Lotus. software

Best Practices for Lotus Domino on System z: z9 and zSeries

- Best practices for implementing Lotus Domino on an IBM mainframe
- Performance, tuning, and capacity planning
- Technical support overview and procedures

Mike Ebbers
Armelle Cheve
Barbara Filippi
Wu W. Huang
DeWayne Hughes
Francelis Ordas
Teri Schenke
Mike Wojton

Redbooks

ibm.com/redbooks

International Technical Support Organization

Best Practices for Lotus Domino on System z

January 2006

SG24-7209-00

Note: Before using this information and the product it supports, read the information in “Notices” on page xi.

First Edition (January 2006)

This edition applies to Releases 6 and 7 of Lotus Domino on System z9.

Contents

Figures

Tables

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

@server®
Domino Designer®
Domino®
DB2®
Enterprise Storage Server®
ESCON®
FICON®
HiperSockets™
IBM®
IBMLink™
Lotus Notes®
Lotus®
Lotusphere®
Notes®
OS/400®
Passport Advantage®
Redbooks™
Redbooks (logo) ™
RMF™
S/390®
System z™
System z9™
Tivoli®
z/OS®
z/VM®
zSeries®
z9™

The following terms are trademarks of other companies:

Java, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Preface

Lotus® Domino® provides much more than just e-mail. IBM® Lotus Notes® and Domino are designed to help organizations increase productivity and responsiveness through business-critical collaboration capabilities and a rapid application development and deployment environment. The IBM System z9™ delivers highly reliable, highly scalable, mission-critical transaction servers in the marketplace. It is designed with industry leading tools for system management which enables customers to deliver robust service for a large number of users with a small support staff.

To most effectively use your system capacity and get the best performance from Domino, we recommend the best practices described in this IBM Redbook.

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

Mike Ebbers is a Consulting IT Specialist in the International Technical Support Organization (ITSO) at IBM Poughkeepsie. He has been with IBM for 32 years and produces documentation on Domino for System z9 as well as other z/OS® Redbooks™ and textbooks.

Armelle Cheve is an IT Specialist in the IBM ATS Products and Solutions Support Center, Montpellier, France. She has five years of experience in Domino for System z9 Advanced Technical Sales Support in EMEA. She has 12 years of IT experience, including development in C, SQL, Java™, LotusScript, Domino administration on distributed and Mainframe environments, and performance benchmark tests for European customer accounts.

Barbara Filippi is a Consulting IT Specialist with the Domino for System z9 Team in the Washington Systems Center. She has worked at IBM for 26 years and has been involved with Domino on System z9 since it initially became available on that platform. Her areas of expertise include Domino installation and administration, capacity planning, performance analysis, and migration to System z9 from other Domino platforms.

Wu W. Huang is a member of the Lotus Domino Performance team in Poughkeepsie. His main focus is on Domino performance for Linux® on System z9, z/OS, and Rhel 4. He joined the IBM System z9 Domino Performance Team in 1998. He is a member of the NotesBench Consortium.

DeWayne Hughes is a Consulting IT Specialist and Systems Architect. He has 16 years of experience at IBM, including roles in many of the IBM server platforms. His areas of expertise include infrastructure architecture, designing Domino deployments across the IBM @server® family, and performing TCO value analysis. He has written about subjects, such as Domino benchmarking, comprehensive TCO analysis for Domino server environments, and Domino for z/OS best practices.

Francelis Ordas is a Senior IT Specialist for the Products and Solutions Support Center, Montpellier, France. He joined IBM in 1977 and has held several specialist positions as well as international assignments, all dealing with S/390® and System z9 technical support. He has more than eight years of experience supporting Europe on Domino on System z9.

Teri Schenke is an IBM Certified Sales Specialist for the System z9 Solution Sales Organization. Teri currently is the Sales Leader for Domino on System z9 for the America's Group. She works directly with customers, Business Partners, software development, ISVs and support organizations. Teri has been directly involved with Domino on System z9 since its release in 1998. Teri has worked at IBM for 26 years in various sales and marketing assignments.

Mike Wojton is a Certified Senior IT Specialist. He has over 23 years of experience in the IT industry working at IBM. He currently works at the Washington System Center supporting Domino for System z9. He has been involved with Domino for System z9 since the first Beta release with 4.5.1 in 1997 and performed the first install with a Beta customer. He has presented and written on areas such as Domino installation, benchmarking, performance, capacity planning, problem determination and problem source identification. He has also participated in the Performance Zone at Lotusphere® since 2000.

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our Redbooks to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- Use the online **Contact us** review redbook form found at:

  ibm.com/redbooks

- Send your comments in an email to:

  redbook@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYJ Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

1

The value of Domino on System z9

Whether you are deploying Domino for simple discussion databases or for core people-driven applications (such as, customer relationship management, help desks, or supply chain management), Lotus Domino offers high-function, integrated, flexible, world class messaging, and collaboration. Organizations have the option of deploying Lotus Domino on several hardware platforms.

1.1 Overview

While Domino on System z9 may appear to be just another Domino server, the System z9 platform provides many unique functions. Along with the Linux and z/OS operating systems, it has a long history. It has matured into a powerful server platform that meets the diverse needs of many organizations.   IBM System z9 offers key functions such as clustering for scalability and availability and systems management tools. These tools, which are important to organizations, are mature and have been available on System z9 for many years. System z9 is able to support large workloads because of the unique capabilities of the hardware and software.

The IBM System z9 delivers highly reliable, mission-critical transaction servers in the marketplace. As diverse workloads and complexity increase, the IBM System z9 continues to be very attractive. IBM System z9 delivers the highest service levels required for advanced processing while allowing customers to use existing policy and procedures, assets and business expertise. IBM System z9 is designed with industry leading tools for system management which enables customers to deliver robust service for a large number of users with a small support staff.

By enabling organizations to consolidate, centralize and simplify their Domino infrastructure, the IBM System z9 delivers unsurpassed quality of service, enhanced manageability and low cost of ownership.

1.2 Reasons to choose System z9

When deciding on a platform on which to deploy Domino, many things should be considered. These include the choice of operating system, total cost, platform features, and support.

1.2.1 Operating system

Linux

Linux allows you to change the cost model for delivering the Domino service. While Linux can be downloaded from the Internet, it is likely that you will pay for the Linux distribution and for Linux support. However, you can isolate the Domino application by deploying Domino on an Integrated Facility for Linux (IFL) processor. A single System z9 system can host many Domino servers spread across multiple images of Linux. This enables a single System z9 system to support tens of thousands of Notes users and applications.

z/OS

Under z/OS, Domino exploits more System z9 hardware features, such as the hardware cryptographic facility that improves SSL performance. z/OS also contains many advanced functions for running mission-critical applications. Half of the operating system kernel is there for error recovery, so the z/OS operating system seldom fails. Many administration tasks can be done without bringing the z/OS operating system down. z/OS can run thousands of Domino users in a single Domino partition (DPAR) under z/OS.

Linux under z/VM

There are a number of reasons why people are choosing to run Linux as a guest operating system under z/VM®. Here are some of them. For a complete list, see:

http://www.vm.ibm.com/linux/benefits.html

- **Sharing resources**: Resources can be shared among multiple Linux images running on the same VM system. These resources include: CPU cycles, memory, storage devices, and network adapters.
- **Virtualization**: The virtual machine environment is highly flexible and adaptable. New Linux guests can be added to a VM system quickly and easily without requiring dedicated resources. This is useful for replicating servers in addition to giving users a highly flexible test environment.
- **System z9 advantages**: Running Linux on VM means the Linux guest(s) can transparently take advantage of VM support for System z9 hardware architecture and RAS features.
- **Automation**: VMs long-standing support for scheduling, automation, performance monitoring and reporting, and virtual machine management is available for Linux virtual machines as well!
- **Horizontal growth**: An effective way to grow your Linux workload capacity is to add more Linux guests to a VM system.

1.2.2 Cost

Distributed server environments can be very difficult to maintain. Multiple sources of data have to be kept uniformly updated and application changes have to be replicated throughout the network. This requires time and skilled resources. As the number of servers increase, so do the number of skilled people it takes to keep the infrastructure running and responsive to the end-users. With System z9 scalability, you support fewer hardware servers and Domino servers, you benefit from less complexity and lower cost of administration and management. System z9 provides additional function that will help reduce your support effort and therefore lower your personnel cost.

1.2.3 Hardware platform

System z9 servers are outstandingly reliable and scalable servers for mission-critical applications in the marketplace. You can add capacity to the server while the system is running. This allows you to start small, then add capacity as the number of users or applications increase.

1.2.4 Support

IBM offers superior Domino support for both z/OS and Linux. Refer to Chapter 5, “Domino for System z technical support overview and recommendations” on page 41 for details of the Support Offerings.

In summary, the System z9 platform has a number of strengths, including:

- A choice of robust operating systems
- A cost-efficient server environment
- World-class hardware reliability and scalability
- Renowned IBM support

When organizations combine the strengths of the System z9 with the business critical collaborative capabilities and rapid application development and deployment capacity of Lotus Domino, the result is an information technology asset that provides exceptional business value to the customer.

Part 1

Best practices

In Part 1 we discuss tips and techniques for implementing Domino on the IBM z9 and System z9 platform. Experience has shown these tips to be best practices.

2

Domino for System z9 best practices

Domino is a complex middleware software product. To most effectively use your capacity and get the best performance from Domino, we recommend following some best practices to optimally run Domino applications and fine-tune your Domino deployment.

In this chapter, we discuss the following topics:

- Domino maintenance recommendations
- File system recommendations for Domino
- Reasons about whether or not to consolidate Domino servers
- How to configure and run Domino to get the most out of available capacity
- The benefits of off-loading a Domino server workload to clients
- The impact of large databases on capacity and performance
- Domino features that should always be implemented
- Tuning Domino for performance and capacity with NOTES.INI parameters
- Monitoring Domino servers

2.1 Recommendations for maintenance schedules

In general, we recommend that you upgrade Domino to a new major release or maintenance release at least once a year to take advantage of enhancements/new functionality and fixes. Additionally, it is important to track the availability of Domino Fix Packs between regularly scheduled maintenance releases. Fix Packs are small, unscheduled maintenance releases consisting of a few fixes to address significant, high-impact issues. Given the nature of the fixes, Fix Packs associated with your specific maintenance release should be tested and installed as soon as possible. For more information about Domino maintenance strategy, see Chapter 5, “Domino for System z technical support overview and recommendations” on page 41.

To optimally run Domino and avoid any issues, all recommended maintenance and configuration settings for z/OS or Linux for System z9 should be followed to the letter. This information is documented in the *Lotus Domino for z/OS Version 7 Installation Guide* (G210-1994-00), *Lotus Domino Version 7 Installing Domino Servers* (G210-1993-00), and the Domino 7 Release Notes. Contact the IBM Support Center for the latest recommendations on additional maintenance and configuration settings. Additionally, the following Web sites contain useful links pointing to general service recommendations by operating system:

- For zOS, http://www-03.ibm.com/servers/eserver/zseries/zos/servicetst/
- For Linux, http://linuxvm.org/
- For zVM, http://204.146.134.18/overview/

2.2 File system recommendations for Domino

For those customers running Domino on z/OS, we recommend that you implement zFS rather than HFS file systems for Domino to achieve better throughput. Additionally, we strongly encourage you to:

- Stay current with zFS maintenance as much as possible.
- Use the NOREADAHEAD parameter for the /names, /notesdata and mail directory mount points.
- Set up IOEZPRM for better problem determination to collect trace table and data set information.

If you are running Domino on Linux for System z9, then the EXT3 file system is recommended because it has considerably less overhead than ReiserFS and has the second best throughput with journaling for error recovery. Only EXT2 has better throughput, but without any error recovery capability.

2.3 To consolidate or not consolidate servers

Here we discuss reasons to add additional Domino servers as well as reasons to consolidate many servers into a fewer number of servers.

2.3.1 Reasons to deploy additional Domino servers

There are several valid reasons to functionally separate Domino servers:

- Applications unrelated to mail should always be deployed on different Domino servers than mail.

Domino tuning parameters can be quite different for mail and other application deployments, making the separation necessary to optimally tune each environment. From the standpoint of the System z9 operating system, application and mail server segregation gives you flexibility to prioritize these workloads differently. For example, applications and mail might normally share the same priority to obtain system resources. But at month-end, it might be more important to complete the application work in the shortest time possible rather than to provide the very best response times to mail users. In this case, separate application and mail servers together with System z9 LPAR management, z/OS Workload Manager, or z/VM guest prioritization can be used to direct system resources to where business requirements dictate they are most needed.

- The Administration server should always be separate from mail or application servers.
  A separate Administration server gives administrators the option to execute AdminP requests against the Domino Directory without concerns about increased contention and CPU utilization on application/mail servers. Additionally, since the AdminP server is usually the first server to receive new code during an upgrade, the impact on end users is minimized.
- For large Domino deployments, a hub and spoke architecture for mail routing and replication is more efficient because every server doesn’t need to initiate sessions with every other server.
  Additionally, server administration is simplified with this architecture. However, for smaller deployments of a few Domino servers, this type of architecture is typically more costly. The CPU overhead, and memory and disk costs of initializing hub servers must be weighed against their benefits. Routing mail through a hub in a small deployment may also have greater associated costs than directly routing/replicating to target servers.
- Sometimes there is a need to set up a stand-alone SMTP server.
  On Domino networks that don't use SMTP for internal mail routing, you can implement a gateway topology for sending outbound mail to the Internet. Internal servers can continue to use Notes routing to transfer mail and send Internet mail to an SMTP server that connects to the Internet.

2.3.2 Reasons to consolidate Domino servers

Although there are some benefits to creating individual servers for specialized functions, there are serious drawbacks in deploying many small servers of the same type, like mail servers, in terms of additional CPU, memory and disk requirements. Figure 2-1 on page 10 demonstrates this. It was taken from a Domino 6 NRPC Notebench, shows a 4% increase in CPU when running the same workload on z/OS in an unconstrained environment for two Domino servers versus one server. This overhead applies to all releases of Domino.

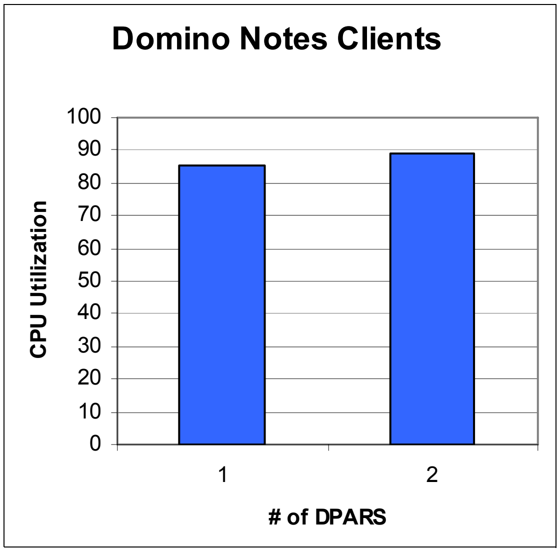


Figure 2-1 Increase in CPU with multiple DPARs - 1

Running a similar test on Linux for System z9 (SLES8 31-bit), Figure 2-2 shows CPU increases of 9% and 36% for two DPARs (Domino partitions) and three DPARs, respectively. The penalties for not consolidating Domino servers in this case are more exaggerated because of swapping issues with the 31-bit kernel. In a Linux image with sufficient memory allocation, the increases would be less, but still noticeable.

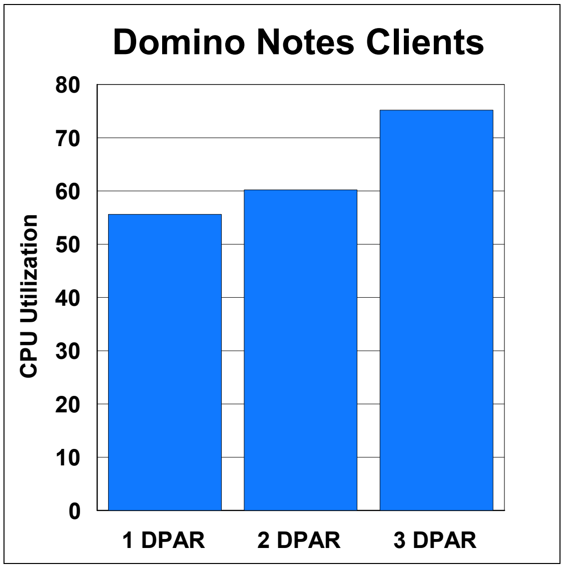


Figure 2-2 Increase in CPU with multiple DPARs - 2

More recent testing has shown that consolidating from four Domino servers to a single server results in a 10% CPU improvement overall. For more information, see the DeveloperWorks article entitled "Best Practices for Large Lotus Notes Mail Files" at:

http://www.ibm.com/developerworks/lotus/library/notes-mail-files/index.html

2.4 Using resources wisely

Using your resources wisely will improve capacity and performance. Here are some tips.

2.4.1 Do not run unnecessary server tasks

Each task loaded by the Domino server consumes some subset of system resources. To optimize the utilization of these resources, it is necessary to minimize the tasks run by Domino, the frequency at which they are run, and the elapsed time in which they will run. At initialization time, Domino starts the Server task, which addresses NRPC requests, and some other tasks defined in the NOTES.INI configuration file:

```
ServerTasks=Update,Replica,Router,AMgr,AdminP,CalConn,Sched,HTTP
ServerTasksAt1=Catalog,Design
ServerTasksAt2=UpdAll
ServerTasksAt5=Statlog
```

The first line above describes tasks which continue to execute until the server is shut down. It is an example of what tasks might be run at server startup; some tasks might be excluded and others included at initialization time. The other lines describe tasks which execute at specific times of the day entered in 24-hour format. Again, the tasks and times may vary from customer to customer.

Capacity and performance can be improved by removing any unnecessary tasks:

- **HTTP:** Turn off this task if users do not require Web access. The HTTP task requires significant resources to even initialize.
- **Replica:** This task adds a fair amount of overhead when loaded and is only required if replication is initiated from the Domino server. Note that without the Replica task, a server can still be a target for replication from other servers.
- **Router:** This task also requires significant resources when loaded. If mail routing is not required on a specific server, like an application server or replication hub, then remove it.
- **AMgr:** Depending on what kinds of agents are being run, this task can consume considerable CPU resources. For mail servers, limit the number of AMgr tasks running during prime shift to 1; off-shift, it may be appropriate to increase the number of AMgr tasks to 2 or more. See 2.4.6, “Controlling the Agent Manager task” on page 16 for more information about controlling the resource requirements for the AMgr task.
- **Collector, Reporter:** Turn off these tasks if you are not using the server to automatically track server statistics on a regular basis. If required, statistic collection can be implemented on a demand basis.
- **Sched, CalConn:** Turn off these tasks if the server is not using scheduling and calendaring. Hub and application servers typically do not need these functions.

One of the recommendations from the Domino Administration Help database for improving end-user response time on Domino servers is to define multiple Update tasks to improve view and full-text indexing update performance. The general guideline for number of Update tasks is to enable a maximum of one Update task per engine/processor on a multi-processor hardware server. What is not stated in this guideline is the underlying assumption that only one Domino server has access to all of the processors. For System z9, it doesn’t take into account that multiple LPARs/z/VM guests with multiple Domino partitions executing in them might all be sharing the same set of processors. For most mail servers, one Update task should be able to sufficiently handle the workload. Other applications, however, might have different requirements for the number of Update tasks. The overall recommendation is to increase the number of Update tasks in small increments, at the same time monitoring any negative impacts to capacity and performance.

Similarly, there is a recommendation to increase the number of Replica tasks to shorten replication cycles. However, the more replication processes running, the greater the impact to DPAR performance. Whereas it is quite appropriate to have many Replica tasks running on a replication hub, there could be adverse effects on a mail server where longer-running replications might contend with the Server/HTTP tasks which service end-user requests. It is important to carefully monitor a Domino server when making changes to the number of Replica tasks.

It is our experience that certain Domino tasks/features can be very resource intensive, such as Domino Message Tracking, Single Copy Template, Clustering, and some others. You need to give careful consideration as to whether this functionality is really needed for your particular user community. For a more in-depth discussion of Domino Clustering considerations, see 2.4.7, “Limit Domino clustering” on page 16.

2.4.2 Scheduling utilities

Schedule CPU-intensive utilities during non-peak processing times. Here are some examples.

Domino administrative utilities

Updall, Compact and Fixup are Domino administrative utilities which when run against a large number of databases (or even a small number of very large databases) can be very CPU intensive. For these scenarios, they should never be run during daytime peak processing periods because of the likelihood of elongated end-user response times.

In general, the following schedules are recommended for these utilities:

- Updall should be run nightly to rebuild view indexes and full-text indexes for all Domino databases.
- Compact should be run against all databases on a weekly basis. With the -B option, all unused/white space is recovered and file sizes are reduced; with the -b option, white space is recovered, but file sizes are not reduced. The -S option followed by a percentage will only compact databases when white space equals or has exceeded the specified percentage. Compact with the -b option is the recommended compaction style, but there are many others documented in the Domino Administration Help database. Which style you choose is dependent on your goals of reducing disk space, reorganizing databases to improve performance, or some other reasons. But keep in mind that running compact against a database with 5-10% of white space will not realize any noticeable performance gains.

  If running transaction logging in conjunction with a backup tool providing point-in-time database recovery, then the style, frequency, and schedules of compactions need to be considered in planning your backup/recovery strategy. For more information, see Chapter 4, “Backup/recovery best practices” on page 35.
- Run Fixup as needed against selected databases to fix corrupted views and documents. For databases which are transaction logged, there should never be a reason to run Fixup (unless requested to by the Support Center) if the Translog_AutoFixup parameter in NOTES.INI is set to 1. This setting will cause Domino to automatically repair any databases with corruption issues.

  If running transaction logging in conjunction with a backup tool providing point-in-time database recovery, a full backup of any databases that have been repaired with Fixup will be required immediately after running this utility. For more information, see Chapter 4, “Backup/recovery best practices” on page 35.

- Updall, Compact and Fixup are intended to execute serially to avoid locking and other issues. Each of these utilities offers the flexibility to run against specific databases, all databases within a folder, and all databases on a Domino server. If CPU cycles are available during off-shift periods, then multiple instances of these utilities may be run against different folders to shorten processing windows as long as there is no overlap of folder processing between any of the utilities.
- Backups of a large number of databases and in some cases recoveries of individual databases should be scheduled to run off-shift and at different times than other Domino utilities because of the potentially large demand on system resources.

The Administration Process (AdminP) automates routine administrative tasks such as name, mail file, Server document, Person document, and user mail file management. If the number of AdminP requests is small, then there are few issues in allowing AdminP to update the Domino Directory and mail/application files during prime-shift. When the number of AdminP requests is large, there is the potential of a huge drain on CPU resources and also contention against the Domino Directory, which could impact end-user response times. The best practice to implement a separate AdminP server allows administrators to create/stage AdminP requests during prime-shift, and then propagate the changes to the rest of the domain at an appropriate low-activity time by replicating the updated Domino Directory or the AdminP request database (Admin4.nsf) for other changes.

2.4.3 Do not implement unnecessary logging and debugging

To improve Domino server performance:

- Limit the amount of information that is logged to LOG.nsf and the console log.
- For those parameters which provide various levels of logging (mail logging, log_replication, log_update, etc.), choose the less verbose versions of logging if possible to reduce the amount of output and potentially lower CPU requirements.
- Disable HTTP server logging to improve Web performance.
  Logging options are stored in the Server document. In the HTTP server “Enable logging to” section there are two fields, Log files and DOMLOG.NSF. Disabling both of these fields improves Web server performance.
- Disable parameters starting with “debug” when troubleshooting has been completed. They can add considerable CPU overhead as well as generate a large amount of output.
- Do not log things which you never review.

2.4.4 Network considerations

Bandwidth is typically not an issue on System z9 unless there are remote clients on constrained networks. For this situation, there are basically three recommendations:

- If possible, have the remote clients deploy local replicas.
  Whereas this approach works very well for mail servers where each client has their own mail file, it may not be a good solution for application servers where many clients might share a large database(s).
- Implement network compression to reduce bandwidth.
  This is a very good solution if there are adequate cycles on the server to support the decompression/compression which needs to take place when processing network data. Certainly, the additional CPU cost of network compression must be weighed against the cost of upgrading remote networks.

  Network compression works only if enabled on both sides of the connection, either a client to server, or server to server connection. To limit costs, compression should be turned on for only those situations which require it. For example, there would be no reason to incur the cost of network compression between a hub/replication server and all of its spoke servers provided that adequate bandwidth exists between them. There would be no reason to activate it for clients local to the Domino server. Through setup or desktop policies, network compression can be activated selectively for those clients who would benefit most from it.

- In some situations, the least costly solution is to support a subset of users on a distributed platform, like Windows®, with a Domino server local to clients with limited bandwidth. Within Domino's architecture, there is no loss of functionality or any other issues when deploying servers across multiple hardware platforms.

2.4.5 Indexing guidelines

The Update task is loaded at server startup by default and runs continually, checking its work queue for views and folders that require updating. When a view or folder change is recorded in the queue, Update waits approximately 15 minutes by default before updating all view indexes in the database so that the update can include any other database changes made during the 15-minute period. The size of the queue and number of minutes requests are held in the queue can be modified through NOTES.INI parameters Update_Suppression_Limit and Update_Suppression_Time respectively. After updating view indexes in a database, Update then updates all databases that have full-text search indexes set for immediate or hourly updates. When it encounters a corrupted view index or full-text index, Update rebuilds the view index or full-text index in an attempt to correct the problem. This means it deletes the view index or full-text index and rebuilds it.

Updall performs a similar function to Update except that it is invoked as needed and does not continuously process requests from a queue. The remaining discussion will focus on Update.

Some general guidelines for managing the utilization of system resources by Update are as follows:

- If not a hard requirement, disable full-text indexing on a server with the NOTES.INI parameter Update_No_Fulltext=1. By default, full-text indexing is enabled on a server.
- Disallow the building of dynamic indexes by agents searching databases without full-text indexes with the NOTES.INI parameter FT_FLY_INDEX_OFF set to 1.
  Note that the parameter Update_No_Fulltext=1 does not prevent agents from dynamically creating indexes. Experience with production customers and recent testing have shown that dynamic full-text indexing incurs significant CPU overhead. The right-hand two-thirds of Figure 2-3 on page 15 shows the steady-state CPU utilizations for the following simulated workload scenarios: FTI updates, but no searches; no FTIs; and no FTIs with agent searches resulting in dynamic FTIs.

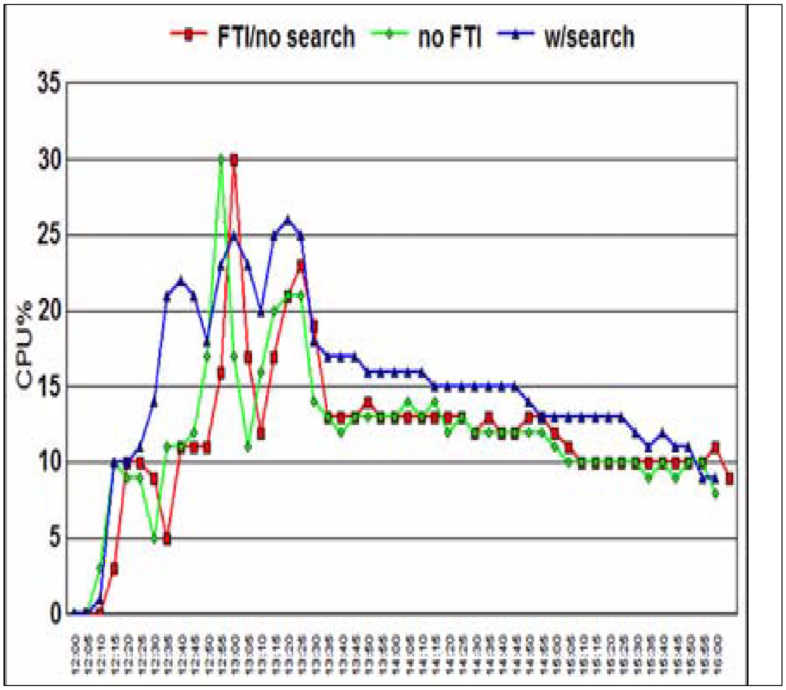


Figure 2-3 CPU utilizations for various workloads

- As you can see from the graph, the penalty for updating the indexes during steady state is much less than the penalty for building them dynamically to satisfy agent searches. If databases are searched on a regular basis by agents, the recommendation is to build the full-text indexes in advance. For more information about this recent testing, see the DeveloperWorks article Best Practices for Large Lotus Notes Mail Files at:

  `http://www.ibm.com/developerworks/lotus/library/notes-mail-files/index.html`

- The rebuild of full-text indexes for large databases can be very CPU intensive and should be scheduled during non-peak processing periods. Otherwise, the Domino server may become CPU constrained and unresponsive to end-user requests.

Domino 7 offers some new tuning options for the Update task with the following NOTES.INI parameters:

- **UPDATE_FULLTEXT_THREAD=1,**
  This parameter causes Update to create a separate thread for full-text index updates which allows view updates to complete in a more timely manner because there is no waiting for the full-text index update to complete. This configuration may use slightly more system resources.
- **UPDATE_IDLE_TIME=** ***# of seconds***
  By default, the delay between each Update operation is 5 seconds. To allow the Update task to use additional system resources, set the delay to less than 5 seconds; to limit its utilization of resources more, set the delay to more than 5 seconds. Always, monitor system resources when making changes.
- For more information about other NOTES.INI parameters which impact the performance of Update, see the section entitled “Indexer Tasks: Update and Updall” in the Lotus Domino Administrator 7 Help database.

2.4.6 Controlling the Agent Manager task

Agents are stand-alone programs that perform a specific task in one or more databases on either Domino servers or workstations. They can be scheduled, triggered-based on events or invoked at will, and run in the foreground or background. Agents can automate complicated database-wide and domain-wide tasks, such as document archiving or updating the Domino Directory. Users can trigger them to perform actions on their private databases, such as processing incoming mail to respond to it, forward it, or file it.

Agents are powerful tools which can consume huge amounts of CPU on Domino servers. For this reason, their CPU utilization needs to be carefully monitored on systems where CPU is constrained or where there are stringent requirements to control CPU growth.

For mail servers, we recommend running only those agents which ship with the standard mail and system database templates. Additional home-grown agents or third-party tools invoking agents must be evaluated separately for their impact on system resources. We strongly discourage giving users the ability to run their own mail agents because this essentially gives them a blank check for system resources. To prohibit users from running private agents, set Enforce_Personal_Agents=1 in the NOTES.INI file.

On application servers, it may be more appropriate to have higher AMgr activity because agents are often an integral part of an application(s). For these environments, agents need to be designed for optimum efficiency, and monitored for changes in resource consumption triggered by updates to them or the database(s) they touch.

The Domino Administration Help database (section entitled “Improving Agent Manager performance”) documents several Server documents and NOTES.INI settings for tuning the Agent Manager in the following areas:

- Controlling how often AMgr runs agents
- Controlling how quickly AMgr queues agents
- Controlling when AMgr runs agents
- Controlling how many concurrent agents are running

Some of these settings will cause agents to run more quickly, but may increase demand on server resources and adversely impact overall system performance.

In general, we recommend that you run administrative agents which process large databases or many databases during non-peak processing periods. Otherwise, Domino servers may have issues with servicing end-user requests.

2.4.7 Limit Domino clustering

Domino clustering provides failover and load balancing capability for Domino users. Multiple replicas of users’ databases are created on two or more Domino servers. Database changes are synchronized across all replicas almost immediately through cluster replication. When a server fails or is too busy, users are directed to other servers with up-to-date replicas of databases that they are trying to access. Disk costs go up because of multiple copies of databases. But more importantly, CPU and storage requirements increase as well. For two-cluster environments (two Domino servers with two sets of replicas), there are twice as many Domino partitions, and potentially twice as many operating system images. Additional DPARs and operating system images incur some CPU overhead and add memory costs. However, the largest portion of the cost is the synchronization of the clustered servers’ databases. CPU costs are directly tied to the number of times the same updates must be applied to multiple databases. In our experience, this overhead could easily be anywhere from 30 to 50%, depending on the database activity.

To reduce the costs of Domino clustering:

- Limit the number of users who are clustered
  Only cluster those users with a real need for failover. Casual users whose day-to-day activities are not dependent on E-mail or other application access typically do not have a need for the high availability which Domino clustering provides.
- Deploy only two-way clusters
  Clusters with more than two servers incur even higher capacity costs because of more redundancy of DPARs, operating system images and database replicas.

Given the stability of the System z9 hardware and operating system environment, Domino for System z9 clustered servers can be deployed on the same hardware footprint. However, we do recommend that cluster mates be deployed on different LPARs or Linux guests to allow for more flexibility in scheduling things like maintenance to the operating system. There is a general recommendation to segregate the cluster replication traffic from the normal user traffic with a private network. On System z9, where clustering is typically implemented on the same hardware server (disaster recovery scenarios with multiple System z9 footprints as the exception), we recommend using something like hipersockets or channel-to-channel communication for the cluster traffic between LPARs on a single hardware server. For more information about setting up a clustering network for Domino z/VM guests, see Chapter 8, “Networking updates” on page 85.

2.4.8 Watch out for mail rules

Starting with Domino 6, mail rules were enhanced to run on the Domino MAIL.BOX and to affect all messages on a server. Mail rules, if not defined carefully, can cause significant CPU overhead. The following best practices should be adhered to in order to minimize the impact of mail rules in either Release 6 or 7:

- Fewer rules are always better.
- Check the logic of your ANDs and ORs, or you will not test for what you may think you are testing.
- Move the most common words or phrases to the beginning of your conditions list. For some formulas, the sooner the logic is successful, the less CPU is used because the searching stops once the correct set of conditions is met.
- Move the most used rules to the top of the Configuration document’s list. The rules are evaluated in the order in which they are listed. There are arrows to change the rules sequence. (Note that all rules are evaluated with one exception-when a delete [or don't accept message] action is encountered. This stops processing of remaining rules.)
- Don't include words that are uncommon. If only one user complains that a particular word needs to be added to the list, set up a rule in the user's personal mail file to search only messages sent to that user, instead of every mail message going to the server.
- Don't search the body of the message unless it is really needed. Searching the body has the most impact on your CPU utilization.
- For more information about mail rules on servers, see the following two DeveloperWorks articles at:

  http://www-128.ibm.com/developerworks/lotus/library/ls-D6ServerMailRules/index.html
  http://www-128.ibm.com/developerworks/lotus/library/nd7features/

2.4.9 Use the extended instead of condensed directory catalog

A directory catalog is an optional directory database that aggregates information from multiple Domino Directories. There are two types of directory catalogs: Extended Directory

Catalogs and condensed Directory Catalogs. An Extended Directory Catalog is larger than a condensed Directory Catalog, but is the recommended directory catalog for server use because it allows faster and more flexible directory lookups and uses less CPU.

The Extended Directory Catalog uses the same design as the Domino Directory, so it includes multiple views that sort names in different ways. Regardless of the format of a name, there's a view in the Extended Directory Catalog that a server can use to quickly find the name. A condensed Directory Catalog has one view used for lookups, which you choose how to sort when you configure it. To look up a name in a condensed Directory Catalog that doesn't correspond to the selected sort order, the server uses the full-text index to search for the name, which takes longer than a view search.

Using an Extended Directory Catalog on servers that route mail is a particular advantage, because a mail server can use views to quickly find an address regardless of the address format. When a mail server uses a condensed Directory Catalog, mail routing can back up if the Router uses the full-text index to look up addresses, for example, some Internet addresses, that don't correspond to the selected sort order.

2.4.10 Limit the impact of SSL

Secure Sockets Layer (SSL) is a security protocol that provides communications privacy and authentication for Domino server tasks that operate over TCP/IP.

It offers these security benefits:

- Data is encrypted to and from clients, so privacy is ensured during transactions.
- An encoded message digest accompanies the data and detects any message tampering.
- The server certificate accompanies data to assure the client that the server identity is authentic.
- The client certificate accompanies data to assure the server that the client identity is authentic. Client authentication is optional and may not be a requirement for your organization

SSL is likely to add significant overhead, especially when deployed for Web clients. To limit the impact to system resources, follow these guidelines:

- Do not deploy SSL for users who access Domino on internally secure networks.
- Deploy SSL only where it is required.
  SSL can be deployed on a system-wide or database by database basis. For more information about how to set up SSL at a Domino server level or for selected databases only, see *Running an SSL connection to a server* in the Domino Administration Help database.
- For Domino 6.5 or later for z/OS, exploit System z9 hardware cryptography to reduce SSL CPU overhead.
  Hardware cryptography is not available to Domino for System z9 running on Linux.
- Deploy a proxy server to off-load some of the CPU associated with SSL.
- Tune the amount of SSL information cached from previous sessions with the NOTES.INI parameter SSL_RESUMABLE_SESSIONS.
  By default, this parameter is set to 50. SSL session resumption greatly improves performance when using SSL by recalling information from a previous successful SSL session negotiation to bypass the most computationally-intensive parts of the SSL session key negotiation. HTTP is the protocol that benefits the most from SSL session resumption, but other Internet protocols may benefit as well. For more information, see the section entitled “SSL session resumption” in the Domino Administration Help database.

2.5 Off-loading mail server workload to Notes clients

Most Domino Notes users access their mail files directly on the Domino server. There is also the less frequently utilized option to access a replica copy of the mail file from a workstation's hard drive rather than from the Domino server. In this scenario, the server is primarily accessed when replicating information to and from the workstation.

There are advantages and disadvantages to using local mail replicas. Some of the advantages are:

- Users can process mail even if disconnected from the server, such as while traveling.
- Users still have access to their mail even if the Domino server is down.
- During peak processing periods on the server, users might have a faster experience while performing common mail tasks on a local replica.
- There might be less of a need to deploy Domino clustering because short periods when servers are unavailable are less noticeable.

  For a detailed discussion of advantages/disadvantages, see the article entitled *Lotus Notes mail: comparing server-based mail and local replica mail* (March 1, 2005) at

  http://www-128.ibm.com/developerworks/lotus/library/serverlocalmail/index.html.

Working from local replicas clearly reduces network traffic to Domino servers and off-loads additional workload, potentially lowering CPU as well as bandwidth requirements. For more sophisticated users who might maintain and search full-text indexes or run personal agents on a daily basis, the potential CPU cycles off-loaded from Domino are even greater. To maximize the benefits of working from local mail files, implement the following recommendations:

- Disable type-ahead addressing on the Domino server in the Configuration Settings document Messaging section.
  If you disable type-ahead addressing on a mail server, users can still use type-ahead addressing to find addresses in their Personal Address Book or a Mobile Directory Catalog. Once users are accustomed to accessing their mail files on workstations, using a Mobile Directory is a logical follow-on step.

  Note: However, keep in mind that confining type-ahead addressing to workstations will save bandwidth and server capacity for users regardless of whether they use a local replica or access mail files directly on the server.

  You can automate setting up a Mobile Directory Catalog on Notes clients by using a Setup policy settings document or a Desktop policy settings document. This process replicates the directory catalog to the client, and adds the directory catalog file name to "Local address books" field in the User Preferences dialog for mail.

- Control the frequency of mail replications between workstations and the Domino server. A replication interval no smaller than every 15 minutes is recommended.
  There have been a few customer situations where CPU requirements increased with local replicas on workstations. This increase was found to be related to the frequency of replication with Domino servers. For a large number of users, a replication interval of 5 minutes or less can drive up CPU utilization substantially.

  The frequency of replications is controlled in the Location document of the Notes client. For Domino 6.5, Policy documents cannot be used to set the appropriate fields. For a work-around to this restriction, see Technote 1106153 ("Can Policies Be Used to Populate Fields in Location Documents or to Create Connection Documents?") in the Lotus Software Knowledge Base. This restriction has been lifted in Domino 7. For more information, see the section entitled “Using policies to assign NOTES.INI or Location document settings to Notes client users” in the Lotus Domino Administrator Help Database.

2.6 Impacts of database size

Some recent tests with large mail files have shown that database size does indeed impact resource consumption. Although all testing was done with mail databases, the observed results and conclusions would also apply to other application databases. For a detailed description of the testing methodology and results, see the DeveloperWorks article “Best Practices for Large Lotus Notes Mail Files” at:

`http://www.ibm.com/developerworks/lotus/library/notes-mail-files/index.html`

It should be noted that these tests were based on simulated workloads. Their results showed certain trends based on specific test environments, which might not totally reflect those seen in production installations. Workload was generated with Server.Load (free tool which ships with the Domino code) and its built-in R6 Mail Routing script. Users were ramped up for some period of time until the desired number of active users was reached, and then workloads were maintained in steady-state for a certain period of time. The ramp-up workload simulated large numbers of users trying to access Domino in a short period of time, which is similar to peak workloads customers might experience early in the morning/after lunch. The steady-state workload showed less variability in resource utilization because the overhead for session initialization and a higher contention for resources had been removed.

Some of the observed results were as follows:

- CPU utilization increases versus mail file sizes were linear, except that during peak times CPU increased at a faster rate. Figure 2-4 shows increases in CPU utilization for mail files ranging from 20 MB to up to 2 GB in size during peak and steady-state periods.

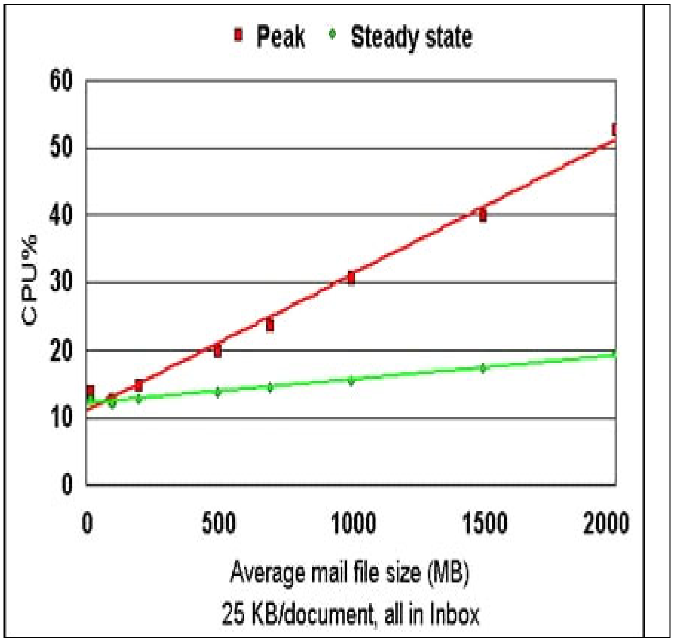


Figure 2-4 Linear increases

Besides the increased CPU demands for session initialization, the sharp increase in ramp-up CPU utilizations were at least partially due to view indexing of the larger Inbox views and the Unread marks processing associated with these views.

- The number of Inbox documents impacted CPU utilization. Again, the increases during ramp-up/peak times were greater than during steady-state. Figure 2-5 shows the CPU differences for users who leave all of their documents in their Inboxes versus users who only keep 25% of their total documents in their Inboxes.

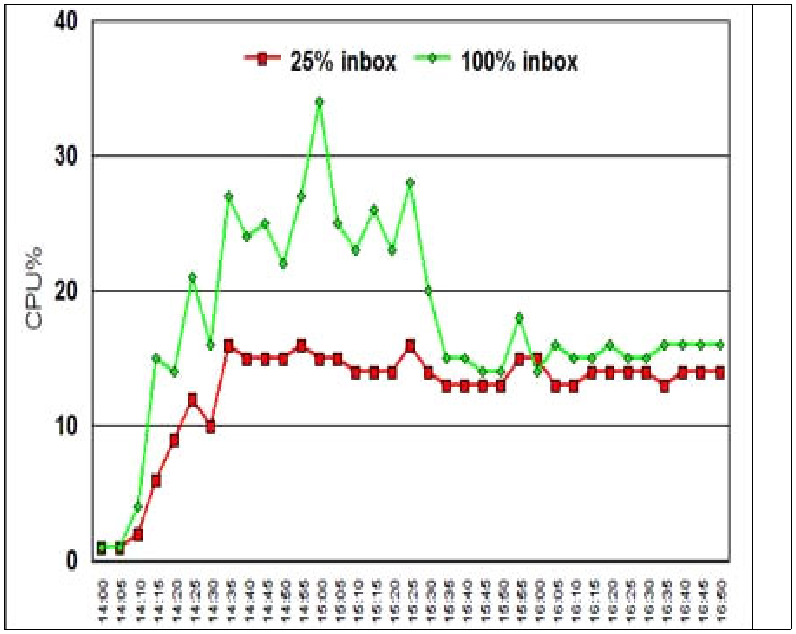


Figure 2-5 CPU utilization by inbox utilization

Several other tests were run, varying the number of documents in the Inbox. From these runs, it was determined that reducing the number of documents in the Inbox improved user response time as well as reduced overall resource consumption, especially when users first accessed their mail files during ramp-up.

- Mail files of the same size used resources differently, depending on the average size and number of documents. Figure 2-6 on page 22 shows the CPU utilization for two groups of users, each with 700 MB mail files, but one group with 28,000 documents (at 25KB each) and the other group with 7000 documents (at 100KB each).

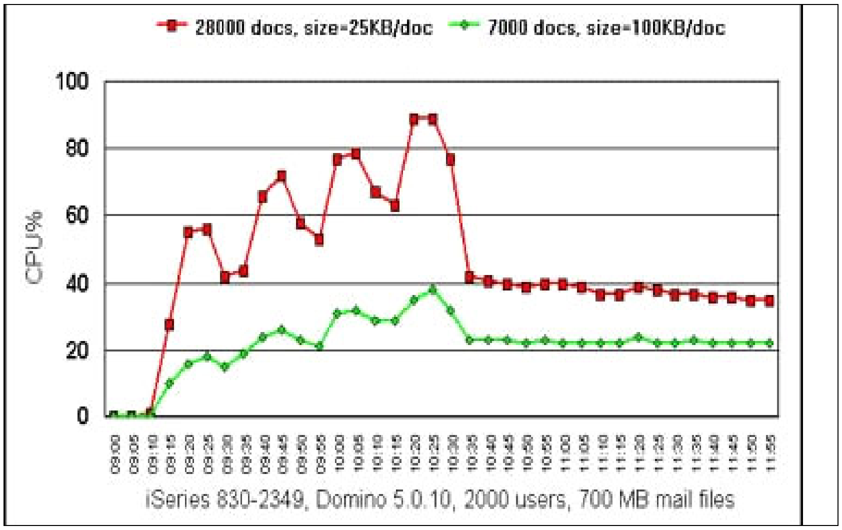


Figure 2-6 CPU utilization by document size

This test indicated that the average size and number of documents in a mail file, especially the Inbox, has a greater impact on CPU utilization than the database size. Users with many small documents use more resources than users with fewer larger documents. But users with large mail files typically have many documents with many views, which could add to processing overhead.

All in all, databases whose size is monitored and minimized show increased performance: database operations require less I/O and fewer CPU resources; view rebuilding and updating is quicker; and memory and disk space allocation is improved. There is the added benefit of reduced processing times for database utilities such as compaction, and backup and restores. Again, these characteristics are not limited to mail files, but are applicable to all Domino applications.

Database sizes can be reduced using the following methods:

- Domino/Notes archiving on the server or client.
- Reducing the number of large attachments sent to many parties.
  A document link to a discussion database is a valid alternative, and there are others.
- The deployment of database quotas.
- The deployment of other third-party tools which prune databases based on some selection criteria.
- Initial application design.
  For more information, see Chapter 3, “Application recommendations” on page 29.

2.7 Always implement

Experience has given us some tips that we can recommend to everyone to implement.

2.7.1 Transaction logging

On System z9, we always recommend Domino transaction logging because it provides better data integrity, point-in-time (forward recovery) of Domino databases, and faster restarts of Domino after an outage. Use transaction logging to:

- Schedule regular backups. Backups based on transaction logs are faster and easier than full database backups that do not use transaction logging.
- Recover from a media failure. If you have a media failure, you can restore the most recent full backup from tape, then use the transaction logs to add the data that was not written to disk. However, you will need a third-party tool which uses Domino's backup and recovery APIs to provide the point-in-time recovery function for databases.
- Recover from a system crash. When the server restarts, it runs through the end of the transaction logs and recovers any writes that were not made to disk at the time of the crash. Logged databases do not require a consistency check.
- Log the database views. You can avoid most view rebuilds.

There are three styles of transaction logging: circular, linear and archival:

- With circular logging, Domino reuses a fixed amount of disk space (up to 4GB) for transaction logs. After the disk space is used up, Domino starts overwriting old transactions, starting with the oldest. When the space fills up, perform a backup on the databases. You may need to do daily backups to capture database changes before they are overwritten, depending on the server activity level. Use circular logging if the size of the log needed between full database backup intervals is less than 4GB.
- Linear logging is like circular logging, except it allows more than 4GB. Use linear logging if the size of the log needed between full database backup intervals is greater than 4GB, and you are not using archive media.
- Archival logging creates log files as needed. It simplifies backup and restoration, and provides online and partial backups. The log files are not overwritten until you archive them. With archived logging, you must have a backup utility to back up the filled log extents so that they are ready if needed. If you do not have a backup utility, the server continues to create log extents, fills up the disk space, and then panics.

Point-in-time recovery of databases requires archival logging. Quick recovery from a system crash can be achieved with either circular or linear logging.

Transaction logging of databases does incur some CPU overhead. But given the benefits of better data integrity and faster restarts, we recommend that you at minimum deploy circular or linear logging. To minimize the cost of transaction logging, follow these guidelines:

- **With Domino 7, enable System z9 hardware compression for Domino transaction logging with the NOTES.INI parameter Translog_HardwareCompress=1**.

  At no extra charge, this System z9 hardware feature can significantly reduce your CPU cost for Domino transaction logging.
- Limit view logging to critical databases and specific views in those databases.
  View logging is transaction logging support for Domino views and folders. All updates to views or folders are recorded in the transaction log for recovery purposes, but at an additional CPU cost and a potential performance degradation in accessing those views. View logging is enabled per database view through Domino Designer®. By default in Domino 6 and later, it is enabled on the $User view of the Domino Directory.
- Disable transaction logging on MAIL.BOXes to avoid routing delays on busy mail servers. Set the NOTES.INI parameter MailBoxDisableTXNLogging to any non-zero value to disable transaction logging on MAIL.BOXes. See Technote 1090763 in the Lotus Software Knowledge Base for more information.
- Set the NOTES.INI parameter Schedule_Disabletxnlogging=1 to disable transaction logging for the Busytime databases clubusy.nsf and busytime.nsf.
  This parameter is recommended to avoid performance issues during rebuilds of Calendar Free Time information. See Technote 1178426.

- Choose the appropriate Runtime/Restart Performance options when configuring Transaction Logging for your servers:
  - Standard (default and recommended) — To record checkpoints regularly.
  - Favor runtime — To record fewer checkpoints. This option requires fewer system resources and improves server run-time performance, but causes more of the log to be applied during restart, which increases the time required to restart.
  - Favor restart recovery time — To record more checkpoints. This option improves restart recovery time because fewer transactions are required for recovery.

For more information about transaction logging, see the Domino Administration Help database.

2.7.2 Fault recovery

You can set up fault recovery to automatically handle server crashes. When the server crashes, it shuts itself down and then restarts automatically, without any administrator intervention. A fatal error such as an internal panic terminates each Domino process and releases all associated resources. The startup script detects the situation and restarts the server. If you are using multiple server partitions and a failure occurs in a single partition, only that partition is terminated and restarted.

It is a best practice to enable fault recovery for Domino servers to minimize server downtime and collect appropriate information for debugging problems. Fault recovery has many configuration options, including running NSD (Notes System Diagnostic) to collect diagnostic information after a fault or crash, and sending an E-mail to a Domino administrator(s) after a restart following a crash. Besides the Domino Administration Help database, several Technotes in the Lotus Software Knowledge Base are available as sources of information about fault recovery:

- 1156547 — NOTES.INI parameters for Fault Recovery
- 1158592 — Fault Recovery Settings/Scenarios on a z/OS/OS390-Based Server
  One of the important points documented in this technote is to set the UNIX® System Services environment variable OS390_DOMINO_SVC_DUMP=1 to obtain a z/OS SVC dump in the event of certain failures.
- 1200076 — NSD Runs Too Long During Fault Recovery on z/OS
  Prior to Domino 6.5.4, there is a recommendation to set FaultRecoveryNSDArgs equal to nomemcheck, regardless of whether Domino runs on z/OS or Linux for System z9. Memcheck is an executable invoked by the NSD utility whenever NSD is run to scan Domino memory segments and report on any errors. There have been some issues in running the memcheck piece of the diagnostics prior to 6.5.4. These issues have been resolved with Domino 6.5.4 and above.

2.7.3 Multiple MAIL.BOXes

You can improve performance significantly by creating multiple MAIL.BOX databases on a server. Using multiple MAIL.BOX databases removes contention for a single MAIL.BOX, allows multiple concurrent processes to act on messages, and increases server throughput. While reading one MAIL.BOX, the Router marks the database “in use” so other server threads trying to deposit mail move to the next MAIL.BOX. As a further benefit, having multiple MAIL.BOX databases provides failover in the event that one MAIL.BOX becomes corrupted.

2.8 Recommendations for NOTES.INI parameters

NSF_Buffer_Pool_Size_MB=*number of megabytes*
The NSF buffer pool is a section of memory dedicated to buffering I/O transfers between Domino and disk storage. This parameter is used to set the maximum size in MB of the NSF buffer pool. On System z9, Domino is limited to two GB of virtual storage which means that you may run into issues if memory is over-allocated for the NSF buffer pool. We recommend that you never allow it to exceed 400 MB.

NSF_DbCache_Maxentries=number of databases
The Domino Dbcache is used to store information about Domino databases so that they can be more quickly opened during subsequent accesses. This NOTES.INI setting sets the maximum number of databases stored in the database cache (if enabled). For short intervals, Domino stores up to 1.5 times the number entered for this setting. If there is sufficient memory on your system, you can increase the maximum number of databases held in cache to improve end-user performance. To determine the appropriate Maxentries value, monitor the Database.DbCache.Hits statistic on your server. This statistic indicates the number of times a database open request was satisfied by finding it in cache, with high values indicating that the cache is working effectively. If the ratio of Database.DbCache.Hits to Database.DbCache.InitialDbOpens is low, then consider increasing the value of NSF_DbCache_Maxentries.

IOCP_DISABLE_ASYNC_NOTIFICATION=1 (see Technote 1087595)
The Domino mail server by default checks every connected user once per minute for new mail and then notifies those users with new mail. Setting the above parameter to '1' disables this functionality, reducing CPU utilization. The frequency of the check may be changed to some value greater than one minute with the NOTES.INI parameter IOCP_Async_Notification_Poll_Time. However, we highly recommend that you disable new mail notifications from the Domino server because of the CPU implications. This function should not be confused with client polling for new mail, which is set in the User Preferences of the Notes client.

Disable_BCC_Group_Expansion=1 (see Technote 1089346)

When a large group is bcc'd in a mail message, it causes a performance degradation on the Domino mail server delivering the mail. The source of this problem are the lookups being performed to expand the group. To prohibit expansion of a large bcc'd group, set this parameter.

Debug_ThreadID=1

With this parameter set, process and thread information useful for debugging problems is made available. Given that CPU overhead is negligible, this is one of the few debugging parameters that we encourage you to always leave enabled.

MinNewMailPoll=*number of minutes*

This parameter determines how often workstations can contact the server to see if new mail has arrived for the user. It overrides the user's selection in the Mail Setup dialog box of User Preferences. You can increase the mail polling interval if there are a large number of mail users on your server, and you want to prevent frequent polling from affecting server performance. We recommend polling intervals of 10 minutes or more.

View_Rebuild_Dir=*path*

When Domino rebuilds views it may generate temporary files to sort the data in order to rapidly update the views; Domino deletes these files after rebuilding the views. By default, these temporary files are located in the Domino Data directory. Putting the temporary folder in a different directory and filesystem distributes disk I/O and ensures that there is enough space to rebuild views. If Domino estimates that there's not enough space available in the temporary folder to rebuild a specific view, Domino uses a slower method to rebuild the view and logs this message to the Miscellaneous Events view of the log file (LOG.NSF).

Server_Session_Timeout=*number of minutes*
There is no default for this parameter, but in the absence of the setting, Domino terminates a session connection after 240 minutes of inactivity (four hours). We recommend that you set a value of somewhere between 30 and 60 minutes. This parameter should never be set for less than 30 minutes.

Server_Pool_Tasks=*number of threads*
Server_Pool_Tasks is used to set the size of the thread pool for NRPC client connections on a Domino server. By default, the pool is set to 100 on z/OS and 40 on Linux for System z9. These defaults should not be adjusted unless end user response time is slow. However, typically, these values do not need adjustment. Domino servers where the thread pools are frequently maxed out should first be investigated for resource constraints prior to making any adjustments to the thread pool value.

Server_Max_Concurrent_Trans=*number of transactions*
This parameter sets the limit for the number of concurrently scheduled transactions on a server. Do not change this parameter unless advised to do so by technical support. By default, there is no limit for concurrently scheduled transactions on System z9.

Server_MaxSessions=*number of sessions*
With this parameter, the maximum number of sessions that can run concurrently on the server may be specified. On System z9, the recommendation is to allow the maximum to default.

2.9 Monitor your Domino servers

We recommend that you define a set of monitoring and maintenance activities for your Domino servers to be performed on a daily, weekly, and quarterly basis. Sections 2.9.1, “Daily monitoring” on page 26 to 2.9.3, “Quarterly monitoring” on page 27 contain sample lists of activities. The primary focus of the daily activities is to assure high availability and good response times for end-users by taking a proactive approach to discovering issues before they turn into serious problems. The weekly list contains some maintenance activities which customers typically perform at this frequency. The quarterly activities list revolves around long-term planning for software upgrades, capacity upgrades to accommodate workload growth, and procedural changes/improvements. These sample lists are in no way all-inclusive; they are intended to be a starting point for defining your monitoring activities. Your particular environment will dictate which activities need to be implemented at what frequency.

2.9.1 Daily monitoring

- **Check server availability and accessibility**
  - Manually or through a software probe, such as:
    - ITMMC (IBM Tivoli® Monitoring for Messaging and Collaboration) and other 3rd party vendors
  - Check if the appropriate Domino tasks are running
    - Are Domino utilities and backups completing in expected windows?

- **Review Domino logs for problems**

  Check for process starts and stops, failed processes or error conditions, DB corruption, network traffic, agents, security. SMTP, etc.

  - Use Domino Log Analysis to facilitate searching against LOG.nsf
  - Implement a script to review console logs
    - Noteslog on z/OS
    - On Linux, generate a console log using instructions from Technotes 1201402 and 1085072
- **Review the Domino statistics database**
  - Use alarms and events to facilitate review
- **Check filesystem size and monitor disk space**
  - Implement a filesystem-full alert on z/OS (fsfull command)
  - Use cron on Linux to mail the output of the *df -h* command to appropriate individuals
  - ITMMC is an option
- **Check high-profile databases (such as Domino Directory)**
  - Did it replicate? Any replication conflicts?
  - Any other issues with Domino utilities?
- **Check for and release dead mail on mail servers**
- **Create FTIs for databases with agents driving dynamic FTIs**

2.9.2 Weekly monitoring

- **Run compact on system, mail, and application databases**
- **Run full backups (needs may dictate a more frequent schedule)**
- **Back up LOG.nsf (might do more frequently)**
- **When using Domino clustering**
  - Run Cluster Analysis to uncover and resolve problems
  - Check clustering queue statistics and adjust the clustering configuration if necessary (See the Lotus Domino Administration Help database)
- **Check for inactive databases**
- **Recycle the z/OS Noteslog**

2.9.3 Quarterly monitoring

- **Monitor system growth**
  - Look at CPU usage / processor / LPAR/ DPAR
  - Using SMF, SAR and Statrep data
- **Monitor Domino growth**
  - Number of transactions / user / hour
  - Number of mail msgs / user / day
  - Number of documents added, updated, deleted / application DB / day or some other application metric

  - Mail and application file size growth
- **Review impact of scheduled changes**
- **Review configuration**
  - Any debug parameters in NOTES.INI
  - Extended logging settings
  - OS Traces and SLIP Traps
- **Review Domino outages**
- **Review maintenance procedures**
- **Review fixlists and plan Domino upgrades accordingly**
- **Review operating system maintenance and plan Domino testing accordingly**

3

Application recommendations

When deploying applications, two important areas of consideration are capacity requirements and user response times. This chapter includes performance tips for dealing with databases and applications.

3.1 General guidelines for better performance

From a capacity standpoint, the primary recommendations for application deployment are a simple design and control of database growth. In general, whenever you add complexity to a solution, you also add to the solution's cost. Ideally, a Domino application should be contained in a single database which resides on one server. That is not to say that there are not valid reasons to deploy applications with multiple databases across more than one server. However, centralization does reduce cost and improves reliability. Fewer servers mean less complex server topologies with fewer server-to-server activities, less redundant data and less replication.

The benefits of controlling database size are that less I/O and CPU resources are required, view rebuilds and updates are faster, and memory and disk allocations are reduced. View processing has a major impact on user response times. Limiting the size of views and consolidating a large number of views to fewer views can greatly improve response times. Indexing views less frequently can also improve response times. For more information about the impacts of database size and view processing, see 2.4.5, "Indexing guidelines" on page 14 and 2.6, "Impacts of database size" on page 20.

3.2 Optimize databases through properties

Properly setting database properties can improve the performance of an active database. Setting database performance properties on many databases or on one large active database can also improve server performance. In addition, some property settings also help reduce the size of databases.

The following common database properties can affect performance and capacity:

- Don't maintain unread marks
- Don't overwrite free space
- Maintain LastAccessed property
- Don't support specialized response hierarchy
- Use LZ1 compression for attachments

These properties can be adjusted from the Database Properties dialog box under the Advanced tab.

Don't maintain unread marks

By default, all read and unread documents are tracked in a database. This is especially useful information to users for certain databases, such as mail or discussion databases. However, there is a performance impact to maintain read and unread marks. When a user first accesses a database on a server, there is a delay because Domino needs to read through the Unread Marks table to determine which documents to display as read/unread for the user. The more documents, the longer the delay. If the read/unread marks are not an important feature of a specific Domino database/application, then the "Don't maintain unread marks" property should be selected in Database Properties for that database.

Don't overwrite free space

When data is deleted from databases, Domino, by default, overwrites the deleted data on disk with a pattern. This pattern prevents an unauthorized user from using a utility to access the data. This overwriting increases disk I/O and can affect database performance. If security isn't an issue for a database or data is being constantly overwritten on a high-activity database, such as MAIL.BOX, then this property should be selected to prevent unnecessary processing and I/O.

Maintain LastAccessed property

If this property is selected, Domino tracks the date when a document was last accessed, either read or modified. By default, Domino only tracks when a document was last modified. For optimal application performance, keep this option deselected. However, keep in mind that this property may be appropriate in certain circumstances, such as when a document archiving solution is implemented which bases its decision to archive on when a document was last referenced, which could mean read/modified. The Maintain LastAccessed property does not apply to Web applications since it ignores Web browser reads.

Don’t support specialized response hierarchy

By default, every document stores information that associates it with a parent or response document. Only the functions @AllChildren and @AllDescendants, which are frequently used in view selection or replication formulas, use this stored information. Maintaining this information has a significant cost in processing cycles. Disable the response hierarchy information in databases that don't use these @functions by selecting the database property “Don't support specialized response hierarchy”.

Use LZ1 compression for attachments

In Lotus Domino Designer, you can choose to compress attachments using the new with Domino 6 LZ1 algorithm instead of the Huffman algorithm. Because LZ1 compression can be performed quickly and efficiently, it is favored over the Huffman method. However, note the caution in the box below.

Caution: If you are working in an environment that uses different versions of client and server software and you choose this option, attachments may be automatically recompressed on the server using the Huffman method. For example, if the LZ1 compression option is set on a Domino 6 database, a Notes 5 client will be unable to process the compressed data. The Lotus Domino 6 server converts LZ1 to Huffman compression for the Notes 5 client. This recompression of attachments can add significant processing overhead. For best performance, use LZ1 in primarily Domino/Notes 6 or more recent environments.

3.3 Performance considerations for database views

Private views or folders

A database ACL can be set up to allow users to create private views/folders and store them on the Domino server where the database resides. Many views, especially large ones, will increase processing requirements due to additional view indexing.

Time/date sensitive views

A time/date sensitive view is one which contains @Now or @Today in the view selection formula or in any column formula. Prior to Domino 6, these views were rebuilt every time the Indexer/Update task ran because by definition the views were out-of date. We saw many cases where there was significant CPU overhead and unacceptable end-user response times stemming from time/date sensitive views.

In Domino 6 and later releases, time/date sensitive views are ignored by the Update task. This means that there is no overhead except in the case of user interaction with the view. Also, in Domino 6 or later, you can set the indexing options for a time/date sensitive view to “manual”, which means that the view will open immediately for users without any refreshes, and the view is only refreshed when the user presses PF9 or clicks on the blue reload arrow.

When @Now or @Today are used in selection formulas, they typically limit the size of the view collection. This reduces the time to refresh and the time to rebuild the view. When @Now or @Today are used in a column formula, they do not similarly restrict the view index size. Therefore, although in theory there is no distinction between using @Now or @Today in selection formulas versus column formulas, in practice it is often the case that the worst performers are those that have these functions in their column formulas. For more information about time/date sensitive views, see the Lotus DeveloperWorks article entitled, “Troubleshooting Application Performance: Part 1: Troubleshooting techniques and code tips”.

For suggestions on alternative coding techniques for time/date sensitive views, see the Lotus DeveloperWorks article entitled “Application Performance Tuning, Part 1”.

Column sorting

Column sorting allows users to sort view columns in ascending, descending or both orders. Each sorting arrow on a column increases the size of the view index and the time it takes to refresh it. To improve performance, the recommendation is to limit the number of sorting arrows. For example, adding two sorting arrows to a view increases the index size and time to process it threefold; adding four sorting arrows increases the index size and time to process five-fold, and so on.

Reader names fields

Domino designers can use Reader Names fields to control who has read access to documents. For views within large databases with many documents, Reader Names fields can have adverse effects because of the document by document processing which takes place to determine whether to display a document within a view to a particular user. The article “Application Performance Tuning, Part 1” documents a few alternatives to avoid this problem.

3.4 Performance tips when coding applications

Use @Db formulas sparingly

View/folder lookup functions like @DbLookup and @DbColumn carry the overhead of searching and potentially re-indexing internal/external views before returning the required results. They should be used sparingly or replaced with the LotusScript methods which are generally faster and perform error checking. If data is referenced more than once, then consider using the “Cache” option on @Db formulas or store the results of lookups in temporary variables. For more information about LotusScript methods to retrieve a collection of documents, see the Lotus DeveloperWorks article “Application Performance Tuning, Part 2”.

Limit computations within documents

The more computations executed in a document, the slower the performance. Typically, whenever documents are opened in read mode, a certain set of computations must occur; when opened in edit mode, another set of computations must take place. The trick is to code the application such that the number of computations are minimized for the majority of accesses.

You can set field values to refresh automatically by selecting the “Automatically refresh field” option in the Form Properties box. With this option set, every time a user moves the cursor through the form fields, all of the previous fields are recomputed. The primary purpose of this option is to check for Input Translation and Input Validation formulas. A better option is to select the “Refresh fields on keyword change” option in Field Properties. It offers better performance because the document is only refreshed when the value changes in the field where this option is specified.

Another recommendation is to use the Computed when composed field type whenever possible instead of the Computed field. Computed fields are computed every time the document is edited, refreshed or saved, which could be very costly if there are many fields. The Computed when composed field type calculates the values for a field only when the user creates the document.

Performance considerations with HTTP and images

There have been reported instances where a small number of concurrent users experience very slow response times when accessing a large document from a Domino Web application with many inline images in the document. The size of a document/form can significantly slow down response times. Additionally, a large number of embedded images such as in the example just cited can cause the HTTP process memory to grow very rapidly. It is a best practice to avoid embedding images directly on a form by using one of the following options:

- Use an Image Resource (<img src> tag) that accesses the image as an attachment to a Notes document, which decreases the size for the document's form and allows images to be stored in a different document.
- Access the images directly from the file system, which allows for the fastest access.

For more information about using images in Domino Web applications, see Technotes 1096054 and 1166257 in the Lotus Software Knowledge Base.

Other recommendations when designing applications

The Lotus DeveloperWorks article "Application Performance Tuning, Part 2" lists additional programming tips for reducing resource utilization and improving end-user response times for your applications:

- Pay attention to the number of events in a form and don't "overcode".
- Shared elements are slightly worse performers, but they compensate for poor performance when used in multiple places.
- Make sure that error checking stops when the error is encountered.
- Large subforms are poor performers.
- Use fewer fields.
- Use view.Autoupdate=False to prevent a view from refreshing.
- Use the StampAll method to modify a large collection of documents at once.
- The ForAll statement is the fastest way to iterate through a loop.
- Fixed arrays are better performers than dynamic arrays.

3.4.1 For more information

For more information about application best practices, see these publications:

- *Performance Considerations for Domino Applications*, SG24-5602
- "Application Performance Tuning, Part 1"

  `http://www-128.ibm.com/developerworks/lotus/library/ls-AppPerfpt1/index.html`
- "Application Performance Tuning, Part 2"

  `http://www-128.ibm.com/developerworks/lotus/library/ls-AppPerfPt2/index.html`

- "Troubleshooting Application Performance, Part 1: Troubleshooting Techniques And Code Tips"

  http://www-128.ibm.com/developerworks/lotus/library/app-troubleshooting1/index.html
- Lotus Domino Designer 7 Help database
- Lotus Domino Designer 6.5.1 Help database

4

Backup/recovery best practices

Domino e-mail data and other application data are quickly becoming more and more mission critical, which means that most customers require a backup and recovery solution that is suited to their needs and that performs well within their Domino environment.

In this chapter, we discuss:

- Understanding your Domino backup/recovery needs
- Which tools to use
- Using the IBM Tivoli Storage Manager product for Domino backup/recovery:
  - Execution and scaling recommendations
  - Capacity/performance considerations and recommendations
  - Other operational considerations

4.1 Understand your Domino backup/recovery needs

To best determine Domino backup/recovery requirements, consider Domino data characteristics, such as criticality, rate of change and specific data types (text, NSF, binaries). The type of tool used is dependent on these characteristics as well as whether Domino can be off-line during the backup/restore process. The following examples describe some possible scenarios in a Lotus Domino for System z9 environment:

- **Discardable data:** The risk/consequences of losing this particular type of data is acceptable. For example, mail data for store clerks in a retail business might not be worth the cost of backup. In this case, it is perfectly acceptable not to take backups. The recovery process for an individual corrupted database might be to delete and recreate the database. However, in a disaster recovery situation, where many/all databases must be rebuilt, this solution could be very time-consuming. To avoid recreating a large number of databases, an acceptable solution might be to take infrequent, but regular volume dumps with an appropriate tool while Domino is down. Through volume restores, all the mailboxes could quickly be repopulated with some previous version of the mail data.
- **Static data:** The Lotus Domino binary directory (/usr/lpp/lotus or /opt/lotus) is an example of data which is relatively, although not totally, static. Full periodic backups of the file system/directory containing this data would be appropriate.
- **Non-mission-critical data**: Data is not mission-critical when the needs of the business make it acceptable to recover the data to a given point in time in the not too distant past. This requirement could be accomplished through periodic backups and then a restore of the last backup when needed. Any updates that may have taken place after the backup would be lost, but this is acceptable by definition. If Domino requires 24X7 availability, then there is a need for online backup/recovery. Among the tools of choice for online backups is IBM Tivoli Storage Manager (TSM) with its clients: IBM Tivoli Storage Manager for Mail - Data Protection for Lotus Domino (TDP) client or the Backup-Archive client.
- **Mission-critical data:** This category involves databases where the needs of the business dictate that the data must be recovered to its state immediately preceding a failure. The only way to accomplish this type of recovery is with periodic backups combined with some journal-based forward recovery software. TSM with TDP in conjunction with Domino archival logging is a solution which provides this capability.

4.2 Which tools should you use?

Different tools are appropriate for different backup/recovery needs:

- Use traditional System z9 tools to back up data which is non-mission-critical or static. This type of backup is typically done on the volume or file system level. To guarantee Domino database consistency and integrity, the Domino server must be down during all or part of the backup. Depending on the size of the file systems and number of volumes being backed up, this type of backup could require an extended Domino outage and should be executed on an infrequent basis.
- Use tools such as TSM with either its Backup-Archive or TDP client to back up files while the Domino server is running. The Backup-Archive client can be used to back up all UNIX/Linux files, including Domino databases, text files and binaries. But keep in mind that only TDP supports online backups of Domino (R5 and later) databases and templates **with full integrity**. The Backup-Archive client can back up Domino databases, but may create “fuzzy” backups which would require Fixup when they are restored. Restores from TDP never require Fixups. TDP can be used for all full (selective) backups with or without Domino transaction logging enabled on the Domino server.

Domino supports three types of transaction logging: circular, linear and archival. Circular and linear logging provide faster restarts of Domino servers because information about in-flight transactions and updates to databases is available and can be applied/backed out at restart time. Archival logging also provides faster restarts, but in addition allows tools like TDP to recover databases to a specific point-in-time by applying Domino transaction logs to a full backup of the database. TDP has the capability to restore the most current version of a Domino database as well as past versions, provided that the appropriate transaction logs are still available. For more information about Domino transaction logging, see the Lotus Domino Administrator Help database.

TDP will not back up the Domino executables or Domino text files like the NOTES.INI file. Use the Backup-Archive client to back up these files.

- Other third-party vendors provide tools for backup and restore similar in function to the Tivoli tools described above.
- For more information about IBM tools, see:
  - *Lotus Domino for S/390 — Running a Large Domino System*, SG24-5984
  - *IBM Lotus Domino 6.5 for Linux on System z9 Implementation*, SG24-7021
  - *IBM Tivoli Storage Management Concepts*, SG24-4877
  - *IBM Tivoli Storage Manager Version 5.3 Technical Guide*, SG24-6638

4.3 Using TSM and its clients with Domino

The Tivoli Storage Manager has components for the Domino environment.

4.3.1 Execution and scaling recommendations

- Both the Backup-Archive (to some degree) and TDP clients support backups of databases while Domino is up and running. But backing up large amounts of data can be CPU intensive and should be scheduled during off-shift hours. As throughput increases, CPU increases due to the overhead of the TSM clients, TCP/IP and file system access.
- To avoid processing conflicts on the same databases, run TDP and Compact at different times. However, Updall and TDP backups can be run during the same window without any harmful effects.
  - If archival logging is enabled, certain kinds of compacts must complete before backups are run. A copy-style compact or compact to reduce file size will change the DBIID (Database Instance ID) of databases. The DBIID is the identifier used to match full backups of databases with their appropriate transaction logs for point-in-time recovery. Backups should be run as soon as possible after compacts which result in changes to the DBIID. For a discussion of which compacts change DBIIDs, see the Lotus Domino Administration Help database.
  - Customers without archival transaction logging enabled typically run nightly full/selective backups of all Domino databases using TDP. Customers with archival transaction logging enabled run less frequent selective backups of all databases – one or more times per week.
  - IBM Poughkeepsie (IBM's premier production location for Domino servers) has archival logging enabled and uses TSM with TDP to back up all Domino for System z9 databases. Selective backups are run three times during the week to reduce the number of logs which need to be restored when recovering databases. Backups of the Domino binaries and text files are run nightly using the Backup-Archive client.

- To achieve scalability, run multiple instances of the client against different Domino directories. Plan your Domino directory structure to facilitate running multiple clients. Otherwise, individual TSM clients must be set up to back up databases using a different, less convenient selection criterion, such as database names beginning with certain letters of the alphabet.
- Since newer tape drives are very fast, direct backup to tape usually achieves the best throughput for a large number of databases. When using tape, each client instance requires a dedicated tape drive.

  If the number of tape drives is an issue, tape drive requirements can be reduced by adjusting the backup schedule. For example, if selective backups are scheduled twice a week, consider extending the backup schedule to four times a week. Back up each file twice a week, but only half the number of files on each scheduled night.
- If backups run into prime shift hours, run more instances of the client to shorten the backup window. Some tuning of the TSM server and its clients may also be necessary.

4.3.2 Capacity and performance considerations and recommendations

- For Domino on System z9, implement the TSM server and its clients on the same hardware server to avoid network bottlenecks/delays and other problems during backup. If the network is faulty (e.g., Network packets are lost), there is overhead when clients resend data to the TSM server.

  With a 64-bit operating system, such as z/OS or SLES9 (supported by Domino 7), Domino and TSM may be installed in the same operating system image. With a 31-bit Linux, such as SLES8 (required by Domino 6.5), implement the TSM server in a separate LPAR/VM guest from Domino. Use CTC communication, hipersockets or VM guest LANs as appropriate to improve network throughput.
- Ensure that the TSM server is not overloaded. If backups are starting to slow down, there could be a CPU problem stemming from too much workload being directed to a TSM server. Check the LPAR/guest where the server resides for high CPU utilization. Also, look at the utilization of the disks that hold the TSM server database. In the event that the network has not been properly tuned, the client may overrun the server with packets. The server will react by pacing the client down to a slower rate.
- Implement the best-performing file system to increase throughput rates for backups and restores. On z/OS, implement zFS instead of HFS. On Linux, implement the EXT3 file system.
  - Use FICON® rather than ESCON® connections to further improve throughput.
  - If still using HFS on z/OS, periodically defragment Domino HFS files. Throughputs for backups may be impacted as an HFS becomes more fragmented over a period of 6 to 9 months. Use the copytree utility or UNIX copy commands to rebuild the fragmented HFS into a new HFS.
- Back up Domino files directly to tape for the best throughput rates.
  - Do not use a VTS (Virtual Tape Server), as it decreases throughput rates.
  - Tip: IBM Poughkeepsie backs up Domino for System z9 databases directly to tape.
- Turn off compression for the TSM clients. Compression should be enabled on the tape drive rather than the client to save on memory and CPU cycles, and possibly improve throughput rates.
- If running with older hardware, consider upgrading to more recent technology. For example, ESS disk systems can provide better throughput than older disk technologies. Newer tape drives may also enhance throughput rates, although some tape drives may

support throughput rates which exceed those observed while running TDP. TDP cannot exceed the throughput capacity of the tape drive.

  - Place the TSM server database and storage volumes on the fastest I/O disk available. Also provide enough separate physical disks to minimize I/O contention.
  - Most System z9 customers, including IBM Poughkeepsie, are using an ESS subsystem with 3390 mod-9 or larger volumes. TDP throughput rates ranging from 4 to 9 Mb have been observed. However, throughput rates may vary widely, depending on available hardware and hardware/software configurations. Lower throughput rates are often tied to tape drive contention or TSM server contention, such as TSM internal housekeeping.

- Define the TSM server to its clients with its IP address rather than host name to avoid DNS lookups. This again will improve throughput rates.
- Fine-tune TCP/IP-related parameters defined to the TSM server and clients.
  - Usually a TCPWINDOWSIZE of 64 and TCPBUFFSIZE of 32 are best. However, values for these parameters are also dependent on your particular implementation. For more information about tuning TSM and its clients, see the *IBM Tivoli Storage Manager Performance Tuning Guide*, SC32-9101.
- If transaction logging is enabled (this is recommended):
  - Store the transaction log on a separate device with a dedicated controller.

    For archival logging, there is no limit for the size of the transaction log, other than the amount of disk space available. However, we do recommend that you allocate at least 4 GB (or more) for this style of logging. The optimal archival log size is dependent on the amount of activity on your Domino deployment and how frequently you archive log extents.

    For circular logging, there is a limit of 4 GB for the log size. For linear logging there is no size limitation other than available disk space. The “Maximum log space” field on the Transactional Logging tab of the Server document can be used to limit the log size for both circular and linear logging styles. It is not necessary to specify the log size with linear logging. But if specified, be aware that Domino will format the number of extents that will fit in the space that is specified. For example, if you specify 200 GB for the logs, you will have to wait for the 200 GB of logs to format at start up time.
  - The general recommendation is to specify the “Standard” option for the Runtime/Restart Performance option in the Transaction Logging section of the Server Document. This field controls how often Domino records a recovery checkpoint in the transaction log, which affects server performance.

    However, if your CPU is constrained during prime shift, you might consider choosing the “Favor runtime” option. This will reduce the number of recovery checkpoints, which will result in lower CPU requirements but more lengthy server restarts.

    There is also an option called “Favor restart recovery time”.
  - With archival logging, archive transaction logs to disk first and then later stage to tape. The time required to recover files significantly increases when the necessary archived logs must be recovered from tape. By keeping some subset of archived logs on disk, requests to recover files in the not-too-distant past can be more quickly processed.

    Ideally, the staging from disk to tape should also be scheduled during non-peak processing times to avoid resource contention.
  - With archival logging, schedule full backups frequently enough to allow for database restores within an acceptable time limit. Full backups which are scheduled too infrequently may require restoring a large number of archived logs, which is very time-consuming and CPU intensive when recovering files.

  IBM Poughkeepsie archives log extents to disk first and then to tape. Due to the high volumes of data being recorded on the transaction logs, they have elected to take full backups three times a week to limit the number of logs which need to be restored to recover a database.

- For point-in-time restores:
  - To avoid contention with Domino and decrease elapsed times for restores, use separate file systems/disk from the file systems/disk for the live transaction log when restoring databases and archived logs. This can be accomplished through a restore to an alternate Domino server. See Appendix B in *Data Protection for Lotus Domino for UNIX and OS/400 Installation and User's Guide*, SC32-9056, for instructions on setting up an alternate Domino server for database restores.
  - Restoring databases can be a very CPU-intensive process, especially if many transaction logs must be restored and applied. If your system is CPU-constrained, consider performing the restores during non-peak periods during the day.
- For performance issues, examine the TSM server activity log, client error file and the appropriate system logs for your operating system. In the TSM activity log, messages are logged for client sessions, scheduled operations, automated server processes and any errors that may occur.

4.4 Other operational considerations

- With archival logging, schedule incremental (full) backups, at a minimum, nightly for those databases whose DBIID has changed.

  Ideally, an incremental backup of a database should immediately follow any Fixup or copy-style Compact or Compact to reduce file size against that database because the database acquires a new DBIID. Otherwise, there will be a period of time when there is no full backup matching current transaction log extents.
- Archive transaction log extents at a frequency so that the transaction log file system does not fill up.

  IBM Poughkeepsie archives its logs hourly with a threshold high value of 70% and a low value of 0%. The high value identifies the point at which log archiving should begin. If the current occupancy of the transaction log equals or exceeds the value for this parameter, then eligible log extent files are archived until the occupancy falls below the low value.

5

Domino for System z technical support overview and recommendations

Efficient and repeatable execution of technical support procedures and policies play a major role in a stable and reliable Domino server environment. System z9 customers have a unique advantage in the experience gained over time on how to properly support applications in their System z9 environment. In addition, these customers have learned how to fully leverage the Technical Support offerings from IBM. This section contains information presented by Domino for System z9 Level 2 Support in July, 2005. It also includes recommendations of production customer environments to provide high quality support for Domino. The majority of this information applies to both the z/OS and Linux environments. Operating system specifics are highlighted for the reader.

5.1 Tips for reporting Domino problems

Many readers have gone through the process of opening a Domino for System z9 Problem Management Report (PMR) with Lotus Technical Support. Most of the tips here will seem obvious. However, a periodic review of the basics is helpful to even the most experienced customer to ensure efficient procedures are in place and followed. The detailed steps involved in opening a PMR will be skipped since that process varies by geography. Here is a simple checklist to make this process work well:

- Stress that you are running the Lotus Domino server on System z9. In the early days of Domino on System z9, the Domino server would be confused with the Lotus Domino Go Web server. If you encounter this problem today, ask to speak to a manager on duty to ensure your issue is resolved quickly.
- Specify whether this specific PMR relates to Domino on z/OS or Linux for System z9.
- Define the problem with as much specificity and detail as possible.
- Document any known changes to the compute environment (hardware, applications, storage, etc.)
- Capture all debugging data immediately. The use of automation is strongly recommended to capture all needed files.
- Be sure to mention if you have previously opened PMRs of a similar nature to this PMR. Indicate if the previous PMR was on the same server or not, and what resolution, if any, was provided.

5.2 Tips for defining Domino problems

Three keys to speeding up Domino problem determination is to accurately and completely define what problem you are experiencing, collect data accordingly, and sharing all of this information with Level 2 Support. The next section notes how data collection varies by problem type. Here is a checklist of questions to ask as you define and describe a Domino server problem:

1. How frequently is the problem experienced? Many Domino problems are sporadic in frequency. They may be difficult to recreate, either by the customer or by IBM Support. The following points will help you determine if links exist between this problem and previous experiences.
2. Is the server running and functioning during the problem?
3. Is there limited functionality of the server, clients, etc. while experiencing the problem?
4. What applications were running at the time of the problem? This includes Domino tasks as defined in notes.ini, other workloads in the same LPAR (z/OS and Linux) or Linux virtual server (when running under z/VM).
5. What has changed in the environment in the recent past that could contribute to the problem? Note this can also include changes in user workload, work patterns, etc.
6. What actions were required to recover the Domino server?

5.3 First time data collection

This section contains the recommendations from Domino System z9 Level 2 Support when collecting data the first time a new problem occurs. Three different problem types will be covered, crashes, hangs, and performance related issues. Of course, specific recommendations and requests by a support analyst working your problem should be followed. However, this is a good starting point for data collection. Again, customers are encouraged to exploit the strong automation capabilities and experiences of the System z9 environment to make this process dependable and repeatable.

5.3.1 Domino server crashes

Data to be collected for a server crash

- NSDs
- Domino Console / Domino log
- SVCDUMP or core dump
- System Logs
- EREP (Error Reporting Data Sets) (z/OS only)—EREP processes the error records from your operating system to produce formatted reports. These EREP reports can show the status of the entire installation, an I/O subsystem, or an individual device depending upon which report you request.
- Debug files

Recommended Settings

- FaultRecovery=1 in Notes.ini—The FaultRecovery environment variable specifies whether automatic recovery is on or off.
- debug_enable_core=1 (Linux only)
- For optimal FFDC (First Failure Data Capture) enable CORE dumps (core.xxxx) with notes.ini variable DEBUG_ENABLE_CORE=1. Core dumps provide significantly more problem determination data to help solve Domino problems. CORE dumps can be rather large and the default location is NOTESDATA, it is recommended that you direct core dumps to a different directory with more space using the notes.ini variable:
- DEBUG_CORE_PATH=$NotesData/IBM_TECHNICAL_SUPPORT
- SVCDUMP Technote 1100734—Technotes are found on the Lotus Software Knowledge Base. This Technote explains what slip traps can be set for debugging and inclusion in the SVC dumps.
- z/OS Environment variable—OS390_DOMINO_SVC_DUMP=1

5.3.2 Domino server hangs

Data to be collected for a Server Hang

- NSDs
- Domino Console / Domino log
- Operator initiated SVCDUMP or core dump
- System logs
- EREP (z/OS only)
- Debug files

Recommended Settings

- DEBUG_CAPTURE_TIMEOUT=10 (Notes.ini setting)—Will add timestamps to the semaphore debugging if used instead of DEBUG_CAPTURE_TIMEOUT=1, which can be extremely useful.

- DEBUG_SHOW_TIMEOUT=1 (Notes.ini setting)—The semaphore time-out messages will be written to SEMDEBUG.TXT, which gets created under the Notes program directory, or under the Notes Data directory on UNIX platforms.
- DEBUG_THREADID=1 (Notes.ini setting)—This prefixes the console output with the process and thread ID information in the format [ProcessID:Virtual Thread ID-Native Thread ID]. This can be helpful in identifying the process or thread holding a semaphore.
- debug_enable_core=1 (Linux only; see 5.3.1, “Domino server crashes” on page 43 for comments)
- SVCDUMP Technote 1100734 (see 5.3.1, “Domino server crashes” on page 43 for comments)
- Environment variable - OS390_DOMINO_SVC_DUMP=1

5.3.3 Domino performance issues

Domino performance issues will use a different process to determine what data collection needs to be performed for problem determination. It is the experience of the writer that many performance issues end up being caused due to changes in the user workload or patterns from a previous environment where the system performed well. It is also helpful to realize that performance can change between versions of Domino due to new features being added to the Domino server or various clients, to fixes and other changes made to the Domino code, or other reasons. A good example of this was the addition of full text indexing on the fly added to the Domino Release 5 codestream around Version 5.08. This function could consume significant CPU resources while the index was being built. Once Level 2 Support understood the impact of this new function, customers were advised how to add a setting to their notes.ini file to disable this feature. Some customers liked this new feature and chose to keep it enabled, but many customers disabled the indexing to have greater control over their server environment.

It is important for a customer to specifically document the problem and all related changes to the environment during the change from acceptable performance to the present performance issues. Many times the initial comment is simply that end users are experiencing very slow server performance or response time. It is important to thoroughly evaluate and understand the Domino environment to determine where the actual problem is starting. It is possible that a workload is running during this time which normally runs at a different time, such as running compact or updall during prime shift rather than off hours. It is possible that workloads in other LPARs on the System z9 system have changed, causing Workload Manager (WLM) or the Intelligent Resource Director (IRD) to reallocate server resources to maintain the defined policies. Newly applied maintenance (hardware/microcode, operating system, or Domino and other key middleware) may have a performance impact as well. Finally, changes to other parts of the infrastructure may impact end user performance. Examples include moving Domino database onto a new disk storage subsystem or making changes to networking hardware and bandwidth.

Data collection requested by Level 2 Support will vary as the performance issue is clearly defined. Here is a list of potential key data sources to collect:

- RMF™ and SMF data files (z/OS only)
- Statrep data to be used in conjunction with SMF data on z/OS or to used on its own on Linux for System z9
- Operator initiated SVCDUMPs (z/OS only)
- Domino and system logs
- Platform specific performance tools or Technotes

- Data from related Domino systems management tools from IBM Tivoli or other key ISVs

5.4 Domino system maintenance

System z9 customers approach upgrades to production server environments carefully due to the expectation for extremely high reliability and availability on the System z9 platform. It is important for customers to have established strategies and plans to perform three common types of upgrades: Major Domino releases (e.g., Domino 6 to Domino 7), Domino Maintenance Releases (MRs) (e.g., Domino 6.5.4 to Domino 6.5.5), and Domino Server Fix Packs (FPs) (e.g., Domino 6.5.4 to Domino 6.5.4 FP1). Additional planning must be done for changes to the operating system. However, most System z9 customers find that upgrading z/OS or z/VM is a straightforward task for middleware environments such as Domino. This is a key benefit of the backward compatibility and automated systems management available on System z9.

5.4.1 Domino major release upgrades

Customers should create test and migration plans for major Domino releases as soon as feasible after the code is generally available. The test plan should provide all testing needed to allow the customer to understand how the upgrade will impact operations along with creating a repeatable method for performing the upgrade. It is common for customers to set up small test servers and have the core administration team (both Domino and hardware/operating system) run on the system. It is likely customers would choose to cluster this server to existing production servers for acceptable availability in case issues are found.

Customers will vary in their interest to upgrade to a new major release of Domino based on their business environment. As stated earlier in this redbook, IBM recommends that customers generally plan to upgrade Domino releases at least every twelve months or sooner to gain new function and fixes. If customers have done sufficient testing in their test plan, then the timing for performing a major upgrade is based more on customer readiness than elapsed time after the product general availability release. All customers should test the code early to ensure the Domino team has worked through all migration issues before any large scale production migration needs to occur.

5.4.2 Domino maintenance releases

Customers are encouraged by Support to stay current on Domino maintenance releases (MRs), especially from a testing view. Certain customer environments do not allow Domino servers to be upgraded to each MR. Each customer will need to study the fixes available in a new MR and determine whether those fixes are needed. Level 2 Support can also assist a customer in determining the best action to take. Some customers like to wait until the first fix pack is released before doing large scale production upgrades to their Domino servers. However, Level 2 Support consistently encourages customers to run the latest version of Domino due to the large number of fixes and maintenance included in maintenance releases.

5.4.3 Domino server fix packs

IBM Support strongly encourages customers to install the first Fix Pack available on a Domino maintenance release. Sometimes additional Fix Packs are released between maintenance releases. Given the nature of these fixes, customers would work to test and install fix packs as soon as possible for their given production release level. Customers can work with Support for further recommendations. Again, the primary principle that is recommended is to stay current with Domino maintenance and releases.

5.4.4 Operating system maintenance

It is critical to keep System z9 operating system maintenance up-to-date on Domino LPARs. It is always recommended to contact Level 2 Support if you have questions about a specific maintenance recommendation or action as these guidelines are general in nature. There are four recommendations from Level 2 Support on this topic:

- Download and run the latest PTF checker available for your specific Domino release
- Review Technotes for issues known by Domino Level 2 Support
- Regularly check the O/S component hyper buckets for critical maintenance. These fixes should generally be applied as soon as possible.
- Keep all other IBM and ISV tools current (backup, antivirus, management, etc.)

5.5 Best practices and recommendations

Here are some general recommendations and best practices related to technical support and systems management. These recommendations come from Level 2 Support, the IBM field team, and Domino System z9 customers.

- **List two primary contacts on all Domino System z9 PMRs, one from the Domino admin team and the other from the hardware-operating systems support team.** It is common for Domino Support to ask questions that can only be answered by one of the two people. If only one person is listed as a primary contact, time is lost while messages are forwarded to the other person. Level 2 Support can direct a question directly to the expert when both people are listed as contacts. Since all communication should be documented in the PMR record, that can serve as the place where all discussion threads can be viewed. The result is faster communication and overall team ownership of problems rather than one half of the team not caring about a problem because it isn't their problem.
- **Record all server related issues in a common repository accessible to the entire admin team.** This process will ensure that sufficient detail is present for later review by IBM or the customer technical teams. Recall that an earlier recommendation was to analyze whether a current problem is similar in nature to previous PMRs opened with Technical Support. This item allows a customer to easily find these correlations. Finally, be sure to include any relevant PMR number in this common repository, again to help future analysis to quickly research and review past actions.
- **Ensure that all admin team members have access and training to the various ways to open and work PMRs, including electronic and voice options.** Since System z9 teams are used to working with IBMLink™ while Domino teams have worked with Passport Advantage® based tools, there usually is some cross training that needs to be done in Domino for System z9 environments on this point. This facilitates better teaming and communication across the Domino admin team.

5.6 Summary

This chapter provided practical experiences and recommendations for Domino System z9 customers to have high quality server support and administration, fast problem determination, and repeatable processes. Customers expect applications running on System z9 to be well managed and highly available. Following these recommendations will help customers meet these expectations and fully leverage the deep technical skills inside the Lotus Domino System z9 support structure.

6

A short course in Domino performance management

Domino has been running on mainframe hardware since 1997. There is a large history from which to look at both customer and IBM experiences in relationship to performance and capacity management of Domino on System z9. This chapter summarizes those experiences and conclusions.

6.1 OS and virtual memory recommendations

We strongly recommend that you run Domino in an operating system that does not page. In order to ensure this, you must have adequate memory available to the operating system where Domino is running. In addition to real storage, you must also understand the virtual memory utilization of Domino.

The amount of real storage your operating system will require will be dependent on not only the number of Domino partitions (DPARs) you are running, but what features will be running in each DPAR. A mail DPAR that is supporting NRPC client will use a different storage profile than a mail DPAR running Domino Web access (DWA). In addition, some components of Domino are always running but still configurable, such as UPDATE, and others are started at the user's discretion, such as FTI, Cluster Replicator, and Agent Manager. Each acquires and utilizes its own amount of both real and virtual storage.

In addition to mail DPARs, you can also have application DPARs running in your system. An application DPAR will use storage differently from a mail DPAR. Also, the way storage is allocated by each of the Domino processes can be very different among your DPARs.

Because of all these differences, the recommendations below are just guidelines to provide you a starting point in building and managing your storage model. Your real-life environments will have differences from these recommendations that you must accommodate.

6.2 OS recommendations

A good starting point is to plan for 2 GB of real storage for each Domino DPAR. This recommendation starts on the high side of the actual storage that you may need. From here you can monitor your system to see what storage you are actually using, depending on the number of users in your DPARs and the features that they are running. You may use SMF, RMF, or any third-party performance monitoring tool to measures the storage utilization of your LPAR.

Keep in mind that not all Domino tasks start immediately nor do all of the Domino tasks immediately allocate their full storage requirements. Domino typically takes several days for its storage utilization to reach a steady state within each DPAR, even though a majority of the storage is allocated fairly quickly.

For example, the UPDATE task allocates storage based on the requests that are sent to this task. The allocation of the storage buffers within the update task are directly related to the indexing that is done. This is affected by the size of the database, number of views, and size of the data within the views in the update request. Since the largest database/views may not be accessed immediately, it will take some time for the full storage requirements of the update task to reach its steady-state values.

Figure 6-5 on page 58 shows several weeks of storage utilization from a set of production DPARs running in Poughkeepsie. You can see that there is not only a daily but also a weekly fluctuation in the amount of storage utilized by the DPARs.

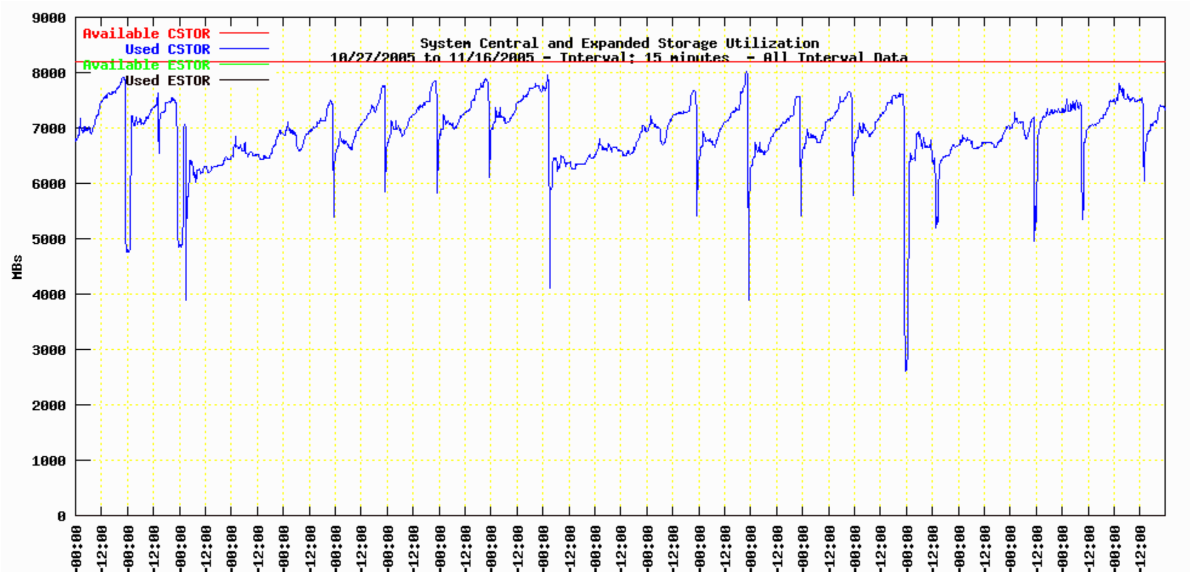


Figure 6-1 Daily and weekly storage usage per DPAR

6.3 Virtual storage requirements

Domino in z/OS and Linux for System z9 for System z9 runs in a 31-bit virtual addressing environment. This means that Domino has 2 GB of virtual addressability. You must not exceed this limit or the Domino server will panic and crash. Domino uses shared resources, so one task will allocate memory resources that come out of this shared 2 GB of virtual addressability. For small and medium-sized DPARs with less than 1300 active users, it is typically not a concern. For larger DPARs with over 1500 active users, this can become an issue.

Running a regular mail workload by itself with this many users will normally not create a problem. However, if you want to mix different types of mail workloads, such as NRPC and HTTP, along with full text indexing, clustering, replication, and agents, you may run into this problem.

As you start more tasks in Domino, each one has its own virtual storage requirement. However, each of these tasks takes up common storage that is shared across all of the processes within a single Domino DPAR. Therefore, for example, it is possible to start the update task and then run out of virtual storage in the HTTP task because the update task allocated more shared memory.

Another area that has a direct correlation to the amount of virtual storage being used by your DPAR is the NSF buffer pool. The default size of the NSF buffer pool is defined differently on z/OS than on other platforms. In z/OS, the buffer pool is initially set to 172 MB. This setting is hardcoded for z/OS and all DPARs will initially use this value when they are started. For small mail DPARs or gateways, this size may be perfectly adequate for your DPAR. However, for larger DPARs you will need to change this value. This can be done with the *NSF_Buffer_Pool_Size_MB* setting in the notes.ini. It is **strongly** recommended that you do not set this value larger than 400 MB. Since this value comes directly out of a 2 GB of virtual storage addressing that each DPAR has, setting this value higher than 400 MB can cause you to run out of virtual storage in your DPAR, which would cause your servers to panic.

Directly related to the NSF buffer pool is the *NSF_DbCache_Maxentries* setting in the notes.ini. In Domino R6 and later, the number of DB cache entries is the size in bytes of the NSF buffer pool divided by 300K. Your default NSF buffer pool of 172 MB would give you 601 DB cache entries. This value will not create the number of DB cache entries needed to support a large number of users in a single DPAR. Since we strongly recommend not to exceed 400 MB for the NSF buffer pool, the largest number of DB cache entries you can create using the default value would be 1398.

In a large DPAR, it is not uncommon to see 3000 or more registered users. Therefore we need to have a DB cache Max entry that is larger than the 1398 entries defined by a 400 MB NSF buffer pool. By using the NSF_DbCache_Maxentries notes.ini settings, you can set the number of DB cache entries that you need to support your workload while leaving the NSF buffer pool at 400 MB or less.

Remember that Domino can dynamically extend these entries by 50% during the normal running of your DPAR. Therefore, if for example you set the number of DB Cache entries to 1000, you can expect to see Domino actually allocate 1500 DB Cache entries if it needs to have them in your cache. Therefore you would need the additional storage to dynamically extend the cache for these new entries. Also Domino will try to manage the cache entries back down to the original DB cache maxentries number. While having Domino dynamically extend the number of cache entries can help keep your server running better, it will cost you more cycles to dynamically extend and reduce your cache rather than to simply find the correct size you need.

To determine the right size for your *NSF_DbCache_Maxentries*, you should monitor the number of db_cache rejects that are occurring in your DPAR. You would want to keep this value to zero during your prime shift when your users activity is at its peak. However, you may have large spikes of db_cache rejects that occur during your evening and early morning hours. This is when most of Domino's maintenance processes run. Some of these processes can sequentially scan all databases, producing db_cache rejects. During these maintenance periods, these are acceptable.

6.4 A good reason why to have two CPUs in your operating system image

In addition to running the tasks that support the end users (server, HTTP, IMAP, etc.), there are several other background tasks that Domino must run to maintain the databases that support the end users. These tasks, such as the indexer, agent manager, and clusterer, can be very CPU intensive. If you only have one CP available to the Domino partition while one of these task executes, you might experience end user response time degradation. Since there is no CPU on which to dispatch the client requests, they must wait for the CPU to be freed by the operating system. During these time frames, a single CP will be 100% busy.

While Domino does not need two CPs to run, having two available to your Domino partition can help you to prevent user degradation when a maintenance task is executing.

6.5 Workload Manager

Workload Manager (WLM) is the z/OS component that manages the flow of work through your system. To use it, you set up business rules for the various workload types executing in your environment. These rules allow you to identify what workload types have priority over other workloads. WLM can apply these priorities to the CPU, storage, and I/O resources in your system.

6.5.1 Terms

This is an overview of some of the terms and ideas that we will be using in our discussion, for those who want a refresher on the features of z/OS.

- Service Policy—This is where you define the definitions that WLM will use to manage your workloads. You can only have one active service policy per z/OS image, so all workload definitions that you manage must reside in a single service policy. WLM reads this service policy and builds in rules for managing your workloads. You can dynamically edit this service policy and reload it into WLM while your system is running. This allows you to make changes to WLM without having to restart either your system or its workloads.
- Service Class—This is the manageable unit of work within WLM. WLM builds a list of all of the service classes that you have defined in your service policy. In addition to these classes, their definitions are loaded into workload manager so that it understands what priority levels each of these classes have. WLM then watches the workloads that are in each of these classes against the definitions that you have defined. If a class is not meeting its performance definitions and there are service classes of lower importance, WLM will take resources from the lower class and give it to the upper class to help meet its workload goals. You can define many different address spaces or jobs in your z/OS environment to run in a single service class. WLM will service all jobs and tasks within a single service class equally.
- Report Class—This is a unit of work that WLM can report on independently. For example, you should place all your Domino tasks into a single service class within a DPAR. However, you might want to see the resource utilization of each individual task. You can place each task into a separate report class that WLM will give you statistics about. However, WLM will only manage the resources at the service class level. Therefore, it will treat all of the Domino tasks equally within that service class. You may define as many different jobs into each report class as you want to.
- Service Period—When you define a service class you, must provide it with definitions. One of these definitions is service periods. Service periods allow you to define multiple performance characteristics for this service class. Each task or job can have its own definition on how many resources it can consume in this period before it moves to the next period. This allows you to let short running jobs have a higher priority than a longer running job. Depending on how many service periods you have defined, a job can successfully cascade through these periods until it reaches the bottom service period for that service class.

6.5.2 Ways to avoid issues with WLM

Domino on z/OS is workload manager tolerant, not workload manager enabled. This means that Domino can run in a WLM environment, but that workload manager cannot manage the individual components within a DPAR. Domino does not have a mechanism to externalize the locking among its components as it is running. WLM cannot see which Domino task (or process or user) is the most critical lock holder at any given time.

It may seem good to protect the server or HTTP task to ensure that end users get good response time when the CPU is under constraint. However, there is no way to tell if the end users in the server task are waiting on another Domino process before they can run. For example, you may have just replicated a database and are in the process of rebuilding its indexes. But you have identified the indexer task as one that can run at a lower priority in a constrained environment. Users trying to access that database could be waiting while the indexes are being rebuilt. Since this task is now running at a lower priority, it will take much longer than it normally would. This can result in more users than normal waiting on the Domino server.

Besides the indexer task, there are several other Domino tasks that can create issues by running in a lower priority service class. These include agent manager, full text indexer, clustering, or replicator. Any one of these tasks could have a critical lock that other tasks are waiting for. By running these tasks in a lower priority service class, users could experience response time delays in your Domino server. A majority of the time, this will not be the case and things will run smoothly. However, you must be aware of the times that having these different priority levels within a DPAR can impact your users negatively.

Until Domino can externalize its locking mechanism so that Workload Manager can determine which user/process/task is most critical, we recommend that you do not set the different tasks in a DPAR into different service classes with different priorities in WLM. However, you can use different report classes inside WLM to see the individual activity of each of these Domino tasks.

While we do not recommend setting individual tasks within a given DPAR to different service classes within WLM, it is possible for each individual DPAR in your LPAR to be set into a different service class within WLM. This allows you to run your mail DPAR at a different service level from your application DPAR. Another example would be to have two mail DPARs that are running at different service levels. For example, you could have the mail DPAR for your executives running at a higher service level than other mail DPARs.

6.5.3 Recommendations on how to use WLM with Domino

If you run multiple DPARs with the same service level in WLM, we recommend that you use only one service class. You should not define multiple service classes with identical service definitions. Since the service classes are treated individually by WLM, users in identically defined service classes could receive different priorities.

When you define a service class for a Domino server, that service class should have only one service period defined to that class. Domino tasks are long-running ones that typically do not end once the DPAR is started. Therefore, if your service class has multiple periods in it, the various tasks will eventually end up in the bottom service period. Any task that does start in this service class will be placed into the first period, which will typically have higher priority than the tasks running in the last period. Also, if a new DPAR is started in the same service class, it would be placed in the first service period and receive better priority than the DPAR that has been running for while and has moved to the bottom period.

If you are running a mixed workload environment with Domino, your service policy must describe the performance criteria that your Domino service classes will have. Ideally, Domino should run below the subsystems that it needs. This can present a problem for Domino if the higher priority subsystem has a runaway task or the CPU usage gets constrained. While WLM gives resources to the higher priority workload to meet its, your Domino end-users may experience response time issues.

By splitting your DPARs into mail, application, and hubs, you can assign each one of them a different priority setting with Workload Manager. This can allow you to place each DPAR into a different service class to ensure that your mission-critical DPARs have a higher priority and access to resources in a constrained environment.

6.6 How many tasks should you run?

Care must be taken when deciding how many instances of each Domino task you want to start within your DPARs in your LPAR. For example, you could start multiple indexer tasks and multiple agent manager tasks. There is documentation on the Lotus Web site giving you recommendations as to how many of the components you can safely start in your DPAR.

Remember that these recommendations **were made based on an assumption that a single Domino server would be the only thing running on that hardware**. With the ability to run multiple Domino servers in the same hardware, you must be careful in using these recommendations.

For example, the recommendation is to run a N-1 indexers, where N is the number of CPs available to your operating system environment. If you have four CPs, then you can safely run three indexers. This allows one CPU to always be free to handle user requests that come in. In the above example, if we have two DPARs running on the same four-way hardware, and we let each DPAR have three indexers, this would gives us a total of six indexers running on the four physical CPUs. With six indexers running (this being a very CPU intensive process.) we could very easily overwhelm the four physical CPUs. It may be possible to stagger the times the different indexers are running, and therefore each DPAR still can run three indexers. However, care must be taken when you do this. Also, if the indexer starts to run longer, you may run into issues when they start to overlap.

At the same time, we must be aware of the other tasks running in Domino. The typical recommendations assume that no other Domino processes are running. If we use the indexer example again, this assumes there is no clustering, no replication, no agents, and no other CPU-intensive task running. Therefore, if you want to run other tasks in addition to the indexer, you must consider their impact on the CPU.

This does not imply that the recommendation of N-1 in the above example must be the sum of all indexers in all the DPARs in the LPARs. As we mentioned above, it is possible to schedule the various components of each DPAR to run at different times. However, if you do want to run them at the same time, you must consider the overall resources available.

6.7 Ongoing capacity planning and performance management

There is an old saying “if you don’t know where you been, how do you know if where you’re going is any better?” This saying it is also true for Domino. It is very hard in a problem situation to understand what your data means if you do not know what your data looked like when Domino was not in a problem situation. So don’t wait until you have a problem before you begin collecting and studying your performance data.

The same is true for capacity. If you do not have a history of what resources your servers were using and how this usage grew, how can you plan for your resource requirements in the next month or six months from now? How do you know that your servers are not creeping close to maximum utilization?

In order to move from a reactive to a proactive environment for supporting Domino servers, you need a methodology for collecting and analyzing your data. This methodology must encompass not only Domino but also your system data and environment. By looking at and integrating these various data sources, you can build a complete picture of your server, both hardware and software. Remember that in a System z9 environment, your Domino servers are running with shared resources. Therefore it is possible for one DPAR to impact the performance of another. For that matter, a non-Domino workload can impact your Domino workload.

When you first planned for your Domino environment, you typically sized your LPARs to handle the anticipated workload for your DPARs and LPARs and defined your LPAR manager settings accordingly. However, over time, you will find that your production workloads may differ from what you anticipated. One DPAR could be taking more resources than originally planned while another might not. So you will want to go back and balance your LPARs and their weightings in the LPAR managers in order to avoid under- or over-allocating the required resources for your Domino LPARs. Otherwise these definition mis-matches will cause problems if your resources become constrained.

For example, if there are extra CPU cycles available, then your LPAR weightings are not used, since all LPARs can get the cycles they need. But if your system becomes CPU-constrained, then the LPAR manager will look at the definitions to decide what is the appropriate level of resources for each LPAR. A year or two after your initial Domino implementation, your Domino LPARs are probably using an entirely different amount of CPU resources than they were initially sized for. If an LPAR is using more then its fair share of CPU and the CPU gets constrained, the LPAR manager will take cycles from this LPAR to give it to another LPAR, who is in need of the cycles and is below their fair share.

The fair share of an LPAR is its weight divided by the sum of all the LPAR weights. For example, you might have a system with four LPARs. LPAR A has a weight of 20, LPAR B has a weight of 30, LPAR C has a weight of 50, and LPAR D has a weight of 100. The sum of all the LPAR weights is 200 (20 + 30 + 50 + 100). Each LPAR's fair share in this example is its own weight divided by 200. LPAR A's fair share is 10%, LPAR B's is 15%, LPAR C's is 25%, while LPAR D's is 50% of the system. In Table 6-1, we show how this fair share would affect the "guaranteed" CPU resources on two different systems.

Table 6-1 LPAR fair share

LPAR Name	LPAR Weight	LPAR Fair Share	4 CP system	10 CP system
A	20	10	40% of 1 CP	1 CP
B	30	15	60% of 1 CP	1 ½ CPs
C	50	25	1 CP	2 ½ CPs
D	100	50	2 CPs	5 CPs

To continue with our example, let's say that LPAR D (in the 10 CP example above) is using 6 CPs, or 20% more than its fair share. As long is the system is not constrained, then this does not become an issue. However, if the system becomes constrained, then the LPAR manager will only guarantee 5 CPs of cycles to this LPAR. With this loss of cycles, we would expect to see end user response times becoming an issue.

Although there was nothing wrong with this LPAR before the system became constrained, the calls you now receive about response times might cause you to think otherwise. If you have not been monitoring your environment, then you would not have a before and after comparison of the workloads and resource usage.

The ability of another DPAR or LPAR to affect your Domino server in a shared environment is called "sympathy sickness." In the above example, we showed how incorrect LPAR definitions can cause this. Another way is for one DPAR to have a large jump in resource utilization. This could be due to a looping task such as a badly coded user agent or a router dealing with a corrupted mail header. Again this would only occur if one of your LPARs was taking more than its fair share.

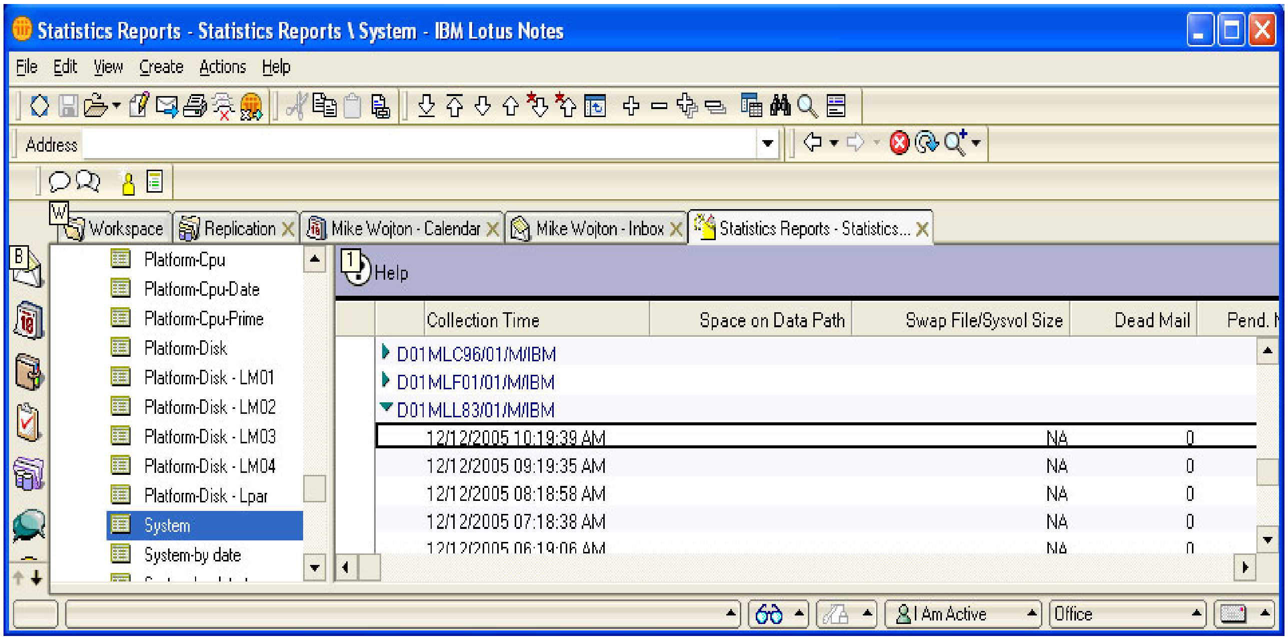
Note that we would see response time issues in the DPAR/LPAR that was not having the runaway task. Our initial inclination would be to look at the LPAR/DPAR that users were complaining about. However by being able to look at the history of the resource utilization of the various LPARs on this system, we could very quickly see that this LPAR was being denied some CPU resources while a different LPAR had a spike in CPU resources. By understanding what has changed, we could quickly focus on the LPAR that had the CPU spike, not the LPAR that had the end-user response time issues.

6.8 What data sources are available to you?

In order to monitor your system for performance and capacity, you should understand what data sources are available to you in the Domino environment on System z9. There are three primary sets of data sources.

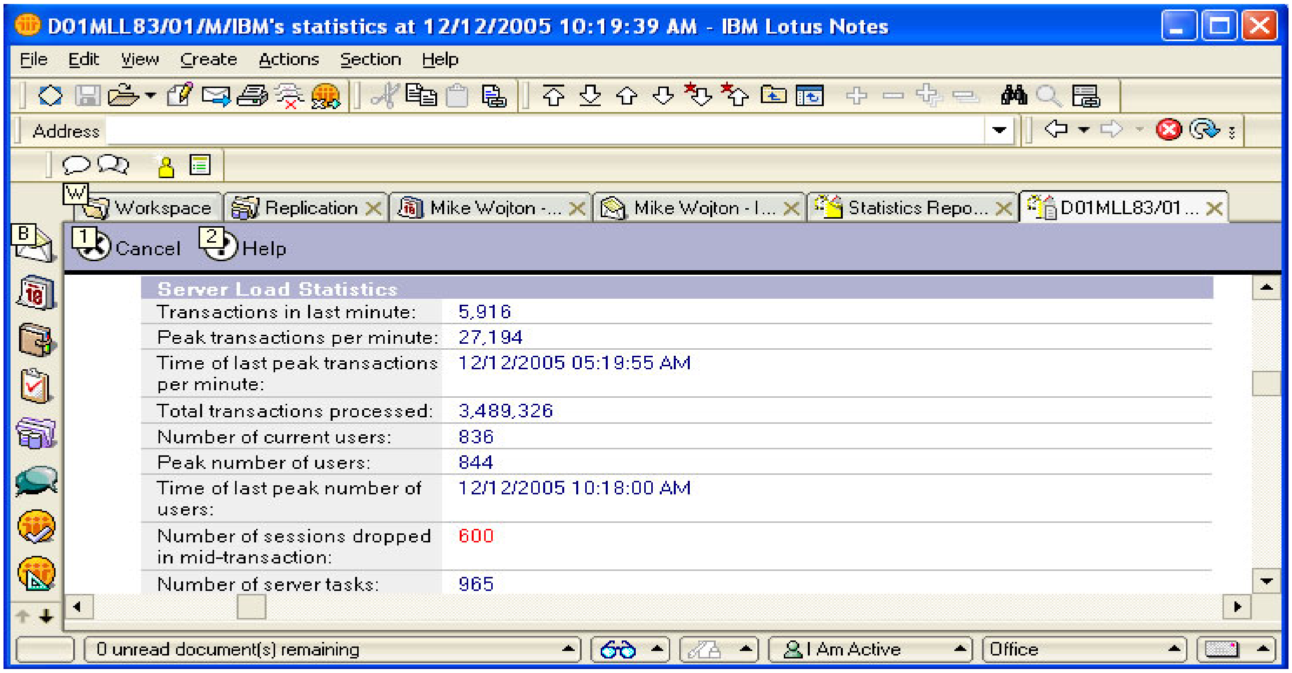
6.8.1 The statrep database

The first data source is the Domino statistics data that is available in the statrep database. This provides a historical view of the various Domino statistics. However, by default, no statistics are collected in the database. You must enable the collect task to take the data from the Domino buffers and write it to the statrep database, or you will have no statistical data. Figure 6-2 and Figure 6-3 on page 56 show how to access this database.



Collection Time	Space on Data Path	Swap File/Sysvol Size	Dead Mail	Pend. M
D01MLC96/01/M/IBM				
D01MLF01/01/M/IBM				
D01MLL83/01/M/IBM				
12/12/2005 10:19:39 AM		NA	0	
12/12/2005 09:19:35 AM		NA	0	
12/12/2005 08:18:58 AM		NA	0	
12/12/2005 07:18:38 AM		NA	0	
12/12/2005 06:19:06 AM		NA	0	

Figure 6-2 Locating a statistics report


D01MLL83/01/M/IBM's statistics at 12/12/2005 10:19:39 AM - IBM Lotus Notes
File Edit View Create Actions Section Help
Address
Workspace
Replication
Mike Wojton -...
Mike Wojton - I...
Statistics Repo...
D01MLL83/01 ...
Cancel
Help
0 unread document(s) remaining
I Am Active
Office

Server Load Statistics

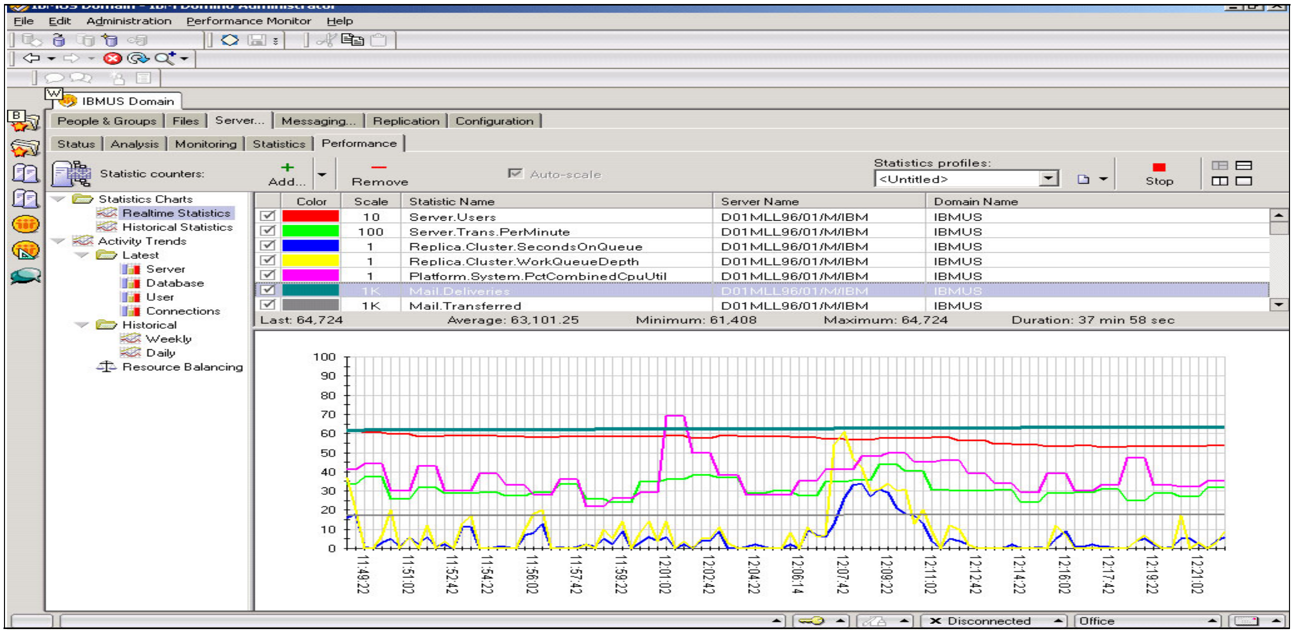
Transactions in last minute:	5,916
Peak transactions per minute:	27,194
Time of last peak transactions per minute:	12/12/2005 05:19:55 AM
Total transactions processed:	3,489,326
Number of current users:	836
Peak number of users:	844
Time of last peak number of users:	12/12/2005 10:18:00 AM
Number of sessions dropped in mid-transaction:	600
Number of server tasks:	965

Figure 6-3 Server load statistics

The default interval for Domino to collect the statistical information is two hours. While a two-hour interval may be useful for capacity planning, it is not very useful from a performance point of view. Therefore, we recommend that you change the default interval to match either your SMF interval in z/OS or your SAR interval for Linux for System z9. Later on you can combine and summarize these intervals to build capacity data, but you cannot break them apart to produce performance data.

6.8.2 Notes administration client

The second data source is the Notes administration client, which can provide both real-time data and historical data about your Domino servers. Also, the trends component, which was formerly a billable feature in the Notes administration client, is part of the base client in Release 7. It allows you to analyze the various workloads that are running in Domino and provides information about what users and databases are the heaviest in your DPAR. Figure 6-4 on page 57 shows a sample of the reports available in the admin client.



	Color	Scale	Statistic Name	Server Name	Domain Name
☑	🟥	10	Server.Users	D01MLL96/01/M/IBM	IBMUS
☑	🟩	100	Server.Trans.PerMinute	D01MLL96/01/M/IBM	IBMUS
☑	🟦	1	Replica.Cluster.SecondsOnQueue	D01MLL96/01/M/IBM	IBMUS
☑	🟨	1	Replica.Cluster.WorkQueueDepth	D01MLL96/01/M/IBM	IBMUS
☑	🟪	1	Platform.System.PctCombinedCpuUtil	D01MLL96/01/M/IBM	IBMUS
☑	■	1K	Mail.Deliveries	D01MLL96/01/M/IBM	IBMUS
☑	■	1K	Mail.Transferred	D01MLL96/01/M/IBM	IBMUS

Last: 64,724 Average: 63,101.25 Minimum: 61,408 Maximum: 64,724 Duration: 37 min 58 sec

Figure 6-4 Server performance tab

6.8.3 Statistical packages

The third data source is the platform statistical packages provided by either z/OS or Linux for System z9. In z/OS, this is typically the SMF and RMF reporting systems. In Linux for System z9, this is typically the SAR reporting package. Both of these packages in their respective LPARs provide you with the ability to analyze the platform data. In addition, if you are running Linux for System z9 under z/VM, then you have the z/VM statistical packages available.

While there has been work to integrate the Domino environment into the z/OS SMF platform platform stats, there is no such capability with the Linux for System z9 and SAR statistics. However, by using either a spreadsheet, DB2®, or some reporting package, you can merge the Linux for System z9 data sources together, as you can in the in z/OS environment.

Your Domino servers have a very dynamic workload profile. Over time, you can build a trend as to what this workload looks like. But at any given instance it can vary greatly, depending on what features and functions your users are using. Therefore in order to interpret what the platform stats are telling you, you must be able to understand and interpret what workloads were running from your Domino servers at the same time as your platform stats. By integrating your platform stats with the Domino stats, you can build a complete picture of what workloads were running with what resources at what time in your environments.

Figure 6-5 on page 58 shows an example of active 15 minute users on a set of DPARs in an IBM Poughkeepsie production environment. You can see the drops in the active 15 minute user count during the holiday periods such as Christmas, Thanksgiving, Labor Day, July 4, and Memorial Day. What is also interesting in this chart is the overall trend in the number of active users over the entire year. We can see that there is a significant jump in a number of active users around Labor Day (beginning of September). Looking back over several years worth of data, we can see that this trend has repeated each Labor Day.

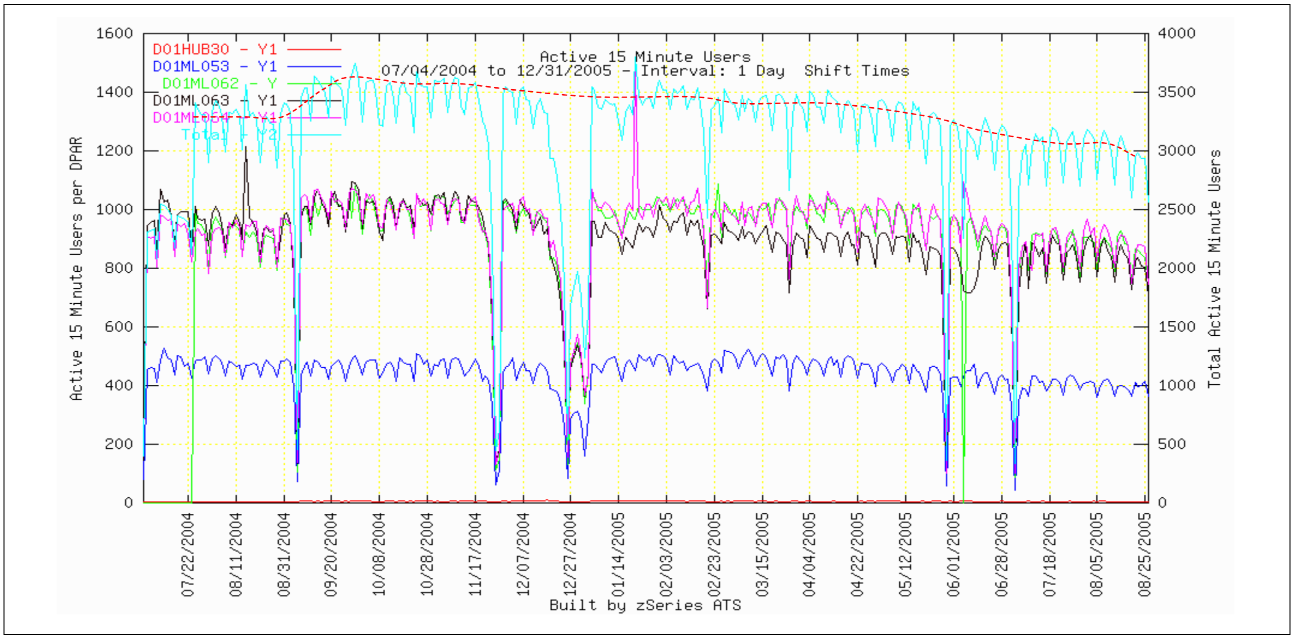


Figure 6-5 Active 15 minute users

This trend is not surprising since schools reopen around Labor Day in the United States and many vacations end around this time period. Note that if we had done a sizing based on the July or August data, then we would not have had a true picture of the normal workloads of these DPARs.

This trend with active users also matches the trend in CPU usage over the same time period. So there is a direct correlation between the number of users on our servers and the amount of CPU resources the servers were taking. If we were just comparing the middle of September's data to that in the middle of August, we could easily reach the conclusion that there is a performance problem. How else could you explain the large increase in CPU usage over a very short period of time when the new users have been registered to your DPARs?

If you are not tracking your servers over a long period of time and understanding their usage patterns, you can fall into a trap of using data that does not truly represent your user workload at any given point in time. These trends can be found in almost every business environment. If we do not understand what these trends are and how they impact our servers, then the data we use for capacity planning can be skewed by taking data from either a peak or a valley in our workload trends.

6.9 What should you be monitoring?

There is a vast amount of data available to you from Domino and the platforms that details how your servers are running. The hardest part is trying to figure out what pieces of data to look at. Keep in mind that what measurements works for one server may not work for another server. What you would monitor in a mail server would be different from a gateway server or an application server. Also how your users access the server can impact what to monitor. For example, you will be looking at different stats for an NRPC Notes Client versus a DWA client or a POP3 or IMAP client.

The key is to figure out what statistics and information give you a good overview of your servers in your environment, then build their baseline measurements. By looking at this high-level view and understanding your baselines, you will determine if there is any need to drill down further. You do not want to view and interpret all the data that your system and servers are producing.

In Figure 6-6 we have plotted the CPU usage for the separate tasks in a Domino server. As you can see over time, there is a very stable trend of what components of Domino are using what resources. We can see in the front part of the chart where the router was taking a lot more resources than normal. However, if this was the first time that you were looking at this data. you would not know that router was using an abnormal amount of CPU.

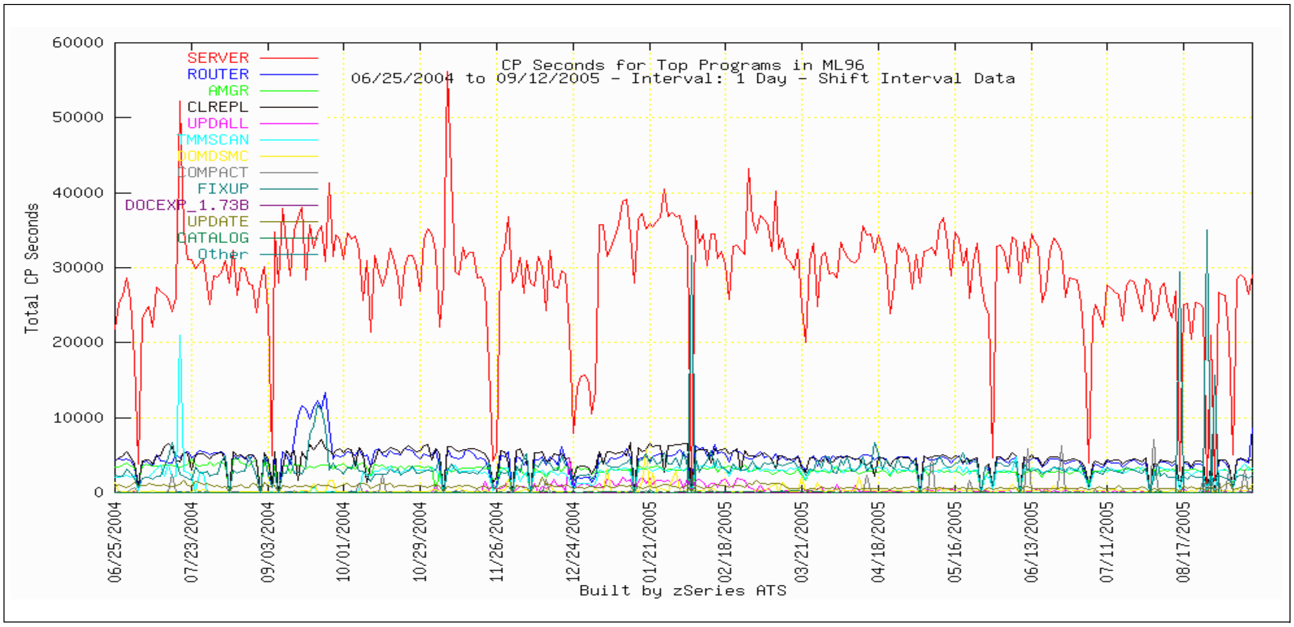


Figure 6-6 CPU usage

By building this profile of typical usage of your DPARs, you can spot abnormalities on your servers. In some cases, the task taking the increased resources could be the victim and not the culprit. In the above example, if we were having a network problem, so that the router was rebuilding its network tables or doing retries, this could cause the high CPU usage. So while the router may not have been cause, at least we have a place to start looking and identifying what is going on. If we see from the Domino stats that there was a large number of mass mailings going on, then this spike could be all workload related. This is why it is important that you have access to both the Domino statistics and the platform statistics together to be able to coordinate what is happening within your DPAR and within your system.

6.10 Interval versus cumulative data

Keep in mind that while your systems typically give you interval data, Domino typically gives you cumulative data. You must therefore reconcile these two data formats before you can use them together. This typically means creating interval data from your cumulative samples in Domino.

Note: if you are using the Domino data from the 108 SMF record type, then this data is already in interval format. If you are planning to use data outside of SMF that is native in Domino, then you must convert it to interval format in order to use it with your platform statistics.

The reason it is important to convert the cumulative data to interval data is that you need to know when the change occurred in the data, not just that it has a large count. For example, you decide to monitor the DB cache rejects in Domino to see how your buffer pools are performing. You may see a large number of rejects showing up in the statistic. When did these rejects occur? If they occurred during the day when you have user activity, this could adversely impact your server's performance. If these rejects are occurring in the middle the night when Domino is doing its maintenance activity and sequentially accessing all of its databases, then they are not a concern. To convert the cumulative data in Domino to interval data, you subtract the two data samples in statrep from each other. The length of your interval is the difference between the two statrep samples and the data value is the difference between the two samples.

It is important that you understand what each statistic represents, since not all data in Domino is cumulative. For example, while the transaction count is cumulative, the DB cache hit ratio statistics is a current rate value.

6.11 How sample periods affect your data

Once you convert to interval data, you must decide what samples you want to look at. Below are two charts. Figure 6-7 shows you the full 24-hour view of a set of LPARs and their CPU utilization. Figure 6-8 on page 61 shows you the same data, but only for prime shift. In these charts, prime shift is defined from 8 a.m. to 4 p.m., Monday through Friday. These charts represent the same days plotting data from the same samples. The only difference is which samples were included in the plot.

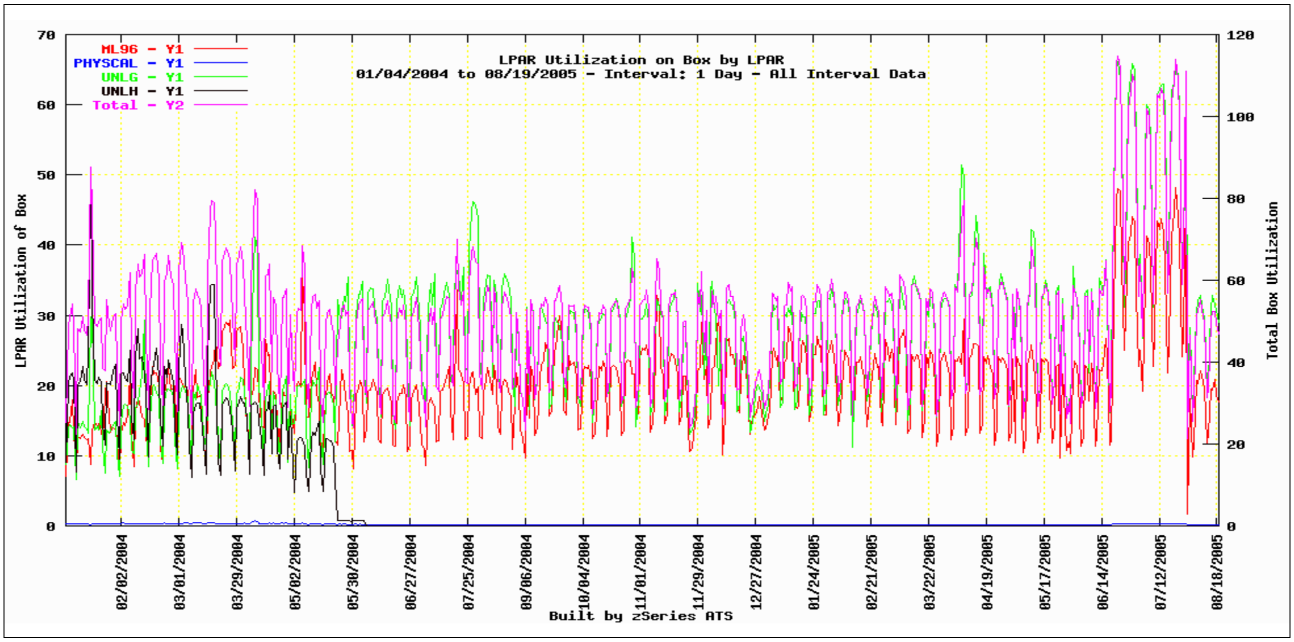


Figure 6-7 LPAR utilization — 24 hours

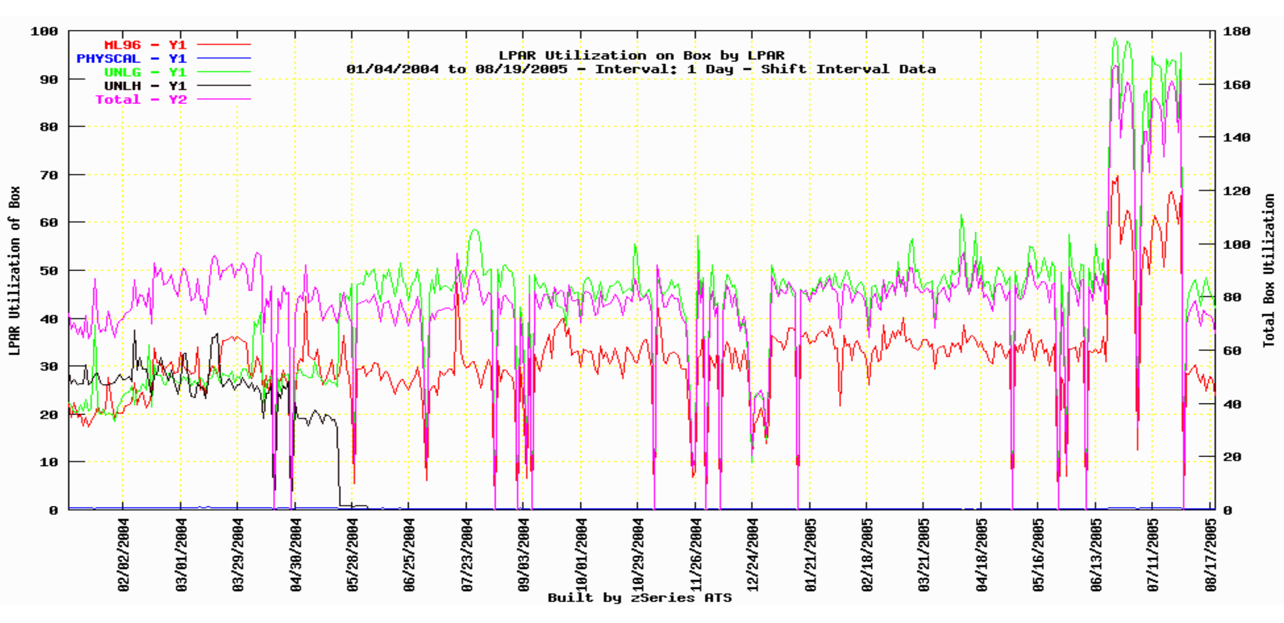


Figure 6-8 LPAR utilization — 8 hour prime shift

As you can see, Figure 6-7 on page 60 shows a much lower overall CPU utilization than Figure 6-8. This is because the samples for prime shift are much higher than those for the rest of the day. By averaging in the full 24 hours you see a lower average than you would see if you just average the prime shift samples.

In Figure 6-7 on page 60, on the far right you can see a CPU spike that averages around 70% for the day. However, if we look at just the prime shift data in Figure 6-8, we see a different picture. Here we can see that the CPU was saturated to almost 100%. You must not only understand the type of data and intervals that you are looking at, but the sample set that is represented in order to truly understand what your data is telling you. Let's discuss an approach.

6.12 Statistics to start with

Domino's workload is very dynamic; it changes with your user activity. Since at any given moment you do not know what your DPARs are being asked to do, looking only at resource utilization can present you with a misleading picture. For example, what if the router spike in the above charts was due to an increase in workload, such as a mass mailing? It seems unlikely that this would be the cause for the extended period of time in the above charts, but a spike in router for an individual day may not be uncommon at all due to mass mailings.

Therefore in order to understand how well or the performance characteristics of your DPARs you need to look at a cost per unit, not at the raw resources being utilized. Rates such as CPU busy and the number of I/Os are normally capacity indicators while response times or cost per unit are performance indicators. If your CPU is at 80%, you can add 20% more workload before you max out at 100%. However, your CPU running at 80% does not give you any indication of how well your DPAR is running.

If you chart the CPU used per active user and this trend is flat but the raw CPU has gone up, then this indicates that your workload has increased by adding more users. You have a capacity issue, not a performance issue, and you are reaching some resource limit. However, if your CPU has gone up and your cost per active user has gone up, then your server is not running as efficiently. or the user workload has increased for some reason. So you have a performance issue rather than the capacity issue.

Here is a starting list of cost-per-unit measurements that you can generate and monitor in your Domino environments for NRPC users. These are especially useful for monitoring your mail DPARs. However, some of these statistics could be useful in monitoring an application DPAR, depending on what features and functions you have enabled. All of the statistics below can be obtained from the SMF 108 records for Domino in your z/OS environment.

Note: The samples below are purely for demonstration and do not represent any recommendations that your DPARs should be measured against.

- CPU cost per active 15 minute user (see Figure 6-9). This is the total number of CPU seconds used by all the processes of a DPAR divided by the total number of active 15 minute users in the sample period. This will give you a good indication of how costly it is for your users on your Domino servers. As this number goes up, your users are getting heavier and it's costing you more cycles per user to support your workload.

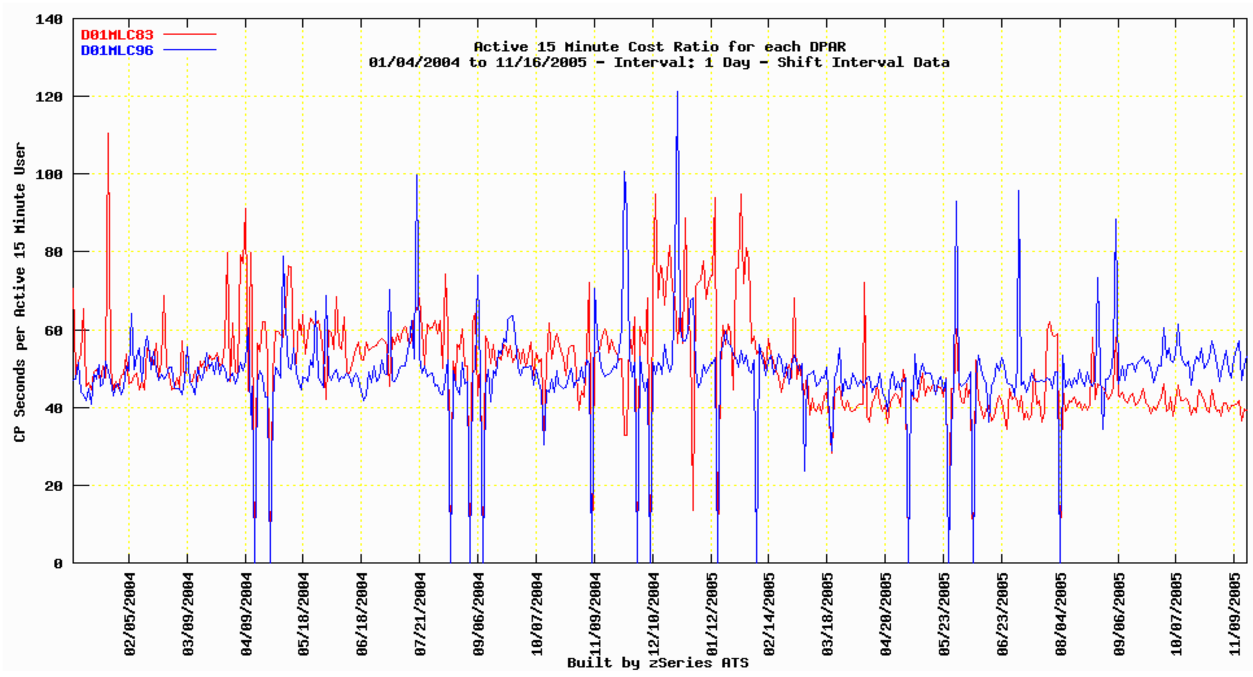


Figure 6-9 Active 15 minute user

- CPU cost per message routed (see Figure 6-10 on page 63). This is the total number of CPU seconds used by the router process divided by the number of messages processed by this task. This includes both local and remote messages delivered by this task. This will give you a good indication of how costly it is for your router to deliver your mail messages. This number should be fairly flat.

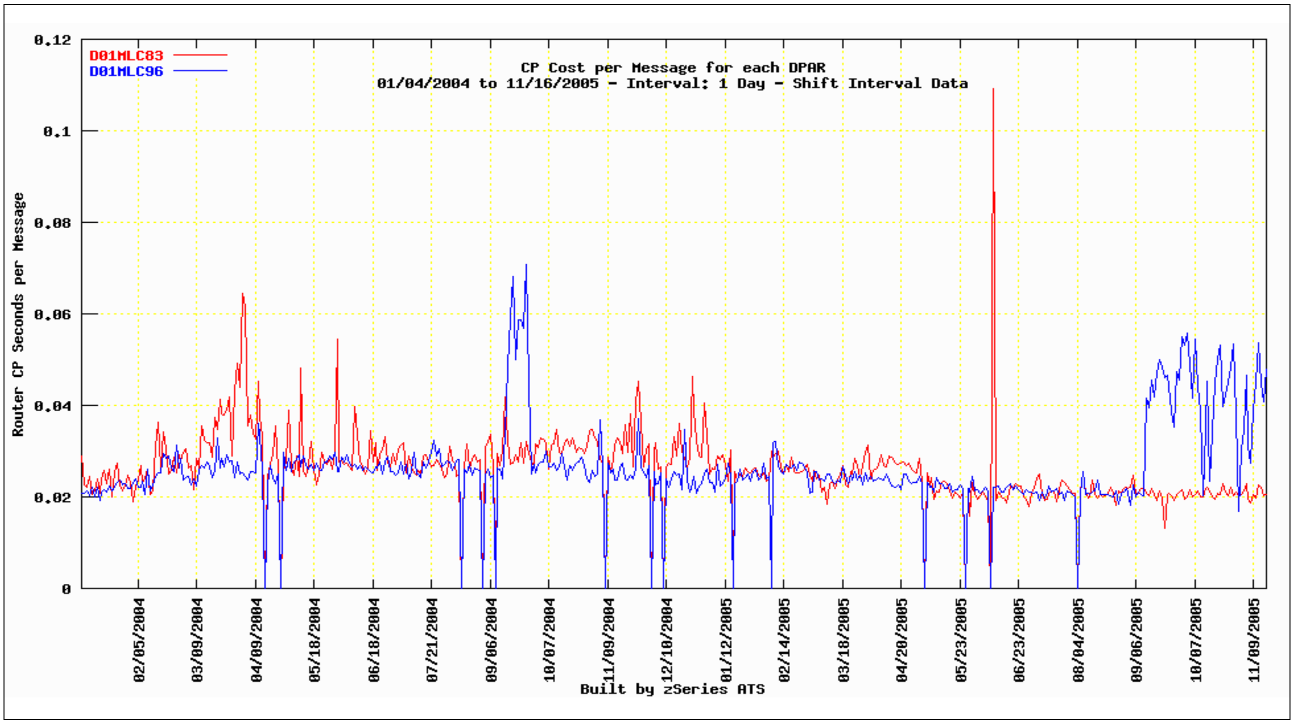


Figure 6-10 Cost per message routed

- CPU cost per Domino transaction (see Figure 6-11 on page 64). This is the total number of CPU seconds used by all the processes of the DPAR divided by the total number of Domino transactions in the sample period. The Domino transaction is a relative indicator of workload on a Domino DPAR. Since customers and DPARs can run differently, this number is not very useful in comparing across different DPARs. However, the statistics can show that something is amiss if there is a large change in the cost of running a Domino transaction within the same DPAR over a short period of time. Keep in mind that things such as server or client upgrades can affect this cost as well.

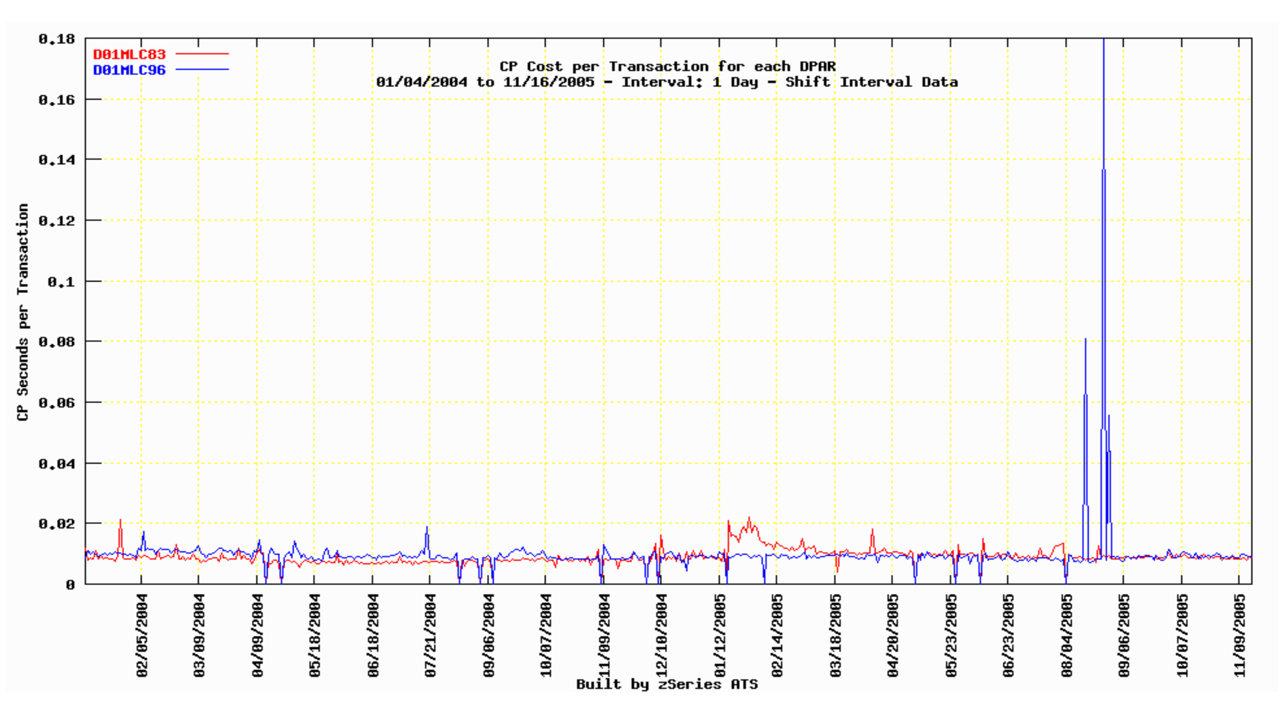


Figure 6-11 Cost per Domino transaction

- Number of transactions per active 15 minute user (see Figure 6-12 on page 65). This is the total number of Domino transactions divided by the number of active 15 minute users in the sample period. As mentioned above, using the Domino's transaction count is not very useful when comparing across multiple DPARs. However, within a DPAR, the statistic can give you a good idea of the amount of average workload being driven by your end users. A change in the statistics can be a good indication that there was a change in the types of workload your users are running within a DPAR.

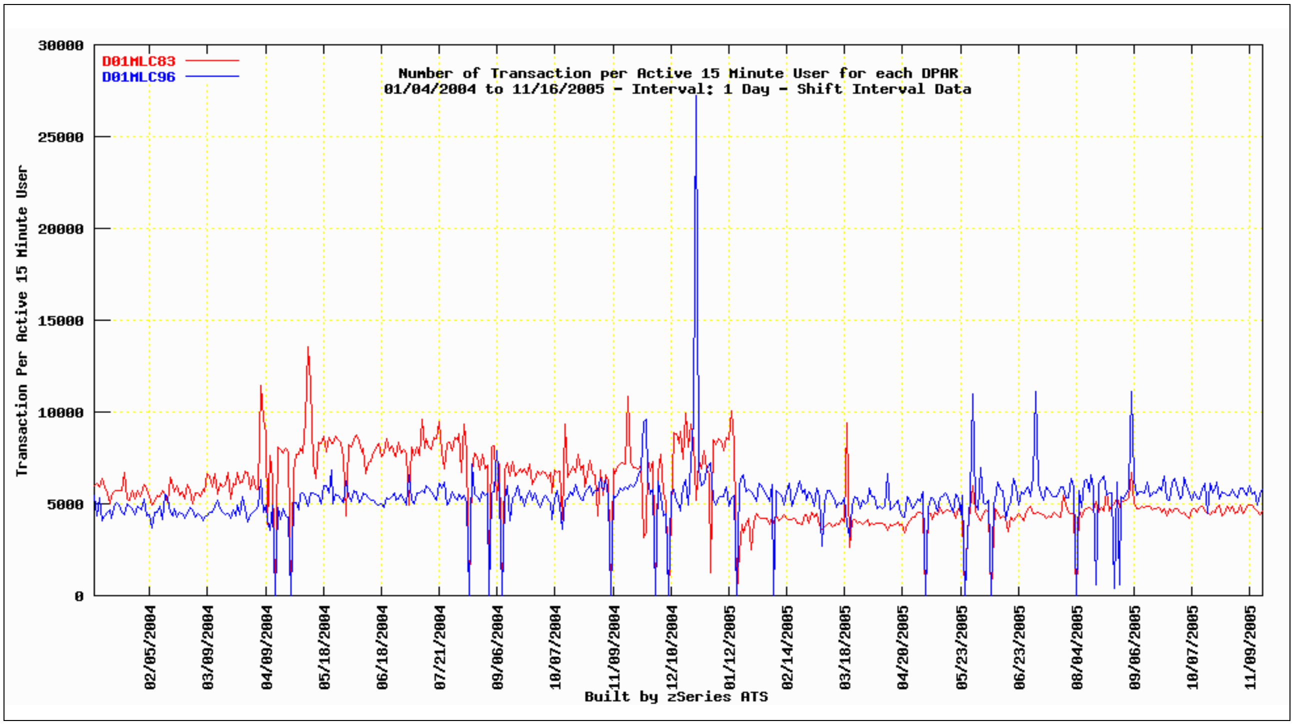


Figure 6-12 Number of transactions per active 15 minute user

- Number of messages per active 15 minute user (see Figure 6-13 on page 66). This is the total number of messages sent across the router divided by the total number of active 15 minute users in the sample period. This measurement represents the number of messages on average been sent by your active 15 minute user. An increase in this value over time will mean that you will need additional resources to support your users (even if you are doing quota management), since more mail is being sent per user.

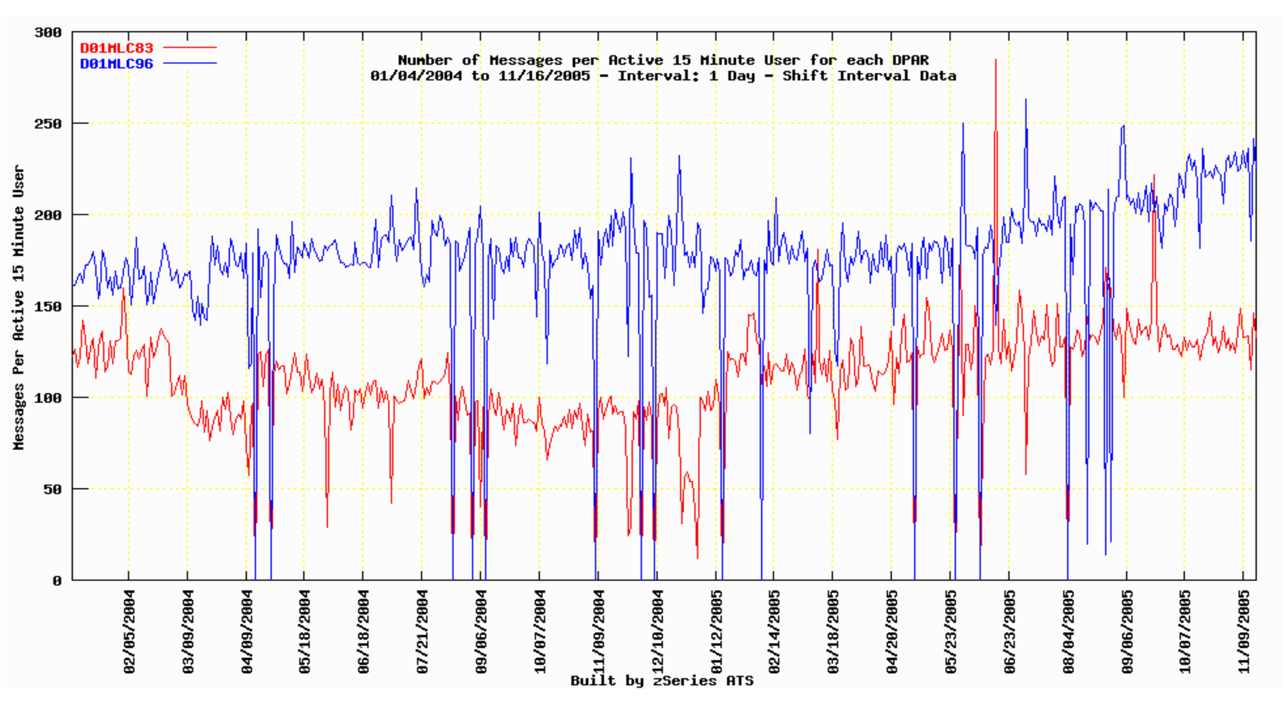


Figure 6-13 Number of messages per active 15 minute user

- Number of active 15 minute users per physical thread in the thread pool (see Figure 6-14 on page 67). This is the total number of active 15 minute users divided by the number of physical threads that were active in this sample period. This statistic is a very good first indicator that something may be going wrong in your DPAR. Typically, your platform statistics tell you when you are running out of resources at the CPU, storage, or I/O level. A sudden change in this statistic can indicate that there is a bottleneck of some sort in your DPAR even before your end users are affected. For example, let’s say you were averaging 20 users per physical thread before and now you are averaging 15 users per thread. This would be an indication that something is slowing your workflow through your system. It is taking you more threads to do the same amount of work.

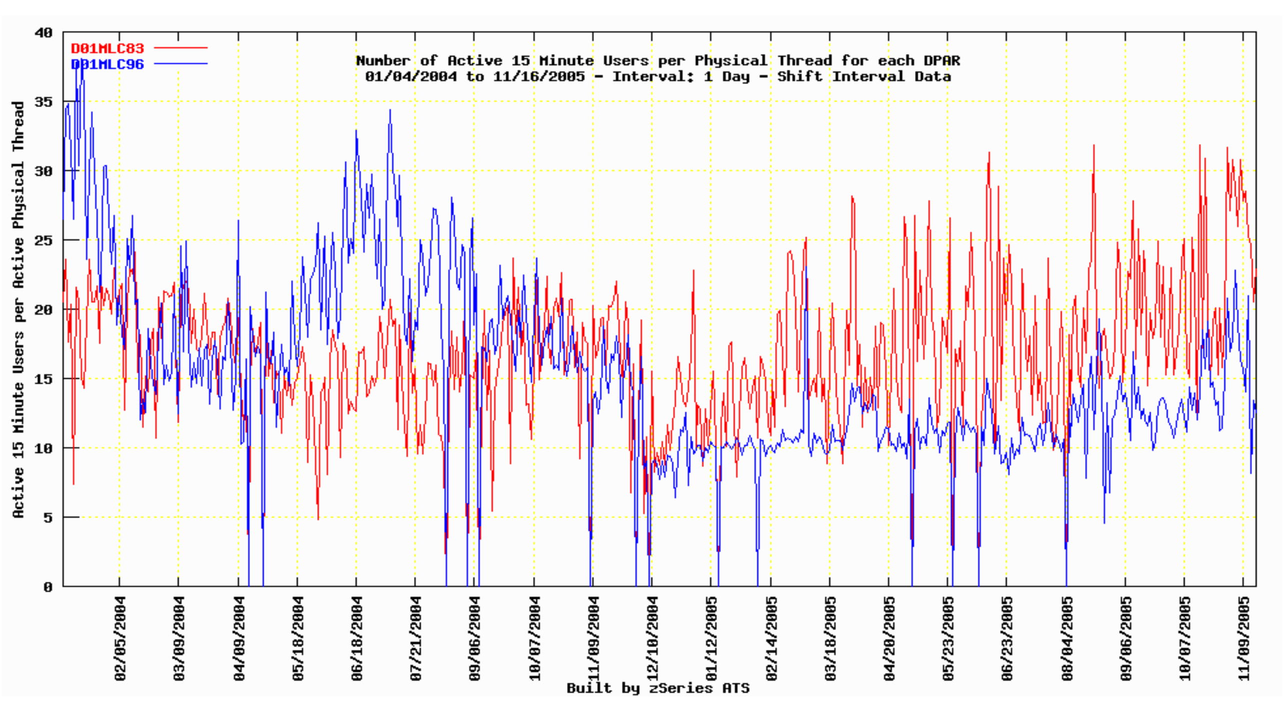


Figure 6-14 Number of active 15 minute users per physical thread

For HTTP users, the following is the starting set of costs-per-unit measurements that you can use.

- CPU cost per HTTP user (see Figure 6-15 on page 68). This is the total number of CPU seconds for all the processes of a DPAR divided by the number of unique IP addresses that access the server in this period. This statistic will give you an idea of what it cost per IP connection to support your HTTP environment. A change in the statistic will indicate that it is more costly (or less) per active user to support your HTTP workload. This can give you some good trending information as to what your users are doing and how much resource you may need to support them.

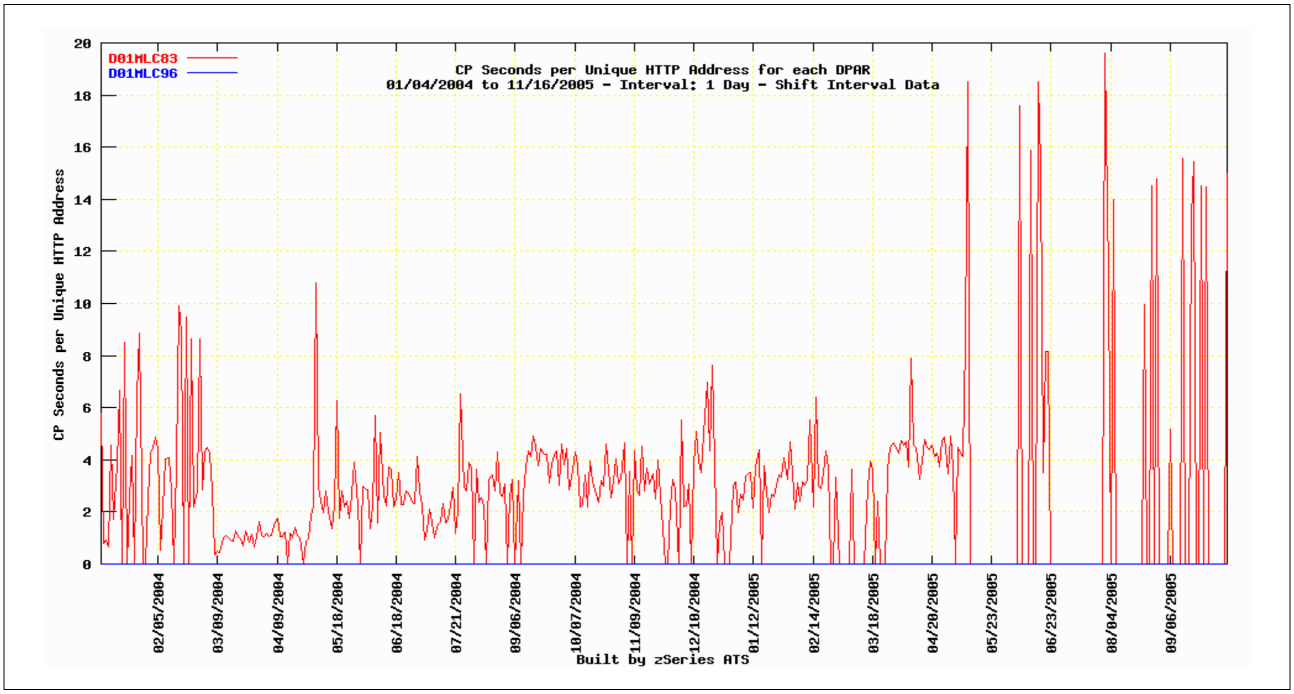


Figure 6-15 CPU cost per HTTP user

- Number of 108 HTTP reads per 108 HTTP writes (see Figure 6-16). This is the number of 108 HTTP reads divided by the number of 108 HTTP writes for this sample period. This gives you an indication of the type of work that is flowing into and out of your system. A change would indicate that your users are doing something different.

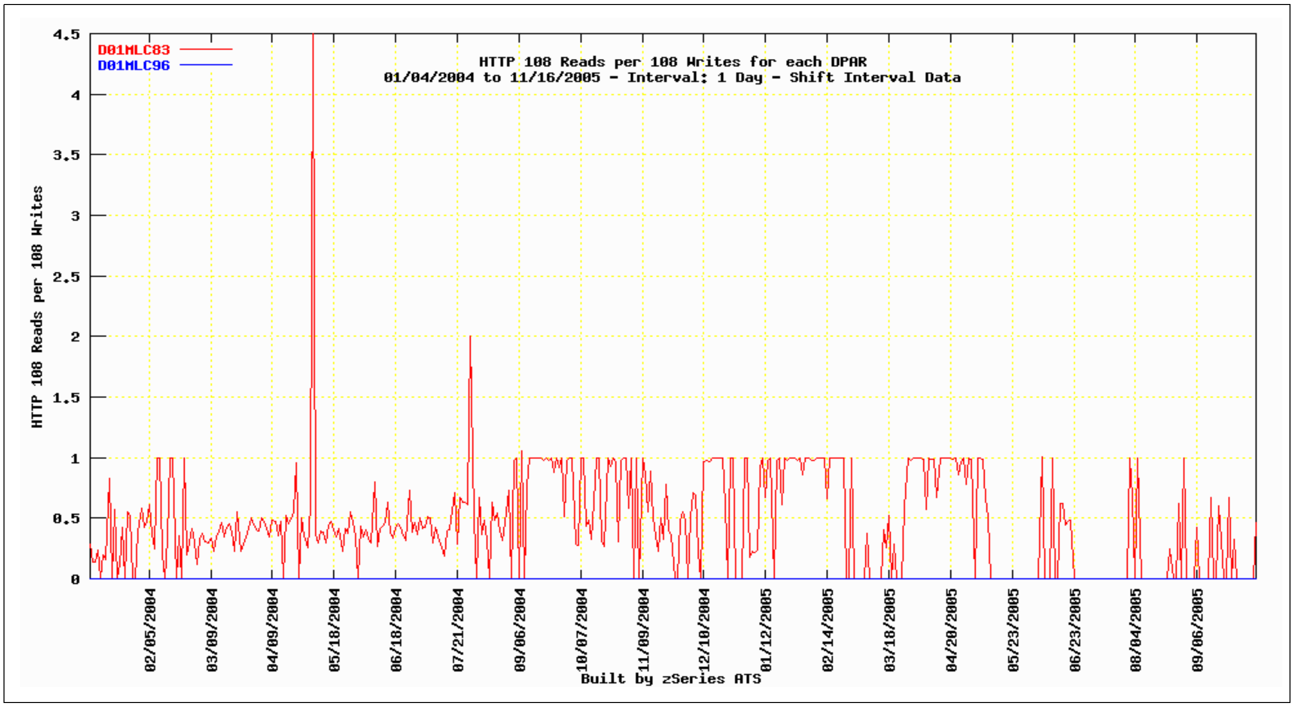


Figure 6-16 Number of 108 HTTP reads per 108 HTTP writes

- Number of 108 HTTP reads per HTTP user (see Figure 6-17). This is the number of 108 HTTP reads divided by the number of unique IP addresses during the sample period. This statistic gives you an idea of how many HTTPs reads each of your active users is performing. As this statistic changes, the amount of work that each HTTP user is performing will change accordingly.

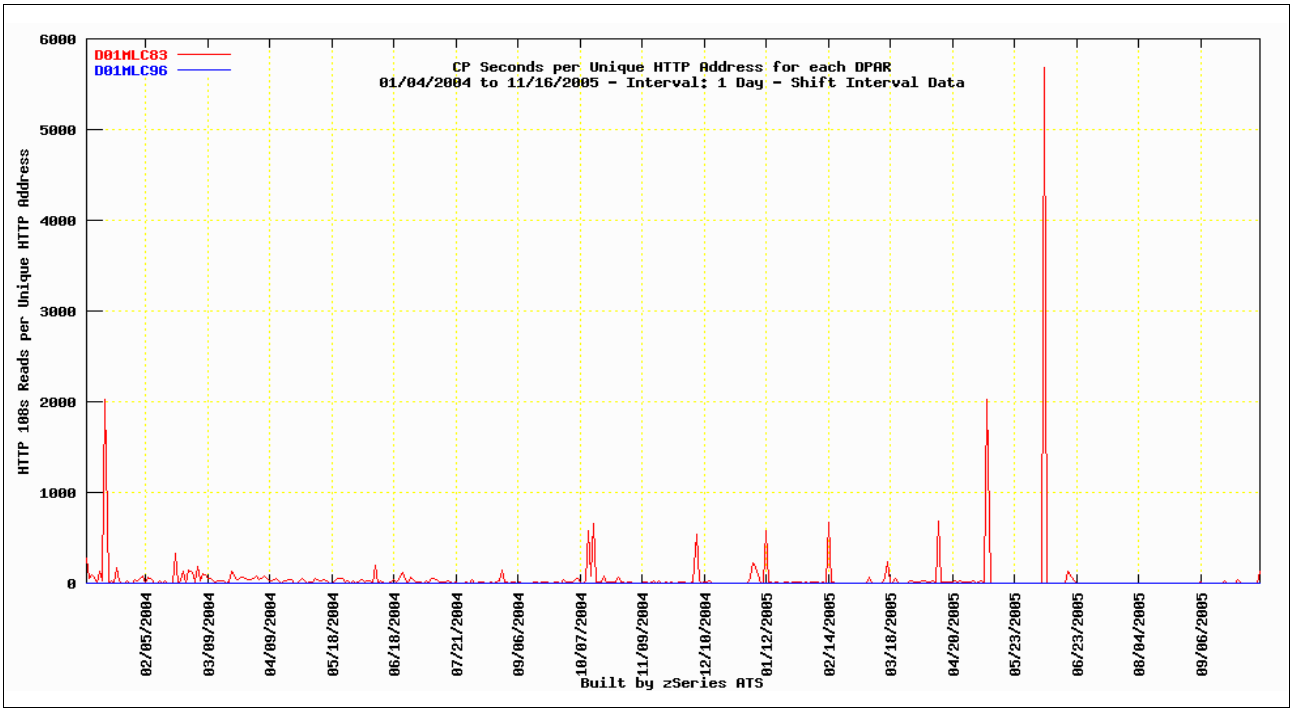


Figure 6-17 Number of 108 HTTP reads per HTTP user

Part 2

Linux update

In Part 2, we discuss performance information that is helpful when supporting Domino for Linux on the System z9 platform.

7

Monitoring with Performance Toolkit for VM

This chapter provides guidance on monitoring the performance of Domino servers running Linux guests under z/VM. There are many tools available to monitor the performance of servers running Domino, some at the Domino level and others at the operating system level. This chapter discusses monitoring at the operating system level using the Performance Toolkit for VM.

7.1 Purpose

Performance Toolkit for VM is a global performance product that provides real-time and historical analysis for z/VM systems. Monitoring the performance of Domino on Linux guests from z/VM requires collecting measurements of resources used by the z/VM LPAR as well as resources used by Linux guests and Domino servers. In our implementation we installed the Linux code from SuSE SLES 9.

A Domino server runs as a set of multiple processes according to the number of activated services/tasks. Some of these processes, such as server and router, have multiple threads to handle end user connections and route the messages. Until Linux Kernel 2.6, most of the tools and commands were not able to make a distinction between processes and threads. For Domino processes handling many threads, such as the server task, it was not easy to get a simple view of resource usage. With Linux Kernel 2.6, tools and commands are now able to show the resources used by processes, making easier to analyze in detail resource used by Domino server tasks. With Domino 7 on SLES 9 and using Linux performance monitoring tools described below, we can retrieve the CPU and memory used by each Domino task.

In this chapter, we will focus on monitoring the performance of Linux guests running Domino in terms of *real time display* and *history* from Performance Toolkit for VM on z/VM V5. We cannot cover all Performance Toolkit functions. For complete product information, refer to the following documentation on:

http://www.vm.ibm.com/library/

- *z/VM Performance Toolkit Version 5 Release 1*
- IBM Redbook: *Performance Toolkit for VM,* SG24-6059

7.2 Starting Performance Toolkit for VM

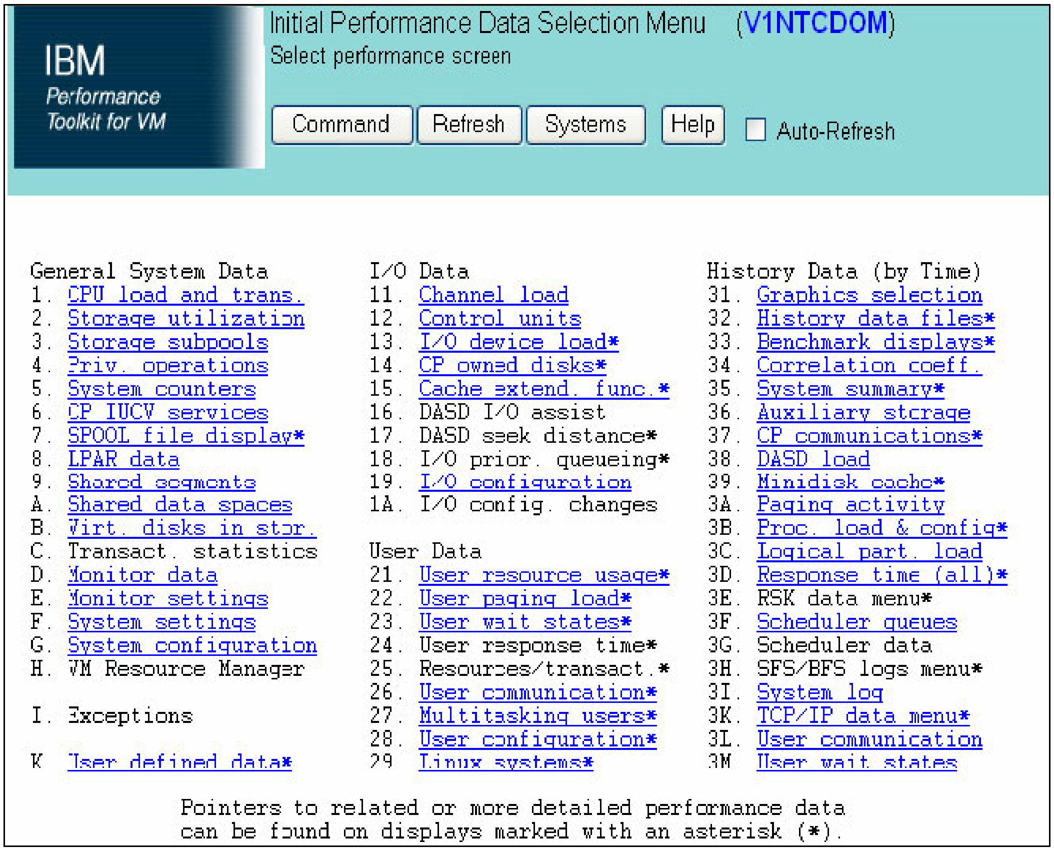
Performance Toolkit for VM is enabled to run by default in the PERFSVM virtual machine logged on or disconnected, and is started using PERFKIT command at PERFSVM user logon. The MONITOR command switches Performance Toolkit into performance monitoring mode and shows the performance data selection menu when using 3270 mode display.

Performance Monitoring Mode displays performance data from CPU, I/O and Users. Data is collected for local and remote machines. It provides real time performance analysis of z/VM systems. The performance data collection has two basic options:

- Data collection into temporary work buffer only
- Data collection into temporary work buffer *and* into history files on disk

The first choice would only allow you to analyze data in real time, while the second method also provides history data that can be used for trend analysis and capacity planning. Scan mode can be entered to display performance data accumulated on some history files.

A Web interface (see Figure 7-1 on page 75) allows retrieval and displays of performance data using a standard browser in addition to 3270 mode display.



Initial Performance Data Selection Menu (V1NTCDOM)

Select performance screen

```
General System Data          I/O Data                     History Data (by Time)
1. CPU load and trans.       11. Channel load             31. Graphics selection
2. Storage utilization       12. Control units            32. History data files*
3. Storage subpools          13. I/O device load*         33. Benchmark displays*
4. Priv. operations          14. CP owned disks*          34. Correlation coeff.
5. System counters           15. Cache extend. func.*     35. System summary*
6. CP IUCV services          16. DASD I/O assist          36. Auxiliary storage
7. SPOOL file display*       17. DASD seek distance*      37. CP communications*
8. LPAR data                 18. I/O prior. queueing*     38. DASD load
9. Shared segments           19. I/O configuration        39. Minidisk cache*
A. Shared data spaces        1A. I/O config. changes      3A. Paging activity
B. Virt. disks in stor.                                   3B. Proc. load & config*
C. Transact. statistics      User Data                    3C. Logical part. load
D. Monitor data              21. User resource usage*     3D. Response time (all)*
E. Monitor settings          22. User paging load*        3E. RSK data menu*
F. System settings           23. User wait states*        3F. Scheduler queues
G. System configuration      24. User response time*      3G. Scheduler data
H. VM Resource Manager       25. Resources/transact.*     3H. SFS/BFS logs menu*
                             26. User communication*      3I. System log
I. Exceptions                27. Multitasking users*      3K. TCP/IP data menu*
                             28. User configuration*      3L. User communication
K. User defined data*        29. Linux systems*           3M. User wait states

            Pointers to related or more detailed performance data
            can be found on displays marked with an asterisk (*).
```

Figure 7-1 Web interface Main selection menu

7.3 Real time monitoring

On the Initial Performance Data Selection Menu, submenus grouped under *General System Data*, *I/O Data* and *User Data* sections provide real time performance display. In the *User Data* section, the *Linux systems** submenu allows you to access Linux monitor information described below.

7.3.1 Linux systems performance monitoring

Linux internal performance data is collected inside the Linux operating system. A data-gathering mechanism is running in Linux and the data is retrieved by a VM performance monitor. **Caution**: Since Linux is monitoring virtual resources, such as processors, it may report more CPU cycles than it really uses. This can be adjusted with *Users Data* reports.The Performance Toolkit can generate Linux internal data reports from two different sources:

- Linux Monitor Stream

  Summary of Linux performance data gathered via APPLMON. The data can also be saved in history files on z/VM.

- RMF PMS for Linux

  Detailed Linux performance data gatherer and server. Data is available through a TCP/IP interface. The Performance Toolkit only uses the information to prepare detailed performance displays for selected Linux systems on request. History data is kept in the Linux file system. None of it is saved in history files on z/VM.

7.3.2 Linux monitor stream for z/VM

A Linux guest exports performance data about CPU, memory, and networks into “APPLDATA” monitor records.

Installation and activation on z/VM

To activate monitoring of Linux performance data, customization is required in z/VM for each Linux guest and PERFSVM guest.

User directory: An option statement in *user directory* must include APPLMON to each Linux guest.

```
OPTION APPLMON
```

PERFSVM machine: Edit PROFILE EXEC A and verify or add the APPLDATA monitor cp command.

```
'CP MONITOR SAMPLE ENABLE PROCESSOR'
'CP MONITOR SAMPLE ENABLE STORAGE'
'CP MONITOR SAMPLE ENABLE USER ALL'
'CP MONITOR SAMPLE ENABLE I/O ALL'
'CP MONITOR SAMPLE ENABLE APPLDATA ALL'
'CP MONITOR EVENT ENABLE STORAGE'
'CP MONITOR EVENT ENABLE I/O ALL'
'CP MONITOR SAMPLE INTERVAL 5 MINUTES'
```

Installation and activation on a Linux guest

To get APPLDATA, monitor stream modules must be loaded. A patch is required on SLES 8. Modules are built in kernel but not loaded on SLES 9. The **lsmod** command can be used to check if following modules are loaded.

```
appldata_os
appldata_net_sum
appldata_mem
```

If modules are not loaded, use **modprobe** command to load them:

```
modprobe appldata_os
modprobe appldata_net_sum
modprobe appldata_mem
```

After the modules have been loaded, the directory /proc/sys/appldata contains files: timer, interval, mem,net_sum, os.

To set the collect interval time, use an *echo* command. The *'time_value'* represents the sample interval, expressed in seconds, of virtual CPU time used by Linux before refreshing the data collected by the CP monitor.

```
echo ‘time_value’ > /proc/sys/appldata/interval
```

To activate the Monitor Stream, enter:

```
echo 1 > /proc/sys/appldata/mem
echo 1 > /proc/sys/appldata/os
echo 1 > /proc/sys/appldata/net_sum
echo 1 > /proc/sys/appldata/timer
```

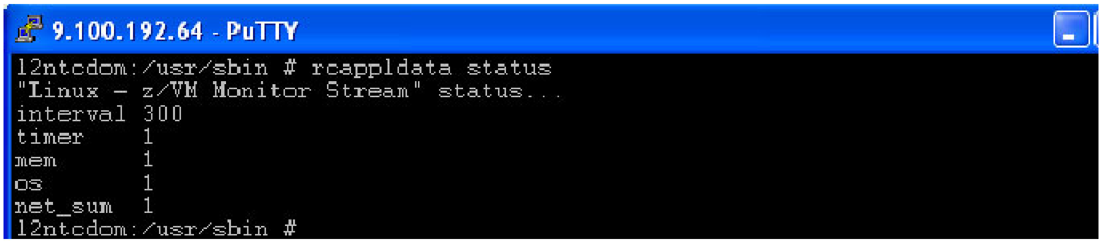
To deactivate the Monitor Stream, enter:

```
echo 0 > /proc/sys/appldata/timer
```

A startup command script is available with SLES 9 to automate load and activation:

```
/usr/sbin/rcappldata
```

The configuration file associated with the script is: /etc/sysconfig/appldata.

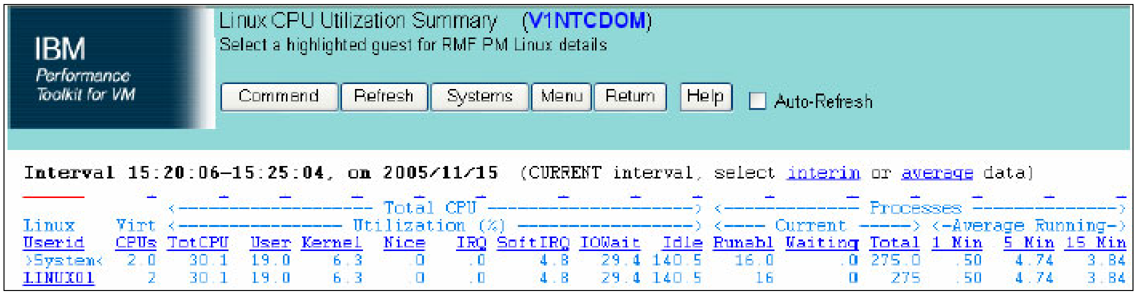


```
l2ntcdom:/usr/sbin # rcappldata status
"Linux - z/VM Monitor Stream" status...
interval 300
timer    1
mem      1
os       1
net_sum  1
l2ntcdom:/usr/sbin #
```

Figure 7-2 rcappldata, arguments: start, stop, status, restart and reload

Displaying a summary of Linux performance

Upon activation, three performance menus are available under the Data Selection menu, option 29. They provide CPU, memory, and networking activity information. See Figure 7-3 through Figure 7-5.

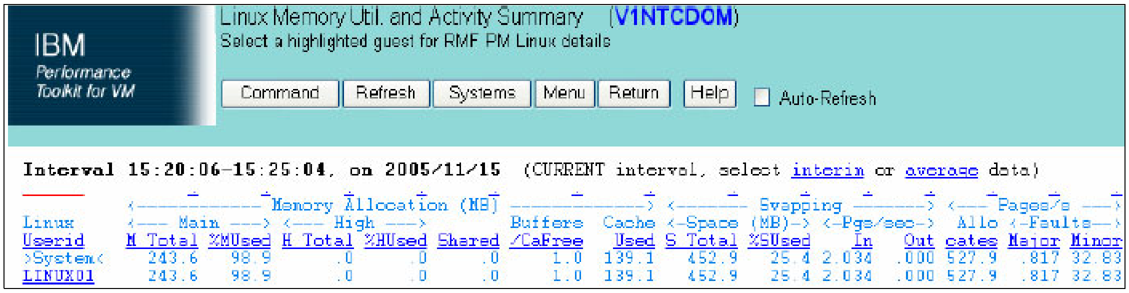


Linux CPU Utilization Summary (V1NTCDOM)

Select a highlighted guest for RMF PM Linux details

```
Interval 15:20:06-15:25:04, on 2005/11/15  (CURRENT interval, select interim or average data)

                <------------------- Total CPU --------------------> <------------- Processes --------------->
Linux     Virt  <--------------- Utilization (%) ------------------> <---- Current -----> <-Average Running->
Userid    CPUs TotCPU  User Kernel  Nice   IRQ SoftIRQ IOWait  Idle Runabl Waiting Total 1 Min  5 Min 15 Min
>System<   2.0   30.1  19.0    6.3    .0    .0     4.8   29.4 140.5   16.0      .0 275.0   .50   4.74   3.84
LINUX01      2   30.1  19.0    6.3    .0    .0     4.8   29.4 140.5     16       0   275   .50   4.74   3.84
```

Figure 7-3 Summary CPU activity display

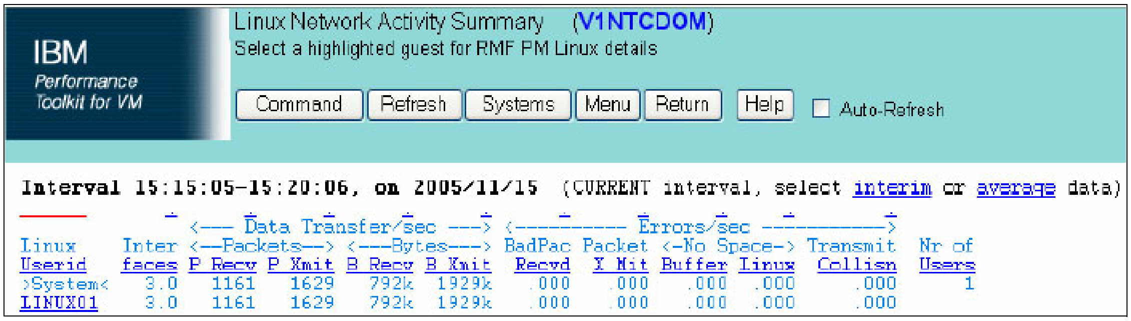


Linux Memory Util. and Activity Summary (V1NTCDOM)

Select a highlighted guest for RMF PM Linux details

```
Interval 15:20:06-15:25:04, on 2005/11/15  (CURRENT interval, select interim or average data)

          <------------- Memory Allocation (MB) -------------> <-------- Swapping --------> <--- Pages/s --->
Linux     <--- Main ---> <--- High --->        Buffers  Cache <-Space (MB)-> <-Pgs/sec->  Allo <-Faults-->
Userid    M Total %MUsed H Total %HUsed Shared /CaFree   Used S Total %SUsed    In   Out  cates Major Minor
>System<    243.6   98.9      .0     .0     .0     1.0  139.1   452.9   25.4 2.034  .000  527.9  .817 32.83
LINUX01     243.6   98.9      .0     .0     .0     1.0  139.1   452.9   25.4 2.034  .000  527.9  .817 32.83
```

Figure 7-4 Summary memory utilization and activity display



Linux Network Activity Summary (V1NTCDOM)

Select a highlighted guest for RMF PM Linux details

```
Interval 15:15:05-15:20:06, on 2005/11/15  (CURRENT interval, select interim or average data)

                <--- Data Transfer/sec ---> <----------- Errors/sec ----------->
Linux     Inter <--Packets--> <---Bytes---> BadPac Packet <-No Space-> Transmit  Nr of
Userid    faces P Recv P Xmit B Recv B Xmit  Recvd  X Mit Buffer Linux   Collisn  Users
>System<    3.0   1161   1629   792k  1929k   .000   .000   .000  .000      .000      1
LINUX01     3.0   1161   1629   792k  1929k   .000   .000   .000  .000      .000
```

Figure 7-5 Summary network activity display

7.3.3 RMF PM server

The RMF PMS is a separate tool, not part of any IBM product. It was originally written for use with the RMF PM client to gather and analyze data. The tool must be installed and active on a Linux guest so that it can monitor from z/VM or from an RMF PM client application. Performance data is permanently collected on the Linux guests themselves, and history data is stored in the file systems of the Linux guests. The RMF PM client application, running on a workstation, can gather real-time and historical performance data from Linux to generate graphical reports or store this data in spreadsheet format. z/VM PERFKIT collects this Linux performance data only in real time. The data is not recorded in z/VM history files. Data retrieval is based on the Distributed Data Server (DDS) interface and using requests sent to the Linux systems through TCP/IP on port 8803. The following reports are provided:

- CPU utilization details
- Memory utilization and activity details
- Network activity (overall and by device)
- File system size and utilization

Installation on Linux guests

You can implement the latest level of the RMF PM Server code on each of the Linux systems to be monitored. The RMF PMS tar file and install instructions can be downloaded from:

http://www.ibm.com/servers/eserver/zseries/zos/rmf/rmfhtmls/pmweb/pmlin.html

The collect interval can be set to same value as sample interval in Performance Toolkit in file: rmfpms/.rmfpms_config. Here we set the interval to 5 minutes:

```
IBM_PERFORMANCE_MINTIME=300
```

Registering and monitoring Linux guests using DDS on z/VM

Register the Linux guest by updating the FCONX LINUXUSR file

```
*LINUX-ID IP ADDRESS FOR DDS INTERFACE:PORT
LINUX00 9.100.192.108:8803
LINUX01 9.100.192.64:8803
```

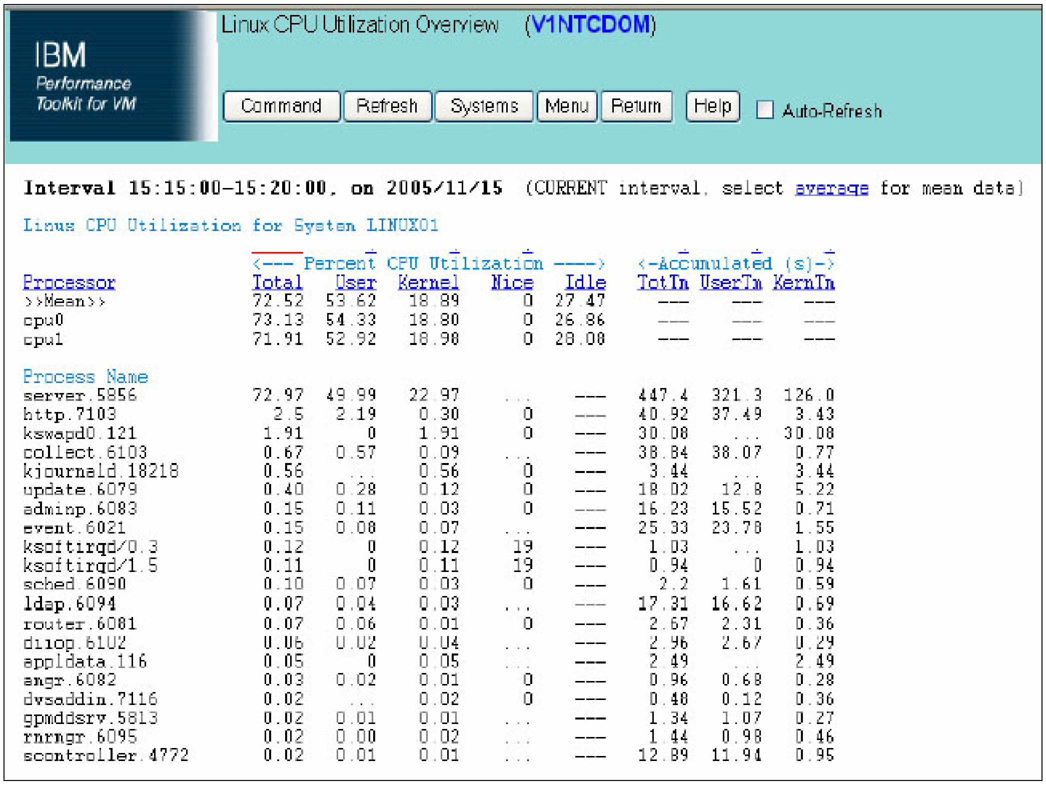
To activate data collection in FCONX $PROFILE file:

```
FC MONCOLL LINUXUSR ON
```

With Linux kernel 2.6, RMF PMS is able to distinguish between threads and processes. With kernel 2.4, that was not true, making analysis of resource usage per process more difficult due to the large number of threads handled by a Domino server and gathered by RMF PM Server.

Displaying detailed Linux performance

Using real-time monitor and when drilling down option 29 (*Linux systems*), new menus show detailed Linux performance data gathered by RMF PMS. See Figure 7-6 on page 79. Domino provides a wide range of services. With Domino 7 on kernel 2.6, we can get a better view and understanding of resources used by each service/process.

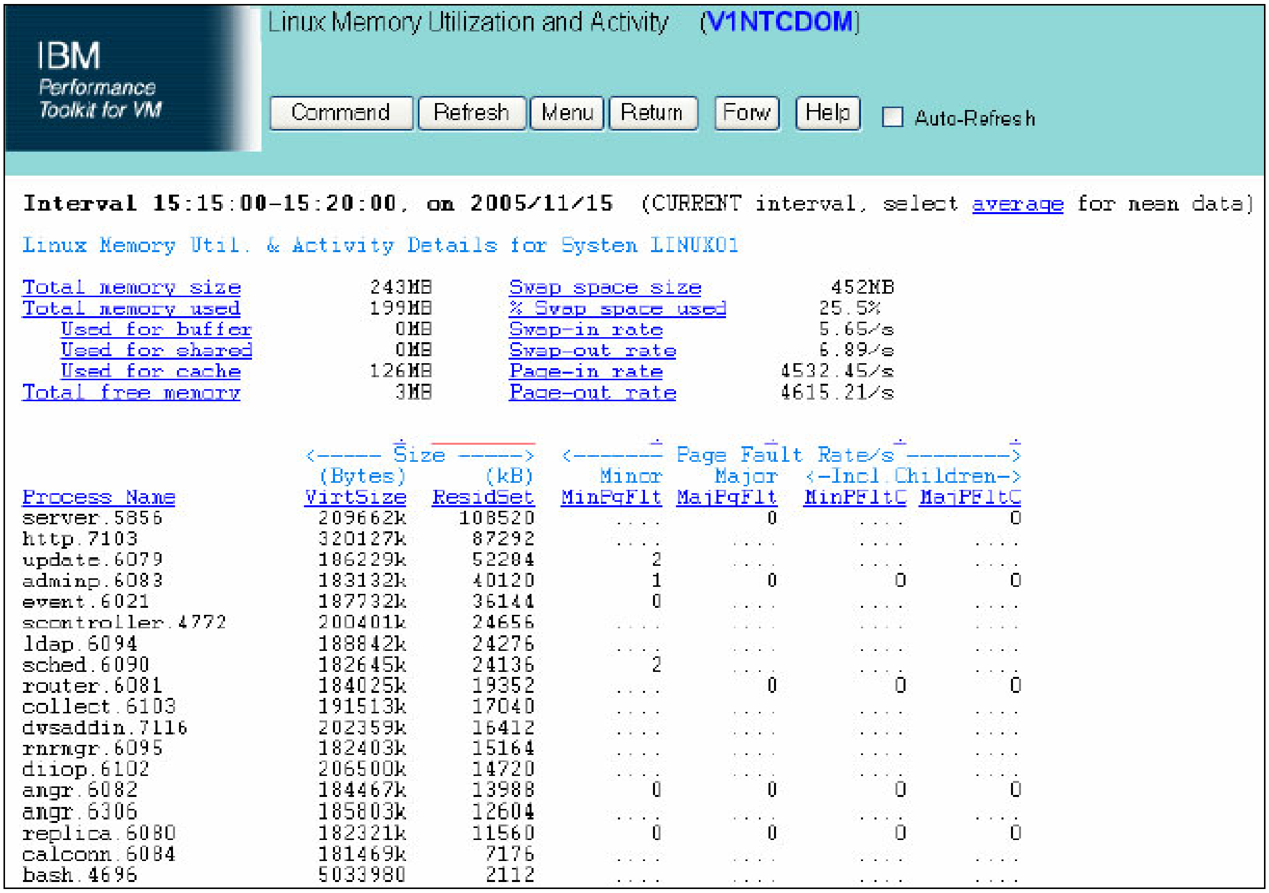


Linux CPU Utilization Overview (V1NTCDOM)

```
Interval 15:15:00-15:20:00, on 2005/11/15  (CURRENT interval, select average for mean data)

Linux CPU Utilization for System LINUX01

                      <--- Percent CPU Utilization ---->  <-Accumulated (s)->
Processor             Total   User  Kernel   Nice  Idle   TotTm UserTm KernTm
>>Mean>>              72.52  53.62   18.89      0 27.47     ---    ---    ---
cpu0                  73.13  54.33   18.80      0 26.86     ---    ---    ---
cpu1                  71.91  52.92   18.98      0 28.08     ---    ---    ---

Process Name
server.5856           72.97  49.99   22.97    ...   ---   447.4  321.3  126.0
http.7103               2.5   2.19    0.30      0   ---   40.92  37.49   3.43
kswapd0.121            1.91      0    1.91      0   ---   30.08    ...  30.08
collect.6103           0.67   0.57    0.09    ...   ---   38.84  38.07   0.77
kjournald.18218        0.56    ...    0.56      0   ---    3.44    ...   3.44
update.6079            0.40   0.28    0.12      0   ---   18.02   12.8   5.22
adminp.6083            0.15   0.11    0.03      0   ---   16.23  15.52   0.71
event.6021             0.15   0.08    0.07    ...   ---   25.33  23.78   1.55
ksoftirqd/0.3          0.12      0    0.12     19   ---    1.03    ...   1.03
ksoftirqd/1.5          0.11      0    0.11     19   ---    0.94      0   0.94
sched.6090             0.10   0.07    0.03      0   ---     2.2   1.61   0.59
ldap.6094              0.07   0.04    0.03    ...   ---   17.31  16.62   0.69
router.6081            0.07   0.06    0.01      0   ---    2.67   2.31   0.36
diiop.6102             0.06   0.02    0.04    ...   ---    2.96   2.67   0.29
appldata.116           0.05      0    0.05    ...   ---    2.49    ...   2.49
amgr.6082              0.03   0.02    0.01      0   ---    0.96   0.68   0.28
dvsaddin.7116          0.02    ...    0.02      0   ---    0.48   0.12   0.36
gpmddsrv.5813          0.02   0.01    0.01    ...   ---    1.34   1.07   0.27
rnrmgr.6095            0.02   0.00    0.02    ...   ---    1.44   0.98   0.46
scontroller.4772       0.02   0.01    0.01    ...   ---   12.89  11.94   0.95
```

Figure 7-6 CPU utilization details

Figure 7-7 on page 80 shows the memory allocation and activity for the Linux. It includes allocation per Domino process and page fault rate from Linux kernel perspective. Major means disk access and Minor means no disk access.


IBM
Performance Toolkit for VM
Linux Memory Utilization and Activity (V1NTCDOM)
Command
Refresh
Menu
Return
Forw
Help
□ Auto-Refresh

```
Interval 15:15:00-15:20:00, on 2005/11/15  (CURRENT interval, select average for mean data)

Linux Memory Util. & Activity Details for System LINUX01

Total memory size          243MB      Swap space size        452MB
Total memory used          199MB      % Swap space used      25.5%
   Used for buffer           0MB      Swap-in rate           5.65/s
   Used for shared           0MB      Swap-out rate          6.89/s
   Used for cache          126MB      Page-in rate         4532.45/s
Total free memory            3MB      Page-out rate        4615.21/s

                          <----- Size ----->  <------- Page Fault Rate/s -------->
                           (Bytes)      (kB)     Minor     Major  <-Incl.Children->
Process Name              VirtSize  ResidSet  MinPgFlt  MajPgFlt  MinPFltC  MajPFltC
server.5856               209662k    108520      ....         0      ....         0
http.7103                 320127k     87292      ....      ....      ....      ....
update.6079               186229k     52284         2      ....      ....      ....
adminp.6083               183132k     40120         1         0         0         0
event.6021                187732k     36144         0      ....      ....      ....
scontroller.4772          200401k     24656      ....      ....      ....      ....
ldap.6094                 188842k     24276      ....      ....      ....      ....
sched.6090                182645k     24136         2      ....      ....      ....
router.6081               184025k     19352      ....         0         0         0
collect.6103              191513k     17040      ....      ....      ....      ....
dvsaddin.7116             202359k     16412      ....      ....      ....      ....
rnrmgr.6095               182403k     15164      ....      ....      ....      ....
diiop.6102                206500k     14720      ....      ....      ....      ....
amgr.6082                 184467k     13988         0         0         0         0
amgr.6306                 185803k     12604      ....      ....      ....      ....
replica.6080              182321k     11560         0         0         0         0
calconn.6084              181469k      7176      ....      ....      ....      ....
bash.4696                 5033980      2112      ....      ....      ....      ....
```

Figure 7-7 Memory utilization details

7.4 History data reporting for the toolkit

7.4.1 History, trend and report files

A large amount of data can be recorded to disk files then post-processed for performance analysis and capacity planning. Here is a description of these files:

Simple history data files

These history files contain only overall system data records.

- *yyyymmdd HISTLOG A*. Detailed data files contain one record per monitor sample interval
- *ACUM HISTSUM A*. Summary data files. They contain fields as HISTLOG but is updated once per hour.

Benchmark display files

The performance of a subset of users and/or I/O devices can be more closely investigated. On our system we activated Benchmark display for LINUX01, which is the Linux guest running the Domino server. Detailed performance data files for LINUX01:

- *LINUX01 USTATLG1 A.* Wait state data log
- *LINUX01 USERLOG1 A.* Resource usage log
- *LINUX01 UCOMMLG1 A.* Communication data log

- *LINUX01 UPAGELG1 A*. Page data log

Summary Linux data performance log files. These files are available only when Linux Monitor Stream is active.

- *LINUX01 LXNETLG1 A.* Linux Network activity log
- *LINUX01 LXCPULG1 A.* Linux CPU load log
- *LINUX01 LXMEMLG1 A.* Linux Memory Utilization log

Detailed performance data files for I/O can also be created.

Extended Sum and Trend files

Extended Summary and Trend files contain data records with general system load information, and can also contain more specific records like ‘Summary Linux performance data’.

- *filename FCXSUM A.* Default, can changed using FC SET SUM fn ft fm
- *filename FCXTREND A.* Default, can be changed using FC SET TRD fn ft fm

VM Monitor data files

The MONWRITE utility collects monitor records and stores them in the file specified in the command. Records are written at the frequency defined by monitor sample interval. This is the most detailed history file.

Print reports

Using the PRINT command, the submenus can be saved to a file.

- *filename LISTING A.* Default, can be changed using FC SET REP fn ft fm

7.4.2 Collecting history data

Histlog and Histsum

These give global performance metrics of applications and users running on this LPAR. Information such as the CPU used by the LPAR, Paging rates, global DASD response time can be retrieved from these files. The writing to these files is activated by using the command:

```
FC MONCOLL PERFLOG ON
```

FC Benchmark display

FC benchmark display provides performance history of users. These history files are useful to analyze performance metrics of specifics Linux guest handling one or more Domino servers. The metrics collected include the Linux APPLMON data. Collect can be activated using following command:

```
FControl BENCHmark USEr user ID FILe hh:mm to hh:mm
```

Extended Trend and Sum data

FCONX TRENDREC and FCONX SUMREC files controls creation of records for extended Summary and Trend files and are located on 5VMPTK10 1CC disk. The period when data is written to disk can be controlled using the commands:

```
FC MONCOLL RESET hh:mmR_S hh:mmS hh:mmR_S hh:mmS
FC MONCOLL RESET hh:mmR_T hh:mmT (merge
```

Where, *R* means reset data counters, and *merge* means that actions specified for a specific time are to be merged with previously specified action. These command can be added to FCONX $PROFILE. To read the extended Summary and Trend files on PERFSVM, the monitor should be stopped using: FC MONCOLL OFF.

To read these extended files enter:

```
TRNDSCAN systemid FCXSUM A
TRNDSCAN systemid FCXTREND A
```

The same Performance Screen Selection Menu as when entering MONITOR appears on display, but with *Trend* on upper right corner to indicate that data is retrieved from Trend file.

If defaults are not used, the hh:mmR creates the new file as defined by the command:

```
FC SETTINGS TRDFILID filename FCXTREND A
```

The Linux APPLMON data can be scanned when using FCXSUM and FCXTREND files.

To navigate through the different interval samples use the following commands:

```
NEXTSAMP
SKIPSAMP
```

VM Monitor data

A default MONWRITE user ID can be logged on to start/stop MONWRITE utility:

```
MONWRITE MONDCSS *MONITOR DISK fn ft fm
MONWSTOP
```

This monitor data file can be scanned on PERFSVM using command:

```
MONSCAN fn ft fm
```

The Monitor should be stopped to enter in MONSCAN. The same Performance Screen Selection Menu as when entering MONITOR appears on display, but with *Monitor Scan* in upper right corner to indicate that data is retrieved from VM Monitor data.

Printing monitor screen on a file

Performance Screen displays can be saved using a PRINT command on file xxxxxxx LISTING A1. File name can be specified with FC command:

```
FC SETTINGS REPFILID fn ft fm
```

Prints can be scheduled using command as shown in the example below:

```
FC MONCOLL RESET 10:00R_P 10:59P 11:00R_P 11:59P (merge
```

These previous FC commands can be stored in FCONX $PROFILE file.

FCONX REPORT file defines which part of the collected performance data to print. The followings settings should be inserted in FCONX REPORT for each Domino Linux Guest, to get the LINUX Monitor Stream and RMF PMS data.

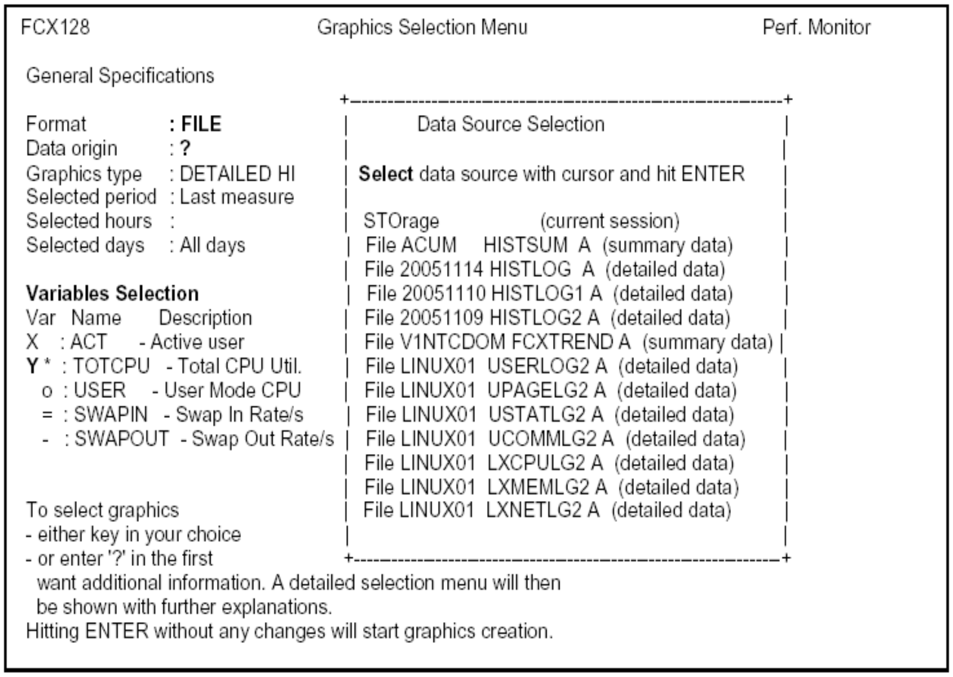
```
* LINUX userid
LINUX LINUX01
LXCPU LINUX01
LXFILSYS LINUX01
LXMEM LINUX01
LXNETWRK LINUX01
```

This is a way to get history-like data on specific Domino processes running, and to analyze their activity from z/VM.

7.5 Producing graphics and exporting history data

Up to four performance metrics under ‘Variable Selection’ can be selected from history data files. See Figure 7-8. You can then create graphics or flat files on z/VM that can be sent to a workstation and imported into a spreadsheet. The Format options are File, PLOT or GDDM.

FCONDATA IMPORT is the name of the saved data file that can be transferred to a workstation using FTP or PCOM.



```
FCX128                        Graphics Selection Menu                       Perf. Monitor

General Specifications
                               +-------------------------------------------------------+
Format          : FILE         |         Data Source Selection                         |
Data origin     : ?            |                                                       |
Graphics type   : DETAILED HI  | Select data source with cursor and hit ENTER          |
Selected period : Last measure |                                                       |
Selected hours  :              | STOrage               (current session)               |
Selected days   : All days     | File ACUM     HISTSUM  A  (summary data)              |
                               | File 20051114 HISTLOG  A  (detailed data)             |
Variables Selection            | File 20051110 HISTLOG1 A  (detailed data)             |
Var  Name      Description     | File 20051109 HISTLOG2 A  (detailed data)             |
X   : ACT     - Active user    | File V1NTCDOM FCXTREND A  (summary data)              |
Y * : TOTCPU  - Total CPU Util.| File LINUX01  USERLOG2 A  (detailed data)             |
  o : USER    - User Mode CPU  | File LINUX01  UPAGELG2 A  (detailed data)             |
  = : SWAPIN  - Swap In Rate/s | File LINUX01  USTATLG2 A  (detailed data)             |
  - : SWAPOUT - Swap Out Rate/s| File LINUX01  UCOMMLG2 A  (detailed data)             |
                               | File LINUX01  LXCPULG2 A  (detailed data)             |
                               | File LINUX01  LXMEMLG2 A  (detailed data)             |
To select graphics             | File LINUX01  LXNETLG2 A  (detailed data)             |
- either key in your choice    |                                                       |
- or enter '?' in the first    +-------------------------------------------------------+
  want additional information. A detailed selection menu will then
  be shown with further explanations.
Hitting ENTER without any changes will start graphics creation.
```

Figure 7-8   Generating files and graphics from history data

8

Networking updates

Domino server on Linux running on System z9 hardware, usually under z/VM, can be configured in two main networking models: real (OSA card) or virtual (z/VM guest LAN). However, other options are available. This chapter focuses on the Virtual Switch feature.

8.1 Network virtualization options and performance

Here are the options to consider when configuring a network for Linux guests under VM:

- **Point to Point connections**: CTC or IUCV. This may be a problem when managing lots of Linux guests.
- The **z/VM Guest LANs** (HiperSockets™ or QDIO) provide excellent bandwidth at very low CPU cost. For communications between guests within a z/VM LPAR, these are the most efficient options. The choice between a HiperSockets Guest LAN and a GbE Guest LAN is dictated more by function than by performance. There is a slight advantage to the HiperSockets Guest LAN in terms of shorter path length in some cases, and the larger available MTU may help.

  The **Linux qdio device driver** reserves approximately 8 MB of storage for each QDIO device, locked in real storage frames below the 2 GB line. This cost needs to be considered when running multiple Linux guests.

  Using **HiperSockets** is the best method for communicating between LPARs. This option provides IP transport connectivity between guest systems within or across logical partitions (LPARs) through an internal hardware LAN segment. HiperSockets is a microcode feature that is part of certain System z9 hardware.
- For those environments that employ a **router** to connect the Linux guests to the external network, there are several choices: use the z/VM TCP/IP stack or a Linux guest as a router, or eliminate the need for an internal router by using the **z/VM** Virtual Switch.

  When using the **Virtual Switch**, the router function is performed by CP. This means that the CPU time that would have been consumed by a router virtual machine is almost eliminated. This can result in a significant reduction in total system CPU time.

  The Virtual Switch does not provide the layer 3 routing that the z/VM or Linux Internal router provided. The Domino server on the Virtual Switch needs to be configured with a gateway (next hop) of an external router.

More performance information is available at z/VM Performance Report Web site:

http://www.vm.ibm.com/perf/reports/zvm/html

8.2 Virtual Switch

The Virtual Switch was introduced in z/VM V4R4. It was improved in z/VM V5R1 to provide enhanced failover support for less disruptive recovery for some common network failures. This helped ensure business continuity as well as infrastructure reliability and availability.

A z/VM Virtual Switch can be created as a special type of Guest LAN. In addition to providing a network of virtual adapters, it can be connected directly to an OSA Express QDIO adapter. This capability provides connectivity to external LAN segments without requiring an internal router.

8.2.1 VSWITCH controller

The VSWITCH connection to an OSA-Express interface is provided by a controller virtual machine. A controller is a z/VM service machine running the TCP/IP stack. At least one TCP/IP service machine must be configured to be a controller.

Unlike previous networking configurations that used the z/VM TCP/IP stack, there is no requirement to manually configure IP addresses or devices when using VSWITCH.

8.2.2 VSWITCH implementation

Here are the steps for installing a VSWITCH:

1. Define a VSWITCH to act as a LAN segment for the virtual machines.
2. Configure one or more z/VM TCP/IP virtual machines (guests) to act as controllers for the VSWITCH.
3. Create a simulated Network Interface Card (NIC) on each virtual machine.
4. Couple each virtual NIC to the VSWITCH.

More details can be found at:

http://www.redbooks.ibm.com/redpapers/pdfs/redp3901.pdf

Important: The Linux 2.6 kernel introduced a new device driver configuration interface, so instructions for configuring devices in Linux are different in the 2.4 and 2.6 kernels. The file /etc/chandev.conf is used to specify the parameters for the device driver in 2.4 kernel. With 2.6 kernel, the device driver creates files in the /sysfs filesystem when it is loaded. The driver and the devices controlled by it are configured by writing values into these files.

Domino network virtualization

Depending on your requirements, you may define Guest LANs, Virtual Switches, or a combination for your system.

A Guest LAN is a closed LAN where each virtual machine can communicate only with the other virtual machines in the same Guest LAN. To connect a Guest LAN to an external network, you must do one of the following:

- Use a router virtual machine that has connectivity to both the Guest LAN (using a virtual network adapter) and a real hardware network device.
- Deploy a Virtual Switch. This is a special-purpose Guest LAN that can be defined with a connection to a real OSA-Express network adapter. You can also define a Virtual Switch without the OSA-Express device, in which case it acts like a Guest LAN. With a Virtual Switch, the requirement for an internal router virtual machine is eliminated.

How can Domino take advantage of z/VM network virtualization?

Defining a guest LAN can help providing several virtual adapters. In this case, a router is needed to connect Notes connections or mail routing over the TCP/IP network. A Virtual Switch can replace the guest LAN and the router, each Linux guest defining its virtual NIC address and IP configuration. This is helpful when, for example, one z/VM partition hosts several Domino Linux guests; it might save some CPU cycles by suppressing the router.

Setting up a private LAN in a Domino clustering configuration is recommended to isolate cluster replication from the rest of the network traffic and prevent the cluster traffic from slowing down the primary network.

If the Domino servers participating in a cluster are located in the same z/VM partition, a Guest LAN can be defined and used as an internal private network dedicated to the cluster replication. Figure 8-1 on page 88 shows an example of a Domino cluster configuration. Each Domino server belongs to a Linux guest, all guests are hosted by the same z/VM. In that case, regular Notes client connections and replications go through the VSWITCH LAN (addresses 192.168.1.x, LAN system VSW1) and the cluster replication activity is handled by a dedicated Guest LAN (addresses 10.10.10.x, LAN system DOMCLUST).

If the Domino servers participating in a cluster are hosted by Linux guests defined in different z/VM partitions on same physical system, a Hipersocket LAN may be dedicated to the cluster replication.

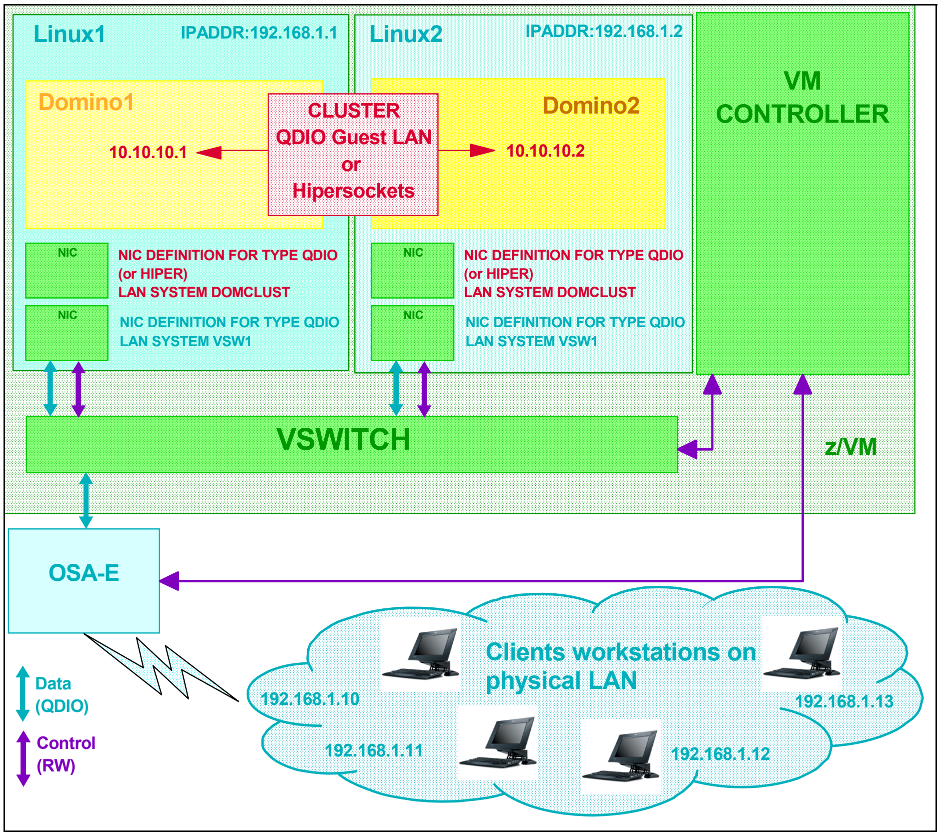

Linux1
IPADDR:192.168.1.1
Linux2
IPADDR:192.168.1.2
VM CONTROLLER
Domino1
10.10.10.1
CLUSTER QDIO Guest LAN or Hipersockets
Domino2
10.10.10.2
NIC
NIC DEFINITION FOR TYPE QDIO (or HIPER) LAN SYSTEM DOMCLUST
NIC
NIC DEFINITION FOR TYPE QDIO LAN SYSTEM VSW1
NIC
NIC DEFINITION FOR TYPE QDIO (or HIPER) LAN SYSTEM DOMCLUST
NIC
NIC DEFINITION FOR TYPE QDIO LAN SYSTEM VSW1
VSWITCH
z/VM
OSA-E
Clients workstations on physical LAN
192.168.1.10
192.168.1.11
192.168.1.12
192.168.1.13
Data (QDIO)
Control (RW)

Figure 8-1 Example of a Domino cluster in a single z/VM

9

Performance testing of Domino 7 for Linux on System z9

A short while ago, Lotus announced support of Red Hat Enterprise Linux 4.0 (RHEL 4) with Domino 7. To provide you with some basic performance data, the Domino for System z9 Development lab performed a series of tests to compare Domino 7 running with Rhel 4 versus Domino 7 with SLES 9. Using the same hardware and software configurations, we simulated NRPC (Notes) mail, DWA (Domino Web Access), and Enterprise mail workloads. The Enterprise mail workload is a new workload we use for performance testing, which comes closer to simulating a production workload than previous test workloads. No performance degradations were found with RHEL 4.

9.1 Overview

All performance test results that we have documented in this chapter come from one logical partition (LPAR) on an IBM @server System z9 990 model 2084-C24. The z990 has 24 CPs available, six of which are dedicated to the performance test LPAR. The remaining 18 CPs, as well as some other machine resources, were shared among thirteen other LPARs used for Domino development and test activities. For Entmail and NRPC tests, we used only three of the six CPUs to drive the load with higher CPU utilization. For DWA tests, we used six CPUs to drive the load with higher CPU utilization.

The performance test LPAR was configured with 12 GB memory. On SLES 9 and RHEL 4, we used 12GB total. One Domino partition was run on this LPAR for NRPC and DWA mail. Two Domino partitions were run on this LPAR for Entmail tests. We used two Gigabit Ethernet Open Systems Architecture (OSA) cards. Our LAN is isolated. All disks are allocated from an Enterprise Storage Server® (2105 Model 800) array with each disk configured as a 3390 model 3.

There were separate filesystems allocated on single volumes (disks) for the Domino execution, data (except for client mail databases), and the Domino address book (names.nsf). Two volumes in a logical volume manager (LVM) file system for transaction logging. Client mail databases were distributed evenly over 52 LVM file systems, each allocated across 5 volumes in a single LVM, providing 11.5 GB of usable space per file system. The EXT3 file system was used on Linux for System z9.

The operating systems installed were SLES 9 with SP1 or RHEL 4. We ran with transaction logging enabled and hardware compression of data instead of LZ1 software compression. This feature is only available with Domino 7 on System z9.

9.2 Hardware configurations

Table 9-1 shows the hardware system specifications that we used for our testing.

Table 9-1 Test specifications

Model	z990 2084-C24
CPUs	Three dedicated CPUs for NRPC, Six dedicated CPUs for DWA
Installed Memory	12 GB
DASD Type	2105 model 800, 3390 model 3 type volumes
File system	52 x 5 LVM mail DBs, 7 other volumes for notesdata, notesbin, NAB, mailbox, utility, and translog per domino partition
Operating System	SLES 9 SP1 / RHEL 4

Table 9-2 shows the changes we made to each server's Notes.ini file.

Table 9-2 Changes to Notes.ini parameters

Domino 6.5	Domino 7
TRANSLOG_Status=1	TRANSLOG_MaxSize=3000
TRANSLOG_Performance=1	NSF_BUFFER_POOL_SIZE_MB=256
Server_TransInfo_Range=9	cluster_replicators=9

Domino 6.5	Domino 7
UPDATE_QUEUE_ENTRY_MAX=20000	NSF_DBCACHE_MAX_CLEAN_HOLD_TIME=9999
ServerTasks=Replica,Router,Update,Adminp	TRANSLOG_Status=1
TRANSLOG_MaxSize=3000	TRANSLOG_Performance=1
NSF_BUFFER_POOL_SIZE_MB=256	Server_TransInfo_Range=9
cluster_replicators=4	UPDATE_QUEUE_ENTRY_MAX=20000
NSF_DBCACHE_MAX_CLEAN_HOLD_TIME=9999	ServerTasks=Replica,Router,Update,Adminp
FTUpdate_Idle_Time=25	Update_Idle_Time=25

Figure 9-1 shows the CPU improvement from Domino 7 on RHEL 4 versus Domino 7 on SLES 9 running a NRPC mail workload, implementing the Mail6 template from Domino 6.5.

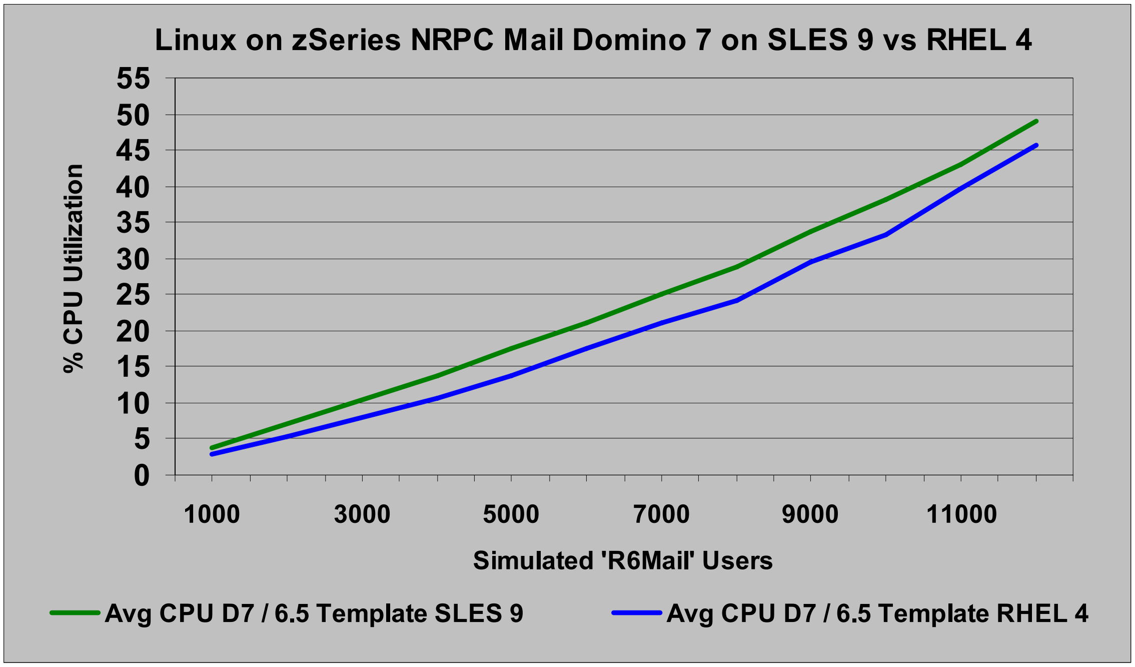


Figure 9-1 CPU improvement with NRPC

Figure 9-2 on page 92 shows the CPU improvement from Domino 7 on RHEL 4 versus Domino 7 on SLES 9 running a DWA mail workload, implementing the iNotes6 template from Domino 6.5.

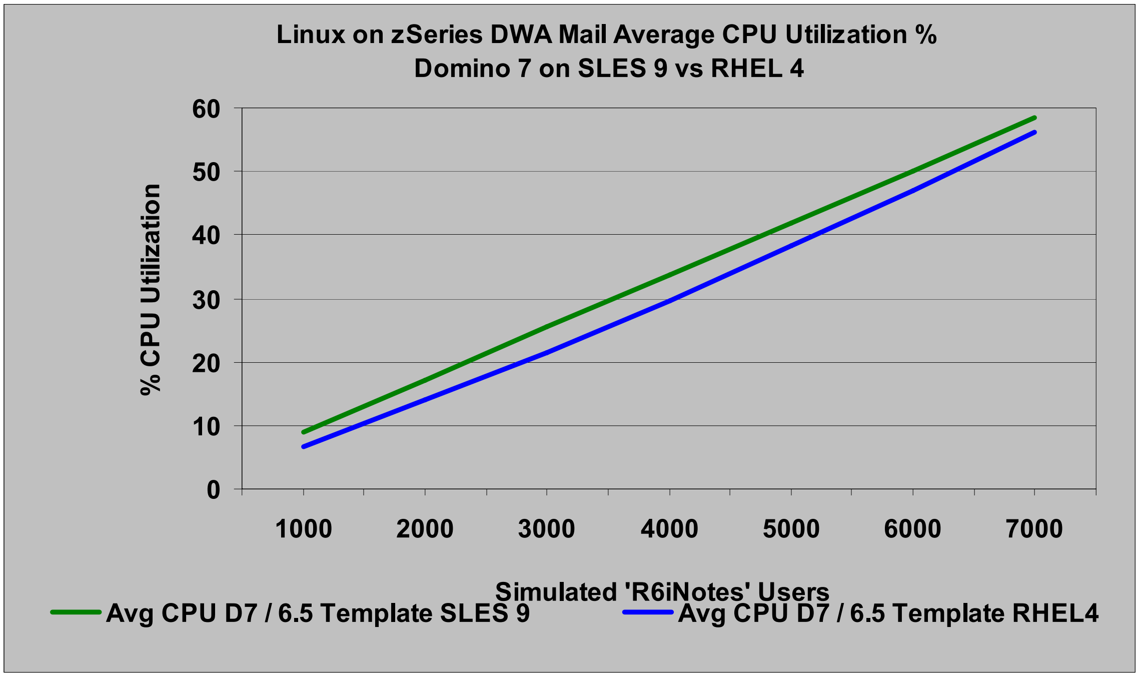


Figure 9-2 CPU improvement with DWA

Figure 9-3 shows the CPU for Domino 7 on SLES 9 versus Domino 7 on RHEL 4 running an Entmail workload, implementing the Mail6 template from Domino 6.5.

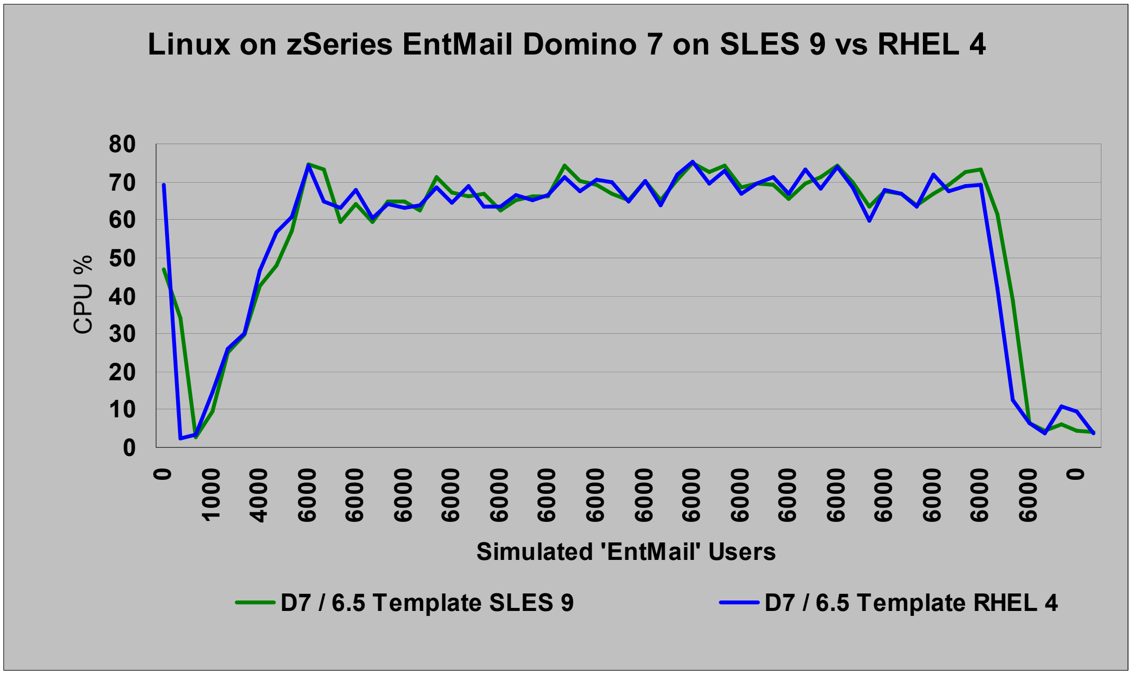


Figure 9-3 CPU improvement with EntMail

Table 9-3 on page 93 through Table 9-5 on page 93 display resource utilization numbers. Table 9-3 on page 93 shows Domino 7 users running the NRPC mail with Mail6 template on SLES 9 and RHEL 4.

Table 9-3 CPU improvement with NRPC

Resource	Domino 7 on SLES 9	Domino 7 on RHEL 4	Change (percent)
Average CPU percent at steady state at 12000 users	48.95	45.60	-6.85
NotesMark	16069	16049	
Response Time (ms)	83	77	-7.2

Table 9-4 shows Domino 7 users running the DWA mail with Mail6 template on SLES 9 and RHEL 4:

Table 9-4 CPU improvement with DWA

Resource	Domino 7 on SLES 9	Domino 7 on RHEL 4	Change (percent)
Average CPU percent at steady state at 7000 users	58.45	56.18	-3.88
NotesMark	5960	5962	
Response Time (ms)	106	104	-1.8

Table 9-5 shows Domino 7 users running the EntMail with Mail6 template on SLES 9 and RHEL 4.

Table 9-5 CPU improvement with EntMail

Resource	Domino 7 on SLES 9	Domino 7 on RHEL 4	Change (percent)
Average CPU percent at steady state at 6000 users	67.88	67.62	-0.38
NotesMark	9405	9407	
Response Time (ms)	109	84	-22.9

The tests showed some CPU improvement with NRPC mail and DWA mail workload. However, it did not show any CPU improvement with the Entmail workload. The result indicated the heavier the workload, the less CPU improvement on RHEL 4 compared with SLES 9. As a result, SLES 9 and RHEL 4 showed as fairly equal to each other.

These are the results from our initial testing. More in-depth testing is planned.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 97. Note that some of the documents referenced here may be available in softcopy only.

Linux

- *Linux on IBM @server zSeries and S/390: Performance Measurement and Tuning*, SG24-6926
- *Linux for S/390 and zSeries: ISP/ASP Solutions*, SG24-6299
- *Linux for zSeries and S/390: Large Scale Linux Deployment*, SG24-6824
- *Linux for S/390 and zSeries: Distributions*, SG24-6264
- *Linux for S/390*, SG24-4987
- *Linux on IBM @server zSeries and S/390: System Management,* SG24-6820

VM

- *Linux on IBM @server zSeries and S/390: Performance Toolkit for VM*, SG24-6059
- *Linux on IBM @server zSeries and S/390: TCP/IP Broadcast on z/VM Guest LAN,* REDP-3596
- *Linux on IBM @server zSeries and S/390: High Availability for z/VM and Linux*, REDP-0220
- *Linux on IBM @server zSeries and S/390: Building SuSE SLES8 Systems under z/VM,* REDP-3687
- *Partitioning DASD for Linux Guests Running under z/VM,* TIPS0277

Domino

- *Domino 6 for Linux*, SG24-6835
- *Lotus Domino 6 Spam Survival Guide,* SG24-6930
- *Upgrading to Lotus Notes and Domino 6*, SG24-6889
- *Domino Designer 6: A Developer’s Handbook*, SG24-6854 (for information about DCRs)
- *Lotus Domino for S/390 Release 5: Enterprise Integration Using Domino Connectors,* SG24-5682 (for information about DECS and zSeries)
- *Domino and WebSphere Together Second Edition,* SG24-5955
- *Lotus Domino for S/390: Running a Large Domino System,* SG24-5984
- *Performance Considerations for Domino Applications*, SG24-5602
- *IBM Lotus Domino 6.5 for Linux on System z9 Implementation*, SG24-7021

Tivoli

- *IBM Tivoli Storage Management Concepts,* SG24-4877
- *IBM Tivoli Storage Manager Version 5.3 Technical Guide*, SG24-6638

Other publications

These publications are also relevant as further information sources:

- *CP Planning and Administration*, SC24-6043
- *CP Command and Utility Reference*, SC24-6008
- *TCP/IP Planning and Customization*, SC24-6019
- *IBM Tivoli Storage Manager for Mail: Data Protection for Lotus Domino for UNIX and OS/400 Installation and User's Guide*, SC32-9056
- *z/VM Guide for Automated Installation and Service, Version 4, Release 4.0*, GC24-6064
- *IBM Tivoli Storage Manager Performance Tuning Guide*, SC32-9101

Online resources

These Web sites and URLs are also relevant as further information sources:

- Linuxvm.org - the Linux on zSeries® portal

  `http://linuxvm.org`
- DeveloperWorks - IBM Boeblingen

  `http://www10.software.ibm.com/developerworks/opensource/linux390/index.shtml`
- ISV applications for Linux on zSeries

  `http://www.ibm.com/servers/eserver/zseries/solutions/s390da/linuxproduct.html`
- The IBM site for z/VM and Linux

  `http://www.vm.ibm.com/linux`
- z/VM publications

  `http://www.vm.ibm.com/pubs/`
- Lotus developers domain, including product documentation

  `http://www.lotus.com/ldd`
- IBM Lotus home page

  `http://www.lotus.com`
- SuSE, a developer of Linux

  `http://www.suse.de`
- UnitedLinux, a private company equally owned by four Linux development companies

  `http://www.unitedlinux.com`

Newsgroups

The linux-390 list server is where the community meets. To subscribe to it, send an e-mail to listserv@vm.marist.edu with this line in the body:

```
subscribe linux-390
```

You should receive a return e-mail in 5 - 15 minutes, asking you to click a URL. When you click that URL, you're in. You'll get another, confirmation e-mail. Save this confirmation e-mail, as it contains many instructions. Then you can send the same address an e-mail to tailor your subscription.

Here are the most useful commands:

```
set linux-390 nomail    // stay subscribed get no e-mail (good for holidays)
set linux-390 mail      // turn e-mail back on
signoff linux-390       // when you've had enough
set linux-390 digest    // get one big e-mail a day
set linux-390 repro     // get a copy of your own appends
get linux-390 log0204   // get a month worth of appends (e.g. April 2002)
```

To append to the list, simply send an e-mail to linux-390@vm.marist.edu. The linux-390 archives are on the Web at:

http://www.marist.edu/htbin/wlvindex?linux-390

Also refer to the SuSE Linux mailing list archive regarding Domino:

http://lists.suse.com/archive/suse-domino/

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

IBM Redbooks

Best Practices for Lotus Domino on System z

(0.2”spine)
0.17”<->0.473”
90<->249 pages

IBM®

Best Practices for Lotus Domino on System z:

z9 and zSeries

Redbooks

Best practices for implementing Lotus Domino on an IBM mainframe

Performance, tuning, and capacity planning

Technical support overview and procedures

Lotus Domino provides much more than just e-mail. IBM Lotus Notes and Domino are designed to help organizations increase productivity and responsiveness through business-critical collaboration capabilities and a rapid application development and deployment environment. To most effectively use your system capacity and get the best performance from Domino, we recommend the best practices described in this IBM Redbook.

Domino on System z9 has matured into a powerful server platform that meets the diverse needs of many organizations. IBM System z9 offers key functions such as clustering for scalability and availability and systems management tools. These tools, which are important to organizations, are mature and have been available on System z9 for many years. System z9 is able to support large workloads because of the unique capabilities of the hardware and software.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-7209-00 ISBN 073849643X