

Capacity planning for a Microsoft Virtual Desktop Infrastructure pooled 2,000-seat virtual machine collection in Windows Server 2012

Microsoft Corporation Published: August 2013

Abstract

The Microsoft Virtual Desktop Infrastructure (VDI) provides each user with a separate virtual machine (VM) and uses a desktop (client-side) operating system for that VM (Windows Server 2012). Microsoft VDI can deliver desktops via three methods: sessions, pooled VMs, or personal VMs. This white paper is a guide for capacity planning of a pooled VM VDI environment running with the VMs hosted across one or more servers running Microsoft Hyper-V.

Powered by Windows Server 2012, VDI allows users to seamlessly access their rich, full-fidelity Windows environment while running in the data center from any device. Organizations employing VDI can realize the following benefits:

Platform. Windows Server 2012 provides a single platform from which to deliver any type of hosted desktop, making it simple to deploy and easy to manage.

Experience. Microsoft RemoteFX provides a consistently rich user experience, regardless of the type of virtual desktop being accessed or from where users are accessing their desktops.

Deployment choices. Although the scope of this document is around pooled VMs, VDI can host either session-based desktops or personal VMs, as well, giving customers the flexibility to deploy the right type of VDI desktop for their users, all from a single platform.

This white paper is a guide for capacity planning a 2,000-seat VDI pooled VM deployment on Windows Server 2012 by using the Dell Desktop Virtualization Services Reference Architecture for Windows Server 2012. It describes the most relevant factors that influence the capacity of a pooled VM deployment evaluated with the Login VSI tool and offers a set of experimental results for the Login VSI medium workload.



Copyright information

This document supports a preliminary release of a software product that may be changed substantially prior to final commercial release, and is the confidential and proprietary information of Microsoft Corporation. It is disclosed pursuant to a non-disclosure agreement between the recipient and Microsoft. This document is provided for informational purposes only and Microsoft makes no warranties, either express or implied, in this document. Information in this document, including URL and other Internet Web site references, is subject to change without notice. The entire risk of the use or the results from the use of this document remains with the user. Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in examples herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2013 Microsoft Corporation. All rights reserved.

Microsoft, Active Directory, Excel, Hyper-V, Internet Explorer, Outlook, PowerPoint, SQL Server, Windows, and Windows Server are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

Contents

Capacity planning for a Microsoft Virtual Desktop Infrastructure pooled 2,000-seat virtual machin	е
collection in Windows Server 2012	4
About this guide	4
Introduction	5
Capacity planning for a specific deployment	6
What determines the capacity of a system?	6
Usage scenario	6
Hardware resources	7
VDI architecture	7
Typical evaluation approaches1	0
Load simulation tests1	1
Scenario definition 1	2
Scenario implementation 1	2
Test execution 1	2
Result evaluation 1	2
Testing methodology1	4
Test deployment overview 1	4
HA implementation 1	5
Pooled VM configuration1	6
Dell R720VDI host servers for the VDI compute and storage nodes specification 1	6
Dell R620 infrastructure hosts specification1	7
Dell 910 Login VSI load launchers specification1	8
Load generation 1	9
Response time measurement 1	9
Test results 2	20
CPU load	21
Network load 2	22
SQL Server load 2	2
Load on the HA RD Connection Broker server 2	23
RD Connection Broker configuration 2	24
Single VM load2	25
Conclusion2	27
Resources 2	29

Capacity planning for a Microsoft Virtual Desktop Infrastructure pooled 2,000-seat virtual machine collection in Windows Server 2012

About this guide

This guide uses a variation of the Dell Desktop Virtualization Services (DVS) Reference Architecture for the Windows Server 2012 operating system and Login VSI 3.7 testing to deliver comparative data for a Microsoft Virtual Desktop Infrastructure (VDI) pooled 2,000-seat virtual machine (VM) collection. Use this guide to help determine your infrastructural needs for deploying VDI.

Introduction

In a server-based computing environment, all application execution and data processing occurs on the server. As a consequence, the server can potentially run out of resources under peak load and cause disruption across the deployment. Therefore, it is valuable to test the scalability and capacity of the server to determine how many client sessions a specific server can support for specific deployment scenarios.

This white paper presents guidelines and a general approach for evaluating the capacity of a system in the context of a specific deployment: a 2,000-seat VDI pooled VM deployment on the Windows Server 2012 operating system similar to the DVS Reference Architecture for Windows Server 2012. The key recommendations are illustrated, with examples based on a scenario using Microsoft Office applications. This paper also provides guidance on the hardware and software parameters that can have a significant impact on the number of sessions a server can support effectively.

Capacity planning for VDI deployments is subject to many variables, and no good off-the-shelf answers are available. Based on usage scenario and hardware configuration, the variance in capacity can reach up to two orders of magnitude. If you need a relatively accurate estimate, either deploying a pilot or running your own load simulation is quite likely the only reliable way to get that. In other words, performance data is sensitive to workload and system configuration: Your mileage will vary!

Capacity planning for a specific deployment

One of the key questions you face when planning a server VDI is the number of users per host, whether that requires figuring out how many users a configuration can run or what configuration you need for *N* users. Determining the system configuration able to support the load generate by users is a typical challenge that any service (such as Microsoft Exchange Server, Internet Information Services [IIS], or SQL Server) faces. This is a difficult question to answer even for server roles that support workloads defined by a relatively small set of transactions and parameters that characterize the profile of a workload. Domain Name System (DNS) is a good example of how DNS queries can define the load well.

VDI requires a separate VM and associated hardware for every user. Although one VM can host a relatively lightweight application that users access infrequently and with low resource costs (like a data-entry application), another may host a demanding computer-aided design (CAD) application requiring a lot of CPU, RAM, disk, and network bandwidth.

In particular, accurate sizing and configuration require clarifying by both the lower and upper bounds:

- The deployment must be sized such that users' applications perform at an acceptable level. If you have 500 users running Microsoft Office 2013 and each needs 20 GB of memory, you would need 10 TB of server disk space to host them all.
- Provision the correct number of resources without significantly exceeding the number required to meet the deployment goals.

Regarding the lower bound, the performance criteria are difficult to state in objective terms because of the large spectrum of applications that can be involved and the variety of ways users can access those applications. A typical complaint that users have about the performance of virtual server applications is that performance is slow or unresponsive, but performance degradation can occur in other ways, such as jittery behavior as opposed to a smooth, even response—sometimes in alternating bursts and lags that may be extremely annoying, even if the average performance may be deemed acceptable. The tolerances to performance degradation vary substantially across deployments: Although some systems are business critical and accept no substantial degradation at any time, others may accept short time spans of peak load where performance is quite poor. Clarity on what the users' expectations are in terms of performance is a key piece of input in the process of sizing the capacity of a deployment.

What determines the capacity of a system?

Before discussing the details of testing a certain scenario on a server, it is important to know what factors affect the server's scalability. At a macro level, these factors fall into two categories: the usage scenario and the hardware resources.

Usage scenario

An extremely important factor in determining the capacity of a given server is the *usage scenario*—the typical sequence of interactions users have with the applications deployed on the

server. A server with a given hardware configuration could support two or 200 users, depending on the scenario. If the scenario is light in resource usage, the server will be able to support many users. An example of such a light scenario is a user entering data in a simple line-of-business application. In contrast, if the scenario is heavy in resource usage, the server will not be able to support as many users. An example of a heavy scenario is a user working with a CAD application or with a complex software development environment that is CPU and input/output (I/O) intensive.

Note When trying to estimate the number of users a server can support, that number only makes sense in the context of a particular scenario. If the scenario changes, the number of supported users will also change.

Generally, the scenario is defined by the system software configuration, applications used, specific features exercised for each application, the amount and content of data being processed, actions performed, and the speed with which actions are being performed. Following are a few examples of significant factors that can influence a simple scenario, such as editing a document:

- Is the user typing in Microsoft Notepad or Word?
- Which version of Word is used?
- Is the spelling checker enabled?
- Does the document contain pictures? Does it contain graphs?
- What is the typing speed?
- What is the session color depth?
- Will the user edit Microsoft PowerPoint decks with heavy animation?

Altering any of these parameters can change the results significantly.

Hardware resources

Server hardware has a major impact on server capacity. The main hardware factors that you must consider are CPU, memory, disk storage, and network. The impact of each of these factors will be addressed in more detail later in this paper.

VDI architecture

VDI offer several ways to address scaling issues. You can configure the VDI environment in three different ways (in order of required server space), as Figure 1 shows:

- Sessions
- Pooled VMs
- Personal VMs



Figure 1. VDI deployment options

With VDI in Windows Server 2012, you can deploy sessions, personal VMs, or pooled VMs, all using the same platform. But how do you choose which architecture is right for you?

These three deployment models share some common benefits:

- Powerful administration capability through the in-box management console, with simple setup, intelligent patching, and unified management using Microsoft System Center technologies
- Consistently rich user experience across LAN and WAN

You can use the following pivot points to determine your architecture:

- **Personalization.** Do your users need the ability to customize their desktops? If so, what level of customization do they need? With sessions and pooled VMs, users have limited personalization capability with user profile disks, like the ability to persist their data across different logins. Although they cannot persist user-installed applications across logins, with a personal desktop (assuming, of course, that the user has administrative rights on their desktop), users can change any aspect of their desktop, including installing their own applications across multiple logins.
- Application compatibility. Session-based desktops share a common server operating system; hence, any applications that installed must be compatible with the Windows 8 operating system. In both pooled VM scenarios, it is the Windows client operating system that is running within the VM, so application compatibility is always higher for VMs than sessions. However, with personal VMs, users can install their own apps, as opposed to pooled VMs, where IT decides which applications are presented to users. Therefore, personal VMs provide the highest level of application compatibility across all three deployment models.
- User density. Because sessions share a single server operating system, the number of users who can be accommodated on a single session-based server is always going to be higher than either VM-based model. In some cases, you can get twice the user density with

sessions than you can with VMs. With pooled VMs, because all VM instances share a common parent disk, each pooled VM typically uses a diff disk in the range of approximately 3–5 GB; also, because user data is either not stored locally or is stored on a separate user profile disk, their sizes are typically smaller than personal VMs. Hence, pooled VMs have significantly higher density from a storage sizing perspective. You can reduce the amount of storage used by introducing user state and application virtualization technologies on the VM, but you will still have lower densities than sessions.

- Images. If getting to a single image is your goal, then the best way to get there is either through session-based desktops or by deploying pooled VMs. For a session-based desktop, all users share a single server image, while in pooled VMs, all users get a cloned copy of a single master image. Single-image configurations are easier to manage and have lower costs compared to personal VMs, where each user gets their own individual image.
- Cost. Because sessions offer the highest densities and are single images, they are often easier to manage and so offer the lowest cost. Pooled VMs get the single image and management benefits of sessions, but higher densities and management efforts mean that they are more expensive to deploy than sessions. Personal VMs have the lowest density and highest management efforts, making them the most expensive of the three deployment models, but bear in mind that Windows Server 2012 helps companies reduce overall VDI total cost of ownership, with support for lower-cost storage (such as Server Message Block [SMB] and direct attached storage), application virtualization, dynamic memory, and user profile disks.

This paper focuses on pooled VMs as the deployment model, which offers the best fit for this user scenario.

Typical evaluation approaches

The above considerations should make it clear that it is not possible to answer the capacity planning questions with reasonable accuracy based on a set of preconfigured numbers. Choosing one of the numbers measured in an actual deployment or simulation and applying it to another deployment that has significant differences in scenario or hardware configuration is not necessarily useful, given the potential errors. Therefore, unless you give careful consideration to the factors affecting the deployment scenario, it is not reasonable to expect high accuracy. Practical approaches exist that can help reduce the estimation error to more reasonable values, and these approaches typically result in different trade-offs between effort invested and accuracy of results. To enumerate a few:

- Piloting. This is probably the most common and the simplest approach. You configure and deploy one test server, and then gradually increase the load over time while monitoring user feedback. Based on that feedback, you adjust the system load up and down until the load stabilizes around the highest level that provides an acceptable user experience. This approach has the advantage of being fairly reliable and simple but requires initial investments in hardware or software that may ultimately turn out to be unsuitable for the deployment goals (for example, the server cannot support enough memory to achieve the desired consolidation). You can further enhance this approach by monitoring various load indicators to determine potential bottlenecks (CPU usage, paging, disk and network queue length, etc.) and overcome them by adding hardware resources (CPUs, RAM, disks, network adapters). However, the lack of control on the level of load makes it difficult to correlate variation in indicators with actual system activity.
- Simulation. In this approach, based on data collected about the specific usage scenario, you can build a simulation by using specific tools to generate various (typically increasing) levels of loads against a test server while monitoring the server's ability to handle user interactions in a timely manner. This approach requires a fairly high initial investment to build the usage scenario simulation and relies significantly on the simulated scenario being a good approximation of the actual usage scenario. However, assuming that the simulation is accurate, it allows you to determine accurately the acceptable levels of load and the limiting factors and offers a good environment for iterating while adjusting various software and hardware configurations.
- Projection based on single-user systems. This approach uses extrapolation based on data collected from a single-user system. In this case, various key metrics like memory usage, disk usage, and network usage are collected from a single-user system, and then used as a reference for projecting expected capacity on a multi-user system. This approach is fairly difficult to implement, because it requires detailed knowledge of system and application operations. Furthermore, it is rather unreliable, because the single-user system data contains a significant level of "noise" generated by interference with the system software. Also, in the absence of sophisticated system modeling, translating the hardware performance metrics (CPU speed, disk speed) to the target server from the reference system used to collect the data is a complex and difficult process.

In general, the first approach proves to be more time and cost-effective for relatively small deployments, while the second approach may be preferable for large deployments where making an accurate determination of server capacity could have a more significant impact on purchasing decisions. This paper uses the second methodology—simulation—using the Login VSI tool.

Load simulation tests

For this guide, we chose to perform load simulation tests based on the DVS Reference Architecture for Windows Server 2012, available from http://www.dell.com/Learn/us/en/555/business~solutions~engineering-docs~en/Documents~dvswindows-server-2012.pdf?c=us&l=en&s=biz and the Login VSI tool, available at http://www.loginvsi.com, to generate load simulation. Our deployment was not identical to the DVS architecture. Specifically, Dell and Microsoft jointly developed a deployment using a scaleout SMB server (versus virtualized SMB in the original architecture) and roaming user profiles (versus user vi rtual hard disks [VHDs]), as shown in Figure 2.

VDI Compute and Storage nodes HA VDI Management infra R720(s) R620 R620 VDI VDI Host -1 VDI Infra srv-1 srv-2 Host Host Pool VM Gateway (s) 14 Same workload as Infra-1 **RDWEB** Pool VM AD SMB Scale-Out Pool VM Storage for User docs & **RD Broker** setting SMB-1 SMB-2 Raid10/equiv SQL srv 15K disks Clustered 15K disks 2x NIC iscsi 2x NIC Clustered IΑ LAN 15K disks 15K disks iSCSI 15K disks 2x DELL S4810 switches OS boot disks 10x 15K disks (Raid1+0) iSCSI Network: 2x10Gig with EQL 6510E VLANs for LAN and iSCSI traffic

2000 seat Pool VM VDI Deployment

Figure 2. The DVS deployment

Load simulation is one of the more accurate techniques for estimating the capacity of a given system. This approach works well in a context in which the user scenarios are clearly understood,

relatively limited in variation, and not terribly complicated. Generally, it involves several distinct phases.

Scenario definition

Having a good definition of the usage scenarios the deployment is targeting is a key prerequisite. Defining the scenarios may turn out to be complicated, either because of the large variety of applications involved or complex usage patterns. Getting a reasonably accurate usage scenario is likely the most costly stage of this approach. It is equally important to capture not only the right sequence of user interactions but also to use the right data content (such as documents, data files, media content), because this also may play a significant role in the overall resource usage on the system. Such a scenario can be built based on interviews with users; by monitoring user activity; and by tracking metrics on key infrastructure servers, project goals, and so on.

Scenario implementation

In this phase, you use an automation tool to implement the scenario so that you can run multiple copies simultaneously against the test system. An ideal automation tool will drive the application user interface from the client, has a negligible footprint on the server, is reliable, and tolerates variation in application behavior because of server congestion. At this stage, it is also important to have a clear idea of the metrics used to gauge how viable the system is at various load levels and to make sure that the scenario automation tools accommodate collecting those metrics. We have chosen the Login VSI Medium Load scenario.

Test execution

Test execution consists of gradually increasing the load against the server while monitoring the performance metrics used to assess system viability. It is also a good idea to collect various performance metrics on the system to help later in identifying the type of resources that come under pressure when system responsiveness degrades. You can repeat this step for various adjustments.

Result evaluation

This is the final step where, based on the performance metrics and other performance data collected during the test, you can make a determination of the acceptable load the system can support while meeting the deployment performance requirements and the type of resources whose shortage causes the performance to start degrading. The conclusions you reach in this step can be a starting point for a new iteration on hardware adjusted to mitigate the critical resource shortage to increase load capacity.

This capacity evaluation approach is what Microsoft recommends when a reasonably accurate number is required, especially for cases like large system deployments, where sizing the hardware accurately has significant implications in terms of cost and a low error margin is desirable. We used the same approach for the experimental data that we used to illustrate various points in this document for the following reasons:

- This approach allowed us to make fairly accurate measurements of the server capacity under specific conditions.
- It makes it possible for independent parties to replicate and confirm the test results.
- It allows a more accurate evaluation of various configuration changes on a reference test bed.

Testing methodology

We included various results obtained in our test labs to illustrate many of the assertions made in this white paper. These tests were executed in the Microsoft laboratories. The tests used the Login VSI set of tools developed specifically for session-based VDI load test simulations so that they meet all the requirements outlined earlier for effective load test execution. These tools were used to implement a few scenarios based on Office 2013 and Internet Explorer. Response times for various actions across the scenarios were used to assess the acceptable level of load under each configuration.

Test deployment overview

We installed Windows Server 2012 Hyper-V hypervisor configured for high availability (HA) and Office 2013. The test tools were deployed on the test controller, client computers, and the test server, as described previously.

Note that the capacity of this deployment is primarily a function of the number of VDI hosts; you could easily add more hosts to grow capacity without upgrading the management infrastructure, but bear in mind that you may need to increase storage for user documents and settings just as you would for traditional desktops with roaming or redirected folders. That said, another benefit of this architecture is that the design of storage for user documents and settings is decoupled from the VDI design, which makes it especially ideal for pooled VMs.

The test deployment was based on but not identical to the DVS Reference Architecture for Windows Server 2012. Specifically, Dell and Microsoft jointly developed a deployment using a scale-out SMB server (versus virtualized SMB in the original architecture) and roaming user profiles (versus user VHDs).

The test bed typically lives on an isolated network and includes three categories of computers (see Figure 3):

- The Hyper-V servers hosting the pooled VMs to be tested
- Infrastructure servers that the scenario requires (such as IIS, SQL Server, and Exchange Server) or that provide basic services (DNS, Dynamic Host Configuration Protocol, Active Directory Domain Services)
- Test clients used to generate the load



Figure 3. Test infrastructure overview

Having an isolated network is an important factor, because it avoids interference of network traffic with either the VDI traffic or application-specific traffic. Such interference may cause random slowdowns that would affect the test metrics and make it difficult to distinguish such slowdowns from those caused by resource exhaustion on the server.

HA implementation

To implement HA for the Management layer, we added a host as well as a few more layers of redundancy. The following elements will protect each critical infrastructure component in the solution:

- The Management hosts will be configured in Hyper-V.
- The storage volume that hosts the Management VMs will be upgraded to a cluster shared volume.
- SQL Server instances will be added to the environment to support RD Connection Broker HA.
 - Optionally, SQL Server mirroring can be configured to further protect SQL Server.
 - The RD Connection Broker will be configured for HA (see Figure 4).



Figure 4. HA configuration

Pooled VM configuration

The pooled VMs had the following configuration:

- Windows 8 x86
- Office 2013
- 1 virtual CPU (vCPU)
- Startup memory: 512 MB
- Dynamic Memory enabled
- Roaming user profile enabled

Dell R720VDI host servers for the VDI compute and storage nodes specification

We configured the servers as follows:

- Operating system name: Windows Server 2012 Datacenter Edition
- Version: 6.2.9200, build 9200
- System Manufacturer: Dell Inc.
- System Model: PowerEdge R720
- System Type: x64-based PC
- Processor: Intel Xeon CPU E5-2690 0 @ 2.90 GHz, 2,900 MHz, 8 cores, 16 logical processors

- Processor: Intel Xeon CPU E5-2690 0 @ 2.90 GHz, 2,900 MHz, 8 cores, 16 logical processors
- BIOS version/date: Dell Inc. 1.6.0, 3/7/2013
- SMBIOS version: 2.7
- Embedded controller version: 255.255
- BIOS mode: Legacy
- Baseboard manufacturer: Dell Inc.
- Platform role: Enterprise Server
- Hardware abstraction layer (HAL): Version = "6.2.9200.16420"
- Installed physical memory (RAM): 256 GB
- Total physical memory: 256 GB
- Page file space: 34.0 GB

Dell R620 infrastructure hosts specification

We configured the servers as follows:

- Operating system name: Windows Server 2012 Datacenter Edition
- Version: 6.2.9200, build 9200
- System manufacture: Dell Inc.
- System model: PowerEdge R620
- System type: x64-based PC
- Processor: Intel Xeon CPU E5-2680 0 @ 2.70 GHz, 2,700 MHz, 8 cores, 16 logical processors
- Processor: Intel Xeon CPU E5-2680 0 @ 2.70 GHz, 2,700 MHz, 8 cores, 16 logical processors
- BIOS version/date: Dell Inc. 1.6.0, 3/7/2013
- SMBIOS version: 2.7
- Embedded controller version: 255.255
- HAL: Version = "6.2.9200.16384"
- Installed physical memory (RAM): 96.0 GB
- Total physical memory: 96.0 GB

• Page file space: 12.5 GB

Dell 910 Login VSI load launchers specification

We configured the servers as follows:

- 2x Dell 910 Hyper-V servers
- 50 launcher VMs per Dell 910 machine
- Each launcher VM launches 20 Remote Desktop Protocol (RDP) clients
- Launcher VM configuration: 8 vCPUs, 8 GB
- Operating system name: Windows Server 2012 Datacenter Edition
- Version: 6.2.9200, build 9200
- System manufacturer: Dell Inc.
- System model: PowerEdge R910
- System type: x64-based PC
- Processor: Intel Xeon CPU L7555 @ 1.87 GHz, 1,862 MHz, 8 cores, 16 logical processors
- Processor: Intel Xeon CPU L7555 @ 1.87 GHz, 1,862 MHz, 8 cores, 16 logical processors
- Processor: Intel Xeon CPU L7555 @ 1.87 GHz, 1,862 MHz, 8 cores, 16 logical processors
- Processor: Intel Xeon CPU L7555 @ 1.87 GHz, 1,862 MHz, 8 cores, 16 logical processor
- BIOS version/date: Dell Inc. 2.8.2, 10/25/2012
- SMBIOS version: 2.6
- Embedded controller version: 255.255
- BIOS mode: Legacy
- Platform role: Enterprise Server
- HAL: Version = "6.2.9200.16420"
- Installed physical memory (RAM): 512 GB
- Total physical memory: 512 GB
- Page file space: 68.0 GB

Load generation

We created roaming user accounts for all users during the testing and configured their profiles. For each user in the Login VSI Medium Workload (v 3.7) scenario, this included copying template files the applications used, setting up a home page in Internet Explorer, and configuring an email account in Microsoft Outlook. We performed an automated restart of the server and client computers before each test run to revert to a clean state for all the components.

The test controller (the Login VSI Manager) initiated 2,000 connections from 100 launchers (each launcher launched 20 RDP clients) in a period of 1 hour, resulting 2,000 logins in 3,600 seconds, or about 33 user logins every minute across the VDI. Each launch goes to the broker to receive network-level authorization, and the broker then passes it on to one of the 2,000 pooled VMs. The 2,000 logins are spread to approximately 140–150 logins for each of the 14 servers, keeping peak CPU usage to 75–80 percent. Note the following points:

- This workload simulates medium knowledge of Office 2013, Internet Explorer, and PDF technology.
- After a session has been started, the medium workload repeats every 12 minutes.
- During each loop, the response time is measured every 2 minutes.
- The medium workload opens up to five apps simultaneously.
- The type rate is 160 ms for each character.
- Approximately 2 minutes of idle time are included to simulate real-world users.

Each loop opens and uses:

- Microsoft Outlook 2013 browsing 10 messages
- Internet Explorer: one instance is left open (BBC.co.uk), one instance each shows Wired.com and Lonelyplanet.com
- Microsoft Word 2013: one instance to measure response time, one instance to review and edit a document
- Bullzip PDF Printer and Acrobat Reader: the Word document is printed to and reviewed as a PDF
- Microsoft Excel 2013: a large, randomized sheet is opened
- Microsoft PowerPoint 2013: a presentation is reviewed and edited
- 7-zip: using the command-line version, the output of the session is compressed

Response time measurement

A user scenario is built by grouping a series of actions. An action sequence starts with the test script sending a keystroke through the client to one of the applications running in the session. As

a result of the keystroke, the application does some drawing. For example, sending CTRL+F to Word results in the application drawing the File menu.

The test methodology is based on measuring the response time of all actions that result in drawing events (except for typing text). The response time is defined as the time taken between the keystroke and the resulting drawing. A timestamp (T1) is taken on the client side when the test tools on the client send a keystroke to the pooled VM client. When the drawing happens in the server application, a test framework tool that runs inside each Remote Desktop session detects it. The test tool on the server side sends a confirmation to the client-side tools, and at this point, the client-side tools take another timestamp (T2). The response time of the action is calculated as T2 - T1. This measurement gives an approximation of the actual response time. It is accurate to within a few milliseconds.

The response time measurement is important, because it is the most reliable and direct measurement of user experience as defined by system responsiveness. Looking at performance metrics such as CPU usage and memory consumption only gives us a rough idea as to whether the system is still within acceptable working conditions. For example, it is difficult to qualify exactly what it means for the users if the CPU is at 90 percent utilization. The response times tell us exactly what the users will experience at any point during the test.

As the number of users increases on a server, the response times for all actions start to degrade after a certain point. This usually happens because the server starts running out of one or more hardware resources. A degradation point is determined for the scenario beyond which the server is considered unresponsive and therefore beyond capacity. To determine the degradation point for the entire scenario, a degradation point is determined for each action based on the following criteria:

- For actions that have an initial response time of less than 200 ms, the degradation point is considered to be where the average response time is more than 200 ms and 110 percent of the initial value.
- For actions that have an initial response time of more than 200 ms, the degradation point is considered to be the point where the average response time increases with 10 percent of the initial value.

These criteria are based on the assumption that a user will not notice degradation in a response time when it is lower than 200 ms.

Generally, when a server reaches CPU saturation, the response time degradation point for most actions is reached at the same number of users. In situations where the server is running out of memory, the actions that result in file I/O degrade faster than others (because of high paging activity, resulting in congestion in the I/O subsystem), such as opening a dialog box to select a file to open or save.

Test results

This capacity evaluation approach is what Microsoft recommends when a reasonably accurate number is required, especially for cases like large system deployments, where sizing the

hardware accurately has significant implications in terms of cost and a low error margin is desirable. We used the same approach for the experimental data that we used to illustrate various points in this document for the following reasons:

- This approach allowed us to make fairly accurate measurements of the server capacity under specific conditions.
- It makes it possible for independent parties to replicate and confirm the test results.
- It allows a more accurate evaluation of various configuration changes on a reference test bed.
- It gives "headroom" in case of an unexpected surge.

CPU load

Although we could have pushed each host to its maximum CPU, we decided to keep them at 80 percent and run 150 VMs per host to test a closer-to-real-world use case, allowing some headroom for unexpected surge.

Figure 5 shows the Login VSI benchmarking results for our 2,000-seat pooled VM deployment, where the Login VSI maximum was not reached.





This diagram shows 2,000 logins in 60 minutes, with the response time in milliseconds where each VM would start running the Login VSI Medium workload within approximately 15 seconds of login. For a detailed discussion on response time as a result of workload configuration, refer to the Login VSI documentation at <u>http://www.loginvsi.com/documentation/v3/analyzing-results/calculating-vsimax</u>. The load on the VDI management infrastructure was low, allowing

such a deployment to grow easily by adding more hosts, because the management infrastructure shows light load based on this 2,000-seat run.

Network load

In the LAN environment that we had set up (2x10 GB), the RDP-generated load on the network was about 400 kbps (average) per pooled VM running the Login VSI Medium workload, so user traffic for this 2,000-seat deployment would be about 800 Mbps—well below the capacity of the network infrastructure we had deployed. Note that because we are using local storage, there is only RDP traffic on the network.

SQL Server load

Figure 6 shows the CPU and I/O load on the SQL Server VM. SQL Server is a key part of the HA RD Connection Broker model in Windows Server 2012, where the HA RD Connection Broker servers use SQL Server to store deployment settings. Customers can choose among several HA models for SQL Server to build enterprise-to-enterprise HA brokering.



Figure 6. SQL Server load during 2,000 connections

We configured the SQL Server instances as follows:

- 4 vCPUs, 8,192 GB (approximately 6 GB free, only 2 GB used)
- 2,000 connections in 1 hour

• SQL Server VM running on a Dell R720 machine

Note that the CPU load (the red line in Figure 6) is shown at 10x its actual value, so the load was approximately 3–4 percent I/O, not 30 percent. This means that the system is barely loaded, using only 4 percent of the CPU during the 1-hour login period. With such a low usage rate, you may be able to use your existing SQL Server deployment. This means that you can easily host more VMs or handle faster logins.

Note that the CPU load scales with time. If you want to scale the number of users/time to the same amount in half the time, the load would increase from approximately 3–4 percent to approximately 6–8 percent. If 2,000 users were to log in in 15 minutes, then load on the SQL Server instance would be quadrupled to about 20 percent of CPU. If you wanted the same number of users in 5 minutes, it would run at approximately 80 percent.

Load on the HA RD Connection Broker server

The broker was configured as follows (see Figure 7):

- 2 vCPUs, 8,192 GB (approximately 6 GB free, only 2 GB used)
- Broker VMs running on Dell R720 machines





As you can see, the broker had plenty of unused capacity and easily handled the 2,000 logins. We probably did not need the second broker. The spike show every user login and on different discs. VMs are in an off state until you log in, and then about 500 M/s of data are loaded into Hyper-V memory. You can reduce this spike by starting the VMs before users log in.

RD Connection Broker configuration

Figure 8 shows the CPU and storage load on one of the 14 VDI hosts.



Figure 8. CPU and disk I/O load on a single Dell 720 machine during the 2,000-seat deployment benchmark with 150 Windows 8 x86 VMs and Office 2013 running a Logon VSI Medium workload

The CPU consumption is about 80 percent, with 150 VMs running the Logon VSI Medium workload. During the stress period, the I/O load was 1,000 reads/second and disc writes were at 800/second. The per-host local storage consists of 10x 15,000 disks configured as redundant array of independent disks (RAID) 1+0, easily handling the necessary I/O per second (IOPS). Note also that disk response time remains low, which tells us that the storage should be able to handle more load, faster logins, and so on.

It is difficult to say exactly when, but a good rule of thumb is to estimate I/O capacity of the 10disk array at about 2,000 read-IOPS and 1,000 write-IOPS, and because the I/O load from 150 VMs over an hour-long login period is about 1,743, the same workload at about 30 minutes will be about 3,500 IOPS, exceeding the I/O capability of the local array.

For a tighter login period (more logins per timeframe), where the I/O load during a shorter login cycle can exceed the I/O capacity of the local spindle disks, Microsoft's recommendation is to replace one or two of the spindles with a small solid-state disk (SSD) for the gold VM, a 250 GB SSD. For deployments with no more than 10 collections, a 250 GB SSD for the virtual desktop template should easily provide the additional performance necessary for faster logins under a heavier workload.

Single VM load

In Figure 9, you see that a guest VM is running/idling initially at about 800 MB and CPU is also flat. Then, at about 11:25 (marked by the vertical green line), a user login finishes, and moments later the Logon VSI workload starts running. When benchmarking starts, you see that CPU usage picks up, as does the memory usage, where Hyper-V Dynamic Memory provides additional RAM. Finally, the guest RAM settles to about 1 GB. This pattern repeats for all VMs on a VDI host, where the 150 VMs consume about 150 GB of memory at a cumulative CPU usage of about 75 percent of CPU. We have plenty of headroom for CPU spikes and real-world workload that could demand more memory as each server is configured with 256 GB of RAM (but remember that some portion of that 256 GB is reserved for the services running on the parent partition).



Figure 9. Workload on a single VM

Figure 10 provides an example of the guest physical memory from one of the 14 VDI hosts, where we had about 150 VMs running (each colored line is the memory usage of a particular VM). We configured Dynamic Memory to allow up to 2 GB, with a startup value of 512 MB. All VMs are logged on and running the Logon VSI workload by the end of the first hour. As you can see, all VMs start running with about 500 MB of memory and settle to about 1 GB while running the Logon VSI Medium workload.



Figure 10. Memory consumption on a single Dell R720 machine: 150 Windows 8 x86 computers with Office 2013 running the Logon VSI Medium workload

Conclusion

These results represent a medium workload scenario chosen as a representative scenario for discussion purposes. Your configuration will be unique. If you need accurate estimates, Microsoft recommends running your own load simulation.

For a Logon VSI 3.7 Medium workload on a dual-socket E5-2690 @ 2.90 GHz server, see the summary in Table 1.

Table 1. Medium Workload per Server

VM density	10 users/core @ ~80% CPU, with one vCPU per user
Memory	1 GB of RAM for Windows 8 x86 with Office 2013
IOPS	10 IOPS/VM
RDP network load on LAN	~400 Kbit/second (average over Logon VSI medium workload)

You can use the estimation in Table 1 to plan the VDI CPU, RAM, and storage requirements for various size deployments, as shown in Table 2.

 Table 2. Storage Requirements for Various User Counts

User count	CPU sockets	RAM	Storage size	Storage load	LAN traffic
150	2	192 GB	1 TB	1,500 IOPS	60 Mbps
600	8	768 GB	3 TB	6,000 IOPS	240 Mbps
1,200	16	1.5 TB	5 TB	12,000 IOPS	480 Mbps
2,100	28	3 TB	10 TB	21,000 IOPS	1 Gbps

Using a server with a dual-socket E5-2690 @ 2.90 GHz, 192 GB of RAM, and 10x 15,000 RAID 1+0 configuration, we can tabulate the required number of servers (see Table 3).

Table 3. Server Number Requirements for Various User Counts

User count	VDI server count
150	1
600	4
1,200	8
2,100	14

We also need two servers to run the management workload (RD Broker, SQL Server, etc.) plus a relatively small amount of shared storage for these VMs as well as HA storage for user settings, profile, docs, etc. VDI can provide good consolidation for certain scenarios if care is taken when configuring the hardware and software. The modified Dell architecture used in our test

deployment readily handled our 2,000-user Logon VSI Medium workload scenario using 14 Dell R720 machines to host the pooled VMs and two Dell R620 machines for the HA VDI management infrastructure.

Resources

"How to Setup Mirroring in SQL Server" at <u>http://www.sqlserver-training.com/how-to-setup-</u> mirroring-in-sql-server-screen-shots/-

Additional resources on Microsoft TechNet:

- "How to: Prepare a Mirror Database for Mirroring (Transact-SQL)" at <u>http://technet.microsoft.com/en-us/library/ms189047.aspx</u>
- "Establish a Database Mirroring Session Using Windows Authentication (SQL Server Management Studio)" at <u>http://technet.microsoft.com/en-us/library/ms188712.aspx</u>

The blog entry, "Hot off the presses, get it now, the Windows 8 VDI optimization script, courtesy of PFE!" at <u>http://blogs.technet.com/b/jeff_stokes/archive/2013/04/09/hot-off-the-presses-get-it-now-the-windows-8-vdi-optimization-script-courtesy-of-pfe.aspx</u>