Lotus Developer Domain
The Technical Resource for Lotus software
LDD Today

The hitchhiker's guide to Sametime deployment at IBM

Level: Intermediate
Works with: Sametime
Updated: 01-Aug-2002

by
Matt
Broomhall

Instant messaging inside of IBM started with a product called VPBuddy, the predecessor of Sametime Connect. Originally deployed as a proof-of-concept pilot by one of our advanced technology labs for a small audience (a couple thousand employees), the pilot quickly reached 65,000 registered users—one of the largest pilots on record inside of IBM. The success of this program was the only proof needed to justify a broader enterprise deployment of instant messaging technology. But along with the success came growing concerns; the pilot would soon become unmanageable as an advanced technology pilot because of scalability issues and lack of formal support as a standard offering for large groups of end-users. This placed urgency on planning for an enterprise production deployment, which we launched in February of 2000, while a formal pilot using Sametime Connect was launched at the same time.

Instant messaging is perhaps the ultimate paradise for the hitchhiker personality. While it is generally true that one person can send only one piece of email at a time, and maybe have a phone conversation (only one) at the same time, those limitations do not apply to those who have discovered instant messaging. With an instant messaging client, this same person can have a virtually *unlimited* number of *simultaneous* conversations (although 42 is reportedly an ideal number). With this level of communication "freedom of choice," the popularity of instant messaging with the hitchhiker personality is not surprising.

We are often asked about how quickly IBM adopted Sametime and how we deployed Sametime to such a large company. The former answer is easy, while the latter is not so simple. This article describes the answer to the latter question—how IBM deployed and adopted Sametime—as well as where we are going next with Sametime inside of IBM. This article is intended for system architects and administrators familiar with instant messaging technologies who are considering or planning a Sametime deployment in their organization. The IBM experience and insight into providing a reliable and scalable instant messaging infrastructure can serve as a guide to other teams planning a Sametime deployment.

The highway to production

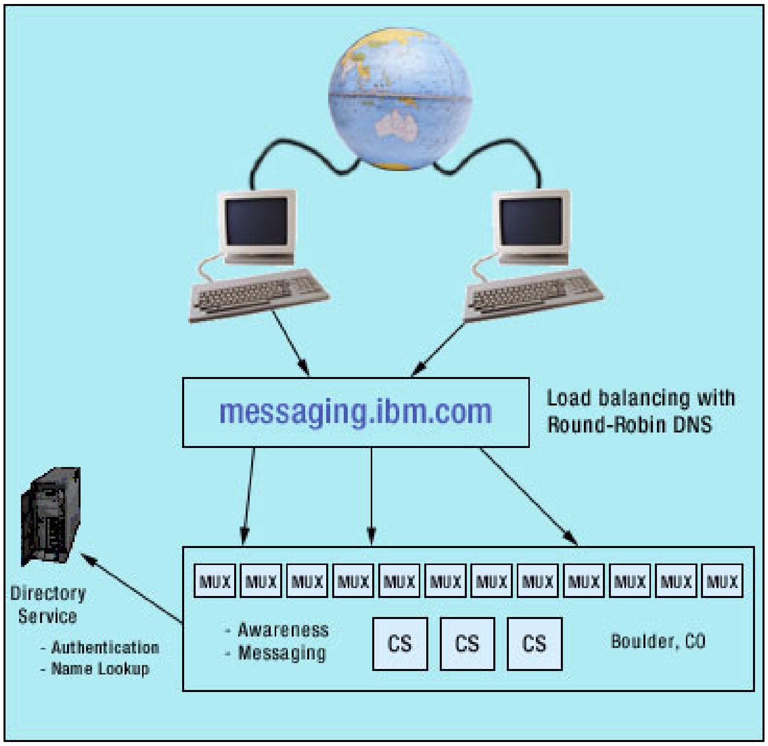
Our original team mission was simple: to provide instant messaging to all IBM employees worldwide while at the same time strengthening the product family by providing "enterprise" experience to the product development teams. The premise of the pilot was to validate the ability to scale Sametime Connect to a population of 330,000 people—the size of the IBM user population at the time. To accomplish this, we first conducted an analysis of how the product behaved under load, with a focus on data flow within the product to optimize the deployment of the individual software components. The result of the analysis highlighted four major components within the software that had very different operational profiles:

- Multiplexor (MUX) Service manages end-user connections
- Community Services (CS) manage presence and messaging traffic
- Meeting Services (MS, IMS) manage shared objects like desktop, applications, and the Whiteboard

- Authentication manages identification and credentials validation

MUX, Community Services, and Meeting Services servers are machines dedicated to their respective service.

At IBM, when users attempt to access the Sametime environment, they are directed to one of twelve MUX servers. The MUX server passes the end-user connection to one of the three available Community Services servers. Before users are added to the community, their user credentials are authenticated against their records in the corporate directory. After the directory validates the credentials, users are granted access to join the community where they become aware of the presence of other members and exchange instant messages with others online. The following figure shows our Sametime 1.55 infrastructure.



Taking the behavior of each of these components into account provided the framework for designing an infrastructure that was both scalable and stable. To implement this framework, we deployed each component to a separate physical infrastructure to properly “right-size" each piece.

In partnership with the Sametime product development team, we chose an initial architecture that separated the MUX and Community Services components and that used a central corporate directory for authentication. Because seamless separation of Meeting Services and Community Services was not possible with the 1.55 release we had at the time, we deferred deployment of the Instant Meeting Service and instead, focused on the components our user community depends upon.

Our focus, therefore, shifted to providing a reliable Instant Messaging Service. To accomplish that, we moved the Meeting Services components to a separate infrastructure. Unfortunately, Release 1.55 did not offer practical means to accomplish this; so we deferred the deployment of instant meetings until a mechanism to separate the services and to operate the two services as a single, integrated environment became available.

Is the road big enough?

The skilled hitchhiker must choose the roads he hitchhikes on with care. The roads needs enough traffic to ensure that he can find a ride. Properly sizing the Sametime components presents a similar concern. The MUX Service performs a high bandwidth connection management process separate from the lower bandwidth presence and messaging components performed by Community Services. Using a single PII 400mhz CPU server with 1GB RAM and 9GB of hard drive, each MUX server had a theoretical connection management upper limit of between 20,000 and 30,000 end-user connections.

In practice, we found that under heavy load at maximum authentication throughput, our environment became unstable beyond 20,000 connections. Throttling the maximum connections at 20,000 resulted in improved stability. However, large spikes in load resulted in a return of instability to the environment. Moreover, when one or more MUX servers dropped large numbers of connections, the thousands of reconnection requests entering the authentication queue at the same time exacerbated the overall instability.

We used Tivoli to monitor memory, CPU, and disk utilization and network sniffer tools to measure traffic. Analysis showed memory utilization as a contributor to this instability. To lower memory utilization (based on the usage patterns within IBM), we lowered maximum connections to 15,000 by modifying one of the INI files on the MUX servers. Further testing indicated that this was an optimal configuration, one that allowed each server to run under load with excellent stability between weekly maintenance cycles.

We used similar logic to right-size the low bandwidth Community Services as well. However, our analysis showed higher CPU utilization requirements of the Community Services servers than with the MUX servers. With few Community Services servers managing the presence data and messaging traffic for peak loads of greater than 100,000 end users, we added a second CPU to each Community Services server to account for spikes in server load.

Beyond this, there were few issues to resolve. In fact, the only significant issue was determining how many MUX servers was optimal for each Community Services server. Accounting for redundant MUX capacity, three Community Services servers were chosen, resulting in a four MUX servers to one Community Services server ratio. This meant that a failure of a single Community Services server or MUX server left ample capacity to continue to service the entire online population. We arrived at this ratio by testing many configurations under full load to determine overall performance and to measure the effects of component failure.

Going my way? Access to the environment

The twelve MUX servers receive their end-user connection requests through rotating or Round-Robin DNS, which is a domain name server configured to pass incoming traffic to a range of IP addresses sequentially. Each user accesses the environment through a single front-end DNS address; in IBM’s case, the address is messaging.ibm.com. Although management of the DNS resource requires manual intervention in the event of a failing MUX server, alerting mechanisms deployed to monitor the environment ensure that any required intervention (taking a failing MUX out of pool) was quick. For example, we configured one of our probes to connect to a specific MUX server in a specific time interval. Two consecutive failures trigger an alert to the operations team, who in turn validate the health of the server.

The authentication mechanism chosen is based on LDAP (lightweight directory access protocol). A central credentials repository, known as Enterprise Directory, validates a user ID and password combination for permission to use Sametime. This strategy allowed us to avoid deploying an additional directory to support this new environment, while at the same time providing the capability to use the same user ID and password combination as most other standard corporate applications.

There were potholes in the road

Because this was our first large-scale experience with technology of this type inside of IBM, the larger expanded pilot highlighted deficiencies in our infrastructure. In fact, the real-time nature of the software showed us each and every place within our infrastructure that was not real-time*.* Our early experience highlighted two major issues:

- While our Enterprise Directory operated as expected for the vast majority of Web-based applications, it was not interacting as expected with Sametime (clearing persistent connections and experiencing soft hangs).
- Several areas within our network caused Sametime to behave unpredictably.

Our Enterprise Directory deployment is made up of several IBM Secureway LDAP servers. These servers have IBM Edge Network Dispatcher servers on the front-end to manage load balancing and failover. In the initial problem determination, we found the Network Dispatcher servers cleared the user connections between the Community Services servers and the Enterprise Directory. This resulted in an unstable environment from the end-user perspective.

Analysis showed this to be the expected behavior based on a Network Dispatcher configuration optimized for transient connections, such as for Web-based applications. Once this was accounted for, the end-user experience improved because their connections to the Sametime environment were no longer being dropped. There was also some anomalous behavior with the directory servers that we termed “soft hang conditions.” These hangs, while very short in duration, were long enough to impact connection status for Sametime users. With the help of the Secureway development team, these minor software problems were identified, fixed, and applied to the Enterprise Directory infrastructure with excellent results.

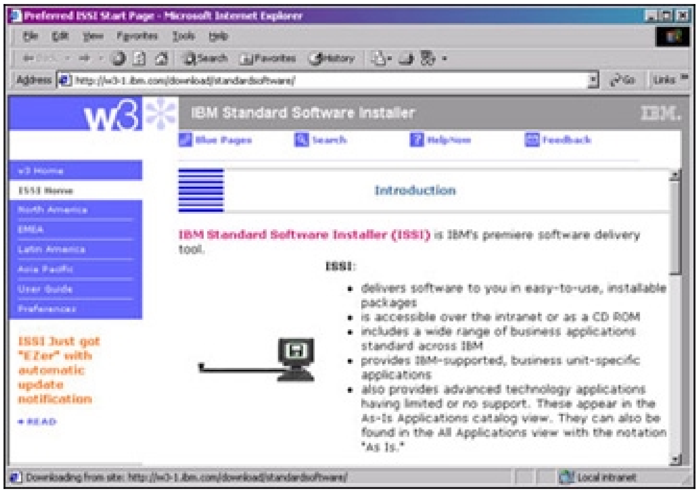
We also found that latency in the delivery center campus LAN introduced stale connection status for end-user connections to the MUX servers, causing drops similar in nature to the cleared connections. A move to a 100 MB switched ethernet for both Enterprise Directory and Sametime, as well as plugging these two infrastructures into the same LAN segment, resolved many of the remaining problems.

We found several minor problems, such as inconsistent MTU size settings and poorly performing routers in the WAN, and addressed them during our deployment ramp-up. The remaining problems centered largely with local campus LANs. In all three cases, we addressed these issues as we found them.

When taken together, resolving these problems resulted in an extremely reliable infrastructure to the point that our user population has become very dependent upon Sametime.

Other factors on the road to deployment
There were other factors that helped to ensure our success:

- Including the help desk as part of our original team ensured readiness of the help desk personnel, while the reuse of existing deployment tools, such as the IBM Standard Software Installer (ISSI), eased the installation and configuration and prevented many calls to the help desk from ever taking place. (ISSI is a one-stop shopping resource for all business software inside of IBM. This resource not only contains all appropriate software but also manages the installation of the selected software, including checking dependencies for each package.)


Preferred ISSI Start Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Search Favorites History
Address http://w3-1.ibm.com/download/standardsoftware/
Go Links
w3
IBM Standard Software Installer
IBM.
Blue Pages
Search
Help/Now
Feedback
v3 Home
ISSI Home
North America
EMEA
Latin America
Asia Pacific
User Guide
Preferences
ISSI Just got "EZer" with automatic update notification
» READ
Downloading from site: http://w3-1.ibm.com/download/standardsoftware/
Local intranet

Introduction

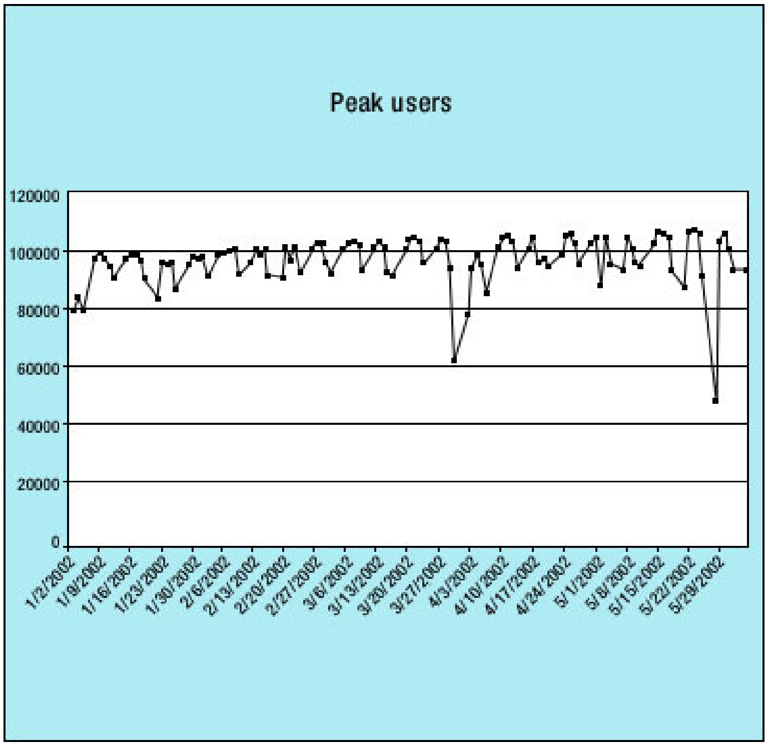
IBM Standard Software Installer (ISSI) is IBM's premiere software delivery tool.

ISSI:

- delivers software to you in easy-to-use, installable packages
- is accessible over the intranet or as a CD ROM
- includes a wide range of business applications standard across IBM
- provides IBM-supported, business unit-specific applications
- also provides advanced technology applications having limited or no support. These appear in the As-Is Applications catalog view. They can also be found in the All Applications view with the notation "As Is."

- Use of existing communications vehicles, such as mailings from the CIO's office and our project Web site, also informed the end-user community of what to expect and how and where to get help on those rare occasions when it was needed.

Success
Today, our current infrastructure, deployed in our Boulder, Colorado Service Delivery Site, experiences remarkable success. We support more than 100,000 peak concurrent users each day, representing 225,000 unique users sending 2.5 million messages daily within an enabled population (IBM employees and contractors) of more than 330,000 people spread between more than 2,000 unique locations worldwide. The adoption by IBM employees and the excellent availability of the service means our original mission is essentially complete. However, because each team enjoys significant benefits, we maintain our close working relationship with product development.



The future of Sametime at IBM

The seasoned hitchhiker constantly seeks a better ride while maintaining the freedom of choice. The improved ride does not imply a fancier or more expensive car, merely one that is more reliable and has more leg and head room.

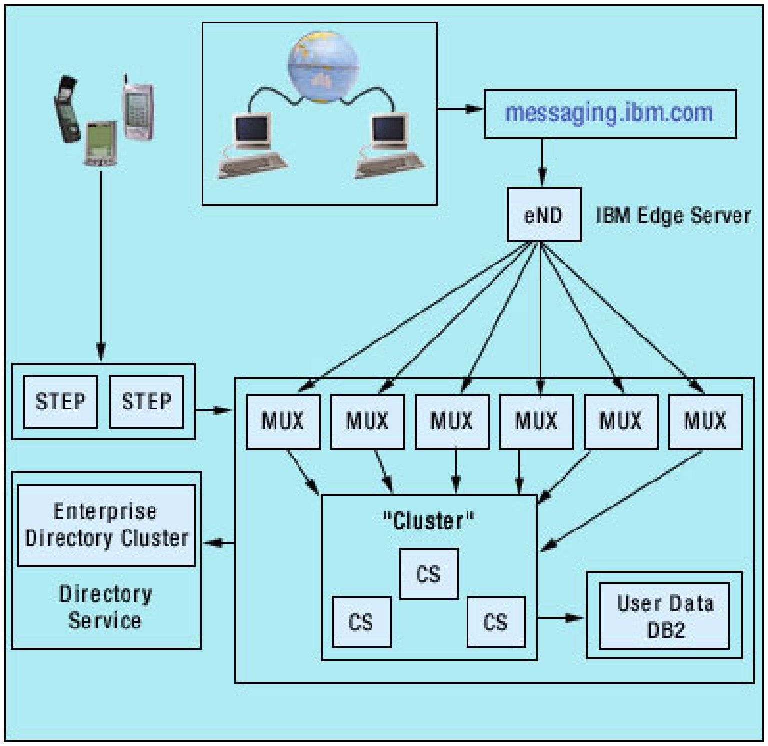
We have several goals for the next version of our instant messaging infrastructure. Most important of these is the requirement to eliminate single points of failure. To accomplish that, we are extending our deployment beyond our Boulder location to reduce the traffic and dependency on the WAN segments leading to our current single site and basing our choices for the location on a couple of goals:

- To maximize the user experience for all users worldwide
- To more effectively manage bandwidth utilization

We plan for additional installations to be nearer to large IBM population centers. We chose our Southbury, Connecticut and Portsmouth, UK delivery centers as viable additional locations.

To further enhance the user experience, we will leverage the new features in Sametime 3.0 to deploy a highly available infrastructure. While we have been very successful with our current deployment, we could not achieve the goal of a highly available infrastructure using the current 1.55 release as the features required to accomplish this are not available in that release.

Release 3.0 addresses the requirements for a highly available enterprise deployment in two very important areas: load balancing and failover. Our target Release 3.0 deployment includes some additional IBM technologies, such as IBM Edge Server Network Dispatcher and IBM DB2. The following illustration shows how we envision the new deployment will look:



The use of DB2 makes possible higher performance server-side storage and maintenance of user information, like privacy and buddy lists. For an enterprise as large as IBM, this solves a significant manageability issue. Off-loading server-side storage from the Community Services server to DB2 removes another function from the core Community Services server, so maintenance of user information does not impact the core services.

Sametime 3.0 includes failover and load balancing between the MUX and Community Services servers. Service remains intact if a Community Services server becomes disabled; we cannot overemphasize how important this capability is when there are more than 300,000 people depending on the service being available. We also plan to deploy Edge Server Network Dispatcher as the front-end to the MUX server pool, replacing the Round-Robin DNS mechanism currently in place. In addition to eliminating the manual intervention required with Round-Robin DNS, Network Dispatcher along with Network Dispatcher health agents installed on the MUX servers (small footprint Java applications) provides load balancing/failover between the Network Dispatcher servers and the MUX server pool. Again, the importance from not only a user perspective but also a serviceability perspective, cannot be overstated.

The illustration also shows the addition of Sametime Everyplace to the infrastructure for pervasive computing device access to the Sametime services. Also included in our plan is the addition of four Instant Meeting servers so that we can add small-group meetings capability.

Like the experiences of a skilled hitchhiker, the IBM experience with Sametime can serve as the proof-point for any company or organization (large or small) wanting to realize the many benefits of enhanced communications that Sametime has to offer. Our experiences and lessons-learned prove that a deployment as large as ours is not only technically possible, but also offers an easily managed, highly available service. And while 42 may not in fact be the optimal number of open Sametime windows on your desktop, Sametime offers the opportunity for you to find the number that is.

ABOUT THE AUTHOR

Matt Broomhall started working at IBM back when Notes R3.0 first shipped. He is now part of the Business Transformation & CIO (BT/CIO) office and is responsible for deploying software tools that allow people and teams to more effectively collaborate. Matt's team deployed Sametime Connect in IBM and is now planning the upgrade to Sametime 3.0. Matt skis in the winter and runs when there is no snow. Matt's fondness for traveling to Disney World with his family as well as his home theater plays prominently with his other leisure preferences.