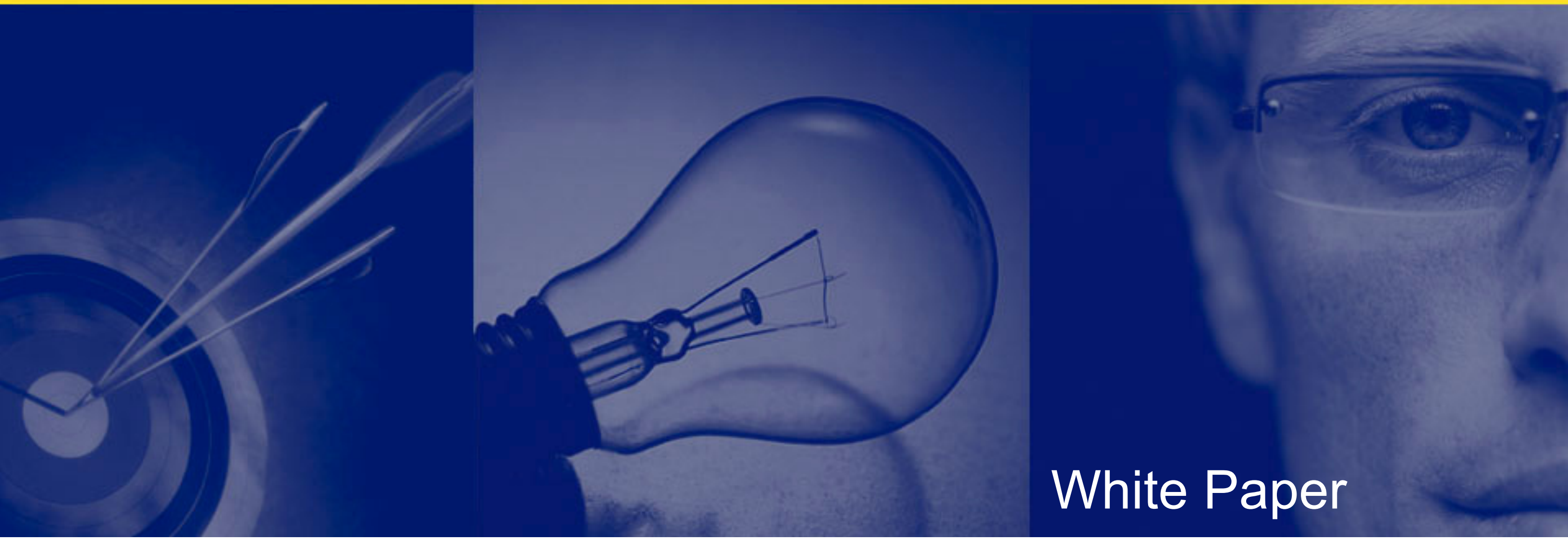
10 Simple Steps for Boosting Database Performance in a Virtualized Environment

Written by
Dr. Bert Scalzo
Quest Software, Inc.



WARRANTY

The information contained in this document is subject to change without notice. Quest Software makes no warranty of any kind with respect to this information. QUEST SOFTWARE SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTY OF THE MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Quest Software shall not be liable for any direct, indirect, incidental, consequential, or other damage alleged in connection with the furnishing or use of this information.

TRADEMARKS

World Headquarters
5 Polaris Way
Aliso Viejo, CA 92656
e-mail: info@quest.com

Please refer to our Web site (www.quest.com) for regional and international office information.

January 2009

Contents

INTRODUCTION 1

BACKGROUND 2

COST OF USING DEFAULT SETTINGS 3

BENEFITS OF MODIFYING DEFAULT SETTINGS 4

10 STEPS FOR IMPROVING PERFORMANCE 5

Optimize the VM Host and Client OS Configuration and Options 5
Adjust VM Client File System Block Size 6

SUMMARY 8

ABOUT THE AUTHOR 9

ABOUT QUEST SOFTWARE, INC. 10

Contacting Quest Software 10
Contacting Quest Support 10

INTRODUCTION

Many people swore that databases on virtual machines would not fly—or at best, would be accepted very, very slowly. I was not one of those people, so I wrote a book on this very subject: *Oracle on VMware: Expert Tips for Database Virtualization*[1]. However, I don't expect people in these tough economic times to go out and buy every new book. So in this white paper, I'll share some other papers worth reading on the subject and show how you can significantly improve database performance simply by adjusting infrastructure default settings in the virtualized technology stack.

[1] www.rampant-books.com/book_0801_oracle_vmware.htm

BACKGROUND

One of the first—and most robust—papers I've seen about databases on virtual machines is from VMware itself: "Deployment of Oracle Databases on VMware Infrastructure[2]." This fine paper is 119 pages of highly useful DBA information for effectively and efficiently deploying Oracle databases on a VMware infrastructure.

At first Oracle was slow to fully embrace databases on virtual machines. But then the company débuted its own virtual machine solution, and things changed. There are now papers that show benchmarks of virtual machines versus bare metal[3] that are achieving acceptable results. There are even papers for successfully deploying RAC (Oracle Real Application Clusters) on virtual machines.[4] So the trend is changing, and virtualization is no longer verboten for databases.

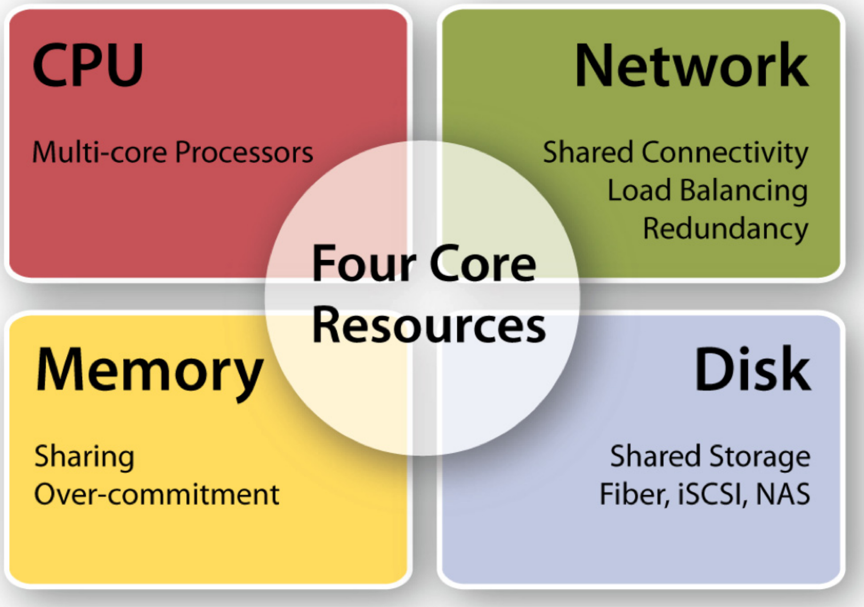
[2] www.vmware.com/partners/alliances/technology/oracle-database.html

[3] www.oracle.com/technologies/virtualization/docs/ovmbenchmark.pdf

[4] www.oracle.com/technology/products/database/clusterware/pdf/oracle_rac_in_oracle_vm_environments.pdf

Cost of Using Default Settings

At Oracle OpenWorld 2008, I presented material that shows that deploying Oracle on VMware with defaults across the board can cost you as much as **440%** in terms of achievable performance. In a virtualized environment, we must approach configuration, optimization and tuning in a slightly different manner, because we must now consider how the four core resources (shown below) are shared.



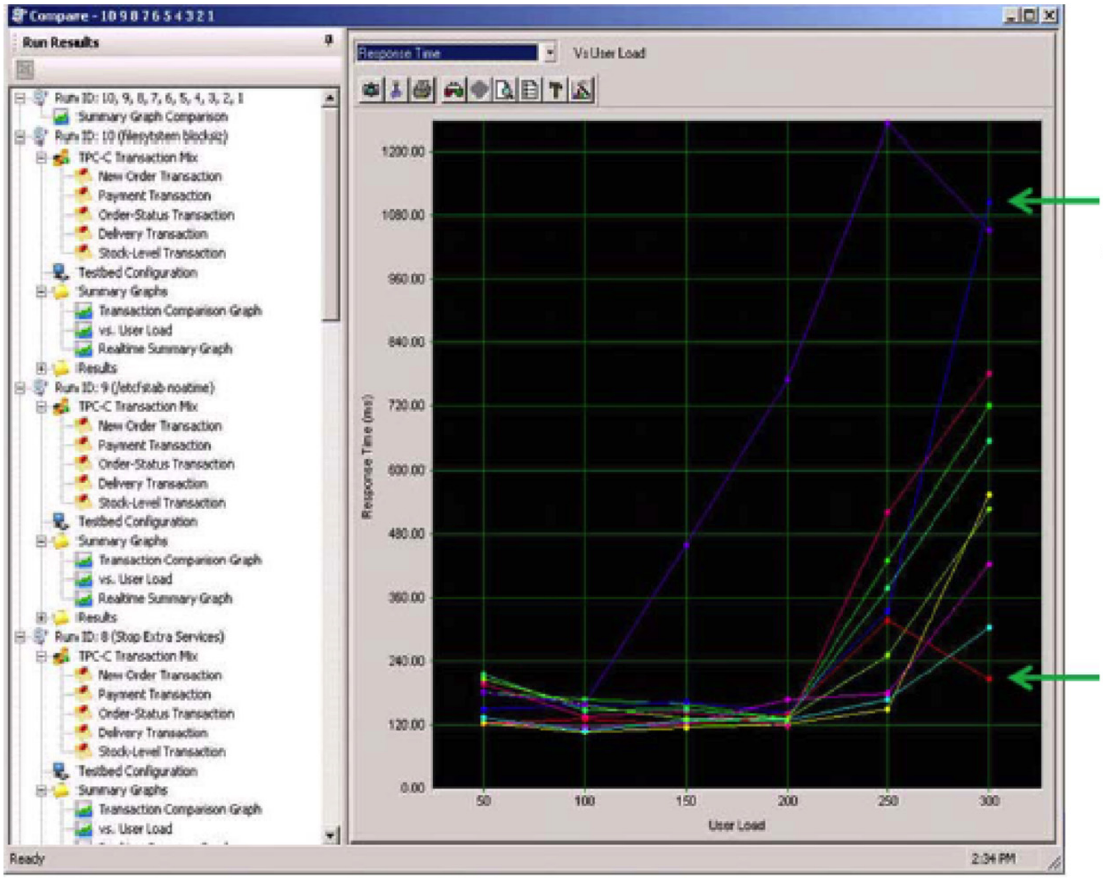
A fundamental challenge is understanding the impact of sharing of the four core resources.

BENEFITS OF MODIFYING DEFAULT SETTINGS

I was able to easily modify default settings and obtain a **440%** improvement for the TPC-C OLTP benchmark (your mileage may vary, of course). I used the following Quest tools to perform these tasks:

- Benchmark Factory® to generate a database load
- Spotlight on Oracle® and Foglight® Performance Analysis to monitor and diagnose database performance
- Foglight® for VMware to monitor and diagnose virtual machine performance

See below how the response time for 300 concurrent users decreased from 1080 ms to 200 ms. And while 1080 ms (i.e., one second) may have been acceptable, what would happen if the user load increased to 1,000 or more users? Obviously the tuned scenario will scale much better to handle a higher concurrent user load.



1080 ms Average Response Time

200 ms Average Response Time

Proper setup with modified default settings yielded performance improvement of 440%.

10 Steps for Improving Performance

So how did I get these results? I followed these 10 simple steps.

(Note that the VM host was running Windows 2003 Server, but similar issues would apply had it been Linux or some other OS.)

On the VM host:

1. Obtain a baseline test for relative comparisons.
2. Exclude VM clients from active, online virus scans.
3. Remove the Windows Indexing Service (because fast file searches aren’t needed).
4. Remove other extraneous Windows services.
5. Change the VM host registry settings to improve file system IO for databases.
6. Optimize the VM host configuration and options.

On each VM client:

7. Optimize the VM client OS configuration and options for Oracle databases.
8. Remove extraneous VM client OS services and daemons.
9. Change VM client file system settings to improve IO performance for databases.
10. Adjust VM client file system block size to more closely match Oracle block size.

My key contention is that as long as there are sufficient resources and bandwidth to handle the net requests, requiring database servers to be islands unto themselves is passé. Even if there is a 10–20% overhead (and I’m inclined to say it’s closer to 10), with cheap hardware these days, the benefits far outweigh the negatives—just spend a little more and get a bigger server to handle multiple databases. But it may be a few years before this opinion is more widely accepted.

The remainder of this paper focuses on three key steps from the list above. These particular steps can be taken in almost any environment and typically result in large performance gains. The other steps are important, and cumulatively they can have a significant effect on performance. But the three outlined in more detail are where you’re most likely to get the biggest bang for your buck.

Optimize the VM Host and Client OS Configuration and Options (steps 6 and 7)

In reality, this is the same improvement applied to two different virtualization levels: the host and each client. Thus, I’ll simply explain how to accomplish this technique for Windows and Linux, and leave it to you to apply it properly across the various virtualization layers.

There are many possibilities for adjusting the OS configuration and options. In the interest of time, I'll focus on one that provides good returns but is often overlooked.

Both the Windows NTFS and Linux ext2/3 file systems maintain a lot of metadata related to file access, such as date created and last time updated. So, an IO request might actually generate multiple physical IOs: one for the data file and one or more for the related metadata. In the case of the VMware host, we really don't care to keep detailed OS file system information about the hosted clients' data file access—it's neither useful nor critical (unless we're doing snapshots). And at the Oracle database level, we know Oracle accesses its files every so many seconds, so why pay to update the metadata with that information? Oracle keeps its own timestamps (i.e., SCN) in the file headers.

For Windows, we simply adjust the following registry entry:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\
Control\FileSystem\NtfsDisableLastAccessUpdate = 1
```

For Linux, there are several ways to accomplish this. We can set the attribute for the individual Oracle files as follows:

```
chattr +A file_name
```

Or we can do it for an entire directory:

```
chattr –R +A directory_name
```

However, the best method (because it automatically handles any file additions) is to edit the /etc/fstab and add the NOATIME attribute:

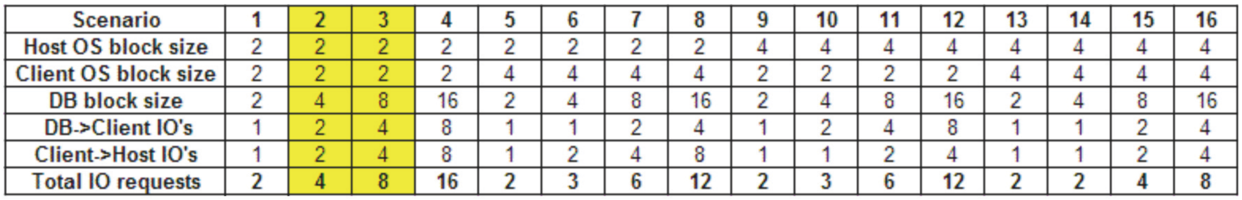
```
/dev/sda6     /            ext3      defaults,noatime    1 1
/dev/sda1     /boot        ext2      defaults,noatime    1 2
/dev/cdrom    /mnt/cdrom   iso9660   noauto,owner,ro     0 0
/dev/sda5     swap         swap      defaults            0 0
```

Adjust VM Client File System Block Size (step 10)

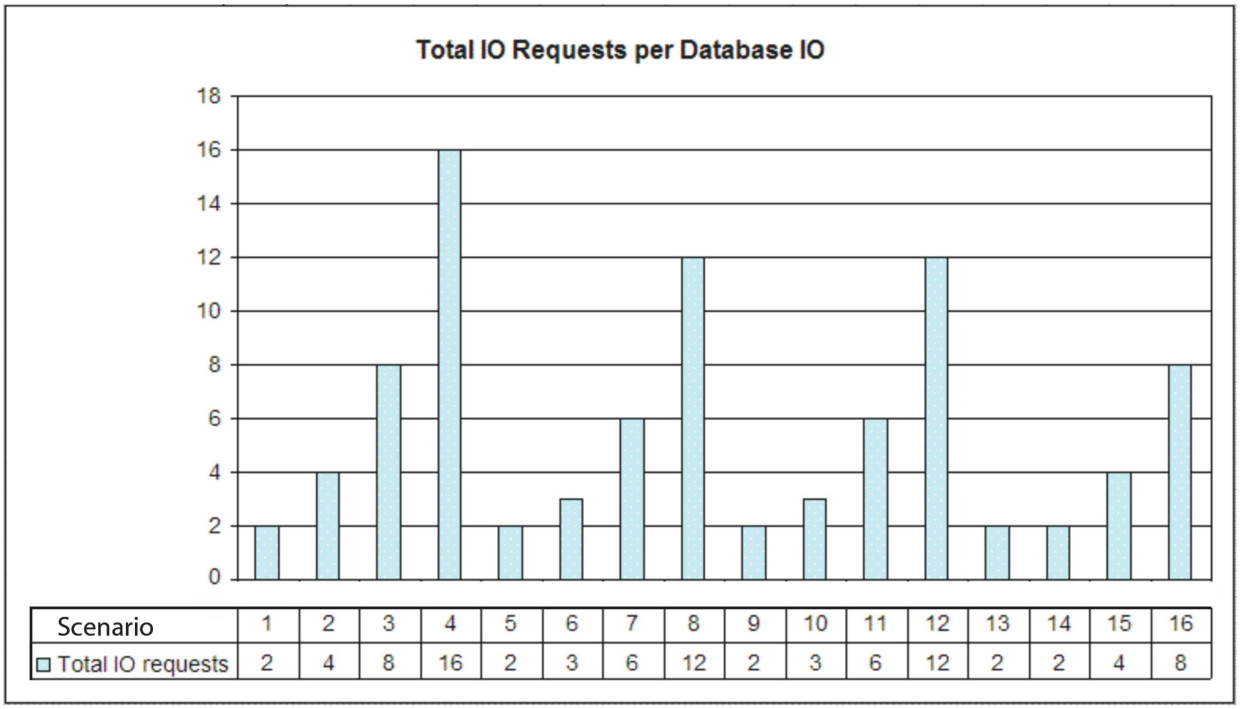
The default block sizes for the host and client OS file systems are generally not the same as your Oracle block size (although hopefully the Oracle block size is a multiple of the OS file block size). Since the host may be servicing multiple Oracle databases with different block sizes, servicing different database platforms (e.g., Oracle and MySQL), or hosting other applications (e.g., web server), we cannot always make this adjustment at the host level. However, we can generally make the adjustment for each Oracle database client.

Suppose we have a Linux host running a Linux-based Oracle database client. Consider **scenario #2** from the chart below. We create the host file system using the default block size of 2K. And we do the same thing on the client—but we size the database blocks at 4K. Thus, each Oracle physical IO request asks the client OS for two IOs, and each client IO asks the host for one IO. That's a total of **four logical IO requests** (although there are only two physical IOs in reality. Note that there is overhead for each logical IO request, so larger numbers are worse.) If the database block size had instead been 8K while the other factors remained at 2K (i.e., **scenario #3**), then the logical IOs would instead be DB -> Client = 4 and Client -> Host = 4 for a grand total of **eight logical IO requests**.

Here's a basic chart identifying some common combinations (and their sometimes surprising overheads):



Scenario	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Host OS block size	2	2	2	2	2	2	2	2	4	4	4	4	4	4	4	4
Client OS block size	2	2	2	2	4	4	4	4	2	2	2	2	4	4	4	4
DB block size	2	4	8	16	2	4	8	16	2	4	8	16	2	4	8	16
DB->Client IO's	1	2	4	8	1	1	2	4	1	2	4	8	1	1	2	4
Client->Host IO's	1	2	4	8	1	2	4	8	1	1	2	4	1	1	2	4
Total IO requests	2	4	8	16	2	3	6	12	2	3	6	12	2	2	4	8



Obviously, there are some very bad choices in the chart above. For example, a seemingly good 16K database block size might actually result in a total of 12 to 16 logical IO requests across the virtualization infrastructure under the worst-case scenario. While it's still effectively just 16K that's being read, the extra overhead added by this mismatch only serves to multiply the negative effects. So choose wisely.

SUMMARY

This paper demonstrated that you can improve database performance up to **440%** simply by adjusting various infrastructure default settings across the virtualized technology stack. Of course, your mileage may vary, but the key point is that taking the time to properly layer your virtual infrastructure settings can have a substantial impact, which is magnified by the net effect of a virtual server hosting multiple database servers. And while my example was specific to Oracle, the same principles generally apply to any database being deployed on a virtual server.

ABOUT THE AUTHOR

Dr. Bert Scalzo is a database domain expert for Quest Software and a member of the Toad™ team. Bert is also an Oracle ACE, a designation that recognizes members of the Oracle technology and application communities for their technical proficiency and willingness to share their knowledge and experience. He has worked with Oracle databases for well over two decades; his key areas of interest are data modeling, database benchmarking, database tuning and optimization, star schema data warehouses and Linux. Bert is the author of several books, including *Oracle on VMware: Expert Tips for Database Virtualization*, *Advanced Oracle Utilities: The Definitive Reference* and *Oracle DBA Guide to Data Warehousing and Star Schema*. He has written articles for many online media outlets and publications, including *Oracle Technology Network* (OTN), *Oracle Magazine* and *PC Week* (eWeek). Bert is a regular presenter at Oracle conferences and user groups.

ABOUT QUEST SOFTWARE, INC.

Quest Software, Inc., a leading enterprise systems management vendor, delivers innovative products that help organizations get more performance and productivity from their applications, databases, Windows infrastructure and virtual environments. Quest also provides customers with client management through its ScriptLogic subsidiary and server virtualization management through its Vizioncore subsidiary. Through a deep expertise in IT operations and a continued focus on what works best, Quest helps more than 100,000 customers worldwide meet higher expectations for enterprise IT. Quest helps organizations deliver, manage and control complex database environments through award-winning products for Oracle, SQL Server, IBM DB2, Sybase and MySQL. Visit www.quest.com for more information.

Contacting Quest Software

Phone: 949.754.8000 (United States and Canada)
Email: info@quest.com
Mail: Quest Software, Inc.
World Headquarters
5 Polaris Way
Aliso Viejo, CA 92656
USA
Web site: www.quest.com

Please refer to our Web site for regional and international office information.

Contacting Quest Support

Quest Support is available to customers who have a trial version of a Quest product or who have purchased a commercial version and have a valid maintenance contract. Quest Support provides around the clock coverage with SupportLink, our web self-service. Visit SupportLink at http://support.quest.com

From SupportLink, you can do the following:

Quickly find thousands of solutions (Knowledgebase articles/documents).

- Download patches and upgrades.
- Seek help from a Support engineer.
- Log and update your case, and check its status.

View the ***Global Support Guide*** for a detailed explanation of support programs, online services, contact information, and policy and procedures. The guide is available at: http://support.quest.com/pdfs/Global Support Guide.pdf