Improving the Availability of Internet2 Applications and Services

(Extended Abstract)
To be submitted to Global Internet 99

G. Carpenter, G. Goldszmidt
IBM Research, Hawthorne
M. Beck, M. Swany
University of Tennessee, Knoxville
B. Dempsey, D. Weiss
University of North Carolina at Chapel Hill

Abstract

We describe the methods of content replication and service resolution location that we plan to use for the Internet2 Distributed Storage Infrastructure (I2-DSI) project. The resolution service will take into account a variety of factors, so that its choice improves the global welfare of the applications. If the resolution is done poorly, applications will not be able to take full advantage of the I2-DSI resources and will likely underperform. We propose combining two resolution technologies to address this problem: SonarDNS *and* Narwhal. *SonarDNS measures and orders resources based on a variety of criteria, and exports its recommendations via DNS. Narwhal provides an application-transparent proxy that can configure, monitor, manage and provide additional functionality to network applications. We intend to deploy extremely lean servers on the periphery "ramp" of I2 that integrate the functionality of both components. Such co-location provides a focal point for administrative control, traffic monitoring, implementation of differentiated services, and enforcement of traffic policies. Content will be replicated using Rsync+, an efficient technique that leverages an existing differential file update algorithm and enables the use of multicast transport solutions for performance and scalability.*

1. Introduction

The aim of the Internet2 (I2) initiative is to facilitate collaboration among researchers in all academic fields. To support this, a variety of techniques and technologies will be developed and put into service over an advanced network infrastructure separate from the current "commodity" Internet operated by commercial Internet Service Providers. The *Internet2 Distributed Storage Infrastructure* project (I2-DSI) will provide replicated services over a large set of servers interconnected via high-speed links [BM98]. The I2-DSI will support the development and testing of services, such as broadcast quality video, and will provide a development test-bed for new applications to enable collaborative research and distance learning [DSI-A]. These applications or "Internet Content Channels" [BM98] will involve large volumes of multimedia data that cannot be reliably delivered in real-time over today's "*commodity*" Internet.

These applications, from replicated information to distributed services, will benefit from being implemented on the DSI platform consisting of servers at multiple sites distributed across the network. Indeed, much of the frequently accessed content on the commodity Internet is replicated in some fashion. It might be done locally and transparently, as in the case of a server cluster farm hosting a popular Web site. It may also be done in an explicit, ad-hoc fashion, such as mirrors of software distribution sites. Reference [GS97] describes some of the problems related to distributing, accessing and managing a complex, worldwide Web site for a high volume sports event on the commodity Internet.

One objective of this research is to determine the most efficient mix of networking and storage for delivering content and ensuring quality of service. Since the I2-DSI channels will serve geographically dispersed users, they will need services that provide for intelligently storing copies of their digital content in close network proximity to their users. The current initial deployment (see Figure 1) has a combined capacity of almost six terabytes, with server nodes installed at some sites directly attached to the Internet2 backbone networks (the vBNS and Abilene) and some removed from any backbone.

The use of replication in I2-DSI immediately creates a need for effective *resolution* of service requests: real-time selection of an appropriate instance of a replicated service. The I2-DSI project aims to resolve accessibility issues associated with sharing and using educational content by creating an infrastructure of server channels for academia [DSI]. Such channels, called *Internet*

Content Channels, enable replicated collections of files to be transparently delivered to end-user communities at a chosen cost/performance point via a flexible, policy-based application of resources. Although files are replicated over multiple domains and multiple servers, a single URL is utilized to access a given file, regardless of the network address of the server called to service the request. This will be accomplished through a combination of techniques that dynamically select the best server to service each request.

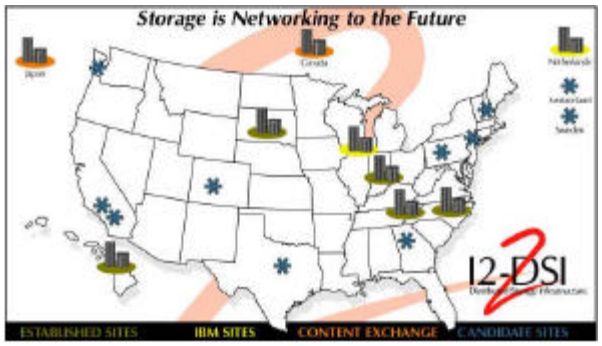

Figure 1. Current Infrastructure Deployment of I2-DSI.

Applications. Many potential applications for I2-DSI were presented at [DSI-AW]. These include:

- Digital libraries, such as
  - *Variations* that provides access to over 5000 titles of near CD-quality digital audio and
  - the California State University Image Consortium that has digitized and catalogued over 12,000 images to-date for art history education.
- Document repositories, like the Internet standards collection at *Normos.org*.
- On-line publishing services such as the Columbia EARTHSCAPE project.
- New applications in scientific collaboration models such as
  - the GIOD project to address the data storage and accesses problems posed by the next generation of particle collider experiments which will start at CERN in 2005
  - and the University of North Carolina distributed, virtual laboratory project that is advancing nanotechnology through development of virtual reality interfaces to scientific instruments.

The rest of this paper is organized as follows. Section 2 describes the problems that we are attempting to address. Section 3 briefly outlines the characteristics of the technologies that we are planning to use, and Section 4 concludes.

2. Problems

2.1 Resolution

A fundamental issue in the I2-DSI project is the *resolution* of the location of an appropriate server to fulfill each service request. Each server will carry a particular set of channels, is located at a different network "proximity" to the client, and is under a different load than the other servers, to name just a few differentiating factors. Further I2-DSI offers the ability to "delegate" a set of channels to a server located on a campus or in a department, complicating the task of resolution. If the resolution is done poorly, applications may have suboptimal performance, perhaps even performing worse than if there were no replication. For example, if the resolution chooses a server that is not currently functioning, the application may stall. Ideally, the resolution service will take into account a variety of factors, so that its choice improves the global welfare of the applications.

2.2 Dynamic Allocation.

In addition to network proximity, the dynamic allocation mechanisms of I2-DSI will need to keep track of the current state of the servers, and provide an effective mechanism to allocate the workload. Such allocation mechanism will need to be distributed and highly available.

On the current commodity Internet, most servers could be defined as being in one of four states, arbitrarily denoted green, yellow, red and black. The green state represents a server with mostly free resources, that is, a server that will welcome additional requests. The red state represents a server that is currently unable to take on new work, as some resources are over-committed. The yellow state represents intervals where the server is not in either green or red state, that is it is not idle, yet retains some spare capacity. The black state represents a server for that there is not known information at this time. For servers with some known state, a plot of their resource utilization will appear similar to a sine wave. Consequently, some inference about current CPU load can be made from previously known state information.

Unfortunately, as network bandwidths approach or exceed the ability of host CPUs to transfer data from memory to the network, the availability of a CPU will appear more like a square sine wave. This will greatly

reduce the utility of mechanisms that use non-real-time data for decision-making purposes.

2.3 Replication.

I2-DSI uses replication in conjunction with the resolution service as a means to achieve high performance in the delivery of services. Replica servers offer identical services and client requests are resolved to one copy. Developing an extensible, portable, and scaleable architecture for server-based replication will require research and experimentation over a range of different problems. For example, a central long-term task is to develop an abstract server and a portable content model for replication [WCW99].

In this paper, we address the low-level replication issue of efficient data movement between I2-DSI servers. While multiple replication mechanisms will eventually be deployed within I2-DSI, the replication process inherently requires data movement from a content gateway to multiple remote servers. Given this situation, a design goal for all replication mechanisms is that they leave open the possibility of using multicasting protocols. As compared with multiple one-to-one transmissions, multicasting offers the efficiency gains of using less network bandwidth, CPU processing, and other computing resources along with the delivery speedups of transmitting a single copy of the data. The Internet2 community has made support for network-level multicast a priority, and reliable multicast protocols are an active area of research within the IRTF with both research and commercial solutions available.

2.4 Advanced Network Capabilities.

A number of advanced network capabilities (including Multicast, IPv6 and Quality of Service connections) will be supported in the Internet2 backbones, but are either difficult to deploy on campus networks or difficult to configure on end-user workstations. I2-DSI will enable the creation of applications that use these capabilities by enabling them on the I2-DSI hosts and then making them available to channel developers through an appropriate API. For example, I2-DSI applications may use IPv6 or Multicast to communicate amongst themselves, but then communicate with end-user workstations using IPv4 or streaming protocols when necessary. The benefits of the advanced networking will only apply on the inter-server communication, but, in some cases, this is an improvement over no use of these capabilities at all.

2.5 Monitoring and Management.

Shared, production-quality services such as I2-DSI are made more robust by the inclusion of remote monitoring and management capabilities. Narwhal, described in section 3.3, uses The Enterprise Management Protocol (TEMP) to provide such management capabilities. An extensive, albeit not exhaustive, list of reasons why existing management infrastructures are not able to handle all the issues of scale posed by client-side Narwhal deployments appears in [TEMP]. Server-side Narwhal deployments represent a much smaller sized problem and could conceivably be dealt with using a conventional approach, such as SNMP [SNMP]. Because of the restrictive nature of SNMP (lexicographic ordering of variable names, scalar-only data types, polling-based, platform-specific sub-agent technologies), we opt instead to utilize the more powerful and efficient capabilities of TEMP.

3. Solution

We describe here on-going work in defining solutions for resolution and replication mechanisms in the I2-DSI project.

3.1 Resolution Components

We propose combining two techniques to address the resolution problem: SonarDNS [MS99] and Narwhal [CG99]. SonarDNS is a system of measuring and ordering resources based on a variety of criteria, and exports its recommendations via DNS. Narwhal provides an application-transparent local proxy that enables dynamic configuration, monitoring and management of networked applications. The combination of these approaches yields a comprehensive, transparent solution to the problems noted above. We intend to deploy lean servers on the periphery "ramp" of I2 that integrate the functionality of both components. The advantages of such co-location include:

- Focal point for administrative/management control, including monitoring traffic.
- Control point to implement differentiated services and enforcement of traffic policies.

3.2 SonarDNS

SonarDNS is a mechanism for enabling what we call *"proximity resolution"*. This term is used to denote the process whereby the "best" of a set of replicated resources is chosen and the client is directed to that

resource. The early SonarDNS work has focused on defining "best" in terms of network proximity, although other possibilities exist. SonarDNS consists of two entities: the SONAR daemon and a modified Domain Name System (DNS) server [RFC1035]. The SONAR daemon implements the SONAR protocol [MS98], which allows network proximity metrics to be queried, thereby informing network-aware applications about the state of the network and its resources.

The implementation of SonarDNS provides proximity resolution by the ordering of Internet addresses returned from requests to the DNS. The modified DNS server makes a request to the SONAR daemon to perform this ordering. The client can then be presented with all resulting Internet addresses, ordered by preference. This technique is similar to that employed by Cisco Systems' product Distributed Director [DD]. SonarDNS makes these ordering decisions primarily on the client-side (although it can operate on the server-side as well), while Distributed Director must be installed on the server-side

The SONAR daemon currently uses two mechanisms to make determinations regarding the order of addresses. The first mechanism is passive and relies on topology information that is learned from the Border Gateway Protocol (BGP). This information demonstrates the distance to locations in the Internet and is the same information that the routers use to forward traffic to these destinations. The second is an active probe using round trip time measurement, which provides a much finer-grained measure of network distance. These mechanisms can be used separately or together. The combination of active and passive network characterization techniques allows SONAR to utilize the benefits of each.

The deployment of the I2-DSI testbed is providing insights into appropriate uses of these measurements. Further, current work is investigating the application of these same techniques to Intranets via interactions with internal routing protocols such as Open Shortest Path First (OSPF) and alternate methods of presentation.

3.3 Narwhal

Narwhal [CG99] is a client-side, dynamic server switching method that improves the availability and performance of network mediated applications. In its load distribution role, Narwhal can be considered the client-side complement of a server-side solution, such as Network Dispatcher [GH97]. A complete Narwhal system consists of a set of global management applications coupled with local (client-resident) intermediary brokers that can intelligently route the traffic among intermediaries (e.g., content replicas). Its benefits include (1) improved availability of intermediary services, (2) load sharing of requests across several intermediaries, (3) bypass of intermediaries whenever possible, and (4) remote administrative control enabling the implementation of domain-specific policies to utilize the shared, limited networking resources. For example, interactive data can be given higher priority than other non-critical data at the client side. Narwhal Client Agents (NCAs) work at the granularity of TCP connections: each new connection is allocated to the best server at the time.

NCAs can be completely controlled by remote management applications using a newly developed management protocol called TEMP [TEMPSPEC]. For example, TEMP will allow the dynamic configuration of NCAs on mobile clients, enabling plug-in-and-go operation. More sophisticated management applications implement globally optimized resource reservation policies that prevent the "tragedy of the commons" syndrome, in which each client tries to locally maximize its individual performance to the detriment of all. Once suitably configured, an NCA provides improved availability of intermediary-based services without requiring modification of existing applications. An NCA will autonomously avoid the use of poorly performing intermediaries and spread workloads across several intermediaries. NCAs are also able to perform protocol conversions and route requests through or around particular intermediaries as appropriate. For instance, NCAs may delegate certain queries through special services, such as HTML pages through an annotation service. Some requests may be routed through alternate service (like requests for GIF images to a proxy cache). NCAs can also give priority to certain kinds of traffic (such as WWW browsing or telnet sessions). This feature permits interactive data transfers to be responsive while a bulk data transfer (like an ftp) is going on at the same time. When a NCA detects a problem with an intermediary, it makes the information available to the management applications at the server. Such management applications in turn can proactively inform other NCAs about the detected problem, enabling them to avoid interactions with the troubled intermediary.

3.4 Rsync+: An Efficient Mechanism for Content Replication

We are modifying a popular open-source file mirror tool, rsync [RSYNC], to create a replication mechanism for content channels. Mirroring tools for site-to-site file synchronization are widely used in the current commodity Internet, and rsync represents a mature, flexible solution. Rsync runs on several platforms, has a rich feature set, and, most distinctively, supports differential file update using a novel checksum algorithm on file blocks.

In its current form, however, rsync carries out point-to-point file synchronization in a single network session between the two sites. To enable its use with reliable multicast protocols, we have added a mode to create an new rsync, which we call rsync+, that decouples the file update process from network transmission of this information. Under rsync+, differential file update information can be captured in a local file, and this file is then transmitted to remote DSI servers for replica update. The solution leverages the efficiency gains of the differential file update algorithm in rsync while also allowing, where operationally feasible, the network transmission to be done with a reliable multicast protocol.

In a preliminary experiment with rsync+, we have created a local mirror of a large portion (8GB) of the Linux software and documents at http://metalab.unc.edu/pub/Linux/. Figure 2 shows the total number of bytes for all files added or modified in each 12-hour interval (blue bars) and the total data bytes that the rsync+ algorithm will move over the network to perform these updates (red bars). The data shown covers 5 days in March 1999. As shown in the Figure, the rsync algorithm has a modest effect for some update periods, those in which updates are largely new files added to the archive. In other periods, however, differential updating is effective since changes to the archive include file modifications as well as additions [DW99].

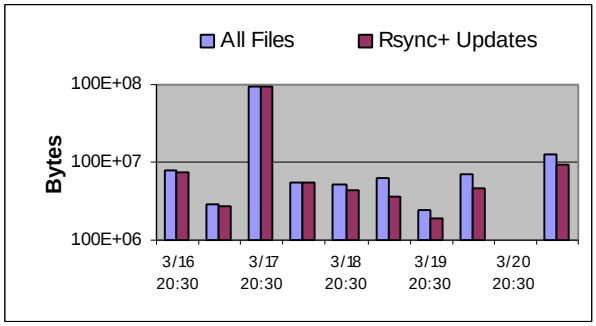

Figure 2: Local Experiment with 8GB Linux Archives

4. Conclusions

The I2-DSI project aims to create a scalable, heterogeneous middleware architecture for a distributed services infrastructure that will benefit the education and research community. By leveraging technology trends in mass storage and high-speed networks, I2-DSI will enable important new applications. The envisioned I2-DSI approach is founded on two fundamental architectural points: replication and transparent resolution. We have described here important issues and on-going research programs that are developing the mechanisms for replication and resolution services.

5. References

[BM98] Beck, M. and Moore, T., "The Internet2 Distributed Storage Infrastructure Project: An Architecture for Internet Content Channels", in Computer Networking and ISDN Systems, 1998, 30(22-23): pp. 2141-2148.

[DD] Cisco Distributed Director, http://www.cisco.com/warp/public/751/distdir/

[DSI-AW] I2-DSI Applications Workshop, http://dsi.internet2.edu/apps99.html

[DSI-A] "Internet2 Distributed Storage Initiative Deploys First Systems to Support Advanced Network Applications", http://www.internet2.edu/html/8_february_1999.html

[DW99] Debra Weiss, "An Efficient, Scaleable Replication Mechanism for the I2-DSI Project", School of Information and Library Science, University of North Carolina at Chapel Hill, Master's Thesis, May 1999 (expected).

[DSI] - Distributed Storage Infrastructure, http://dsi.internet2.edu/

[CG99] Carpenter, G. and Goldszmidt, G., "Improving the Availability and Performance of Network Mediated Services", Submitted to INET 99.

[GH97] Goldszmidt, G. and Hunt, H., ShockAbsorber: A TCP Connection Router, In Proceedings of the 2nd Global Internet Conference, GLOBECOM, Phoenix, Arizona, November 1997.

[GS97] Goldszmidt, G., and Stanford-Clark A., "Load Distribution for Scalable Web Servers: Summer Olympics 1996 - A Case Study", In Proceedings of the 8th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management, Sydney, Australia, October 1997.

[MS98] Moore, K., Swany, M., "Sonar: A Network Proximity Service", Technical Report, University of Tennessee Department of Computer Science.

[MS99] Swany, M., "Proximity Resolution with Sonar and SonarDNS", Submitted to USITS99.

[RFC1035] Mockapetris, P., "Domain Names – Implementation and Specification", RFC-1035, November 1987.

[RSYNC] http://samba.anu.edu.au/rsync

[TEMP] "Enabling the Management of Everything using TEMP", G. Carpenter. Submitted to Globecom99.

[TEMPSPEC] "The Enterprise Management Protocol", G. Carpenter, Unpublished Report.

[WCW99] Beck, M., T. Moore, B. Dempsey, and R. Chawla, "Portable Representation of Internet Content Channels in I2-DSI", in 4th International Web Caching Workshop, San Diego, CA, April 1999.