The Internet2 Distributed Storage Infrastructure Project: An Architecture for Internet Content Channels

Micah Beck and Terry Moore
mbeck@cs.utk.edu, tmoore@cs.utk.edu
Innovative Computing Laboratory
Department of Computer Science
University of Tennessee, Knoxville

Keywords: replication, server channel, Internet2, I2-DSI.

1. Internet Engineering Issues and the I2-DSI Channel Strategy

The mission of the University Corporation for Advanced Internet Development's Internet2 project is to accelerate the next stage of Internet development in academia. One approach to this mission is the development of new networking paradigms based on technologies that are not available on the commodity Internet. Another approach, taken by the ***Internet2 Distributed Storage Infrastructure*** project (I2-DSI) is to improve the engineering characteristics of services currently available to allow them to be used in ways that are not currently practical. The resulting architectural developments may be as important to the commodity Internet as to next generation applications.

The performance problems of the commodity Internet are well known: Not only can access times be long or unpredictable, availability can be unreliable. Because the performance of the Web is dependent on the real-time performance of wide area networking it cannot be improved without the global application of resources. Finally, and of most importance to the Web caching community, the Web is wasteful in its use of wide area networking due to repetitive transmission of the same data. While caching and mirroring strategies continue to evolve to address these problems, the underlying client/server architecture of the Web does not allow us to improve access to certain services by applying resources to those services alone. This structure inhibits the development of a viable economic model for differential investment in new infrastructure.

But the need for further action on this front becomes all the more urgent as some of the underlying causes of these problems get worse over time. For instance, new media types and complex services delivered across the Web, as well as the exponential growth of the Web user population, are putting an ever-increasing load on network backbones and servers. Moreover, the Web and the Internet are constantly being used in new ways and for mission critical activities, such as the ubiquitous delivery of new software (e.g. Netscape Communicator). The I2-DSI project is based on the premise that any new infrastructure designed to improve Internet performance should also accommodate the relentless movement toward innovation and expansion of services.

The I2-DSI strategy is to promote the development of innovative applications by focusing on *Internet channels* that enable differential investment in infrastructure to support specific services and content. In our sense *a **channel** is a collection of content which can be transparently delivered to end user communities at a chosen cost/performance point through a flexible, policy-based application of resources*. Three key elements of this definition need to be clarified:

- ***Collections of content*** — The kinds of content that can go into a channel is a superset of what is accessible via the Web. It includes the familiar collection of portable files and programs available through Web protocols (e.g. text, graphics, Java applets, multi-media, etc.) and also services that utilize non-Web protocols, such as backend database systems and remote search engines. But an aggregation of such content becomes a channel only when it is more or less deliberately *collected* and made available to end users. For example, the entire ensemble of digital content used during an academic course — papers, videos, application and applet software, large data sets for analysis, running simulations, on-line examinations, etc. — could be made into a channel.
- ***Policy based applications of resources*** — Channels use a combination of network and storage resources to localize collections of content that people want, so that it can be access it quickly and reliably when they want it. Any given channel will manage its resources according to policies implemented by its creators, the owners of the infrastructure, and the users. So-called "push" channels bring collections of content all the way to the desktop machine, and therefore need no re-engineering of the infrastructure. *Server channels*, of the kind I2-DSI is designed to support, allow the application of resources at the server level, enhancing the delivery of collections of content for a wider population of users.
- ***Transparent delivery to end users*** — The default Internet paradigm is to resolve a domain name to a single host in a globally consistent manner. The Web builds on this model by supporting a uniform naming scheme (viz. URLs) for objects stored on servers and accessible via Web protocols. Server channels continue this paradigm by redirecting service requests to localized content replicas using a transparent access mechanism that preserves the global consistency of domain name resolution. For instance, two students taking our hypothetical course in different parts of the world could click on a the same hyperlink on their teachers Web page and be directed to **local** copies of the same object

While other ways of using distributed storage resources to address the performance problems of the Internet support some of these features, I2-DSI's focus on the creation of server channels can support them all in an open way. But to see that this is so, we need to examine and compare the alternatives in more detail.

2. Caching, Mirrors and Server Channels

The success of Web caching is due largely to the fact that it transparently addresses a prime cause of the Web's inefficient use of bandwidth. Specifically, the Web model does not take account of, and therefore cannot make use of, the relative network proximity between client and server. In most cases, a URL determines the network location of the server with no possibility of redirection. The strategy of Web caching is to introduce both a level of indirection and a distance metric, and then to use these mechanisms to automatically redirect user requests to the best available cache server. In this way Web caching both replaces wide area network access with more local access and offloads service requests to the Web cache hierarchy. As reflected in Table 1 below, the effects of this strategy are twofold: it reduces average access times and it increases the efficiency with which wide area networking is used.

Table 1: Comparison of Caching and Mirroring

Goals of Web engineering	Caching	Mirror
Support transparent redirection of service requests	✓	
Reduce access times	✓	✓
Use wide area bandwidth efficiently	✓	✓
Take account of network proximity	✓	✓
Make access times more predictable		✓
Support the application of resources to get proportionately better service		✓
Support high-bandwidth & compute intensive services, e.g. multimedia		✓

But as this table also shows, caching does not address the problem of long access times, or the unpredictable nature of access times. Moreover, it does not give us a way to improve the performance of certain services by an expenditure of resources proportional to the level of improvement we want to see. Lastly, it does not easily handle some significant protocols for new media types and services, such as streaming audio and video. When these remaining problems are of paramount importance, the deployment of mirror sites is the commonly used remedy.

Mirrors do indeed achieve significant goals for Web engineering that caches do not. But they do so at a very high price from the user's point of view. Since mirrors do not support the automatic redirection of service requests, requiring instead that users manually choose a particular mirror, they burden the user with the problem of determining which URL represents the host that is "best" for them. Users, however, are poorly positioned to make this choice. The performance of different mirror servers is unpredictable and the consistency and correctness of the replicas at a given site is generally unknown. (For example, see the status page on mirror sites for the popular Netlib software repository. [1]) In consequence most publishers and information providers do not find mirrors valuable enough to invest the resources necessary to set them up and operate them well. One goal of I2-DSI is to make professional quality replication open to more general use by promulgating standard interfaces and practices.

From its beginning, the I2-DSI project has recognized the importance of retaining the power of user level transparency that Web caching provides. We developed the concept of an Internet channel described above from the idea of preloading caches with definite collections of content, and we began there precisely because caching's ease of use is so important.[2] But the effort to surmount caching's limitations as a channel delivery mechanism eventually forced us to consider the alternative approach. As a result, the I2-DSI architecture is a form of mirroring which, when *combined with a layer of indirection or resolution,* regains some of the power of the cache

hierarchy that standard mirroring forgoes. We hope to build on the Internet2 community's level of technical advancement to introduce resolution services that have not yet emerged in the commodity Internet and that make I2-DSI's Internet channels and the replicated storage services that underlie them transparently accessible in a way that current mirrors are not.

Space not permitting a more thorough treatment, Table 2 presents an overview of the advantages that can be achieved by server channels as compared to the standard approaches to caching and mirroring:

Table 2: Comparison with I2 Server Channels

Features of alternative approaches to distributed storage infrastructure	**Caching**	**Mirror**	**Server Channels**
Saves bandwidth	+	–	–
Open and dynamic	+	–	–
Permits content filtering/monitoring	+	–	–
Scalable	+	+	+
Improves end-user performance	+	+	+
Maintain global name space	+	–	+
Deployable	–	+	+
Facilitate technologies that support wholesale delivery of information services (e.g. multicast support)	–	+	+
Respect the semantics of http	–	+	+
More reliable	–	–	+
Extensible to new protocols	–	+	+
Provide support for active server content and cookies	–	+	+
More deterministic (so you *depend* on it for better performance).	–	–	+
Accountability to content provider	–	+	+

3. I2-DSI Architecture

The I2-DSI architecture has three main components: a *system of replicated servers*, *a mechanism to transfer files* from the file systems of content authors to the replicated servers, and *a mechanism for the intelligent redirection of service request*s to particular replicas.

3.1. A System of Replicated Servers

The key architectural feature of our design is that a server can be located at any point in the network in order to achieve the desired proximity to clients and other servers. This allows I2-DSI to flexibly deliver the needed performance characteristics through differential investments in servers and bandwidth. One key factor underlying this architectural element is that the network requirements of client-server interactions may change significantly with the introduction of a new replica server. Once we understand the trade-offs, this design will allow us to engineer the use of different network segments to fit cost and performance parameters.

The system of replicated servers supported by I2-DSI has two basic components: a set of core servers located at strategic points in the network, and a mechanism for delegating service of a particular channel to a local server.

3.1.1 Core Servers

The purpose of the core servers is to provide the broadest possible access to I2-DSI content by implementing reliable, high-quality mirroring. The cost of maintaining these servers will be borne by the project sponsors and contributors, including government grants. A fee structure is under consideration to enable us to support users who require substantial server resources.

One problem encountered by current caching technology and mirror sites is that the replica servers are not perceived as giving the content provider control and accountability. That is, a cache or mirror is not always an acceptable surrogate for the content provider’s own server. While there is a proposed extension to HTTP 1.1, which allows caches to report back to the origin server, [3] the issues of whether and how those solutions will be competently and consistently applied remain unsettled. The I2-DSI takes a combined business and technical and approach to this issue: the servers used to implement I2-DSI will be operated by Internet2, an organization with a known reputation, and according to strict standards of accountability.

3.1.2 Delegated Servers

A delegated server is a secondary participant in the replication scheme. It supports (or subscribes to) a subset of the channels served by I2-DSI and serves them to a restricted set of clients, as shown in Figure 1:

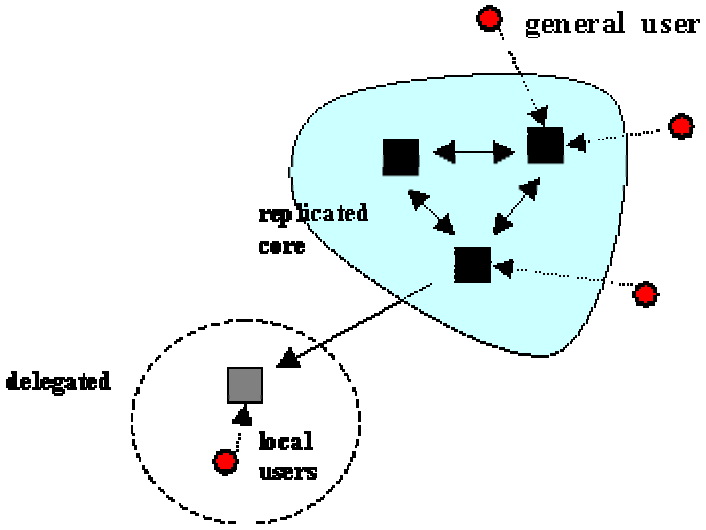


Fig. 1: Access to I2-DSI core and delegated servers

For instance, a campus server could be a delegated server for the channels that delivers collections of content of interest to the campus community and that serve only clients within that community. While we will initially support delegation only on the basis of static policy, a more dynamic, usage sensitive system is possible. Such a system would more closely approach the functionality of caching. There are technical problems in the design of HTTP servers and clients which may make it difficult to use server delegation as a caching mechanism for Web pages, but it could eventually evolve into a cache for FTP and other protocols.

Responsibility for I2-DSI services will be delegated to other organizations only under agreements that allow Internet2 to continue meeting its obligations to content providers. Internet2's role is to serve all of its member institutions by enforcing the controls that will allow them to interact and share resources freely. Delegation will have the effect of improving the performance of channel delivery on a given campus without forcing that campus' server to spend external bandwidth servicing off-campus requests.

3.2 Distribution of Content

In general I2-DSI services are implemented on a set of replication hosts using a collection of portable files and programs, uniformly available backend services (e.g. a database system), and assorted network-accessible services (e.g. remote search engines). In building replicated servers the basic choices are the hardware and software platform, the file replication mechanism, and the selection of other backend services. For the purposes of I2-DSI, we treat *portability* as relative to

the set of replication hosts. If that set of replication hosts is single-vendor, then proprietary solutions may be acceptable. Uniformly available backend resources must be replicated using techniques specific to each resource. Network-accessible services are considered to be uniformly available.

To distribute files through this replicated I2 storage service, a user (say a faculty member with a series of course modules, laden with Java simulations and multimedia) interacts with the storage service through a simple Web form interface (Fig.2)

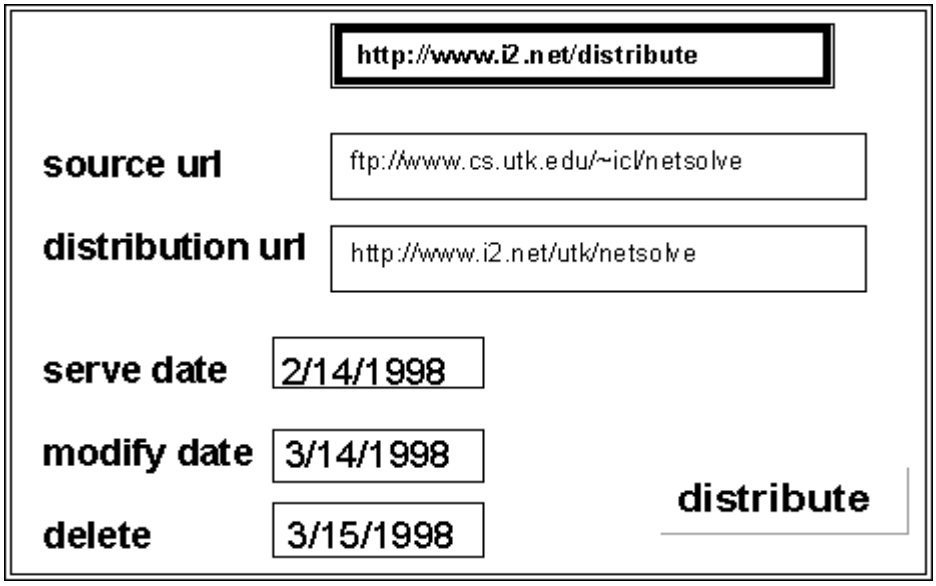


Fig. 2: Prototype author submission interface

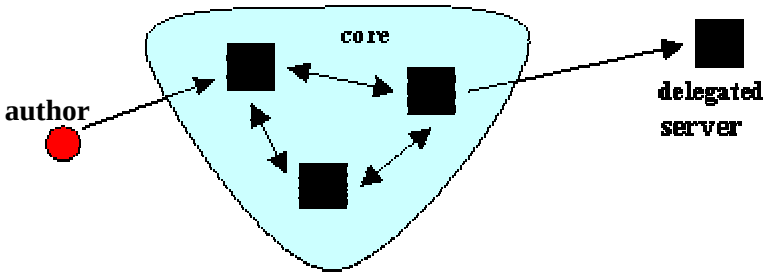
This form displays several key properties of the service:

1. The files to be distributed are specified by a single URL, which names either
    a. a directory to be copied *in its entirety* using FTP, or
    b. an (optionally compressed) archive file (tar, zip or binhex), or
    c. a Web site to be copied by a Web crawler
2. Files will be distributed but not served until the serve date. A new version cannot be distributed until the modify date. After the modify date, the files will continue being served until the delete date or until a new version is distributed.
3. After submitting the distribution form, the distribution status can be queried. It will respond that the files have not yet been retrieved, are retrieved but not fully distributed, or are fully distributed and are being served.

The distribution architecture is both simple and powerful. There is a set of core servers that fully replicate a set of distributed domains. We consider that a copy of the domain i2.net is present on every core server, but other domains may be fully replicated on some subset of the core servers.

The distribution function described above will eventually be distributed across the core servers, but may initially be centralized (Fig 3).

Fig.3: Publication model for I2-DSI



3.3 Resolution

The possibility of using any one of a number of servers to fulfill a given service request opens up many possible criteria for choosing the server. The most fundamental question is this: Which server is most capable of fulfilling the request? The answer to this question has both static and dynamic aspects. We can statically determine which servers have the fundamental resources required to fulfill the request. Dynamic aspects include the current load on the server and the level of congestion on the network connection to it. In order not to burden the user with the job of resolution it is necessary to implement a ***resolver*** which performs the task. The use of a resolver creates a level of indirection between the service requests specified by the user and the service request fulfilled on the network.

Our approach to the static aspect of resolution is to identify a channel with an Internet domain and use the DNS database to store IP addresses of a set of servers all of which have the resources required to fulfill the request. We then use an enhanced DNS server, which we call *SonarDNS*, to implement resolution that uses criteria that are not part of the usual DNS model. [4] For example, *Sonar*DNS will initially use network proximity information obtained from the SONAR service. (see Fig. 4). While the current implementation of SONAR uses only ICMP echo packets to measure round trip delays, improving this implementation is an area of active research.

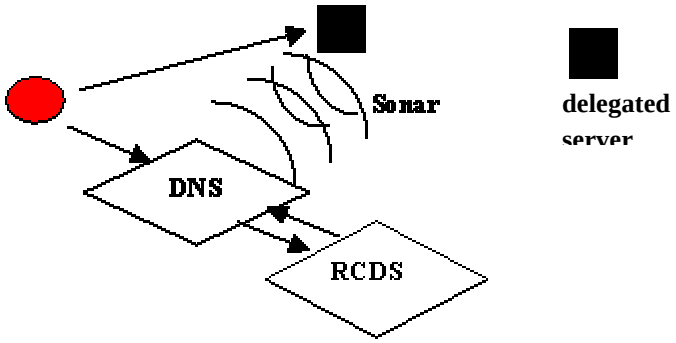


Fig. 4: SonarDNS

We have a number of reasons for choosing a DNS-based deployment strategy rather than one based on a Web proxy, which was our initial impulse. One reason is generality, since the DNS serves all Internet protocols rather than just HTTP and FTP. Another reason is deployment — in

the US, Web caches are not widely used and are viewed with some skepticism by users and system network administrators alike. As a consequence of this decision we have to deal with issues such as backward-compatibility and the lack of expressiveness of the DNS interface. Certainly we do not hold up the current DNS resolution interface as adequate for the evolving needs of the Internet, but we do find it adequate for our present purposes in developing I2-DSI.

A better solution might be obtained either through the evolution of the DNS interface, or by the development of a completely new resolution system. The *Resource Cataloging and Distribution System* (RCDS) developed by Keith Moore at the University of Tennessee's Innovative Computing Laboratory is such a system, engineered to provide more capable and higher level resolution services.[5] We intend to use the Resource Catalog servers which form the core of RCDS when DNS servers are inadequate to our purposes, but for now we will hide the light of this new system under the bushel of DNS compatibility.

Cache servers and other DNS clients can coexist well with intelligent caching as long as they leave the management of multiple IP addresses (A records) to the SonarDNS by always choosing the first record returned, a common but not required behavior. Currently Squid caches DNS responses and manages multiple IP addresses itself, using a round-robin load balancing technique which defeats dynamic proximity measurement. A forced compatibility mode is available which causes it to return only the IP address of the closest host for each DNS request and to report a TTL of zero.

4. Internet2

The status of I2-DSI as part of the Internet2 project at the University Corporation for Advanced Internet Development has several important implications. One is that Internet2 can gain access to network machine rooms and campus facilities across the US and around the world and so I2-DSI can be widely deployed. Another is that research universities, which make up the membership of Internet2, employ some of the most capable network engineers and tend to give them the time and freedom to participate in innovative projects like ours. Most important, it is university researchers who have the greatest need and the best political support for new and demanding applications of internetworking. I2-DSI provides a mechanism for allowing application developers to take advantage of advanced internetworking while continuing to program to a familiar file access API. We present a few scenarios for academic use of I2-DSI below.

4.1 Timely Access to Massive Experimental Data Sets

Data sets on the order of tens of gigabytes in size are generated daily from instruments used in high energy physics research. Geographically distributed researchers need to analyze each day's results, and these analyses form the basis for ongoing discussions that can result in modifications to the experimental regimen. Such discussions require free access to the stored data using visualization tools. With increasing frequency the visualization also needs to take place in real-time collaborative mode between different campuses. A massive data library maintains a complete record of the experimentation in a high latency terabyte store, but the most recent data must be accessible at high bandwidth and low latency to all participants. In addition, researchers who find features of interest in the data need continuing access to those data sets that they choose

for further analysis. Analogous situations exist across the spectrum of fields in science and engineering.

By creating channels for the data sets produced by their experiment, researchers can use I2-DSI storage services to delegate each day's data to a server on their own campus. The flexible delegation mechanism will then allow channels to be replicated where needed, and after some period of time all the replicas at each site can be purged except for the archival store. Because channel replication is transparent to the users, the URLs used to reference this data are globally valid, so that differences in performance correspond to the localization choices made at each participating institution.

4.2 Low Latency File Access for Virtual World Simulations.

One scenario for the application of information technology in higher education calls for the use of a high performance, multimedia client/server protocol that can deliver interactive, three-dimensional virtual worlds through audio, video and tactile interfaces. Such protocols make heavy use of primary memory on the client and, for each virtual world, require low latency, high bandwidth (100 Mb/s) secondary storage on the order of multiple gigabytes, as well as higher latency, high bandwidth tertiary storage on the order of hundreds of gigabytes. Faculty are willing to subscribe in advance to make these worlds accessible to their students, but they want them to be available on demand from student workstations on the campus network.

The use of such a demanding and specialized protocol requires that a channel for virtual worlds be served by a highly capable host system that has a server for that protocol. The low latency, high bandwidth data can be replicated on that server, while the higher latency, high bandwidth data might be stored either at a central repository, or at some regional replica site if overloading of the central host's I/O capabilities were to occur. Advance subscription to the channel for a particular world through I2-DSI storage services would allow the appropriate files to be replicated as needed.

4.3 High-quality Educational Media Delivery over Low Bandwidth Links

Publishers of educational materials for K-12 classrooms are moving aggressively into the use of electronically delivered media, including video, software, and Web-based multimedia. The political rhetoric of the information age is that connection of every school to the Internet allows every child equal access to this wealth of electronic information. The truth is that most schools connect to the Internet at 128Kb/s or less and the delays encountered there rule out use of the Internet in interactive classroom work. It is typically relegated to use as a source of reference materials, and even there it has limitations. Schools with sufficient resources move to higher bandwidth network connections or private networks that offer better access characteristics.

Delivery of educational channels to a school's local area network is an affordable alternative to ever-increasing external bandwidth requirements. The needs of students are largely determined by their curriculum, and so can be anticipated and addressed in advance. However, a channel delivery platform must be available if publishers are going to make it the target of their educational materials. The current infrastructure allows publishers few options: only small files that can be downloaded at low bandwidth will reach most schools, while large files reach only

those locations with premium networking. The I2-DSI replicated storage service can dramatically transform this situation, since it can easily be extended to K-12 schools by the purchase of a cheap server with a big disk. Costly bandwidth can then be reserved for truly interactive purposes, including teacher conferences and professional education.

5. Channels and the Future of the Internet

I2-DSI is creating an infrastructure and defining a strategy for the support of server channels for academia. However, the engineering benefits of organizing Internet content into channels and delivering them through a replicated server infrastructure may be equally important to ISPs and developers of commercial content. The concentration of resources which academia can achieve through government funding and the focused demand imposed by curriculum and research programs are not currently present in the commodity Internet. Today, content from CNN, Disney and Microsoft is carried over the same infrastructure and with roughly the same performance as educational materials and personal home pages. Now consider the market impact of an infrastructure technology that can deliver highly subscribed commercial content reliably, with high performance and full accountability to the content provider. With such technology powerful media conglomerates that stand to benefit from consolidation in the Internet industry may be in a position to determine who fights for access to their content across crowded and chaotic backbones, and who can offer their users advanced services available through proprietary channels. We would prefer to see the Internet community deploy an open system of channel distribution before a closed system is imposed on them.

6. References

1. Netlib Editors. *Netlib Mirrors*, 1998. http://www.netlib.org/bib/mirrors.html.
2. Micah Beck, Joseph Kirby, Terry Moore, Donia Nance, Melissa Wauford. *The Tennessee Cache Box Project* in *NLANR Web Cache Workshop*,1997. Boulder, Colorado. http://ircache.nlanr.net/Cache/Workshop97/Papers/Kirby/kirby.html.
3. J. Mogul, P.Leach., *RFC 2227: Simple Hit-Metering and Usage-Limiting for HTTP*, 1997. http://www.cis.ohio-state.edu/htbin/rfc/rfc2227.html.
4. Martin Swany. *Sonar DNS*, 1998. http://www.cs.utk.edu/~swany/sonardns/
5. Keith Moore, Shirley Brown., Jason Cox, and Jonathan Gettler, *Resource Cataloging and Distribution System*. 1997. http://www.netlib.org/utk/projects/rcds/rcds-tr/.