VMware PRESS

Virtualizing SQL Server with VMware®

Doing IT Right

Michael Corey

Jeff Szastak

Michael Webster

Foreword by **Duncan Epping**

Virtualizing SQL Server with VMware®

Doing IT Right

VMware Press is the official publisher of VMware books and training materials, which provide guidance on the critical topics facing today's technology professionals and students. Enterprises, as well as small- and medium-sized organizations, adopt virtualization as a more agile way of scaling IT to meet business needs. VMware Press provides proven, technically accurate information that will help them meet their goals for customizing, building, and maintaining their virtual environment.

With books, certification and study guides, video training, and learning tools produced by world-class architects and IT experts, VMware Press helps IT professionals master a diverse range of topics on virtualization and cloud computing and is the official source of reference materials for preparing for the VMware Certified Professional Examination.

VMware Press is also pleased to have localization partners that can publish its products into more than forty-two languages, including, but not limited to, Chinese (Simplified), Chinese (Traditional), French, German, Greek, Hindi, Japanese, Korean, Polish, Russian, and Spanish.

For more information about VMware Press, please visit vmwarepress.com.

This page intentionally left blank

Virtualizing SQL Server with VMware®

Doing IT Right

Michael Corey
Jeff Szastak
Michael Webster

vmware® PRESS

Upper Saddle River, NJ • Boston • Indianapolis • San Francisco
New York • Toronto • Montreal • London • Munich • Paris • Madrid
Capetown • Sydney • Tokyo • Singapore • Mexico City

Virtualizing SQL Server with VMware®

Published by Pearson plc

Publishing as VMware Press

ISBN-13: 978-0-321-92775-0

ISBN-10: 0-321-92775-3

Library of Congress Control Number: 2014941961

Printed in the United States of America

First Printing August 2014

Warning and Disclaimer

Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an "as is" basis. The authors, VMware Press, VMware, and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the CD or programs accompanying it.

The opinions expressed in this book belong to the author and are not necessarily those of VMware.

Special Sales

For information about buying this title in bulk quantities, or for special sales opportunities (which may include electronic versions; custom cover designs; and content particular to your business, training goals, marketing focus, or branding interests), please contact our corporate sales department at corpsales@pearsoned.com or (800) 382-3419.

For government sales inquiries, please contact governmentsales@pearsoned.com.

For questions about sales outside the U.S., please contact international@pearsoned.com.

ASSOCIATE PUBLISHER
David Dusthimer

ACQUISITIONS EDITOR
Joan Murray

VMWARE PRESS PROGRAM MANAGER
David Nelson

DEVELOPMENT EDITOR
Eleanor C. Bru

MANAGING EDITOR
Sandra Schroeder

PROJECT EDITOR
Mandie Frank

COPY EDITOR
Bart Reed

PROOFREADER
Sarah Kearns

INDEXER
Erika Millen

EDITORIAL ASSISTANT
Vanessa Evans

DESIGNER
Chuti Prasertsith

COMPOSITOR
Tricia Bronkella

This book is dedicated to my wife of 28 years, Juliann, who has supported me in every way possible, and my three children, John, Annmarie, and Michael.
-Michael Corey

This book is dedicated to my wife, Heather, and my three children, Wyatt, Oliver, and Stella.
-Jeff Szastak

This book is dedicated to my wife, Susanne, and my four sons, Sebastian, Bradley, Benjamin, and Alexander, for their ongoing support. I also dedicate this book to the VMware community.
-Michael Webster

This page intentionally left blank

Contents

This page intentionally left blank

Foreword

About 10 years ago, I started a new job. The company I started working for had a couple hundred physical servers at the time. When several new internal software development projects started, we needed to expand quickly and added dozens of new physical servers. Pretty soon we started hitting all the traditional datacenter problems, such as lack of floor space, high power consumption, and cooling constraints. We had to solve our problems, and during our search for a solution we were introduced to a new product called VMware ESX and Virtual Center. It didn't take long for us to see the potential and to start virtualizing a large portion of our estate.

During this exercise, we started receiving a lot of positive feedback on the performance of the virtualized servers. On top of that, our application owners loved the fact that we could deploy a new virtual machine in hours instead of waiting weeks for new hardware to arrive. I am not even talking about all the side benefits, such as VMotion (or vMotion, as we call it today) and VMware High Availability, which provided a whole new level of availability and enabled us to do maintenance without any downtime for our users.

After the typical honeymoon period, the question arose: What about our database servers? Could this provide the same benefits in terms of agility and availability while maintaining the same performance? After we virtualized the first database server, we quickly realized that just using VMware Converter and moving from physical to virtual was not sufficient, at least not for the databases we planned to virtualize.

To be honest, we did not know much about the database we were virtualizing. We didn't fully understand the CPU and memory requirements, nor did we understand the storage requirements. We knew something about the resource consumption, but how do you make a design that caters to those requirements? Perhaps even more importantly, where do you get the rest of the information needed to ensure success?

Looking back, I wish we'd had guidance in any shape or form that could have helped along our journey—guidance that would provide tips about how to gather requirements, how to design an environment based on these requirements, how to create a performance baseline, and what to look for when hitting performance bottlenecks.

That is why I am pleased Jeff Szastak, Michael Corey, and Michael Webster took the time to document the valuable lessons they have learned in the past few years about virtualizing tier 1 databases and released it through VMware Press in the form of this book you are about to read. Having gone through the exercise myself, and having made all the mistakes mentioned in the book, I think I am well qualified to urge you to soak in all this valuable knowledge to ensure success!

Duncan Epping

Principal Architect, VMware

Yellow-Bricks.com

This page intentionally left blank

Preface

As we traveled the globe presenting on how to virtualize the most demanding business-critical applications, such as SQL Server, Oracle, Microsoft Exchange, and SAP, it became very clear that there was a very real and unmet need from the attendees to learn how to virtualize these most demanding applications correctly.

This further hit home when we presented at the VMworld conferences in San Francisco and Barcelona. At each event, we were assigned a very large room that held over 1,800 people; within 48 hours of attendees being able to reserve a seat in the room, it was filled to capacity. We were then assigned a second large room that again filled up within 24 hours.

Recognizing that the information we had among the three of us could help save countless others grief, we decided to collaborate on this very practical book.

Target Audience

Our goal was to create in one book—a comprehensive resource that a solution architect, system administrator, storage administrator, or database administrator could use to guide them through the necessary steps to successfully virtualize a database. Many of the lessons learned in this book apply to any business-critical application being virtualized from SAP, E-Business Suite, Microsoft Exchange, or Oracle, with the specific focus of this book on Microsoft SQL Server. Although you don't have to be a database administrator to understand the contents of this book, it does help if you are technical and have a basic understanding of vSphere.

Approach Taken

Everything you need to succeed in virtualizing SQL Server can be found within the pages of this book. By design, we created the book to be used in one of two ways. If you are looking for a comprehensive roadmap to virtualize your mission-critical databases, then follow along in the book, chapter by chapter. If you are trying to deal with a particular resource that is constraining the performance of your database, then jump to Chapters 5 through 8.

At a high level, the book is organized as follows:

- Chapters 1 and 2 explain what virtualization is and the business case for it. If you are a database administrator or new to virtualization, you will find these chapters very helpful; they set the stage for why virtualizing your databases is "doing IT right."

- Chapters 3 through 9 are the roadmap you can follow to successfully virtualize the most demanding of mission-critical databases. Each chapter focuses on a particular resource the database utilizes and how to optimize that resource to get the best possible performance for your database when it is virtualized. We purposely organized this section into distinct subject areas so that you can jump directly to a particular chapter of interest when you need to brush up. We expect that you will periodically return to Chapters 5 through 8 as you are fine-tuning the virtualized infrastructure for your mission-critical databases.
- The last two chapters walk you through how to baseline the existing SQL Server database so that you adequately determine the resource load it will put onto the virtualized infrastructure. In these chapters, we also provide detailed instructions on how to configure a stress test.

Here are the three major sections of the book with the associated chapters:

What Virtualization Is and Why You Should Do It

In this section, the reader will learn about the benefits of virtualization and why the world is moving towards 100% virtualization. The reader will learn the benefits of breaking the bond between hardware and software, and the benefits this brings to the datacenter and why virtualization is a better way to do IT.

Chapter 1: Virtualization: The New World Order?

Chapter 2: The Business Case for Virtualizing a Database

Optimizing Resources in a Virtualized Infrastructure

In Chapters 3-9, the reader will gain knowledge on how to properly architect and implement virtualized SQL Server. The reader will start off learning how to put together a SQL Server virtualization initiative, and then dive into an in-depth discussion on how to architect SQL Server on a vSphere platform. This section includes deep dives on storage, memory, networking, and high availability.

Chapter 3: Architecting for Performance: The Right Hypervisor

Chapter 4: Virtualizing SQL Server: Doing IT Right

Chapter 5: Architecting for Performance: Design

Chapter 6: Architecting for Performance: Storage

Chapter 7: Architecting for Performance: Memory

Chapter 8: Architecting for Performance: Network

Chapter 9: Architecting for Availability: Choosing the Right Solution

How to Baseline and Stress Test

The final two chapters walk the reader through the importance of setting up a baseline for their virtualized SQL Server implementation. Chapter 10 speaks to the why and the how of baselining, which is critical to successfully virtualizing SQL Server. In the final chapter, the reader will put all the knowledge presented in the previous chapters together and will be walked through a beginning-to-end configuration of SQL Server 2012 with AlwaysOn Availability Groups running on Windows Server 2012 on a vSphere 5.5 infrastructure.

Chapter 10: How to Baseline Your Physical SQL Server System

Chapter 11: Configuring a Performance Test—From Beginning to End

A database is one of the most resource-intensive applications you will ever virtualize, and it is our sincere intention that with this book as your guide, you now have a roadmap that will help you avoid the common mistakes people make—and more importantly, you will learn how to get optimal performance from your virtualized database.

We want to thank you for buying our book, and we hope after you read it that you feel we have achieved our goal of providing you with a comprehensive resource on how to do IT right. Feel free to reach out to us with any questions, suggestions, or feedback you have.

Michael Corey (@Michael_Corey) Michael.corey@ntirety.com

Jeff Szastak (@Szastak)

Michael Webster (@vcdxnz001)

This page intentionally left blank

About the Authors

Michael Corey (@Michael_Corey) is the President of Ntirety, a division of Hosting. Michael is an experienced entrepreneur and a recognized expert on relational databases, remote database administration, and data warehousing. Microsoft named Michael a SQL Server MVP, VMware named him a vExpert, and Oracle named him an Oracle Ace. Michael has presented at technical and business conferences from Brazil to Australia. Michael is a past president of the Independent Oracle Users Group; he helped found the Professional Association of SQL Server, is a current board member of the IOUG Cloud SIG, and is actively involved in numerous professional associations and industry user groups. Michael currently sits on the executive committee for the Massachusetts Robert H. Goddard Council for Science, Technology, Engineering, and Mathematics.

Jeff Szastak (@Szastak) is currently a Staff Systems Engineer for VMware. Jeff has been with VMware for over six years, holding various roles with VMware during his tenure. These roles have included being a TAM, Systems Engineer Specialist for Business-Critical Applications, Enterprise Healthcare Systems Engineer, and a CTO Ambassador. Jeff is a recognized expert for virtualizing databases and other high I/O applications on the vSphere platform. Jeff is a regular speaker at VMworld, VMware Partner Exchange, VMware User Groups, and has spoken at several SQL PASS events. Jeff holds a Master of Information Assurance degree as well as the distinguished CISSP certification. Jeff has over 13 "lucky" years in IT and is passionate about helping others find a better way to do IT.

Michael Webster (@vcdxnz001) is based in Auckland, New Zealand. He is a VMware Certified Design Expert (VCDX #66), author of longwhiteclouds.com (a top-15 virtualization blog), and a Top 10 VMworld Session Speaker for 2013. In addition, he is a Senior Solutions and Performance Engineer for Nutanix, vExpert, MCSE, and NPP. Michael specializes in solution architecture and performance engineering for Unix-to-VMware migrations as well as virtualizing business-critical applications such as SQL, Oracle, SAP, Exchange, Enterprise Java Systems, and monster VMs in software-defined data centers. Michael has more than 20 years experience in the IT industry and 10 years experience deploying VMware solutions in large-scale environments around the globe. He is regularly a presenter at VMware VMworld, VMware vForums, VMware User Groups, and other industry events. In addition to this book, Michael was technical reviewer of *VCDX Boot Camp* and *Virtualizing and Tuning Large-Scale Java Platforms*, both published by VMware Press.

This page intentionally left blank

About the Technical Reviewer

Mark Achtemichuk (VCDX #50) is currently a Senior Technical Marketing Architect, specializing in Performance, within the SDDC Marketing group at VMware. Certified as VCDX #50, Mark has a strong background in data center infrastructures and cloud architectures, experience implementing enterprise application environments, and a passion for solving problems. He has driven virtualization adoption and project success by methodically bridging business with technology. His current challenge is ensuring that performance is no longer a barrier, perceived or real, to virtualizing an organization's most critical applications on its journey to the software-defined data center.

This page intentionally left blank

Acknowledgments

We would like to thank the entire team at VMware Press for their support throughout this project and for helping us get this project across the line—especially Joan Murray for her constant support and encouragement. We would like to thank our editorial team. Thank you Ellie Bru and Mandie Frank for your attention to detail to make sure we put out a great book, and last but not least, we would especially like to thank our technical reviewer, Mark Achtemichuk (VCDX #50).

Michael Corey

Anyone who has ever written a book knows first hand what a tremendous undertaking it is and how stressful it can be on your family. It is for that reason I thank my wife of 28 years, Juliann. Over those many years, she has been incredible. I want to thank my children, Annmarie, Michael, and especially John, who this particular book was hardest on. John will know why if he reads this.

Jeff and Michael, my co-authors, are two of the smartest technologists I have ever had the opportunity to collaborate with. Thank you for making this book happen despite the many long hours it took you away from your families. Mark Achtemichuk, our technical reviewer, rocks! He helped take this book to a whole new level. To my friends at VMware—Don Sullivan, Kannan Mani, and Sudhir Balasubramanian—thank you for taking all my late-night emails and phone calls to discuss the inner workings of vSphere. To the publishing team at Pearson, what can I say? Thank you Joan Murray for believing and making this book possible.

Special thanks go to my Ntirety family—Jim Haas, Terrie White, and Andy Galbraith are all three incredible SQL Server technologists. And special thanks to people like David Klee and Thomas LaRock and to the entire SQL Server community. Every time I attend a SQLSaturday event, I always think how lucky I am to be party of such a special community of technologist who care a lot and are always willing to help.

Jeff Szastak

I would like to thank my loving wife, Heather, for her love, support, and patience during the writing of this book. I want to thank my children, Wyatt, Oliver, and Stella, for it is from you I draw inspiration. A huge thank-you to Hans Drolshagen for the use of his lab during the writing of this book! And thanks to my mentor, Scott Hill, who pushed me, challenged me, and believed in me. Thanks for giving a guy who couldn't even set a DHCP address a job in IT, Scott.

Finally, I would like to thank the VMware community. Look how far we have come. I remember the first time I saw a VMware presentation as a customer and thought, "If this software works half as well as that presentation says it does, this stuff will change the world." And it has, because of you, the VMware community.

Michael Webster

I'd like to thank my wife, Susanne, and my four boys, Sebastian, Bradley, Benjamin, and Alexander, for providing constant love and support throughout this project and for putting up with all the long hours on weeknights and weekends that it required to complete this project. I would also like to acknowledge my co-authors, Michael and Jeff, for inviting me to write this book with them. I am extremely thankful for this opportunity, and it has been a fantastic collaborative process. Finally, I'd like to thank and acknowledge VMware for providing the constant inspiration for many blog articles and books and for creating a strong and vibrant community. Also, thanks go out to my sounding boards throughout this project: Kasim Hansia, VMware Strategic Architect and SAP expert, Cameron Gardiner, Microsoft Senior Program Manager Azure and SQL, and Josh Odgers (VCDX #90), Nutanix Senior Solutions and Performance Architect. Your ideas and support have added immeasurable value to this book and the IT community as a whole.

We Want to Hear from You!

As the reader of this book, *you* are our most important critic and commentator. We value your opinion and want to know what we're doing right, what we could do better, what areas you'd like to see us publish in, and any other words of wisdom you're willing to pass our way.

We welcome your comments. You can email or write us directly to let us know what you did or didn't like about this book—as well as what we can do to make our books better.

Please note that we cannot help you with technical problems related to the topic of this book.

When you write, please be sure to include this book's title and author as well as your name, email address, and phone number. We will carefully review your comments and share them with the author and editors who worked on the book.

Email: VMwarePress@vmware.com

Mail: VMware Press
ATTN: Reader Feedback
800 East 96th Street
Indianapolis, IN 46240 USA

Reader Services

Visit our website at www.informit.com/title/9780321927750 and register this book for convenient access to any updates, downloads, or errata that might be available for this book.

This page intentionally left blank

Chapter 6

Architecting for Performance: Storage

All aspects of architecting your SQL Server Database for performance are important. Storage is more important than most when compared to the other members of the IT Food Group family we introduced in Chapter 5, “Architecting for Performance: Design,” which consists of Disk, CPU, Memory, and Network. Our experience has shown us, and data from VMware Support validates this belief, that more than 80% of performance problems in database environments, and especially virtualized environments, are directly related to storage. Understanding the storage architecture in a virtualized environment and getting your storage architecture right will have a major impact on your database performance and the success of your SQL Server virtualization project. Bear in mind as you work through your storage architecture and this chapter that virtualization is bound by the laws of physics—it won’t fix bad code or bad database queries. However, if you have bad code and bad queries, we will make them run as fast as possible.

> **TIP**
>
> Greater than 80% of all problems in a virtualized environment are caused by the storage in some way, shape, or form.

This chapter first covers the key aspects of storage architecture relevant to both physical and virtual environments as well as the differences you need to understand when architecting storage, specifically for virtualized SQL Server Databases. Many of the concepts we discuss will be valid for past versions of SQL Server and even the newest release, SQL Server 2014.

We provide guidance on what our experience has taught us are important database storage design principles. We present a top-down approach covering SQL Server Database and Guest OS Design, Virtual Machine Template Design, followed by VMware vSphere Hypervisor Storage Design and then down to the physical storage layers, including using server-side flash acceleration technology to increase performance and provide greater return on investment. We conclude the chapter by covering one of the biggest IT trends and its impact on SQL Server. Throughout this chapter, we give you architecture examples based on real-world projects that you can adapt for your purposes.

When designing your storage architecture for SQL Server, you need to clearly understand the requirements and have quantitative rather than subjective metrics. Our experience has taught us to make decisions based on fact and not gut feeling. You will need to benchmark and baseline your storage performance to clearly understand what is achievable from your design. Benchmarking and baselining performance are critical to your success, so we've dedicated an entire chapter (Chapter 10, "How to Baseline Your Physical SQL Server System") to those topics. In this chapter, we discuss some of the important storage system component performance aspects that will feed into your benchmarking and baselining activities.

The Five Key Principles of Database Storage Design

When architecting storage for SQL Server, it's important to understand a few important principles. These will help guide your design decisions and help you achieve acceptable performance both now and in the future. These principles are important because over the past decade, CPU performance has increased at a much faster pace than storage performance, even while capacity has exploded.

Principle 1: Your database is just an extension of your storage

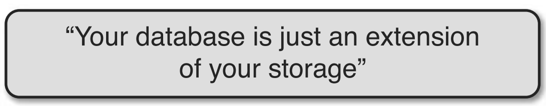


Figure 6.1 Quote from Michael Webster, VMworld 2012

The first principle is highlighted in Figure 6.1: that your database is just an extension of your storage. A database is designed to efficiently and quickly organize, retrieve, and process large quantities of data to and from storage. So increasing the parallelism of access to storage resources at low latency will be an important goal. Later in this chapter, we cover how to optimize the architecture of your database to maximize its storage performance and parallelism. When you understand this principle, it's easy to understand why getting your storage design and performance is so critical to the success of your SQL Server Database virtualization project.

Principle 2: Performance is more than underlying storage devices

The next key principle is that storage performance is more than just about underlying storage devices and spindles, although they are very important too. SQL Server storage performance is multidimensional and is tightly coupled with a number of different system components, such as the number of data files allocated to the database, the number of allocated vCPUs, and the amount of memory allocated to the database. This is why we like to use the term "IT Food Groups," because it is so important to feed your database the right balance of these critical resources. This interplay between resources such as CPU, Memory, and Network and their impact on storage architecture and performance will be covered in subsequent sections of this chapter.

Principle 3: Size for performance before capacity

> "The bitterness of poor performance lasts long after the sweetness of a cheap price is forgotten"

Figure 6.2 Quote from Michael Webster, VMworld 2013

Figure 6.2 is loosely based on the eighteenth-century quote "The bitterness of poor quality remains long after the sweetness of low price is forgotten," by Benjamin Franklin. Both quotes are extremely relevant to SQL Server database and storage performance.

This brings us to the next key principle. In order to prevent poor performance from being a factor in your SQL Server virtualization project (refer to Figure 6.2), you should design storage for performance first (IOPS and latency), then capacity will take care of itself. Capacity is the easy part. We will show you later in this chapter how compromising on certain storage configurations on the surface can actually cost you a lot more by causing unusable capacity due to poor performance.

CAUTION

A lesson from the field: We were working with a customer, and they wanted to design and run a database on vSphere that could support sustained 20,000 IOPS. After we worked with the customer's vSphere, SAN, Network, and DBA teams, the customer decided to move forward with the project. The customer then called in a panic saying, "In our load test, we achieved 1,000 IOPS. We are 19,000 short of where we need to be." Trust me, this is a phone call you don't want to get. Playing the odds, we started with the disk subsystem. We quickly identified some issues. The main issue was the customer purchased for *capacity*, not *performance*. They had to reorder the right disk. Once the new (right) disk arrived and was configured, the customer exceeded the 20,000 IOPS requirement.

TIP

When it comes to storage devices, HDDs are cents per GB but dollars per IOP, whereas SSDs are cents per IOP and dollars per GB. SSDs should be considered cheap memory, rather than expensive disks, especially when it comes to enterprise SSDs and PCIe flash devices.

Principle 4: Virtualize, but without compromise

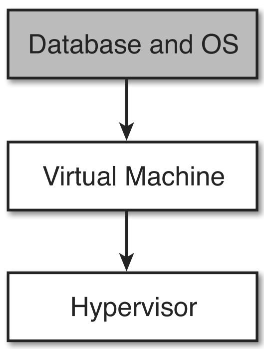
The next principle is that virtualizing business-critical SQL Server databases is all about reducing risk and not compromising on SLAs. Virtualize, but without compromise. There is no need to compromise on predictability of performance, quality of service, availability, manageability, or response times. Your storage architecture plays a big part in ensuring your SQL databases will perform as expected. As we said earlier, your database is just an extension of your storage. We will show you how to optimize your storage design for manageability without compromising its performance.

Believe it or not, as big of advocates as we are about virtualizing SQL Server, we have told customers in meetings that now is not the right time for this database to be virtualized. This has nothing to do with the capability of vSphere or virtualization, but more to do with the ability of the organization to properly operate critical SQL systems and virtualize them successfully, or because they are not able or willing to invest appropriately to make the project a success. If you aren't willing to take a methodical and careful approach to virtualization projects for business-critical applications, in a way that increases the chances of success, then it's not worth doing. Understand, document, and ensure requirements can be met through good design and followed by testing and validation. It is worth doing, and it is worth "Doing It Right!"

Principle 5: Keep it standardized and simple (KISS)

This brings us to the final principle. Having a standardized and simplified design will allow your environment and databases to be more manageable as the numbers scale while maintaining acceptable performance (see Principle 4). If you have a small number of standardized templates that fit the majority of your database requirements and follow a building-block approach, this is very easy to scale and easy for your database administrators to manage. We'll use the KISS principle (Keep It Standardized and Simple) throughout this chapter, even as we dive into the details. Once you've made a design decision, you should standardize on that decision across all your VM templates. Then when you build from those templates, you'll know that the settings will always be applied.

SQL Server Database and Guest OS Storage Design



The starting point for any storage architecture for SQL Server Databases is actually with our last design principle: KISS (Keep It Standardized and Simple). But all of the principles apply. We will determine the smallest number of templates that are required to virtualize the majority (95%) of database systems, and anything that falls outside this will be handled as an exception.

Your first step is to analyze the inventory of the SQL Server Databases that will be virtualized as part of your project (refer to Chapter 4, "Virtualizing SQL Server 2012: Doing It Right"). From this inventory, you will now put each database and server into a group with similar-sized databases that have similar requirements. The storage requirements for all of these existing and new databases, based on their grouping, will be used to define the storage layouts and architecture for each of the SQL Server Databases, Guest OS, and VM template.

> **TIP**
>
> If you are virtualizing existing databases, you might consider using a tool such as VMware Capacity Planner, VMware Application Dependency Planner, Microsoft System Center, or Microsoft Assessment and Planning Toolkit to produce the inventory. VMware Capacity Planner and Application Dependency Planner are available from VMware Professional Services or your preferred VMware partner. When you're baselining a SQL Server database, a lot can happen in a minute. We recommend your sample period for CPU, Memory, and Disk be 15 seconds or less. We recommend you sample T-SQL every minute.

SQL Server Database File Layout

Database file layout provides an important component of database storage performance. If you have existing databases that will be virtualized, you or your DBAs will likely have already developed some practices around the number of database files, the size of database files, and the database file layout on the file system. If you don't have these practices already in place, here we provide you with some guidelines to start with that have proven successful.

Your SQL Server database has three primary types of files you need to consider when architecting your storage to ensure optimal performance: data files, transaction log files, and Temp DB files. Temp DB is a special system database used in certain key operations, and has a big performance impact on your overall system. The file extensions you'll see are .mdf (master data file), .ndf (for secondary data files), and .ldf for transaction log files. We will go over all of these different file types later in this chapter.

Number of Database Files

First, we need to determine the number of database files. There are two main drivers for the number of files you will specify. The first driver is the number of vCPUs allocated to the database, and the second is the total capacity required for the database now and in the future.

Two design principles come into play here: The parallelism of access to storage should be maximized by having multiple database files, and storage performance is more than just the underlying devices. In the case of data files and Temp DB files, they are related to the number of CPU cores allocated to your database. Table 6.1 provides recommendations from Microsoft and the authors in relation to file type.

NOTE

It is extremely unlikely you will ever reach the maximum storage capacity limits of a SQL Server 2012 database system. We will not be covering the maximums here. We recommend you refer to Microsoft (http://technet.microsoft.com/en-us/library/ms143432.aspx).

Table 6.1 Number of Data Files and Temp DB Files Per CPU

File Type	Microsoft Recommended Setting	Author Recommended Setting
Temp DB Data File	1 per CPU core	< 8 vCPU, 1 per vCPU > 8 vCPU, 8 total (increase number of files in increments of four at a time if required) Max 32
Database Data File	0.25 to 1.0 per file group, per CPU core	Min 1 per vCPU, max 32
Database Transaction Log File	1	1*
Temp DB Transaction Log File	1	1*

*If Temp DB and Transaction Log are deployed on local SSD or flash storage, especially when using AlwaysOn Availability Groups, then it is recommended to have an additional copy on SAN.

Microsoft recommends as a best practice that you should configure one Temp DB data file per CPU core and 0.25 to 1 data file (per file group) per CPU core. Based on our experience, our recommendation is slightly different.

If your database is allocated eight or fewer vCPUs as a starting point, we recommend you should configure at least one Temp DB file per vCPU. If your database is allocated more than eight vCPUs, we recommend you start with eight Temp DB files and increase by lots of four in the case of performance bottlenecks or capacity dictates.

TIP

Temp DB is very important because it's extensively utilized by OLTP databases during index reorg operations, sorts, and joins, as well as for OLAP, DSS, and batch operations, which often include large sorts and join activity.

We recommend in all cases you configure at least one data file (per file group) per vCPU. We recommend a maximum of 32 files for Temp DB or per file group for database files because you'll start to see diminishing performance returns with large numbers of database files over and above 16 files. Insufficient number of data files can lead to many writer processes queuing to update GAM pages. This is known as GAM page contention. The Global Allocation Map (GAM) tracks which extents have been allocated in each file. GAM contention would manifest in high PageLatch wait times. For extremely large databases into the many tens of TB, 32 files of each type should be sufficient.

Updates to GAM pages must be serialized to preserve consistency; therefore, the optimal way to scale and avoid GAM page contention is to design sufficient data files and ensure all data files are the same size and have the same amount of data. This ensures that GAM page updates are equally balanced across data files. Generally, 16 data files for tempdb and user databases is sufficient. For Very Large Database (VLDB) scenarios, up to 32 can be considered. See http://blogs.msdn.com/b/sqlserverstorageengine/archive/2009/01/04/what-is-allocation-bottleneck.aspx.

If you expect your database to grow significantly long term, we would recommend that you consider configuring more data files up front. The reason we specify at least one file per CPU is to increase the parallelism of access from CPU to data files, which will reduce any unnecessary data access bottlenecks and lower latency. This also allows for even data growth, which will reduce IO hotspots.

CAUTION

Having too few or too many Temp DB files can impact the overall performance of your database. Our guidance is conservative and aimed to meet the requirements for the majority of SQL systems. If you start to see performance problems such as higher than normal query response times or excessive database waits in *PAGELATCH_XX*, then you have contention in memory and may need to increase the number of Temp DB files further and/or implement trace flag 1118 (which we recommend), which prevents single page allocations. If you see waits in *PAGEIOLATCH_XX*, then the contention is at the IO subsystem level. Refer to http://www.sqlskills.com/blogs/paul/a-sql-server-dba-myth-a-day-1230-Temp DB-should-always-have-one-data-file-per-processor-core/ and Microsoft KB 328551 (http://support.microsoft.com/kb/328551).

TIP

The number of data files and Temp DB files is important enough that Microsoft has two spots in the Top 10 SQL Server Storage best practices highlighting the number of data files per CPU. Refer to http://technet.microsoft.com/en-us/library/cc966534.aspx.

NOTE

When you're determining the number of database files, a vCPU is logically analogous to a CPU core in a native physical deployment. However, in a native physical environment without virtualization, each CPU core may also have a hyper-thread. In a virtual environment, each vCPU is a single thread. There is no virtual equivalent of a hyper-thread.

Figure 6.3 shows an example of data files, Temp DB files, and transaction log files allocated to a SQL Server 2012 Database on a sample system with four vCPU and 32GB RAM.

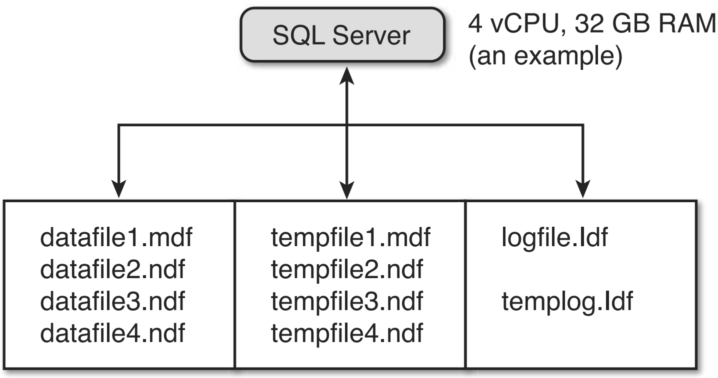


Figure 6.3 SQL Database data file allocation.

NOTE

As Figure 6.3 illustrates, there is only one transaction log file per database and per Temp DB. Log files are written to sequentially, so there is no benefit in having multiples of them, unless you exceed the maximum log file size (2TB) between backups. There is a benefit of having them on very fast and reliable storage, which will be covered later.

This page intentionally left blank

Index

Numbers

A

B

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X-Y-Z