Tesis de Posgrado

Fallas de escalabilidad en Internet

Righetti, Claudio Enrique

2000

Tesis presentada para obtener el grado de Doctor en Ciencias de la Computación de la Universidad de Buenos Aires

ıbilidad en Internet

C. Enrique Righetti

Departamento de Ciencias de la Computación
Facultad de Ciencias Exactas y Naturales
Universidad Nacional de Buenos Aires

Disertación entregada como requisito
Parcial para obtener el titulo de
Doctor en Ciencias de la Computación

Director : Leandro Navarro Moldes , Universidad Politécnica de Catalunya

Jurados

Dr. Manel Medina Linares, Universidad Politécnica de Catalunya, España
Dr. Alberto Bartoli, Universidad de Trieste, Italia
Dra. Patricia Borensztejn, Universidad de Buenos Aires, Argentina

28 de Julio del 2000

Agradecimientos

Son muchos anos, mucha gente que hizo posible este día . Creo que debo empezar por Patricia Borensztejn, ella fue quien me contacto con Leandro Navarro. Leandro quien me ofreció su apoyo guía y amistad , junto a su familia Mildred las niñas. A la gente de la UPC , Paco , Isabel, Pau con sus contradicciones , tantos momentos allá en Barcelona.
A todos mis compañeros de doctorado en la UBA , en especial a Juan Santos , Vero, Isa, El ferme , que decir compartimos estos duros momentos .

A mis amigos de todos estos anos que me aguantaron a todos los chicos de Redes , a la gente del INTI , a los del CCC.

Tantos momentos de vacilaciones, de discusiones, a todos mis alumnos que dirigí sus tesis de licenciatura, muy especialmente a Maria Eva Lijding que fue parte de esta búsqueda.

A toda familia , a mi Padre , que al fin entiende a que me dedico , a mi hermana Graciela .

A cada uno de aquellos que me apoyaron , a Irene que allá en el 86 me ofreció ser ayudante en Exactas .

A todos los que este tiempo he dejado de ver .

Se que en este día , alguien estará contento allá en el Cielo .

A Alejandro Rios y Juan Santos que hicieron posible que llegara este día desde la subcomisión de doctorado. A los miembros del jurado .

Por ultimo a " mi familia" a Mariela que me apoyo y brindo todo su Amor , y a mi hijo Nicolas que nunca me imagine tenerlos en estos momentos aquí enfrente míos .

A todos gracias , y al Flaco sin el nada hubiese sido posible.

Buenos Aires , Julio del 2000

En memoria de mi Madre

Abstract

This dissertation presents an Object Distribution System (ODS), a distributed system inspired by the ultra-large scale distribution models used in everyday life (e.g. food or newspapers distribution chains). Beyond traditional mechanisms of approaching information to readers (e.g. caching and mirroring), this system enables the publication, classification and subscription to volumes of objects (e.g. documents, events). Authors submit their contents to publication agents. Classification authorities provide classification schemes to classify objects. Readers subscribe to topics or authors, and retrieve contents from their local delivery agent (like a kiosk or library, with local copies of objects). Object distribution is an independent process where objects circulate asynchronously among distribution agents.

ODS is designed to perform specially well in an increasingly populated, widespread and complex Internet jungle, using weak consistency replication by object distribution, asynchronous replication, and local access to objects by clients. ODS is based on two independent virtual networks, one dedicated to the distribution (replication) of objects and the other to calculate optimized distribution chains to be applied by the first network.

The IRTF (Internet Research Task Force) has stated that resource discovery tools (RDT) have scalability problems. Danzing et. Al. [Dan94] has classified these problems in three dimensions: Data volume, number of users, and RDT diversity.

Our work is focused on providing solutions to the lack of scalability mainly on the first two dimensions. The goal is to provide local access to relevant and pre-selected information; obtaining the best service from the ordered use of global interconnections where bandwidth is scarce, quality is unstable and network partitions occur too often; and providing a global and cooperative mechanism for content classification and qualification (meta-information).

We focus on a model centered on communities or organizations that are producers and consumers of information: they may produce, classify, label, offer and publish information, and also look for and consume information produced by other distant communities. These interactions occur with a local (region, organization) service agent, while object distribution is done asynchronously, reliably and cooperatively among agents located anywhere in Internet. This model is appropriate because intra-community networking is usually adequate meanwhile external networking is usually poor and more expensive.

Contenidos

Capitulo 1

Introducción

El objetivo de este trabajo es el de contribuir en los aspectos relacionados con los fallos de escalabilidad en los sistemas de área global proponiendo además un modelo de referencia cuya arquitectura permita superar dichos fallos. Esto se realiza fundamentalmente mediante un sistema que permite el acceso a la información a comunidades de gran escala. El mismo permite la oferta, suscripción y distribución organizada de contenidos en redes de conectividad débil (Internet). Nos inspiramos en los modelos redes de distribución de la vida real y aplicamos ese paradigma en Internet, logrando un esquema sencillo y novedoso para el acceso a los recursos.

1.1 Motivación

Es indiscutible que la única red Global que permite implementar sistemas de muy gran escala es Internet y que el World Wide Web ("Web") [BCGP92] se ha transformado en la herramienta por excelencia que permite el acceso a los recursos en la misma.

Sin embargo, lamentablemente, la Internet de los 70 está realizando la tarea de transportar los contenidos y proveer la creciente información útil en forma lenta y caótica. Esto es debido a la combinación de diversos elementos a saber: el explosivo crecimiento en la población de usuarios; el tráfico generado por la utilización del Web; la proliferación de un gran número de contenidos de diversa y en muchos casos dudosa calidad .
Otro problema habitual en una red global son las *particiones de red*, producidas básicamente por fallas en un nodo o en un enlace.

Un aspecto importante a considerar es que no sólo hay que tener en cuenta el crecimiento en el número de usuarios, sino también el comportamiento de los mismos. El hecho de que una cantidad importante de usuarios tenga un interés común en un momento dado, provoca un acceso no uniforme a los contenidos de un servidor y, en casos extremos, un fenómeno de

inundación en dicho servidor y en las proximidades de la red, denominado *flash-crowd* [Nie95]. El acceso a los diversos contenidos no está ecualizado.

Ante el gran volumen de información disponible en Internet surge el inconveniente que en la mayoría de los casos no hay garantía alguna de la calidad y confiabilidad de la misma. A excepción de News y algunos catálogos en servidores Web, la mayoría de la información disponible no está clasificada. Otra cuestión a tener en cuenta es cómo el usuario localiza la información de su interés.

El exceso de información y los problemas asociados a la misma, produce una desinformación total. Muchas veces el exceso de información no clasificada y sin garantía de calidad es un problema tan grave como la falta de la misma.

Todos estos factores conllevan a que los múltiples recursos existentes en Internet se encuentren inalcanzables para los usuarios produciendo, de esta forma, frustración en los mismos.

Ante algunas de estas problemáticas muchos plantean la producción de un colapso de Internet [Met96], de no mediar soluciones que puedan ser implementadas en su actual infraestructura.

Estos problemas se manifestaron a nivel mundial y también fueron corroborados en nuestras experiencias realizadas entre la red de la Universidad de Buenos Aires (RedUBA) y la Red Científica de España (RedIris) desde 1994 y posteriormente con la Red de Interconexión Universitaria (RIU) desde 1996.

Durante el desarrollo de los trabajos vinculados con esta disertación, que se iniciaron en 1994, Internet pasó a ser un producto de consumo masivo y se lo presenta a la sociedad como tal. Sin embargo aún existen deficiencias que deben ser solucionadas, si es que se busca lograr la tan mentada "Sociedad de la Información".

1.2 Los probiemas

Los problemas fundamentales que surgen es la falta de escalabilidad que poseen los actuales sistemas de acceso a los recursos en Internet y la conectividad débil que en conjunto presenta. Tanto el WEB (no fue concebido teniendo en cuenta la escalabilidad [1]), como cualquier otra aplicación que permita acceder a la información en Internet deben hacer frente a:

- Escalabilidad
- Funcionamiento ante particiones de red
- Conectividad Débil

1.3 Objetivos

El objetivo de esta disertación es contribuir en los aspectos relacionados con los fallos de escalabilidad en los sistemas de área global y proponer además un modelo de referencia cuya arquitectura permita superar dichos fallos. Dicho objetivo es considerado primordial porque actualmente no existen modelos que escalen correctamente en redes globales. Esto se puede comprobar en la mayor red de área global existente, Internet.

1.4 Contribuciones de nuestro trabajo

El corazón del trabajo presentado en esta disertación es como hacer frente a la escalabilidad y conectividad débil mediante un sistema que mejora el acceso a la información de una comunidad de gran escala, organizando y distribuyendo dicha información entre comunidades de intereses similares.

[1] En realidad, casi ninguna de las aplicaciones fue desarrollada teniendo en cuenta la escalabilidad. Cuando el acceso a la Información en Internet era llevada a cabo mediante FTP y ARCHIE , fue necesaria la replicación de los sitios mas populares a los efectos de sostener su funcionamiento.

En particular la disponibilidad de ciertos contenidos *importantes suscriptos populares* puede ser óptima usando mecanismos de replicación de consistencia débil y (oferta, suscripción y distribución), en redes como Internet de conectividad débil por naturaleza.

En particular las contribuciones son las siguientes:

- Un comprensiva caracterización de las fallas de escalabilidad en Internet.

- ODS un sistema que provee una efectiva replicación de objetos mediante la construcción de una red virtual dinámica sobre Internet. La topología de esta red virtual optimiza el uso de los recursos de la red subyacente, mientras se satisfacen los requerimientos de los usuarios. Muchas redes virtuales pueden crearse y administrarse independientemente. Mediante un modelo de replicación asincrónico se mejora la localidad de la referencia (Objetos replicados disponibles localmente).

- Una taxonomía de los mecanismos de replicación y caching en Internet.

- Limitaciones y Potenciales de dichos mecanismos.

- Determinación del tráfico de Web proxy que permitan determinar cotas de popularidad en documentos Web a los efectos de luego distribuirlos mediante ODS.

- Un modelo simple de ODS con escenarios que muestra como podemos escalar .

Nuestro modelo de referencia esta influenciado por dos principios clásicos en el diseño de sistemas: favorecer la simplicidad sobre una generalidad no garantizada [Lam83] y respetando la funcionalidad de los niveles de punta a punta [Sal84]. Enfatizando la elegancia y la simplicidad, de esta manera logramos funcionalidad buscada con un mínimo de mecanismos y un máximo de claridad [2] [Cor91]

[2] En la conferencia que dio Corbató [uno de los padres de los sistemas de time-sharing] cuando recibió el premio Turing, plantea que todos los sistemas complejos finalmente fallarán y que para tener alguna posibilidad de éxito, es esencial evitar la complejidad y procurar un diseño simple y elegante.

1.4.1 Etapas del trabajo presentado en esta disertación

Durante el desarrollo de los trabajos relacionados con esta disertación la red que fue utilizada a nivel nacional a los efectos de validar los resultados de la misma fueron, la Red de Interconexión Universitaria Red ("RIU") , de la Universidad de Buenos Aires (RedUBA). Y a nivel global con la Red Iris de España.

Si bien existía una experiencia previa en lo que respecta a la utilización de lo que se dio a llamar las redes TCP IP desde el año 1988 [3] [IBM90], básicamente hasta fines de 1993 la red Internet en Argentina consistía de un acceso Internacional de 9,6 Kbps en la Cancilleria Argentina y un par de nodos, uno de ellos en el departamento de computación de la Facultad de Ciencias Exactas de la UBA. La primera etapa de los trabajos de esta disertación comenzó en 1994, durante ese año la UBA contó con un acceso a Internet propio de 64 Kbps. La inmediata utilización del Mosaic para acceder al Web por la comunidad académica, comenzó a plantear los problemas que dieron la motivación al trabajo final presentado en esta disertación.. A nivel nacional en las universidades (RIU) recién en 1997 se logró una conectividad con enlaces de 64 kbps

La problemática presentada era demasiado amplia para se atacada individualmente y como usualmente el trabajo en el área de sistemas es realizado por equipos de trabajo [Cor91], con un numeroso grupo de alumnos que realizaban sus trabajos de licenciatura comenzamos a atacar cuestiones que estaban directa o indirectamente con las problemáticas planteadas.

En aquel momento, comenzamos a implementar una red de distribución de servidores Web basados en la red Usenet[4]. En realidad, mientras desarrollábamos esta aplicación debíamos solucionar constantemente las problemáticas relacionadas con la red subyacente que surgían con nuestra joven Internet.

---

[3] En esa época aun no estaba claro a nivel mundial cual era el modelo red que se utilizaría a nivel global, fundamentalmente en las redes académicas , se debatía entre OSI, BITNET, TCP/IP etc. El panorama en arquitectura de redes de los 80 es el mismo que se presenta en cuanto a un modelo de acceso a la información a fines de los 90.

[4] HT-NN (HyperText - Network News), un mecanismo diseñado para replicar en forma automática y distribuida páginas Web . El sistema propuesto provee un conjunto integrado de herramientas configurables para distribuir información proveniente de diversos repositorios, replicándola en forma masiva a través de Internet y usando la estructura no centralizada de Usenet. Para obtener una red de distribución de recursos Web explotamos las ventajas Para armar esta red el administrador de un servidor Web replicará sus páginas transmitiéndolas a través de un grupo de noticias de Usenet. Todo servidor que se encuentre subscripto a dicho grupo tendrá la posibilidad de capturarlas, incorporándolas a su información disponible.

Si bien muchos de los trabajos realizados no se describen en esta disertación de alguna manera cada uno de ellos permitieron una mejor compresión de toda la Internet. Entre ellos podemos destacar: investigaciones relacionadas con los protocolos de ruteo de nivel de red y la implementación de un nuevo protocolo de ruteo, OP. El motivo fue tratar de comprender sobre qué infraestructura debíamos trabajar y la notable escalabilidad que presenta IP.

Durante estos años también desarrollamos un sistema EROS [SF98] que facilita la recolección de información de los servidores Web para sus posterior indexación y procesamiento de los servidores de búsqueda. Simultáneamente se investigaba si los paradigmas de comunicación en grupos podían ser un modelo para la programación de ODS y en particular los Sistemas de archivos Distribuidos eran sostenibles en la infraestructura de Internet.

Paralelamente sabíamos de la falta de escalabilidad y de la replicación masiva que imponía Usenet, con lo cual se comenzó el desarrollo del primer modelo de referencia de ODS. Simultáneamente se investigaba si los paradigmas de comunicación en grupos y en particular los Sistemas de archivos Distribuidos eran sostenibles en la infraestructura de Internet.

Finalmente, cuando contamos con ODS, nos preguntamos si este modelo propuesto había surgido como un producto de la pobre infraestructura de comunicaciones con que cuenta aun la comunidad académica nacional, o era un modelo válido para toda la "Internet". El incremento de las investigaciones en lo que se ha dado en llamar últimamente Content Delivery y Distribution corrobora de alguna manera que estamos en el camino correcto.

Nuestros estudios del modelo de Internet 2 y la política de telecomunicaciones a nivel mundial nos permiten asegurar que en Internet la conectividad débil estará siempre presente.

Por lo tanto el escenario en el cual se valida esta tesis sigue siendo válido, además nos permite ODS acceder a Contenidos clasificados, calificados y populares en la comunidad que tenemos afinidad.

1.5 Guía de esta disertación

Este capítulo presenta la motivación y etapas de este trabajo, los problemas que estudiamos y una descripción global de la estructura de esta tesis.

El capitulo 2 describe el contexto en el cual se desarrolla el trabajo, fundamentalmente en lo que se refiere a la Red Subyacente (subsistema de comunicaciones) en Internet, Escalabilidad y una visión desde la localización de los recurso.
El capitulo 3 presenta el Sistema de Distribución de objetos ODS y sus diferencias con las técnicas tradicionales que mejoran únicamente la localidad de la referencia.
El capitulo 4 presenta una caracterización de los esquemas de replicación y caching enfatizando en los sistemas de caching y un modelo de caches cooperativo con comunicación multicast. En el capitulo 5 presentamos criterios de diseño basados en invariantes que justifican a ODS y un modelo de escenario que nos brinda una idea de su perfomance. En el capitulo 6 se describen los protocolos de replicación y en particular los de consistencia débil que sirvieron como marco de desarrollo de ODS, así como también los trabajos relacionados con ODS. Finalmente en el Capitulo 7 se desarrollan las conclusiones y los trabajos futuros.

Capitulo 2

Red Subyacente y Escalabilidad de los Sistemas Globales en Internet

2.1 Introducción

En el presente capítulo se presentan los inconvenientes, intrínsecos a su propia naturaleza, que presentan las redes globales y en particular Internet. Todos éstos necesariamente han impulsado el desarrollo de mecanismos que mejoren la localidad de la referencia de los recursos disponibles en Internet, en particular de los recursos Web.

Internet es la red global mas amplia que actualmente existe[5] y se prevé que para el 2001 contará con mas de 200 millones de usuarios.

Cualquier sistema de acceso a los recursos disponibles en Internet necesariamente debe hacer frente a las siguientes cuestiones:

- Escalabilidad
- Particiones de red
- Conectividad Débil

[5] Dejamos de lado la red telefónica PSTN. Sin embargo, en un futuro se prevé que dicha red telefónica va a ser absorbida lentamente por Internet.

2.2. Modelo de Red Subyacente – Características Redes Globales

Si bien la red global por excelencia es Internet, describiremos las características de las redes globales en general. Cualquier aplicación soportada por una red global debería ser diseñada teniendo en cuenta el modelo de red subyacente[6]. Las principales características que definen a las redes globales (WAN)[7] son:

- La administración de la red no centralizada, existen múltiples organizaciones administrativas involucradas [8]
- Alcance geográfico muy extenso y con diferentes husos horarios
- Diversos proveedores de telecomunicaciones involucrados
- Heterogeneidad debido a múltiples equipamientos, diversas plataformas de sistemas operativos y redes
- Gran número de usuarios y recursos
- Particiones de red (debida a fallas de los enlaces, nodos, inconsistencias en las tablas o políticas incorrectas de ruteo)
- Latencia entre nodos de la red elevada y pérdidas de paquetes (Conectividad Débil).

En una red global cualquier aplicación debe enfrentarse con: escalabilidad en todas las dimensiones, latencias, fallas de comunicaciones impredecibles y la semántica de fallas de los nodos. Las particiones de red producen sistemas autónomos o grupos que siguen funcionando en forma separada al resto.

---

[6] Otra área que busca sistemas que se adapten a la red subyacente es la de computación móvil, pero los mecanismos propuestos no tienen como objetivo de máxima la escalabilidad.

[7] Preferimos enunciar las características que la definen , antes de dar una definición de WAN o LAN al estilo de IEEE de años 80. El avance tecnológico a dejado sin sentido tales definiciones, hoy es común enlaces tanto para WAN o LAN cuya velocidad de transmisión es del orden de los Gbps.

[8] Además para ser escalable debe soportar pasar por múltiples dominios administrativos

2.2.1. Modelo de Red Subyacente – Internet

Los estados de la red subyacente que cualquier Herramienta de Acceso a los Recursos ("RDT") debe enfrentar son el resultado de las características intrínsecas de la red y del tráfico generado por dicha RDT, en definitiva, el comportamiento de los usuarios.

El modelo de red subyacente en Internet básicamente es: *En ausencia de fallas, la red es conectada y cada proceso puede comunicarse con cualquier otro proceso. La comunicación entre nodos es punto a punto, asincrónica (con demoras de transmisión no acotadas) y no confiable.*

En el área de Sistemas Distribuidos se han propuesto aplicaciones que tienen soluciones bien conocidas para este modelo pero con uno o dos órdenes de magnitud en la escalabilidad.

Los estados de la red subyacente a su vez también dependen del comportamiento de los usuarios, consecuencia de los patrones de tráfico por ellos generado. El comportamiento de los usuarios lo incluimos en la escalabilidad.

Es el subsistema de comunicaciones con que cuentan las aplicaciones el que permite ver a Internet como un grafo conexo, en consecuencia la estabilidad de los enlaces punta a punta es un factor crítico. Dicha estabilidad depende de: sistemas de conmutación de telecomunicaciones subyacentes; componentes de retransmisión de paquetes y la arquitectura de ruteo. Actualmente, la interacción de todos estos componentes es pobremente comprendida [LAJ98], sin embargo a los efectos de nuestro trabajo se ha observado que :

Las Particiones de redes en Internet presentan las particularidad que cuando dos nodos pueden no comunicarse entre si es posible que ambos puede comunicarse con algún otro nodo en común Ej. : FTP de A a C no es posible, pero puedo realizar un telnet a B y de ahí un FTP a C).

La probabilidad que existan particiones es : $P_p \rightarrow 1$, [Gol92a]

Tráfico asimétrico [Pax96]

2.2.1.1. Conectividad débil

Denominamos enlaces de conectividad débil a aquellos que presentan un alta latencia y alta pérdida de paquetes mientras que aquellos que presentan baja tasa de pérdida de paquetes y baja latencia los denominamos de conectividad fuerte.

Existe una latencia inherente entre los nodos [9] de una red global. En el caso de Internet esta latencia se ve incrementada por la congestión causada por el ancho de banda insuficiente producto del:

- Incremento en el número de usuarios que provoca un importante aumento en el tráfico.
- Tipos de objetos; multimedia

Berners-Lee argumentó que un tiempo razonable de latencia es del orden de los 100 mseg [Ber95] [10]; sin embargo, Viles y French encontraron que el mismo está en el orden de los 500 mseg [VF95]. Nuestras mediciones reflejan que actualmente el panorama es más desalentador, encontramos latencias promedio a nivel de red de 2000 ms entre las redes académicas de España y Argentina. A su vez, en el mejor de los casos (horario diurno) la mayoría de los proveedores comerciales en Argentina de acceso a Internet (ISPs) obtienen una RTT del orden de los 650 ms ya que la mayoría del tráfico a USA utiliza satélite y con varios saltos entre routers.

La latencia a nivel de aplicación se incrementa con la pérdidas de paquetes [11]. Se ha planteado que son comunes las pérdidas de paquetes en el orden del 40% o más [Gol92], sin embargo nosotros hemos encontrado valores entre el 60% y 70% (Octubre-Noviembre/95, enero a febrero 97) en los horarios de mayor carga. Metcalfe[12] 3 a 4% media en 24horas y horas de carga >30%. Diversas experiencias que realizamos en los horarios de mayor tráfico hacen casi imposible el acceso a recursos mediante HTTP o FTP.

---

[9] Solo de velocidad de propagación en una Fibra Óptica tenemos 5ms cada 1000km.

[10] En red vBNS enlaces no saturados OC3/OC12, 155Mbps/622Mbbps tienen aproximadamente 40ms de RTT. http:www.internet2.edu

[11] Si bien TCP no fue diseñado para perdidas de paquetes mayores al 30% , es robusto y continua trabajando pero no en forma optima . Actualmente el IETF esta trabajando en una idea de ACK selectivo (actualmente el ACK es acumulativo , si transmito N segmentos y pierdo uno debo retransmitir los N).

[12] www.computer.org/internet/9702/metcalfe9702.htm

2.2.1.2. Ancho de Banda

En cuanto a la conectividad débil, se podría suponer que el aumento del ancho de banda es una solución pero existen varios problemas: John Sidgemore, director ejecutivo de UUnet, afirmó que ninguna otra tecnología en la historia creció tan rápido como Internet al decir: "La demanda de ancho de banda de Internet es diez veces mayor cada año". A esto hay que sumar lo expuesto por la denominada ley de Andy Grove, CEO de Intel,: "El ancho de banda se duplica cada cien años". Si bien la ley de Grove expresa un posición un tanto extrema, si uno la analiza con detalle y su ámbito de aplicación es toda una red global, como Internet, esto no es tan descabellado. Es verdad que por ejemplo el backbone de NSFnet pasó de T1 (1,5Mbps) a T3 (45 Mbps) en su momento, pero analizando Internet como un todo siempre encontramos enlaces de ancho de banda limitado.

En EEUU existen iniciativas como Internet 2 (Con velocidades de enlaces OC32, 2.4Gbps) o en Europa TEN 155 con un notable incremento en los ancho de banda usados. Sin embargo, la arquitectura propuesta asegura ancho de banda para las aplicaciones críticas tales como voz (VoIP), aplicaciones *Premium*, realidad virtual etc., dejando la política de best-effort para el tráfico por Ej. HTTP.

La perspectiva de crecimiento por ejemplo en la región de Latinoamérica es interconectar las redes académicas y científicas mediante enlaces de 256 Kbps.

Todo esto no son sino islas aisladas de conectividad fuerte ya que las demás redes se encuentran conectadas débilmente, o tal vez regiones con islas con conectividad débil.

- *En definitiva tenemos siempre una red con conectividad débil y un ancho de banda insuficiente.*

2.2.1.3. Características del tráfico en Internet

Si tenemos en cuenta que existen 7.000.000 de servidores Web, julio 1999, podremos comprender que el tráfico total HTTP en enlaces de los backbones de wide-área de algunas

áreas de USA llega al 80%[13] del tráfico total. En los enlaces de nuestros backbones nacionales este tráfico supera el 87%[14]. Con lo cual el tráfico hoy esta determinado por el Web.

Pruebas estadísticas sobre datos reales sustentan la hipótesis que en Internet el tráfico es asintoticamente *self-similar* [15]. En particular el tráfico HTTP puede ser atribuido al comportamiento periódico de los usuarios Web, el protocolo, la estructura de los documentos en suma con la distribución de tamaño de los archivos. Dicho comportamiento periódico aparece en intervalos de minutos a semanas [Abd88].

2.3. Escalabilidad

La escalabilidad en forma genérica puede definirse como la habilidad de incrementar el tamaño del dominio de un problema con un pequeño o insignificante incremento en el tiempo de solución y complejidad espacial. Con lo cual la solución de un problema es escalable si continúa siéndola (o trabajando) aun cuando alguna variable del problema cambia a un valor usualmente grande. Es normal que una solución que trabaja bien en un *pequeño* problema se vuelva impracticable cuando se escala a gran tamaño. Para lograr escalabilidad el impacto del crecimiento de las variables debe minimizarse.

Es conocido el efecto de la escalabilidad en los algoritmos. Una búsqueda lineal en una lista trabaja bien si es pequeña. Si hay un gran número de elementos una búsqueda binaria a través de un árbol ordenado balanceado es mas escalable. O por ejemplo el modelo de señales débiles para dispositivos semiconductores, si bien ese modelo es suficiente para esas condiciónes de contorno.

Se consideran exitosos generalmente a aquellos sistemas que escalen uno o dos órdenes de magnitud [BLN82]. Sin embargo los órdenes de magnitud en Internet superan ampliamente a cualquiera de los sistemas implementados en las áreas de bases de datos, sistemas distribuidos o sistemas de archivos distribuidos.

Si asumimos escalabilidad en función de número de usuarios en un sistema, dicho sistema tendrá un costo a lo sumo de adición de recursos $O(n)$ y una degradación de la performance $O(\log n)$.

---

[13] http://squid.nlanr.net(web port 80 0,1% tráfico del backbone de la NSFnet marzo 93). [historia WWW en w3.org]
[14] http://WWW.riu.edu.ar.
[15] Con parámetro de Hurst , H = 0.8 , para un proceso de Poisson se demuestra que H=0.5. En [Abd98] entre 0.59 y 0.94

2.3.1. Escalabilidad en los sistemas de área global

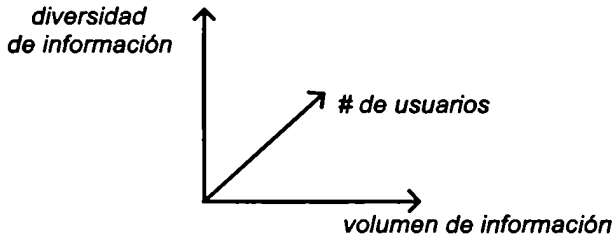
El Web en un principio era escalable (uno dos órdenes de magnitud), pero cuando las variables pasaron a dominios grandes comenzaron los problemas. Tanto que algunos irónicamente comenzaron a denominarlo World Wide Wait.
El Web es un sistema que utiliza componentes escalables: clientes, servidores, TCP/IP y DNS [Moc87], pero el sistema como un todo tiene serios problemas de escalabilidad.

Usualmente, la descripción de un protocolo ignora dónde existen probabilidades que haya un gran valor. Una solución que trabaja bien para un problema, puede tornarse impracticable cuando se escala a un gran tamaño.
Podría ser que un sistema que no es escalable pueda ser reemplazado, sus datos convertidos, y los usuarios reeducados para hacer uso de una nueva solución escalable. Pero esto sería actualmente casi imposible de realizar con los usuarios de Internet (o al menos Web). Desde 1994 se han comenzado a aplicar las técnicas tradicionales de escalamiento de sistemas: replicación y caching, con la particularidad y complejidad que impone Internet.

Nuestro problema es el acceso de recursos y producción por miles de comunidades en Internet., si nosotros podemos anticipar la necesidad para soluciones que son escalables con respecto a una variable conocida, esto puede ser importante para diseñar como corresponda el sistema.

La escalabilidad en Web es la habilidad par incrementar en número de servers, clientes, usuarios, tipos de datos, tamaño de los mismos y la habilidad de manejar servers desparramados en amplias localizaciones geográficas, con un mínimo de cambio en la QoS.

Los esfuerzos del IRTF [Dan94] este trabajo es examinar el impacto de la escalabilidad y colocar los problemas resultantes en un framework adecuado . Generan un marco donde plantean tres niveles conceptuales :



Dimensión	Nivel Conceptual	Problemas	Investigaciones
Volumen de datos	Interface de Información	Sobrecarga de información	Especialización de los tópicos ; Scalable content-indexing
Usuarios	Dispersión de Información	Replicaron Insuficiente ; Topología de distribución manual	Replicaron Masiva Estadísticas de acceso; Caching de objetos
Diversidad de Datos	Gathering de Información	Extracción de datos baja calidad de los datos .	Mapeo de Operaciones; Mapeo de datos .

2.4. Entorno de Trabajo

La red de Interconexión Universitaria (RIU) ha sido caso de estudio y el entorno de nuestro trabajo. La misma fue la primera red Internet de alcance nacional en Argentina e interconecta a 33 universidades nacionales. Topológicamente esta formada por una anillo de cuatro Universidades, interconectadas con enlaces de 128 Kbps y con accesos Internacionales cada uno de 128 Kbps y uno de 2 Mbps. A su vez las restantes universidades se conectan mediante enlaces de 64 kbps a cada uno de estos nodos. La misma permitía el acceso, en mayo de 1999, a 70.000 usuarios y contaba con unos 250 servidores Web. Un aspecto importante es que el tráfico Internacional representa el 85 % del tráfico total del backbone nacional.

Resulta interesante observar la reacción frente al aumento de ancho de banda del enlace internacional [16] . En el gráfico de la figura 1 podemos ver la utilización del ancho de banda de uno de los enlaces internacionales de la RIU. Claramente se aprecia que el mismo se encuentra desde las 8 a las 20hs totalmente saturado, no resultando el tráfico ecualizado.

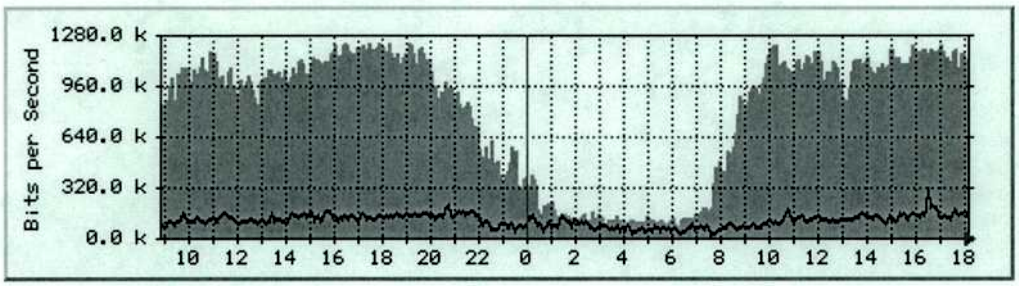


Fig1

En el gráfico de la figura 2 se muestra el momento en que se incrementó la capacidad de dicho enlace en agosto de 1998.

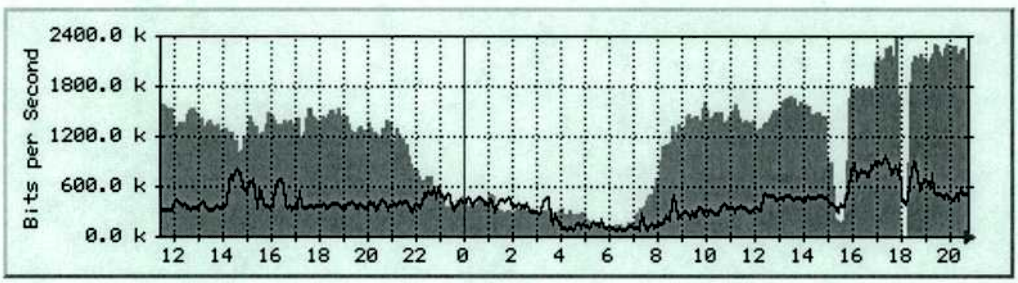
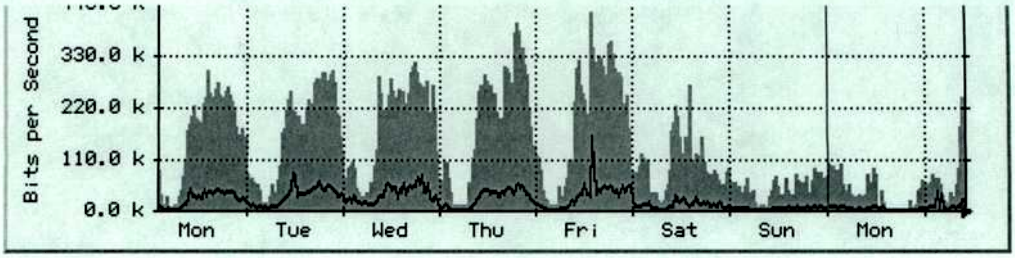


Fig. 2

[16] Denominamos así al enlace hacia el mayor exportador de tráfico IP, USA.



- *En definitiva por mas que se aumente el ancho de banda, éste inmediatamente será ocupado*

Hemos encontrado un comportamiento de tráfico *self-similar*, en períodos de horas, días y semanas. Este comportamiento lo hemos verificado en la RIU durante 1998. Asimismo, hemos encontrado evidencias de self similaridad en el tráfico de unos de los más grandes ISP de Argentina en el periodo julio-noviembre 1999.

En el caso particular de un ISP, accesos Dial-up, mejor de los casos 5Kbps (2Kbytes, 3 Kbytes) con lo cual existe una limitación (Modems V.90 , 56Kbps) en el consumo de ancho de banda. En el caso de la RIU o RedUBA el aumento de ancho de banda es ocupado por el tráfico de los usuarios tal la expansión de un gas.

2.5. Localización de los Recursos

Podemos ver las problemáticas planteadas desde otra visión , de que manera localizamos un recurso en Internet.
Actualmente en Internet, un recurso (cualquier objeto al que se puede acceder o hacer referencia en una red [DE94]), puede ser accedido si se conoce su ubicación dentro de un host y su forma de acceso. Estos identificadores reciben el nombre de URLs [BMM94], Uniform Resource Locators. Los URLs son ampliamente usados, pero tienen básicamente tres problemas están ligados : nombres de hosts[1] , nombres de archivos en un host en particular y protocolos de acceso[2]

Con lo cual tenemos un mapeo uno a uno entre nombre y una única copia física de un recurso. Los nombres de hosts imponen dos problemas: *estabilidad y escalabilidad.*

---

[1] host: máquina destinada a correr programas de usuario (aplicaciones) en una red [Tan91].
[2] Ej. http, news, ftp , gopher, etc.

Además, un recurso que pertenece o que es mantenido por una organización puede migrar a otra organización, necesitando por lo tanto una actualización, dentro del URL, acerca del nuevo nombre de host en el cual el recurso está alojado. Todos estos cambios invalidan a los viejos URLs.

Dado que la única forma de acceder a un recurso nombrado por un URL es contactar al host cuyo nombre aparece en el URL, es difícil proveer un acceso escalable a dichos recursos.

Los nombres de archivos embebidos en URLs sirven a dos propósitos que están en conflicto. Por un lado necesitan ser estables para que una referencia a un recurso nombrado continúe siendo válida tanto tiempo como sea necesario. Por otra parte, los archivos usualmente están organizados en jerarquías para facilitar a los usuarios su búsqueda a través del file system[3]; no obstante las jerarquías necesitan cambiar continuamente a medida que se agregan archivos nuevos y que se borran otros. Cuando la jerarquía del file system cambia, los viejos URLs dejan de ser válidos.

El hecho de que el URL lleve implícito el protocolo de acceso al recurso impone una barrera adicional: el cliente debe tener soporte para el protocolo de acceso para poder acceder al recurso. Inclusive, aunque se contara con el soporte para cualquier protocolo de acceso, se necesitarían conocer de antemano todos los URLs del recurso, para poder elegir la forma de acceder al mismo.

La alternativa es proveer recursos replicados , antes que preservar la ilusión de un única copia visible . Con el objeto de desarrollar un esquema abstracto de nombres independiente de la locación es que diversos grupos de investigación comenzaron a trabajar sobre la problemática planteada. En particular, uno de los trabajos que se llevo a cabo es aquel realizado por el URI Working Group[4] , siendo URI el nombre que recibe según dicho grupo, un identificador genérico en Internet. El Grupo de trabajo URI [17] definió a los URIs como la unión de los URLs (Uniform Resource Locators), URNs (Uniform Resource Names) [WD94] y URCs (Uniform Resource Characteristics). Estos identificadores fueron pensados para trabajar conjuntamente en un esquema general para nombrar, describir y obtener recursos en Internet.

---

[3] Conjunto de programas del sistema operativo cuya función es la administración de archivos.

[4] El URI Working Group es un grupo de investigación encargado de definir un conjunto de estándares que sirvan de marco de trabajo a otros investigadores que implementen servicios de información para la localización e identificación de recursos.

[17] "Uniform Resource Identifiers (URI) Charter".<http://www.ietf.cnri.reston.va.us/uri-charter.html.

El propósito de los URNs es el de proveer un nombre persistente para los recursos en la red. Fue pensado para usarse en conjunto con un servicio de directorio que provea un mapeo de URN a URL . La arquitectura URN-URL permite referencias permanentes a los recursos sin importar su ubicación actual en la red.

Un recurso tendrá sólo un identificador a lo largo de su existencia, cuyo significado deberá ser el mismo en cualquier lugar donde se lo referencie y que continuará siendo válido ante la aparición de nuevos protocolos y tecnologías.

El esquema propuesto el URN se mapea con el URC (la meta información del recurso) junto a los URL´s de las copias del mismo.

Si bien se definieron claramente los URN y URC , como se realiza la replicación de recursos no estaba dentro de sus objetivos y que esquema implementar para la resolución no se avanzo casi nada salvo implementaciones parciales.

En [BCR+99] presentamos un modelo que independizará al usuario de tener que conocer los URLs de los Objetos de ODS asociado , al utilizar como identificador del mismo un URN en lugar del URL. Desde ya las cadenas de distribución nos aseguraran la transparencia ante la migración .

2.6. Conclusiones

Cualquier sistema que permita acceder a los recursos en Internet debe tener en cuenta las características intrínsecas de la red subyacente y las producidas por el comportamiento de los usuarios a saber:

- El acceso a los recursos en el tiempo no esta ecualizado
- Los fenómenos de Flash crowd suceden
- Particiones de redes siempre suceden
- Conectividad débil siempre presente, producto en muchos casos de un ancho de banda limitado.

Por ultimo, en cuanto a la cantidad de usuarios que generan información, en muchos casos de escasa calidad , y limitada capacidad humana de producción y acceso a los recursos hace que se genere un volumen de contenidos casi caótico.

Frente a todas estas cuestiones se deben desarrollar mecanismos que mejoren la localidad de la referencia de los recursos en Internet y que brinden un orden y garantía de calidad en el espacio de recursos disponibles.

Capitulo 3

Un Sistema de Distribución de Gran Escala

3.1. Introducción

Presentamos en este capítulo nuestro *Sistema de Distribución de Objetos (ODS).* El objetivo de ODS es proveer en Internet mecanismos de distribución a semejanza de los utilizados en la vida cotidiana (ej. distribución de productos alimenticios, diarios, libros etc.), con el fin de hacer frente a las cuestiones que hemos analizado en el capítulo 2.
El sistema cuenta con un mecanismo de ruteo, mecanismos de replicación de consistencia débil y de indirección que le permite escalar correctamente en una red de área global y mejorar la localidad de la referencia de los recursos. Es por esto que ODS está diseñada para funcionar especialmente bien en la cada vez más poblada, extendida y compleja jungla de Internet.

Está formado por dos redes virtuales independientes que funcionan en Internet. Una dedicada a la distribución de objetos (replicación selectiva) y la otra dedicada a construir los canales de distribución óptimos [18] de acuerdo a intereses comunes de las comunidades que utilizan ODS. Debido a que en Internet algunos sitios frecuentemente se vuelven inaccesibles, ya sea por la latencia entre los nodos de la red, particiones de red o el fenómeno flash-crowd, los objetos de nuestro sistema se acceden localmente y se actualizan fuera de línea. El sistema también provee métodos para la clasificación de los objetos. Esto permite implementar mecanismos de distribución selectiva y brindar orden en el caos reinante actualmente en Internet.

Las cadenas de distribución se construyen en forma dinámica para proveer a los usuarios finales de los objetos que desean consumir optimizando el uso de los recursos de la subred disponibles. Estas cadenas de distribución se arman utilizando protocolos de ruteo a nivel de

[18] El concepto de optimalidad en Internet es muy variado, asumimos el que nos permite una distribución eficiente en función del estado de la red subyacente.

aplicación. La distribución de objetos es un proceso independiente donde los mismos circulan asincrónicamente entre los agentes de distribución (SA).
Organizando las réplicas en grupos limitamos el tamaño del estado de consistencia que cada réplica mantiene minimizando, de esa forma, el tiempo necesario para alcanzar un estado de consistencia.

Presentamos finalmente un modelo de referencia de ODS, llamado *Red de Distribución de Documentos (DDN)*. Los autores presentan sus contenidos a los agentes de publicación, los cuales se basan en los esquemas propuestos por las *Autoridades de Clasificación* para clasificarlos. Los lectores se suscriben a *tópicos* o *autores*, y acceden a los contenidos desde su agente local de distribución (igual que un Kiosco o biblioteca, con copias locales de los objetos).

3.2. Motivación

En los últimos años Internet creció en forma exponencial en cantidad de usuarios y servidores de recursos y se prevé que este fenómeno seguirá sucediendo. Sin embargo, la infraestructura global de comunicaciones no acompaña en forma equilibrada dicho crecimiento. A esto hay que sumarle la aplicación incorrecta de políticas de ruteo entre sistemas autónomos[19].

Este crecimiento provoca un constante e importante aumento en el tráfico experimentando los usuarios un tiempo de latencia muy grande en el acceso a los recursos.

En particular se estima [Sch90] que los tiempos de repuesta de los sistemas de información deben ser menores a los 1000 mseg, para consultas que presentan información nueva y mucho menor para consultas que requieran detalles adiciónales.

No sólo hay que tener en cuenta el crecimiento en el número de usuarios, sino también el *comportamiento* de los mismos. El hecho de que una cantidad importante de usuarios tenga un interés común en un momento dado, provoca un fenómeno de inundación en un servidor de recursos y en las proximidades de la red, denominado *flash-crowd* [Nie95]. El fenómeno se presenta cuando aparece un recurso que miles de usuarios tratan de acceder simultáneamente (ej. 880.000 accesos a los servidores Web [Ber92] de la NASA durante el choque del cometa Levy-Schoemaker-9).

[19] Grupo de Routers bajo una única política de administración de ruteo

La conjunción de todos estos elementos tornan inalcanzables o de difícil alcance a los recursos, produciendo frustración en los usuarios.

Ante estas problemáticas muchos plantean que se produzca un "colapso en Internet" [Met96], de no mediar soluciones que puedan ser implementadas en la actual infraestructura de Internet.

Otro de los problemas es el gran volumen de información disponible en Internet. Generalmente no hay garantía alguna de la calidad y confiabilidad de la misma, y la información ruido tiende a ocultar la información útil (e.g. en el sistema de noticias News [Ada87] se considera que más del 90% de los contenidos de los newsgroups es ruido [Sal92]). En forma genérica el alto ruido para la información útil es solamente en una búsqueda por tópicos y no en una lectura sistemática. La información en Internet no está clasificada en base a algún criterio de calidad [20] (News y algunos catálogos en servidores Web proveen algún grado de clasificación). Esto lleva a otro problema: cómo localizar la información relevante para el usuario (o como encontrar lectores relevantes para una información dada). El exceso de información y los problemas asociados, produce una desinformación total. Muchas veces el exceso de información es un problema tan grave como la falta de la misma.

Nuestro trabajo trata de solucionar esta problemática proponiendo un sistema de clasificación y autoridades de publicación .

El IRTF (Internet Research Task Force) ha planteado que las herramientas de descubrimiento de recursos (RDT) (e.g. WAIS [Kah90], Gopher [Ank93], Web, ARCHIE [ED92], Prospero [Neu92], etc.) tienen fallas de escalabilidad. Esta escalabilidad la han representado Dantzig et al. [BDM+94] en tres dimensiones con el objetivo de situar los problemas en un framework adecuado: *Volúmen de Datos, Número de Usuarios y Diversidad de RDT* [21].

---

[20] Tim Berners-Lee, inventor de la World Wide Web, responsabilizo a los usuarios de que la Web sea segura: "Es importante tomar conciencia de que nosotros hacemos la Web. Somos los que leemos, los que enseñamos a los niños a navegar en la Web, los que ponemos información en la Web y, en especial, los que conectamos unos sitios con otros a traves de links... La Web no te obliga a hacer nada. Si a Uds. les preocupa que sus hijos lean información de baja calidad, entonces enséñenles que deben leer y de que manera juzgar esa información." Scientific American Dec 97

[21] En la época que se planteo marco de trabajo el uso de RDT que permitían Navegación gopher , ftp , WWW etc y los que permitían búsqueda WAIS , ARCHIE, VERONICA etc era muy pronunciado , hoy una única interfaz Web se navega y se realiza la búsqueda.

A la dimensión usuarios le corresponde el nivel de "diseminación de la información". En este ámbito vemos que las investigaciones están orientadas a: *cache de objetos, estadísticas de acceso y replicación masiva.*

Nuestro trabajo trata de dar una solución a la falta de escalabilidad de las dos primeras dimensiones y asegurar la disponibilidad de los recursos ante particiones de red, para ello, principalmente proponemos la replicación selectiva basada en grupos de interés (esquemas de clasificación) y mecanismos de oferta/demanda.

El objetivo es proveer acceso local a información relevante y preseleccionada; obteniendo el mejor servicio para las interconexiones globales donde el ancho de banda es escaso, la calidad inestable, las particiones de red ocurren frecuentemente y los costos de conexión son elevados.

Nosotros nos enfocamos en un modelo centrado en comunidades u organizaciones que producen y consumen información: ellos pueden producir, clasificar, rotular, ofrecer y publicar información. También consumen y distribuyen información de otras comunidades distantes. Esas interacciones ocurren con un agente de servicio local (región, organización), mientras las distribución de objetos es realizada asincrónicamente, confiablemente y cooperativamente entre agentes localizados en cualquier lugar de Internet.

El modelo es apropiado porque la red subyacente dentro de una comunidad es adecuada (conectividad fuerte) mientras que interconexión externa es usualmente pobre (conectividad débil) y mas costosa. ODS permite la distribución selectiva, suscripción y notificación brindando orden y economía en el caótico y redundante tráfico de la actual Internet.

Las cadenas de distribución son construidas dinámicamente para encontrar la manera mas efectiva de proveer a los usuarios finales con una replica de los objetos que ellos buscan obtener, mientras se realiza un buen uso de los siempre limitados recursos (fundamentalmente de los enlaces WAN). Lectores y autores tienen un servicio especializado en seleccionar y distribuir económicamente la información en gran escala.

Los objetos en nuestro sistema son accedidos en forma mas robusta en una región local y actualizados asincrónicamente (políticas: en batch, en horas de bajo tráfico, reintentando ante fallas, etc.). ODS mantiene metainformación sobre los objetos, incluyendo información sobre su clasificación (bajo que esquema). En definitiva la metainformación provee a los usuarios

de notificación sobre los nuevos objetos (evento: oferta) en su área de interés a un costo mínimo.

3.3. Mecanismos para acercar los contenidos a los usuarios

Actualmente es indiscutible tomar el Web como paradigma de un sistema de acceso a recursos usado por un número muy grande de usuarios, que maneja una gran variedad de objetos. El mismo es responsable de la mayor cantidad del tráfico de Internet.
Analizaremos entonces los dos mecanismos mas populares[22] para mejorar el acceso a la información en el Web (acercando ésta a los usuarios y en consecuencia reduciendo las transferencias sobre enlaces entre backbones de Internet): *caching y mirroring.*

3.3.1. Caching

Un cache es una memoria intermedia mas lenta que el registro de un procesador y mas rápida que la memoria principal o discos. Caching explota la *localidad de la referencia*: se ahorra tiempo cada vez que el cache provee los datos salvando accesos a dispositivos de almacenamiento mas costosos. Necesita de políticas de reemplazo para decidir el mejor contenido de la cache y sufre de los problemas de *consistencia* en caso de cambios en el repositorio original.

Generalmente en el Web el cache se realiza en la memoria del cliente, en su disco o en un repositorio cache compartido en la cercanía del cliente: un proxy cache[23] [LA94]. Todas las solicitudes "hacia el mundo exterior" son enviadas y respondidas por el proxy. Algunas son resueltas localmente (cuando recientemente alguien hizo la misma solicitud) y otras son redireccionadas a la fuente. Todas las solicitudes deben ser validadas frente a la copia original.
El caching ahorra ancho de banda y mejora la latencia del acceso a los objetos solicitados por los clientes, pero presenta los siguientes inconvenientes:

[22] A mediados de 1999 surgieron una serie de iniciativas comerciales que implementan redes de caches, dichas redes se han dado en llamar actualmente Redes de Distribución de Contenidos , CDNs .
[23] Por cuestiones históricas existe la denominación proxy cache, en este capitulo usaremos proxy cache o cache en forma indistinta. El concepto de proxy cache implica la configuración manual en el cliente del usuario , en cambio un sistema de caching asume que se realiza mediante diversos mecanismos la redirección de las solicitudes HTTP a el para actuar como proxy (intermediario) con los servidores Web exteriores.

a) ante particiones de red: esto es debido a que la verificación de la consistencia ocurre sincrónicamente con las solicitudes de los clientes (se debe realizar previamente una validación de la copia que posee el cache con el objeto original).
b) la latencia o falla ante cache misses, cuando un documento tiene que ser recuperado sincrónicamente de la fuente original. (Un problema importante es que el primer usuario que solicita un documento debe soportar la latencia de la red en ese momento[24]) .
c) no redistribuye el acceso a la red: la carga de la red se mantiene siguiendo las preferencias temporales de los usuarios (horas picos).
d) las técnicas de cache fallan al manejar grandes volúmenes de información (políticas de reemplazo)
e) cuando la información se modifica rápidamente (consistencia de la cache).

En definitiva los clientes son afectados ante particiones de red en cada solicitud, y de la latencia ante cache misses. Si bien se han propuesto formas más eficientes de caching (e.g. geographical push cashing [GS94], demand based dissemination [Bes95], cache cooperativo [MLB95]) todos ellos adolecen de los problemas ya mencionados.

Además, el funcionamiento del cache, solicitudes sincrónicas de los clientes, impone una fuerte carga a los servidores. Usualmente quedaban pendientes varias conexiones del protocolo HTTP 1.0 [25][Ber94], que abre una conexión TCP [Pos81b] para cada archivo dentro de un documento html, solicitado por cada usuario. Si bien actualmente se utiliza HTTP 1.1 que abre una única conexión TCP los órdenes de magnitud en los cuales crecieron los usuarios mantiene fuerte carga en lo servidores por manejo de las conexiones TCP .

3.3.2. Replicación (Mirroring)

El objetivo de la replicación es el de incrementar la disponibilidad de los datos y reducir (balanceo) la carga en las maquinas servidoras de documentos. Los usuarios deben usar y conocer los sitios mirror cercanos. Si son usados, la latencia de acceso de los clientes y el fenómeno de flash-crowd disminuyen, mientras que el tráfico es ecualizado en el tiempo.
El mirroring consiste en tomar una copia exacta de un área FTP remota y presentarla como un área FTP local, usando en muchos casos un software que periódicamente chequea actualizaciones [McLo93] (ej. la base de datos de Netfind que se replica globalmente mediante esta técnica [Sch94]).

[24] Al menos a la fecha existe un cache, CacheFlow, que minimizaría esta demora .
[25] Aunque este problema se busca minimizar con HTTP 1.1

Sin embargo la utilización de sitios espejados plantea diversos inconvenientes:

a) es necesario mantener consistencia entre el servidor original y sus mirrors: la mayoría de los documentos son read-only, y los cambios se producen en los sitios originales por el autor del documento. Los usuarios usualmente no confían en el origen y actualidad de la información. Ellos necesitan conocer las políticas de actualización de cada sitio mirror y la historia de cambios de los documentos de su interés.
   En [Bae97] se propone una solución parcial: si un *proxy client-side* redirecciónа un solicitud HTTP a un servidor dado, una lista de servidores replicados (espejados) pueden incluirse en un header especial de HTTP en la respuesta. Esta información puede ser usada en forma transparente en pedidos futuros, o presentarse al usuario cuando él decida.

b) Los usuarios ignoran (o no recuerdan) la existencia de muchos sitios mirror y muchas veces no pueden decidir cuál es el mejor (mas cercano y/o menos cargado). Ellos deberían conocer la calidad del servicio (y política de update , etc.) de un sitio mirror site antes de decidir cuál mirror (u original) es el mejor sitio para acceder.

c) El mantenimiento de sitios espejados no restringe el acceso de los usuarios al sitio original, por lo anteriormente mencionado los usuarios continúan accediendo al sitio original.

d) La replicación se realiza en forma centralizada y se configura manualmente generando gran complejidad en la administración.

Existen otros utilitarios que realizan mirroring de recursos Web, tales como WebCopy[26] y WebWhacker[27]. Todos permiten copiar información remota en forma local, la cual luego puede consultarse sin necesidad de estar conectados a la red. La mayoría de los browsers actuales permiten navegar a través de esta información almacenada localmente. Esta técnica se utiliza para disminuir el costo de las comunicaciones, para realizar demostraciones o con fines educativos ya que a diferencia de FTP mirror, ésta es una copia no autorizada.

[26] Copyright 1994, 1995 by Victor Parada. Copia archivos en forma recursiva vía HTTP.
[27] Copyright © 1995-97 CNET, Inc. All rights reserved. Permite la copia de páginas Web al disco de la PC.

3.4. Acceso a los Recursos

Cómo los usuarios acceden[28] a los recursos es de vital importancia para el desarrollo de cualquier sistema. Las herramientas de acceso a los recursos desarrolladas en Internet pueden ser clasificadas según la forma en que "diseminan" a los mismos [Wei94] en:

- *Ir y conseguir:* Es la técnica más popular en donde hay una única fuente original de información; ejemplos de ello son FTP [Pos85], Gopher, Web, etc. . Esta técnica produce una gran pérdida de tiempo al usuario, como combinación de latencia en el acceso y el exceso de información (relevante y ruido). Son sistemas basados en el modelo cliente-servidor.

- *Enviar a todos lados:* Creemos más conveniente llamar a esta técnica "enviar a todos los miembros del grupo". Podemos clasificar en esta categoría a News, Mbone [Eri94], Harvest [BDH94], Hyper_G [KAM95], etc. Son sistemas distribuidos, en los que distintos agentes cooperan para replicar/distribuir la información. Un punto importante en esta técnica es la topología utilizada para la difusión de la información y cómo se construye.

3.4.1. Diseminación Selectiva de Contenidos

Dado el volumen de información existente en Internet es necesario adoptar Sistemas de Distribución de Contenidos Selectivos.
El concepto de la distribución de información selectiva (DIS) surge en los sistemas de computación centralizados [Sal68] en los cuales los documentos son filtrados en función de algún profile definido por el usuario. Mientras que en [LT92] se plantean filtros más efectivos, García et. al. [Gar94] proponen una forma de replicar y distribuir entre servidores SDI, profiles y documentos de solución intermedia en el número de profiles y de documentos. Basan el manejo de la replicación de datos en un algoritmo de quórum [29].

[28] Aquí el concepto de acceder esta relacionado con el de distribución de contenido , estrictamente hablando los usuarios interactúan con una RDT mediante dos paradigmas , navegación y búsqueda Obr93 , Bow94.

[29] Los principales protocolos de control basados en quórum son: consenso mayoritario protocolo de Grilla; protocolo de árbol; y protocolos jerárquicos.

En realidad se han centrado en un sistema que permita el filtrado de la información [30]. El mismo fue pensado para que escale correctamente (*el método de indexación*) frente al volumen de información y números de usuarios.

En ODS, si bien se realiza una distribución selectiva, la misma no es producto del filtrado de información. En cada cadena de distribución circulan únicamente los documentos relacionados con el grupo de interés de la comunidad de agentes de servicios intervinientes. Además, la distribución (replicación) se hace efectiva cuando se acepta la oferta de determinado objeto, con lo cual la granularidad es a nivel de objeto.

3.5. Red de distribución de contenidos

El modelo propuesto, ODS, enfrenta la problemática planteada utilizando como referencia los modelos de distribución existentes en la sociedad (ej. cadenas de distribución de productos alimenticios, publicaciones, etc.). En la vida cotidiana, los consumidores no van a los lugares donde se producen los productos (e.g. fabricas, casa del autor, etc.), sino que los compran en el negocio minorista más cercano. Los mismos obtienen productos que ya se encuentran en los negocios, no esperan a que estos lleguen. Aunque la fábrica esté produciendo nueva mercadería constantemente, los consumidores compran la que ya ha llegado a los negocios, de otra forma nunca comprarían nada. Este sistema funciona porque los consumidores confían en los negocios minoristas, pues creen que los productos serán lo más frescos (o actuales) que sea posible [31], pagando por los mismos un precio razonable. Para los negocios minoristas es mejor si el mismo producto es consumido por varios clientes que por uno solo.

Algunos consumidores pueden intentar obtener los productos directamente de los productores, pero los productores pueden negarse a la venta directa. Incluso si los productores aceptan realizar o vender en forma directa a los consumidores finales, resulta más incomodo que adquirirlos en el negocio minorista mas cercano (e.g. distancia, horas de atención, presentación del producto, atención al cliente, etc.), salvo que la fabrica se encuentre cerca del consumidor. Uno de los beneficios del negocio minorista es que permite al cliente elegir entre productos de diferentes productores.

Al desarrollar nuestro trabajo analizamos el modelo de las bibliotecas. Ellas compran libros cuyos temas pueden ser de interés para sus lectores. En el caso de las bibliotecas temáticas

[30] Las técnicas de filtrado se basan en el modelo de espacio vectorial (VSM) utilizada también por otros sistemas. VSM representa profiles y documentos como vectores de términos ponderados.

[31] Por otro lado el costo de conseguir un producto más fresco puede ser muy alto o las molestias pueden ser intolerables

(e.g. biblioteca universitaria), generalmente alguien especializado en cada área es quien decide cuáles libros deben tener.En algunos casos lectores relevantes (e.g. profesores) sugieren qué libros se necesitan.

Una comunidad de lectores comparten los libros, ya que el mismo libro no puede ser leído al mismo tiempo por más un lector. Una biblioteca puede comprar varios ejemplares e incrementar de esa forma el stock permitiendo que más de un lector pueda acceder a mismo libro simultáneamente. Algunos libros aún no siendo populares, son poseídos por las biblioteca ya que son considerados relevantes en un tópico determinado, o la gente que los usa son usuarios *significativos* .

ODS es un Sistema de Distribución inspirado en los modelos anteriormente mencionados los cuales han demostrado ser eficientes y escalables globalmente. Sin estos modelos no existiría la economía moderna ni la distribución de conocimientos.

En ODS, los canales de distribución son los que permiten bajar los costos para los productores y consumidores.

Hemos dotado a ODS de esquemas de clasificación, los cuales son colecciones de rótulos o tópicos producidos por una *autoridad de clasificación*, utilizados para asociar la metainformación de los objetos. Estos son usados por los autores o los editores para describir los objetos que publican a los efectos de sus distribución y ruteo por la red para distribuir colecciones de documentos. Los lectores utilizan los esquemas de clasificación para seleccionar los contenidos en los cuales están interesados.

El contenido, estructura y propósito de los rótulos y procedimientos asociados a los rótulos de los objetos es arbitrariamente diverso. Por ejemplo *Usenet News* proveen un esquema de clasificación bajo diversos tópicos *(newsgroups),* con tópicos públicos o moderados. En cambio, IEEE o ACM provee de un catálogo de tópicos en términos de publicaciones con reglas específicas para la selección del contenido y su revisión.

ODS fue pensado para objetos que no sufren modificaciones constantes, como por ejemplo documentos, librerías de software, archivos MP3.

Mas allá de los mecanismos tradicionales que acercan los contenidos a los usuarios (e.g. mirroring, caching, tecnología push) ODS permite la publicación, clasificación y suscripción a volúmenes de objetos (e.g. documentos, eventos, etc).

Mientras que muchos recursos en Internet comienzan a ser inaccesibles debido a la latencia, particiones de red o flash-crowd, los objetos en nuestro sistema son accedidos en forma robusta en regiones locales y actualizados asincrónicamente (con políticas: operación en batch, en horas de bajo tráfico, siempre exceptuando fallas, etc).

ODS permite distribución selectiva, suscripción y notificación, brindando orden y economía en el tráfico caótico y redundante de la actual Internet. Las cadenas de distribución son construidas dinámicamente para encontrar la manera más efectiva de proveer a los usuarios finales una réplica de los objetos que desean obtener, mientras se realiza un buen uso de los recursos disponibles y siempre limitados. Lectores y autores tienen un servicio especializado de distribución de información selectiva y económica a gran escala.

3.4.1. Conectividad Débil, Mecanismos de Consistencia Débil

ODS distribuye en forma selectiva contenidos entre comunidades de intereses comunes, dicha distribución es llevada a cabo mediante mecanismos de replicación. Replicar los recursos aumenta la performance y la disponibilidad de los mismos, ya que no es necesario el acceso a sitios lejanos y además aumenta la probabilidad de que por lo menos una copia del recurso este disponible. Sin embargo cuando los recursos se replican debe tenerse en cuanta la *consistencia* de cada copia, se define como consistencia al hecho de que todas las copias del mismo ítem lógico de datos deben coincidir en exactamente un valor

En sistemas como ODS que además utilizan redes de conectividad débil creemos que es necesario basarlos en modelos cuya estrategia de replicación sea optimista, con lo cual utilizaremos mecanismos de consistencia débil.

Los protocolos de consistencia débil son aquellos cuyo grado de confiabilidad es *fiable* y cuya latencia es *eventual* . En el capitulo 6 definiremos este marco de trabajo.

3.6. Sistema de Distribución de Objetos (ODS)

Está formado por dos redes independientes: La *Red de Distribución de Objetos (ODN)* y la *Red de Ruteo de Objetos (ORN).*

ODN maneja *objetos* [32] que son persistentes y se mantienen replicados en todos los sitios donde hay usuarios interesados. ODN esta formado por Agentes de Servicio (SA) que se unen formando distintos grupos de acuerdo a los intereses de sus usuarios. Una comunidad de usuarios accede los objetos son replicados en cada SA.

ORN es la encargada de generar las redes lógicas de comunicaciones que se utilizarán para realizar la distribución de los objetos entre los usuarios des sistema. Su función es informar a la ODN la forma en que se deben rutear los objetos, para este fin se generan las cadenas de distribución. Es ORN quien construye cadenas de distribución para cada grupo de ODN.

Los SA representan a los negocios, minoristas, fabricas y distribuidores en uno de los modelos, como también bibliotecas y editores en el otro. La diferencia entre SA y una tienda es que en el SA existe solo un ítem de cada objeto[33], y que entre agentes cooperan sin tener que mantener necesariamente una estructura jerárquica.

Las cadenas de distribución son las que permiten bajar los costos debido a que los usuarios acceden localmente a los objetos en que están interesados, aún si son producidos en un SA remoto en la red. Los productores no necesitan saber a donde serán enviados, ellos delegan la distribución al sistema. El sistema encontrará la forma de distribuir los objetos teniendo en cuenta los intereses de los usuarios y minimizando el costo de la distribución. ODS garantiza que todos los objetos alcanzarán a todos los consumidores interesados en un tiempo finito pero no acotado [34].

Manejamos objetos write-one/read-many (recursos Web, FTP , etc.), esto quiere decir que cada objeto puede ser modificado sólo en el AS donde se registró (produjo). Los peor que puede suceder es que alguien acceda a versión de un objeto que no sea la última [35]. ODS ha sido diseñado para objetos que no se modifican frecuentemente.

---

[32] no en el mismo sentido que en la programación orientada a objetos, ni en el modelo de ODP(ISO/IEC 10746-X) sino el aplicable a Bibliotecas digitales

[33] Todos los clientes / lectores interesados pueden consumir / leer solamente un ítem al mismo tiempo.

[34] Si bien el tiempo de convergencia no esta acotado, la experiencia bajo las peores condiciones de red subyacente , 12 horas , tomar el t - > infinito

[35] Herlihy dio la siguiente motivación para control de concurrencia optimista, que es igualmente aplicable al control de replicación optimista: *"... está basado en la premisa de que es más efectivo pedir disculpas que pedir permiso."*

Ambas redes funcionan en forma independiente, para esto se definió un protocolo de comunicaciones por medio del cual se realizan las peticiones y se reciben las respuestas por parte de los componentes de las redes. Los algoritmos para establecer las Redes de Distribución son independientes de los objetos que se estén distribuyendo, esto aumenta la independencia entre ambas redes

Nuestro sistema está diseñado para la estructura actual de Internet, sin modificar los protocolos y standares existentes. Creemos que los modelos que proponen cambios radicales no pueden encontrar viabilidad inmediata, porque definir nuevos modelos y protocolos para reemplazar los actuales, no se puede hacer con la rapidez que demanda la comunidad de Internet.

3.6.1. Entidades

Definimos las siguientes entidades :

- **Objetos**: entidad que se distribuye en ODS.
- **Actores**: usuarios de ODS. Ellos producen y consumen objetos. Un actor puede se una persona o un grupo de trabajo. En ODS interactuan los siguientes *actores* :

Productor: una o varias personas que producen objetos registrados en ODS para la distribución (bajo uno o varios rótulos o esquemas de clasificación).

Autoridad de Clasificación: una o varias personas que producen listas de rótulos (esquemas de clasificación) útiles para la clasificación de objetos.

Consumidor: una o varias personas que se suscriben a listas de rótulos (o autores) y consumen (leen) objetos (documentos), o al menos son notificados de nuevos objetos de su interés (evento de notificación).

- **Workspaces**: relación entre objetos y actores. Un workspace define un subconjunto entre el conjunto de actores y un subconjunto dentro de los objetos. Los Actores están relacionados con los objetos en un workspace como *productores (autores)*, *consumidores (lectores)* o consumidores de *meta-información (observadores)*. Cada workspace tiene un *esquema de clasificación* formado por *rotulos*. Los objetos en el workspace son clasificados bajo al menos uno de esos rotulos.

3.6.1.1. Estructura de los objetos

Todos los objetos están formados por dos partes: *sobre* y *contenido*.

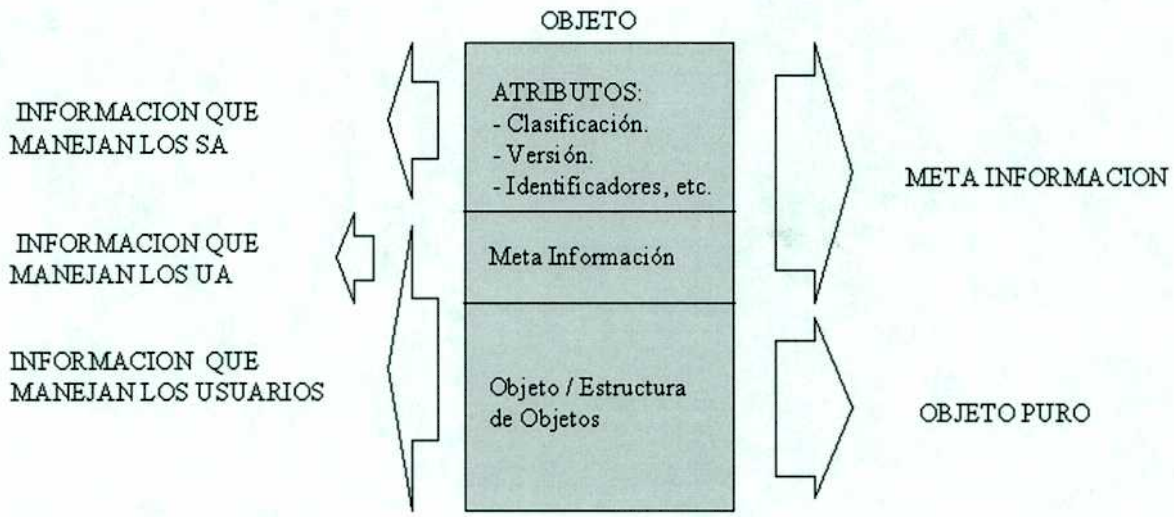
- El sobre contiene los atributos básicos que permiten a cualquier objeto indentificarse en forma única y determinar su versión, además de dar sus clasificación (en el sobre uno de los atributos pueden ser seguridad y una lista de control de acceso, ACL). Los SA son los que interpretan los sobres.
- El contenido está definido por un *header* y un *cuerpo*. El cuerpo está constituido por los datos (archivo MP3, documento, etc) y el header es la información acerca del cuerpo (meta-información básica).

Denominamos *meta-información* extendida al header y al sobre de un objeto. ODS maneja en forma transparente a los objetos. Estos se distribuyen a través de la red basándose solamente en la información contenida en los sobres. Esta información es usada en las *ofertas* (eventos de ODS) que los SA realizan a sus pares. El contenido de un objeto es enviado únicamente si la oferta es aceptada.

ODS mantiene meta-información extendida sobre los objetos, incluyendo información sobre los esquemas de clasificación con los cuales fueron rotulados. Esta meta-información permite la notificación a los usuarios sobre los nuevos objetos en las áreas de sus intereses con un costo mínimo. Esta estructura incluye las necesidades de control para la administración de dichos mensajes en la red y la codificación de los mismos según el tipo.

Los usuarios interactuan con ODS mediante *agentes usuario (UA)*. Los UA deben ser capaces de interpretar la meta-información extendida, con el objeto de proveer a los usuarios de la correcta representación y manejo de cada objeto en la forma mas conveniente de acuerdo a su tipo.

Fig 4: Estructura de los Objetos de ODS



ODS maneja *objetos simples* y *compuestos*. Un objeto compuesto es un conjunto ordenado de objetos simples, los cuales pueden pertenecer a diferentes autores. El sobre de un objeto compuesto contiene una lista de los objetos simples que lo componen.

Esta definición brinda las facilidades necesarias para componer volúmenes o documentos cuyos capítulos son redactados por diferentes autores. También se habilita la posibilidad de modificar o agregar un capítulo de un documento sin necesidad de cambiar todo el documento, sino solamente la parte que fue modificada o agregada.

3.6.2. Arquitectura

ODS esta formado por dos redes virtuales independientes (Fig 4) : Red de Distribución de Objetos (ODN) y Red de Ruteo de Objetos (ORN). ODN acerca los objetos a los usuarios finales en función de sus intereses y ORN construye las cadenas de distribución que ODN necesita para la distribución en forma óptima de acuerdo a esos intereses y el estado de la red subyacente.

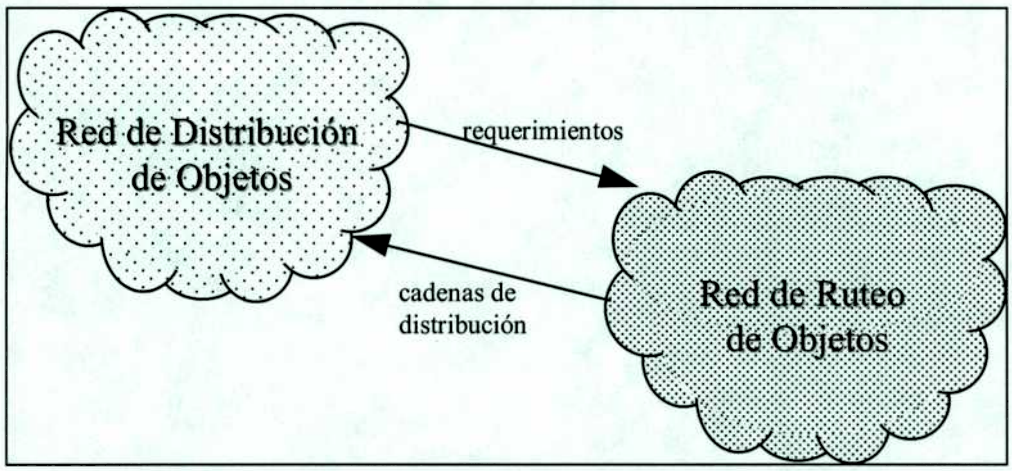


Fig. 2: Sistema de Distribución de Objetos

Los Workspaces definen grupos de SA en ODN (*grupos ODN*). Los SA ingresan a los grupos de ODN respetando la subscripción de sus usuarios. Los SA se unen a los grupos como productores, consumidores de objetos o de meta-información. Decimos que un SA adquiere un *tags* como productor de un workspace X (P[X]), consumidor de Y (C[Y]), o consumidor de meta-information de Z (MIC[Z]), etc.
Los SA cooperan con el fin de obtener una replicación eficiente dentro del grupo. Los objetos son únicamente replicados dentro del grupo. De esta forma nosotros buscamos poner algún orden en el caos que lleva tener información que no esta clasificada.

ORN arma cadenas de distribución para cada grupo en forma dinámica. Para construir estas cadenas los *agentes de ruteo (RA) (fig. 5)* (miembros de ORN) tienen en cuenta el tipo de membrecía al grupo de cada AS y el estado de la red subyacente.

Si bien News también replica los objetos (noticias) únicamente en los grupos de interés, la distribución es siempre masiva y no construyen cadenas de distribución en forma dinámica sino manual.

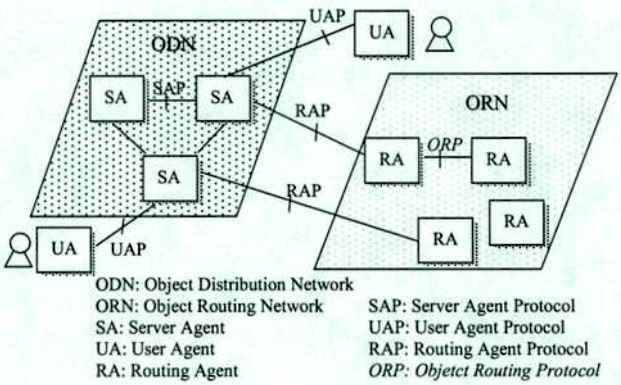


Fig. 3: Redes y Protocolos de ODS

Los mecanismos de ruteo utilizados en ORN para construir las cadenas de distribución son totalmente independientes de la clase de objetos que maneja ODN. Ambas redes fueron diseñadas para funcionar en forma independiente, definiendo una clara interface entre ellas, para que ORN provea servicios a ODN en forma transparente y además posibilito el desarrollo de los prototipos en forma independiente.

3.6.2.1. Red de Distribución de Objetos (ODN)

Una ODN está compuesta por un conjunto de SA que cooperan con el fin de replicar objetos para usuarios en sitios dispersos geográficamente, sin que estos deban conocer el sitio origen de los mismos y menos aún si éste se encuentra alcanzable. Cada usuario accede a una réplica de la colección los objetos a los que se suscribe en el SA (mas cercano) que le provee un punto de acceso a ODN y también registra allí los nuevos objetos que desea distribuir a través de la misma. De esta forma la replicación resulta transparente a los usuarios.
Los usuarios acceden a ODN a través de Agente Usuario *UA* que actúan como interfaces (Fig 6). Un UA se comunica con el SA utilizando el *Protocolo de Agente Usuario (UAP).*
En nuestro modelo de referencia hemos utilizado dos variantes de UAP, uno mediante HTTP y como UA un cliente Web y la otra mediante el desarrollo de un protocolo UAP e interfase adaptada un una red de distribución de documentos (DDN).

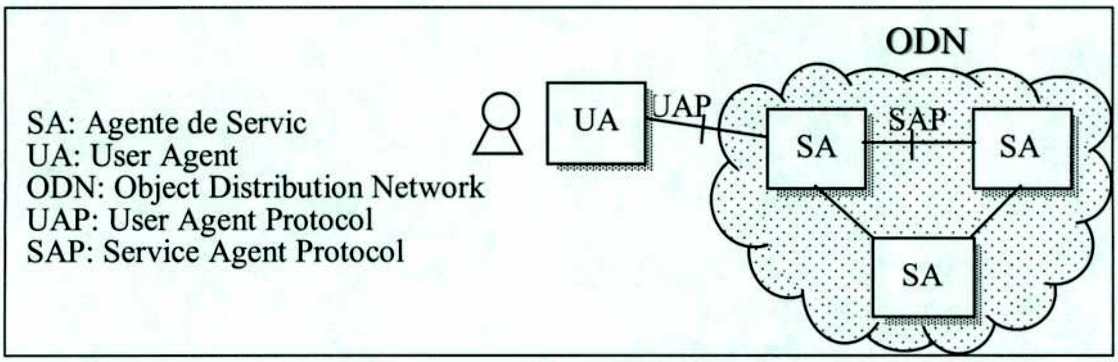


Fig 4 : Protocolos de los Agentes de Servicio

Los objetos son unívocamente identificados. Los objetos sólo pueden ser actualizados o eliminados por sus *dueños*. En los sobre de los objetos el campo *versión*, que permite a los SA decidir si se encuentra actualizado (ante objetos con el mismo nombre). Otra posibilidad para determinar la versión de un objeto es tener un atributo que indique la fecha de última modificación del mismo y si es que fue eliminado, una marca de tiempo (fecha de baja) que indique cuando fue retirado de ODN. Estos atributos sólo se pueden modificar en el SA en el que se encuentra registrado el objeto home SA.

Cada SA mantiene una únicamente la ultima versión de un objeto recibida, ya que una nueva versión reemplaza a la anterior, sin conservar un registro de las versiones anteriores (aunque se podría, depende del escenario). Cuando un objeto es eliminado de ODN, los SA deben mantener cierta información sobre el mismo para poder determinar si reciben una oferta desactulizada de ese objeto.

Los SA pueden tener los objetos completos o sólo la meta-información de los mismos. Los usuarios se subscriben a determinadas colecciones de objetos en sus SA, pudiendo seleccionar recibir el objeto completo o solo la metainformación (para notificarse de nuevos objetos).
Algunos objetos están formados a su vez por *sub-objetos*, llamamos a estos objetos, *objetos compuestos*. El cuerpo de un objeto compuesto es el conjunto de los sub-objetos que lo forman. La meta-información de un objeto compuesto debe contener su estructura , es decir la relación entre los sub-objetos que lo componen. Cada sub-objeto puede modificarse en forma independiente y tiene por lo tanto un identificador único y un número de versión asociada.
Tanto los objetos como los sub-objetos tienen un atributo que indica el tamaño, que tendrá significado distinto de acuerdo a la clase del objeto. Los sub-objetos sólo tienen cuerpo, no

tienen meta-información, heredan algunos atributos de la meta-información de los objetos que forman. Los atributos a heredar están definidos por la clase del objeto..

Cada SA tiene un rol dentro de los grupos a los que se suscribe, definido por tags que representan el tipo de objetos que produce y consume.

Decimos que un SA *produce* los objetos que registran sus usuarios y *consume* los objetos en que sus usuarios están interesados. ODN maneja tres tipos de tags: *Productor[X] (P[X]), Consumidor de Objetos[X] (C[X]) y Consumidor de Meta-información[X] (CMI[X]).* Donde *X* indica el tipo de objeto al cual se refiere el tag . Un grupo es un conjunto de SA que comparten un interés común. Habrá pues un grupo para cada X posible.

3.6.2.2. Cadenas de Distribución

Para cada grupo construimos y mantenemos una *cadena de distribución.* La misma asegura que todos los consumidores del grupo reciben los objetos que se producen en él. Un agente de servicio considera *clientes* dentro de un grupo a aquellos a los que debe enviar objetos. Por otro lado, el agente de servicio que recibe objetos considera a los que le envían como *productores.*

Loas SA trabajan en forma activa para distribuir los objetos sin esperar que les sean solicitados ('enviar a todos lados'). Cuando un SA recibe una nueva versión de un objeto, ofrece el mismo a sus clientes y SA que pertenecen al mismo grupo. De esta manera los objetos puedan llegar a todos los SA, sin necesidad de que los usuarios tengan que solicitarlos. Los usuarios no necesitan realizar ningún esfuerzo para obtener los objetos a los que se suscriben, simplemente acceden a la última versión disponible en el SA. Es tarea de los SA mantener los objetos actualizados.

La distribución de un objeto comienza en el SA en el que se registró el mismo. Denominemos al mismo *home SA.* El *home SA* lo ofrece a sus clientes y estos, a su vez, hacen lo mismo con sus clientes. *Si* Como la cadena de distribución provee al menos un camino desde el *home SA* a todos los consumidores del grupo, este mecanismo de inundación asegura que la nueva versión de un objeto o uno nuevo, llegue a todos los consumidores (SA interesados en el grupo).

Una cadena de distribución puede proveer más de un camino entre dos SA. Sin embargo, los objetos no entran en ciclos, ya que los agentes descartan toda oferta de un objeto con una versión menor o igual al que poseen.

Las cadenas de distribución están formadas sólo por SA del grupo. Los SA sólo reciben objetos que están interesados en consumir. Cada consumidor mantiene una réplica local persistente de los objetos del grupo, además de los objetos que produce. De esta forma los SA pueden repetir la oferta del mismo objeto a sus clientes. También pueden determinar si la versión de un objeto es más actual que la que poseen, sin que esto signifique un proceso costoso para ellos.

En otras redes, tales como Mbone, mantener información para determinar si la versión de un objeto es más actual que la que el agente ya inundo anteriormente puede resultar muy costoso y se exige por lo tanto que las cadenas de distribución no provean caminos alternativos [Sem97]. Dado que Mbone utiliza IP multicast, que al igual que IP es un protocolo con entrega best-effort, no repite las ofertas ya realizadas.

Las redes tienen una topología física y una topología lógica. La topología física está determinada por las conexiones entre componentes físicos. La topología lógica de Internet es un grafo completamente conexo, ya que IP [Post81] esconde la topología física para permitir que todo host se pueda comunicar con cualquier otro host.

Sobre esta topología lógica, debemos encontrar cadenas de distribución que permitan que todos los objetos producidos en el grupo lleguen a todos los consumidores, optimizando el costo de distribución de los mismos. Cuando se construyen las cadenas de distribución cada agente de servicio es tratado en forma diferente según su rol.

Algunos autores proponen el uso de IP multicast [Dee89] para la *replicación masiva* (e.g. DPM [Don95], Muse[]). IP multicast implica envío best-effort de datagramas a un grupo de hosts que comparten una única dirección IP multicast. Esto se adecua correctamente a aplicaciones de tiempo real para audio y video, donde la pérdida de un datagrama no constituye un gran problema[36] y donde lo realmente critico es que lleguen datagramas duplicados o con una gran demora. Sin embargo cuando se busca consistencia de los datos, la utilización directa de IP multicast en esquemas globales no funciona, pues el envío de mensajes no es confiable y todos los miembros del grupo deben estar activos para recibir una actualización, ya que la distribución se realiza siempre desde el productor del objeto.

[36] Acá hay que ver , ya que en MPEG producen algunas distorsiones de la imagen

Los algoritmos de ruteo para construir cadenas de distribución multicast en Internet (e.g., CBT Core Based Trees [Bal93], DVMRP Distance Vector Multicast Routing Protocol [Wai88], MOSPF Multicast OSPF [Moy94], PIM Protocol Independent Multicast [Dee96], etc.), pueden ser analizados en forma independiente al uso de IP multicast. Ya que nuestro problema es de nivel de aplicación y no es correcto depender del nivel de red para solucionarlo. Proponemos por lo tanto adaptar los protocolos de ruteo utilizados en el nivel de red al de aplicación. (Si bien actualmente se han propuestos Multicast Transport Protocols tales como MDP [MD96], AFDP[CKW96],RMTP [LP96] MFTP [MRT+97] etc).

Quien construye las cadenas de distribución es la *Red de Ruteo de Objetos (ORN)*. Los agentes de servicio no conocen la forma en que opera esta red. ORN está formada por *agentes de ruteo* que cooperan para construir cadenas de distribución adecuadas a los requerimientos (suscripción) de los agentes de servicio, utilizando un protocolo que llamamos genéricamente *Object Routing Protocol (ORP)*.

Cada SA es usuario de un agente de ruteo (fig. 6), el cual determina qué conjunto de clientes debe unirse a cada grupo. Los SA se comunican con los agentes de ruteo utilizando el *Routing Agent Protocol (RAP)*.

Determinar cadenas de distribución eficientes no es trivial y la forma más apropiada para realizarlo depende fuertemente del escenario de trabajo. Hay que considerar la cantidad de SA, como también el comportamiento de los mismos, siendo la infraestructura de la red subyacente muy importante. En determinados escenarios (simples) las cadenas de distribución se pueden configurar en forma manual, sin embargo es necesario utilizar ruteo dinámico para proveer un sistema de distribución que escale correctamente. Todos estos factores determinan el tipo de algoritmo de ruteo en que se basa ORP.

Sólo precisamos armar cadenas de distribución dentro de los grupos que tienen por lo menos un productor y un consumidor. Dentro de cada grupo, la cadena debe proveer un camino desde cada agente productor a todos los agentes consumidores. Las cadenas no deben proveer necesariamente un solo camino entre un productor y un consumidor, podrían tener ciclos, ya que como hemos visto, los SA tienen un mecanismo independiente para evitar que los objetos entren en ciclos.

El mecanismo de ruteo debe cumplir las siguientes condiciones:

- Manejar políticas : Se deben respetar los roles (tags) de los agentes de servicio dentro de cada grupo. Los agentes de servicio no deben ser obligados a consumir objetos que no desean.
- *Minimizar* costos de distribución para una colección de objetos: Las cadenas de distribución deben ser de costo mínimo según alguna métrica (e.g. latencia, bandwidth, pérdidas de paquetes).
- Ser adaptativo: Las cadenas de distribución deben adaptarse a los requerimientos de los agentes de servicio y al estado de la red lógica sobre la cual está montada ODN. Sin embargo por el mecanismo de distribución de ODN, no es deseable que las cadenas se modifiquen con mucha frecuencia. El criterio para considerar si los protocolos que arman las cadenas tienen un comportamiento adaptativo es más laxo que el utilizado en el nivel de red, ya que los tiempos resultan menos críticos.
- Ser escalable: Las cadenas de distribución debe asegurar escalabilidad a ODN, ya que un sistema distribuido debe poder manejar la adición de usuarios y sitios, con un aumento mínimo de costo, degradación de performance y complejidad administrativa.
- Robusto : debe garantizar un eventual envío de cada objeto a cada SA aun cuando algunos de estos se encuentren inalcanzables. Cuando un SA vuelve a estar operativo debe recibir todos los objetos que se perdieron en esos momentos.

3.6.3. Red de Ruteo de Objetos (ORN)

Es función de esta red armar y mantener las cadenas de distribución para cada grupo de ODN que tenga al menos un productor y un consumidor. Cada cadena debe proveer un camino a cada productor con todos los consumidores. Sin embargo una cadena de distribución puede proveer más de un camino entre dos SA. No obstante ello los objetos no entran en ciclos, ya que los agentes descartan toda oferta de una versión menor o igual a la que poseen.
Podemos visualizar ODN como un grafo, donde cada SA es un nodo. Dado que trabajamos sobre una red lógica que permite que todos los nodos se comuniquen entre si, el grafo es completo. Cada eje del grafo tiene asignado un peso que representa el costo de utilizar el enlace lógico. Una métrica posible a utilizar para determinar el costo, es realizar mediciones estadísticas del estado del enlace (e.g. usando topology-d [katia97]).
Los agentes de ruteo deben armar cadenas de distribución teniendo en cuenta que la cantidad de miembros del grupo puede cambiar y en particular considerando que puede crecer en forma significativa. Las cadenas también deben considerar el estado de la red subyacente.

Presentamos un mecanismo de ruteo distribuido, que puede ser utilizado en un esquema plano o jerárquico. Primero presentamos el algoritmo básico a ser utilizado en un esquema plano y luego las modificaciones que deben realizarse para adaptarlo a un esquema topología jerárquico. Por último presentamos una suite de protocolos a utilizar en una topología jerárquica.
Los agentes de ruteo determinan dinámicamente la estructura de los grupos. Cada RA anuncia a sus pares los rótulos de los SA que representa. Con esta información cada RA puede conocer a los miembros de cada grupo. Los RA deben realizar cálculos para todos los grupos a los que pertenecen sus SA.

Los nodos en el grafo son clasificados según sus roles, en función de los rótulos que poseen. Trataremos a este grafo como un conjunto de grafos en distintos planos, uno para cada grupo. Los cálculos se realizan utilizando estos grafos.

Cada SA posee información de acuerdo a su rol en el grupo. Los nodos (SA) con el mismo rol en el grupo, forman un *nivel.*

Nivel	Rol de los Nodos SA	Objetos que poseen
1	Productores Puros	Solamente los producidos localmente
2	Productores/Consumidores	Los que son producidos localmente (dado que son productores) y todos los del grupo (ya que son consumidores)
3	Consumidores de Objetos	todos los del grupo
4	Consumidores de Meta-info.	meta-información de todos los del grupo

Información que poseen los nodos de cada nivel

Nosotros construimos las cadenas de distribución de los grupos respetando los niveles. Cada nodo tiene clientes dentro del mismo nivel o en un nivel superior (fig 4). De esta forma eliminamos la posibilidad de tener algunas cadenas de distribución óptimas que no respeten esta división de niveles. Dentro de cada nivel se busca minimizar el costo global de la distribución.

Los productores/consumidores forman un subgrafo completamente conexo, que denominamos *core.* Dentro del core hay un camino entre todo par de nodos. Cada productor puro debe elegir un nodo dentro del core como cliente. De esta forma cada nodo dentro del core tendrá también todos los objetos producidos por los productores puros del grupo, en otras palabras cada nodo en el core eventualmente tendrá todos los objetos del grupo.

Siempre que hayan consumidores en el grupo, tiene que existir un core. En el caso de no haber productores/consumidores, se elige un consumidor de objetos para formar el core, y si tampoco hubiese nodos del ese tipo, se elige un consumidor de meta-información.

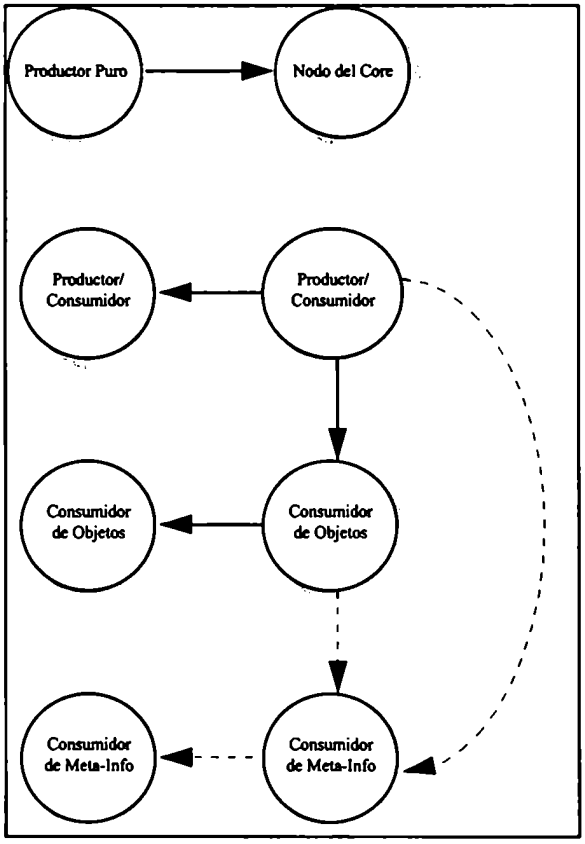


Fig. 5: Comunicación entre nodos

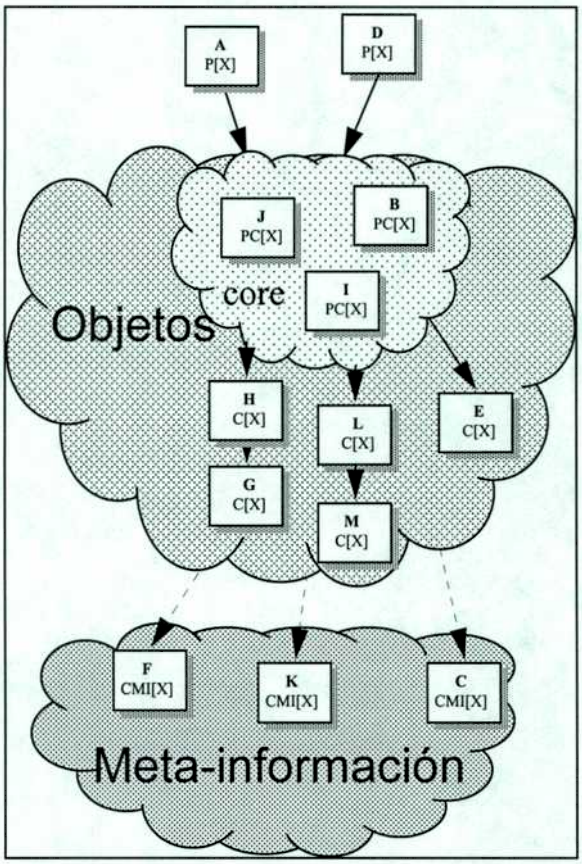


Fig. 6: Ejemplos de cadenas de distribución

Los RA deben tener una visión global del grupo para poder armar las cadenas de distribución. Cada RA inunda a sus pares el estado de los enlaces de sus nodos, tal como en un algoritmo link-state [McQ80]. La inundación se realiza en forma independiente a los grupos a los que pertenecen los SA. Se mantiene esta información en una tabla topológica en cada RA. Los RA realizan los cálculos en forma independiente para cada grupo, utilizando la tabla topológica.

En consecuencia cada RA anuncia a los demás los rótulos y el estado de los enlaces de sus nodos. Los RA necesitan conocer cómo difundir esta información a los demás RA de ORN. Definimos entidades denominadas *coordinadores* que se hacen cargo de proveer a cada RA con esta información. Cuando un RA quiere comenzar a operar se suscribe con el coordinador. Entonces el coordinador anuncia la incorporación de un nuevo miembro a los demás RA que ya estaban operando y envía una lista de miembros de ORN al nuevo RA.

3.6.3.1. Algoritmo

Cada nodo del grafo tiene un identificador único. Los identificadores de los nodos se pueden ordenar en forma creciente (e.g. direcciones IP o cadenas de caracteres). Por lo tanto siempre se puede decidir, dado un conjunto de nodos, cuál tiene el menor identificador.
Los ejes del grafo se identifican en forma única por el identificador de los nodos que unen, en forma de par ordenado (i, j) con $i < j$. Los ejes se pueden ordenar entonces por orden lexicográfico ($\forall$ par de ejes (i,j), (k, l): $(i,j) < (k, l) \Leftrightarrow i < k$ o $i = k$ y $j < l$).

En el modelo que planteamos siempre debe existir un core, por lo tanto si no hay nodos productores/consumidores en el grupo, se elige como core al consumidor de objetos con menor identificador. En el caso de que tampoco hubiese ninguno, se elige al consumidor de meta-información con menor identificador.

Un RA que representa a un nodo productor puro busca la forma de comunicar al nodo con el core con costo mínimo, elige al nodo del core al que tiene menor distancia como primer salto (cliente).

Los RA que representan a los productores/consumidores calculan un árbol de cubrimiento mínimo que incluye a todos los nodos del core, que llamamos *árbol de productores*.

Para calcular un árbol de cubrimiento proponemos el algoritmo de Prim. Cuando hay conflictos con dos ejes de igual costo se elige él de identificador menor. De esta manera todos los RA que realizan el cálculo obtendrán el mismo árbol de cubrimiento mínimo, siempre y cuando partan de las mismas tablas topológica. Dado que el árbol de cubrimiento mínimo se calcula sobre un grafo no dirigido, consideramos a la tabla topológica como una matriz triangular superior. (A*) debemos asignar $a^*_{ij}=a^*_{ji}=(a_{ij}+a_{ji})/2$

Cada uno de los SA representados por los nodos del árbol de productores debe considerar cliente a todos los SA correspondientes a los nodos adyacentes. Todo par de nodos adyacentes en el árbol se verá mutuamente como proveedor y cliente. Dado que el árbol no tiene ciclos se pueden eliminar los ciclos en las ofertas fácilmente, no enviando una oferta al proveedor que envío el objeto correspondiente.

Los RA que representan a los consumidores de objetos calculan un árbol de cubrimiento mínimo en donde se incluyen a todos los consumidores de objetos. Este árbol se denomina

árbol de consumidores y se arma utilizando el algoritmo de Prim partiendo de S = {productores/consumidores}. En forma similar, los RA que representan a los consumidores calculan un árbol en donde se incluyen a todos los nodos interesados en consumir meta-información. Este árbol se denomina *árbol de meta-información,* para el cálculo se parte de S = {consumidores de objetos}.
Cuando un nodo es productor puro y consumidor de meta-información, los RA realizan los cálculos como si fuesen dos nodos distintos. Aparecerá por lo tanto dos veces en la cadena de distribución, enviando objetos al core y recibiendo meta-información.

3.6.3.2. Ruteo Jerárquico

En una topología jerárquica los nodos se dividen en dominios. Dentro de cada uno se utiliza un *protocolo de ruteo intra-dominio (protocolos de ruteo interno).* Dicho protocolo puede ser distinto al utilizado en los demás dominios. Los nodos deben poder recibir los objetos generados en otros dominios, hay que establecer caminos entre los mismos, para esto se utiliza un *protocolo de ruteo inter-dominio (o protocolo de ruteo externo).*

Los agentes de servicio no tendrían que notar que la topología de ruteo es jerárquica, pues el armado de cadenas de distribución debe resultarles transparente. Podemos realizar una similitud con el ruteo de Internet, donde el nivel de red oculta todos los detalles de la división en sistemas autónomos.

También podemos ver a los dominios como un grafo completo, donde cada dominio es un nodo y los ejes tienen pesos asignados en forma estática. Cada eje representa el enlace lógico que une un dominio con otro. El peso de un eje podría ser una estimación del comportamiento promedio de los enlaces físicos subyacentes.

Cada dominio tiene además de su coordinador que maneja la suscripción de los RA del dominio, una entidad que denominamos *router externo*, que se ocupa del ruteo inter-dominio. Los routers externos resumen los rótulos de los nodos de su dominio para informarlos a sus pares. El resumen se realiza asignando al dominio todos los rótulos de los nodos del mismo. A su vez inundan el estado de sus enlaces inter-dominio a los demás routers externos

Utilizando esta información cada router externo computa el algoritmo propuesto para cada grupo en el que tiene nodos interesados. El algoritmo se computa sobre el grafo que representa a los dominios, por lo tanto cada router externo obtiene como resultado un

conjunto de dominios a los que debe considerar clientes. Sin embargo los SA no saben nada sobre dominios y necesitan información sobre los SA que deben considerar clientes.

Un grupo puede tener miembros en distintos dominios. Los routers externos deben determinar para qué grupos hay que calcular *cadenas de distribución inter-dominio* (por lo menos un productor y un consumidor, y miembros en más de un dominio). Los routers externos informan a los RA del dominio sobre dichos grupos que son de interés local.
En cada dominio debe haber un core para cada grupo con consumidores. Elegimos al nodo con menor identificador del core del grupo para que lo represente hacia los demás dominios. Este nodo se convierte en cliente de algún nodo de los dominios proveedores (en el ruteo inter-dominio). Si el dominio tiene productores, este nodo también actuará como proveedor hacia los dominios que sean clientes en el ruteo inter-dominio.
Un problema que se plantea es la ausencia de consumidores de objetos en el dominio (ej.. sólo hay productores puros). Como el core siempre está formado por nodos consumidores, en este caso el dominio no tiene un core que posea objetos. Podríamos tener un core que posea sólo meta-información, que tampoco sirve para distribuir objetos hacia otros dominios. Adoptamos como solución en este caso, que cada productor del dominio considere clientes a los nodos elegidos en los dominios clientes.
El RA del SA que debe representar al core del dominio, informa sobre el mismo al router externo. El router externo, a su vez, informa sobre el SA a los routers externos de los dominios proveedores. Los routers externos de los dominios proveedores, informan a los RA de los SA que deben actuar como proveedores (nodo del core o productores puros) sobre los SA clientes de otros dominios.

3.7. Un Modelo de Referencia : Red de Distribución de Documentos (DDN)

El modelo de Red de Distribución de Documentos (DDN) es una aplicación particular de ODS, donde el principal objetivo es la distribución de documentos (aunque no son los únicos objetos, DDN también maneja objetos con información sobre los autores y esquemas de clasificación).
Dado que la consistencia de los documentos en los AS se mantienen débilmente (o la granuralidad de sincronización puede estar en el peor caso en el orden de horas), asumimos que : Los documentos son persistentes, pueden ser clasificados y no cambian a menudo.

Nos basamos en las siguientes invariantes en cuanto a la mutabilidad de los documentos:

- Bestavros [Bes95b] plantea que el porcentaje de actualización de los documentos que denomina 'globalmente populares' es a lo sumo de un 0,5% por día, restringido además a un pequeño subconjunto de ellos.
- Blaze [Bla93] plantea que la probabilidad de que un archivo vuelva a modificarse va disminuyendo a medida que transcurre el tiempo desde su última modificación

Si bien el volumen de documentos disponibles en Internet es enorme y la producción de los mismos crece día a día, la cantidad de documentos que se distribuiría por DDN no sería tan grande como en caso de las News ya que en las News cuando se produce un documento, éste se postea directamente al grupo y se replica masivamente. En cambio en DDN existe una relación de oferta-demanda. Dicha oferta puede basarse en preferencias definidas a priori o en una oferta basada en los intereses, sobre determinados documentos, que tiene una comunidad a otra comunidad similar que pertenece al mismo workspace. Una forma que se tenemos de determinar los intereses de una comunidad es en base a un análisis logs de los servidores web o de servidores proxies . En el capitulo 5 discutiremos en mayor profundidad la validez de estos invariantes.

El cuerpo de un documento puede ser un archivo o una colección de archivos con texto multimedia. Por ejemplo colecciones de documentos HTML, archivos PDF, documentos LaTex , etc. Cada documento pertenece a su autor y puede ser modificado solo por él.
Los documentos pueden ser agrupados para formar objetos compuestos, que denominamos *volúmenes* (e.g. libros, revistas, technical report, diarios , etc). Un volumen también puede contener otros volúmenes. Los volúmenes pueden estar formados por documentos de diferentes autores. El cuerpo de volumen tiene información sobre su estructura y también puede tener una introducción, índice , etc.

Dependiendo del modelo (*escenario*) usado, los documentos necesitan ser publicados o hechos públicos por sus autores. En el primer caso necesitamos tener un actor que cumpla con el rol de *editor*. Los editores garantizan de alguna manera la calidad de los documentos que son publicados (ej.. ACM, IEEE, etc).
En el otro caso cada autor puede considerarse un editor. Esto tiene sentido en dos escenarios : publicaciones sin referatos o en estado de producción. Filtros de Distribución pueden ponerse en el estado de los objetos (e.g. solamente recibir documentos publicados).
Los documentos (o volúmenes) necesitan ser clasificados por una *autoridad de clasificación* o únicamente por sus propios autores. La clasificación no garantiza en principio calidad, únicamente nos dice que determinado documento trata sobre determinado tópico.

3.7.1. Agente usuario (UA)

Cada agente usuario actúa como una interfaz entre los usuarios y la red de distribución. Depende del ambiente de trabajo y de las necesidades de los usuarios permitirles configurarlos de acuerdo a sus preferencias. Los agentes usuarios no necesitan ser los mismos en todos lados. A continuación mencionamos algunas posibles implementaciones:

- Acceso , potencialmente fuera de línea , a publicaciones por una aplicación cliente en una estación de trabajo. El usuario inicia la conexión para acceder a la nueva información de su interés. Nuevos documentos o cambios producidos desde la ultima conexión son también transferidos .
- Una aplicación CGI corriendo en un server http localizado en el nodo que es agente de servicio de DDN. Nuestra actual Implementación utiliza esta.
- Un proceso que coopera con un proxy server [Luo94] que permita a las solicitudes HTTP de los usuarios ser redireccionadas como una solicitud to local requests to the service agent. Esto usa solamente una funcionalidad provista por un SA, pero es transparente a los usuarios los cuales no necesitan saber de la existencia de DDN.

3.7.2. Suscripción

Proponemos para suscribirse a los objetos de DDN, dependiendo si se está interesado en documentos, eventos o esquemas de clasificación, los siguientes :

Documentos: con un rótulo; de un Autor; de un autor con un rotulo

Información (eventos): idem como un documento; sobre autores de Documentos ; sobre autoridades de esquemas de clasificación.

Esquemas de Clasificación: de una Autoridad; un esquema dado.

3.7.3. Agentes de Servicio

En definitiva para manejar la publicación, suscripción y cooperación con otros agentes de servicio, algún agente de servicio debe especializarse con funcionalidades internas que configure su comportamiento de diversas maneras, algunos ejemplos son:

[esquema] repositorio de colecciones de esquemas de clasificación.

[metadata] repositorio de metainformación sobre documentos en general (sin contenido), sobre documentos en papel, etc.

[contenidos] editor (editorial) con diferentes políticas de admisión de documentos, cobro y envío.

3.7.4. Escenarios de Utilización

Mediante DDN buscamos que el volumen de documentos disponibles para un usuario no sea extremandamente grande, y proveer al mismo referencias sobre la calidad de los documentos. Como vimos esto debe realizarse de una forma que mejore el acceso a los documentos y al uso eficiente de los recursos.
Un lector puede expresar sus intereses de lectura en término de suscripciones. Esto puede expresarse en términos de uno o varios esquemas de clasificación, un selector de documento o de metainformación. Típicamente puede ser un lista <ítem, modo>, donde ítem es:
<nombre de autoridad de clasificación authority, categoria>
y modo es : <documento o metadata>. Por ejemplo interesado en cualquier documento de SIGOIS de ACM, e interesado en ser notificado de cualquier evento sobre sistemas distribuidos que organizados por la asociación ATI, puede expresarse de la siguiente manera:

```
//acm.org/sigois/; document
//x.org/distributed_systems/; metadata
```

En los ejemplos siguientes, delineamos dos escenarios completos. Un ambiente literario (Fig 8) donde DDN puede distribuir libros producidos por los escritores a los lectores a través de editores y librerías. La interacción entre escritor-editor y libreria-lector son locales (agente usuario a agente de servicio) mientras que la relación editor y libreria son globales (distribución entre un numero N de SA de DDN).

Un modelo similar puede ser aplicado a la publicación científica en un ambiente de investigación universitario (Fig9).

Fig 8: DDN en un ambiente literario

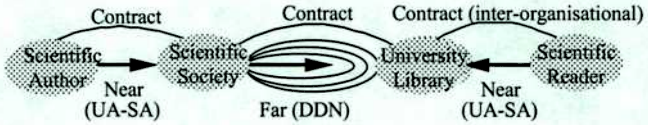


Fig. 9: Una DDN Académica

3.7.5. Distribución de objeto versus caching y mirroring

ODS difiere de las técnicas de caching y mirroring en varios aspectos:

- Eventos (creación o modificaciones de un documento) circula mas rápido que la circulación de un documento. Esto asegura la consistencia de la información y provee información de notificación a los clientes con un costo muy bajo.

- La Distribución se ejecuta en unidades de contenido (volúmenes) antes que parte/todo de un sitio (mirrors), o partes de un documento (caching de una pagina).

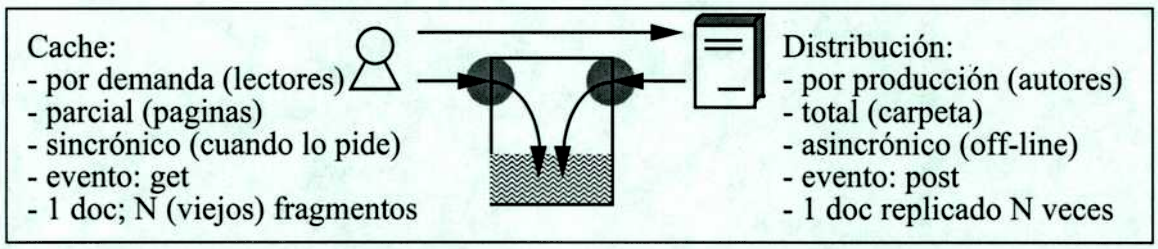
- Asincrónico: los cambios son realizados por el autor (evento producción), antes que una solicitud (caching) o actualizaciones periódicas (mirroring). Las rutas de distribución son optimizadas para la distribución y ayudan a mejorar el uso del ancho de banda. Una eventual conexión fuera de banda de los lectores puede ser implementada bajo este modelo.

- Escalable

- Un documento replicado N veces puede mantenerse más consistente que un documento con M fragmentos cacheados. Invalidación o actualización puede hacerse con ODS, mientras que los mirrors convergen periódicamente y las diversas caches no pueden ser todas notificadas o invalidadas.

- El tráfico puede ser más ecualizado, organizado y automatizado porque los usuarios pueden expresar sus intereses en ciertos documentos por suscripción, y los documentos pueden ser rotulados bajo una o varias categorías.

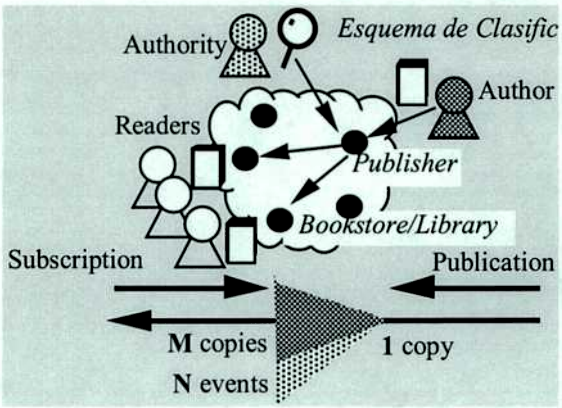
- Cuando el rotulado es realizado bajo algún criterio de calidad de alguna Institución (Universidad, ISOC, IEEE, etc.), puede ayudar a mejorar la calidad de la información que reciben los clientes.



Técnicas de Cache vs. Técnicas de Object Distribution

En definitiva son los esquemas de clasificación (que determinan los canales de distribución) lo que permite a consumidores y productores optimizar la distribución de los objetos.

Este modelo es adecuado para muy gran escala y no existe en la actual comunidad Internet, pero debe ser introducido sobre la actual infraestructura de red. Su introducción gradual proveerá de ventajas inmediatas a los usuarios. En la Fig siguiente se puede observar una comunidad de gran escala de autores, lectores y autoridades de clasificación que colaboran para distribuir en forma efectiva objetos y eventos, siendo esto ODS.



3.8. Seguridad en ODS

Si bien esta fuera de los objetivos de esta disertación los aspectos relacionados con la seguridad a medida que realizamos la Implementación del modelo de referencia planteamos una arquitectura de seguridad para ODS. Proponemos que en [BMT97] que la seguridad se encapsule en el objeto mismo, en forma de atributos dentro de la cabecera. El nivel de seguridad de un objeto lo establece el autor en el momento de su publicación. A su vez se indica que operaciones puede realizar cada usuario sobre el objeto a través de una lista de control de accesos (ACL). Se utilizan flags de seguridad a través de los cuales se puede especificar:

- Confidencialidad: indica si el objeto será transmitido en forma encriptada.
- Autenticación Simple: indica si se requiere autenticación del productor del objeto.
- Autenticación Mutua: indica si se requiere autenticación del productor del consumidor.
- No repudiación : indica que la autenticación de las entidades deberá ser realizada de manera de no negar la realización de dichas actividades.

3.9. Conclusiones

Hemos presentado un sistema asincrónico de replicación de consistencia débil, Sistema de Distribución de Objetos (ODS) que utiliza los recursos disponibles en Internet y se adapta a la red subyacente. Pueden haber múltiples sistemas basados en ODS trabajando en Internet al mismo tiempo. Cada uno con sus propias clases de objetos, mecanismos de seguridad, autoridades de clasificación, etc. Estos sistemas de distribución se pueden configurar fácilmente para eventos especiales (workshops, cursos virtuales, etc.).

Los objetos son actualizados automáticamente en forma off-line. Los usuarios se suscriben para recibir ciertos objetos y el sistema les asegura que siempre tendrán acceso a la última versión disponible de dichos objetos. El modelo se ajusta correctamente a objetos persistentes que no cambian constantemente y cuando cambian no es necesario informar inmediatamente a los usuarios.

Las dos redes virtuales que forman ODS: Red de Distribución de Objetos (ODN) y Red de Ruteo de Objetos (ORN), fueron diseñadas para trabajar en forma independiente. Esto nos permitirá continuar el desarrollo de cada red en forma separada. Además deseamos generalizar el concepto de los mecanismos de ruteo utilizados en ORN a otros sistemas y definir los servicios de ORN en forma más general.

Actualmente contamos con un prototipo que tiene un mecanismo incorporado para armar las cadenas de distribución. Las cadenas de distribución se arman partiendo de un grafo que no es completamente conexo. Decimos por lo tanto que cada SA tiene vecinos (SA con los que se puede comunicar). Los SA adoptan entonces una conducta aún más cooperativa, pues no sólo consumen los documentos de interés para sus usuarios, sino también los documentos de interés para sus vecinos. De esta forma cuando el grafo es conexo, cada SA podrá recibir los documentos de su interés. Sin embargo como todos sus vecinos harán propios sus intereses, ellos también recibirán los documentos y a su vez transmitirán el interés a sus vecinos. En el peor caso todos los SA poseerán todos los documentos, perdiendo las ventajas de la distribución selectiva, tendiendo esto a ser una *replicación masiva*.

A partir del modelo de referencia concluímos que era necesaria la construcción de cadenas de distribución que respetasen los rótulos de cada SA y que por lo tanto era necesario que todos los SA se pudiesen comunicar entre si. Podemos decir que en ORN todos los SA son vecinos, aunque un SA no se comunique con todos sus vecinos. De hecho en ORN dos SA se comunican sólo cuando mantienen una relación cliente-proveedor dentro de una cadena de distribución. Esto implica que la comunicación entre SA queda restringida a SA del mismo grupo (SA con un interés común).

Podemos comparar el mecanismo utilizado en el modelo de referencia a utilizar un broadcast, en donde los paquetes llegan a todos los destinos posibles. Un broadcast utiliza siempre el mejor camino, pues utiliza todos los caminos haciendo un muy mal uso de los recursos de red. En este caso no sólo se da un mal uso a los recursos de comunicación, sino también a los recursos locales de cada SA, pues cada uno debe almacenar todos los documentos que recibe. Partiendo de este modelo de referencia, hemos desarrollado una variante en la que la relación de vecinos no es bidirecciónal. Conociendo la topología total de la DDN y los intereses de cada agente de servicio, se pueden construir cadenas de distribución eficientes. La performance de esta variante del modelo de referencia es muy buena, ya que presenta una gran robustez ante particiones de red (teniendo en cuenta también las grandes latencias de Internet).

Utilizamos este modelo para distribuir grandes volúmenes de información entre SA ubicados en España y Argentina. Dicha información resultaba imposible de ser transferida utilizando por ejemplo FTP, HTTP o replicadores Web, dada la alta latencia y alta proporción de paquetes descartados, producto de la congestión de las redes académicas de ambos países. Los documentos que transferíamos no eran tan grandes como para provocar este tipo de problemas ya que apenas excedían en tamaño a un Mb. y sin embargo la conexión FTP o HHTP se cerraba antes de poder transmitir el archivo en forma completa.

Capitulo 4

Esquemas de Caching

4.1. Introducción

En este capitulo se presentan los aspectos generales de los mecanismos de caching en Internet y se definen claramente las diferencias con las técnicas de replicación. Se realiza un análisis de los potenciales y las limitaciones que tienen las diferentes técnicas y esquemas de caching para brindar escalabilidad (en una dimensión) fundamentalmente a sistemas Web. Presentamos un esquema global de caching de servidores proxies cooperativos que utilizan comunicación IP Multicast, que prácticamente no introduce una sobrecarga de comunicación y trae aparejado los beneficios de los esquemas de caching cooperativos.

El crecimiento del volumen de contenidos, número de usuarios y la consiguiente demanda de ancho de banda en Internet, es una cuestión crítica para el desarrollo de las nuevas y de las existentes herramientas de acceso a los recursos (RDT), en particular el Web. Las técnicas de cache y replicación se han propuesto como una suerte "middleware" (en general la modificación en los servidores o clientes Web no es bien vista) para las herramientas existentes, tales como el Web.

Junto a los mecanismos de replicación, las técnicas de caching buscan lograr una mayor escalabilidad a nivel de usuarios, disminuyendo la utilización del ancho de banda de los enlaces, la latencia en el acceso a los recursos y la carga en los servidores Web.

En particular las técnicas de caching mejoran la perfomance de la interacción cliente-servidor, se basan en el principio de la *localidad de la referencia*, la cual asume que el objeto más recientemente accedido tiene alta probabilidad de que sea accedido en el futuro. La mayoría

de las RDT en Internet fueron diseñadas sin esquemas de cache[37] y menos aun de replicación, posteriormente algunas de ellas fueron dotadas de mecanismos ad-hoc.

Una esquema ideal de cache es aquel que maximiza la disminución de la utilización del ancho de banda, latencia y carga de los servidores, minimizando los costos de mantenimiento, administración del esquema, consistencia de los objetos, etc. . La eficiencia de un esquema de caching depende de múltiples factores (muchos de ellos dependientes entre si):

- política de reemplazo,
- topología y esquema de la caches,
- mecanismo de consistencia,
- "cacheabilidad" de los objetos,
- política de colocación ,
- quien inicia el proceso (servidor o cliente) y
- *fundamentalmente los intereses de la comunidad que comparte un sistema de cache (o las políticas de acceso a la información de dicha comunidad).*

En el presente capítulo analizaremos estos factores y las diversas propuestas existentes a los efectos de lograr la mayor eficiencia.

El comportamiento del los esquemas de caching los hemos validado fundamentalmente en el escenario que nos plantea la RIU y el comportamiento de una comunidad académica representativa, la Universidad de Buenos Aires (UBA).

4.1.1. Los inicios - Algunas Propuestas Experimentales

El surgimiento del Web y la apertura de Internet, a principios de los 90', a la iniciativa privada a producido un aumento del numero de usuarios. Estos factores contribuyen a un aumento en el tráfico de la red y carga en los servidores. Si sumamos que el ancho de banda *nunca* es suficiente, surge la necesidad de implantar esquemas de caching y replicación.
Es así que en el 94 surgen los primeros servidores cache entre ellos el desarrollado en Digital [Gla94], en el CERN [LB95], y el Unix Hensa Project [HEN97]. Sin embargo ya se preveía que estos servidores por si solos cache no eran suficientes, es por eso que surgen las primeras propuestas experimentales.

[37] Una de las excepciones , Si consideramos al DNS una RDT , dado que fue diseñado a los efectos de escalar correctamente . Antes de su introducción se utilizaba en Internet para resolver nombre , dirección IP , un archivo que se distribuía a todos los hosts (hosts.txt).

El Applied Network Research Group (ANRG) del San Diego Supercomputer Center, analizando los accesos logs de los servidores más populares (octubre1994), demuestran que un caching geográfico ayudaría a reducir el tráfico de la red, la latencia y la carga de los servers de una manera significativa.
ANRG sugiere que para ayudar a distribuir la carga la preferencia de un sitio cache para un cliente debe ser función no solo de la ubicación sino también del estado de la red y la carga del server . Por lo que propone dividir Internet en regiones geográficas y colocar Web Caches en cada región. También sugiere que una forma de resolver el problema de localización de un recurso es modificar el DNS[38], ellos sugieren adicionar la habilidad de devolver la dirección IP de la replica mas cercana de un recurso [39](cercanía puede incluir distancia física, numero de hops, latencia etc).

Estas áreas de investigación coinciden en parte con las propuestas por la National Science Foundation [FS94] (nuevas técnicas de organización de caches; división y distribución de los recursos existentes mediante sistemas distribuidos) que las considera críticas para las aplicaciones proyectadas en la futura Infraestructura Nacional de la Información (NII, National Information Infraestructure) de los Estados Unidos.

A su vez el Internet Architecture Board (IAB) propone las siguientes recomendaciones: Incrementar el estudio de arquitecturas de replicación y caching en general , amen de problemas de resolución de nombres de los recursos y articulación de una arquitectura común de seguridad para los sistemas de información. [RFC1862].

Muchos trabajos se impulsaron a los efectos de cubrir estas áreas de investigación y desarrollo, sin embargo a excepción de [MBC+97] y nuestro trabajo con un enfoque totalmente diferente [NR95] [LNR97], todos los cubren en forma parcial .

Desde los primeros trabajos experimentales a la fecha se produjo un fuerte desarrollo en los sistemas de cache[40] . Actualmente existen decenas de sistemas comerciales los cuales

[38] Actualmente CISCO esta desarrollando un protocolo que junto al DNS en función de las tablas de ruteo que manejan determinar la copia mas cercana que solicita el cliente que consulta al DNS .
[39] Unos de los primeros trabajos que presenta una taxonomía [GS95]
[40] Consideramos por la facilidad de instalación y administración que supone su uso exclusivo por una única comunidad , por ejemplo la UBA , un ISP o una empresa.

podríamos clasificarlos en : software-based proxy server [41] y hardware dedicado conocidos como "appliances" [42], los cuales tenderían a ofrecer una mejor performance.

Una explicación que podemos dar a que únicamente los sistemas de caching tengan tan fuerte desarrollo se debe a la necesidad fundamentalmente de los proveedores de servicio, ISPs a disminuir el tráfico de sus enlaces y a la administración centralizada que ellos requieren .

4.1.2. Cache y Replicación

Se puede considerar a los esquemas de caching y replicación (mirroring y distribución) como una suerte de "middleware" para los sistemas que se utilizan en Internet. En la literatura muchas veces no esta clara la diferencia entre caching y replicación siendo utilizadas en forma indistinta. Algunas diferencias que los definen son [RFC1862][W3096][LNR97] :

Caching

- Es un servicio con QoS, " best effort ".
- La copia de un objeto puede ser descartada en cualquier momento (política de remoción).
- Bajo demanda de los clientes (pull).
- Contiene un conjunto parcial de objetos.

Replicación

- es un servicio " garantizado " .
- la copia (*replica*) de un objeto es persistente
- iniciada por los servidores (push)
- Contiene volúmenes de objetos

---

1. CSM-USA (Salt Lake City), Deerfield Communications Inc. (Gaylord, Mich.), Microsoft Corp. (Redmond, Wash.), Netscape Communications Corp. (Mountain View, Calif.), Novell Inc. (Orem, Utah), and Ositis

[42] Cacheflow Inc. (Sunnyvale, Calif.), Cisco Systems Inc. (San Jose, Calif.), CobaltNetworks Inc. (Mountain View), Entera Inc (Pleasanton), Eolian Inc. (Fairfax, Va.), IBM,Infolibria Inc. (Waltham, Mass.), Inktomi Corp. (San Mateo, Calif.).

Dentro de los sistemas de replicación debemos distinguir al los sistemas de espejado (mirroring) donde se realiza un *replicación masiva* de grandes volúmenes de objetos a lo igual que News y a ODS en donde la replicación se realiza en función de grupos de interés oferta/demanda . En todo los sistemas de replicación los periodos de la persistencia de la replica están impuestos por la administración de los sistemas, en esos casos definimos a las copias como *replicas.*
Todos estos procesos de copia de un objeto traen aparejado mantener una garantía de consistencia con el objeto *master* (alojado en el servidor que lo creo). Tal garantía depende de la *granularidad de la sincronización* (mseg, seg., horas, dias,) que esta impuesta por la aplicación [43].

Si bien se realizaron propuestas como push-caching[GS95] y caching-in-advance[Bes95b] donde la copia de un objeto es iniciada por el servidor , este es iniciada en función de la demanda de los clientes (e.g. en función de las logs se determina la popularidad de un objeto, lo cual determina realizar una copia del mismo), es por ello que las consideramos sistemas de cache. Los mencionados mecanismos de cache se diferencian del sistema propuesto en [BBM+96][BBM+97] en el cual se copia un objeto en un grupo de servidores, en función de la popularidad del mismo, asociando a su vez un servicio de nombres [BMS96]. Es importante que estos sistemas manejan copias de un objeto y en ODS manejamos replicas.

A su vez nosotros dotamos a ODS de un subsistema [BM00] que permite determinar la popularidad de los recursos accedidos por una comunidad , a los efectos de ser ofertados a comunidades de intereses similares. La cota de popularidad que se determina buscar maximizar el ancho de banda ahorrado y las cantidades de solicitudes, minimizando el tráfico generado en ODS.

Uno de los problemas del Cache es que los objetos son tratados en la mayoría de los casos, democráticamente, ya que objetos muy importantes son limpiados (política de reemplazo) de la cache a favor de otros mas recientemente solicitados menos importantes.[44]

4.2. Cache en Internet

La mayoría de herramientas de acceso a recursos clasificadas como "ir y conseguir" [Wei94] se basan en un modelo cliente-servidor. En particular podemos ver al Web como un enorme

[43] Por ejemplo en Globe [STKS99] se define en cada objeto compartido distribuido, en cambio en W3Objects se define también de diferentes formas [Ing97].
[44] Menos importante, por ahora tiene un carácter subjetivo , en al apéndice I presentamos un survey de protocolos de remoción en lo cual los objetos no son tratados democráticamente.

Sistema de Archivo Distribuidos (DFS) [45] [DP96]. En [SS97] plantean que desde el punto de vista de los clientes el Web esta compuesto efectivamente por objetos read-only, con propagación perezosa (lazy) de actualizaciones desde los servidores a los clientes. Con lo cual naturalmente muchas de las investigaciones sobre caching en Internet han sido fuertemente influenciadas por los subsistamos de caching de los DFS. Por ese motivo analizaremos que relaciones pueden existir y si el paradigma de DFS es sostenible en el entorno de Internet.
La eficiencia de un DFS de gran escala depende de como el cache reduce la *carga en* los servidores de archivos [Bla92], este objetivo no es fundamental en los sistemas utilizados en Internet, donde la prioridad se centra en la disminución del ancho de banda utilizado y los tiempos de latencia [46]. Una cuestión a tener en cuenta es la granularidad en ambos casos, en un FS es a nivel de bloques , en el Web a nivel de archivos [47].

4.2.1. Caching en espacios FTP

Hasta principios de los 90 la herramienta por excelencia de acceso a los recursos era FTP[48], debido al gran tráfico generado se propusieron diversos esquemas de cache entre los cuales describiremos dos que han inspirado muchos de trabajos de caching en Web.

4.2.1.1. Alex Filesystem

Alex [Cat92] es un gateway FTP-NFS que permite a un servidor FTP remoto mapear en el Sistema de Archivos (FS) local que archivos pueden ser accedidos usando las operaciónes de archivos tradicionales.

Un servidor Alex que se comunica con un servidor remoto FTP mediante FTP, cachea los archivos localmente y se lo comunica al local FS a través de NFS. Cuando un usuario busca acceder a un archivo remoto, Alex transfiere el archivo usando FTP y entonces lo cachea localmente. En definitiva traslada instrucciones NFS a comandos FTP. El mecanismo de consistencia débil utilizado por Alex, es utilizado en diversos proxies web.

4.2.1.2. Caches Jerárquicos

[45] Amen de otras diferencias, un sistema de cache para paginas un SO la tasa de hit rate es del 99.99%
[46] Los proveedores de contenidos por su lado definen N servidores Web a los cuales se acceden mediante un único URL.
[47] No usaremos acá el concepto de objeto, objeto distribuido, objeto fragmentado.
[48] Un herramienta muy popular en ese momento era ARCHIE , la cual mantiene índices de los repositorios FTP. Posteriormente dichos servidores ARCHIE comenzaron a replicarse mediante esquemas manuales vía FTP mirror.

Danzing et. al. demuestran en [DHS93] que colocando múltiples caches FTP en puntos estratégicos del entonces Backbone NSFNET, usando una estructura jerárquica, el volumen del tráfico podía reducirse un 42%. A partir de este trabajo se penso que estructuras jerárquicas de cache podrían utilizarse en todo Internet. Ellos propusieron colocar los FTP cache en todos los puntos de unión entre redes así también entre backbones y redes regionales. Esta propuesta condujo además al desarrollo de un subsistema jerárquico de cache de objetos, para el sistema Harvest [49] [CDN+95].

4.2.2. Caching en la WWW

Las técnicas de caching pueden ser *bastantes efectivas* [50]para reducir la utilización de ancho de banda como también la carga en los servidores y el tiempo de latencia en el acceso a los recursos.

Uno de los problema del Web es que es por naturaleza ineficiente ya que se lo diseño sin esquemas de cache como los primeros DFS. Cuando N clientes "cercanos" solicitan el mismo objeto de un servidor, el mismo es trasmitido N veces, sin tomar en consideración la proximidad de los clientes (*localidad geográfica*). Esto produce mayor carga en el servidor y usa mayor capacidad de red de la estrictamente necesaria. Lo mismo sucedía cuando un cliente solicitaba el mismo objeto en diferentes instantes (*localidad temporal*). Posteriormente se implementaron mecanismos de cache en los clientes.
En cuanto a clientes *cercanos* [51]un método para reducir la carga del server y el ancho de banda usado fue desarrollo de los proxies caches [LA94] [Gla94] que permiten que muchos clientes compartan un cache . El proxy sirve los requerimientos del cliente siempre que sea posible, devolviendo los objetos desde su cache. Los beneficios del caching crecen con el número de clientes compartiendo el mismo sistema de cache (se plantea que a partir de 50 clientes).

Hemos comprobado, a lo igual que otros trabajos, que existe [52] un gran proporción de tráfico a causa de cache miss, sin embargo esos objetos pueden estar en servidores caches de otras comunidades. Si los clientes pudieran compartir esas caches se lograría disminuir la carga del

[49] A su vez este sistema de cache dio lugar al Squid y NetCache
[50] Mostraremos cuan efectivo y útil puede llegar a ser un middlweare de cache.
[51] Cercanos nos referimos a usuarios de una misma comunidad que utilizan un cliente Web, con el mismo enlace a Internet.
[52] Durante 1998 la tasa de miss rate en la UBA fue superior al 60%, aun con diferentes estrategias de reemplazo y espacio de memoria.

servidor donde se encuentra el objeto master [53] y el ancho de banda utilizado en enlaces de alto costo[54]; con lo cual es posible brindar mayor escalabilidad.

4.2.2.1. Estrategias de Caching

Las técnicas de caching se pueden clasificar en función que la demanda (iniciación del proceso) del caching provenga del lado del servidor web, cache o del cliente.

- *iniciada por el cliente* : las técnicas tradicionales de caching de área extendida son iniciadas por los clientes , cualquier requerimiento de un cliente a un proxy genera una copia del objeto.

- *iniciada por el server* : en estas técnicas que el caching es iniciado por el servidor Web. La decisión de caching es del servidor y puede ser función del conocimiento de la topología de red, patrones de acceso, popularidad, etc., para decidir copiar un objeto en servidores Cache. Como las propuestas en [GS95] [Best95] .

- *iniciada por el cache* : en este caso en función de los patrones de acceso se copia un objeto. Esto puede ser realizado mediante un configuración manual , cachear cada determinado periodo de tiempo un sitio web (una especie de pre-fetching[55] por ej. el de un diario) o en forma automática[56]

4.2.2.2. Perfomance

Las métricas mas usuales para evaluar la performance de los esquemas de caching son:

---

[53] Métodos de Round-robin permiten la selección de un proxie particular de un cluster.
[54] El concepto de alto costo es relativo. En Argentina fundamentalmente son los enlaces internacionales, pero existen enlaces nacionales en algunas provincias que tienen un alto costo. Todo esto gravita a la hora de definir un esquema de cache cooperativo. Mediciones en topologías jerárquicas han mostrado que el aumento de perfomance es del 10%.
[55] Caches como Netscape, NetcCache
[56] www.akamai.com

- Hit Rate.
- Hit Miss.
- Hit Byte.

Se dice que ocurre un "hit" en la cache cuando un cliente solicita un documento y debido a un requerimiento previo se encuentra en la cache. Ocurre un "miss" cuando el documento no se encuentra en la cache [57]. La métrica de Hit Byte surge debido a que es una medida que me permite computar el ahorro de ancho de banda efectivo [58], es la relación entre la cantidad de bytes devueltos por la cache y el total de bytes requeridos por los clientes (El hit rate muchas veces computa el acceso a documentos que no se transfieren totalmente al cliente, debido a que el usuario aborta el proceso, y la distribución de tamanos de los objetos no es uniforme).

Diversos investigadores y proveedores de servidores proxy presentan diversos valores de perfomance, pero no aclaran en si es producto trace driven[59] simulado, o en condiciones reales de funcionamiento y cuales son las condiciones de contorno. Por ejemplo Netscape asegura que con tamaño de cache de 2 a 3 GB y un único WEB proxy puede soportar miles de usuarios internos y que puede proveer un cache hit ratio tan alto como el 65 % [marzo95], por otro lado Microsoft asegura una perfomance desde 35 al 65%.

Sin embargo diversos trabajos plantean que la tasa de hit rate de un único proxy va del 30% al 50% [ASA+96] [CDN+95][Gla94]. En cambio en [CY97] se asegura que con un método de prefetching han logrado un hit-rate del 64 %. Mediante simulación y una cache virtual de tamaño ilimitado en [BBM+97] llegan a un hit rate 56.5% y 50.3% de hit byte.

Cabe destacar que todos estos valores de perfomace son para un único proxy accedido por una comunidad de clientes, o simulando el tráfico que generaría una comunidad en particular.
Mediante análisis de los logs [60]del servidor proxy (Netscape) de la UBA que además actúa como firewall[61], con lo cual el tráfico generado es representativo de toda la comunidad académica arrojan :

- 28 al 40 % Hit Rate
- · 12 al 18 % Hit Byte

[57] La utilización de cache trae aparejado un aumento de la latencia en los objetos que son accedidos una sola vez o aquellos que se solicitan por primera vez.
[58] Muchos autores consideran la tasa de hit rate equivalente al ahorro de ancho de banda .(gla94)
[59] Mediante la ejecución de solicitudes típicas registradas en los logs de alguna comunidad
[60] Los periodos analizados fueron julio a agosto de 1998.
[61] Mediante el filtrado de HTTP, port 80, en el router.

Estos rango de valores en la perfomance coinciden con los reportados[62], con plataformas de cache similares, en otras comunidades académicas de la RIU.

A los efectos de determinar si la perfomace dependía del sistema o de la comunidad analizamos los logs del proxy cache Squid con plataformas de hardware similares[63] (Agosto 1998). Los valores típicos obtenidos son los siguientes :

- ISP's : 50-55 % Hit Rate , 30-35 % Byte Rate
- Academic : 25 -35% Hit Rate, 10-20% Byte Rate

Pruebas posteriores (Noviembre1999) con diversas plataformas [64], cuyos autores reportaban diversos valores de perfomance , obtuvimos los mismos rangos de perfomance.

En consecuencia la eficiencia de un sistema de cache depende fundamentalemente de la comunidad a quien da servicio.

La efectividad de técnicas de caching iniciadas por el cliente para sistemas gran escala es bastante limitada comparadas cuando se las utilizada en otros áreas. Esto fue previsto por Glassman [Gla94] y confirmado por los estudios de Bestvros *et al* [BCC+95] y los realizados para proxy caching genérico por Abrams *et al* [ASA+95]. Básicamente la limitación en los proxy caching esta dada por el bajo nivel de documentos remotos que son compartidos entre clientes de un mismo sitio.

Como podemos ver las tasas de éxito son bajas , con lo cual tenemos con gran porcentaje de solicitudes que deben ser satisfechas desde el servidor original , con el agravante que esa solicitud ve incrementada su latencia por el proxy. La tasa de miss rate es función de tres parámetros: Tamaño de cache; Política de reemplazo; Política de colocación (cache iniciado por cliente o por demanda del servidor). A los efectos de mejorar los tiempos de respuesta constituye esencial tener en cuenta que los proxies puedan realizar un prefeching pero esto nunca logra bajar el miss rate más que en el caso de caches bajo demanda [Bla93].

---

[62] Primer Encuentro Nacional de Representantes Técnicos de la Red de Interconexión Universitaria. Facultad de Ingeniería. UBA, Buenos Aires Julio de 1998.

[63] Disco de 4 a 10 Gbytes. RAM entre 254 y 512 Mbytes.

[64] NetCache, Cisco Engine y Compaq

4.3. Localidad de la referencia

Dado que las técnicas de caching basan su funcionamiento en la localidad de la referencia, es importante ver que comportamiento tiene la comunidad en este aspecto y que relación existe con los DFS. En [ABC+96] se analizan los patrones de acceso mediante los logs de un servidor cache observando que estos exhiben tres propiedades de localidad de la referencia : *temporal, geográfica y espacial*.

- Temporal : que objetos accedidos recientemente es probable que sean accedidos en el futuro por un mismo cliente. En un cache en el cliente se ha reportado [BCC+95] [ABC+96] que la temporal (la única que puede ser explotada) es bastante limitada, con un tamaño de cache infinito el *byte rate* promedio llega al 36% y puede ser tan bajo como del 6 %. En Abd88 17%. En cambio en DFS esta localidad es muy importante lo cual hace efectivo el cache en los clientes (Cache hit rates en AFS > 96 % [SS94] , Client hit rates > 90% [Bla93]).

- Geográfica : un objeto accedido por un cliente es probable que también sea accedido en un futuro por clientes de una misma comunidad. En cambio aquí con un proxy uno esperaría que la perfomance se viese mejorada en un virtud de la propiedad de localidad geográfica . Se observa en [BCC+95] que para documentos remotos el grado en cual lo comparten los clientes es relativamente bajo. Han encontrado que el byte rate con tamaño de cache infinito va del 36% al 50% en el mejor de los casos. Mejor que una única comunidad como en [LA94] , se esperaría que muchas comunidades compartiendo proxies geográficos [GS95] se mejorase la performace En cambio con caches multi-level [MH92] en DFS tienen valor bajo. A lo igual que el acceso de datos en AFS (volúmenes) ocurre dentro de una organización , y el porcentaje de acceso remoto es insignificante , 5 % [SS94].

- Espacial : que un objeto vecino a un objeto recientemente accedido es probable que en el futuro sea también accedido. Cadenas de URL tienden a suceder en la misma secuencia cada vez [ABC+96], lo cual hace útil la utilización de mecanismos de prefetching (los cuales son complejos de implementar [65]). En cambio en DFS luego de un open viene un read con lo cual un simple mecanismo de read-ahead es útil.

[65] Algunos trabajos utilizan cadenas de Markov

4.4. Consistencia

La copia objetos en los sistemas de caching requiere del mantenimiento de la consistencia del mismo. El problema de mantener actualizadas las paginas Web en la terminología de los DFS es llamada *coherencia.* Sin embargo las técnicas de coherencia para DFS no se pueden aplicar en forma directa en los Web caches, ya que el web se enfrenta con las características propias de Internet .
Un mecanismo de consistencia debe proveer un reducción en el ancho de banda de la red y carga en el servidor a muy bajo costo y que retorne un bajo porcentaje de objetos desactulizados.

Diversos trabajos han evaluado el comportamiento de varios mecanismos de consistencia desde los *débiles* a los *fuertes.* Actualmente en los Web caches la consistencia se mantiene débilmente, aunque esto tal vez se deba a la creencia generalizada del costo extra que implicaría generar otros mecanismos.

Clasificamos los mecanismos de consistencia en función de quien es el encargado de iniciar el proceso de verificación de la consistencia de los objetos (a lo igual que en el área de DFS) :

- Conducida por el cliente
- Conducida por el servidor

4.4.1. Mecanismos de Coherencia en los DFS

Analizando los sistemas de archivos distribuidos mas populares vemos que dentro de las conducidas por los clientes tenemos:
La utilizada en Sun NFS [Lyo85]: cuando un cliente lee y cachea una pagina, este marca al archivo con un timestamp. Cuando el cliente usa la pagina cacheada un tiempo después, realiza un *chequeo de validación*, envía el timestamp al servidor para verificar que la pagina no ha sido usada desde que fue cacheada. Para mejorar la perfomance se realiza un chequeo de validación al menos una vez para una pagina dada un tiempo fijo . En Sprite File Systems [NWO88] , cuando un cliente open/close un archivo contacta al servidor . (similar a **polling-every-time)**
En cambio dentro de las Conducidas por los servidores encontramos:

callback newer (Andrew ,AFS [HKM+88]) : En estos files systems cuando un cliente recibe un archivo desde un servidor, este también recibe un *callback promise* y se garantiza que el servidor notificara al cliente si la pagina es modificadas. En definitiva el servidor promete notificar antes de permitir una modificación por cualquier otra estación de trabajo . Esta promesa llamada callback , se traduce en una considerable reducción del tráfico en la validación de la cache. (los archivos son servidos a los clientes - cache con el acuerdo que antes que otro cliente puede modificar el file , el server invalidara todas las otras copias cache , este tipo de esquema disminuye la comunicación entre el server y los clientes).Trabaja bien con R/W muy grande. Existe el concepto de volumen, que son frecuentemente R pero raro W, con lo cual generan r-only replicas en múltiples servidores. (Similar a Invalidación)

Como podemos ver el chequeo de validación con un tiempo de 3 seg y callback nos proveen un mecanismos de coherencia estricta, y hemos visto que estos mecanismos son demasiados costosos de implementar en redes como Internet.

4.4.1.1. Consistencia en el Web

Existen diversos mecanismos para mantener la consistencia de los objetos en Internet, en base a los trabajos de [GS96][Gwe95] [LC97] y [LC98] podemos presentar la siguiente clasificación :

Consistencia Débil

- *campos TTL (e.g. DNS, CERN Proxy)* TTL fijo. Clásico (basada en expiración)
- *client polling*
- *client polling (e.g Alex* modificado*)* con TTL adaptativo

Básicamente mecanismos basados en campos TTL, es aquel en el cual un cliente considera actualizado el objeto en la cache sino expiro su TTL. Cliente Polling , en el cual el cliente chequea periódicamente al servidor Web para verificar la actualidad de la copia.

Consistencia Fuerte

- *polling- every- time :* cada vez que un usuario solicita un objeto , si hay copia en la cache el proxy contacta al servidor web para validarla. Si es valida la devuelve al cliente.

- *protocolos de invalidación :* Un servidor mantiene una lista de todo los clientes que tienen cacheados objetos , cuando este cambia se les envía un mensaje de invalidación a los clientes.

La mayoría de los Proxies Web existentes utilizan la combinación de los dos primeros mecanismos de consistencia débil. Cada copia es considerada valida hasta que vence su TTL (el campo expires) en cuyo caso el próximo request realiza un polling mediante el envío de un " if-modified-since " (IMS) para verificar si la copia aun sigue siendo valida.

Una consistencia débil debería ser aceptable para el Web ya que los objetos tiene un comportamiento similar a los manejados por DFS o FTP: el archivo mas viejo, es el que tiene menos probabilidad de ser modificado. Desde este único punto de vista ambos tienen una distribución bimodal.
Por lo tanto al archivo mas viejo que el servidor proxie cachea, le corresponde la menor frecuencia de chequeo de consistencia. Esto es muy eficiente comparado a chequear en cada requerimiento, y el cliente debería poder forzar un chequeo si le resulta esencial utilizar la última copia. Esto es usado por Alex (FTP-NFS gateway) y otros.

Client polling es una técnica donde los clientes periódicamente chequean con el server si los objetos cacheados siguen siendo validos. Una variante , *TTL adaptativo* es la usada en Alex FTP cache [Cat92] y se basa en asumir que los archivos mas jóvenes son modificados mas frecuentemente que los viejos y que estos tienden a ser menos modificados.
Lo cual implica que los clientes necesitan realizar menos poll para los objetos mas viejos. Este protocolo particular adoptado por Alex utiliza un umbral de actualización (update threshold) para determinar la frecuencia de poll al servidor (el mismo se expresa como un porcentaje de la edad del objeto). Un objeto es invalido cuando el tiempo desde la ultima validación excede el umbral de actualización. En definitiva el administrador de la cahe asigna un atributo TTL que es función de un porcentaje de la edad del mismo (Time_current – Last-modified [66])

Por ejemplo para evitar un polling excesivo, Alex solamente garantiza que el recurso no será nunca mas del 10% fuera de la edad reportada (umbral de update). Es verdad que durante dicho periodo el objeto pudo cambiar , pero a lo igual que los campos TTL no soporta consistencia perfecta.

[66] Esta información es provista por el header de replay HTTP.

De la misma manera que con TTL, client polling puede implementarse fácilmente en el protocolo HTTP actual. El campo del header de solicitud "if-modified-since" indica que el server devolverá el documento solicitado únicamente si el mismo cambio desde la fecha especificada. La mayoría de los proxies actualmente usan este campo.

Los protocolos de invalidación son usados cuando una consistencia débil no es suficiente, como hemos visto varios DFS lo usan, cuando el archivo es modificado el server notifica a las caches que las copias ya no son validas. Tratan de esta manera de asegurar que nunca una copia en la cache este desactualizada. Estos protocolos mantienen una lista de los datos cacheados a los efectos de notificar a las caches. El problema de estos protocolos es que son costosos. El mantenimiento de la lista trae problemas de escalabilidad a menos que se utilice un esquema de cache jerárquico. Además ante particiones de red, caídas de clientes, el servidor tratara de notificar la invalidación de objetos. Por otro lado los protocolos de invalidación requieren de modificaciones en el servidor, mientras que los otros protocolos pueden ser implementados a nivel de web-proxy .

4.5. Políticas de reemplazo

Uno de factores críticos en la performance de la cache es el algoritmo de reemplazo utilizado. Los algoritmos de remoción de la cache usualmente maximizan la tasa de cache hit intentando dejar en la cache los objetos que son mas posibles de ser referenciados en el futuro lo cual resulta muchas veces difícil de predecir. Algunas de las políticas de remoción utilizados son las mismas que se utilizan para caching de file system , memoria y discos como las presentadas en . Sin embargo la aplicación directa de estos políticas nuevamente no tienen en cuenta las características de la comunidad de usuarios de Internet y el master objeto a cachear esta en millones de servidores distribuidos geográficamente.
Con lo cual la política a utilizar debería ser mas compleja que la utilizada para memoria virtual y cache de memoria del computador .
Existen además en los caches web *políticas de admisión*[67], con lo cual varios algoritmos de remoción se complementan con algoritmos de colocación.
El estado de la cache podría verse como un conjunto de documentos almacenados en ella y para cada documento un conjunto variable de estados que típicamente incluyen información estadística asociada con el documento. Algunos ejemplos de estados variables asociados con el documento son: el tiempo desde la última referencia al documento, el tamaño del

[67] A partir de configuraciones de sistema de cache se pueden excluir cachear determinados objetos según diferentes criterios

documento, el tiempo que tomo recuperar el documento, el tiempo de vida del documento, etc.
Lo mas conveniente sería tener una cache tan grande como sea posible y de esa manera no tener que remover ningún documento. Sin embargo la realidad es que el tamaño de la cache es limitado[68] .
Además los métodos tradicionales para caching de los file system no son buenos para el tráfico de la Web porque: granularidad de caching es diferente: la política de los métodos de file system tienden a hacer cache de bloques individuales; la Web debe hacer cache de archivos, debe traer el archivo con un acceso a disco mientras que el file system necesita traer el archivo de a un bloque a la vez. Intervención del sistema operativo: el Web server tiene un overhead por la llamada al file system para abrir y leer el archivo, aunque tenga al archivo en cache. Los documentos de Web son accedidos casi siempre para lectura solamente.

4.5.1. Descripción de algunos algoritmos de remoclón.

Los algoritmos de reemplazo en particular, deben diseñarse de acuerdo a los objetivos generales de los servidores web . Sin embargo la mayoría de los algoritmos de remoción propuestos se direccionan solo a minimizar el ancho de banda utilizado, carga en los servidores web dejando de lado minimización los tiempos de latencia para acceder a un objeto (Apendice 1) .

4.6. Esquemas de Cache

Muchas de las sistemas de caching iniciada por el cliente consisten en que los clientes compartan un único servidor proxy (estructura planas como en muchos DFS), pero este esquema presenta los siguientes inconvenientes: no es tolerante a fallos, se produce un cuello de botella (creando un limite en el número de clientes que pueden compartir una cache), se limita la tasa de hit que cada cliente puede obtener .

Además un único proxie no ayuda a comunidades cercanas, como el cache en el cliente a los clientes de la misma comunidad. Un cliente cache no reduce el tráfico en su comunidad y un proxy cache no reduce el tráfico entre comunidades cercanas.

[68] Muchas veces esa limitación no esta impuesta por el espacio en disco , sino por cuestiones de diseño tal es el caso del Squid o NetCache.

Una solución es la que permite a los clientes compartir el espacio de cache de servidores proxies cooperativos. Se deben usar técnicas que permitan que múltiples caches proxy cooperen y compartan sus caches[69] incrementando así la tolerancia a fallos, escalabilidad y perfomance[70].

Clasificamos los esquemas de cache en :

- Locales : son los que brindan servicios a una comunidad (ej. usuarios dial up, la comunidad de la UBA) y los servidores cache están conectados fuertemente.

- Globales: son los que brindan servicios a diversas comunidades y los servidores están conectados débilmente.

Muchos de los trabajos de esquemas de cache se han inspirado en los realizados en el área de DFS.
En el área de DFS se ha demostrado la eficacia de esquemas de cache cooperativos [DWAP94], dentro de una misma organización. El sistema xFS [WA93] desarrollado en Berkeley se le propone un esquema de cache cooperativo [DMW+94][DMW+94a] que es similar al adoptado en [Gwe95][GS95], con la diferencia que se distribuye funcionalidad de los servidores entre los clientes (eg mantienen las copias master) y la motivación fue mejorar el desempeño ante el avance de las LAN's de alta velocidad.
Muntz y Honeyman en [MH92] demuestran que la perfomance del sistema es mejorada por los niveles intermedios de AFS cache debido a la habilidad jerárquica de reducir los access rates para los servidores. Sin embargo le llamo la atención el bajo hit-rate, que es mas bajo lógicamente[71] del encontrado por Danzing et al. [DHS93].

Blaze [Bla93] también simula multi-level caching para AFS las mismas muestran que el tráfico puede reducirse a la mitad. En cambio en [DSH93] estiman que el cache jerárquico propuesto llega a disminuir el tráfico en un factor de 10. Blaze direccionó el problema de

---

[70] Como beneficio adicional que se consigue es que cada cliente tenga la efectividad máxima sobre la cache; sobre un gran número de clientes compartiendo una cache mayor será la probabilidad de obtener un cache hit.

[71] Desde ya los archivos o/write clientes compartidos es muy bajo a lo igual que en el trabajo de Blaze [Bla93].Fundamentalmente se encuentra una disminución en la carga de los servidores. Los trabajos de Danzing ,donde con cache infinito, los clientes tienden a compartir un gran numero de archivos o/read.

Cache en large scale systems proponiendo también un esquema cooperativo el cual soporta un suerte de replicación cuando la demanda es alta y porque los clientes usan cualquier cache cercano de otros clientes. Cuando un server recibe muchos request deja de servir el file y devuelve un lista de los clientes que los accedieron [72](y por lo tanto cachearon el file). En estos trabajos con DFS los clientes tienen la funcionalidad que pueden ser accedidos por otros clientes , a diferencia de los clientes Web.

Protocolos de cooperación

Definimos a los protocolos de cooperación como aquellos que permiten que ante un requeriemiento de un cliente a un cache que no pueda responder , redirecciónar el pedido a otros caches del esquema. El primero de dicho protocolo ha sido ICP(Internet Caching Protocol) (Standard IETF) en su version unicast para esquemas globales, el propuesto en [MLB95] multicast para un equema local y el que hemos propuesto multicast para un esquema global..

4.6.1. Global Jerárquico

Harvest [BDH+94][BDM+95] que es un sistema de descubrimiento de recurso en Internet. Esta compuesto por diversos susbsistemas , entre ellos el de cache. A partir de este sistema y de las experiencias obtenidas con un esquema de cache-FTP jerárquico [DSH93], el grupo de Danzig et. al. propone un esquema jerárquico de servidores proxies a lo largo de backones regionales. Denominaron a este sistema Harvest Cache [CDN+95][CDN+96].
Harvest provee un mecanismo para compartir las caches de múltiples organizaciónes formando una estructura jerárquica. En definitiva un esquema cooperativo de caches jerárquicos para comunidades que comparten los recursos de backbones y redes regionales. Esta jerarquía establece relaciones Padre-Hijo, y hermanos . Los mismos cooperan mediante un protocolo que ha pasado a ser un standard para la cooperación entre servidores cache, ICP.

Los clientes WWW son configurados para usar una cache de un Servidor Proxy en particular usando el protocolo HTTP-proxy estándar. Si un requerimiento no puede ser resuelto por él, entonces pregunta a su padre y hermanos usando ICP[73] . La cache toma el objeto del primero que responda que lo tiene de entre el servidor dueño del objeto, su padre o sus hermanos. En el caso de que el padre o todos sus hermanos hayan respondido negativamente se contacta a otra cache padre (ya sea usando el protocolo HTTP-proxy o el ICP) y el proceso se repite. Así

[72] De alguna manera esto inspiro también el trabajo [GS94] de Gwertzman y Margo.
[73] ICP utiliza datagramas UDP

continúa el proceso en forma recursiva hasta que el requerimiento se ve cumplido (ya sea por un padre, un hermano o por el servidor dueño del objeto).

De esta forma, el arreglo de caches en forma jerárquica puede reducir la demanda de ancho de banda en redes WAN, reduciendo transferencias redundantes de información en sucesivos niveles de la red. Esto es una mejora sobre el cacheo a nivel organizacional, ya que no sólo reduce el nivel del tráfico en el gateway de la organización misma, sino en todos los gateways hasta el nodo raíz de la cache. Más aún, la implementación también puede ayudar a reducir la carga en los servidores más accedidos, decrementando en gran medida el número de acceso a estos nodos.
Una de las desventaja del diseño de cache jerárquica es que usan comunicación "unicast" con sus padres y hermanos; sería mas eficiente que usaran comunicación multicasting como la que hemos utilizado. Además la latencia de acceso a un objeto se ve incrementada, por primera vez. En realidad las características de escalabilidad de ICP es pobre [WC97].
En cuanto a los niveles de la jerarquía en [BBM+97[a]] muestra que con tres niveles las demoras son aceptables.

Diferentes iniciativas a nivel mundial han comenzado a utilizar esquemas globales jerárquicos de cache cooperativos. Dichos Sistemas de cache han sido diseminados para interconectar backbones nacionales , regionales y caches locales, creando en definitiva una estructura global de caching . Entre los varios proyectos de Web Cache podemos destacar el NLANR (National Laboratory for Applied Network Research) (USA) ; CHOC Project (Europa) ; National JANET Web Cache (Reino Unido) ; NZIX Cache (Nueva Zelandia) .

4.6.2. Global Multicast

Al tratar de implementar un esquema de cache jerárquico, en una estructura de red plana como el de la RIU no resultaba natural. A tales efectos presentamos un esquema simétrico de servidores proxy cooperativos , cuyas requerimientos se llevan acabo mediante IP Multicast [PMR96]. A tales efectos utilizamos CERN httpd 3.0 [LB95] que puede ser configurado como Servidor Proxy Cache y que permite configurarse para redireccionar un requerimiento a otro servidor Proxy (opción outer proxy)
Un trabajo relacionado con el nuestro es el presentado en [MLB95], que propone un esquema local con modificaciones en los clientes . Una de las grandes problemáticas planteadas a los efectos del desarrollo del esquema fue que protocolo de ruteo multicast seleccionar en función de la conectividad débil de la RIU y los routers disponibles.

Los protocolos de ruteo multicast más difundidos son: DVMRP (Distance Vector Multicast Routing Protocol) MOSPF (Multicast Open Shortest Path First) PIM (Protocol-Independent Multicast)
Estos, como cualquier otro protocolo de ruteo multicast, siguen generalmente uno de dos enfoques básicos dependiendo de la distribución de los miembros de los grupos multicast [SM97].
El primer enfoque asume que los miembros de los grupos están densamente distribuidos a través de la red y que la mayoría de estos están interconectados con gran ancho de banda. Este enfoque es llamado generalmente "dense-mode" y, en este modelo, están incluidos los protocolos DVMRP, MOSPF y PIM-DM (Protocol Independent Multicast - Dense Mode). Por el contrario, el segundo enfoque asume que los miembros de los grupos multicast están esparcidos por la red y tenemos conectividad débil . Este enfoque es llamado "sparse-mode" y es importante destacar que no implica que los integrantes de los grupos sean pocos sino que ellos pueden estar dispersos. En estos casos, técnicas de inundación como las que se utilizan en el caso anterior podrían realizar un mal uso del ancho de banda disponible y tener una baja performance del sistema. Estos protocolos de ruteo utilizan algunas técnicas y algoritmos más selectivos para obtener los árboles de ruteo multicast y dentro de ellos se encuentran el CBT (Core Based Trees) [Bal95] y el PIM-SM (Protocol Independent Multicast - Sparse Mode) [EFH97]. Nuestra elección se encontró limitada al segundo grupo de protocolos de ruteo ("sparse-mode") y al equipamiento de comunicaciones disponibles en la RIU para la interconexión de las redes. Este equipamiento soporta el protocolo PIM-SM y, por otra lado dicho protocolo es independiente del mecanismo provisto por cualquier protocolo de ruteo, es decir, que puede trabajar sobre cualquiera de los protocolos de ruteo IP unicast existentes.

En los protocolos denominados "sparse-mode", los routers que poseen miembros de un grupo multicast directamente conectados a él necesitan enviar explícitamente un mensaje ("join message") para unirse al denominado árbol de distribución de mensajes "sparse-mode". Si un router no es parte de este árbol predefinido, éste no recibirá tráfico multicast dirigido al grupo interesado. En contraste, los protocolos de ruteo "dense-mode" reenvían el tráfico multicast sobre todas las interfaces hasta que un mensaje denominado "prune message" le es enviado. De esta manera podemos ver que en los protocolos de ruteo "dense-mode" la acción por defecto es reenviar el tráfico multicast, mientras que en los protocolos "sparse-mode" la acción por defecto es bloquear el tráfico a menos que este sea solicitado explícitamente.

Los protocolos "sparse-mode" (tanto el PM-SM como el CBT) utilizan el concepto de RP ("rendezvous point"). Bajo este esquema, uno o más routers son designados como RPs y sólo ellos son los que obtienen la información acerca de quienes envían mensajes multicast, y ponen esta información disponible para los potenciales receptores. Cuando alguien desea

enviar un mensaje, primero debe enviarlo a algún RP y cuando un receptor desea recibir un dato, éste debe registrarse primero con un RP. De este modo, el flujo de datos va de emisor a RP y de RP a receptor y en este camino, los routers van optimizando las rutas eliminando saltos innecesarios. Esto requiere que se mantenga algún estado previo al arribo de paquetes. Los protocolos "dense-mode" no mantienen ningún estado previo al arribo del primer paquete.

Implementamos multicast solamente en el protocolo de cooperación entre los proxies sin realizar modificaciones del lado de los clientes. El cliente envía cada requerimiento a un servidor proxy especificado en su configuración (1) y si el servidor posee el objeto en su cache local, se lo envía (6). Caso contrario, si el servidor no tiene el objeto, lanza una consulta multicast para averiguar si alguno de los servidores que pertenecen al grupo cooperativo lo posee (2). Si al cabo de un tiempo no se recibe respuesta, obtiene la información directamente del servidor origen y se la envía al cliente (6). Si alguno de los servidores que pertenece al grupo posee el objeto, éste le envía una respuesta afirmativa (3) con lo cual el servidor contactado inicialmente se comporta como cliente, realizando el requerimiento al servidor con el objeto cacheado (4), recibiéndolo (5) y enviándoselo al cliente (6). El servidor deja cacheado el objeto en cuestión. En nuestro modelo se toma como respuesta valida la primera que llega.
La implementación de este esquema no requiere realizar ningún tipo de modificación en los clientes, con sólo configurarlos para que utilicen proxy. De esta manera, el cliente en forma transparente accede a un conjunto de servidores caches.

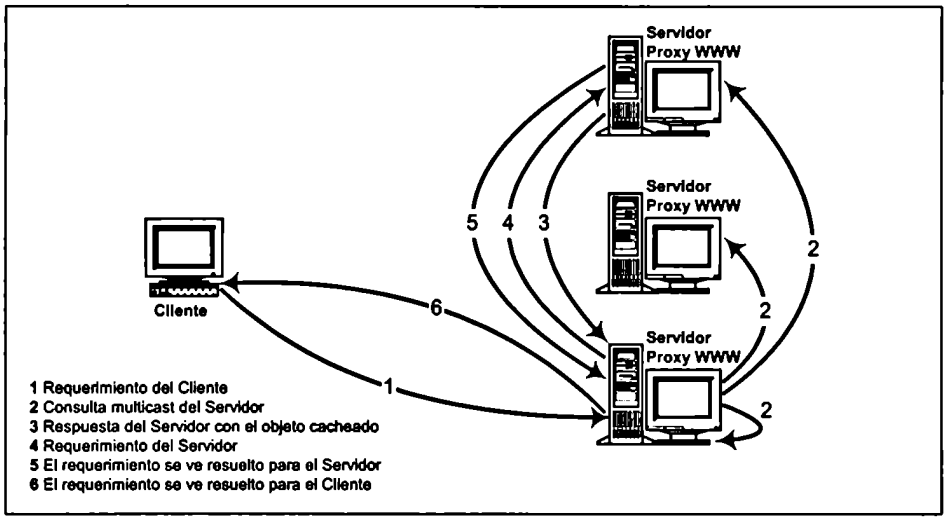


Fig. 1

Las diferencias en la implementación que se presentan con [MLB95]: para cada requerimiento, un cliente selecciona aleatoriamente un proxy cache server de una lista de servers cooperativos y envía su requerimiento a dicho servidor (*proxy master* de ese requerimiento). contrario este realiza un consulta multicasting a los otros s. Si alguno de los proxy tiene el objeto cacheado, ese proxy cache informa al master, así el master puede redireccionar al cliente a ese proxy cache. El cliente entonces hace un nuevo requerimiento del objeto a ese proxy cache, y obtiene el objeto. Si bien son mínimas fue necesario que realizaran modificaciones en el cliente. Concluyen en sus experiencias realizadas con servidores cooperativos en un *campus* que la utilización (como es lógico) de técnicas multicast no aumenta el significativamente el ancho de banda usado .

La desventaja frente a la propuesta de Malpani es que la conexión por parte del cliente a un servidor en particular no provee mayor robustez frente a la posibilidad de fallas, ni escalabilidad en lo que al balanceo y distribución de carga en los servidores se refiere. Sin embargo consideramos que, hoy en día, existen mecanismos y herramientas para que, en conjunto con el esquema que planteamos, se puedan obtener la robustez y escalabilidad deseadas y necesarias

4.6.3. Esquemas Globales conducidos por el servidor

Gwertzman y Margo Seltzer proponen en [Gwe95] [GS94] una técnica de cache iniciadas por los servidores web en función de la historia de accesos de un objeto y del ahorro de ancho de banda que implicaría cachearlo. Básicamente esta orientada a disminuir la carga en el servidor Web.
En esta técnica se utiliza el conocimiento de la topología de la red, geográfica y los patrones de acceso para minimizan el tráfico de la red y la carga en los server. Su arquitectura se inspira en las de Blaze y xFS. En push-caching los servers registran la historia de los accesos y usan esta información para decidir donde colocar las copias de los recursos. Su visión es la de una infraestructura de red con miles de push-cache servers en los cuales los objetos pueden ser pusheados.

Por simulación trace-driven concluyen que la combinación de la técnica de push-caching, con las iniciadas por el cliente es la mas adecuada [74]. Un server tiene la habilidad para monitorear los requests y decidir donde y cuando (definen un umbral de replicación [75], push-threshold) los objetos deben ser cacheados.
Un servicio centralizado de registros tiene la lista de push-servers, que permite a los servers decidir donde replicar los objetos por la provisión por demanda de una lista de push-cache server. La política de remoción en los push-servers, es la de documentos mas populares reemplazan a menos populares (least popular). En su trabajo no queda claro como determinan cual es el umbral para considerarlo popular a un documento.

Bestravos propone en [Bes95a] [Bes95b] un protocolo para "descargar", reducir la carga, servidores muy populares por duplicación (en otros servidores) de un pequeño porcentaje de los objetos que estos proveen. Sin embargo la idea original la publica en [Bes95] donde propone reducir el tiempo de respuesta mediante el caching a nivel de aplicación antes que caching a nivel de file systems. Propone un sistema que permita lo que denomina *replicación por demanda especulativa.* Plantea que ofrece una solución global la cual "replica" información multimedia en base a "supply/demand", basado en la oferta y la demanda, por la diseminación desde los productores a los servers que están cercanos a los consumidores. El nivel de diseminación dependerá del nivel de popularidad de documento.

---

[74] Ellos usaron una simulación *trace-driven* de Internet para evaluar las distintas estrategias de caching (cliente). El *cache cliente* reduce la carga del servidor y el ancho de banda por encima de un 30 % y el *server cache* un 20 % el ancho de banda y un 10 % la carga del servidor. (no queda claro si la combinación de ambas produce un efecto lineal).

[75] Mediante simulación trace-driven determinaron un valor de umbral razonable. Aunque esperan que este sea dinámico.

4.6.4. Locales

Los esquemas locales están orientados fuertemente a la tolerancia a fallas y al Balanceo de Carga , generando una suerte de Clustering de Servidores de cache. Mediante este esquema un conjunto de servidores se comportan como un único servidor web distribuido o proxies caches.
El balanceo de carga es una cuestión importante cuando un sitio popular no puede satisfacer los pedidos de miles de clientes con un único servidor[76] . A tal efecto esta técnica consiste en desplegar un grupo de servidores (cluster) que trabajen conjuntamente como un servicio de un simple host. Solo debe publicarse una dirección para todo el cluster.
La distribución de los pedidos a múltiples servidores requiere de artificios a nivel de red tal como round-robin DNS [AL92] o network address translation basadas por ejemplo en las estrategias propuestas en [EF94][Cisco96] [Dam97] para crear la ilusión de un único espacio de URL en el top de un set de servidores distribuidos.

Existen dos técnicas en uso [ODA+97] : por Software y basadas en ruteo (podría decir stwiching). Dentro de los por SW tenemos : autoconfiguración de proxy y redirección Server –side. Los basado en router : Cache director y content-based redirection.

4.7. Perfomance de esquema globales

La utilización de nuestro esquema global de cache (mayo1997) con cuatro servidores cooperando en los cuatro nodos principales de la RIU arrojaron que aproximadamente un 7 % eran respondidas por alguno de los ellos. Estos valores están relacionados con los reportados en ODAY97 en el cual con esquemas globales jerárquicos un 10 % eran resueltos por los vecinos. Si bien por simulación en [KS98] reportan que la cooperación puede mejorar el hit rate desde un 0,7% a un 28,9% . Preferimos quedarnos con los valores obtenidos en entornos reales , asumimos un 10%.
La pregunta es si este 10% justifica la utilización de esquemas cooperativos , en cualquier entorno de red . En redes como las existentes en USA tal vez no , pero en redes con un ancho de banda tan escaso como la nuestra se justifica.

[76] Por ejemplo en Octubre del 94, el servidor de la NCSA tuvo mas de 300 accesos por minuto para hacer frente a esa carga los archivos se pusieron en tres grandes back-end servers, de que AFS automáticamente hace cache desde el front-server a los back-end sistemas , un hacked -up DNS server maneja distintas direcciones IP por cada lookup de www.ncsa.uiuc.edu [KBM94].

Es probable que el beneficio de esquemas cooperativo de cache en ambientes comerciales no sea tan alto, pero por razones obvias no hemos podido verificar esto

4.8. Reemplazo del WEB por DFS

A lo largo del capitulo hemos realizado comparaciones con los DFS, ya que el Web se comportaría como un enorme DFS. Para que un middleware de replicación y caching, si los DFS actuales tienen el tema resuelto[77]. Porque no aplicar directamente el paradigma de los DFS a escala global en Internet.

Es indiscutible que DFS como los descriptos en SatyaBuscar Scale DFS, [HKM+88]], NFS, LanManager y Netware (NOS) tienen una amplia aceptación en áreas locales para compartir datos. El orden de escabilidad ha sido de 100 para NFS , 1000 en xFS [Buscar] y 100.000 para AFS [SS94]. El crecimiento de la interconexión de redes llevo en forma natural que muchos autores hayan tratado de aplicar el paradigma de los FS a muy gran escala .

En [SS94] se examina a AFS como un proveedor de Wide -Área files services sobre cientos de organizaciones en el mundo (buscan dar respuesta dando un punto de vista basado en la caracterización de uso de AFS) Analizando los motivos se plantean si el paradigma del DFS es sostenible a muy gran escala . En otras palabras cuan grande puede ser un DFS teniendo en cuenta la simplificación de la administración del sistema, poder soportar un efectivo comparición de datos y preservando la transparencia. Tratan a partir de la experiencia con AFS dar respuesta a estas preguntas. Aseguran AFS es el mas grande DFS que esta funcionando seriamente [SS94].

En [SBS94] propuso usar AFS para replicar los recursos de WEB, reemplazar HTPP por AFS (u Sin embargo llega al punto, como es lógico del reemplazo de protocolos tan difundidos como FTP y HTTP, cuando la idea de nuestro sistema es precisamente hacer uso de ellos y de futuras mejoras en http. Por otro lado plantean que el paradigma DFS es sostenible a muy gran escala.
En [SBS94] demuestran que las funcionalidades del Web se pueden mantener almacenando documentos en AFS . Una simple modificación en los browser permitiría recuperar un documento a través de un files systems de área global. Esto permitiría construir un servicio mas escalable . Estudios preliminares muestran que la velocidad de transferencia es mucho

[77] los mecanismos de replicación, caching y concurrencia embebidos

mayor que en AFS que con HTTP [SBS94]. Un administrador puede replicar un volumen popular en varios servidores de archivos para distribuir el acceso, pero no explican cual es el criterio a utilizar para la replicación.

Justifican con una serie de observaciones su utilización en grandes áreas. Pero la gran pregunta, puede este modelo ser utilizado en Internet. Desde ya ven el problema de la escalabilidad y la falta de caching, replicación, transparencia en la locación y mecanismos de autentificación y protección . El dramático crecimiento de datos compartidos, incrementos en la carga de la red y server , alta latencia y inadecuada seguridad .

Nuevamente aquí no se tiene en cuenta las características que implican el concepto de global de Internet y las semánticas de cada aplicación. La semántica del Web puede no concordar con la de DFS, los sistemas de archivos necesitan crear rápidamente y borrar archivos muy pequeños que nunca serán compartidos por varios usuarios. En cambio en el Web se necesita proveer rápido acceso a muchos archivos a través de toda la Internet, los cuales son compartidos por miles de usuarios y pocos de esos archivos son locales .

El gran problema es que proponen el reemplazo de HTTP como protocolo de comunicación entre los clientes y los servidores, una situación que consideramos imposible de llevar a cabo en la Internet actual. Finalmente [SS96] no plantean ya el reemplazo de HTTP por AFS y llegan también a la conclusión que a lo sumo se pueden usar como tecnologías complementarias.

Durante los inicios de los trabajos presentados en esta disertación exploramos todas las técnicas bien conocidas utilizadas en el área de DFS, sin embargo su aplicación directa a Internet no es factible, toda la experiencia obtenidas en esta área si bien no se aplicaron permitió el desarrollo de un gateway entre NFS y CODA [RSR99].

4.9. Conclusiones

Los sistemas de caching están orientados a reducir el ancho de banda utilizado la latencia a recursos frecuentemente solicitados en determinados periodos . Su eficiencia depende fuertementemente de las comunidades que hacen uso de el .

Un esquema global con protocolo de cooperación multicast no introduce demoras como el impuesto en ICP [78].

Es probable que el beneficio de esquemas cooperativo de cache en ambientes comerciales no sea tan alto, pero por razones obvias no hemos podido verificar esto.

No existe técnica de cache que permita satisfacer todas las restricciones, con lo cual se debe complementar con mecanismos de replicación.

Los sistemas de caching locales en comunidades de ISP proveen un fuerte reducción en la latencia y reducción de tráfico en los enlaces.
En comunidades académicas si bien esta reducción es sensiblemente menor , su utilización se justifica por los escasos recursos.

En el caso de Redes Académicas, RIU , Red Iris o Internet2 para el tráfico de calidad no garantizada es necesario la utilización de esquemas globales de cache cooperativos con protocolos de cooperación Multicast .

[78] Durante 1999 en Internet2, la red vBNS tienen un esquema global de caches que usan multicast para enviar ICP.

Capitulo 5

Invariantes y Escenarios

5.1. Introducción

Dado que los logs de los servidores Web como los sistemas de Cache proveen gran información sobre los hábitos de las comunidades de usuarios que consumen y crean recursos, innumerables trabajos han buscando determinar invariantes o patrones regulares tales como ranking de popularidad , vida media de un objeto, tamaño medio, etc . Dicho patrones de comportamiento permiten definir un escenario para analizar la escalabilidad .

La estrategia de diseño de cualquier sistema que ayude a superar las fallas de la escalabilidad en Internet debe ser función del comportamiento de la comunidad de usuarios . Dicho sistema será aplicable a una comunidad en particular o permitir funcionalidades que se adapten a otras comunidades de usuarios.

5.1.1. Estrategias de Diseño, función del comportamiento de la comunidad de usuarios

En el área se sistemas esto es una practica *común* [79], una de las tareas fundamentales en el diseño de DFS ha sido el analizar el comportamiento de la comunidad que utiliza dicho sistema, tal es el caso de Satya tuvo en cuenta con el diseño de Andrew, AFS y CODA.

Un problema siempre presente en las mediciones de cualquier sistema existente es saber que tan típica es la población observada [Tan95]. Las mediciones de Satya fueron realizadas en el ámbito de la universidad, cabe preguntarse si es extrapolable a otras comunidades (eg empresa). Otra pregunta es las conclusiones del análisis no se encuentra sesgado por condiciones de contorno, en un ambiente MS-DOS uno podía decir que los usuarios no usan nombres de archivos de mas de ocho caracteres [Tan95]. Con el mismo criterio se puede

[79] Aunque muchas veces el usuario se deba adaptar al sistema.

plantearse cuan representativos son los logs de un proxy de una comunidad académica o los de proxies de los ISP´s.

En sistema Unix (comunidad universitaria) Satya [Sat93] encontró los siguientes invariantes : archivos a menores 10 K (Tan, Mulle. 84 lo mismo) , Tiempo de vida cortos, R y W son secuenciales , raro el acceso aleatorio, Poco usual compartir archivos[80], Procesos promedio uno solos cuantos archivos, R es mas común que W. (alto relación R/W) etc . A partir de ellos desarrollo las estrategias de diseño de todos sus DFS.

La pregunta es si estos invariantes pueden aplicarse a cualquier sistema distribuido. Blaze [Bla93] para diseñar su DFS de gran escala, previamente registro las trazas de un NFS server para determinar la estrategia de cache optima . Una de las principales observaciones es que los archivos tienden a mostrar una fuerte *inercia*, esto es un mismo archivo tiende a ser abierto nuevamente y de la misma forma por un mismo usuario. Blaze también encontró que el archivo mas viejo es el que tiene menos probabilidad de ser cambiado. Además un recurso recientemente cambiado es muy probable que sea cambiado nuevamente .

Un análisis de la carga de trabajo en los DFS revela que son considerados regulares los patrones de acceso de un cliente y que una gran cantidad de archivos tienden a ser compartidos.

Uno puede pensar que los esquemas de cache utilizados (basadas en los diferentes estudios de carga) en los DFS podrían ser usados en el Web. Pero una de las diferencias fundamentales entre ellos son los patrones de acceso (debido al gran numero de usuarios), la localidad de la referencia y los problemas que produce su uso a gran escala en redes globales.

A lo largo de los trabajos relacionados con esta disertación podemos definir que la invariante que mas se ajusta en Internet , es la hallada por Blaze.

Si un archivo no fue cambiado enseguida después de su creación la probabilidad que este sea cambiado cae bruscamente.

[80] Diversos trabajos dan cuenta de este hecho. Uno de ellos : A trace-driven analysis of the 4.2BSD file system . Proceedings of the 10th ACM Symposium on Operating Systems Principles. Dec 1985.

5.2. Invariantes en el WWW

Desde 1994 cientos de trabajos han tratado de determinar esos invariantes en el WWW. Uno de los primeros [CBC95] Bestavros muestra que muchas características del uso de la WWW pueden modelarse usando una distribución power-law (ley de potencia)[81], incluyendo la distribución del tamaño del documento, la popularidad del documento como una función del tamaño, la distribución de requerimientos de usuarios para documentos, y el número de referencias a documentos como una función de su rango global en popularidad .

Sin embargo mayoría de ellos son han sido utilizados para mejorar los mecanismos involucrados en los esquemas de caching, el objetivo de la presentación de los mismos en esta disertación es ver como ODS se ajusta a estos invariantes. A partir de la utilización masiva de ODS deberíamos encontrar nuevos invariantes .

5.2.1. Tiempo de media vida de un documento

En [Bes95b] en base a los logs de servidores Web determina que existen tres clases de documentos: los *localmente* , *remotamente* y *globalmente populares*. Hallo que los remotos y globales tenían un 0.5 % de actualización al ida mientras que los locales 2% actualizaciones/ida. En todos los casos los updates están confinados a un muy pequeño subconjunto de documentos, denominados *mutables*. En cambio en [GS96][Gwe95] encuentran que las actualizaciones son del 1.8% en un periodo de 25 dias. En [Wor94][GS96] concluyen que el tiempo de vida media se encuentra en los 50 dias .

Con estos valores vemos que para la mayoría de los recursos que existen en Internet , ODS es un modelo valido.

5.3. Distribución de las preferencias

En particular sobre la popularidad de los archivos solicitados los primeros trabajos que sugieren un distribución Zipf para el acceso han sido desde el lado de un proxy [Gla94] un cliente [CBC95] y un servidor web [ABC+96] .

La ley Zipf [Zip49] es empírica y fue originalmente aplicada a la relación entre la popularidad de las palabras en términos de su ranking y su frecuencia de utilización en un

[81] La forma de la distribución power-law es una hipérbola. El tráfico de documentos actual, generado por requerimientos de usuarios también sigue este tipo de distribución.

texto dado. La frecuencia de una palabra es inversamente proporcional a su ranking r (la palabra mas utilizada tiene ranking uno y así sucesivamente) . La ley cumple con (1) , siendo N el numero de palabras distintas halladas en el texto.

$$(1)\ f(r) \cong C/r \quad \text{con} \quad C = \frac{1}{\left(\sum_{i=1}^{N} \frac{1}{i}\right)}$$

En el escenario planteado por un negocio que ofrece videos , también se asume una Zipf pero con algunos parámetros de ajuste [82] . Asumen que la probabilidad de que el *r-esimo* video mas popular sea solicitado, sobre N disponibles , por un usuario es :

$$P_r = \frac{G}{r^{\xi}} \quad , r = 1,...N$$

$$G = \frac{1}{1 + \sum_{r=2}^{N} 1/r^{\xi}}$$

Incrementando el paramento ξ se incrementa la popularidad relativa del video. En particular estos estudios sobre video están orientados a definir estrategias de desarrollo en sistemas de video bajo demanda, en los cuales los videos mas populares estarían disponibles en sistemas de almacenamiento de rápido acceso .

Pero debemos tener en cuenta que tanto en un texto como en un video club en principio el acceso a esos items tienen igual costo.

Con lo cual dado que el proxy accede a recursos que no tienen igual costo , no es natural su aplicación. No a si en un servidor web donde en principio el acceso a todos sus objetos tendría igual costo para un usuario.

[82] (http://www.eurocom.fr/~ross/Mmtutorial/) [Ross98]

Sin embargo varios autores sugieren que sigue fuertemente la ley , introduciendo un factor α que dependería de la correlación de intereses de la comunidad que comparte el cache y la cardinalidad de la misma :

$$f(r) \cong \frac{1}{r^{\alpha}} \text{ con } 0.55 < \alpha < 0.97$$

En [BCF+97] concluyen que el rango de α = [0.57; 0.67], ambiente con gran numero de usuarios alfa pequeño ,ambientes con pocos usuarios u homogeneos alfa grande. En cambio en [NHM+98] toman $\alpha = 0.75$.

Hemos tratado de encontrar evidencias de estas relacionados en el proxy de la UBA , pero las dispersiones son muy grandes con las funciones mencionadas, actualmente estamos procesando los logs de una comunidad de un ISP a los efectos de determinar si se ajustan a las distribuciones Zipf.

Sin embargo esta distribución no determina cuales son los mas globalmente populares y que ahorro en ancho de banda producirían si fuesen distribuidos a los efectos de mejorar la localidad de la referencia de los mismos.

Uno de nuestro objetivos en [BM00] fue la determinación de una cota que permita la identificar tales documentos no en un servidor web, sino en el conjunto de servidores web accedidos por una comunidad dada. Determinados dicho objetos ofrecerlos para su distribución a comunidades de intereses similares mediante ODS. Nos planteamos un ranking de los mas populares , de dicho conjunto cual el subconjunto que de ser distribuidos produce la mejor relación disminución de tráfico latencia tráfico de ODS.

5.3.1. Tráfico de los documentos populares

En [Bes95][Bes95a] Analiza los logs de un servidor Web de la Universidad de Boston: solo el 10 % de todos los bloques de datos registran el 91% de los pedidos. Corrobora las observaciones con los logs del servidor Web de la gira de los Rolling Stones, que a diferencia

de la Universidad Boston, sirvió exclusivamente a clientes remotos. En el periodo de la gira sirvió 1GB/dia (60.000 clientes request al menos 10 archivos) De los 400 MB accedidos al menos una vez, solo 21 MB (5.25 %) fue responsable del 85 % del tráfico.

En [BC94] se analizan las 837.046 solicitudes recibidas en el server web de NCSA en un periodo de dos dias. Los 25 documentos mas populares fueron responsables del 59% de los requests y del 45 % de los bytes enviados. El sitio tenia unos 8500 documentos de los cuales el 95% menores a 60Kb, siendo el tamaño promedio de 18Kb.

Realizamos análisis de los logs generados (7-95 al 10-96) por el servidor Web del Instituto Nacional de Tecnología Industrial [83] en el cual encontramos que el 5 % de lo objetos mas populares le corresponde el 75 % de las solicitudes.En reportan que [BBM+97] que el 1 % de los documentos mas populares son responsables del 10% del tráfico

Tenemos una regla, 90/10 , en un sitio popular del total de objetos el 10% es responsable del 90% del tráfico . A su vez existen sitios muy populares esta relación tiende a 80/20 [84] *y la gran mayoría de los sitios esto puede llegar a 98/2 .*

Podríamos concluir que el enorme espacio de recursos disponibles en Internet muy pocos son los que realmente son accedidos.

5.4. Escenario

Definimos el concepto de *escenarios* [CR92] [RCC97] que pueden ser usados, entre otras cosas, para evaluar la escalabilidad de sistemas de acceso de la información en Internet como WWW [Abd98] y ODS. Pondremos énfasis en sistemas académicos.

Un escenario es un ejemplo representativo o una instancia de la utilización de una RDT. El escenario puede focalizarse en los clientes, servidores , usuarios, enlaces o cualquier otro objeto . Para ser útil, un escenario debe tener bien definido los paramentos de salida que se necesita estudiar y los de entrada que pueden cambiar durante el análisis. Por ejemplo nosotros podemos querer estudiar el numero de clientes de una comunidad que pueden ser soportados por enlace de red siendo que esa comunidad tiene ciertos comportamientos en el

[83] http:www.inti.gov.ar
[84] Surgen de los logs de uno de los sitios mas visitados de Argentina , julio 1999.

acceso a la información[85] . El objetivo del escenario puede ser el de entender mejor al sistema, predecir futuros comportamientos, y realizar ajustes en el modelo entre otras cosas.

Se denomina escenario porque los valores de los parámetros utilizados son representativos de los valores encontrados en algunas mediciones para una comunidad determinada, pero no son necesariamente representativos de otras comunidades de usuarios .

Es por ello que los parámetros que tomamos son el la comunidad de la UBA y en principio serian aplicables a las otras comunidades de la RIU[86].

Si bien ODS trae aparejado un conjunto de mejoras en el acceso a los recursos, las que produce en el acceso ordenado y en principio calificado a los contenidos es de difícil cuantificación los beneficios. Sin embargo podemos estimar el tráfico en ODS en función de la distribución de determinados documentos populares y cual es la mejora que obtendrían otras comunidades.

El escenario que planteamos es el siguiente : tenemos a la comunidad de las UBA en la cual todos los dias realizamos un análisis de los logs del proxy para determinar los intereses de esa comunidad , si asumimos que las demás universidades tienen intereses similares los mismos cuando se realice las oferta de esos documentos serán aceptadas y distribuidas a las restantes 33 universidades nacionales que componen la RIU.

En este escenario no cuantificamos cuanto aumentara el acceso de los recursos al hacerlo local, y tener un costo cero a su acceso, y el hecho de realizar su oferta , de alguna manera si las demás comunidades académicas saben de la existencia de recursos ya accedidos por otra comunidad hay evidencias que las solicitudes aumentarían con respecto a la comunidad inicial.

De alguna manera genera un efecto de realimentación , si es muy requerido por algo será , ese fenómeno se verifica cuando en un video club listan los videos mas solicitados . La oferta de algo muy requerido genera mas requerimientos .

Ninguno de los estudios que comentamos a lo largo de esta disertación plantea cual es la cota para considerar a un recurso popular. Del análisis de varios conjuntos de datos tomados

[85] Por ejemplo se considera para el diseño del números de enlaces telefónicos de para acceso dial up a Internet una relación de 10 usuarios por líneas , el las comunidades del sur de la Argentina esta relación no es valida ya que los hábitos de la misma hace que el tiempo de conexión sea superior a las de otras regiones del país.

[86] De alguna manera esto es corroborado por la perfomance obtenida en nuestros esquemas de cache cooperativo

durante agosto de 1988 y agosto de 1999 en los logs del proxy de la UBA , sobre el conjunto total de recursos O solicitados obtuvimos los siguientes resultados : $O_{NP} \cong 75\%$ fueron los accedidos únicamente una vez y los que tuvieron mas de un acceso $O_P \cong 25\%$.

O_{NP} : *Cantidad de objetos un único acceso en el periodo de la muestra*

O_P : *Cantidad de objetos que tienen mas de un acceso en el periodo de la muestra*

O_{pi} : *Cantidad de objetos que tienen mas de i accesos en el periodo de la muestra*

S_{pi} : *Cantidad de solicitudes realizadas a objetos con mas de i accesos*

M_{pi} : *Cantidad de Mbytes que tienen los objetos con mas de i accesos*

$\eta_i = \frac{O_{pi}}{O_p}$ porcentaje de objetos con i>=2 a distribuir por ODS

$\chi_i = \frac{S_{pi}}{S_p}$: porcentaje de solicitudes a objetos que se le mejora la localidad de la referencia

$\kappa_i = \frac{M_{pi}}{M_p}$: porcentaje de tráfico que se genera en ODS por la distribución de objetos

Asumiendo las siguientes cotas C_i con $i[2,5,10,15,10,50]$, obtuvimos los siguientes resultados :

C_i	η	κ	χ
2	49.01%	53.14 %	78.51
5	14.75 %	9.19 %	57.94
10	6.36%	2.41 %	45.92
15	4.09%	1.37 %	40.32

20	2.99%	0.55 %	36.51
35	1.51 %	0.26 %	28.76
50	0.93%	0.19 %	24.16

En conclusión de los conjunto de documentos con cota mayor o igual a 2 , si distribuyo por ODS objetos con Cota 20 al 36 % de las solicitudes se le mejora la localidad de la referencia y con un costo mínimo , 3 % en tráfico generado al distribuir los recursos por ODS.

Capitulo 6

Replicación y Trabajos Relacionados

6.1. Introducción

Los protocolos de replicación son de interés en diversas áreas de las ciencias de la computación y tecnologías de la información , tales como sistemas operativos tolerantes a fallos, distributed y parallel databases, y distributed computing systems . Su objetivo es proveer confiabilidad y disponibilidad en los sistemas. En Internet la replicación se hace necesaria para supervivencia. Nuestro sistema no utiliza una replicación masiva, pero dentro de un grupo de interés (workspace) tenemos protocolos de replicación. Surge entonces como una cuestión importante que las actualizaciones a las replicas necesitan sincronizarse con el costo que esto trae aparejado . El mecanismo mas directo de sincronización es el que requiere que todas las replicas estén en el mismo estado todo el tiempo, llamado *one-copy equivalence*

6.1.1. Comunicación en grupos

Comunicación en grupos es una abstracción a nivel de sistema operativo que ofrece conveniencia y claridad a los programadores [LCN90]. En definitiva es una abstracción de comunicaciones de alto nivel que busca simplificar la tarea de desarrollar sistemas distribuidos .

Tanto en [Bir93] como [Bab97] hablan de grupos de procesos, en cambio en [Tan92] [Gol92] comunicación de grupos.

Nosotros adoptaremos como nombre el de comunicación en grupos. Este mecanismo organiza un conjunto de nodos en un grupo [Bir87] . El grupo actúa como una única entidad abstracta. El grupo soporta dos tipos de operaciónes : *group multicast (para enviar mensajes para los miembros del grupo) y membresía.*

El mecanismo de comunicación de grupos puede ser usado para construir un servicio replicado (algunos autores denominan *distributed process group*).

En [Tan92] al grupo se puede enviar un mensaje ya sea broadcast o multicast y cuando no se dispone de ellos se envian n mensajes a n procesos. En cambio Birman entiende como multicast enviar un único mensaje a todos los miembros del grupo.
En [Tan95] los grupos pueden ser dinámicos o estáticos. En cuanto a su estructura interna simétricos o jerárquicos . Nota que fisicamente no es necesario que cada nodo este conectado con los demás .En grupos jerárquicos existe un coordinador (decide quien realiza una tarea).

6.1.2. Consistencia

La replicación de recursos mejora la performance y disponibilidad acercando la información a los lectores. Mediante el almacenamiento de datos compartidos en procesadores donde son frecuentemente accedidos la necesidad de costosas lecturas remotas se decrementa. La copia de datos críticos en procesadores, con independencia del modelo de fallas, aumenta la probabilidad de que al menos una copia este accesible. Sin embargo cuando los recursos se replican debe tenerse en cuanta la consistencia de cada copia.

Sin embargo cuando se replican recursos hay que tener en cuenta la correctitud de cada copia. Se define como consistencia al hecho de que todas las copias del mismo ítem lógico de datos deben coincidir en exactamente un valor [DGS85].

Mantener la correctitud (consistencia) y la disponibilidad de los datos son objetivos que entran en conflicto ante particiones de red. [DGS85]. La correctitud se puede asegurar permitiendo las operaciónes sólo en una partición. Esto requiere de un mecanismo confiable que pueda detectar una partición. En una red muy congestionada como Internet, no es fácil distinguir una partición de la latencia de la red [87]. (TCP asume por ejemplo que si no llegan ACK se debe a congestión) .

[87] We assume that site failures are detectable : while a site is "down " , other sites can detect this fact. Some networks provide this feature internally (eg , the early ARPANET NCP protocol and SNA's virtual circuit protocol) . Others do not (eg. the currente ARPANET TCP/IP protocol). If the network does not provide this feature , the DBS (data base system) must implement it through high-level time-outs-which are not completely satisfactory , but are workable in the practice. acm december 1984 An algorithm for concurrency control and recovery Cito pp 600

Esta forma de mantener la correctitud afecta a la disponibilidad del sistema. Si en cambio queremos asegurar la disponibilidad de los recursos en todo momento, podemos permitir la operación normal aunque existan particiones. Esto provoca que las replicas no sean consistentes y obliga a aplicar algún mecanismo de corrección de la información una vez superada la partición [DGS85].

Se plantea entonces un compromiso entre disponibilidad y consistencia. Si buscamos asegurar disponibilidad de los recursos todo el tiempo, debemos permitir la operación normal del sistema en presencia de particiones. Frente a cada aplicación se desarrollan estrategias optimistas o pesimistas, las cuales tienen en cuenta el compromiso entre consistencia y disponibilidad. Las pesimistas evitan inconsistencias por limitación de la disponibilidad. En cambio la optimista no limita la disponibilidad, y pueden existir inconsistencias globales. Asume que, si bien existen, raramente ocurren. Cuando una partición de red es superada, el sistema primero debe detectar posibles inconsistencias y resolverlas (desde el punto de vista sintáctico y semántico).

En particular a partir de una estrategia optimista se desarrollan los sistemas con mecanismos de replicación de consistencia de débil, que permiten que ante una partición de red sus réplicas diverjan y continúen brindando servicio. Una vez superada la partición, las réplicas en algún momento , *eventualmente* ,convergen a un estado de consistencia. Nosotros incluimos dentro del concepto de partición, las significativas latencias que se producen en Internet actualmente.

En función de la naturaleza del problema y sus condiciones de contorno se pueden aplicar *estrategias optimistas* o *pesimistas*. La definición del tipo de estrategia resulta de vital importancia para el diseño de un sistema distribuido en redes globales.
Las condiciones de contorno que podemos adoptar es en función de la conectividad a nivel de red. Esta puede ser *fuerte* (alta velocidad, tasa de errores y fallos bajo) o *débil* (baja velocidad, tasa de errores y fallos alto.

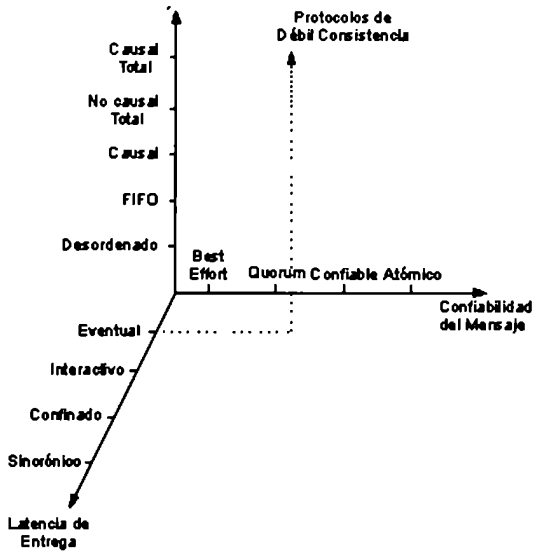
6.2. Marco de Trabajo: Protocolos de Replicación con Mecanismos de Consistencia Débil

El grado de consistencia de un sistema distribuido, depende en última instancia de las restricciones de la aplicación, no es lo mismo informar sobre las cotizaciones de la bolsa a brokers, que sobre nuevos artículos científicos a un grupo de investigadores.

El objetivo de ODS es tratar con objetos cuya consistencia no es critica, la mayoría de los objetos que generan el tráfico HTTP, cuyo grado de sincronización requerido es del orden de las horas .

Los sistemas con mecanismos de replicación de consistencia de débil, permiten que ante una partición de red sus réplicas diverjan y continúen brindando servicio. Una vez superada la partición las réplicas en algún momento , *eventualmente* ,convergen a un estado de consistencia. Nosotros incluimos dentro del concepto de partición, las significativas latencias que se producen en Internet actualmente.

En un sistema distribuido los nodos de un grupo intercambian mensajes para mantener las replicas consistentes. Se considera que el envío de mensajes tiene tres componentes (Fig1) [Gol92]:



Latencia: tiempo que deben esperar las réplicas para recibir un mensaje. Esto depende de cuando empieza y termina el proceso de replicación.

Sincrónico: comienza en forma inmediata y se completa en un tiempo acotado.
Acotado: los mensajes pueden ser encolados o retrasados, pero el envío se completa en un tiempo acotado.
Interactivo: toma un tiempo finito, pero no acotado. El envío es inmediato.
Eventual: toma un tiempo finito, pero no acotado. El envío puede ser demorado y los mensajes encolados.

Ordenamiento: representa el orden en que los mensajes son enviados a la aplicación.

Causal total: los mensajes son enviados en el mismo orden a todas las réplicas y se mantienen las relaciones causales entre los mensajes.
No causal total: garantiza que los mensajes son enviados en el mismo orden a cada réplica, pero no garantiza que siempre se mantenga la causalidad.
Causal: los mensajes son enviados en orden tal que preservan la potencial relación causal.
FIFO: sólo garantiza que los mensajes de una misma fuente son enviados en orden.
Sin orden: los mensajes son enviados sin tener en consideración el ordenamiento.

Grado de confiabilidad: cuantas réplicas podrán recibir copias de un mensaje. Los grados de confiabilidad son los siguientes:
Atómico: garantiza que el mensaje es enviado a todas las réplicas o a ninguna. Si una réplica falla el envío es abortado.

Fiable: asegura que un mensaje es enviado a todas las réplicas que funcionan, pero los miembros que fallan no necesitan recibir el mensaje. Si el que envía falla, el envío no es garantizado.
Quórum: garantiza que por lo menos a una parte de las réplicas les llegue el mensaje. Si el que envía el mensaje falla, el envío no es garantizado.
Best effort: intenta enviar a cada réplica, pero no garantiza que llegue el mensaje. Este mecanismo no necesita mantener una copia del mensaje.

Los protocolos de consistencia débil son aquellos cuyo grado de confiabilidad es fiable y cuya latencia es eventual.

Consideramos que se requieren en principio, protocolos de consistencia débil para implementar mecanismos de replicación escalables en Internet.

Los sistemas con mecanismos de replicación de *consistencia fuerte* como Isis [Birman] Psync [Mis89] Arjuna[REF] y Lazy replication [REF] vemos que son costosos y no escalan correctamente en redes de conectividad débil.

Para que un mecanismo de replicación resulte escalable en redes de comunicación poco confiables y de gran tamaño, es necesario utilizar en principio protocolos de consistencia débil. Estos ya han sido utilizados en una gran variedad de sistemas debido a alta disponibilidad del sistema, buena escalabilidad y simplicidad de diseño (e.g. Grapevine [Bir82], Clearinghouse [Opp83], Locus [Pop85], Coda [Sat92], GNS [Lam86], AFS [Satya 93], News, Refdbms [Gol92a], mirror-d, flood-d Harvest [Obr95], OSCAR [Dow90], Hyper_G[Kap95] etc).

Un eficiente algoritmo de replicación inunda los mensajes entre las replicas. Notar que los esquema de inundación a nivel de red de los algoritmos de ruteo (simplemente siguen la topología de la red física e inundan las actualizaciones a todos los enlaces físicos de la red) difiere ya que las replicas inunda datos a sus vecinos lógicos o replicas pares . El esquema propuesto de inundación a nivel de aplicación en ODS hace un uso eficiente del ancho de banda.

Debido a que los protocolos de nivel 3 ocultan la topología de a la red a los de nivel 7, las replicas por si solas no pueden seleccionar cuales son sus pares de inundación para optimizar el uso de la red.

Grapevine y su sucesor Clearinghouse ignoran la red y la topología de actualización . El GNS asume la existencia de un único administrador el cual configura manualmente la topología de update. Coloca las replicas en un ciclo Hamiltoniano y reconfigura el anillo cada vez que se agrega o saca una replica.

News emplea inundación para distribuir las actualizaciones entre miles de replicas. Como GNS se configura manualmente la topología de actualización.

6.3. Trabajos Relacionados

6.3.1. El Proyecto I2-DSI (Internet2 Distributed Storage Infrastructure)

Si bien la red subyacente en Internet 2 no presenta conectividad débil , para aplicaciones descriptas a lo largo de este trabajo si la tiene ya que por ej. el tráfico FTP o HTTP seguirá manejándose con política de best effort , reservando para el tráfico de otras aplicaciones que requieran calidad de servicio, QoS [88].
Es por ello que uno de los proyectos es una Infraestructura de Almacenamiento Distribuida (DSI)[89] la cual propone una Arquitectura para Canales de Contenidos en Internet .
Hemos visto que las herramientas replicar de archivos sitio a sitio son ampliamente usadas entre sitios Internet para replicar HTTP o FTP (sitios mirror) . Generalmente son front-end scripts para FTP (mirror , ftp-mirror) . Otras herramienta que permite la sincronización de archivos es *Rsync* y *rsync+*[90]
En realidad es un replicación masiva de grandes repositorios de datos , librerías de software , etc.

[88] QOS networking se refiere a la tecnología de nivel red que permite garantía de servicio . Dicha garantía puede ser estadística , RFC 2475, Diff Services , comunidad Internet o Garantía determinística (e.g. ATM tiene soporte intrínseco QoS).

[89] http://dsi.internet2.edu/

[90] El modelo es una modificación de rsync , que sigue el modelo de actualización asincrónico de NetLib en combinación con el fine-grained uso de checksums de rsync

6.3.2. Productos Comerciales

Durante 1999 surgieron dos redes comerciales Akamai e Island. Las mismas se podrían denominar de acuerdo a las discusiones del WREC , redes de distribución de contenidos , CDN (Content Distribution Networks)[91] . En realidad un CDN es un red global de caches, con dos diferencias fundamentales : los caches reciben solicitudes de otros caches antes que de los clientes y es coordinado por un mecanismo que rutea las solicitudes de los clientes al mejor cache. Deben realizarse modificaciones en las paginas de los servidores Web. Al momento todo el mecanismo no han sidos publicados y se mantienen casi como secreto industrial.. Si bien se denominan redes de distribución de contenidos , las mismas en realidad son esquemas globales de caches .

[91] http://www.digisle.net/
http://www.mirror-image.com/
http://www.adero.com/
http://www.akamai.com/

Capitulo 7

Conclusiones y Trabajos Futuros

Los principales objetivos principales detrás del trabajo descriptos en esta disertación que se han cumplido son :

- Identificar y comprender los fallos de escalabilidad en los sistemas de área global
- Proponer además un modelo de referencia cuya arquitectura permita superar dichos fallos
- Un comprensiva caracterización de las fallas de escalabilidad
- Una taxonomía de los mecanimos de replicación y caching en Internet.
- Limitaciones y potenciales de los mecanismos de caching y mirroring..
- Determinación del tráfico de Web proxy que permitan determinar cotas de popularidad en documentos web a los efectos de luego distribuirlos mediante ODS.
- Un modelo simple de ODS con escenarios que muestra como podemos escalar .

El corazón del trabajo presentado en esta disertación fue como hacer frente a la escalabilidad y conectividad débil mediante un sistema que mejora el acceso a la información de una comunidad de gran escala, organizando y distribuyendo dicha información entre comunidades de intereses similares.

A tal efecto hemos presentado un sistema asincrónico de replicación de consistencia débil, Sistema de Distribución de Objetos (ODS). Que utiliza los recursos disponibles en Internet y se adapta a la red subyacente. Pueden haber múltiples sistemas basados en ODS, trabajando

en Internet al mismo tiempo. Cada uno con sus propias clases de objetos, mecanismos de seguridad, autoridades de clasificación, etc. Estos sistemas de distribución se pueden configurar fácilmente para eventos especiales (workshops, cursos virtuales, etc.).

Los objetos son actualizados automáticamente en forma off-line. Los usuarios se suscriben para recibir ciertos objetos y el sistema les asegura que siempre tendrán acceso a la última versión disponible de dichos objetos. El modelo se ajusta correctamente a objetos persistentes que no cambian constantemente y cuando cambian no es necesario informar inmediatamente a los usuarios.

Para satisfacer los requerimientos de los usuarios en forma más óptima y permitir escalabilidad, los agentes de servicio se agrupan *en grupos*. Nosotros utilizamos replicación selectiva, construyendo cadenas de distribución dinámica para cada grupo. Las cadenas de distribución se adaptan al estado de la red subyacente y siempre respetan los rótulos de los agentes de servicio, que a su vez representan los intereses de los usuarios finales

Las dos redes virtuales que forman ODS: Red de Distribución de Objetos (ODN) y Red de Ruteo de Objetos (ORN), fueron diseñadas para trabajar en forma independiente. Esto nos permitirá continuar el desarrollo y mejoramiento de cada red en forma separada. Además deseamos generalizar el concepto de los mecanismos de ruteo utilizados en ORN a otros sistemas y definir los servicios de ORN en forma más general.

Actualmente contamos con un prototipo que tiene un mecanismo incorporado para armar las cadenas de distribución. Las cadenas de distribución se arman partiendo de un grafo que no es completamente conexo. Decimos por lo tanto que cada agente de servicios tiene vecinos (agentes de servicio con los que se puede comunicar). Los agentes de servicio adoptan entonces una conducta aún más cooperativa, pues no sólo consumen los documentos de interés para sus usuarios, sino también los documentos de interés para sus vecinos. De esta forma cuando el grafo es conexo, cada agente de servicio podrá recibir los documentos de su interés. Sin embargo como todos sus vecinos harán propios sus intereses, ellos también recibirán los documentos y a su vez transmitirán el interés a sus vecinos. En el peor caso todos los agentes de servicio poseerán todos los documentos, perdiendo las ventajas de la distribución selectiva, tendiendo esto a ser una *replicación masiva*.

A partir del modelo de referencia concluimos que era necesaria la construcción de cadenas de distribución que respetasen los rótulos de cada agente de servicio y que por lo tanto era necesario que todos los agentes de servicio se pudiesen comunicar entre si. Podemos decir que en ORN todos los agentes de servicio son vecinos, aunque un SA no se comunique con

todos sus vecinos. De hecho en ORN dos SA se comunican sólo cuando mantienen una relación cliente-proveedor dentro de una cadena de distribución. Esto implica que la comunicación entre SA queda restringida a SA del mismo grupo (SA con un interés común). Comparamos el mecanismo utilizado en el modelo de referencia a utilizar un broadcast, en donde los paquetes llegan a todos los destinos posibles. Un broadcast utiliza siempre el mejor camino, pues utiliza todos los caminos haciendo un muy mal uso de los recursos de red. En este caso no sólo se da un mal uso a los recursos de comunicación, sino también a los recursos locales de cada SA, pues cada uno debe almacenar todos los documentos que recibe.

Partiendo de este modelo de referencia, hemos desarrollado una variante en la que la relación de vecinos no es bidirecciónal. Conociendo la topología total de la DDN y los intereses de cada agente de servicio, se pueden construir cadenas de distribución eficientes. La performance de esta variante del modelo de referencia es muy buena, ya que presenta una gran robustez ante particiones de red (teniendo en cuenta también las grandes latencias de Internet.

El prototipo ha sido desarrollado en Java , JDK 1.1 , a los agentes de servicio se interactúa mediante un agente usuario desarrollado para el caso de DDN o mediante un cliente web.

En particular hemos visto que el tráfico Web tiene una componente determinística y cíclica muy importante para los ISP. Este hecho puede ayudar la ecualización del tráfico .

Los sistemas locales de caching en un ISP además tienen una ventaja económica ya que hemos visto que ahorros del ancho de banda de un 30% se logran en el escenario de un ISP , del cual solamente el 10% correspondería a tráfico nacional.

El trabajo de en esta disertación muestra que la escalabilidad de los RDT en Internet es posible y en particular del WWW . En definitiva a ODS se puede acceder mediante Web.

No hay un único mecanismo que permita la escalabilidad en Internet , se deben utilizar en forma conjunta mecanismos de caching para el tráfico cotidiano y redes de distribución para los contenidos de calidad.

Otro aspecto es que si bien el ancho disponible en la comunidad académica prácticamente no ha crecido en los últimos cuatro años con lo cual el escenario de conectividad débil en el que validamos ODS fue el de la peor condición, estamos lejos aun de contar con velocidades como las de Internet 2 , así mismo el incremento futuro que se prevé en el 2000 como resultado de la desregulación de las Telecomunicaciones en Argentina , las necesidades aun

insatisfechas de ancho de banda de la comunidad académica nos dejara siempre con conectividad débil entre las 9 AM y 9PM .

Durante 1999 comenzamos a realizar un extenso estudio del tráfico en uno de los ISP mas grandes de argentinas , 140.000 usuarios , esperamos durante el 2000 presentar un reporte de las conclusiones. Pero siguen valiendo las motivaciones que dieron lugar a este trabajo entre ellas conectividad débil presente , aun con ancho de bandas a Internet de mas de 40 Mbps, particiones de red y los problemas de escalabilidad propios de la naturaleza del Web y del caos de contenidos .

Comercialmente el caos de información se busca organizarlo mediante los llamados portales horizontales que permiten acceder a portales temáticas dentro del mismo host u otro , en definitiva tenemos siempre el mismo esquema centralizado. Los motivos son básicamente la facilidad de implementación y la idea que la propaganda con la cual dichos portales ofrecen es manejable desde un único punto de acceso. Creemos que el mismo motivo a llevado a la proliferación de buscadores de contenidos centralizados y no distribuidos.

Estos portales son en definitiva quienes tienen una cotización en el índice Nasdaq. Pero a lo que igual que Microsoft o Cisco tienen una red de distribución a nivel mundial de sus productos, los portales deberían contar con sus redes de distribución de contenidos para seguir cotizando en la bolsa tecnológica.

Utilizamos este modelo para distribuir grandes volúmenes de información entre SA ubicados en España y Argentina. Dicha información resultaba imposible de ser transferida utilizando por ejemplo FTP, HTTP o replicadores Web, dada la alta latencia y alta proporción de paquetes descartados, producto de la congestión de las redes académicas de ambos países. Los documentos que transferíamos no eran tan grandes como se puede esperar para que provoquen este tipo de problemas, apenas excedían en tamaño a 1 MB, sin embargo la conexión FTP o HHTP se cerraba antes de poder transmitir el archivo en forma completa.

2. Trabajos Futuros

- Comparar el protocolo propuesto para ORN con los protocolos de ruteo multicast de Internet y mejorar en el sistema final la perfomance de la red de ruteo, mediante la determinación de heurísticas basadas en el conocimiento topológico de la red subyacente (por ejemplo conectividad fuerte en determinados sistemas autónomos) .

- Implementar localización del *miembro más cercano*, para posibles usuarios que no pertenezcan a ODS. La localización del grupo, o de algún miembro de un grupo que ya este funcionando es similar al problema de localización de recursos [Guy95].

- Implementación de copyrights, firmas digitales, certificación, pago y registros de distribución (e.g. número de usuarios que acceden a un objeto replicado, cantidad de replicas de un objeto, etc.).

- Pasar de la etapa de prototipo a la de sistema de producción en la nueva Red Académica .

- Extender a el modelo de ODS a objetos que requerían actualizaciones desde varios agentes con un sincronización , basados en el paradigma y modelo de programación de vistas sincrónicas extendidas.

Bibliografia

[Abd98] Abdulla G. Analysis and Modeling of World Wide Web Traffic. PhD Thesis. Virginia Polytechnic Institute. May1998.

[ABC+96] Virgilio Almeida; Azer Bestavros; Mark Crovella; Adriana de Oliveira: Characterizing Reference Locality in the WWW . In *Proceedings of PDI'96: The IEEE Conference on Parallel and Distributed Information Systems,* Miami Beach, Florida, December 1996.

[AH87] Adams, Rick; Horton, Mark Standard for Intrechange of USENET Messages RFC 1036. Marina del Rey , CA: Information Sciencies Institute. 1987.

[AL92] P. Albitz and C. Liu, "DNS and BIND in a Nutshell," *O'Reilly and Associates*, 1992.

[Ank93] Anklesaria, Johnson et.al .The Internet Gopher Protocol Request for Comments 1436, University of Minnesota ,March 1993.

[ASA+95] Marc Abrams; Charles R. Standridge; Ghaleb Abdulla; Stephen Williams; Edward A. Fox Caching Proxies: Limitations and Potentials. In *Proceedings of the Fourth International WWW Conference*, Boston , MA, December 1995.

[AWY+96] Aggarwal, C.C.; Wolf, J.L.; Yu, P.S.; Epelman, M.: On caching policies for Web objects, *IBM Research Report,* Yorktown Heights, NY, November 1996.

[AY96] Aggarwal, C.C.; Yu, P.S.: On disk caching of Web objects in proxy servers, *IBM Research Report,* Yorktown Heights, NY, November 1996.

[BA91] Blaze, Matthew; Alonso, Rafael: Long-term caching strategies for very large distributed file systems. Proceedings of the USENIX Summer Conference, pp 3-16 Junio 1991.

[BA92] Blaze, Matthew; Alonso, Rafael: Dynamical hierarchical caching for large-scale distributed file systems. *Proceedings of the Twellvth International Conference on Distributed Computing Systems*, June1992

[BFC93] Ballardie, A.J.; Francis, P.F.; Crowcroft, J. Core Based Trees (CBT) . *Proc. of the ACM SIGCOMM '93.* San Francisco, CA. August 93.

[BBM+97] Baentsch, Michael; Baum, Lothan; Molter, George; Rothkugel, Steffen; Sturm, Peter: Enhancing the Web's Infrastructure: From Caching to Replication. IEEE Internet Feb, 1997

[BBM+97] Baentsch, M.; Baum, L.; Molter, G.; Rothkugel, S.; Sturm, P.: World-Wide Web Caching - *The Application level view of the Internet IEEE Communications Magazine*, Vol. 35, No. 6, June 1997 Postscript (1MB) (accepted draft - revised version only in journal).

[BC95] Bestavros, Azer; Crovella, Mark: Explaining World Wide Web Traffic Self-Similarity.Technical Report TR-95-015, *Computer Science Department,* Boston University, MA - October 1995.

[BC94] Braun, H.; Claffy, K.: Web traffic characterization: an assessment of the impact of cacheing documents from the NCSA's web server.

[BC96] Bestavros, Azer; Crovella, Mark: Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes. In *Proceedings of ACM SIGMETRICS'96*, Philadelphia, PA - May 1996.

[BCC+95] Bestavros, Azer; Carter, Robert L.; Crovella, Mark E.; Cunha, Carlos R. Heddaya, Abdelsalam ; Mirdad, Sulaiman A.: Application-Level Document Caching in the Internet. In *IEEE SDNE'95 : Proccedings of the Second International Workshop on Services in Distributed and Netsworked Environmentes*, Whistler , Canada , June 1995.

[BCR+99] Baleani, Mariana; Cervini, Carolina; Rodriguez, Liliana: Localizadores universales de recursos en Internet. Tesis de licenciatura. Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires. Diciembre 1999

[BDH94] C.M. Bowman, P.B. Danzig, D.R. Hardy, U. Manber, and M.F. Schwartz, The Harvest Information Discovery and Access System, in: *2nd International World Wide Web Conference,* Chicago, IL, 1994, pp. 763–771.

[BDH+94] Bowman, C. Mic; Danzig, Peter B.; Hardy, Darren R.; Manber, Udi; Schwartz, Michael F.Harvest: A Scalable, *Customizable Discovery and Access System Technical Report CU-CS-*732-94, Computer Science Department, University of Colorado, CO- August 1994.

[BDM+94] Bowman, M.; Danzing, Peter; Mander, Udi, Schwartz, Michael F. "Scalable Internet Resource Discovery : Research Problems and Approaches". *Comm. of the ACM. Vol 37 - num 8. August 1994.* Pp 98-107.

[BDM+95] Bowman, M.; Danzing, Peter; Mander, Udi; Schwartz, Michael F. The Harvest Information Discovery and Acces System. *Computer Networks and ISDN Systems.* Vol 28. December 1995. Pp 119-125.

[BDM+94] Bowman, M.; Danzing. Peter; Mander Udi, Schwartz Michael F. Scalable Internet Resource Discovery : Research Problems and Approaches . Comm. of the ACM. Vol 37 - num 8. August 1994. Pp 98-107.

[Ber92] Berners-Lee, T.J.; Cailliau, R.; Groff, J.F.; Pollermann, B. World Wide Web: The Information Universe Electronic Networking: Research, Applications and Policy. vol 2 - nro 1. pp 52-58. 1992.

[Ber94] Berners-Lee, T.; Fielding, R.T.; Nielsen, H. Hypertext Transfer Protocol - HTTP/1.0 RFC 1945. Mayo 1996.

[Ber95] Berners-Lee, Tim. Propagation, Replication and Caching . Web Consortium, MIT. December 1995.

[Bes95] Azer Bestavros , Robert Carter , Mark Crovella. Application-Level Document Caching in the Internet . In Proccedings of the Second Intl Workshop on Services in Distributed and Netsworked Environmentes (SDNE'95) , Whistler , Canada , June 1995.

[Bes95] Bestavros, Azer. "Demand-based dissemination for Distribute Multimedia Aplication". *Proceedings pf the ACM/ISMM/IASTED International Conference on Distributed Multimedia Systems and Aplications.* Stanford, CA. August 1995 .

[Bes95] Bestavros, Azer. Demand-Based Document Dissemination for the World-Wide Web Computer Science Department, Boston University, MA - February 1995.

[Bes95] Bestavros, Azer.Using Speculation to Reduce Server Load and Service Time on the WWW. In *Proceedings of CIKM'95: The Fourth ACM International Conference on Information and Knowledge Management*, Baltimore, MD - November 1995.

[Bes95] Bestavros, Azer. Demand-Based Dissemination to Reduce Traffic and Balance Load in Distributed Information Systems. In *Proceedings of the 1995 Seventh IEEE Symposium on Parallel and Distributed Processing*, San Antonio, Texas, October 1995.

[Bes96] Bestavros, Azer. Speculative Data Dissemination and Service to Reduce Server Load, Network Traffic and Service Time for Distributed Information Systems. *In Proceedings of ICDE'96: The 1996 International Conference on Data Engineering*, New Orleans, Louisiana. March, 1996.

[Bes96a] Bestavros, Azer . Middleware Support for Data Mining and Knowledge Discovery in Large-scale Distributed Information Systems* . I*n Proceddings of ACM SIGMOD 96 Data Mining Workshop*, Montreal, Canada, June, 1996

[BFN96] Berners-Lee, T.; Fielding, R.; Frystyk, H.: Hypertext Transfer Protocol – HTTP/1.0 RFC 1945, MIT/LCS, UC Irvine, MIT/LCS - May 1996.

[BH96] Bolot, Jean-Chrysostome; Hoschka, Philip: Performance Engineering of the World Wide Web: *Application to Dimensioning and Cache Design Fifth International World Wide Web Conference*. May 6-10, 1996, Paris, France.

[Bir93] Birman Kenneth P. The process Group Approach to realiable Distributed Computing. CS Cornell University. January 1993.

[Bla93] Blaze, Matthew. Caching in Large Scale Distributed File Systems . PhD Thesis, Technical Report 397-92. Princeton University, Dept. of Computer Science. January 1993.

[BLN+82] Birell, Andrew; Levin, Roy; Needham, Roger; Schoroeder, Michael. "Grapevine: An Exercise in Distributed Computing". *Communication of ACM. Vol 25-num 4. April 1982.* Pp 260-274.

[Bla93] Blaze, Matthew. "Caching in Large Scale Distributed File Systems". PhD Thesis, Technical Report 397-92. Princeton University, Dept. of Computer Science. January 1993.

[BMM94] Berners-Lee, T., Masinter, L., McCahill . Uniform Resource Locators (URL)", RFC 1738, CERN, Xerox Corporation, University of Minnesota, Diciembre 1994.

[BMS96] Baentsch, Michael; Molter, George; Sturn Peter: Introducing Application-level Replication and Naming into today's Web. *Fifth International World Wide Web Conference*

[BMT97] " Definición de un Marco de Seguridad para un Modelo de Distribución de Documentos". Tejedor C., Bolivio S., Meiter M. Tesis de licenciatura. Facultad de Ciencias Exactas y Naturales. Universidad de Buenos Aires. Diciembre 1997

[Cat92] Cate Vicent, Alex: A global file system. *USENIX File Systems Workshop Proceeding*, p 1-12, Ann Arbor , MI , May 21-22 1992.

[Cau93] Caughey S.J. et al, "SHADOWS: A Flexible Support System for Objects in a Distributed System," *Proceedings of the 3rd IEEE International Workshop on Object Orientation in Operating Systems (IWOOOS),* Ashville, North Carolina, USA, December 1993.

[CB96] Crovella, Mark E.; Bestavros, Azer: Explaining World Wide Web Traffic Self-Similarity *Technical Report,* Boston University, CS Dept, Boston. Aug., 1996 .

[CB96] Crovella, Mark E.; Bestavros, Azer: Self-Similarity in World Wide Web Traffic Evidence and Possible Causes* To Appear in the *ACM SIGMETRICS International Conference on Measuring and Modeling of Computer Systems*, May 1996.

[CDH+95] Carr, L.; De Roure, D.; Hall, W.; Hill, G.:"The Distributed Link Service: A Tool for Publishers, Authors and Readers," *World Wide Web Journal 1*(1), 647--656, O'Reilly & Associates Inc., 1995.

[CDN+95] [14] Chankhunthod, Anawat; Danzing, Peter; Neerdaels, Chuck; Schwartz, Michael F.; Worrel, Kurt. A Hierarchical Internet Object Cache . *CSD UCB.* November 1995.

[CHD+94] Carr, L.; Hall, W.; Davis, H.; De Roure, D.; Hollom, R.: The Microcosm Link Service and its Application to the World Wide Web 1994, *Presented at the 1994 WWW Conference*, Geneva .

[Che94] Chervenak A. Tertiary Storge : An evaluation of the New Applications. PhD Thesis , CSD , Univ, of California at Berkeley 1994.

[Cor91] Corbato, Fernando. On Building Systems That Will Fail. Communications of the ACM Vol 34 9 September 1991.

[Cis96] "Cisco LocalDirector," Cisco Systems, Inc. White paper, 1996.

[CL97] Cao, P.; Liu, C.; Maintaining strong cache consistency in the World-Wide Web, *In Proc. of the 17th IEEE International Conference on Distributed Computing Systems*, May 1997, y IEEE Transactions on Computers Vol 47 n 4 pp 445 –457 April 1998.

[CPK95] Chervenak, Ann L.; Patterson, David A.; Katz, Randy H.: Choosing the best storage system for video service . In: *Proceedings of the third international conference on Multimedia '95*, pages 109-119.

[CR92] Carroll, John M.; Rosson, Mary Beth: Getting around the task-artifact cycle: how to make claims and design by scenario. *ACM Transactions on Information Systems Volume 10 , Issue 2 Pages* 181-212 (1992)

[CY97] Chinen, K., Yamaguchi, S. : An Intearactive Prefetching Proxy Server For Improvement of WWW Latency . *Procceding of INET 97.*

[Dam97] Damani O. M. et al., "ONE-IP: Techniques for Hosting a Service on a Cluster of Machines," *Proceedings of the 6th International World Wide Web Conference*, Santa Clara, California, USA, April 1997.

[DC90] Deering, S; Cheritond Multicast Routing in datagram Internetworks and extended LAN's ACM Transactions on Computer Systems V8, N 2, May 1990. pp85-110.

[DDO93] Peter Danzing , Dante Delucia ,and Katia Obraczka . Massively replicating services in wide area internetworks . Technical Report 93-541 , University of California at Santa Cruz , 1993.

[DEF+96] Deering, S.; Estrin, D.; Farinacci, D.; Jacobson, V., Liu, C., Wei, L. "Protocol Independent Multicast (PIM): Motivation and Architecture". <draft-ietf-idmr-pim-arch-04.ps>. September 1996.

[DE94] Deutsch, P.; Emtage, A. "Publishing Information on the Internet with Anonymous FTP". <draft-ietf-iiir-publishing-01.txt>. May 1994.

[DFK+97] Douglis, F.; Feldmann, A.; Krishnamurthy, B.; Mogul, J.: Rate of change and other metrics: live study of the World Wide Web, in: *Symposium on Internet Technology and Systems, USENIX Association*, December 1997,

[DGM+85] Davidson, S.; García-Molina, H.; Skeen, D. "Consistency in Partitioned Networks". *ACM Computing Surveys 1985. Vol 17 - num 3. September 85.* Pp 341-370.

[DGP90] Downing, A.R.; Greenberg, I.B.; Peha, J.M. "OSCAR: A systems for weak-consistency replication". *Proceedings Workshop on the Management of Replicated Data. Houston Texas. November 1990.* Pp 26-30.

[DGS85] Davidson, S.; García-Molina, H.; Skeen, D. Consistency in Partitioned Networks . ACM Computing Surveys 1985. Vol 17 - num 3. September 85. Pp 341-370.

[DHS93] Danzig, Peter B.; Hall, Richard S.; Schwartz, Michael F.: A Case for Caching File Objects Inside Internetworks *Technical Report CU-CS-642-93,* Department of Computer Science, University of Colorado, CO - March 1993.

[DMW+94] Dahlin, M.; Mather, C.; Wang, R.; Anderson, T; Patterson, D.: "A Quantitative Analysis of Cache Policies for Scalable File Systems," *Proceedings of the 1994 Sigmetrics Conference,* May 1994, 150-160.

[DMW+94] Dahlin, M., Mather, C., Wang, R., Anderson, T, Patterson, D., " Coperative Caching: using Remote Client Memory to improve File Systems Perfomance". *OSDI.* 1994.

[Don95] Donnelley, James. "WWW media distribution via hopwise realiable multicast". *Computer Networks and ISDN Systems. Vol 27-num 6. April 1995.* Pp 781-788.

[DP96] Dingle, Adam; Partl, Tomas:Web Cache Coherence. *In Proceedings of Fifth International World Wide Web Conference.* May 6-10, 1996, Paris, France.

[ED92] A. Emtage and P. Deutsch Archie : An electronic directory service for the Internet. *In Proceedings of the Winter 1992 Usenix Conference* , pp 93-110, January 1992

[EF94] Egevang, K.; Francis P.: The IP Network Address Translator (NAT). RFC 1631, Network Information Center, *SRI International*, May 1994.

[EP92] A. Emtage and P. Deutsch . Archie : An electronic directory service for the Internet Proceedings of the Winter 1992 Usenix Conference , pp 93-110, January 1992

[Eri94] Eriksson, Hans. "MBONE: The Multicast Backbone". *Comm. f the ACM. Vol 37 - num 8. August 1994.*

[FCL92] Franklin, M.J.; Carey, M.; Livny, M.:Global memory management in client–server DBMS architectures, in: *18th VLDB Conference*, 1992.

[FGM+97] Fielding, R.; Gettys, J.; Mogul, J. ; Frystycs, H.; Berners-Lee, T.: Hipertext Transfer Protocol – HTTP/1.1 RFC 2068, UC Irvine, DEC, MIT/LCS – January 1997

[FJ94] Foster, M.; Jump, R. :Solicitation 94-75 , *STIS Database, National Science Foundation* , Arlignton , Va. , 1994.

[Fos94] M. Foster, R. Jump Solicitation 94-75 , STIS Database, National Science Foundation , Arlignton , Va. , 1994.

[Gla94] Glassman Steven. A Caching Relay for the World Wide Web. *First International Conference on the World Wide Web,* Geneva. Elsevier Science BV pp. 69–76. May 1994.

[Gol92] Golding, Richard. "Weak Consistency group communications and memberships". Ph.D. Thesis. *University of California, Santa Cruz. December 1992.*

[Gol92a] Golding, Richard. "End-to-end performance prediction for the internet". Technical Report UCSC-CRL-92-26. *University of California, Santa Cruz. June 1992.*

[GS94] Gwertzman, James; Seltzer, Margo. The case for geographical push-caching. *HotOZ Conference.* 1994.

[GS95] Guyton, James; Schwartz, Michael. Locating Nearby Copies of Replicated Internet Servers .Technical report CU-CS-762-95. University of Colorado at Boulder , Boulder , Colorado 80309-430. February 1995

[GS96] Gwertzman, J.; Seltzer, M.: World-Wide Web cache consistency. In: *Proceedings of the USENIX Technical Conference*, USENIX Association, January 1996, pp. 141–152,

[Gwe95] Gwertzman James. Autonoumus Replication in Wide-Area Internetworks . *Thesis Computer Science Harvard College* Cambridge , Massachusetts , April 1995.

[HEN97] The Higher Education National Software Archives. The UK Academic National Web Cache at HENSA Unix January 1997.

[HKM+88] Howard, J.H.; Kazar, M.L.; Menees, S.G.; Nichols, D.A.; Satyanarayanan, M.; Sidebotham, R.N.; West, M.J.: Scale and Performance in a Distributed File System ACM Transactions on Computer Systems. Feb. 1988, Vol. 6, No. 1, pp. 51-81.

[IBM 90] Righetti et. al. Redes Academicas . IBM Argentina Diciembre 1990.

[ILC+95] Ingham, D. B.; Caughey, S. J.; Little, M. C.; Shrivastava, S.K.:"W3Objects: Bringing Object-Oriented Technology To The Web,"The Web Journal, 1(1), pp. 89-105, *Proceedings of the 4th International World Wide Web Conference*, Boston, USA, December 1995.

[ICL96] Ingham, D. B.; Caughey, S. J.; Little, M. C.: "Fixing the Broken-Link Problem: The W3Objects Approach," Computer Networks and ISDN Systems, 28(7-11), pp. 1255-1268, *Proceedings of the 5th International World Wide Web Conference*, Paris, France, May 1996.

[ICL97] Ingham, D. B.; Caughey, S. J.; Little, M. C., "Supporting Highly Manageable Web Services," *Proceedings of the 6th International World Wide Web Conference*, Santa Clara, California, USA, April 1997.

[ICP] Internet Cache Protocol Specification I.4.

[Ing97] Ingham, D. B. et al.; Flexible Open Caching for the Web. *In Proceedings of the 6th International World Wide Web Conference*, Santa Clara, California, USA, April 1997.

[Kah90] Kahle, B. et al. Wais Interface Prototype Functional Specification , Thinking Machines Corporation, Abril 1990.

[KAM95] Kappe, F.; Andrews, K.; Maurer, H.: "The Hyper-G Network Information System," J.UCS Vol. 1, No. 4 (Special issue: *Proceedings of the Workshop on Distributed Multimedia Systems, held in Graz*, Austria, Nov. 1994), pp. 206-220, Springer, April 1995.

[Kap95] Kappe, Franz .Maintaining Link Consistency in Distributed Hyperwebs. Proceedings of the INET 95. April 1995.

[KBM94] Katz, E. Kazar; McGrath, M.; R. A Scalable HTTP Server : The NCSA Prototype . Computer Networks and ISDN Systems Vol 27 , September 1994.

[Knut73] D. Knuth, The Art of Computer Programming, Vol. 3, Addison Wesley, *Reading, MA,* 1973.

[KS98] Krishnan, P.; Sugla, Binay: Utility of Co-operating Web Proxy Caches,.*7th International World Wide Web Conference*, Australia , April 1998

[KW97] Krishnamurthy, B.; Wills, C.E.: Study of piggyback cache validation for proxy caches in the World Wide Web, in: *Symposium on Internet Technology and Systems, USENIX Association,* December 1997,

[KW98] Krishnamurthy, Balachander; Wills, Craig E: Piggyback Server Invalidation for Proxy Cache Coherency.*7th International World Wide Web Conference*, Australia , April 1998

[LA94] Luotonen, A.; Altis, K.: World Wide Web Proxies. *First International Conference on the World Wide Web,* Geneva May 1994. Y en Computer Networks and ISDN Systems 27(2), Elsevier Science BV 1994.

[LA94] Luotonen, A.; Altis, K. World Wide Web Proxies . Computer Networks and ISDN Systems. First International Conference on the World Wide Web, Elsevier Science BV May 1994.

[LAJ98] Labovitz C, Ahuja A, Jahanian F. Experimental Study of Internet Stability and Wide-Area Backbone Failures.

[Lam83] Lampson, B. W. Hints for Computer System Design. *Proceedings of the Ninth ACM Symposium on Operating Systems.* Bretton Woods,NH, October 1983

[Lam86] Lampson, B. W. "Designing a Global Name Service". Proceedings of the Fifth ACM Annual Symposium on Principles of Distributed Computing. Calgary, Canada. August 1986. Pp 1-10.

[LB95] Luotonen, A.; Berners-Lee, T.: *CERN httpd Reference Manual. The World Wide Web Consortium* - July 1995.

[LC97] Liu, C.; Cao, P.: Maintaining strong cache consistency in the World-Wide Web, in: *Proc. of the 17th IEEE International Conference on Distributed Computing Systems*, May 1997.

[Lit89] Little M. Goals and Functional Requeriments for Inter-Autonomous System Routing RFC 1126. SAIC. October 1989.

[LOM93] Kurt Lidl, J. Osborne y Joseph Malcom . Drinking form the Firehose: Multicast USENET News .

[LWY93] Leff, A.; Wolf, J.L.; Yu, P.S.:Replication algorithms in a remote caching architecture, *IEEE Transactions on Parallel and Distributed Systems*, 4(11), 1993, pp.1185–1204.

[Mark96] Markatos, E.; Main memory caching of Web documents, *Computer Networks and ISDN Systems,* 28, 1996, pp. 893–905.

[McL93] L. McLoughlin. Mirroring Software disponible June 1993.

[MH92] Muntz, D.; Honeyman, P.: Multi-level caching in distributed file systems - or - your cache ain't nuthin' but trash. *USENIX Winter Conference*, January 1992.

[MLB95] Malpani, Radhika; Lorch, Jacob; Berger, David.: Making World Wide Web Caching Servers Cooperate . *In Proceedings of the fourth International Conference on the World Wide Web.* Boston, MA , pp107-117 December 1995.

[MQR+80] Mc Quillan, John; Richer, Ira; Rosen, Eric C. "The New Routing Algorithm for the ARPANET". *IEEE Transaction on Communications. Vol 28-num 5. May 1980.*

[Moc87] Mockapetris, P. Domain Names - Concept and Facilities RFC 1034. Noviembre 87.

[Mog94] Mogul, J.: Recovery in spritely NFS, Computing Systems, 7(2): 201–262, Spring 1994.

[Moy94] Moy, John. "Multicast Extensions to OSPF". *RFC 1548. March 1994.*

[MPS89] Mishra S. , Peterson L., Schlichting R., Implementing fault-tolerance replicated objects using Psync. Proccedings of the 8th Symposium on Realiable Distributed Systems (Seattle WA) pp 42-52 October 1989.

[MS97] Manley,S.; Seltzer, M.: Web facts and fantasy, in: Symposium on Internet Technology and Systems, USENIX Association, December 1997.

[Nab97] Nabeshima, M.: The Japan Cache Project: an experiment on domain cache, in: *6th International World Wide Web Conference*, Santa Clara, CA, 1997.

[Neu92] Neuman, B.C. Prospero: A Tool for Organizing Internet Resources Elec. Netw. Appl. Pol 2, 1. 1992. pag 30-37.

[Nie95] Nielsen, J. "Multimedia and Hypertext". *Academic Press San Diego. 1995.*

[NHM+98] Nishikawa, Norifumu; Hosokawa, Takafumi; Mori, Yasuhide; Yoshida, Kenishi; Tsuji, Hiroshi: Memory-based Architecture for Distributed WWW Caching Proxy *7th International World Wide Web Conference*, Australia , April 1998.

[NWO88] Nelson, M. Welch B. Ousterhout J. Caching in the Sprite File Systems. *ACM Transactions on Computer Systems.* Vol6 n 1 pp 134-154 February 1988.

[Obr94] Obraczka, Katia. "Massively replicating services in wide area internetworks". Ph.D. Thesis. University of Southern California. December 1994.

[Oca96] O'Callaghan, Daniel A Central Caching Proxy Server for WWW users at the University of Melbourne Department of Information Technology Services, University of Melbourne, Australia – November 1996.

[OD83] Oppen, D. C.; Dalal, Y. K. "The Clearinghouse: A decentralized agent for locating named objects in a distributed enviroment". *ACM Transactions on Office Information Sytems. Vol 1 - num 3. July 1983.* Pp 230-253.

[Pax96] Paxson V. End-to-End Routing Behavior in the Internet. *In Proceeding of the ACM SIGCOMM'96*. Stanford CA August 1996.

[Pos85] Postel, J.; Reynolds, J. File Transfer Protocol RFC 959. Octubre 1985.

[Pop85] Popek; Walker; Kiser; English; Matthews; Butterfield; Thiel . "The LOCUS Distributed System Architecture". *The MIT Press Cambridge London England. 1985.*

[Pov95] Povey, Dean A Distributed Internet Cache School of Information Technology, University of Queensland, Australia - November 1995.

[PR94] Pitkow J.E.; Recker, M.M.: A simple yet robust caching algorithm based on dynamic access patterns, in: *2nd International World Wide Web Conference*, Chicago, IL, 1994.

[RCC97] Rosson, Mary;. Carroll, John M; Chin, George: Participatory analysis: shared development of requirements from scenarios, ; conference proceedings on Human factors in computing systems , 1997, Pages 162 – 169.

[RSR99]. Romano S. Scherman P. Righetti C. Un Gateway NFS-CODA

[Salz92] Salz, Rich: InternetNetNews : Usenet trasnport for Internet sites , - Open software Foundation Summer'92 USENIX - June8-12 , 1992 San Antonio Tx.

[Satya88] Satya et al:.Scale and perfomance in DFS Communications of ACM

[Sat89] Satyanarayanan , M . Integrating Security in a Large Distributed Systems . ACM Transactiones on Computer Systems 7, 3 August 1983 , pp 247-280.

[Sat93] M. Satyanarayanan , Distributed File Systems , Distributed Systems 2 ° Edicion Addison Wesley Pag353 , 1993.

[SBS94] Spasojevic , Mirjana; Bowman, Mic; Spector, Alfred: Using Wide_Area Systems Within the World-Wide Web . *Second Conference WWW* September 1994

[SBS94] Mirjana Spasojevic , Mic Bowman , Alfred Spector Using Wide-Area Systems Within the World-Wide-Web In Electronic Proceedings of the Second World Wide Web , Chicago IL , October 1994 http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/DDay/spasojevic/paper.html

[SF98] Mariela Sassi, Ariel Fabian . EROS: Un servicio para los robots de descubrimiento de recursos Tesis de licenciatura. Facultad de Ciencias Exactas y Naturales. Universidad de Buenos Aires. Septiembre de 1998.

[Sha92] Shapiro, M.; Dickman, P.; Plainfosse, D.: "Robust, Distributed References and Acyclic Garbage Collection," *Symposium on Principles of Distributed Computing*, Vancouver, August 1992.

[SK92] [27] Satyanarayanan, M.; Kistler, James. "Disconnected Operation in the CODA file system". *ACM Transactions on Computer Systems. Vol 10 - num 1. February 1992.* Pp 3-25.

[SM97] Semeria, C.; Maufer, T. "Introduction to IP Multicast Routing". <draft-ietf-mboned-intro-multicast-00.txt>. 3Com Corporation. January 1997.

[Smi78] Smith , A. J.: Bibliography on Paging and Related Topics. *Operating Systems Review*, Vol 12 , pp 39-56 . October 1978.

[SP94] Michael Schwartz y Calton Pu . Applying an Information Gathering Architecture to Netfind : A White Pages Tool for a Changing and Growing Internet. . IEEE/ACM Transaction on Networking Vol 2, NO 5 October 1994 . pp 426-439.

[SRC84] Saltzer, J. H. , Redd, D. Clark D. D. End-to-End Arguments in System Design. *ACM Transactiones on Computer Systems Vol 2, 4* November 1984 , pp 277-288.

[SS94] M . Satyanarayanan , Mirjana Spasojevic An Emprical Study of a Wide-Area Distributed File System , MDA972-90-C-0036 ARPA order number 7312 November 1994

[SS94] Spasojevic, M., Satyanarayanan, M. A Usage Profile and Evaluation of a Wide-Area Distributed File System . *Proceedings of the USENIX Winter Technical Conference* Jan. 1994, San Francisco, CA.

[SS96] Satyanarayanan, M., Spasojevic, M. AFS and the Web: Competitors or Collaborators? *Proceedings of the Seventh ACM SIGOPS European Workshop* September 1996, Connemara, Ireland.

[SSV97] Scheuermann, P.; Shim, J.; Vingralek, R.: A case for delay-conscious caching of Web documents, in: *6th International World Wide Web Conference*, Santa Clara, CA, 1997.

[VF95] Viles, C.L.; French, J.C. "Availability and Latency of World Wide Web Information Servers". *Computing Systems. Vol 1. 1995.* Pp 61-91.

[WA93] Wang, R.; Anderson, T, . xFS : A Wide Area Mass Storage File System. TR CSD-93-783. University of California. 1993.

[WA97] R.P. Wooster and M. Abrams, Proxy caching that estimates page load delays, In: 6th International World Wide Web Conference, Santa Clara, CA, 1997.

[WAS+96] Williams, S.; Abrams, M.; Standridge, C.R.; Abdulla, G.; Fox, E.A.: Removal policies in network caches for World Wide Web documents, in: Proceedings of the ACM SIGCOMM, 1996, pp. 293–304.

[WC97] Wessels, D.; Claffy, K.: Internet Cache Protocol (ICP), Version 2, RFC 2186, National Laboratory for Applied Network Research, UCSD, Sept. 1997.

[WC97] Wessels, D.; Claffy, K.: Application of Internet Cache Protocol (ICP), Version 2, RFC 2187, National Laboratory for Applied Network Research, UCSD, Sept. 1997.

[Wei94] Weider C. Wild Beasts and Unapproachable Bogs Proceedings INET'94/JENC5 123-1,123-3.

[Wei94a] Weider: A vision of an Integrated Internet Information Service Deutsch - Bunyip Information Sustems , Inc .July 1994.

[Wes95] Wessels, Duane. Intelligent Caching for World Wide Web Objects. Master's Thesis University of Colorado y Proceedings INET-95. 1995.

[WPD88] Waitzman, D.; Partridge, C.; Deering, S. "Distance Vector Multicast Routing Protocol". *RFC 1075. November 1988.*

[W3096] MEETING ON WEB EFFICENCY AND ROBUSTNESS Cambridge , Massachusetts Abril de 1996 .

[Wor94] Worrell, Kurt . Invalidation in Large Scale Network Object Caches. Master's Thesis, Department of Computer Science, Faculty of the Graduate School of the University of Colorado, Boulder, CO – 1994

[YM98] Yu, Philip S.; MacNair, Edward A.Yu .Performance Study of a Collaborative Method for Hierarchical Caching in Proxy Servers. *7th International World Wide Web Conference*, Australia , April 1998.

[Zip49] Zipf, G. K. Human behavior and the principle of least effort: An introduction to human Ecology Cambridge , *MA Addison*-Wesley 1949.

Apendice 1

Politicas de Reemplazo

Cuando un web proxy que recibe una solicitud de un recurso que no está en cache, siendo *S* el tamaño del objeto referenciado por la URL, las posibles politicas de reemplazo pueden ser :

- *LFU* : reemplazar el objeto Menos frecuentemente usado.
- *LRU*: remover los objetos Menos recientemente usados. Cuando el espacio libre en la cache es menor a S, descarta el/los objetos menos recientemente usados hasta que el espacio en cache sea por lo menos *S*.
- *LRU-MIN [ASA+95]*: Busca de minimizar el número de objetos reemplazados. LRU no produce una buena perfomance, pues ignora el tamaño de los objetos. Ante la necesidad de cachear un objeto grande puede reemplazar muchos objetos pequeños con la consiguiente disminucion de perfomance. Una mejor politica es la utilización de LRU-MIN. Basicamente el algoritmo es el siguiente : Sea L una lista y T un entero. 1) T = *S* 2) Colocar en L todos los objetos iguales o mayores que T. 3) Utilizar LRU en L hasta que la lista esté vacia o el espacio libre de cache sea menor T. 4) Si el espacio libre de cache no es menor que *S*, hacer T = T/2 e ir a 2).

-*LRU-THOLD*: [ASA+95]: es otra version de LRU en la cual los objetos mayores que un cierto umbral no son cacheados, buscando evitar a lo igual que LRU-MIN que un objeto grande comparado con el tamaño de la cache cause el reemplazo de muchos objetos pequeños. Si el tamaño del disco es pequeño comparado con el tamaño de cache necesario para que no occurran reemplazos en la cache, LRU-THOLD mantiene el tamaño de la cache moderado pero otorga una performance comparable a LRU-MIN. La desventaja de LRU-THOLD es que el valor óptimo del umbral depende principalmente de la carga y del tamaño disponible para cache.

- *LRU-ADAPTABLE*: [Mar96] Es similar a LRU-THOLD, salvo que el umbral no es fijo, sino que es calculado dinámicamente por un algoritmo el cual trata de sintonizar el umbral adecuado. La política es una variación de LRU, cuando se acaba el espacio de la cache, los documentos que son mayores que el umbral son desalojados usando LRU. Cuando todos estos ya han sido desalojados de la cache, el resto de los documentos son desalojados por LRU hasta que el espacio de la cache sea suficiente.
En [Mar96] estudiando los resultados obtenidos llega a la conclusión que el umbral óptimo depende de el tamaño de los documentos solicitados y del tamaño de la cache. La elección del umbral adecuado para obtener una buen performance es un procedimiento delicado. Desarrollaro una política de caching adaptable para estimar el mejor umbral en tiempo de ejecución sin la intervención del usuario. La intuición detrás de esta política es simple. Se debe comenzar con un umbral inicial, periódicamente incrementar / decrementar el umbral. Si la performance obtenida es mejor, continuar incrementando/decrementando sino comenzar a decrementar/ incrementar el umbral.

- LAT: *El objetivo es mantener en cache los documentos que tienen el tiempo de recuperación más largo.*

Proponen en [WA97] dos algoritmos de remoción nuevos que tienen en cuenta el tiempo que toma traer un documento y estimaciones de las condiciones de la red. En definitiva la politica de remoción en función del BW y la demora en traer un documento.

Como hemos visto hasta ahora los algoritmos de cache no toman como parámetro de entrada el tiempo que toma traer un documento remoto y ponerlo en la cache. No obstante, parece natural que una desición de descartar de la cache un documento traído desde un sitio lejano, hablando en terminos de latencia de la red, debería tomarse con mayor cuidado que la desición de descartar un documento traído desde un sitio mas cercano. El algoritmo de reemplazo de la cache debería tener en cuenta el tiempo de recuperación asociado con c/ documento en la cache. Un modo de hacer esto es asignar un peso a cada item en la cache, y usar un algoritmo de reemplazo basado en dicho peso el cual reemplaza el item con el menor peso corriente en la cache. En general el peso podría ser una función del tiempo de la última referencia al item, el tiempo que tomo traerlo, el tiempo de vida esperado para el item, el tamaño del documento, etc.
Un algoritmo de remoción que incorpora tiempo de recuperación debería aparecer directamente dirigido a experiencias en las demoras de los usuarios para acceder sitios sobre enlaces debiles como los utilizados en Latinoamerica o entre U.S.A y Europa.

- *Hibrido:* **[WA97]** Se propone un algoritmo híbrido que tiene en cuenta diversos factores, tratando de mantener en la cache objetos de servidores que tienen un alto tiempo de recuperacion, que deben ser cargados sobre los links de Internet mas lentos, que han sido referenciados muy frecuentemente, y que son pequeños. El algoritmo trabaja estimando la demora de la carga de la página Web o próximo al ancho de banda del Web server usando páginas recién alcanzadas. Tienen un proxy cache en el cual corren algoritmos de reemplazo simultaneamente: LRU, LFU, SIZE (tamaño del documento), LAT e Híbrido. De esta manera pueden comparar resultados en tiempo real y con la misma carga de trabajo. Encontraron que la utilizacion de LAT usualmente trabaja peor que LRU, LFU y SIZE. Sin embargo, el algorítmo de reemplazo Híbrido fue el mas robusto que estudiaron, la mayor parte del tiempo este dio la mejor performance sobre todas las medidas, optimisando bandwidth de la red, carga del server, y tiempo de baja carga, o su performance fue estadisticamente indistinguible con la mejor politica de reemplazo.

- ***LNC-R:[SSV97]*** Este tambien es un Algoritmo de reemplazo conciente de la demora. En [1] argumentan que que maximizar la tasa de cache hit solamente no garantiza el mejor tiempo de respuesta a los usuarios de la Web, sino que también se debería minimizar el costo de cache miss. Los documentos que toman un largo tiempo para ser traídos a la cache deberían ser preferentemente retenidos en la cache. Definen una medida de performance llamada *tasa de demora de grabado* la cual genera la métrica tasa de hit por considerar los costos de cache miss. En particular LNC-R Maximiza la tasa de demora de grabado, este algoritmo se aproxima al algoritmo óptimo de reemplazo de cache, pero es NP-completo. El diseño de LNC-R tiene una sólida fundamentación teórica. Estima la probabilidad de futuras referencias a un documento moviendo el promedio de las últimas K horas de requerimientos a el documento.

- ***LNC-R-W3:[SSv97]*** Los usuarios de la Web tienen una fuerte preferencia por acceder documentos pequeños. Uitilizan una función de ganancia que combina el efecto de demoras y preferencia a los documentos pequeños para minimizar el tiempo de respuesta percibido por los clientes. Obtiene beneficios de performance sobre una mas confiable estimación de tasas de documentos referenciados basados sobre un movimiento de ventanas promedio de las últimas K referencias. La experimentación demostró que LNC-R-W3 provee consistentemente mejor performance que LRU y LRU-MIN. Dependencia entre la tasa de referencia y el tamaño de un objeto. Correlación entre demora a cache y tamaño (25% clientes < 1Kb.