IBM

Lotus Domino 7 on Linux for IBM System z

Capacity Planning and Performance Updates

Scalability under Linux

Performance under z/VM

Multiple Domino servers

Don Corbett
Juergen Doelle
Barbara Filippi
Wu Huang
Mike Wojton

Redpaper

ibm.com/redbooks

International Technical Support Organization

Domino 7 for IBM System z: Capacity Planning and Performance Updates

December 2006

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

First Edition (December 2006)

This edition applies to Release 7 of IBM Lotus Domino for System z.

Contents

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 illustrate 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.

Trademarks

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

Domino®
Informix®
IBM®
Lotus®
Lotusphere®
Redbooks™
Redbooks (logo) ™
System z™
System z9™
z/OS®
z/VM®
zSeries®
z9™

The following terms are trademarks of other companies:

SAP R/3, SAP, and SAP logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries.

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

Intel, Xeon, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States, other countries, or both.

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

IBM® Lotus® Domino® 7, which first became available in the Fall of 2005, includes many enhancements in the areas of capacity and performance. Foremost among them is Domino 7 support of a 64-bit Linux® kernel on IBM for System z™, which improves Domino's throughput and offers more vertical scaling than previous releases. Additionally, some I/O performance enhancements became available with z/VM® 5.2, which also benefit customers who wish to implement Domino to run in z/VM Linux guests. Testing has shown that Domino 7 and z/VM 5.2 together have some clear advantages over their predecessor releases, and greatly improve the total cost of ownership for running Domino.

This IBM Redpaper discusses the results of those tests and provides some recommendations.

The team that wrote this Redpaper

This IBM Redpaper was produced by a team of specialists from around the world.

Don Corbett is a Senior Software Engineer in the System z9™ Development Software Performance Department in Poughkeepsie, New York. Don has more than 40 years of experience in IT working at IBM. His areas of expertise include operating system performance analysis and design for Linux on System z9 and z/OS®. He has led multiple performance benchmarking and capacity planning projects, including Domino for Linux on System z9 and other middleware products.

Dr. Juergen Doelle is the project leader of the Linux end-to-end performance project, which examines the performance of various middleware products and applications on Linux on System z. Since 1994, he has worked as a developer for products on z/OS and USS, and since 2001 as performance analyst for Linux on System z. His areas of performance experience are SAP® R/3, Informix®, and Oracle databases, in addition to Linux kernel performance with the focus on disk I/O, storage servers, and the FCP attachment for SCSI.

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 27 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 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.

Mike Wojton is a Certified Senior IT Specialist. He has more than 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 on and written about Domino installation, benchmarking, performance, capacity planning, problem determination and problem source identification. He has also participated in the Performance Zone at Lotusphere® since 2000.

Thanks to the following people for their contributions to this project:

Mike Ebbers	International Technical Support Organization, Poughkeepsie Center, IBM

The team that did the system setup and performed the benchmark measurements:

Stephen McGarril	IBM System z Benchmark Center, Poughkeepsie
Eugene Ong	IBM System z Benchmark Center, Poughkeepsie
Judy Viccica	IBM System z Benchmark Center, Poughkeepsie
Chris Williams	IBM Project Manager for zSeries® Benchmark Center, Poughkeepsie

Other people who supported our efforts:

Cherie Barnes	IBM Endicott z/VM development performance team - z/VM sizing tool
Evanne Bernardo	IBM Size390 team Gaithersburg- sizing tools
Bill Bitner	IBM Endicott z/VM development performance team - z/VM sizing tool
John Campbell	IBM Size390 team Gaithersburg - sizing tools
Jon Entwistle	IBM System z Performance, Poughkeepsie
Clark Goodrich	Former Domino for IBM System z Development team leader, Poughkeepsie

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 will team with IBM technical professionals, Business Partners, and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you will 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 papers to be as helpful as possible. Send us your comments about this Redpaper 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 e-mail to:

  redbook@us.ibm.com

- Mail your comments to:

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

1

Introduction

This chapter describes the background behind this testing study, the objectives, and how to put the results to use in your installation.

1.1 Background for Domino 7.0 capacity and performance study

About 18 months ago, the Scalability Center for Linux on zSeries (SCL) in Poughkeepsie performed a study on Domino 6.5 for Linux on IBM System z. During that study, capacity planning and performance data was collected using an industry-standard benchmarking tool to drive both NRPC (Notes) and Domino Web Access (HTTP) e-mail users. The results of this benchmark showed two bottlenecks that prevented linear scalability of throughput for Domino 6.5 for Linux on System z.

The first bottleneck was due to the fact that Domino 6.5 does not support the 64-bit Linux kernel on either zSeries or Intel®. To scale Domino within an OS image, multiple Domino servers or partitions (also called DPARs) are typically deployed to extend Domino scalability. Each DPAR has its own set of virtual address spaces to support its users. The SCL's testing showed that a Linux instance with multiple DPARs and many users per DPAR caused the Linux kernel to swap heavily as the number of users was increased past a certain threshold because of the 2 GB real memory limitation of the 31-bit kernel. To prevent any one kernel from swapping heavily, multiple Linux images, each with only a single production Domino 6.5 server, must be deployed for production environments.

The second bottleneck was tied to z/VM4.4, the release that was used to test Domino 6.5 running in Linux guests. The I/O support modules for z/VM 5.1 and earlier releases still use 31-bit addressing, which implies that any I/O that is done by a guest running above the 2 GB line must be moved below the 2 GB line to be completed. The fact that the available storage for I/O below the 2 GB line is limited in size is a major issue. If there is enough I/O being driven by Linux guests, it causes z/VM to page this area, which can cause z/VM to page heavily under high disk I/O demands. During the SCL's testing, they were not able to scale Domino 6.5 beyond 13,000 active NRPC users (spread across multiple z/VM guests) running under a single z/VM LPAR. For larger Domino 6.5 production deployments with robust I/O requirements, this implies that multiple z/VM LPARs might have to be deployed to support Domino in order to avoid z/VM paging issues.

Domino 7.0 and z/VM 5.2, which both became generally available in the latter half of 2005, offer relief for the two bottlenecks that are associated with their predecessor products. Domino 7 supports a 64-bit kernel which allows a Linux instance to support multiple Domino DPARs as long as enough memory is available to Linux to minimize swapping. With z/VM 5.2, I/O processing has been changed to not move the target page for I/O below the 2 GB line. With this release of z/VM, disk I/O no longer contributes to z/VM paging. For more information about z/VM 5.2 changes that improve performance, see:

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

A few months after Domino 7.0 became available, a study was organized by the Linux End-to-End Performance Group in Germany to verify the extent to which the bottlenecks had been removed and to provide updated capacity and performance information for Domino 7 for IBM System z running on Linux.

1.2 Study objectives

The primary focus of the Linux End-to-End Performance Group's study was to show improvements and differences between running Domino 7.0 and Domino 6.5 in a Linux for System z environment. Although all testing was done at the Domino 7.0 level, the results that were reported for this study would be applicable to follow-on maintenance releases for Domino 7. The high-level objectives of the study were as follows:

- Show that prior Linux memory and z/VM I/O bottlenecks have been removed.
- Determine how linearly Domino scales as the number of CPs and users are increased.
- Measure the overhead of running multiple DPARs versus a single DPAR in a Linux LPAR.
- Measure the overhead of running Domino within Linux z/VM guests versus Linux LPARs.
- Measure the impact of virtual-to-real CP ratios for Domino in a z/VM Linux guest environment.
- Define new guidelines for running Domino 7 in Linux images:
  - Number of active 15-minute users per DPAR

    (This means users who have been active in a 15-minute period, which should not be confused with connected users who are not necessarily active during each 15-minute period.)
  - Number of DPARs per Linux image
  - Memory usage per DPAR and Linux image
  - Any other performance and tuning recommendations for Linux, z/VM, and Domino

1.3 How to use this paper

This Redpaper presents the latest capacity and performance guidelines for Domino 7 and is intended to be used in conjunction with the IBM Redbook *IBM Lotus Domino 6.5 for Linux on zSeries Implementation*, SG24-7021. This book has valuable implementation information applicable to both Domino 6.5 and 7 on Linux. It covers a broad range of topics for deploying a Domino for IBM System z server on Linux, including:

- Planning the operating system, disk, and network environment
- Installing and administering Linux and Domino
- Capacity planning and performance

The capacity and performance discussion in the Domino 6.5 implementation book has two components: how to monitor capacity and performance, and capacity and performance configuration guidelines for running Domino 6.5 on Linux. The discussion about monitoring capacity and performance is valid for both Domino 6.5 and 7 releases. In this paper, we cover the new capacity and performance guidelines specific to Domino 7.

Another IBM Redbook that has valuable information about tuning Domino is *Best Practices for Lotus Domino on System z: z9 and zSeries*, SG24-7209. Again, the recommendations documented in this book are applicable to both Domino 6.5 and 7, unless otherwise stated.

When installing a new release of Domino, always review the latest installation documentation and release notes for the most up-to-date procedures and recommended configuration settings.

2

Domino and the benchmarking tool used

This chapter discusses the benchmarking tool and specific workloads that are used to test Domino 7.0. It also describes the differences between production and benchmark workloads, and offers some guidelines for interpreting data derived from benchmark testing.

2.1 Benchmark driver and workloads

The same industry-standard benchmarking tool and specific workloads were used for testing Domino 7.0 as were used for Domino 6.5 to guarantee valid comparisons between the two releases. For each run, this tool generated a transactions-per-minute (TPM) metric, not to be confused with the Domino transaction statistic, along with a value for the maximum capacity (number of active users) supported and their average response time.

From a tool perspective, a workload is a defined script that is used to simulate user activity through specific applications. Workloads typically cover a variety of protocols such as IMAP, NRPC, and HTTP. Workload scripts provide a common method to apply a consistent, repeatable load against the Domino server in order to assess the effects of various operating systems, hardware, and configuration changes.

For this study, only NRPC and HTTP workloads were considered. For the NRPC client testing, a workload script called R6Mail was used, which emulates Notes users with their own mail file systems. It repeats a set of user actions every 90 minutes. For the DWA testing, a script called R6iNotes was used to emulate HTTP users with their own mail file systems. Again, this script repeats a series of actions every 90 minutes. Table 2-1 lists the user actions that were invoked by both workloads. The R6Mail and R6iNotes scripts were also used in previous testing for Domino 6.5.

Table 2-1 User actions invoked by workloads

Actions every 90 minutes	R6Mail NRPC workload	R6iNotes DWA workload
Open Inbox	6	6
Read Message	30	30
Delete Message	12	12
Add Message to Inbox	2 (50 KB)	2 (100 KB average)
Send Message to 3 Recipients	1 (100 KB average)	1 (100 KB average)
Send Calendar Invitation to 3 Recipients	1	1
Send RSVP	1	1
Close Inbox	6	6

For these scripted workloads, the size of each inbox is around 20 MB. Keep in mind that the scripted actions and inbox and message sizes reflect benchmark workloads. Production workloads would be quite different with much larger inboxes and message sizes, movement of documents from the inboxes to folders, and an increased frequency of creating memos and replying to received mail.

Before each run for this study, all new copies of the mail files were allocated and initialized to minimize variability. Then users were ramped up (logged in) at 1000 to 1500 at a time until the desired number of users was attained for a particular test case. After steady-state had been reached (all users actively executing the test scripts/workloads and no more users logging in), the test runs were allowed to execute for a period of time during which performance and capacity data were collected. Runs with average response times of greater than one second were discarded as unacceptable and not included as input into the results of the study.

2.2 Benchmark versus production workloads

The user workloads that were simulated through the tool showed that Domino servers running in Linux LPARs and z/VM guests can support several thousand users if properly configured. However, as discussed earlier, simulated users are not equivalent to production users. The processing scenarios for the simulated clients did not include all the functions, features, and third-party software that production users might access, such as Domino administration tasks, view/full-text indexing, anti-virus, and others. Also, simulated users are constantly accessing the Domino server with a steady amount of work, which is not comparable to the ebb and flow of a production workload.

Domino servers that are used for benchmark testing are typically brought up with a minimal configuration of two to four Domino tasks (Server, Router and HTTP tasks, and in the tests under discussion Logasio for Domino transaction logging). Most Domino production servers will be running at least 10 to 20 tasks or more. Besides the tasks brought up on benchmark servers, production environments might include Update to index views and databases, AMgr to invoke event-driven or scheduled processing, AdminP to perform administrative tasks on the Domino server, Collect to collect Domino statistics, and many others.

Production users have different CPU needs, ranging from very light users in, for example, a manufacturing facility, to very heavy users at a corporate headquarters. Although benchmark users frequently have lighter CPU requirements than production users, this does not negate the value of benchmark testing. Because benchmark testing guarantees a standard workload, it is a valid way of driving CPU utilization for measuring things such as workload scalability, differences in CPU costs when configuration changes are made, and some CPU improvements between Domino releases.

3

Test environments and scenarios

This chapter describes the hardware, software, and network configuration (LPAR and z/VM guests) used to test Domino 7.0. Additionally, it shows how we were able to configure the Linux and network environments to run with multiple Domino partitions or servers, a benefit of the new support for a 64-bit Linux kernel in Domino 7.

3.1 Hardware used to test

The following hardware was used for this study:

- z9 system (2094-S18) with up to eight CPs for most of the measurements
  - Four additional CPs as needed.
  - The number of available CPs varied according to workload and test scenario.
- 42 GB memory

  The amount of memory used varied according to workload and test scenario.
- 1 DS 8000 DASD unit with 6 TB of space
- Workload driver workstations (2-way Intel Xeon® boxes at 3.06 GHz with 4 GB RAM)
  - Nine x-345 Linux workstations, including one spare, to drive benchmark clients
  - One x-345 Windows® workstation to act as the master controller
- Cisco 6500 series switch between workstations and z9 server
- One OSA-Express2 1000Base-T Ethernet card for the z9 server

Figure 3-1 depicts the hardware configuration and client network connectivity to the z9.

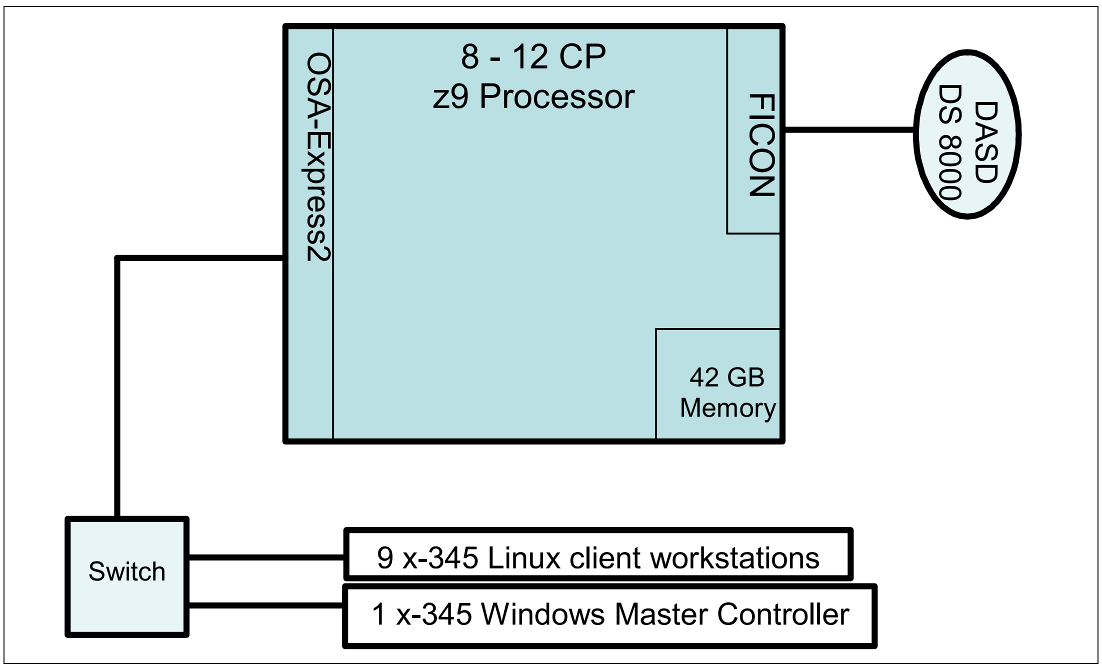

Figure 3-1 Hardware configuration and connectivity

3.2 Software used to test

The following products and software levels were installed:

- Domino 7.0
- Linux SUSE SLES 9 SP 2 Kernel without Fixed I/O Buffers patch on the Linux server
- SUSE SLES 9 SP 2 on driver workstations
- EXT3 file systems
- z/VM 5.2

3.3 Domino network configuration

A host system in the System z environment has the ability to run multiple copies of a Domino server. Domino partitions (DPARs) can run in a native logical partition (LPAR) or in a guest machine inside a z/VM LPAR. This section gives an example of each.

3.3.1 DPARs within a single Linux LPAR or guest

Figure 3-2 shows the network setup for the Domino Linux LPAR tests. Using this configuration, multiple Domino partitions were created to share the same Domino set of executables. Each DPAR had its own set of processes and each process had a 2 GB virtual address space, which were accessed by a predefined number of users.

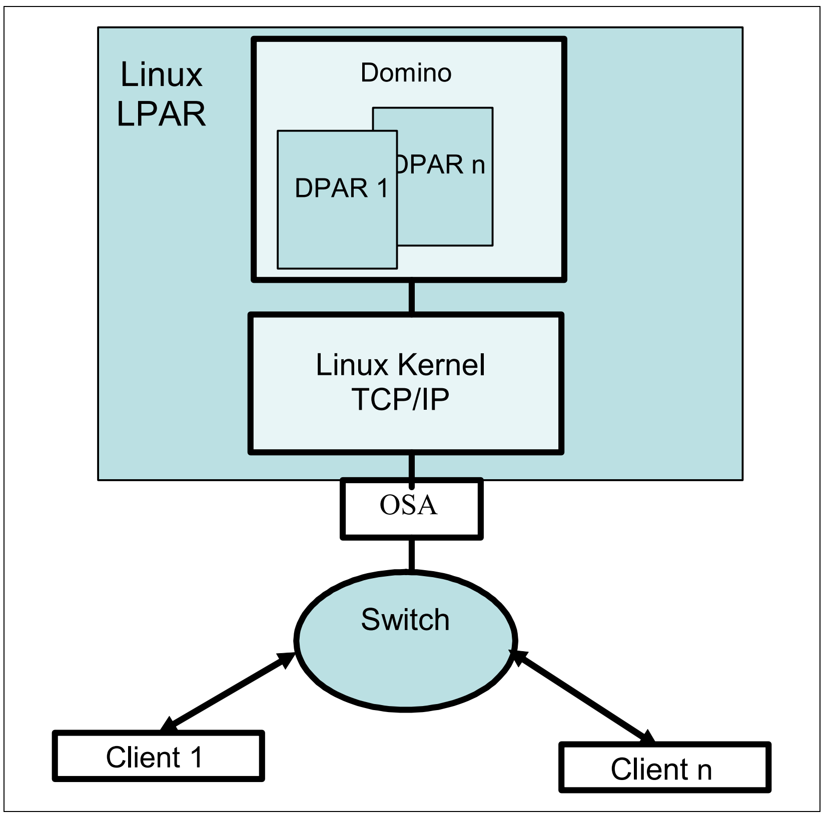

Figure 3-2 Network setup for Linux LPARs with multiple DPARs

Figure 3-3 shows the configuration for multiple DPARs in a single z/VM guest. Again, the guest had a single set of Domino executables shared by multiple Domino partitions.

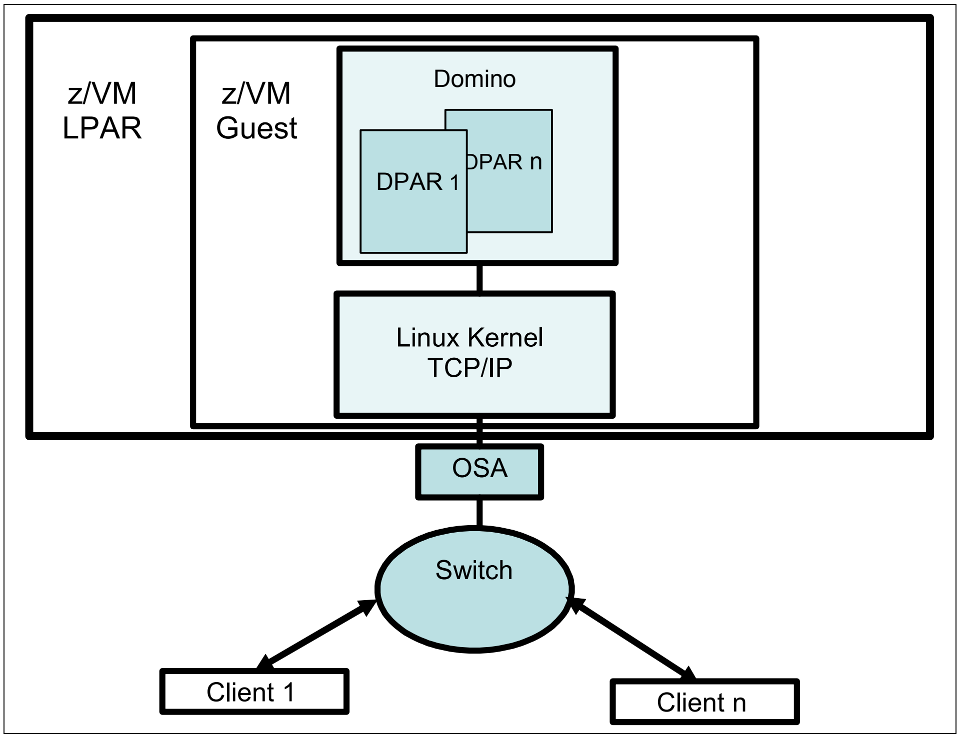

Figure 3-3 Network setup for z/VM guests with multiple DPARs

3.3.2 DPARs within multiple z/VM Linux guests

For the z/VM multi-guest runs, OSA routing by IP address was implemented. Figure 3-4 shows the configuration that was used with multiple z/VM guests. Each z/VM guest had a dedicated connection to the OSA card. Only one OSA card was required to handle all of the network traffic that was generated during the benchmarks. In a production environment, you might want to use multiple OSA cards if the throughput is more than what the OSA card can support. If multiple OSA cards are deployed, consider using VIPAs (virtual IP addresses) to allow load balancing of the network traffic. With the availability of multiple cards, VIPA also provides the added benefit of failover. Even if one of the cards should fail, traffic can be rerouted automatically through the other card.

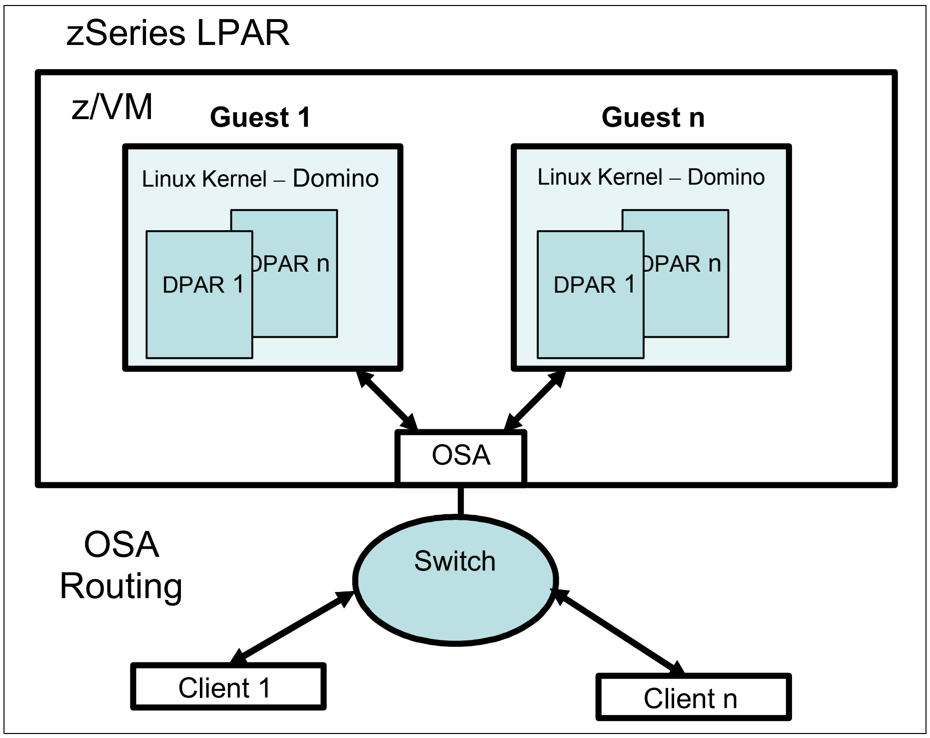

Figure 3-4 OSA routing for a multi-guest environment

4

Background information and terminology for interpreting data runs

This chapter provides definitions and background discussions about External Transaction Rates (ETRs), CPU data, and Internal Transaction Rates (ITRs), as preparation for understanding the test run data in Chapter 5. It is important to understand what these values represent and how to apply them. A sample scenario, outside the scope of the study under discussion, is used to illustrate what these measurements mean and how to interpret them. This will lay a foundation for the discussion of actual results in the following chapter.

4.1 ETR data

Note: The discussion in this chapter is for background information. See Chapter 5, “Study results” on page 19 for actual data runs.

ETR is the measurement of some external workload that is being performed during a benchmark. In Figure 4-1, the amount of workload (ETR) that was performed during a benchmark is plotted at five different sample times.

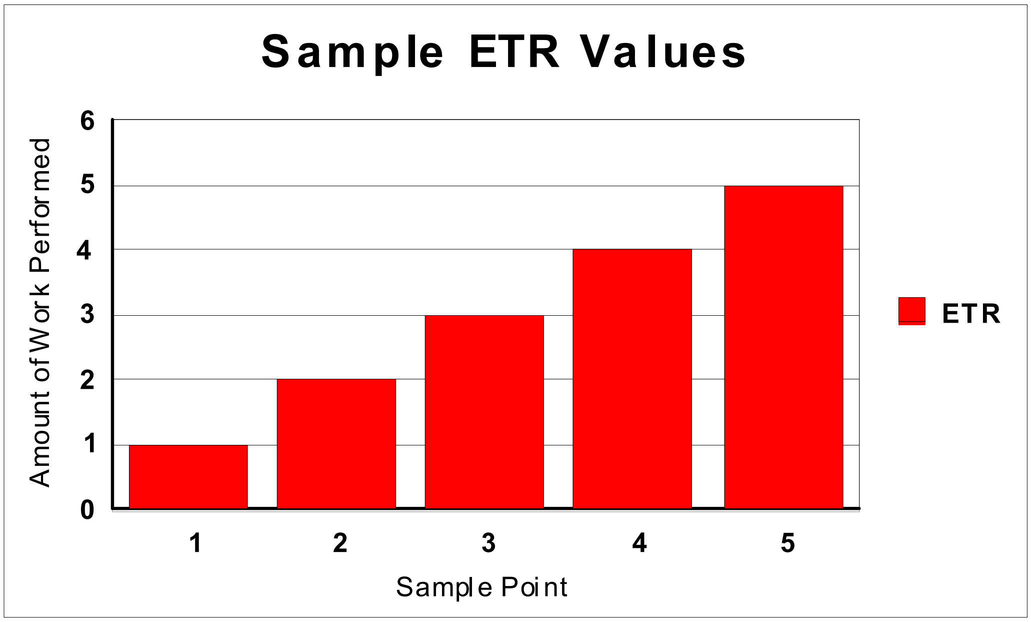

Figure 4-1 ETR values

For each successive sample point (each point could be from single or multiple benchmark runs), the amount of work increased by 100% when compared to the initial point. In the case of a Domino benchmark, this might indicate that there was twice as much mail being sent between points one and two, and five times as much mail between points one and five. Figure 4-1 indicates that the system was able to handle the increase in workload over these samples, which is a good trend.

4.2 CPU data

However, ETR growth is only part of the picture. Another factor to consider is CPU. How much CPU does the workload cost? Figure 4-2 shows the amount of CPU that was consumed for the same sample points plotted in Figure 4-1 on page 16. The ETR measurements are also included in this chart to facilitate the comparison of trends between ETR and CPU.

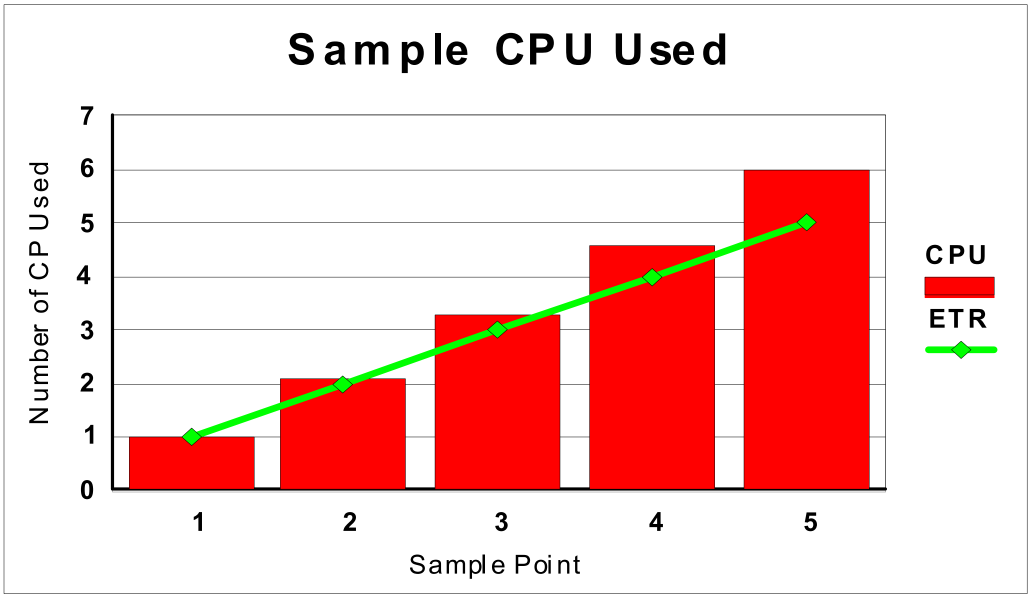

Figure 4-2 CPU values

Figure 4-2 clearly shows that the amount of CPU used for the workload did *not* grow in the same linear fashion as the workload driving the system. The workload on the right side of the chart is taking proportionally more cycles than the workload on the left side of the chart.

4.3 ITR data

ITR is the relationship of how much CPU is required to run specific benchmark workloads. It is calculated by taking the workload (ETR) and dividing it by the CPU percentage times 100. Figure 4-3 shows the ITR values for the same sample points plotted in the charts shown in Figure 4-1 on page 16 and Figure 4-2 on page 17.

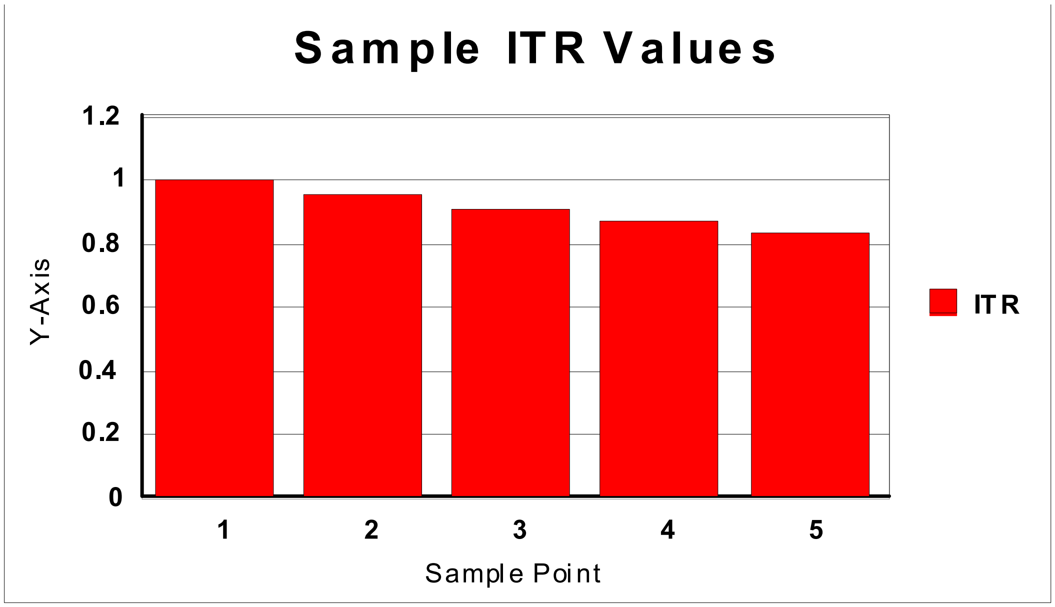

Figure 4-3 ITR values

The downward ITR trend shows is that as the workload was growing, it was getting more expensive to run. An upward trend would mean that as the workload was growing, processing it was becoming more efficient: less cost for the same types of transactions.

All of the Domino 7.0 benchmark results were analyzed in terms of ETR and ITR. The ETR was calculated from the benchmark TPM (transactions per minute) divided by 60 to get transactions per second. (TPM is described in 2.1, “Benchmark driver and workloads” on page 6.) We used the ETR by itself to compare workload consistency between test runs where factors such as number of DPARs, CPs, memory, and other configuration options might be varied. For example, if the ETRs are very close in value between runs or they scale almost linearly when capacity is increased between runs, then the workload is very consistent. This means that users accessed the same functions at the same rate with very comparable response times. We calculated ITR from the ETR divided by the CPU percent utilization times 100. ITRs were used to do all of the analysis for the Domino 7.0 measurements, which included things like determining the scalability factor of workloads as CPs were added, the cost of multiple DPARs compared to a single DPAR, the impact of virtual-to-real CP ratios when running with z/VM, and so far.

In summary, although ETR gives you the overall view of how much workload is being performed in benchmarks, ITR lets you know how efficiently that workload is being processed.

5

Study results

This chapter describes the various test scenarios that were run with Domino 7.0 in the general areas of:

- Domino workload processor scalability
- Capacity requirements of single versus multiple Domino partitions
- Capacity requirements of running Domino in z/VM Linux guests versus Linux LPARs
- Impact of virtual-to-real CP ratios for Domino in a z/VM environment
- Maximum number of Domino NRPC users supported by single Linux LPAR

Each section contains the details and results of running a specific test case, and a comparison of Domino 7.0 results to results from a comparable Domino 6.5 test run, if such a run was executed in the past.

Compared to Domino 6.5, Domino 7.0 showed significant improvements with its ability to support multiple Domino partitions and up to 50,000 NRPC users within a single Linux image.

5.1 Domino workload processor scalability

IBM publishes the Large Systems Performance Reference (LSPR) to compare capacity on the various zSeries and z9 processor models. A variety of standard workload mixes running on z/OS and Linux are defined to describe application scalability. A series of tests was run with Domino 7.0 to see how it compared to some standard LSPR workloads.

Because the behavior of NRPC and DWA workloads are quite different, tests were run for each type of client to determine Domino's workload behavior. DWA is a much more CPU-intensive workload than NRPC because all of the processing is placed on the Domino server, causing it to consume approximately four times or more as many cycles as NRPC. Consequently, DWA supports significantly fewer users than NRPC for a given number of CPs. DWA clients in general are more memory intensive, which also limits the number of supported users on a Domino server.

Within a single LPAR, several scenarios were run with NRPC and DWA clients to determine how linearly the Domino Notes and browser workloads scale as the number of users and CPs were increased.

All tests were executed in a single LPAR running four Domino DPARs. A single 3390 mod 9 volume was defined for Linux swap space for all runs that were executed for this study (Linux LPARs and z/VM guests). However, little or no swapping was seen on any of the runs because enough main memory was allocated to the Linux images.

5.1.1 NRPC

On zSeries, scalability is stated in terms of the transaction rate (not equivalent to Domino transaction rate!) that can be obtained with a given workload and how that transaction rate behaves when the number of processors is increased. Perfect scalability would have the transaction rate double when the number of CPs is doubled; however, due to a number of factors (among them the overhead associated with managing more CPs), this does not happen. There are two LSPR curves (among others) that characterize workload with very good and efficient scalability, CB-L (Commercial Batch Long Job Steps), and not so efficient scalability, OLTP-T (Traditional On-Line Workload). For this study, these workloads were used as the baseline to show how well Domino scales.

Measurements

To determine how Domino 7 compares to the LSPR workloads, tests were run on a two-way and four-way processor. These scenarios were specifically designed to meet the following requirements for workload scalability testing:

- The number of users per CP must be the same across all runs.

  In other words, the number of NRPC users was doubled between the two-way and four-way runs.

- The overall processor CPU must be driven to somewhere between 85% and 95% for all runs.

There was also an attempt to triple the workload of the two-CP scenario on a six-CP processor. However, this run was unsuccessful because of a dramatic increase in CPU utilization, and consequently was not included in the NRPC scalability analysis. For most applications, there normally would also be a one-CP run. This scenario was not run because of Domino's architecture, which requires access to at least two CPs for optimum performance, which is a cross-platform recommendation not specific to zSeries and System z.

Figure 5-1 shows the scalability behavior for NRPC clients distributed across four DPARs in a single LPAR with two and four CPs. When increasing from two to four CPs, the ITRs varied by a factor of 1.7, indicating that throughput decreased when the workload increased by a factor of two. (See Chapter 4, “Background information and terminology for interpreting data runs” on page 15, for a description of ITR or Internal Transaction Rate.) From an LSPR point of view, this is not a good scaling factor. Ideally, workloads scaled between a two-way and four-way processor should fall somewhere between 1.89 (OLTP-T) for those workloads with less scalability and 1.95 (CB-L) for those that scale very well. Perfect scalability would be a factor of 2.00, but that is not possible because there is associated overhead in managing multiple CPs.

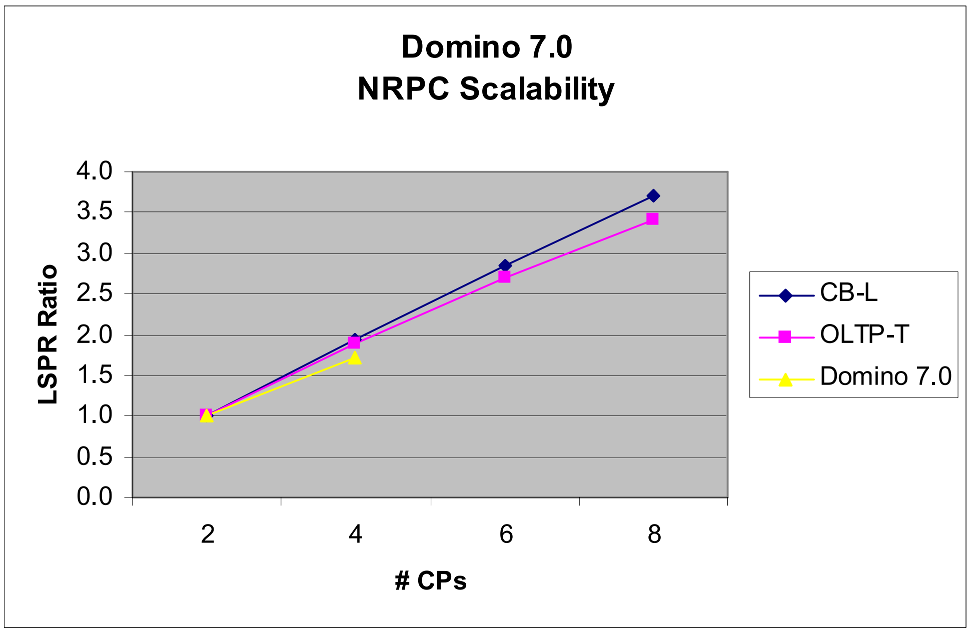

Figure 5-1 NRPC scalability for Domino 7

Given these results, the conclusion is that this type of NRPC workload, a very large number of light users (low CPU usage per client), does not scale very well at four CPs. However, we do not believe the NRPC scalability results to be a problem in production environments because production NRPC users, even light ones, will have heavier CPU requirements. In the Domino 7.0 testing, more CPU-intensive workloads, such as DWA, showed a scaling factor very close to 1.95 on a four-way. See 5.1.2, “DWA” on page 22, in which the results of the DWA testing are described. So, we believe that a more CPU-intensive NRPC workload will scale very well up to at least four CPs, and possibly more, depending on the CPU intensity of the users.

Similar tests, documented in Figure 5-2 on page 22, were run for Domino 6.5. At that time, Domino was compared to the LSPR workloads CB-L and CB-S (Commercial Batch Short Job Steps, by definition a workload with less scalability), and had a scaling factor of just over 1.5. The results for Domino 7.0 were somewhat better at 1.7. Also, keep in mind that the Domino 6.5 runs were done with a single DPAR in an LPAR, which meant fewer users than the Domino 7.0 runs. The lower number of users made a third test possible on six CPs.

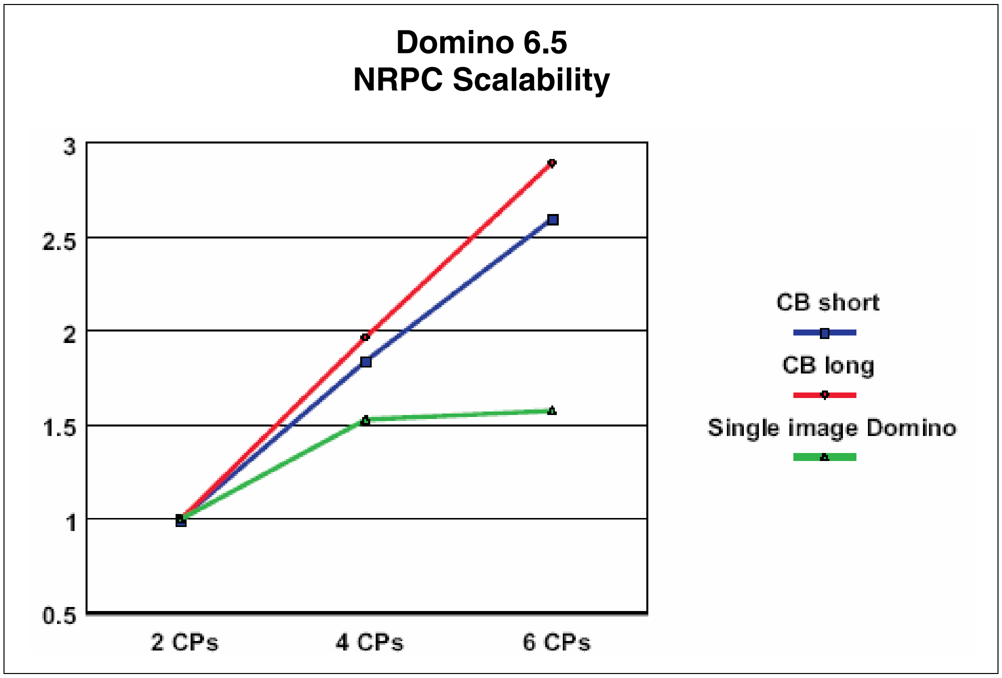

Figure 5-2 NRPC scalability for Domino 6.5

Summary

The NRPC benchmark workload is characterized by low CPU utilization per user, but high numbers of users, and high I/O rates. Based on LSPR workload ratios, this workload does not scale well on four CPs. Compared to production environments, however, the NRPC benchmark workload is very light. Consequently, we see no issues with recommending up to four CPs (possibly more for more CPU-intensive workloads) for NRPC users in a single LPAR. Although the six-CP tests showed that scalability dropped off significantly after four CPs on a single LPAR, good scalability can be achieved with multiple four-CP LPARs for larger user deployments. Furthermore, Lotus recognizes that the current benchmark workload is very light and is developing new NRPC benchmark workloads more inline with production workloads for the next release of Domino.

5.1.2 DWA

To measure the DWA workload scalability, similar runs to those for NRPC, but with fewer users to accommodate heavier CPU demands, were executed on two-way, four-way, and eight-way processors.

Measurements

Using the two-way run as a base, the number of DWA clients was doubled on the four-way run and quadrupled on the eight-way run. From an LSPR point-view, the DWA workload scaled very well: nearly the same as CB-L on the four-way and slightly better than CB-L on the eight-way as shown in Figure 5-3 on page 23.

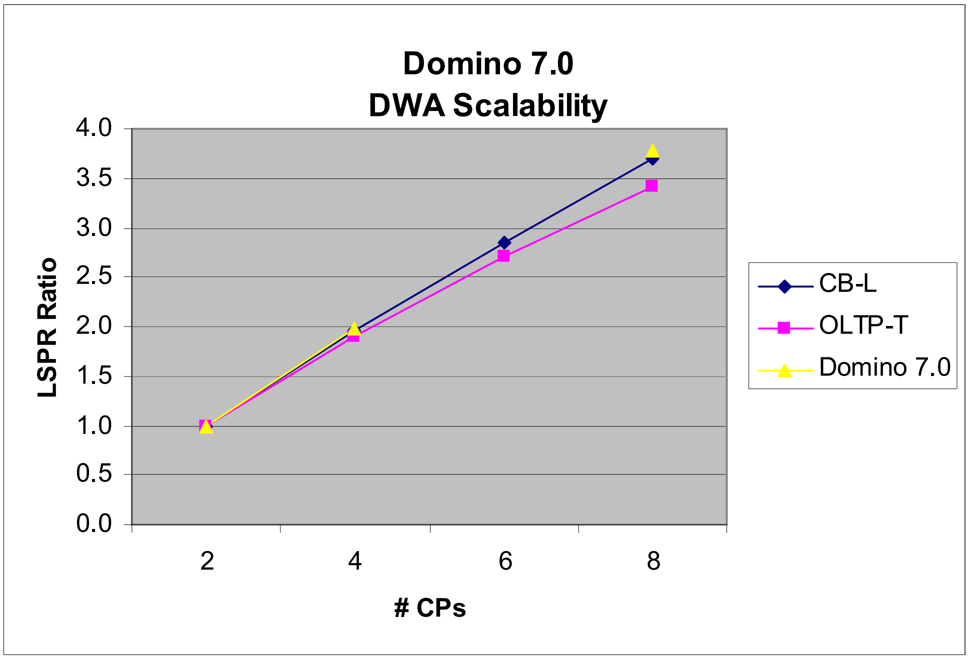

Figure 5-3 DWA scalability for Domino 7

Domino 6.5 also showed good scalability for the DWA workload as shown in Figure 5-4. It closely followed the scalability curve for CB-L, even slightly exceeding it at the upper end.

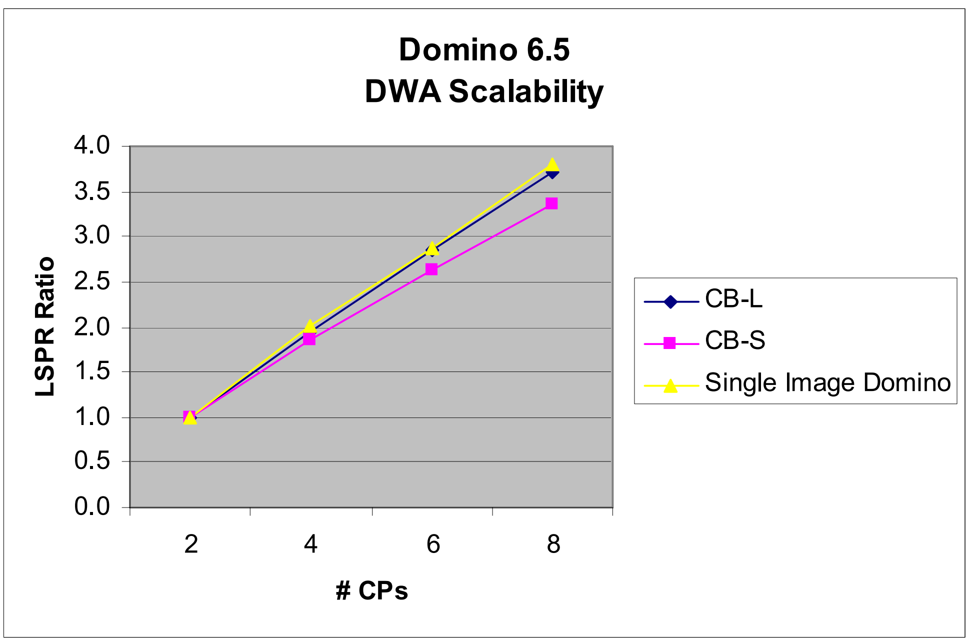

Figure 5-4 DWA scalability for Domino 6.5

Summary

The DWA workload is characterized by a high CPU utilization per user and fewer active users when compared to NRPC. Based on Domino 7.0 testing, it scales very well up to eight CPs, and we expect would scale well beyond eight CPs if more hardware had been available for additional testing.

5.2 CPU cost of single versus multiple DPARs

Domino as an application is architected to run with multiple tasks: some in the foreground to satisfy end-user requests (such as Server, HTTP, and POP3), others in the background to perform tasks such view indexing, event-driven processing, mail routing, anti-virus, and so on. Unlike some applications, Domino continuously polls for work; even an "idle" Domino server uses some number of cycles. This set of runs was designed to measure the CPU overhead of running a given NRPC and DWA workload in one large versus several smaller DPARs executing in a single LPAR, and to determine the most CPU-efficient number of DPARs for Linux images. Although all runs were executed in Linux native mode, these results would also be applicable to Domino running in z/VM guests.

5.2.1 NRPC

For this set of tests, 12,000 NRPC users were used to drive one, two, three, and four DPARs. For the multi-DPAR runs, the 12,000 users were evenly distributed across the DPARs—in other words, for the two-DPAR run, half of the users were assigned to each DPAR; for the three-DPAR run, one-third of the users were assigned to each DPAR; and so on. Transaction logging and mail routing were enabled on all of the servers.

Measurements

Figure 5-5 and Figure 5-6 on page 25 show the results of these tests.

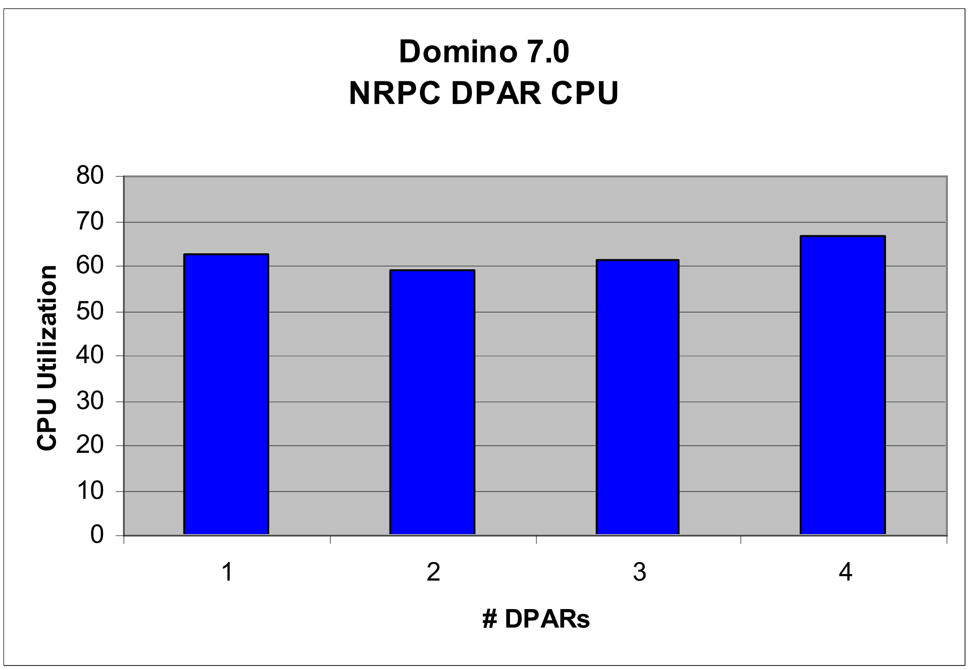

Figure 5-5   NRPC DPAR CPU utilization for Domino 7

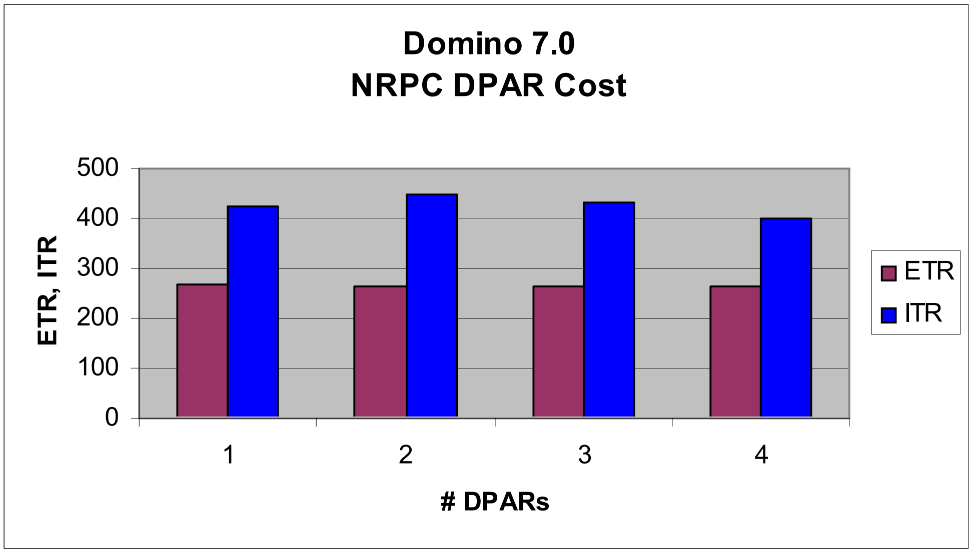

Figure 5-6 DWA DPAR CPU utilization for Domino 7

The CPU utilization for the single-DPAR run was higher than initially expected when compared to the two-DPAR, three-DPAR, and four-DPAR runs. But given that previous Development testing has shown that 12,000 active NRPC users is the scalability limit for a single Domino server running on zSeries Linux, this anomaly was not surprising. As Domino approaches this upper limit, it incurs more and more overhead to manage the large number of NRPC clients.

The CPU utilization in Figure 5-5 on page 24 shows that when going from two to three DPARs, CPU load increased 3.7%, and when going from two to four DPARs, 12.9%. It should also be noted that the ETR (transactions per second) values were almost the same regardless of the number of DPARs, which proved that the workload was consistent across all four runs. Again and as expected, the ITR (throughput by cost) value was lower for the one-DPAR run, which was constrained by the large number of active users. Compared to the two-DPAR run, the ITR decreased 3.6% with the third DPAR and 11.3% with the fourth DPAR.

In the past, a similar set of tests was run with Domino 6.5. As shown in Figure 5-7 on page 26, the Domino 6.5 test results were significantly worse than those obtained with 7.0. Domino 6.5 was tested with 4,200 NRPC users on one, two, and three DPARs. Again, the DPAR costs are expressed in terms of changes to ITR values. When comparing the first with the second DPAR, the ITR degraded by 7.6%; when comparing the first with the third DPAR, the ITR degraded by 25.8%. The large cost of the third DPAR was due to the fact that Linux was experiencing very heavy swapping, which also prevented additional testing with more users and a fourth DPAR. Compared to the one-DPAR scenario, CPU utilization increased 8.3% with the second DPAR and 35.2% with the third. The large discrepancy between change in CPU and ITR on the third DPAR shows the value of going to a cost-based metric such as ITR.

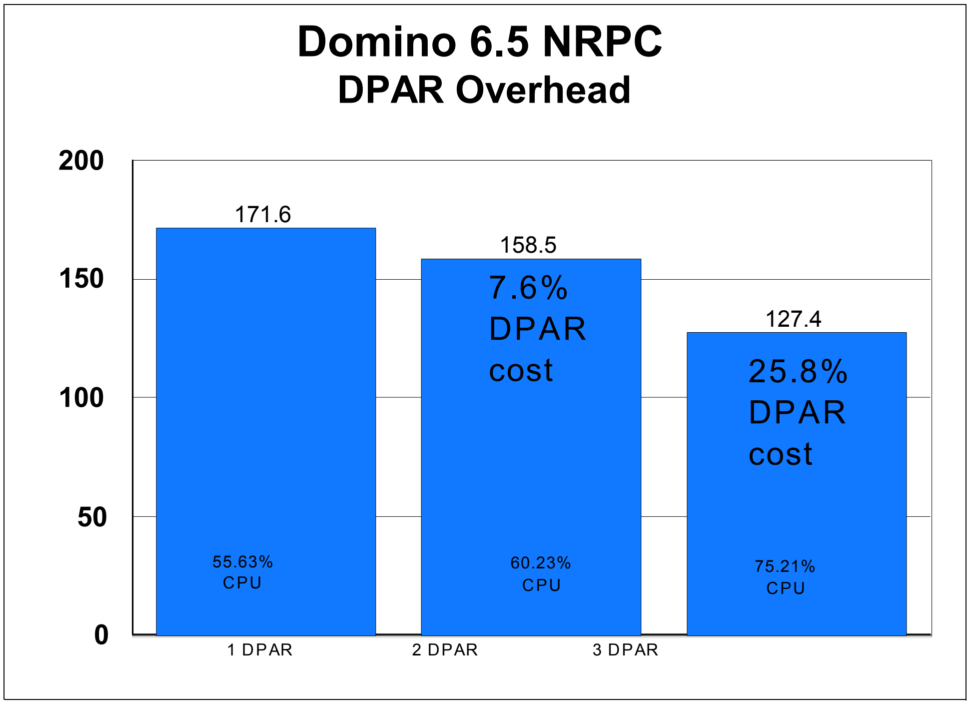

Figure 5-7 NRPC DPAR cost for Domino 6.5

Summary

Based on the low overhead of the Domino 7.0 3 DPAR run compared to the two-DPAR run, a single Linux image would be able to efficiently support up to three DPARs, provided that sufficient memory is allocated to the image (see 5.2.2, “DWA” on page 26, for Domino 7 memory recommendations). This number of DPARs is recommended for an NRPC workload characterized by users with light CPU requirements and high active rates. An NRPC workload characterized by more CPU-intensive clients and a lower activity rate should be able to efficiently support more DPARs within a Linux image.

However, production DPARs typically run with more Domino tasks and third-party tools than what was implemented during the testing for this study. Therefore, the costs of running multiple smaller DPARs versus a large DPAR may be higher in production environments than what benchmark results indicated. Consolidation of smaller DPARs into larger ones will in most cases result in measurable CPU savings.

5.2.2 DWA

In this set of tests, one to four DPARs were driven with 8,000 DWA users. As was the case for the NRPC tests, the 8,000 users were evenly distributed across each multi-DPAR run. Additionally, some testing was done to determine optimal memory configuration for DPARs. Memory was allocated using the following guidelines: <number of DPARS>*2 GB + 2 GB (for the kernel). Little or no paging was seen with these memory allocations.

Measurements

Because the DWA workload scales very well and there were no resource constraints during these runs, varying the number of DPARs had minimal impact on the test results, as shown in Figure 5-8 on page 27 and Figure 5-9 on page 27.

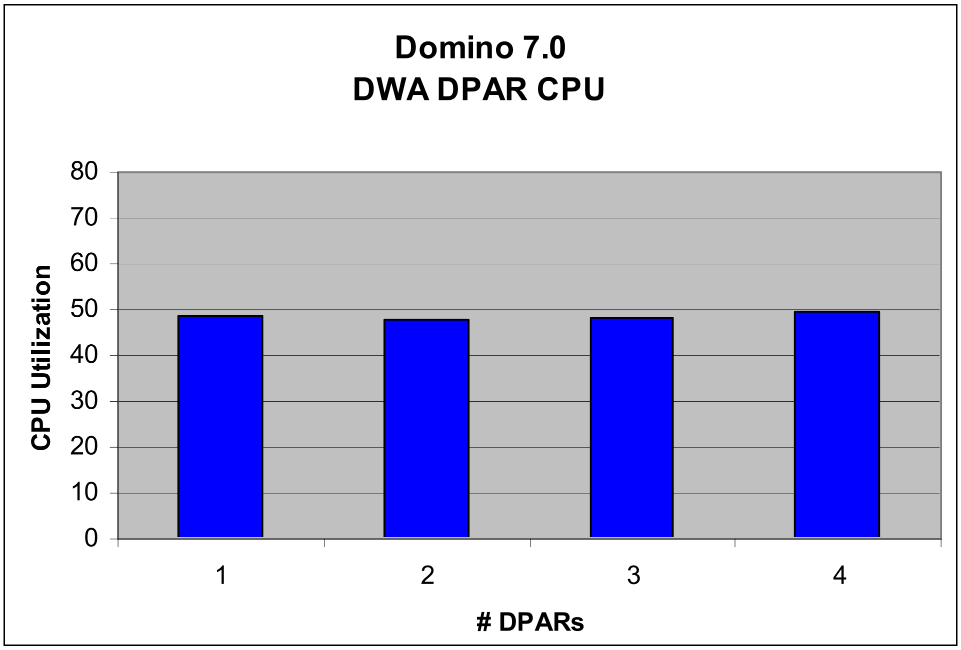

Figure 5-8 DWA CPU utilization for Domino 7

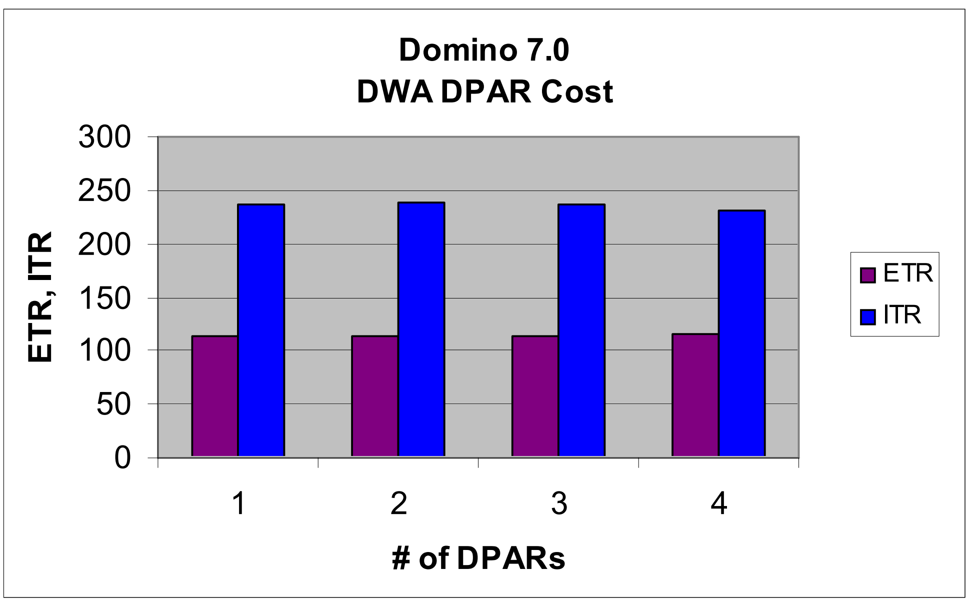

Figure 5-9 DWA DPAR cost for Domino 7

Again, the CPU utilization was somewhat inflated on the one-DPAR run because 8,000 users is close to the DWA user scalability limit for a single DPAR. The CPU increased by 3.7% and the ITR degraded by 3.3% between two and four DPARs. The ETR was flat for all of the runs, showing workload consistency. The small amount of overhead associated with distributing the DWA workload across several DPARs sharply contrasts with the much larger overhead for NRPC users as discussed in the previous section, and can be correlated with the larger number of less-CPU-intensive users. In fact, we would expect DPARs with more-CPU-intensive NRPC clients to behave more like DPARs driven by DWA users.

For this set of tests, we cannot make comparisons with Domino 6.5 because no multi-DPAR DWA runs were executed for this release.

Summary

Based on this set of tests, we can recommend deploying up to four DPARs within a single Linux image. More DPARs per Linux image also might be possible. However, as mentioned earlier, the costs of running many production versus one larger production DPAR may be much higher due to the additional overhead associated with running multiple instances of additional Domino and third-party tasks. Also, a single overloaded DPAR, as was the case for the one-DPAR runs, might incur some additional CPU costs and affect throughput. In general, for production environments we recommend about 1000 active users in a 15-minute interval per DPAR. This number of users scales the DPAR sufficiently to avoid the overhead of many small instances, yet does not stress the DPAR to the point where scaling limitations add cost.

5.3 z/VM Linux guests versus Linux LPARs cost

There are several advantages to deploying Domino in z/VM guests, including enhanced manageability and better performance monitoring and tuning of Linux images. However, there is some CPU overhead associated with virtualization of system resources and providing services to z/VM guests. A series of tests was executed with Domino on z/VM to show the relative effect on throughput and cost of running Domino on z/VM versus running Domino on native LPARs. The same NRPC and DWA workloads, in terms of number of clients, number of CPs, and memory, were executed in each environment to ensure comparable measurements.

To simplify the analysis, the cost of running Domino on z/VM will be discussed solely in terms of the percentage reduction in ITR (Internal Transaction Rate described in Chapter 4, “Background information and terminology for interpreting data runs” on page 15) for the case being compared.

5.3.1 NRPC single guest

The purpose of this set of runs was twofold: measure the cost of a single guest versus a single Linux LPAR running Domino, and to determine the impact of scaling CPs on a single guest versus a single LPAR. For these scenarios, three test cases were run with one, two, and four CPs, keeping the number of users per CP constant for all of the runs. Users were distributed across four DPARs for each test run.

Measurements

The comparison of the results for the LPAR versus the guest NRPC runs showed that the overhead of z/VM for a single guest is affected by the number of CPs being managed by z/VM as well as the characteristics of the workload. In Figure 5-10 on page 29, you can see that the internal transactions per second value was 30 fewer for the one-CP guest run versus the one-CP LPAR run. This is a 14.3% cost for the first guest with one CP. Compared to the corresponding native runs, the cost of the guest with two CPs was 19.85%, and the cost of the guest with four CPs was 19%.

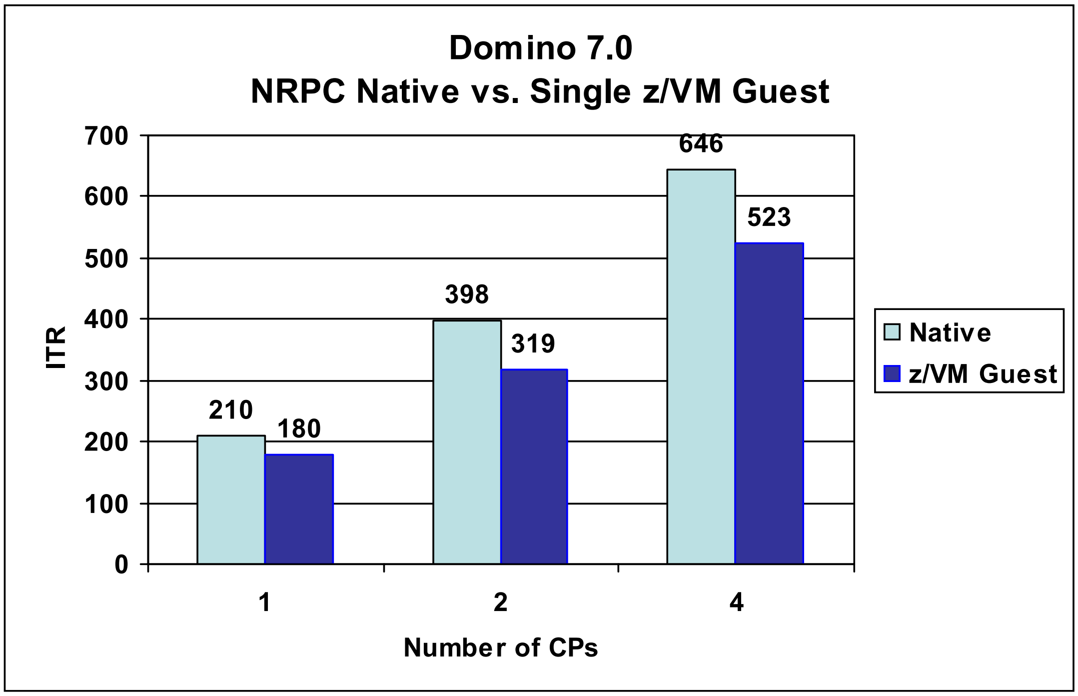

Figure 5-10 NRPC LPAR versus guest cost for Domino 7

For Domino 6.5, a single guest using two CPs was tested and compared to a two-CP LPAR. The ITR measurements for this first guest showed a cost (again reduction in ITR) of approximately 11 percent, which is lower than the cost of the first guest running Domino 7.0. However, it is important to note that the Domino 7.0 z/VM overhead included a much larger number of active users than Domino 6.5. The maximum number of NRPC benchmark users that Domino 6.5 supports in a single guest/LPAR is around 10,500. Some of the Domino 7.0 single-guest and LPAR test scenarios included more than twice this number of users. Also, keep in mind that the Domino 6.5 measurements were performed with a single DPAR within a Linux guest due to the 2 GB memory limitation for the 31-bit Linux OS. For Domino 7.0, four DPARs were defined for all of the runs, which incurred some additional overhead. Additionally, mail routing was enabled for Domino 7.0, which was not the case for the Domino 6.5 testing. With mail routing, a certain percentage of mail items is not delivered locally, but routed to other DPARs for delivery, also driving some additional CPU.

Summary

There is overhead associated in deploying Domino under z/VM because of the virtualization of services. The amount of overhead increases with a lightweight and I/O-intensive workload, such as the NRPC benchmark workload. A more-CPU-intensive and less-I/O-intensive workload, such as the DWA benchmark workload, will incur less overhead (see 5.3.3, "DWA single guest" on page 32). The amount of overhead is also affected by the number of CPs. Based on this study's measurements, for a workload similar to the NRPC workload under consideration, the recommendation is to define up to four CPs per guest. As demonstrated by the test results, after the second CP the overhead levels off. With more CPU-intensive NRPC workloads, more than four CPs per guest might be justified.

The measurements for Domino 7.0 with z/VM 5.2 indicated that the number of users supported on a single guest has increased dramatically, mainly because of the 64-bit OS which provides access to more memory to support multiple DPARs. If in the past you had many guests, each with a small number of users, you would be able to consolidate to larger and fewer guests.

5.3.2 NRPC multiple guests

Multiple guests require more virtualization of resources and management on the part of z/VM than a single guest. These test scenarios measured the CPU overhead of running the same workload on one large z/VM guest versus two and four smaller guests. For these runs, the number of guests was varied; but the total number of real CPs (four in this case), users, and DPARs remained the same across all of the guests.

Measurements

Figure 5-11 shows the results of these tests. The overhead seen with one guest versus a native Domino image remained the same, but the overhead for two guests was reduced slightly with an ITR change of 527 to 562, resulting in a gain of 6.6%. Four guests produced an ITR of 548 or a 4% improvement over the first guest. With four guests, the z/VM cost compared to the native LPAR run was around 15% versus the 19% overhead of the single guest running with four CPs discussed in the previous section. The increase in ITR for multiple guests was due to improvements in Linux page cache hit rates (each guest had 2 GB for kernel usage), which reduces disk I/O and possibly path length per transaction.

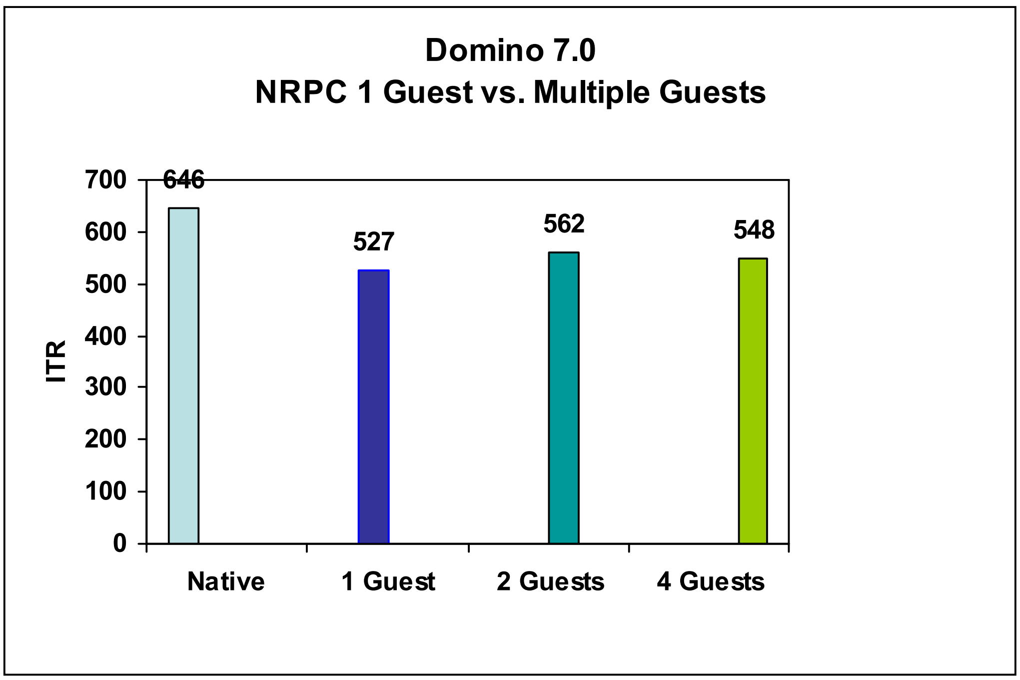

Figure 5-11 NRPC cost of multiple guests for Domino 7

For Domino 6.5, four measurements were collected for one native, one guest, two guests, and eight guests. Figure 5-12 on page 31 shows the results of these tests. Similar to Domino 7, the largest drop in ITR compared to the native run was in the initial instance of z/VM managing a single guest: about 11% for Domino 6.5. After the first guest, the cost of each additional Domino guest, up to eight guests, was about 2%. For the Domino 6.5 testing, all guests were configured with a single DPAR because of memory constraints, and serviced significantly fewer users. Consequently, there were no ITR gains associated with increased page cache availability for the second and subsequent guests as was the case for Domino 7.0.

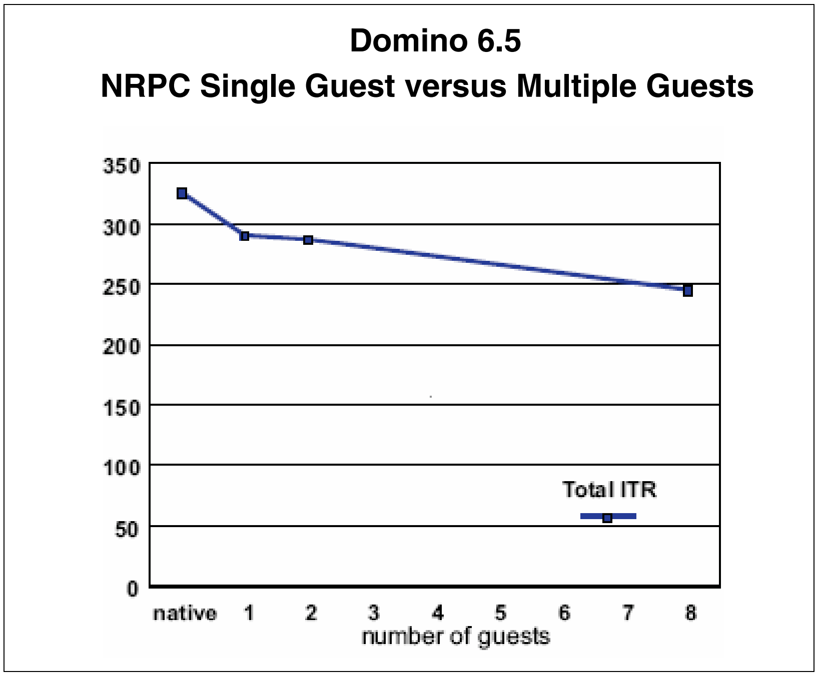

Figure 5-12 NRPC cost of multiple guests for Domino 6.5

For Domino 6.5, we ran into the z/VM I/O bottleneck for a single z/VM LPAR at just over 13,000 users using multiple guests to drive the I/O. In contrast, with Domino 7.0, we were able to drive a single z/VM LPAR with up to 44,000 NRPC users distributed across multiple guests. This run was not included in the previous analysis because of disk contention issues encountered with this number of users. From a z/VM 5.2 point-of-view, however, there were no issues with servicing I/O requests for 44,000 users. If additional disk had been available, the z/VM LPAR would have possibly been able to support more users and process more I/O requests.

Summary

In general, if an application has good scalability, then there is a small measurable cost for each additional z/VM guest. This was not the case with Domino 7.0. The measurements with up to four guests showed that the ITR and cost actually improved as guests were added. This gain can be partly attributed to improved Linux page cache hits. Therefore as the number of users was spread over a larger number of guests, the total ITR improved and the smaller guests with fewer users became more efficient than the first guest. So, the cost of z/VM went down as additional guests were added.

These observations are the basis for the recommendation that it is better to add guests at some point than to drive one guest to a very large number of active users. For production systems, we typically recommend around 1000 active users per Linux DPAR and three to four DPARs per Linux image, which is an efficient configuration and should be appropriate for most deployments.

The number of Domino users that can be supported by a single z/VM LPAR has increased significantly because z/VM 5.2 performs I/O above the 2 GB line. If in the past you were forced to implement multiple z/VM LPARs to support a large number of Domino 6.5 users, you will now be able to reduce the number of LPARs needed with Domino 7 and z/VM 5.2.

5.3.3 DWA single guest

For this scenario, a single native LPAR with eight CPs running four DPARs was compared to a similarly configured z/VM guest driven by the same number of DWA users. Figure 5-13 shows the results of the DWA workload test case. The ITR for the native run was 225.4 versus 208.6 for the single z/VM guest, showing a cost of 7.4%.

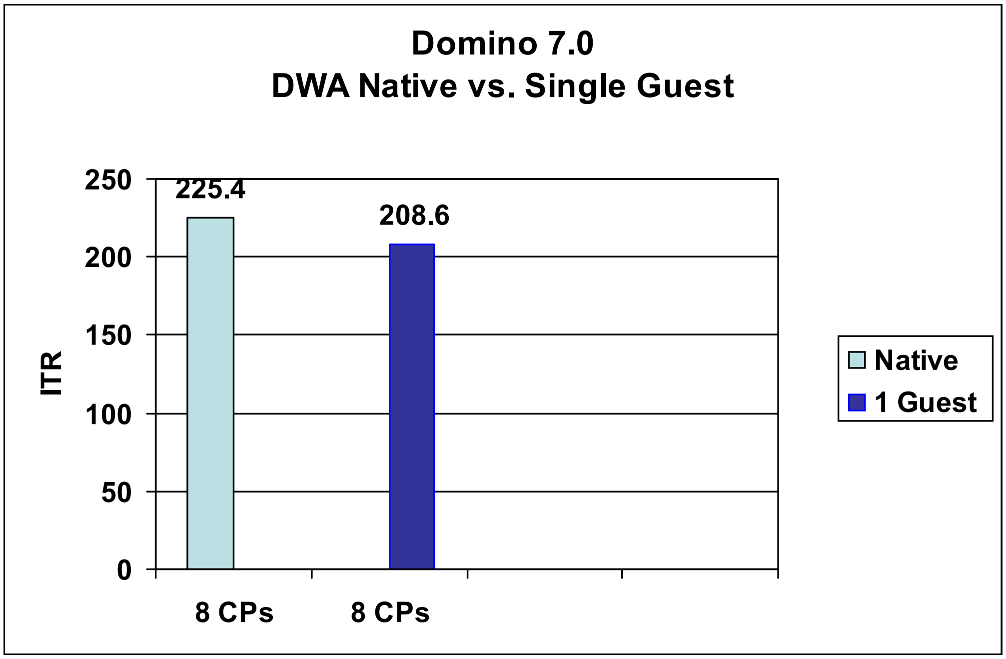

Figure 5-13 DWA LPAR versus guest cost for Domino 7

Measurements

For Domino 6.5, the cost of the first guest was around 3%, a little less than half the cost of the Domino 7.0 first guest. However, when comparing the Domino 7.0 and 6.5 single guest overheads, we must again take into consideration that the Domino 7.0 test case was run with four DPARs, significantly more users, and mail routing enabled; all of these elements added some CPU overhead.

Summary

The cost of the first z/VM guest with the more-CPU-intensive DWA workload was about half of what was measured for the NRPC single guest because a higher percentage of time is spent in the application rather than I/O processing with this workload.

The NRPC cost for typical production systems is somewhere between our NRPC and DWA results, most likely closer to DWA, because the NRPC benchmark workload is so much more I/O-intensive on an overall basis than production environments. The increased I/O activity in the NRPC benchmark workload scenario results in an increased usage of z/VM services.

5.3.4 DWA multiple guests

The multiple-guest DWA scenario was executed on an eight-CP z/VM LPAR with the number of guests being varied while all other variables were held constant. The users, DPARs, and virtual CPs were distributed equally among all of the guests for one-guest, two-guest, and four-guest runs.

Measurements

In Figure 5-14, the ITR value for the first DWA guest was 208.6. This was a 7.4% degradation for this guest compared to the native LPAR results. Compared to the first guest, the cost improved (degradation was less) for the two-guest and four-guest runs: 2.3% for the second guest and 3.4% for the fourth guest. For the four guests combined, the total ITR degradation was roughly 4.3% compared to the native Linux LPAR. Similar to the NRPC runs, the cost reduction for multiple guests versus a single guest was due to improvements in Linux page cache hit rates (each guest had 2 GB for kernel usage).

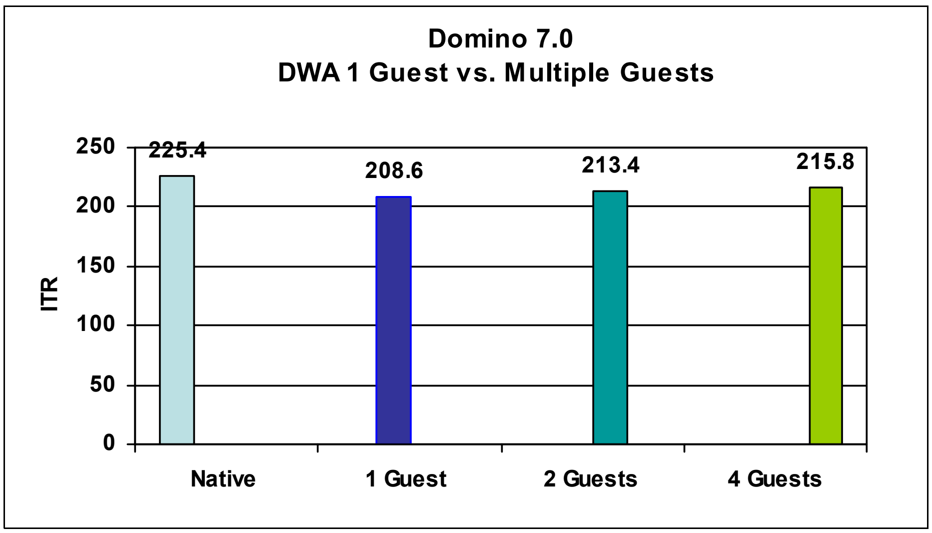

Figure 5-14 DWA cost of multiple guests for Domino 7

We collected five Domino 6.5 DWA measurements consisting of one native, one guest, two guests, eight guests, and 16 guests.

Figure 5-15 on page 34 shows the results of these tests. The largest cost was in the initial instance of VM managing a single guest. The cost of the first Domino 6.5 guest was approximately 3%, and the cost of each additional guest, up to 16 guests, was less than 1% per guest.

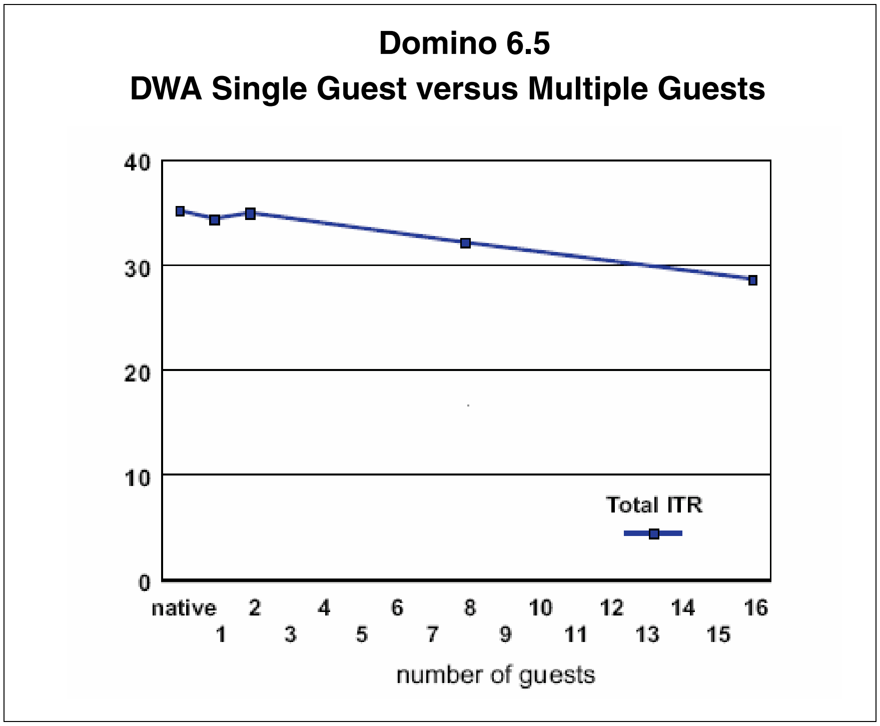

Figure 5-15 DWA cost of multiple guests for Domino 6.5

Summary

Again, the recommendation is that it is better to add guests at some point than to drive one guest to a very high number of active users. Using the guideline of 1000 active users per DPAR, we would recommend up to three or four DPARs per guest for production environments, provided that the guest has access to sufficient memory to limit swapping. Given the relatively low z/VM overhead of a more-CPU-intensive workload, it might even be appropriate to deploy more than four DPARs per guest.

5.4 Impact of virtual-to-real CP ratios for z/VM guests

In this scenario, the z/VM LPAR had four dedicated real CPs. Measurements were taken with four guests while changing the number of virtual CPs given to each guest. The ratio of virtual CPs to real CPs was varied while keeping other variables constant, such as the number of users, number of DPARs, and other configuration parameters. Measurements were taken with 1:1, 2:1, and 4:1 virtual-to-real processor ratios, the last two scenarios introducing CP overcommitment factors of two and four for active guests.

Measurements

The 1:1 run produced a baseline with an ITR of 564 (shown in Figure 5-16 on page 35). When the virtual CP ratio was varied to 2:1 (two virtual processors per guest), the resulting ITR measurement was 547, a 3% decrease or overhead. With four virtual processors per guest, the ITR was 524, a 7.1% overhead from the base measurement.

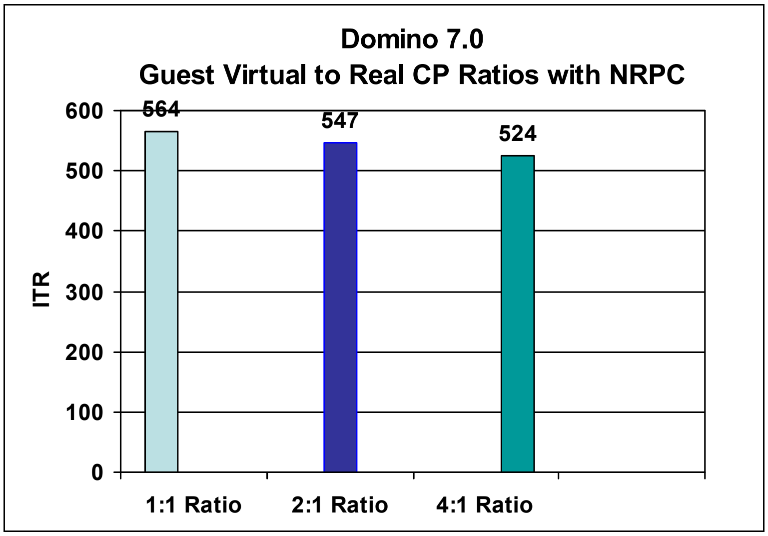

Figure 5-16 Virtual-to-real CPs with Domino 7

Domino 6.5 comparison

The Domino 6.5 measurements showed that virtual-to-real CP ratios of 2:1 or less have little effect on ITR, and ratios of 4:1 add an ITR cost of 4-6%. This cost is not significantly different from the Domino 7.0 results, given that these runs were executed with a larger number of users, four DPARs per guest, and mail routing enabled.

Summary

The Domino 7.0 measurements show that the ratio of virtual to real processors should be kept at 4:1 or preferably less to minimize overhead costs.

5.5 Maximum number of NRPC users in Linux LPAR

A series of tests was run to find the maximum number of NRPC users that Domino 7 could support in an LPAR. Unlike all of the other runs for this study, these specific runs were done without transaction logging. Domino transaction logging incurs some CPU overhead; without it, more benchmark users can typically be supported in a test. However, we do not recommend disabling transaction logging in production environments. Its benefits far outweigh the CPU cost: faster restarts, better data integrity, and support of more production users because of asynchronous I/O.

Measurements

Using eight DPARs, we were able to demonstrate that Domino 7 can support 50,000 NRPC users in a single LPAR. Although more CPU was still available with 50,000 users, Domino did not scale beyond this number in our tests because of poor DASD response times. There was not enough hardware available to tune the disk to support a larger number of mail files and achieve acceptable response times.

During Domino 6.5 testing, memory constraints limited a single Linux image to 10,000 NRPC users.

Summary

Compared with Domino 6.5, Domino 7.0 showed significant improvements with its ability to support multiple DPARs and up to 50,000 NRPC users within a single Linux image.

6

Conclusions and recommendations

Among many others, the following important points were uncovered as a result of the Domino 7.0 study:

- Domino 7 supports more users in a single Linux image than Domino 6.5.
- z/VM 5.2 is able to process more Domino I/O than previous z/VM releases.
- Domino 7 has a better total cost of ownership (TCO) than Domino 6.5.

Additionally, there are several new configuration guidelines for optimizing Domino 7 performance and capacity utilization relating to:

- Number of CPs per Linux image
- Number of DPARs within a Linux image
- Memory and swap space for Domino and Linux
- Number of z/VM guests
- Number of virtual versus real CPs with z/VM guests

6.1 Linux memory and z/VM bottlenecks removed

- Domino 7 supports more users in a single Linux image than Domino 6.5

  With Domino 6.5, we were only able to support 10,000 NRPC benchmark users on a single Domino server running in a Linux image because of the 2 GB memory limitation by the 31-bit supported Linux OS. Domino 7.0, running on a 64-bit Linux kernel, showed significant improvements with its ability to support multiple DPARs and up to 50,000 NRPC benchmark users within a single Linux image. It should be noted that the 50,000-user cap is no way a Domino or Linux limitation; additional users could have been supported during testing if more hardware had been available for this study.

- z/VM 5.2 is able to process more Domino I/O than predecessor z/VM releases

  Previous tests with Domino 6.5 and z/VM 5.1 showed that z/VM started to experience serious I/O bottlenecks at about 13,000 NRPC benchmark users because of the 2 GB constraint of z/VM 5.1 and its predecessor releases. This issue has been fixed in z/VM 5.2, which greatly improves I/O throughput. With Domino 7.0, we could show that a single z/VM 5.2 LPAR can support up to 44,000 NRPC users without experiencing I/O issues.

- Domino 7.0 has a better TCO than Domino 6.5

  Domino 7 support of a 64-bit Linux kernel and the I/O throughput improvements for z/VM 5.2 will enable you to consolidate your Domino Linux images (LPARs or z/VM guests) and to run with fewer z/VM LPARs if z/VM is part of the solution. Domino 7 and VM 5.2 greatly improve the TCO for running Domino on Linux for System z.

6.2 Domino scalability

- There are differences between the NRPC and DWA benchmark workloads

  A series of tests was performed to see how linearly the Domino Notes (NRPC) and browser (DWA) workloads scale as the number of users and CPs were increased. These results were compared to standard LSPR (Large Systems Performance Reference) workloads, which are used to correlate capacity on zSeries and z9 processor models.

  To put the test results into proper perspective, it is necessary to understand that the NRPC and DWA workloads have very different characteristics. The NRPC workload is comprised of very lightweight users from a CPU perspective whose overall I/O demands are much larger than those for DWA. DWA is a much more CPU-intensive workload than NRPC (roughly using four times or more as much CPU) because all of the processing is moved to the Domino server. DWA clients also consume more memory than NRPC clients. Consequently, for a given number of CPs, DWA supports fewer users on a Domino server than NRPC, which implies that the overall I/O demands are also lower.

- Domino workload characteristics influence scalability

  Compared to standard LSPR zSeries workloads, the NRPC benchmark workload did not scale well, even at four CPs. However, the poor scalability of the NRPC benchmark workload is not believed to be a problem for production environments because the more CPU-intensive DWA workload scaled very well when compared to LSPR workloads. In fact, in tests of up to eight CPs, it scaled slightly better than the best-performing standard LSPR workload. Compared to production environments, the NRPC benchmark workload is very light in terms of CPU load; most Domino deployments have users with CPU demands at least four times as heavy as NRPC benchmark users, which means fewer concurrent users with less overall I/O and memory demands. Therefore, in a production environment, we would expect the scalability of NRPC clients to be much closer to that of DWA benchmark clients.

- Guidelines for running Domino 7 to maximize scalability
  - For NRPC clients, define up to four CPs for a Linux image.

    If the CPU requirements for NRPC users are very robust, then more CPs per Linux image may be appropriate.
  - For DWA users, define up to eight CPs for a Linux image.

    We did not run any scalability tests with more than eight CPs; so, more CPs might be possible for a Linux image running this type of workload.
  - For large user deployments, scale Domino horizontally by implementing multiple four-CP Linux images for NRPC clients and multiple eight-CP Linux images for DWA clients.

    Again, the number of CPs per Linux image might vary depending on the CPU intensity of users.
  - One critical bottleneck is the transaction logging file. Place it on a device with a high I/O bandwidth, and make sure that no other processes perform I/O on this device. The flow of sequential I/O must not be interrupted by other I/Os.

6.3 Running with multiple Domino servers (DPARs)

This study's test results indicated that there are costs associated with running multiple smaller DPARs versus a larger DPAR. This is not surprising because every Domino server is comprised of multiple tasks, each task invoking one or more processes. Domino servers continuously poll for work, which implies that no Domino server, even a server without client requests, is ever idle.

Based on the results of this study, we recommend that you:

- Run three or four DPARs per Linux image for NRPC clients.

  An NRPC workload characterized by more CPU-intensive clients and a lower active user rate should be able to efficiently support more DPARs per Linux image. Most of our tests were executed with four DPARs per Linux image.
- Run four DPARs per Linux image for DWA clients.

  More DPARs might be possible as long as each DPAR is sufficiently scaled with users and has enough memory to limit swapping. Most of our tests were executed with four DPARs per Linux image.
- Allocate enough memory to the Linux image to limit swapping from disk, which can significantly increase the overhead of the DPAR.
  - Based on testing, the recommendation is 2 GB of memory for each fully-loaded DPAR and 2 GB of memory for Linux.

    By allocating 2 GB for Linux, there is additional page cache, which might minimize real I/O.

    DPARs with only a few hundred active users might have smaller memory requirements provided that a minimal number of Domino tasks is running on the Domino server. Domino servers running extra tasks, such as addin tasks or multiple instances of the replicator and indexer, will need the full 2 GB to avoid issues with swapping.
  - Define a minimal amount of disk swap space to allow Linux to move out unused storage pages. This frees up space to hold more frequently used pages, such as from files in memory. A small amount of Linux swapping does not indicate a memory issue.

- Scale Domino servers as close as possible to 1000 active 15-minute (not connected) users.

  Active 15-minute users are users who have requested Domino services at least once in a 15-minute period. The recommendation of 1000 active users per Domino server is based on our experience with production deployments, both internal and external to IBM. We consider servers with this number of active users to be fully loaded. Smaller DPARs with fewer than 1000 active users will incur additional CPU overhead.

6.4 Domino and z/VM

- z/VM 5.2 supports more Domino users with fewer z/VM LPARs and guests

  The objectives of the Domino z/VM measurements were to show the scalability of Domino on z/VM and the throughput that can be achieved with z/VM 5.2 and Domino 7.0. Based on this study's test results, the conclusion is that Domino 7 and z/VM 5.2 can support a significantly larger number of users in a single z/VM guest than Domino 6.5 and predecessor releases of z/VM. If in the past you had many guests, each with a small number of users, you will now be able to consolidate to fewer guests. Also, if you had to implement multiple z/VM LPARs to support a large user community, you will now be able to reduce the number of LPARs.

- The cost of running Domino on z/VM is dependent on several variables:

  – The characteristics of the workload

    There is overhead associated in deploying Domino under z/VM because of the I/O service requests to z/VM and the virtualization of other resources. For NRPC benchmark users, this overhead is somewhat high because the workload is a very lightweight in terms of CPU and uses a high percentage of I/O service calls per transaction. When a workload is more CPU-intensive and generates fewer I/O requests, such as DWA, then the cost of z/VM is lower.

    In our tests, the cost of running with Domino NRPC clients in a single z/VM guest versus a Linux LPAR ranged from around 14 to 19%. The low end of the range was with a single CP, and the high end of the range with four CPs. With a comparable number of DWA users running on four CPs, the z/VM overhead for a single guest was only about 7%, less than half the cost of the NRPC scenario. Given that production NRPC clients tend to use more CPU and have lower active rates, we would expect production z/VM costs to be closer to DWA benchmark results.

  – The number of real processors VM is managing

    As the number of additional processors being managed by z/VM increases, the cost of z/VM increases. This was evident from the range of z/VM overheads observed when varying the number of CPs from one to four. See 5.3.1, "NRPC single guest" on page 28 for more information.

  – The ratio of the virtual to real processors being managed by z/VM

    We saw an overhead of about 7% when going from a 1:1 to a 4:1 ratio. Processors are typically overcommitted for most z/VM deployments. To limit the overhead, do not go beyond the 4:1 ratio and preferably stay as close as possible to the 1:1 ratio.

- The number of guests being managed by z/VM

  If the application has good scalability, then there is typically a small but measurable cost for each additional guest. This was not what we saw with Domino 7.0. Given a specific workload, measurements with up to four guests showed that the cost actually improved as guests were added up to four guests. The costs of driving four guests with NRPC users versus a single guest were 15% and 19%, respectively. Similarly, the costs of four guests with DWA versus a single guest were 4% and 7%, respectively. These results were in part due to the availability of additional file cache with more Linux guests.

Based on the results of this study, when running with z/VM we make the following recommendations:

- Assign up to four CPs for individual guests.

  More CPs per guest might be possible for customer environments that tend to have heavier and consequently fewer active users per guest.

- Deploy multiple guests rather than driving one guest to a very high number of users.

  Using the general rule of 1,000 active users per DPAR, implement anywhere from three to four DPARs per production Domino guest with enough memory to limit swapping.

  This configuration will avoid overloading a single guest.

- Keep the virtual-to-real processor ratio at 4:1 or less.
- Use the same memory recommendations as documented in 6.3, “Running with multiple Domino servers (DPARs)” on page 39.

Glossary

CP. Central Processor

DASD. Direct Access Storage Device

DPAR. Domino Partition (an instance of a Domino server)

DWA. Domino Web Access is one of the two types of user access used in the benchmark workload. It is a CPU-intensive browser type of user access.

ETR. External Transaction Rate: transaction rate measured per second of execution.

ITR. Internal Transaction Rate: the maximum number of transactions that could be run if the processor were running at 100% of capacity.

LPAR. Logical Partition: hardware partitioning

LSPR. Large System Performance Ratio: the scalability ratio between IBM systems of different types or models.

NRPC. Notes Remote Procedure Call: one of the two types of user access used in the benchmark testing. It is an I/O-intensive user access emulating a Notes client.

SAR. Linux System Activity Report

SLES. SUSE Linux Enterprise Server

SP. Service Pack

Users. In this document, “users” refers to workload users for NRPC or DWA benchmark workloads only. It does not imply the number of production users that can be supported by a particular configuration.

z/VM. Software product that provides hardware virtualization services to emulate a virtual system environment for multiple hosts.

Related publications

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

IBM Redbooks

For information on ordering these publications, see "How to get IBM Redbooks" on page 45. Note that some of the documents referenced here may be available in softcopy only.

- *Best Practices for Lotus Domino on System z: z9 and zSeries*, SG24-7209
- *IBM Lotus Domino 6.5 for Linux on zSeries Implementation,* SG24-7021

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

Lotus Domino 7 on Linux for IBM System z

Capacity Planning and Performance Updates

Redpaper

Scalability under Linux

Performance under z/VM

Multiple Domino servers

In early 2005, the Scalability Center for Linux on zSeries (SCL) in Poughkeepsie performed a study on IBM Lotus Domino 6.5 for Linux on IBM System z. During that study, capacity planning and performance data was collected using an industry-standard benchmarking tool to drive both NRPC (Notes) and Domino Web Access (HTTP) e-mail users. The results of this benchmark showed two bottlenecks that prevented linear scalability of throughput for Domino 6.5 for Linux on System z. Domino 7.0 and z/VM 5.2, which both became available later in 2005, offer relief for these two bottlenecks.

This IBM Redpaper discusses the results of those tests and provides some recommendations.

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