Lotus. Developer Domain
The Technical Resource for Lotus software
LDD Today

Level: Intermediate
Works with: Sametime
Updated: 01-Apr-2003

by
Matt
Broomhall

For most IBMers, Sametime has been synonymous with the Sametime 1.55 client. IBM used the release 1.55 client and server combination for more than two years. It worked very well and provided end users with the “freedom of the open road”—the ability to find and chat with anybody in our company at any time. In fact, most employees were not even aware that Sametime 3 even existed—they remained dutiful users of the “old model.” The new model, now appearing on showroom floors however, is shinier and faster. It handles better and is more reliable—it's the “Sport Utility” of the instant messaging world.

Our older model, while providing plenty of smiles-per-mile, was more like the sports car of the mid-1950s: lots of fun, but make sure you bring your tools with you on the journey. Our quality of service was excellent; however, it took some work to stay there. With Sametime 3 deployed in IBM since November 16, 2002, the fast lane has become much more inviting.

While this article does not in fact assume great knowledge of the auto industry, it does assume that you have knowledge of or experience with the administration of Lotus Sametime. This article uses the automobile metaphor to describe the technical background of our upgrade to Sametime 3. For background on IBM's Sametime deployment, see the *LDD Today* article, "**The hitchhiker's guide to Sametime deployment at IBM**."

Trading in and trading up

The transition to Sametime 3 was more than a simple engine transplant—it was a wholesale trade-in for the new model right as it hit the showroom. What this means from a practical standpoint is that our transition to release 3 was an excellent opportunity to switch to all new infrastructure using the new release and not just an upgrade to existing servers. This gave us an opportunity to test and add new features and capabilities as well as to do a few things differently to provide improved service—improvements especially important to larger companies such as IBM.

These improvements stemmed from a couple of key areas: our experience with Sametime 1.55, as well as through collaboration with product development to address weaknesses we uncovered through our previous deployment. This collaboration involved using IBM to “field test” the software under enterprise conditions to insure reliability and performance objectives were achieved. We achieved desired gains in authentication rate and performance, in particular with buddy list loading times as well as achieving phenomenal reliability. Lotus Product Introduction Engineering (PIE) validated these gains by comparing Sametime 1.55 performance with Sametime 3.

For IBM, the enterprise model is a completely modular deployment, that is, each primary software component—Multiplexer and Community Servers, Edge Server Network Dispatcher, and LDAP-based enterprise directory—is deployed on a separate infrastructure as a means to scale and enhance reliability. More horsepower with the MUX (multiplexor) server software has allowed us to increase the maximum connection volume from 15,000 to 30,000 connections per MUX server. Before the upgrade, the previous release required that each MUX be bound directly to an individual Community Server. The MUX is also now capable of connecting to all Community Servers in the pool, effectively minimizing the impacts of a failed Community Server.

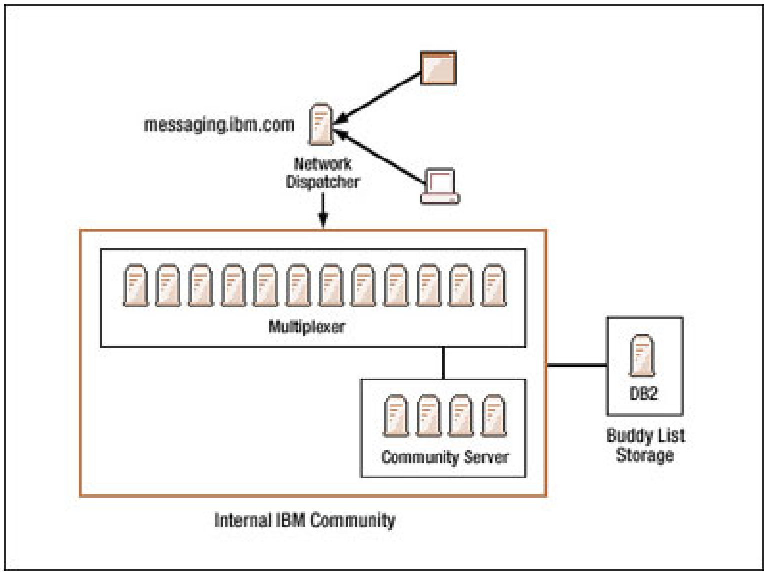
Why do we still have 12 MUX servers if the upper limit for connections is so improved? We still have 12 for three primary reasons:

- To insure adequate capacity remains in place should more than one MUX server fail
- To provide us adequate capacity to give us the capability to sequentially upgrade parts of the infrastructure with little risk for loss of service
- To accommodate a continued steady increase in overall volume due to higher usage and exploiting applications

Again, this is also the same reason why we have four Community Servers in the infrastructure – to minimize impacts of potential failure. In addition, the community servers can now operate as a “cluster.” From an end-user perspective the four Community Servers become a single logical server, negating the need for an end user to be tied to a discrete server—further reducing perceived loss of service when a single component fails. What this means is that an end user can disconnect and log back in ten minutes, ten hours or ten days later, and it does not matter in the least which MUX and Community Server combination the connection is made with—an achievement essential to how reliant IBM employees are on Sametime.

In keeping with this strategy, with our new Sametime 3 deployment, we also manage buddy list storage on a separate infrastructure. This allows us to not only off-load this function to limit the impact to the Community Servers where the buddy list is normally stored, but also to position us to use a solution capable of very high performance for very large numbers of people—like IBM. The solution chosen for this was IBM DB2. Off-loading this job essentially provides the “engine” the ability, in our case the Community Server, to carry much less cargo. While this capability was done by product development originally for IBM, the goal all along was to provide this same capability to our enterprise customers. Essentially, this functionality works by replacing the logic that directs the storage to an NSF file, and instead, the replacement software adds capability to redirect the storage to DB2 servers.

This then allowed the Community Servers to focus completely on their assigned task: managing the presence data and passing messages for our very large community. And, because the Community Servers act as a single logical server, as the following diagram shows, there was no need to deploy an instance of DB2 for each Community Server, they all share the single deployed instance servicing *all* 330,000 IBM employees.



Life in the fast lane

Although we thoroughly tested the environment before switching our user community to it, in the weeks since the Sametime 3 deployment we have “live” stress tested the environment, reaching peak concurrent user loads of nearly 120,000 users. We have achieved ”life in the fast lane.”

Just like the wind tunnels and proving grounds the car companies use, we also did in fact very thoroughly test the software before we let our population use it. Our testing started with the Lotus Product Introduction Engineering

(PIE) group. This team constructed an environment that was a mirror of the IBM infrastructure design and performed a wide range of functional tests as the starting point.

The next step was to drive an artificial end user workload against the environment to measure overall stability and performance as a means to prepare for the third step. The end user load is generated by “robots” that are designed to simulate user logon/logoff, message exchange, buddy list updates, and status change with the goal of approximating the load the production system will face. In this next step, the proving grounds begin to simulate more closely real world conditions; in fact, the load is applied across the wide-area network to the systems at the service delivery center that would actually become the production environment. The remote stress test is the key difference that makes this last road test more representative of real world conditions. (See the Performance Perspectives column, "**Sametime 3 vs 1.5.5 performance comparison**."

Using the Network Dispatcher Advisors
Like most car owners, we are not content to simply leave well enough alone; tuning and improving our investment are continuous goals. Our team continues to make our environment even more “roadworthy,” while at the same time adding new capabilities. To make our environment more roadworthy, we are preparing to use WebSphere Edge Server Network Dispatcher on the front-end of our instant messaging environment. This isn't simply the stock version of the product—we are developing (and will have tested by the time you read this) Network Dispatcher Advisors. These advisors are small Java programs that provide the Network Dispatcher critical information regarding the relative health of each of the MUX servers. Why is this important? The advisors not only help the Network Dispatcher determine the best MUX servers to pass connections to, but they also help decide *if* a connection should go to a particular MUX server at all. This prevents end user connections from being passed to a failing or failed MUX server, providing the user a smoother ride.

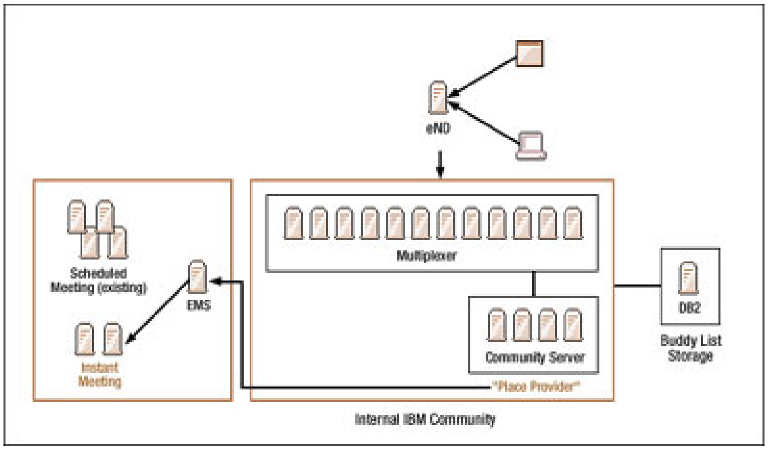
The Edge Servers act like a fuel pump, delivering just the right amount of fuel to the engine. In our case, the fuel is actually end user connections. When an end user connects to our environment, he does not connect to a single discrete server. Messaging.ibm.com is a high-level DNS entry for our entire environment. In our case, messaging.ibm.com maps to the IP address for a primary ND (Edge Server Network Dispatcher).

This Network Dispatcher performs two major functions. First, it determines whether the MUX server is up and functional. Second, the ND measures the response time of each MUX and compares the response times. This insures the connection is never passed to an unresponsive MUX and instead is delivered to the most available MUX in the MUX server pool. In this way, the environment not only enjoys redundant capacity, but is also highly available due to the failover and load balancing introduced through the use of Edge Server Network Dispatcher.

For features new to IBM, testing is under way to determine how best to scale and deploy instant meeting capability, the new IM Gateway and Sametime Everyplace.

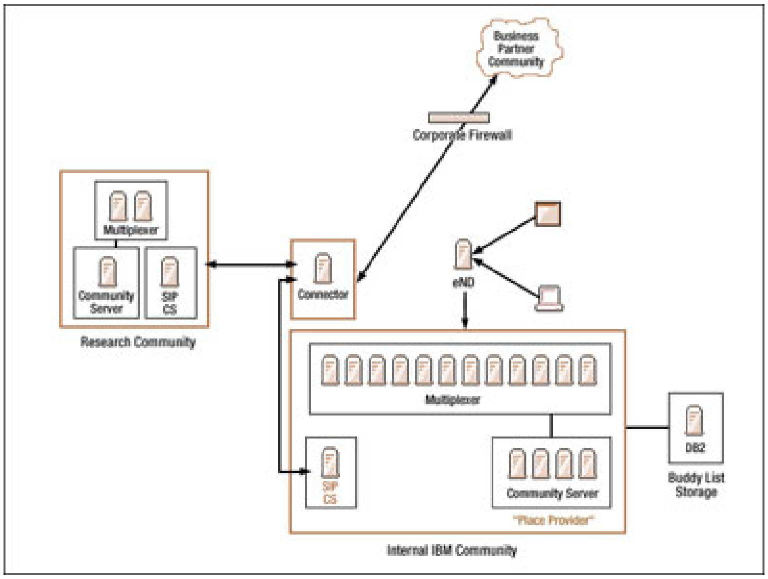
Instant meetings and the IM Gateway @ IBM

In the case of Instant Meetings, we load new services on our Community Servers called “place providers.” Place providers are services that route end user connections to the proper location for each “activity provider.” An activity, in this case, can be instant messaging, instant meetings, or scheduled meetings. IBM’s Community Servers are configured to support only one activity, that being the instant messaging role. However, place providers are configured to pass other types of connection requests on to the new Sametime 3 Enterprise Meeting Server (EMS). The EMS acts as a front-end proxy for all meeting services. This means that an instant meeting request initiated through the Sametime Connect 3 client is routed to the EMS, which in turn locates an activity provider configured to supply instant meeting services with the same scenario in place for scheduling larger meetings—our sport utility carries people and luggage, while the eMeetings infrastructure includes the trucks and trains that carry the heavier meeting loads.



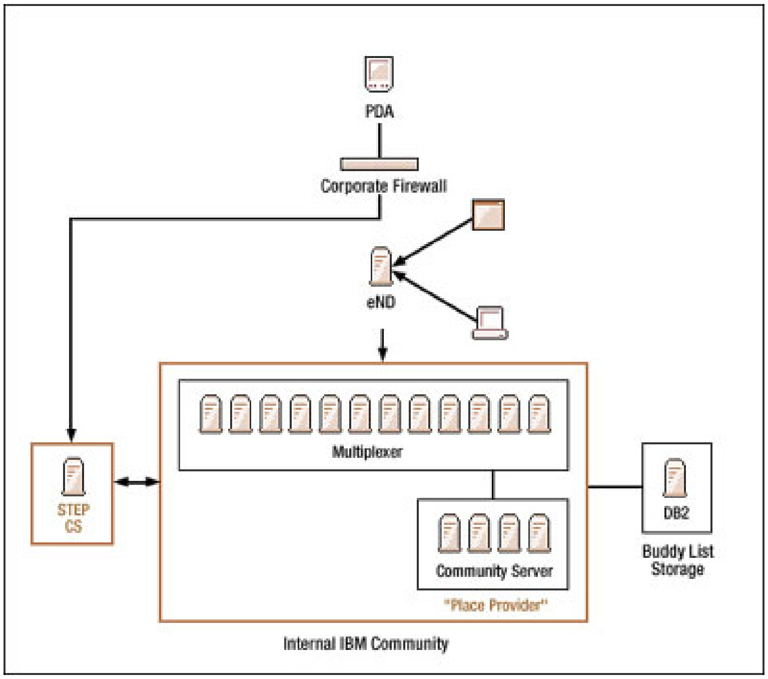
Crossing the IM Gateway

The IM Gateway is increasingly viewed as an important means to keep in close contact with our business partners—"real time." In essence, the IM Gateway becomes a secure proxy or private highway between corporate instant messaging infrastructures. With Sametime 3, we have the capability to deploy a "custom" Community Server that will also run the gateway process. This process sends and receives messages through a "connector server," the component that acts as the intermediary between the instant messaging environments of two companies. The Community Services component of this custom Community Server exchanges the messages and presence data with the rest of the internal IBM instant messaging environment. We are currently preparing this component for stress testing to insure we deploy with sufficient capacity to meet the internal demand. Our functional testing and pilot will be accomplished by bridging the existing IBM Sametime environment to the new IBM Research Community using the gateway, while the PIE team will again drive testing.



Sametime Everyplace @ IBM

Sametime Everyplace (STEP) is a tool that allows your pervasive computing device to access Sametime. A device registered to the owner uses the STEP server as a proxy to access standard infrastructure, allowing the user to exchange messages with users of the standard client. Using a specialized Community Server and Domino Everyplace SMS, short message service (SMS) messages are passed from Sametime Everyplace to the SMS carrier the pervasive device is using, while incoming messages are transformed from wireless access protocol (WAP) to TCP/IP-HTTP. This becomes very useful when the end user is literally on the open road and must maintain communication with the team back in the office. This tool is currently deployed as an extended pilot while we complete the stress testing.



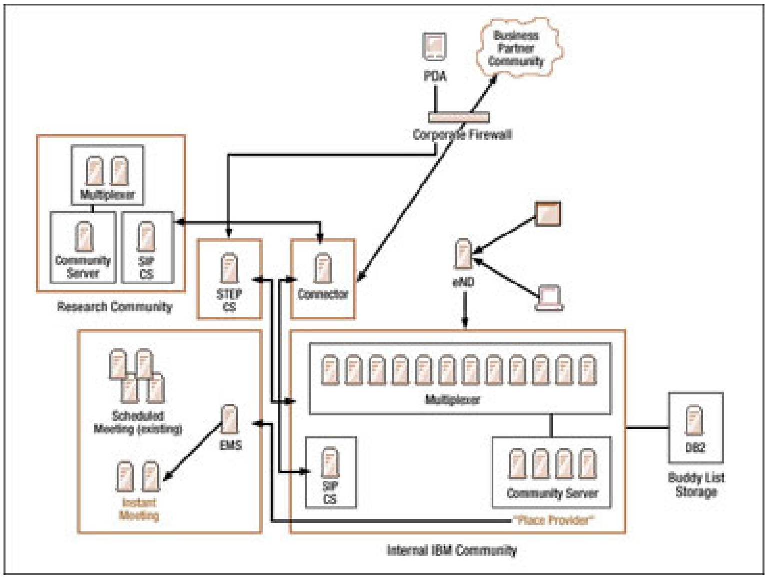
Testing this environment was complex and involved the following steps:

- Validate that the STEP server was functioning properly with the other required component, Domino Everyplace SMS.
- Verify that this custom STEP Community Server was able to properly pass messaging and awareness data between itself and a standard-build Community Server.
- Validate that messages could be routed outbound to a registered PDA.
- Insure incoming messages were properly transformed via the WebSphere WAP Gateway and passed through this gateway to the internal infrastructure and received by a standard client.

Stress testing involves simulating PDA load through this infrastructure.

Conclusion

Our experience getting Sametime 3 up and running in IBM has been extremely positive; it is one *very* smooth ride. Because our experiences have been so positive, we are already taking a holistic view of our entire environment in terms of how we can extract more value out of the base environment. In addition to the previously described infrastructure components that we are piloting and are soon to deploy, we are in the process of determining how best to leverage the application integration options so that applications of all types can utilize the capabilities Sametime offer. Although that is a road not yet traveled, it soon will be as we plug the remaining pieces into our Sametime 3 environment.



Over 225,000 IBMers use our Sametime instant messaging infrastructure each day to send 3,000,000 messages. The reason we do this is very clear: It helps us do our jobs more easily and efficiently. That's all for now—the road beckons.

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.