YouServ: A Web-Hosting and Content Sharing Tool for the Masses*

Roberto J. Bayardo Jr., Rakesh Agrawal, Daniel Gruhl, Amit Somani
IBM Almaden Research Center
San Jose, CA 95120
bayardo@alum.mit.edu, ragrawal@acm.org, dgruhl@almaden.ibm.com, asomani@us.ibm.com

ABSTRACT

YouServ is a system that allows its users to pool existing desktop computing resources for *high availability* web hosting and file sharing. By exploiting standard web and internet protocols (e.g. HTTP and DNS), YouServ does not require those who access YouServ-published content to install special purpose software. Because it requires minimal server-side resources and administration, YouServ can be provided at a very low cost. We describe the design, implementation, and a successful intranet deployment of the YouServ system, and compare it with several alternatives.

Categories and Subject Descriptors

H.3.4 [**Systems and Software**]: *Distributed systems, information networks, Performance evaluation (efficiency and effectiveness)*; H.4.1 [**Office Automation**]: *Groupware*; H.5.4 [**Hypertext/ Hypermedia**]: *Architectures, User issues*.

General Terms

Algorithms, Performance, Reliability, Security, Human Factors

Keywords

Web hosting, peer-to-peer networks, p2p, decentralized systems

1. INTRODUCTION

People want to share files over the internet. Whether web pages, audio clips, photographs, or other content, the preferred method of sharing files is through the web. While the web makes accessing content simple (almost anyone has access to a browser), publishing content on the web is far more difficult both because of technical hurdles and cost.

This difficulty in publishing content has led, at least in part, to alternative peer-to-peer (P2P) file sharing mediums and protocols such as Napster [20] [21] and Gnutella [11]. These systems allow you to easily publish as much content as you can store on your system's hard drive, and are for the most part free. Unfortunately, people who wish to access this content must install special purpose software, which limits the audience capable of reaching the content.

We propose and describe the YouServ system, which exploits peer-to-peer techniques to provide easy to use, low-cost *web* publishing of content. With YouServ, ordinary PC owners can use their own hardware and internet connections to cheaply put content on the web. Unlike standard web server software such as Apache httpd [3], YouServ is a complete web hosting solution in that with YouServ:

- You are immediately assigned a convenient domain name which always directs to your site content, even if your ISP assigns IP addresses dynamically.
- By pooling resources of a group of friends, your content can remain accessible even after your computer is turned off or disconnected from the network.
- You can publish content on the web from your own machine even if you are behind a firewall (or cannot accept inbound connections for any other reason).
- You can secure content on your site without assigning, distributing, and maintaining any accounts or passwords.

While P2P systems such as Napster and Gnutella have also been engineered to overcome some of the problems addressed by YouServ (such as firewalled content access), they do so using proprietary protocols, and hence require proprietary clients for file access. YouServ makes content available to anyone with a web browser.

YouServ is also extremely simple to use. Users who wish to publish a YouServ site on the IBM intranet install the *YouServ peer software* with a standard wizard based installer, enter their corporate intranet id and password, and two clicks later their first file is accessible on the web at a convenient domain name based on their corporate e-mail address. Deployed as such on the intranet, YouServ is a convenient alternative to mailing around e-mail attachments, which consume precious (and expensive) server-side storage.

We see an internet-based deployment of YouServ serving a class of users whose needs are not well met by existing solutions. Free web hosting systems impose highly restrictive space and transfer quotas, and also bombard the site visitor with advertisements, almost certain to discourage repeat visits. Other for-fee web hosting services require high monthly fees (or impose restrictive space and transfer limits), and many in addition require technical expertise beyond that of a typical web user.

YouServ is cost-effective because the machines of its end users do almost all of the work. The only server-side components in YouServ are a dynamic DNS server and the *YouServ coordinator*, which acts as a matchmaking service for the end-user machines and performs other lightweight administrative tasks.

1.1 Paper Overview

We begin with a survey of related work in Section 2. Section 3 describes the YouServ system from the perspective of an end user, and discusses our deployment of YouServ within the IBM corporate intranet. Section 4 catalogs the various components that comprise the YouServ system, and provides protocol and implementation details. Section 5 addresses scalability of the YouServ system, and Section 6 security and access control. Section 7 summarizes the contributions, and concludes with a discussion of future work.

2. RELATED WORK

The usefulness of webservers for sharing files is well known, and numerous organizations provide webserver software allowing a PC to serve its own website. One webserver explicitly geared towards "personal" web serving is BadBlue [29]. In addition to easy to use

*The system as deployed in IBM is known as *uServ*.

WWW 2002, May 7-11, 2002, Honolulu, Hawaii, USA.
ACM 1-58113-449-5/02/0005.

webserving features, BadBlue implements the Gnutella protocol to support dynamic search over BadBlue and other Gnutella-shared content. Other software programs that facilitate web publishing of local files include the MacOS Personal Web sharing function [4] and Microsoft's Personal Web Server. YouServ differs from these personal webservers because it solves the problems that conspire to prevent them from providing a good alternative to paid web hosting services: firewalls, dynamic IP address assignment, identity management and access control, and limited machine uptime or network connectivity.

File sharing is also supported by collaborative work tools such as Lotus Notes [14] and Groove [12], instant messenger systems such as ICQ [13], and the aforementioned Gnutella and Napster/ OpenNap P2P networks. These systems require installation of special software for accessing content. Many people would not be willing to download and install special software just to, for example, casually view some photos from a friend, no matter how streamlined the installation process. (We foresee photo sharing becoming one of the most common uses of YouServ when deployed on the open internet.)

Some collaboration tools such as Lotus QuickPlace [15] support web access of content, but do so through centralized servers, failing to scale to a large number of users without a significant hardware investment and administration overhead. While some decentralized systems such as Gnutella accept the browser-supported HTTP GET command for file downloads, network-specific protocols are required for sharing files when the party hosting the content is behind a firewall. In addition, many Gnutella implementations explicitly prohibit browser access to discourage people from accessing files without sharing any content.

XDegrees Corp. offers software similar to YouServ in that it assigns location independent URLs to files throughout an organization, providing a uniform namespace for the organization's distributed information assets. Components of the XDegrees system also provide intelligent storage services to improve availability and performance. At the time of this writing, the technical information publicly available from XDegrees consists of only a brief whitepaper [30], so a detailed comparison is not possible. The whitepaper does, however, note that XDegrees employs a naming system that resolves the physical location of content based on its entire URL. In contrast, existing Web browsers consult the internet domain name system (DNS) to resolve the machine location using only the domain name portion of the URL. The DNS protocol [18] predates HTTP [9] and has no provisions for resolving IP addresses from a full URL.

There are several other systems implementing file replication and distribution across a network of peers. The Farsite project [5], for example, provides a serverless, fault-tolerant distributed filesystem. Farsite aims to make your personal files always available through replication and redundancy, while using encryption to ensure that files are available only to those who are authorized to access them. Mojo Nation [19] replicates content to support swarm distribution, allowing multiple bandwidth limited peers to cooperatively serve files faster and with improved availability. Unique to Mojo Nation is its use of a digital currency called "Mojo" for reimbursing peers for the resources they contribute. Freenet [7] is a system that supports anonymous, uncensorable content publishing. Freenet replicates content as it is pulled across the network in response to a query, ensuring the most highly accessed content remains the most highly replicated. While similar to YouServ in some respects, these systems do not directly solve the problem of providing highly available, easy to use file sharing via standard web protocols with very low cost. Another important difference is that YouServ replicates content only in response to explicit agreements between end users, rather than having replication driven by uncontrollable protocol decisions. While this limits the extent with which content is replicated and its associated availability benefits, it eliminates objections users often have to unknown content being hosted on their machine (whether or not it is encrypted).

Project JXTA [27] is an open source project led by Sun Microsystems that is creating a common platform for building a range of distributed services composed of addressable and communicating peers running on arbitrary devices. We mention JXTA because it could be exploited to build a YouServ-like system, much as we used the Vinci distributed services architecture [1] in our implementation.

One feature not currently provided by YouServ, but common in other P2P networks including Gnutella and Napster, is the ability to search over dynamic content [31]. Rather than requiring people always search for content, YouServ instead allows its content to be accessed through location independent URLs. Many YouServ sites maintain availability high enough for standard web search engines to index them. Nevertheless, many YouServ sites remain more transient than the typical website. We also expect YouServ sites to be more dynamic, given that adding new content is easy and unconstrained by centralized storage quotas. We therefore believe the improved freshness of search results offered by a dynamic search capability would be of significant value to YouServ users.

One final piece of related work is the WebDAV project [28]. WebDAV is an extension of the HTTP protocol to better support web authoring. YouServ does not currently require or use any HTTP protocol extensions. Instead, it strives to expand the audience that can make use of the existing protocol. Nevertheless, WebDAV support would be another natural extension of YouServ, particularly since recent releases of popular software packages (Microsoft Office) and operating systems (Windows XP and Windows 2000) already support it.

3. A TOUR OF YOUSERV

This section describes our deployment of YouServ inside the IBM intranet, primarily from the perspective of an end user. Note that by *user* we are typically referring to someone who uses YouServ to publish a site. Those who access YouServ-hosted content are a larger class of individuals that subsumes the user population.

3.1 System Usage

YouServ has been running inside the IBM corporate intranet for nine months, where it can be freely downloaded and installed by any employee with intranet access. Files hosted on this internal deployment of YouServ can only be accessed from within the IBM intranet, allowing it to be safely used for work purposes.

The full set of functionality described in this paper has been gradually rolled in over the lifetime of the deployment. For example, access control has been available for only one month, and site replication functionality for four months. We advertise YouServ as an experimental system to be used "at your own risk." Despite the warning, the system has already been used by over 2900 people spread across the world, and the userbase is consistently growing (mostly due to word-of-mouth). Its usage in the presence of our disclaimers demonstrates the need and desire to share files over the web, even in a technology company where there are many high powered and traditional alternatives available for free (e.g. Lotus Notes, other web server tools, and centralized, network-accessible storage).

At the time of this writing, anywhere from 1200-1300 users are active during the week, with 700-800 YouServ sites available simultaneously during peak hours. During non-peak hours, fewer sites are available, bottoming out around 400 during weeknights and 300 over the weekend. Site replication is so far actively used by over 60 users. We believe these numbers are extremely modest

compared to what might be expected from an internet or "official" intranet deployment

3.2 Getting Started

Hosting a site with YouServ is simple: after completing the download and installation of the YouServ peer software, a user must simply enter a valid corporate user ID and password to have his YouServ site become visible on the web. Every employee in IBM has an e-mail address that serves as his or her corporate user ID. YouServ assigns the employee's site a domain name based on this e-mail. For example, the e-mail address bayardo@us.ibm.com maps to the domain name bayardo.userv.ibm.com. Locating someone's YouServ website is then as trivial as looking the person up in the employee directory, or remembering his or her e-mail address.

The first time the YouServ peer software is run on a machine, it creates a brand new empty folder and populates this folder with a default homepage and two access-controlled subdirectories. We elaborate on the access control features provided by YouServ in Section 6. The default homepage is populated with information extracted from the IBM corporate directory, and includes information such as job title, phone number, snail mail, and so on. The user can manually change the shared folder at any time. Content outside the access controlled folders is accessible by anyone on the network capable of determining the URL, e.g. by obtaining it directly, by browsing the site, or by simply guessing it.

To share a file (or an entire folder of files), a user can either copy it to his or her shared folder, or simply right-click over it and select a "Publish to YouServ" (Fig 1) desktop menu option that is installed in Windows environments along with the peer software. When publishing through this menu option, YouServ prompts the user to choose a specific shared destination folder. YouServ copies the file (or folder) to the designated destination and displays a URL that addresses the published content on the web.

YouServ also integrates with Lotus Notes, IBM's standard corporate e-mail client. This feature creates a toolbar icon that mimics the behavior of the standard "create attachment" icon. Instead of creating a true attachment, however, this icon copies the selected file to the user's shared folder, and pastes a URL pointing to the file into the message. This keeps attachments from being replicated across every mailbox receiving the message, and also

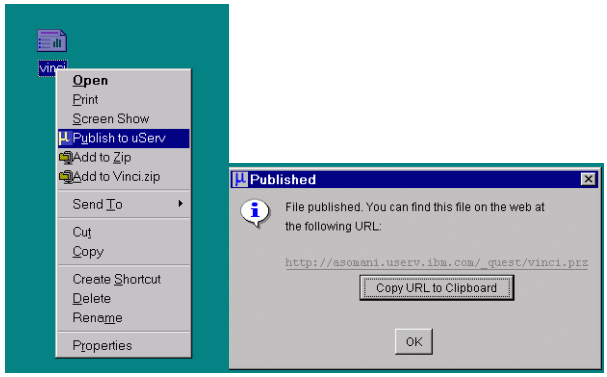

Figure 1. YouServ lets a user quickly publish a file on the web.

allows the sender to monitor when and from where the attached content was viewed.

We have found that most people like to share files without maintaining sophisticated HTML pages linking to them. By default, YouServ lists the contents of any folder visited on the web (Fig. 2). Folder browsing allows users to find content without having to remember the exact URL. Users who do maintain HTML links to their content can rename their homepage.html file (or some other file) to index.html in order to have that file served in place of a folder listing. This behavior is consistent with that of other webserver software including Apache httpd and Microsoft IIS. One unique feature of YouServ is that it allows site visitors to download the entire contents of a shared folder hierarchy with one click in ZIP format (note the link in Fig. 2). Most users find creating ZIP files manually to be too cumbersome to use regularly, thus the feature has proven valuable when sharing multi-file content such as photo albums or source code trees.

The YouServ GUI console (Fig. 3) displays a log that lists any files that have been accessed, when they were accessed, from which host, and the referring page (if available). This log also flags error requests (such as file not found), which facilitates site debugging. The identity of anyone who access a file is also displayed in the log if it can be determined. YouServ requires that anyone who attempts to access secured content reveal their identity

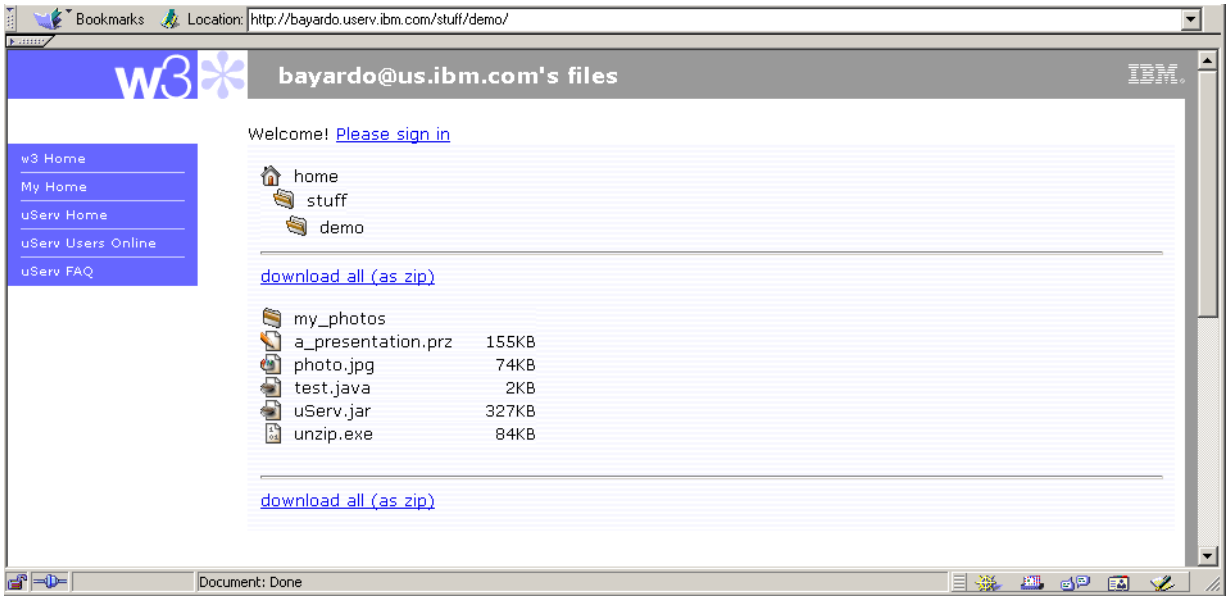

Figure 2. Folder listing provided by YouServ. Note the ability to download all files as a ZIP file in one click.

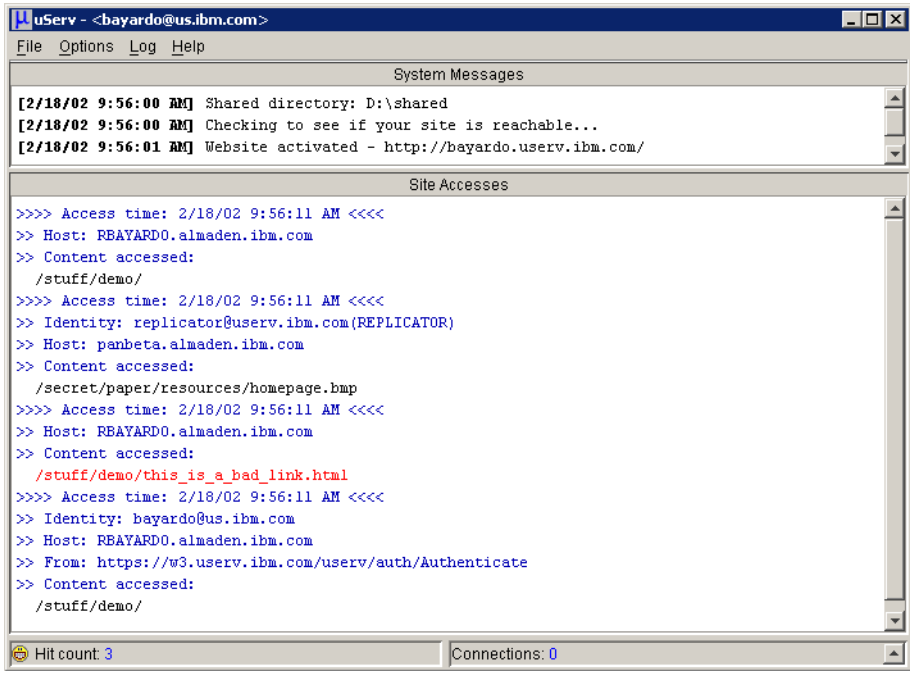

Figure 3. YouServ reports file accesses through its GUI as well as in a more traditional file-based log.

by signing in with the site. Signing in initially involves providing a corporate e-mail id and password to a trusted authentication server (see Section 6.1). Subsequent sign-ins, even with other YouServ sites, requires only a single click for the duration of the browser session.

These "voyeuristic" features give YouServ a distinct advantage over other file-sharing methods such as e-mail attachments, which may remain unopened without the sender ever knowing. Users who are not interested in monitoring the activity of their site can close the GUI window. A YouServ control-tray icon allows the GUI to be restored as desired.

3.3 Replication

We have noted that with typical personal webserver software, content becomes inaccessible once the hosting machine goes offline. This malady greatly hinders asynchronous collaboration, in which it is not known exactly when a shared file will be downloaded. YouServ supports site replication and shared hosting in order to overcome this problem. Any YouServ user can list other users ("replicators") who are willing to host their content when they are offline. These other users must also specify the users whom they are willing to host ("masters"), thereby enforcing a two-way agreement. Many groups or teams in our company have at least one member who is willing to leave a desktop machine running continuously. This member is typically used as a replicator by the other members of the team. Some people have multiple machines, e.g. a desktop and a laptop machine. Such users tend to use their desktop machine as a YouServ replicator system and maintain the master copy of their site on their mobile laptop. Other users are satisfied with transient availability, using YouServ primarily to serve files on the spot with others in a face-to-face meeting or in an instant messaging conversation.

Site replication is performed transparently to the user. Once the masters and replicators are specified, replicas synchronize with the master site automatically, and replicas are activated automatically by the YouServ coordinator when the user disconnects, even when the user does not cleanly shut down. Those who access YouServ content are also oblivious to replica usage -- content is always accessed through the same location independent URLs regardless of how or from where it is served. Replication implementation and protocol details appear in Section 4.2.2 and Section 4.3.

3.4 Proxying

Approximately 20% of the current YouServ userbase use machines that cannot accept what are known as *inbound port 80 connections*, which must be allowed for standard web server software to function. Several reasons prevent inbound port 80 connections, the most common of which is firewall software. Corporate security guidelines often mandate mobile laptop owners install firewall software. Others install firewall software because of general security concerns. While firewall software can be configured to allow inbound port 80 connections, quite often this configuration step is beyond the capability or patience of the average user. Virtual private networks (VPNs), network address translators (NATs) [26], and even the presence of other webserver software running on the same machine can also forbid or otherwise prevent YouServ from accepting inbound port 80 connections.

In order to accommodate this population, YouServ provides what we call *peer-to-peer proxying*. Put simply, members of the YouServ community who are able to accept inbound port 80 connections can be called upon to accept them on behalf of users who are not. Other systems such as Groove, JXTA, and Mojo Nation employ similar relay techniques for traversing firewalls.

A user who accepts connections on the behalf of someone else is referred to as a *proxy*. By default, any user who runs YouServ is willing to serve as a proxy for at most 4 other users. Users can change this limit or even disable the feature completely.

The YouServ peer software that a user runs on his own machine to publish a YouServ site will detect if a proxy is needed when it first starts up. Should a proxy be needed, the YouServ coordinator forwards the contact information of another YouServ user willing to serve as a proxy. The user's machine connects to the proxy's machine which will then accept connections on his behalf. More technical details on proxying are provided in Section 4.2.3.

As with replication, the use of proxies in YouServ is for the most part completely transparent to the end user. Whenever a proxy is used, YouServ will non-intrusively indicate which YouServ user is serving as the proxy, and also encourage the user to check if his or

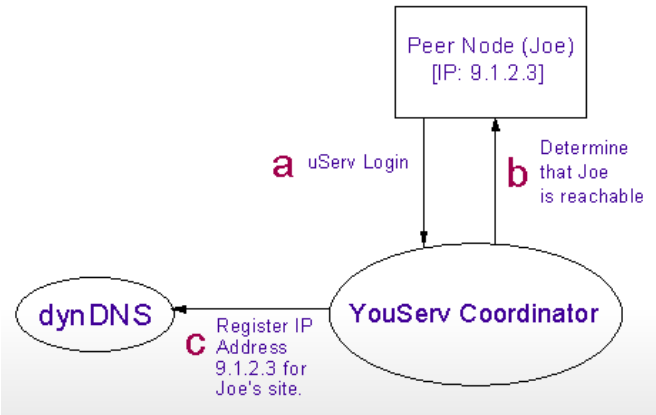

Figure 4a. A peer node that can accept inbound connections comes online.

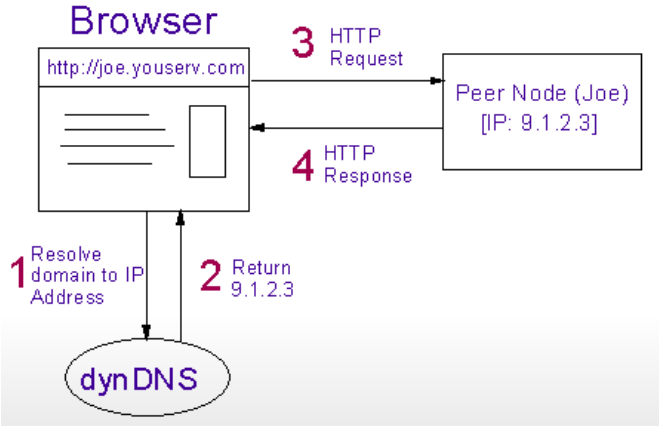

Figure 4b. Accessing content from a peer node that can accept inbound connections.

her machine can be reconfigured so that proxying is not necessary. YouServ also informs users who serve as proxies precisely when and for whom they are serving. We have found a wide majority of users (>80%) are willing to serve as proxies for the community. In most cases, a user notices no performance or bandwidth degradation when serving as a proxy. Proxying only consumes bandwidth when someone is downloading files from the proxied user's site. The CPU time consumed by proxying is typically negligible.

4. TECHNICAL DETAILS

This section describes the inner workings of YouServ. We overview the various components that comprise the YouServ system, and detail how they interact and operate to provide the features outlined in the previous section.

4.1 System Components and Protocols Overview

The components of the YouServ system are as follows:

- Browsers - any machine running a standard web browser and accessing YouServ content.
- Peer nodes - the machines of users who have set up a YouServ site by running the YouServ peer software. These components do all of the "heavy lifting" in that all content is served directly from them, not any centralized resource.
- Dynamic DNS - a centralized component that speaks the DNS protocol for resolving YouServ domain names to machine IP addresses.
- YouServ Coordinator - a centralized component that provides user authentication, proxy and replica matchmaking, IP sniffing and firewall detection, site availability monitoring, and other administrative tasks. The coordinator is the first contact point of any peer node, which must authenticate itself before YouServ will set up the appropriate domain name to IP mapping with the dynamic DNS component.

The communication protocols used in YouServ are DNS and HTTP (for supporting standard web browsers), and YouServ specific protocols that are implemented using Vinci libraries [1]. The Vinci system includes high performance libraries supporting XML document exchange via a lightweight XML document encoding. Vinci applications can be easily evolved without breaking existing code. This evolvability proved invaluable during YouServ development, allowing us to release numerous upgrades of the software with improved functionality, all the while maintaining backwards compatibility without adding undue complexity to our code-base. Because Vinci libraries are compact and fast, this evolvability comes with no serious drawbacks compared to using a low-level structure oriented (and hence brittle) binary protocol such as those employed by Napster, Gnutella, and many other P2P applications.

4.2 Accessing YouServ Content

We have allocated a subdomain (userv.ibm.com) specifically for our YouServ deployment, and assign each user's site a unique name in this space corresponding to his or her IBM e-mail username. For example, the domain name bayardo.userv.ibm.com is assigned to bayardo@us.ibm.com. YouServ takes advantage of dynamic DNS so that a browser can map these domain names to the location (IP address) of an available peer node capable of serving the requested YouServ content. In a typical scenario, the YouServ DNS maps a domain name to the machine of the user to whom the domain name belongs. However, should this machine be offline, it could instead map to the machine of another peer node who is capable of serving the content from a site replica. In the third case, if the user's machine is firewalled, this could instead map to a machine which is serving as a proxy for the site.

The YouServ implementation uses BIND [2] to provide the dynamic DNS service. Recent versions of BIND allow updates to be performed on a running nameserver. This allows the YouServ coordinator component to immediately push any updates to the DNS server. These entries have a very short time to live (2 minutes), ensuring that changes in the hosting machine are quickly propagated (e.g. if the host goes offline and a replica takes over).

We describe in turn each of the three different modes in which YouServ content can be served.

- Basic: A peer node is online and capable of accepting inbound connections, and serves its own site.
- Peer-hosted: A peer node is offline, and a replica of its site is served by another peer node.
- Proxied: A peer node is online but unable to accept inbound connections, and serves its site through a proxy which accepts connections on its behalf.

To keep the presentation clear, we only describe cases where a peer node assists exactly one other peer node by either serving as its proxy or serving a replica of its site. Any peer node in YouServ is capable of simultaneously serving as a proxy for multiple users at once, while also simultaneously serving replicas of multiple YouServ sites

4.2.1 Scenario 1: Basic

Fig. 4a depicts the initialization step for the first scenario where a user's machine is capable of accepting inbound connections. The peer node, in the depicted case run by a user named Joe, comes online and authenticates itself with the YouServ coordinator. The

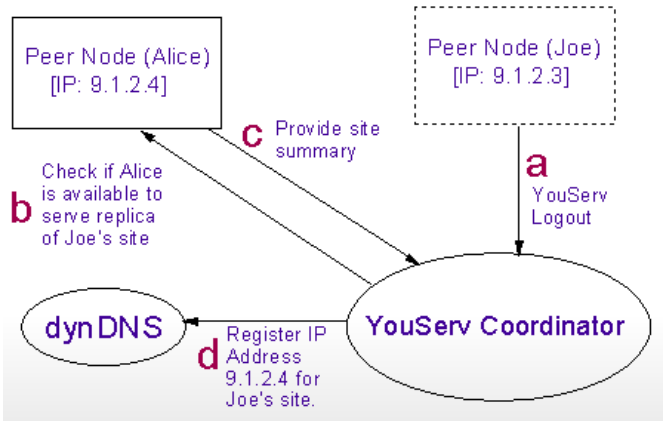

Figure 5a. A peer node goes offline and another node that is replicating his site takes over.

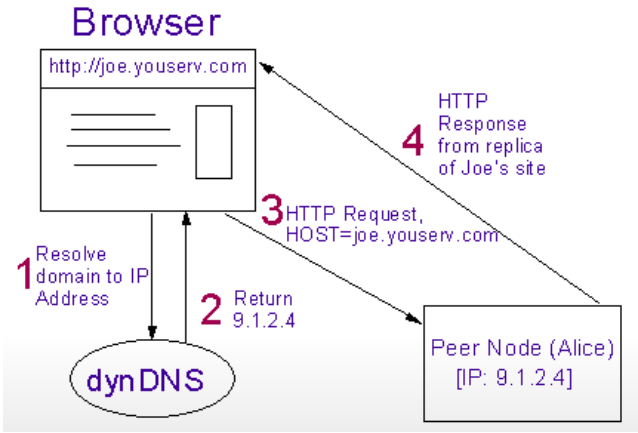

Figure 5b. Accessing content from a site whose owner is offline.

YouServ coordinator successfully establishes a connection back to Joe's peer node which signals that it can accept inbound connections. The coordinator immediately updates the DNS entry of Joe's site with the IP address of Joe's machine.

In Fig. 4b, a standard web browser is used to access Joe's site from some arbitrary network connected machine. The user of the web browser does not need to install YouServ or other special software on this machine. The browser resolves Joe's domain name to Joe's machine, and executes an HTTP request to retrieve the desired content. Though the figure depicts the browser communicating directly with the YouServ DNS, the DNS protocol allows the browser to communicate with a local nameserver or fetch the IP address from a local cache. Ultimately, however, the domain name to IP mapping information arises from the YouServ dynamic DNS component.

This scenario is also supported by dynamic DNS services existing on the internet today [8] (minus the inbound connection check). The YouServ system is unique in that it can also serve content in two additional ways for higher availability.

4.2.2 Scenario 2: Peer Hosted

The next figure (Fig. 5a) depicts a scenario where Joe's machine goes offline (for whatever reason). Sometime before going offline, Joe and Alice agreed to allow Alice's peer node to serve Joe's content while Joe's peer node is unavailable. (The protocol for maintaining replicas is discussed in Section 4.3.) When Joe disconnects, the coordinator will check if Alice's peer node is available and willing to serve Joe's content. Alice indicates willingness by returning a site summary (essentially a checksum plus timestamp) of Joe's site. The coordinator may use this summary to determine whether to activate Alice's replica. In our implementation, if Alice is the only replica, the coordinator will activate her replica unconditionally. The summaries are only used to determine which of multiple replicas are the most up to date. Assuming Alice is the only available replica, the coordinator activates the replica by updating the IP address for Joe's site to the address of Alice's machine. If a replica goes offline, the coordinator will determine whether another replica is available and activate it. Should no replica of Joe's site be immediately available, the coordinator will monitor newly active peers in case one should come online.

After the replica of Joe's site is activated on Alice's machine, web requests for Joe's content are directed to Alice's peer node. Alice's peer node checks the value of the HTTP HOST header within the incoming request. Browsers will set the HOST header value to the domain name used to resolve the IP address of the requested site. In this case the web request will contain Joe's domain name, which causes Alice's peer node to return the requested content from the replica of Joe's site (Fig. 5b). Delegating requests based on the HOST header is known as *virtual hosting*.

4.2.3 Scenario 3: Proxied

In the next and final scenario (Fig. 6a), imagine again that Joe is online and capable of accepting inbound connections. Another user, Bob, comes online and registers with the coordinator, which is unable to open a new connection back to him. Bob's peer node recognizes that it didn't receive the expected connection from the coordinator, indicating it is incapable of accepting the necessary inbound connections to serve its own content. Bob's peer node therefore requests that it be directed to an available proxy. The coordinator responds with contact information of an available proxy, in this case Joe. The coordinator returns its response through the connection established by Bob, so an inbound connection is not needed to get this information to him.

Contact information consists for the most part of an IP address and an authenticating token. Bob's peer node uses this contact information to establish an outgoing, persistent TCP/IP connection to port 80 on Joe's peer node, and reports back to the coordinator that a proxy connection was successfully established. The coordinator updates the DNS entry of Bob's site with the IP address of Joe's peer node. This persistent connection will be used to forward content requests from Joe's peer node to Bob's without having to establish an inbound connection to Bob. Should the persistent connection fail at any time, Bob's peer node will immediately attempt to re-establish it. If unsuccessful, then the scenario restarts with Bob asking the coordinator for contact information of another available proxy.

The protocol spoken across the persistent proxy connection is not HTTP, but instead a YouServ-specific protocol similar to BEEP [24] that allows multiple requests to be served in parallel on a single connection. A special protocol is used here because the HTTP protocol allows no more than one request be active at a time on a single connection. While HTTP/1.1 supports pipelining of requests on the same connection, this would not prevent someone with a slow connection who is downloading a large file from blocking out all other requests during the transfer. In addition, browsers will often open multiple concurrent connections to a site at once to, for example, allow multiple images to load concurrently. By using a special protocol, peer nodes can parallelize proxied content requests while maintaining only a single persistent connection. This protocol has an added benefit of avoiding the high connection establishment overhead of multiple concurrent HTTP requests. Note that this special protocol need

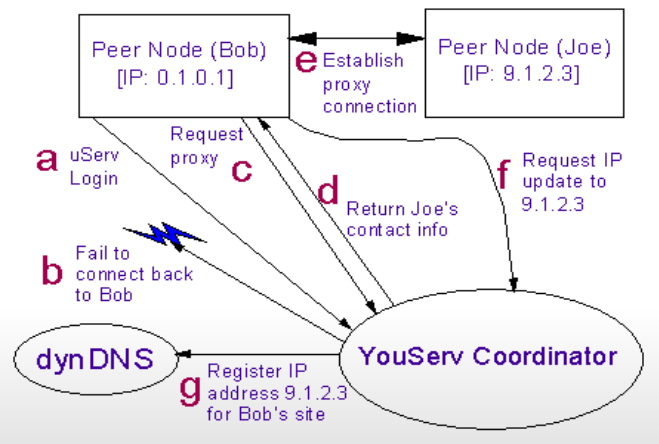

Figure 6a. A peer node that cannot accept inbound connections (Bob) comes online.

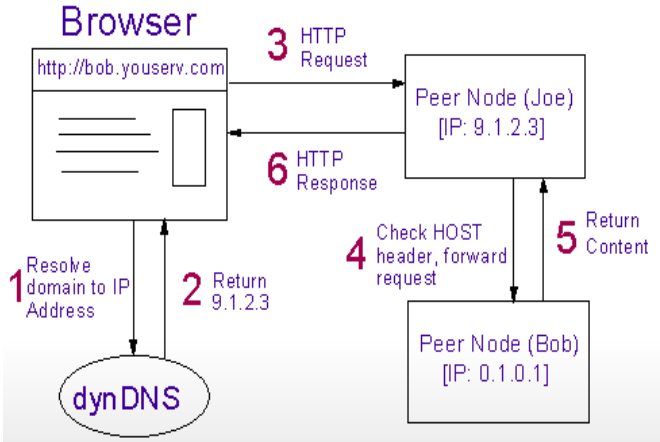

Figure 6b. Accessing content from a peer node unable to accept inbound connections.

only be spoken between peer nodes, and not by the machines requesting YouServ content.

The next figure (Fig. 6b) displays what happens when a browser attempts to access Bob's content. In this case, the browser directs the HTTP request to Joe's peer node, which performs the HTTP HOST header check and determines the request is intended for Bob's content. Joe forwards the request to Bob's machine through the previously established persistent connection (thereby not requiring it establish any inbound connections with Bob). Bob returns the requested content to Joe who returns it back to the browser through the HTTP response.

Proxying can be bandwidth intensive. We delegate the task to peer nodes instead of performing it centrally in order to spread the load across the entire network. A proxied request roughly doubles the bandwidth and latency required, and will consume the proxy's bandwidth in addition to the bandwidth of the node hosting the requested site. Though not yet implemented, it is possible to have a proxy node cache often requested content from the proxied user in order to lessen the consumed bandwidth and latency. A heavily accessed node could conceivably exploit more than one caching proxy, with multiple DNS entries used to effectively load balance across them.

Another potential proxying optimization would be to have Bob directly forward the HTTP response back to the browser instead of routing it through Joe. The problem with this idea is that the HTTP protocol requires that the HTTP response travel down the same incoming connection as the request. It is possible in some situations to have Bob spoof the IP packets to make them appear as a response from Joe (e.g. see the Triangle Boy system for anonymous and secure web access [25]). Unfortunately, any such hack involving IP spoofing would be thwarted by software or hardware that performs IP rewriting, including SOCKS proxies [22] (which are commonly used for outbound firewall traversal) and network address translators [26]. A solution to this problem would be extremely valuable, since HTTP responses typically require far more bandwidth than the requests. If proxies were required only to forward requests, then the primary objection users have to serving as a proxy -- excessive bandwidth consumption -- can be eliminated.

4.3 Maintaining Replicas

Peer nodes are themselves entirely responsible for the bulk of replica maintenance. The YouServ coordinator's part in this task is simply to provide the contact information and authenticating tokens necessary for sites to directly (or via a proxying peer node) communicate with one another. Because of the obvious security implications, YouServ requires permissions to be granted in both directions between trusted parties before the coordinator will activate a site replica. Interesting future work would be to extend the system to support use of less trusted or untrusted parties for replica hosting; content could potentially be encrypted and made tamper resistant, as in Farsite or Freenet, while maintaining compatibility with web protocols.

The site synchronization scheme we have implemented is designed with the assumption that the typical site change involves the addition or removal of files from a site, with file modifications taking place less frequently. In most cases, this scheme requires very little data to be exchanged between sites in order to keep a replica up to date. Some users in our deployment are maintaining replicas of several gigabytes and tens of thousands of files.

In this synchronization scheme, *replicator* sites (sites which host replicated content) initiate contact with their *master* sites, and also initiate content synchronization when necessary. A replicator determines when its replicated content is out of date by periodically (3 minutes by default) comparing a short summary of its replicated content with the master's summary. If these summaries fail to match, the replicator site will proceed by providing a more detailed summary to the master which allows it to determine precisely which directories need to be updated or deleted. For each directory that needs to be updated, the replicator summarizes the directory contents in order to determine precisely which files need to be updated or deleted. For each file that needs to be updated, the replicator site will download the entire file from the master site using a standard HTTP GET request.

While checking for site synchronization, replicators also effectively monitor the availability of their masters. Should any of its masters go offline, a replicator will immediately notify the YouServ coordinator. The YouServ coordinator also monitors site availability, but it must do so on a much larger scale. The replicator's assistance in this task reduces site unavailability due to situations such as improper shutdown of a YouServ site or network problems, and is consistent with our attempt to reduce the centralized roles of the YouServ system in order to minimize the cost of the service.

4.4 One Caveat: DNS caching

Some DNS servers and most browsers do not properly abide by the time-to-live (TTL) contract for caching DNS mappings. The result is that sometimes a YouServ site can become inaccessible for several minutes when a replica of the site is just activated, or the IP address of the site changes. This problem is for the most part a minor nuisance that affects a very small percentage of all accesses

to YouServ sites. An individual YouServ site that is not heavily accessed is unlikely to have its IP address cached within a browser or a local nameserver when it is accessed. Further, users aware of the problem can typically cure it by launching a new browser instance, since indiscriminate caching of DNS entries by the browser is usually the culprit.

YouServ DNS entries should be cached for at most the value of the TTL setting (2 minutes), allowing a replica to become accessible by users very shortly after it is activated by the coordinator. In a perfect world, site inaccessibility could be eliminated completely by implementing a delayed shutdown where a peer node will remain running for 2 minutes after activating a replica. Some DNS server software unfortunately allows configurations that override low TTL values with a global minimum. Most popular browsers ignore TTL values completely and use their own fixed cache timeout settings.

We have identified a handful of nameservers in IBM that appear to be configured to use no less than a 5 minute TTL. Even worse, Netscape and Mozilla browsers cache DNS entries for 15 minutes by default [16]. Internet Explorer appears to use a similar caching policy. This problem is not one unique to YouServ, but also affects systems such as dynamic DNS services. As dynamic IP address assignment and services impacted by dynamic IP address assignment become more common, we are optimistic that operating system libraries, DNS servers and their configuration, and browser implementations will adapt by properly abiding by the DNS specification.

5. SCALABILITY

The peer nodes perform the most resource intensive tasks in YouServ: proxying, storing and replicating content, and serving web content. Since the load of these tasks is distributed across a large number of peer nodes, the system bottlenecks are limited to the centralized dynamic DNS and YouServ coordinator components. In our deployment, we have one machine running the DNS component and another running the coordinator. Both machines are Pentium III-class desktop systems running at 400 MHz, with modest amounts of memory (256MBytes) and disk (16GBytes). This configuration easily supports the current user base of over 2900 with little CPU, memory, and secondary storage utilization.

Because of their nearly identical roles, the YouServ dynamic DNS component and existing dynamic DNS services on the internet have similar operating costs and scalability characteristics. Dynamic DNS services available today on the internet handle hundreds of thousands of users and charge minimal fees. For example, Dynamic DNS Network Services [8] has over 150,000 domain names registered and gets by on donations and advertising revenue alone. Its ISP costs are a mere $3500 a month, which as of today comes out to 2 cents a month per domain name. This service can be offered cheaply because DNS is a lightweight, low bandwidth protocol for which freely available implementations (including BIND) are highly optimized. DNS also supports redundant servers if needed. To reduce DNS traffic in YouServ, the coordinator could in addition be reprogrammed to recognize sites that use static IP addresses and rarely if ever fail over to replicas, and increase their TTL values accordingly.

The YouServ coordinator component spends most of its time performing user authentication and site availability monitoring. As we have noted, however, the peer nodes assist in availability monitoring, and the protocol could be extended to further push roles other than authentication to the peer nodes should scalability becomes a problem. Authentication thus becomes the primary bottleneck for the coordinator component. Each authentication requires the exchange of only a small amount of data (the encrypted user ID and password) and a single database lookup. Assuming very conservatively that our system can handle 100 authentications per second and that each YouServ site authenticates on average twice daily, the capacity of the coordinator would be over 4 million YouServ sites. Even more capacity could be provided by running additional coordinator instances on other servers.

6. SECURITY

Security is one of the primary concerns of YouServ users, and this concern has been heightened by recent worm attacks on Microsoft's IIS web server software (e.g. Code Red and its variants). In addition to worms, users are worried about hackers who might exploit holes to install unauthorized programs on their machine, or access files that were not designated for sharing. Other specific areas of concern include denial of service attacks and restricting access of certain content to designated users.

6.1 Secure Content Access

A secure access mechanism must deliver only encrypted data, and also authenticate users to sites and vice versa. Web protocols seamlessly allow browsers to authenticate websites to users and encrypt transmitted data through secure HTTP extensions and third-party issued security certificates [23]. Unfortunately, there is no widespread, uniform method for websites to authenticate their users [23]. HTTP does offer a simple mechanism for having the browser prompt the user for an id and password when requesting secured content [10]. Passwords can also be requested through HTTP forms which then set an authenticating cookie. Most websites thus implement their own user authentication scheme which typically requires the user to register a user ID and password with the site.

It would be unwieldy and unreasonable for someone to register with each YouServ site requiring secure access. The alternative is a single sign-on scheme, in which case the peer nodes cannot be responsible for performing authentication through password verification; if the sites themselves are responsible for verifying a "YouServ global" ID and password, then a malicious peer node could record the passwords presented to it, allowing it to impersonate anyone that accesses its secured content.

The solution we have implemented in YouServ is similar to that used by Microsoft Passport [17] -- a single-sign-on scheme for the web in which only a central site accepts passwords in order to perform authentication on behalf of its member sites. When someone whose identity is unknown attempts to access secured content, YouServ forces a redirect to the main YouServ authentication site, where the person attempting to access the site must provide his or her ID and password. Our IBM deployment is integrated with the IBM corporate ID and password service, allowing a YouServ site to authenticate any employee, not just YouServ users.

After successfully verifying the id and password, the authentication site will redirect the request back to the member site with the identity of the person requesting the content encrypted in the request. Encryption and decryption of identity information is performed using a symmetric key that was securely exchanged between the YouServ coordinator and the member site during initialization. A secure hash algorithm is also used to ensure this information is not tampered with.

Our implementation of secured access in YouServ is thus far incomplete, as encrypted data transmission is used only for protecting passwords, for example when providing id and password information to the YouServ authenticating server. One implication is that it is possible for a malicious hacker to "sniff" and reuse encrypted identify information (but not passwords) to temporarily impersonate any person whose request for secured content can be intercepted. YouServ sites periodically change the

identity encryption key, and identity encryption keys are site specific. The impact of such a breach is therefore limited since someone can be impersonated only on sites from which a request is intercepted, and only until the encryption key changes.

One solution to this security hole is to use SSL for secured content access, thereby protecting both identity information and the downloaded content from sniffing. SSL could also be used to additionally authenticate the server to the client. Unfortunately, use of SSL poses a few problems in the YouServ environment. First, to avoid intrusive and disturbing browser warnings and to support server authentication, each peer node would require a security certificate signed by a recognized certificate authority, which is a non-negligible expense. Virtual hosting (used when a site is served through a replica) is, in addition, not entirely compatible with the current version of SSL, which requires the proper signed certificate before the intended host is known.

Until the above problems are addressed, we warn users not to treat YouServ identity information as bulletproof. The YouServ authentication scheme is nevertheless considerably more secure than the still commonplace method of basic HTTP authentication over unencrypted connections.

6.2 Access Control

All YouServ-hosted content is publicly accessible unless it is contained either within the protected or private subfolders of the shared YouServ folder. The protected folder simply requires that any access be authenticated. Its purpose is to provide a way to determine precisely who is downloading what content. It also serves as a simple method for keeping some content out of the reach of web crawlers.

The private folder and any of its subfolders are accessible only to the site owner, unless the owner grants broader access privileges by explicitly listing the names of users and user groups who can access them. These names are listed in a file named access.txt residing within the folder to which it applies. Access control lists can also be propagated to subfolders with a special command to easily grant access to entire folder hierarchies. In YouServ, one never restricts access to public content, instead, one grants access to private content. This minimizes the potential for mistakenly publishing content to an audience broader than intended. It also minimizes problems should a replica fail to properly download an access control file.

User group support in our IBM YouServ deployment is provided by integrating with the IBM user group management server. This server allows any employee to securely associate a list of employees with a group name. Group membership queries can then be performed over an authenticated SSL connection by any application. In YouServ, the peer nodes directly query the user groups server whenever they need to determine if an authenticated user belongs to a designated group.

6.3 Malicious Attacks

YouServ is written in Java which makes it robust (if not immune) to buffer overflow attacks such as those used by Code Red and other hacking tools to install unauthorized programs. In addition, because of its content sharing focus, the YouServ webserver does not allow executing scripts (e.g. CGIs) -- another common source of security holes.

Because the webserver within each peer node is quite simple, there are only a few code paths that need to be thoroughly scrutinized in order to improve security. Our implementation provides only one code path through which all content, for whatever purpose, is accessed. This code path always explicitly verifies that any delivered content resides within the designated shared folder hierarchy and that the requesting user has the necessary access permissions.

The YouServ system as a whole is more robust to denial of service attacks than a typical web hosting service. Because of its distributed nature, a denial of service attack must target multiple machines in order to take out a significant fraction of the system's content. While it is conceivable that YouServ's DNS and coordinator components could be targeted, the system is somewhat resilient to DNS and coordinator failures. Even if DNS is unavailable, IP addresses are often cached in local nameservers. Coordinator unavailability does not prevent access to existing sites, but it does prevent access to sites requiring activation.

An individual YouServ site with no replica is generally more prone than a hosted site to denial of service attacks since end-user machines typically have rather limited bandwidth and compute power compared to those used by hosting services. A node with an available replica, however, would lose contact with its replicator during an attack, forcing the replica to become active. The attack would thus have to keep track of DNS updates and target multiple machines in order to succeed.

7. CONCLUSIONS AND FUTURE WORK

YouServ makes publishing content on the web as easy and universal as accessing it. Because YouServ exploits existing web protocols, YouServ content can be accessed with any standard web browser without installing special software. Additionally, by relying primarily on existing desktop infrastructure, the YouServ service can be provided at an extremely low cost. Our internal deployment utilizes two low-end Intel Pentium based systems. Currently handling over 2900 users, the system is projected to scale to at least tens of thousands more. The use and growing user base of our deployment lends credence to our thesis that its low cost, wide accessibility, and high availability make YouServ a superior alternative to paid hosting services and other content sharing networks for a wide class of users.

There are numerous extensions to YouServ that we feel are worth pursuing, several of which we have already mentioned: up-to-date search over YouServ content, WebDAV support, stronger secure access protocols, caching-enhanced (multi-)proxying, and an ability to safely exploit less trusted and untrusted parties for replica hosting. Another valuable extension might be a plugin API for endowing YouServ peer nodes with a dynamic content capability. Most users would have little use for full scripting support, but many might be interested in one-click installation of simple dynamic content generators for common tasks such as web counters and guest books. Such plugins could even provide generic services for sharing compute resources. In effect, we believe YouServ could evolve into a platform for "personal web services." YouServ already provides a solution for routing SOAP requests [6] to end-user machines over HTTP. Interesting open questions arising from such an architecture include how to deal gracefully with the transient availability of end-user provided web services, how to ensure validity of results when end-user machines are not necessarily trusted, and most importantly, what applications might be compelling enough to drive end users to adopt such technology?

8. ACKNOWLEDGEMENTS

We thank Andreas Dieberger and Christopher Campbell for many helpful discussions, Yirong Xu, Jan Pieper, and Daniel Meredith for their code contributions, Glenn Deen for advise on corporate guidelines, Eyitayo Akins and Jim Spohrer for help with formulating a business model, and many users of our internal YouServ deployment for their valuable feedback. We also thank Ted Anderson for his comments on this paper, and his insightful observation regarding the use of multiple caching proxies in YouServ to distribute the serving load of heavily accessed sites.

9. REFERENCES

[1] Agrawal, R.; Bayardo, R.; Gruhl, D.; and Papadimitriou, S. Vinci: A Service-Oriented Architecture for Rapid Development of Web Applications. In *Proc. of the Tenth International World Wide Web Conference (WWW10)*. http://www10.org/cdrom/papers/506/index.html, 2001.

[2] Albitz, P. and Liu, C., *DNS and BIND*, 4th edition. O'Reilly & Associates, 2001.

[3] Apache HTTP Server Project home page. http://httpd.apache.org/.

[4] Apple Computer Corp., Personal Web Sharing description page. http://www.apple.com/macosx/whatyoucando/websharing.html.

[5] Bolosky, W. J.; Douceur, J. R.; Ely, D.; and Theimer, M., Feasibility of a Serverless Distributed File System Deployed on an Existing Set of Desktop PCs. In *Proc. of the Int'l Conf. on Measurement and Modeling of Computer Systems (SIGMETRICS-2000)*, pp. 34-43, 2000.

[6] Box, D.; Ehnebuske, D.; Kakivaya, G.; Layman, A.; Mendelsohn, N.; Nielsen, H. F.; Thatte, S.; and Winer, D., *Simple Object Access Protocol*. http://www.w3.org/TR/SOAP/, May 2000.

[7] Clarke, I.; Sandberg, O.; Wiley, B.; Hong., T. W., Freenet: A Distributed Anonymous Information Storage and Retrieval System. In *Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability*, LNCS 2009, ed. by H. Federrath. Springer: New York, 2001.

[8] Dynamic DNS Network Services home page. http://www.dyndns.org/.

[9] Fielding, R.; Gettys, J.; Mogul, J.; Frystyk, H.; Mastiner, L.; Leach, P.; and Berners-Lee, T., *Hypertext Transfer Protocol -- HTTP/1.1 - Draft Standard RFC 2616*. http://www.ietf.org/rfc/rfc2616.txt, June 1999.

[10] Franks, J.; Hallam-Baker, P.; Hostetler, J.; Lawrence, S.; Leach, P.; Luotonen, A.; and Stewart, L., *HTTP Authentication: Basic and Digest Access Authentication - Draft Standard RFC 2617*. http://www1.ics.uci.edu/pub/ietf/http/rfc2617.txt, June 1999.

[11] Clip2 Distributed Search Services, *The Gnutella Protocol Specification v0.4*. http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf.

[12] Groove Networks, Inc., *Groove Product Backgrounder.* Corporate whitepaper, 2001.

[13] ICQ home page. http://web.icq.com.

[14] Lotus Notes. http://www.notes.net/.

[15] Lotus QuickPlace. http://www.lotus.com/home.nsf/welcome/quickplace.

[16] Mozilla.org, netlib documentation. http://www.mozilla.org/docs/netlib/ (netlib preferences section).

[17] Microsoft Corp. *Microsoft Passport Technical Whitepaper*, http://www.passport.com/downloads/TechnicalWhitePaper.doc, March 2001.

[18] Mockapetris, P., *Domain Names - Implementation and Specification -- STD 13, RCF 1035*. http://www.ietf.org/rfc/rfc1035.txt, November 1987.

[19] Mojo Nation, Mojo Nation Technical Overview. Feb. 14, 2000.

[20] Napster home page. http://www.napster.com/.

[21] Open Source Napster Server project. http://opennap.sourceforge.net/.

[22] Permio Technologies, Inc., SOCKS home page. http://www.socks.nec.com/.

[23] Rescorla, E., *HTTP Over TLS*, IETF Internet Draft. http://www.ietf.org/rfc/rfc2818.txt, Sept. 1999.

[24] Rose, M. *The Blocks Extensible Exchange Protocol Core*. IETF Network Working Group request for comments: 3080. http://www.ietf.org/rfc/rfc3080.txt, March 2001.

[25] SafeWeb, Inc. *Triangle Boy Network* (Technical Whitepaper). http://fugu.safeweb.com/sjws/solutions/white_papers_triangle_boy.html, 2001.

[26] Senie, D., *NAT-Friendly Application Design Guidelines*. IETF Network Working Group RFC: 3235. http://www.faqs.org/rfcs/rfc3235.html, Jan. 2002.

[27] Sun Microsystems, Inc., *Project JXTA: An Open, Innovative Collaboration*. http://www.jxta.org/project/www/docs/OpenInnovative.pdf, April 25, 2001.

[28] Whitehead, J. and Goland, Y. Y., WebDAV: A Network Protocol for Remote Collaborative Authoring on the Web. In *Proc. of the European Computer Supported Cooperative Work Conference (ECSCW'99)*, 1999. Available at http://www.webdav.org.

[29] Working Resources, Inc., BadBlue website and product description. http://www.badblue.com, 2001.

[30] XDegrees, Inc., *Introducing XDegrees Technology.* Corporate whitepaper, 2001.

[31] Yang, B. and Garcia-Molina, H., Comparing Hybrid Peer-to-Peer Systems. In *Proc. of the 27th Int'l Conf. on Very Large Data Bases*, 561-570, 2001.