Predicting Domino cluster performance

by
Masud
Khandker

Level: Advanced
Works with: All
Updated 12/01/99

Inside this article:

Related links:

Domino clusters (Part 1)

Domino clusters (Part 2)

Workload balancing with Domino clusters

Fine points of configuring a cluster

NotesBench Web site

Get the PDF:

One of the main advantages of Domino clusters is that they provide higher database availability than non-clustered systems. Domino uses application-level clustering, where copies of databases on one server are replicated on one or more different servers. However, high availability comes at a cost to performance overhead. In order to maintain database consistency among cluster servers, Domino needs to replicate database changes as they occur to all replica copies of the database in the cluster. The extent of the performance overhead involved depends on how the servers are configured and how frequently the databases are updated. For example, a Domino server hosting database replicas can be configured either as a standalone server for handling cluster failovers, or as an active server handling mail user requests. Databases can be replicated once, or more than once, depending upon how critical their availability is. All these variations have performance implications.

Therefore, organizations planning to deploy Domino clusters need to consider several factors, such as the number of users that can be supported by a given system configuration, or the additional resources (CPU, disk space, memory) that are required to support clustering. Ordinarily, system administrators rely on past experience, rules of thumb, and often, just plain trial-and-error to find optimal clustered configurations for their users. This article offers an alternative to trial and error, in the form of a predictive model that provides administrators with key information about optimal cluster configuration and resource planning.

While this sounds daunting, building and using the model is not as complicated as it sounds. The article will guide you through the major steps. These include:

- Testing a system under workload.
- Explaining the basic measurements of the test machines and show how to build the model.
- Using the model to predict the performance of various configurations under varying load.
- Using the model to predict the optimal number of users for a given set of configurations.
- Determining the number of users that can actually be supported with those configurations.
- Comparing the number predicted by the model with the actual number to validate the model.

Note: Although the model was developed using Notes/Domino release 4.6.x, it can be applied to Domino R5 as well.

Explaining the workload

Workload has always been a critical issue in capacity planning. In this section, we explain the workload used in our test.

Typically, administrators ask what type of workload should be used in testing. The process by which measurements are obtained depends on the state of the Domino cluster itself. In many cases, clustering is implemented after the Domino server is deployed. For such cases, we can take the initial set of measurements using the existing server workload.

In other cases, where the Domino server has not been deployed, but the hardware has been selected, we obtain the initial measurements using an artificially-generated workload. Because users in real environments could be radically different from those in an artificially-generated workload, concessions should be made to convert the number of model-predicted users to real-life users when a synthetic workload is used.

Still, in other cases, where hardware needs to be selected, we can use NotesMark, produced by NotesBench, to find suitable machines for non-clustered Domino servers and use them as a starting point for deriving workload measurements.

For the purposes of this article, the test workload is generated using Testnsf, a version of NotesBench, which allows us to fine-tune the workload. The workload is modeled after an IBM production site (known as IBM Geoplex site). It includes three levels of users -- light, medium, and heavy. All users behave like typical mail users and perform basic mail operations, such as sending mail, navigating through Notes mail databases, refiling and deleting mail documents, and checking for new mail messages. The majority of the operations performed by all three levels of users involve frequent database updates.

The main differences among the different levels of users are the frequency and the size of mail messages. For example, medium-level users send e-mail messages more frequently than light users do. In addition, medium users tend to make group responses. Heavy users tend to use the Name and Address book to find addresses, and they send larger e-mail messages on a more frequent basis.

Using a multiple-intensity workload makes it easier to obtain model parameters closer to a real-environment workload, by combining them in different proportions. The following table compares the intensity of workload generated by the three levels of users.

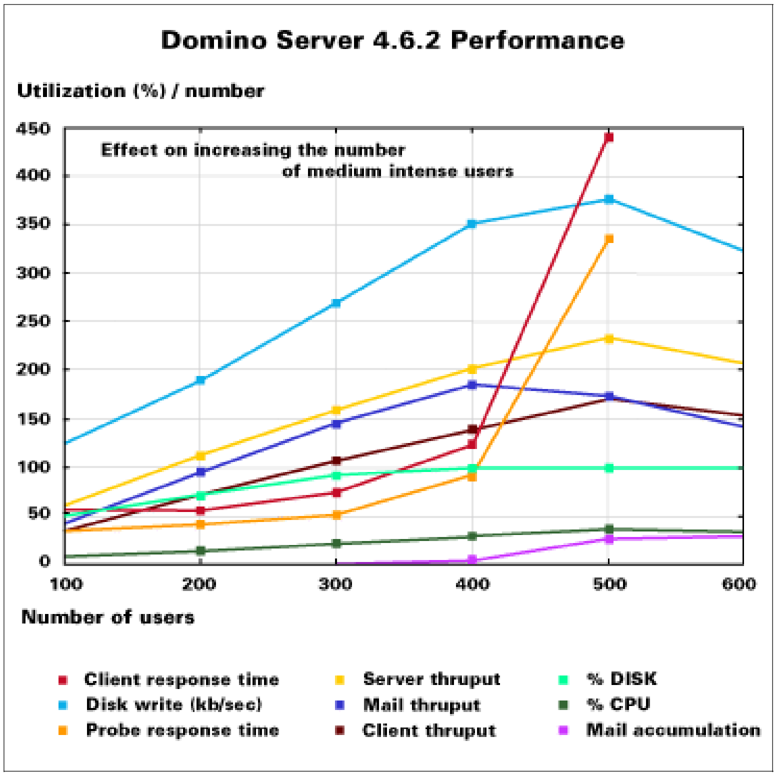
	Light users	**Medium users**	**Heavy users**
Client throughput (number of APIs/min)	2.45	3.37	8.50
Server throughput without cluster (number of transactions/min)	4..88	5.00	9.01
Mail throughput (number of mail messages/min)	0.11	0.46	2.40
Average mail size (bytes)	1,000	1,000	6,888

Note: Each light user in the test simulates two Geoplex light users because of limited disk space in the test servers. Therefore, if a test configuration turns out to be capable of supporting 100 light users, in reality it can support 200 such users. However, for the sake of simplicity, one test user is equivalent to one real life user. Although it affects absolute performance numbers, it does not affect the modeling methodology described here.

Developing an approach to building the model

Next, we see how Domino servers typically behave under increasing load. The following graph shows the performance of a non-clustered test server under medium load. The number of users ranged from 100 to 600, in increments of 100. The graph shows values for:

- Client response time -- Average of all API response times seen by a client.
- Probe response time -- Time to open a database on the server; it is measured periodically from a client and then averaged.
- Server throughput -- Number of transactions/second at the server.
- Client throughput -- Number of transactions/second at the client.
- Mail throughput -- Number of mail messages delivered by the server/second.
- %CPU and %DISK -- CPU and disk utilization at the server.
- Disk write -- Number of Kbytes written to the disk/second at the server.
- Mail accumulation -- Number of mail messages queued at the server, waiting to be delivered.



From the graph, we can see clearly that the CPU usage increases linearly with the number of users when the total number of users is less than 300. Client and server throughput values are also linear with fewer than 300 users. Response times increase only slightly between 100 and 300 users. And, while disk utilization is not linear, the number of bytes written to disk does increase linearly.

At 400 users, the disk becomes 100% utilized and mail messages can no longer be delivered. As a result, mail messages start accumulating and the mail throughput (and eventually the client and the server throughput) decreases. CPU utilization does not increase beyond this point, which indicates that no additional work gets done. Requests are queued up, resulting in a significant increase in response times above 400 users. This is what is known as the **critical point**. The critical point indicates the optimal number of concurrent users that the system can support. For the test server, that number is 350 medium users. A performance objective of system administrators is to keep systems operating below this critical point at all times.

The behavior shown above is typical for any Domino server with or without clustering. Our approach for building the predictive model is to find a linear increment in resource usage under low load, and extrapolate it to find the maximum number of users, using the critical point to limit the number of users.

In a computer system, any one of its resources -- CPU, disk space, memory, and network -- can become a performance bottleneck. However, the performance of some of these resources are influenced by factors other than usage, and hence are out of scope for this article. Network bottlenecks are easy to detect and resolve, and are out of the scope for this article. Memory requirements are easy to determine and are excluded from our model. Disk space can be a bottleneck either because of limited capacity or because of I/O bandwidth. Disk capacity requirement is a function of the total number of users supported by a server (as opposed to the number of concurrent users) and of policies such as the amount of disk space allocated per user; hence, it too is out of the scope of this article. Therefore, the key resource usage parameters for the predictive model are disk I/O bandwidth and CPU utilization.

Building the model

The following notations are used in the model:

Notation	Equals
$U_{a,b}$	CPU utilization for doing type *a* work by type *b* users. a ε {W(workload), I(initiating replication), R(receiving replication)} b ε {L(light), M(medium), H(heavy), X(mixed)}
U_{max}	Maximum allowable CPU utilization
$D_{a,b}$	Disk write in KBytes/sec for doing type *a* work by type *b* users. a ε {W(workload), R(receiving replication)} b ε {L(light), M(medium), H(heavy), X(mixed)}
$D_{max,b}$	Maximum allowable disk write in Kbytes/sec by type *b* users.
N	Number of users supported by a particular configuration.
N_a	Number of type *a* users supported by a particular configuration.
$N_{a,r}$	Number of type *a* users supported if constrained only by the resource *r.* r ε {CPU, Disk}.
S_I^{active}	Number of times an active server initiates replication due to an update of one of its databases.
S_R^{active}	Average number of times an active server receives replication request due to an update of one database at each of the active servers.
$S_R^{standby}$	Average number of times a standby server receives replication request due to an update of one database at each of the active servers.

K_R^{active}	Average number of times an active server performs replication due to an update of one database at each of the active servers.
$K_R^{standby}$	Average number of times a standby server performs replication due to an update of one database at each of the active servers.
F_a	Fixed disk write in Kbytes/sec when subjected to doing type *a* work. a ε {W(workload), R(replication)}
F	Fixed disk write in KBytes/sec. (does not depend on the number and type of users). = F_W (for servers that only initiate replication) = F_R (for servers that only receive replication) = $F_W + F_R$ (for servers that initiate as well as receive replication)

Obtaining model parameters

The test environment consists of six server and four client machines, connected by a 100-Mbit private Ethernet.

Each server machine:

- Has a single Pentium II 333 MHz processor, 512 MB RAM, 523 MB pagefile and 2 SCCI hard disks.
- Runs Domino extended server version 4.6.2 on Windows NT Server 4.0. On these machines, the operating system and Domino data files are placed on one drive and the Domino executables and the pagefile on the other.

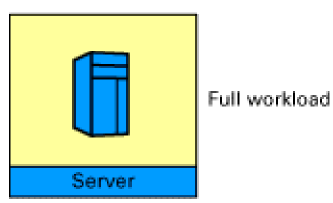
Each client machine:

- Has a single Pentium II 400 MHz processor, 256 MB RAM, 267 MB pagefile and a single SCCI hard disk.
- Runs Notes version 4.6a on Windows NT Workstation 4.0.

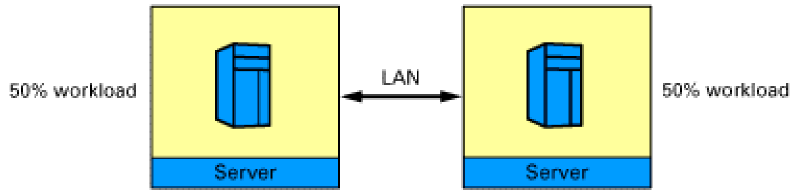
Base configurations

The figures below shows the three configurations, known as base configurations, that are required to build the model.

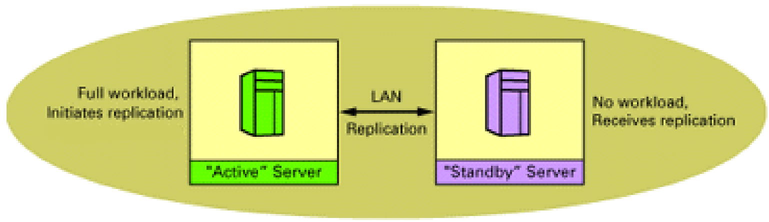
Configuration 1 consists of a single server with no clustering enabled; all workload is handled by the one server.



Configuration 2 has two servers with no clustering, and the workload is divided equally between the servers.



Configuration 3 has two servers in a cluster, but the workload is directed towards one server, referred to as the "active" server. All databases are replicated to the second server which acts as standby waiting for failover. Because all users are on one server, no messages are routed.



Base measurements
Using at least three different numbers of users for each of the three types of workload described in the test workload section, we measure the CPU utilization and disk-write bytes/sec at the servers for all three base configurations described above. The NT Performance Monitor Tool is used to determine these measurements, which we call the base measurements.

Using the base measurements, we find the CPU and disk utilization per user. CPU utilization for active servers due to user workload is obtained directly from the measurements of the two active servers in Configuration 2. CPU utilization for initiating cluster replication is obtained by subtracting the server CPU utilization in Configuration 1 from that for the active server in Configuration 3. CPU utilization for receiving replication is obtained from the measurement of the standby server in Configuration 3.

CPU usage for a single user can be obtained either by plotting the numbers or using mathematical calculations. We can plot the CPU utilization for the five different number of users and find the CPU requirements for our test configuration from the slope of the lines. Plotting provides a visual check for a linear relationship of CPU utilization with the number of users.

$U_{W,L} = 0.065$, $U_{I,L} = 0.053$, $U_{R,L} = 0.044$
$U_{W,M} = 0.093$, $U_{I,M} = 0.077$, $U_{R,M} = 0.061$
$U_{W,H} = 0.200$, $U_{I,H} = 0.180$, $U_{R,H} = 0.170$

Similarly, we find the disk parameters for the model:

$F_W = 50$, $F_R = 50$

$D_{W,L} = 0.50$, $D_{R,L} = 0.50$
$D_{W,M} = 0.80$, $D_{R,M} = 0.55$
$D_{W,H} = 3.30$, $D_{R,H} = 2.35$

$D_{max,L} = 350$, $D_{max,M} = 350$, $D_{max,H} = 500$

For all cases we assume $U_{max} = 75\%$.

Calculation of model parameters for a mixed workload
From the parameters for the three types of users, we can find model

parameters for a mixed load. We chose to use a mixture of light, medium, and heavy users with a ratio of 60:24:16. This ratio can be varied widely depending on the real mix of the users. We chose this ratio because it produces a workload close to what has been observed at the IBM Geoplex site.

$U_{W,X} = U_{W,L} * 0.60 + U_{W,M} * 0.24 + U_{W,H} * 0.16 = 0.093$
$U_{I,X} = U_{I,L} * 0.60 + U_{I,M} * 0.24 + U_{I,H} * 0.16 = 0.079$
$U_{R,X} = U_{R,L} * 0.60 + U_{R,M} * 0.24 + U_{R,H} * 0.16 = 0.068$

$D_{W,X} = D_{W,L} * 0.60 + D_{W,M} * 0.24 + D_{W,H} * 0.16 = 1.02$
$D_{R,X} = D_{R,L} * 0.60 + D_{R,M} * 0.24 + D_{R,H} * 0.16 = 0.81$

$D_{max,X} = D_{max,L} * 0.60 + D_{max,M} * 0.24 + D_{max,H} * 0.16 = 374$

Model usage and validation

In this section, we predict the maximum number of users supported by a series of configurations using our model, and then compare these numbers with the actual number of users obtained from our tests. We used the following four configurations:

Configuration 1: Two identical servers in a cluster with the workload evenly distributed between the servers. There is no standby server. All mail databases are replicated.

Configuration 2: Three identical servers in a cluster -- two active and one standby. Workload is evenly distributed between the two active servers. All mail databases in the two active servers are replicated with the standby server.

Configuration 3: Three identical servers in a cluster with workload evenly distributed among all the servers. No standby servers. All mail databases are replicated in all three servers.

Configuration 4: Six identical servers in a cluster -- four active and two standby. Workload is evenly distributed among the four active servers. All mail databases in the four active servers are evenly replicated in two standby servers.

Configuration	1	2	3	4
Total number of servers	2	3	3	6
Number of active servers	2	2	3	4
Number of standby servers	0	1	0	2
Number of copies of a database	2	2	3	2
S_I^{active}	1	1	3	1
S_R^{active}	1	0	3	0
$S_R^{standby}$	0	2	0	2

K_R^{active}	1	0	2	0
$K_R^{standby}$	0	2	0	2

Configuration 1 -- Light users

Predicted server performance (all servers are active servers).

$U_L = U_{W,L} + U_{I,L} * S_I^{active} + U_{R,L} * S_R^{active}$
= 0.065 + 0.053 * 1 + 0.044 * 1
= 0.162

$F = F_W + F_R$
= 50 + 50
= 100 Kbytes/sec

$D_L = D_{W,L} + D_{R,L} * K_R^{active}$
= 0.5 + 0.5 * 1
= 1.0 Kbytes/sec

$N_{L,CPU} = U_{max/UL}$ = 75.0 ÷ 0.162 = 462
$N_{L,Disk} = (D_{max} - F) \div D_L$ = (350-100) ÷ 1.0 = 250

Therefore, the number of users per server
$N_L = MIN(N_{L,CPU}, N_{L,Disk})$
= 250 (limited by the disk I/O rate)

Results from the actual measurement

Number of users	100	200	300	400
Client response time (ms)	49.4	69.9	148.5	246.2
Probe response time (ms)	65.3	106.0	152.0	227.0
CPU utilization (%)	16.6	34.6	34.9	34.2
Disk write (Kbytes/sec)	201	295	342	346
Disk utilization (%)	82.0	98.5	99.9	100.0

Comparison of predicted and actual performance

	Predicted	Actual
Maximum number of users	250	200 to 300
CPU utilization per 100 users (%)	16.2	16.9
Disk write (Kbytes/sec) for 250 users	350	295 to 342

Configuration 1 -- Medium users

Predicted server performance (all servers are active servers).

$U_L = U_{W,M} + U_{I,M} * S_I^{active} + U_{R,M} * S_R^{active}$

= 0.093 + 0.077 * 1 + 0.061 * 1
= 0.231

$F = F_W + F_R$
= 50 + 50
= 100 Kbytes/sec

$D_M = D_{W,M} + D_{R,M} * K_R^{active}$
= 0.80 + 0.55 * 1
= 1.35 Kbytes/sec

$N_{M,CPU} = U_{max/UM} = 75.0 \div 0.231 = 325$
$N_{M,Disk} = (D_{max} - F) \div D_M = (350-100) \div 1.35 = 185$

Therefore, the number of users per server
$N_M = MIN(N_{M,CPU}, N_{M,Disk})$
= 185 (limited by the disk I/O rate)

Results from the actual measurement

Number of users	100	200	300
Client response time (ms)	82.0	161.0	360.0
Probe response time (ms)	85.0	147.0	393.0
CPU utilization (%)	23.2	40.5	34.0
Disk write (Kbytes/sec)	245	384	365
Disk utilization (%)	85.6	99.0	99.0

Comparison of predicted and actual performance

	Predicted	**Actual**
Maximum number of users	185	100 to 200
CPU utilization per 100 users (%)	23.1	23.2
Disk write (Kbytes/sec) for 185 users	350	245 to 384

Configuration 1 -- Heavy users

Predicted server performance (all servers are active servers).

$U_H = U_{W,H} + U_{I,H} * S_I^{active} + U_{R,H} * S_R^{active}$
= 0.20 + 0.18 * 1 + 0.17 * 1
= 0.55

$F = F_W + F_R$
= 50 + 50
= 100 Kbytes/sec

$D_H = D_{W,H} + D_{R,H} * K_R^{active}$
= 3.3 + 2.35 * 1

= 5.65 Kbytes/sec

$N_{H,CPU} = U_{max} \div U_H = 75.0 \div 0.55 = 136$
$N_{H,Disk} = (D_{max} - F) \div D_H = (500-100) \div 5.65 = 71$

Therefore, the number of users per server N_H
$= MIN(N_{H,CPU}, N_{H,Disk})$
= 71 (limited by the disk I/O rate)

Results from the actual measurement

Number of users	20	40	60	80	100
Client response time (ms)	30.3	55.4	70.0	109.6	146.6
Probe response time (ms)	50.3	65.4	82.5	85.8	109.2
CPU utilization (%)	14.6	25.4	33.5	34.0	35.2
Disk write (Kbytes/sec)	164	312	417	503	529
Disk utilization (%)	61.2	89.2	99.3	99.9	100.0

Comparison of predicted and actual performance

	Predicted	Actual
Maximum number of users	71	60 to 80
CPU utilization per 100 users (%)	55.0	64.0
Disk write (Kbytes/sec) for 70	496	417 to 503

Configuration 1 -- Mixed users

Predicted server performance (all servers are active servers).

$U_x = U_{W,X} + U_{I,X} * S_I^{active} + U_{R,X} * S_R^{active}$
= 0.093 + 0.079 * 1 + 0.068 * 1
= 0.24

$F = F_W + F_R$
= 50 + 50
= 100 Kbytes/sec

$D_x = D_{W,X} + D_{R,X} * K_R^{active}$
= 1.02 + 0.81 * 1
= 1.83 Kbytes/sec

$N_{X,CPU} = U_{max}/U_x = 75.0/0.24 = 313$
$N_{X,Disk} = (D_{max} - F)/D_x = (374-100) \div 1.83 = 150$

Therefore, the number of users per server N_x
$= MIN(N_{X,CPU}, N_{X,Disk})$
= 150 (limited by the Disk I/O rate)

Results from the actual measurement

Number of users	100	125	150	175	200	225	250	300
Client	58.9	67.0	80.8	106.	117.	150.	188.	207.

response time (ms)				1	9	8	5	8
Probe response time (ms)	78.7	112.1	110.4	112.6	122.4	133.5	119.1	119.4
CPU utilization (%)	29.6	36.9	41.4	43.0	37.6	37.2	36.7	34.2
Disk write (Kbytes/sec)	272	327	361	392	431	443	479	440
Disk utilization (%)	92.8	98.5	99.4	99.9	100.0	100.0	100.0	100.0

Comparison of predicted and actual performance

	Predicted	Actual
Maximum number of users	156	150 to 175
CPU utilization per 100 users (%)	24.0	29.6
Disk write (Kbytes/sec) for 156	386	361 to 392

Configuration 2 -- Mixed users

Predicted server performance (has both active and standby servers).

Active servers

$U_x^{active} = U_{W,X} + U_{I,X} * S_I^{active} + U_{R,X} * S_R^{active}$
$= 0.093 + 0.079 * 1 + .068 * 0$
$= 0.172$

$F = F_W = 50$ Kbytes/sec

$D_X^{active} = D_{W,X} + D_{R,X} * K_R^{active}$
$= 1.02 + 0.81 * 0$
$= 1.02$ Kbytes/sec

$N_{X,CPU} = U_{max}/U_X^{active} = 75.0 \div 0.172 = 436$
$N_{X,Disk} = (D_{max} - F)/D_X^{active} = (374 - 50) \div 1.02 = 318$

Therefore, the number of users per active server N_X^{active}
$= MIN(N_{X,CPU}, N_{X,Disk})$
= 318 (limited by the Disk I/O rate)

Standby servers

$U_X^{standby} = U_{R,X} * S_R^{standby}$
$= 0.068 * 2$
$= 0.136$

$F = F_R = 50$ Kbytes/sec
$D_X^{standby} = D_{R,X} * K_R^{standby} = 0.81 * 2 = 1.62$ Kbytes/sec

$N_{X,CPU} = U_{max}/U_X^{standby} = 75.0 \div 0.136 = 551$

$N_{X,Disk} = (D_{max} - F)/D_X^{standby} = (374-50) \div 1.62 = 200$

Therefore, the number of users per standby server $N_X^{standby}$
$= MIN(N_{X,CPU}, N_{X,Disk})$
= 200 (limited by the Disk I/O rate)

Therefore, the number of users supported by the configuration
$= MIN(N_X^{active}, N_X^{standby})$
= 200 (limited by the Disk I/O rate at the standby server)

Results from the actual measurement

Active servers

Number of users	100	150	200	225	250	275
Client response time (ms)	34.6	36.2	41.7	45.7	49.5	56.9
Probe response time (ms)	55.8	66.2	87.3	76.3	89.0	85.4
CPU utilization (%)	19.8	28.1	31.9	34.7	36.2	38.8
Disk write (Kbytes/sec)	196	235	291	334	358	386
Disk utilization (%)	74.2	85.9	94.1	97.4	98.0	99.2

Standby servers

Number of users	100	150	200	225	250	275
CPU utilization (%)	16.6	25.1	25.5	24.9	24.8	25.5
Disk write (Kbytes/sec)	205	313	372	383	388	393
Disk utilization (%)	88.6	98.8	99.9	100	100	100

Comparison of predicted and actual performance

	Predicted	**Actual**
Active server		
Maximum number of users	318	> 275
CPU utilization per 100 users (%)	17.2	18.2
Disk write (Kbytes/sec) for 200 users	254	291
Standby server		

Maximum number of users	200	200 to 225
CPU utilization per 100 users (%)	13.6	15.4
Disk write (Kbytes/sec) for 200 users	374	372

Configuration 3 -- Mixed users

Predicted server performance (all servers are active servers).

$U_x = U_{W,X} + U_{I,X} * S_I^{active} + U_{R,X} * S_R^{active}$
= 0.093 + 0.079 * 3 + 0.068 * 3
= 0.534

$F = F_W + F_R$
= 50 + 50
= 100 Kbytes/sec

$D_x = D_{W,X} + D_{R,X} * K_R^{active}$
= 1.02 + 0.81 * 2
= 2.64 Kbytes/sec

$N_{X,CPU} = U_{max} \div U_x$ = 75.0 ÷ 0.534 = 140
$N_{X,Disk} = (D_{max} - F) \div D_x$ = (374-100) ÷ 2.64 = 104

Therefore, the number of users per server N_x
= MIN($N_{X,CPU}$, $N_{X,Disk}$)
= 104 (limited by the Disk I/O rate)

Results from the actual measurement

Number of users	25	50	75	100	125	150
Client response time (ms)	41.4	42.3	56.8	63.6	76.2	82.6
Probe response time (ms)	53.3	65.4	79.6	81.0	95.9	133.8
CPU utilization (%)	16.0	29.9	40.8	30.6	32.7	36.0
Disk write (Kbytes/sec)	160	229	307	323	350	373
Disk utilization (%)	68.2	90.3	98.6	97.8	98.3	99.4

Comparison of predicted and actual performance

	Predicted	**Actual**
Maximum number of users	104	75 to 100
CPU utilization per 100 users (%)	53.4	59.8
Disk write (Kbytes/sec) for 104 users	374	> 307

Configuration 4 -- Mixed users

Predicted server performance (has both active and standby servers).

The model parameters for this configuration are the same as those for

Configuration 2. Therefore, the predicted performance of these two configurations are identical.

Results from the actual measurement

Active server

Number of users	50	100	150	200	250	300
Client response time (ms)	31.4	34.9	39.6	47.2	64.8	88.6
Probe response time (ms)	45.9	57.7	65.3	70.8	93.3	95.2
CPU utilization (%)	9.0	18.0	25.7	27.6	33.4	36.7
Disk write (Kbytes/sec)	134.4	206.9	255	278.2	400	464.2
Disk utilization (%)	48.3	74.8	89.1	88.9	99.0	99.0
Mail accumulation (Number/min)	0.0	0.0	0.0	0.0	0.0	0.0

Standby server

Number of users	50	100	150	200	250	300
CPU utilization (%)	9.6	18.3	23.8	21.2	22.1	22.3
Disk write (Kbytes/sec)	176.1	234.5	336.8	349.4	379.3	392.3
Disk utilization (%)	71.8	91.5	99.2	99.2	99.8	99.8

Comparison of predicted and actual performance

	Predicted	**Actual**
Active server		
Maximum number of users	318	> 200
CPU utilization per 100 users (%)	17.2	19.4
Disk write (Kbytes/sec) for 150 users	203	255
Standby server		
Maximum number of users	200	150 to 200
CPU utilization per 100 users (%)	13.6	15
Disk write (Kbytes/sec) for 150 users	293	337

Conclusion

It is possible to create a performance model for Domino clusters, given a set of initial measurements for three basic configurations. Using the performance model, the number of users supported by different numbers of active and standby servers with different levels of replication can be predicted. Actual measurements from a test configuration shows that the predicted numbers can be very close to actual figures.

These models can help Domino administrators in capacity planning of Domino clusters, as well as can provide helpful insights regarding performance bottlenecks in the system. For example, the model for the hardware used in our test reveals that the disk I/O is the bottleneck in all configurations and, in some cases, the number of users supported by the servers can almost be doubled by increasing the bandwidth for disk I/O (for example, by using RAID disks).

ABOUT THE AUTHOR
Masud Khandker joined the DCE performance group in IBM, Austin in 1997 after receiving his Ph.D. and M.S.E degrees in Computer Science and Engineering from the University of Michigan. He is currently involved in the performance analysis of IBM PKIX products. One of his primary areas of interest is performance modeling of distributed systems. When he is not working, Masud likes to play bridge and spend time with his wife and son.

ACKNOWLEDGEMENTS
Many thanks to Ginny Roarabaugh for leading the Notes Cluster performance project and for her interest, ideas and comments. Special thanks to Harry Murray of Iris Associates for reviewing and to Amy E. Smith of Lotus for editing.

What do you think about this article?

Register Here!