

Remote Desktop Virtualization Host Capacity Planning in Windows Server 2008 R2

Microsoft Corporation

Published: June 2010

Abstract

The Remote Desktop Virtualization Host (RD Virtualization Host) role service lets multiple concurrent users run Windows®-based applications on a remote virtual machine running Windows client operating systems. This white paper is intended as a guide for capacity planning of RD Virtualization Host in Windows Server 2008 R2. It describes the most relevant factors that influence the capacity of a given deployment, methodologies to evaluate capacity for specific deployments, and a set of experimental results for different combinations of usage scenarios and hardware configurations.



Copyright Information

*The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.*

*This White Paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.*

*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.*

* 2010 Microsoft Corporation. All rights reserved.*

*Microsoft, Hyper-V, Windows, and Windows Server are trademarks of the Microsoft group of companies.*

*All other trademarks are property of their respective owners.*

Contents

[Introduction 4](#_Toc264889900)

[Capacity planning goals and approaches 5](#_Toc264889901)

[Testing methodology 6](#_Toc264889902)

[Test bed configuration 6](#_Toc264889903)

[Load generation 8](#_Toc264889904)

[Response time measurement 9](#_Toc264889905)

[Scenarios 11](#_Toc264889906)

[Examples of test results for different scenarios 13](#_Toc264889907)

[Tuning Your Server to Maximize Capacity 14](#_Toc264889908)

[Memory 15](#_Toc264889909)

[Storage 15](#_Toc264889910)

[CPU 17](#_Toc264889911)

[Conclusions 19](#_Toc264889912)

[Appendix A: Test Hardware Details 21](#_Toc264889913)

[Appendix B: Testing Tools 22](#_Toc264889914)

[Test control infrastructure 22](#_Toc264889915)

[Scenario execution tools 22](#_Toc264889916)

[Appendix C: Test Scenario Definitions and Workflow 24](#_Toc264889917)

[Knowledge Worker v2.1 24](#_Toc264889918)

[Appendix D: Remote Desktop Virtualization Host Settings 27](#_Toc264889919)

[Appendix E: Connection Broker Settings 28](#_Toc264889920)

[Appendix F: Guest Virtual Machine Settings 29](#_Toc264889921)

# Introduction

Remote Desktop Virtualization Host (RD Virtualization Host) is a new Remote Desktop Services role service which supports Virtual Desktop Infrastructure (VDI) scenarios and lets multiple concurrent users run Windows®-based applications in virtual machines hosted on a Windows Server® 2008 R2 server running Hyper-V. This white paper is intended as a guide for capacity planning of an RD Virtualization Host server running Windows Server 2008 R2.

In a Remote Desktop Virtualization Host-based computing environment, all application execution and data processing occurs on the server on a virtual machine. Given that many users can be hosted simultaneously, the server is one of the most likely systems to run out of resources under peak load and cause disruption across the deployment. Therefore it is very valuable to test the scalability and capacity of the server to determine how many client virtual machines a specific server can support in specific deployment scenarios.

This document presents some preliminary guidance and data around capacity planning for RD Virtualization Host and should be regarded as an update to the “[Remote Desktop Session Host Capacity Planning in Windows Server R2](http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=ca837962-4128-4680-b1c0-ad0985939063)” document. As such, it focuses mostly on the RD Virtualization Host-specific aspects of the capacity planning exercise, and briefly summarizes most of the facts that are equally applicable to both types (virtual and session) of Remote Desktop Services deployments. For a more complete understanding of all the considerations and guidelines, it is highly recommended that you read the RD Session Host white paper. The results presented in this document are based on a few scenarios that use Microsoft® Office applications. The document also provides basic guidance on the hardware and software parameters that can have a significant impact on the number of virtual machines that a server can support effectively.

Dynamic Memory enables customers to more efficiently utilize the memory resources of Hyper-V hosts by balancing how memory is distributed between running virtual machines. Memory can be dynamically reallocated between different virtual machines in response to the changing workloads of these machines. It lets you increase virtual machine density with the resources you already have without sacrificing performance or scalability resulting in more efficient use of expensive server hardware resources, which can translate into easier management and lower costs. For more information about Dynamic Memory, see [Hyper-V Dynamic Memory Configuration Guide](http://go.microsoft.com/fwlink/?LinkId=214695).

# Capacity planning goals and approaches

The key question capacity planning efforts try to answer is: “How many users will this server be able to host?” Other variations of the question could also be: “How much hardware is required to properly host all my users?” or “What kind of server is required to host <N> users?”.

Answering this question for RD Virtualization Host servers is relatively difficult because the load is defined by the deployed applications, the hardware configuration, the system software configuration, and the user interaction, which can differ substantially from deployment to deployment. While one deployment may host a relatively lightweight application that users access infrequently with low resource costs (like a data entry application), another may host a very demanding CAD application requiring a lot of processor, memory, disk, and/or network bandwidth.

Furthermore, the key question implies two additional requirements:

1. The deployment needs to be sized such that users’ applications perform at an acceptable performance level.
2. The number of resources that servers are provisioned with does not significantly exceed the number required for meeting the deployment goals.

The performance criterion is difficult to state in objective terms because of the large spectrum of applications that may be involved and the variety of ways that users can interact with those applications. One of the most typical complaints that users have about the performance of their Remote Desktop sessions applications is that performance is slow or unresponsive, but there are other ways in which performance degradation may occur, such as jittery behavior as opposed to a smooth, even response, sometimes 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: while 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.

The most significant factors that determine the capacity of a given deployment are:

1. **Usage scenario** - including the nature and implementation of applications, their configuration, usage pattern, data content being processed etc.
2. **Hardware resources** - including available memory, processor speed number and speed, disk speed etc.

In practice there are multiple ways in which a capacity planning exercise can be conducted (piloting, simulation, extrapolations etc.) that make different trade-offs regarding completion time, accuracy, complexity. The results presented in this document are based on a **load simulation** approach because it best fits the requirements for empirical testing:

* It 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 our test results.
* It allows a more accurate evaluation of various configuration changes on a reference test bed.

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

## Testing methodology

The RD Virtualization Host tests were executed in the Microsoft laboratories. The tests used a set of tools developed specifically for the purpose of Remote Desktop load test simulations that meet the requirements outlined for effective load test execution. These tools were used to implement a few scenarios based on Microsoft Office 2007 and Internet Explorer. The acceptable level of load under each configuration was assessed using response times for various actions across the scenarios.

### Test bed configuration

The Remote Desktop test laboratory configuration is shown in Figure 1.

****

**Figure 1 – Test setup configuration**

All the computers used for the test were connected on a private/isolated network.

The operating system installed on the test server was Windows Server 2008 R2 Enterprise Edition on which Hyper-V and Remote Desktop Virtualization Host are also installed. A virtual network switch was created on the server running Hyper-V and connected to the virtualization server’s network adapter that is connected to the test network. A set of virtual machines configured to use one virtual processor, the test case defined amount of memory, and a network adapter connected to the virtual switch was created for each test case. The virtual machines used a reference image configured according to test requirements (operating system, Office 2007 installation, etc.) as described in Appendix D.

The operating system installed on the Remote Desktop Connection Broker (RD Connection Broker) server was Windows Server Standard 2008 R2 with the RD Connection Broker and RD Session Host (running in redirection mode) role services installed. The RD Connection Broker was configured with a virtual desktop pool that included all test virtual machines assigned to exactly one test user in the domain.

For details on the configuration of the other infrastructure servers, see Appendix A.

A roaming profile hosted on a file share on the Microsoft Exchange server was configured for each user in the Knowledge Worker scenario, including copying template files used by the applications, setting up a home page on Internet Explorer, and configuring an e-mail account in Microsoft Outlook. Before taking a snapshot of each virtual machine, a user’s profile was configured for each user, and the associated virtual machine was allowed to stabilize for a few hours and was then changed to a Saved state so that all virtual machines could be easily reverted to a clean Saved state snapshot before each test run.

The server and client workstations were restarted automatically before each test run to revert to a clean state for all the components. All virtual machines were reverted to the state captured in the associated snapshot.

The test tools were deployed similarly to the RD Session Host methodology (see Appendix B, Scenario Execution Tools section, for the exact layout), with one significant difference: instead of being installed only on the test server, the TSAccSessionAgent.exe is installed on each virtual machine because RDP connections are directed to individual virtual machines as opposed to the test server, and the machine that is the Remote Desktop target (the virtual machines in this case) needs to have that component installed to support the scenario execution.

### Load generation

The test controller was used to launch automated scenario scripts on the workstations. Each script, when launched, starts a remote desktop connection as a test user to the target VDI virtual machine and then runs the scenario.

A user is connected to a virtual machine in the following way:

1. All connections are initially directed to the RD Session Host server in redirection mode installed on the server hosting RD Connection Broker.
2. The RD Session Host server running in redirection mode authenticates the user, and then forwards the request to the RD Connection Broker server.
3. The RD Connection Broker server queries Active Directory Domain Services and retrieves the name of the virtual machine that is assigned to the requesting user account.
4. The RD Connection Broker server selects the virtual machine associated with the user, and sends a request to the RD Virtualization Host server to start the virtual machine.
5. The RD Virtualization Host server returns the IP address of the fully qualified domain name, of the virtual machine, to the RD Connection Broker server. The RD Connection Broker server then sends this information to the RD Session Host server running in redirection mode.
6. The RD Session Host server running in redirection mode redirects the request to the client computer that initiated the connection.
7. The client computer connects to the virtual machine.

The test tool requires two specific configuration settings to successfully connect to the virtual machine through the RD Session Host server running in redirection mode /RD Connection Broker server:

1. Allow authentication based on the server name provided by the RD Session Host server running in redirection mode. This requires changing the following property of the Remote UI object:

RUIDCOM.SetProperty "UseRedirectionServerName", "1"

1. Configure RD Connection Broker to use the default virtual machine pool

The Remote Desktop users were started by the test controller in groups of 10 with 1 minute between successive users. After the group of 10 users was started, a 5-minute stabilization period was observed in which no additional sessions were started before moving on to the next group. This means that it takes 9 minutes for the test to start 10 users. The controller takes 2 hours and 20 minutes to start 100 users; taking into account the 5-minute stabilization period, 10 users log on in average every 14 minutes.

This approach of logging on users one at a time has two advantages. First, it ensures that we do not overwhelm the server by logging on 100 users at the same time. Second, we can look at the resulting data from the test, and point to a specific number of users after which the responsiveness of the virtual machines running on the server degrades significantly. From the results in the following sections it can be seen that the number of supported users has been reported to the nearest 10. The reason for this is that we use a group size of 10 users and the level of precision that we get from the test data is not sufficient to clearly distinguish between users from the same group.

### Response time measurement

Coming up with a *single* application-independent criterion for defining when an application performance degrades is fairly difficult. However, there is an interaction sequence that captures the most fundamental transaction of an interactive application: sending input, such as from a keyboard or mouse, to the application and having the application draw something back in response. The most trivial case of this would be typing, but other interactions like clicking a button, or selecting a check box or menu item also map in a very straightforward way to this type of transaction. The reason this interaction pattern stands out is that it captures the fundamental intention of connecting to a remote end point: allowing a user to interact with a rich user interface running on a remote system the same way he or she would if the application were running locally. Although this metric will not cover all relevant metrics for tracking application performance, it is a very good approximation for many scenarios, and degradation measured through this metric correlates well in general with degradation from other metrics.

A user scenario is built by grouping a series of actions. An action sequence starts with the test script sending a key stroke through the client to one of the applications running in the session. As a result of the key stroke, the application does some drawing. For example, sending ALT-F to Microsoft 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 key stroke and the drawing that happens as a result. A timestamp (T1) is taken on the client side when the test tools on the client send a keystroke to the Remote Desktop Connection (RDC) client. When the drawing happens in the server application, it is detected by a test framework tool that runs inside each Remote Desktop session. 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 (ms).

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 processor 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 processor is at 90% utilization. The response times tell us exactly what the users will experience at any point during the test.

In the scenario where there is one virtual machine for each one user, 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 RDP sessions hosted by the virtual machines, running on the server, are 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% 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% of the initial value.

These criteria are based on the assumption that a user experience is not significantly impacted when a response time is lower than 200 ms.

Generally, when a server reaches processor saturation, the response time degradation point for most actions is reached at the same number of users. In situations where the server’s I/O subsystem nears saturation, the actions that result in file input/output degrade faster than others, such as opening a dialog box to select a file to open or save. For the purpose of this testing, the degradation point for the whole test was determined to be the point where at least 20% of the user actions have degraded. A typical user action response time chart is shown in Figure 2. According to the criteria described above, the degradation point for this action is at 150 users.

**Figure 2 – Response time evaluation**

### Scenarios

The scenarios used for testing are automated and meant to simulate real user behavior. Although the scripts used in these scenarios simulate tasks that a normal user could perform, the users simulated in these tests are tireless—they never reduce their intensity level. The simulated clients type at a normal rate, pause as if looking at dialog boxes, and scroll through mail messages as if to read them, but they do not get up from their desks to get a cup of coffee, they never stop working as if interrupted by a phone call, and they do not break for lunch. The tests assume a rather robotic quality, with users using the same functions and data sets during a thirty-minute period of activity. This approach yields accurate but *conservative* results.

Initial testing using the v2 version of the Knowledge Worker scenarios against the RD Virtualization Host server exhibited some instability, with glitches in user interface (UI) automation scripts and connection failures. These UI automation failures were concentrated around a relatively small set of script actions, most of them dealing with disk I/O related application activities (Opening the “open file” dialog and “save file” dialogs in PowerPoint, for example). This is reasonably explained by the substantially higher level of disk I/O activity generated by the RD Virtualization Host server when compared to its RD Session Host server counterpart. Given the random nature of these failures, it is unlikely that increasing storage performance would be able to help mitigate the issues (although it would likely make them less common), so we decided to alter some of the timings/synchronization conditions in the script to increase the UI automation reliability because this can be done without significantly affecting either the average workload generated by user or the real-life simulation characteristics of the test. The measured difference in cycle time between the KW v2 and KW v2.1 scenarios is less than 1.5% which makes it reasonable to expect a variation in the same order of magnitude between the results the two versions would generate. The new scripts that include these changes were assigned a new version and will be designated in this document and any subsequent documentation as “Knowledge Worker v2.1” and “Knowledge Worker without PowerPoint v2.1”.

For the tests using Windows XP as the guest operating system, the scripts were changed in a few places to use different accessibility events for synchronization. These changes do not affect the load generated by the scenario in any significant way.

#### Knowledge Worker v2.1

The Knowledge Worker scenario consists of a series of interactions with Microsoft Office 2007 applications (Word, Excel, Outlook, and PowerPoint) and Internet Explorer. The set of actions and their frequency in Office segments of the scenario are based on statistics collected from the Software Quality Management data submitted by Office users and should represent a good approximation of an “average” Office user. The scenario includes the following:

* Creating and saving Word documents
* Printing spreadsheets in Excel
* Using e-mail communication in Outlook
* Adding slides to PowerPoint presentations and running slide shows
* Browsing Web pages in Internet Explorer

This scenario is described in detail in Appendix C: Test Scenario Definitions and Workflow.

#### Knowledge Worker v2.1 without PowerPoint

This scenario is similar to the Knowledge Worker scenario in most ways. The significant difference in this case is that the light Knowledge Worker scenario does not use PowerPoint. The duration of the scenario is the same as the Knowledge Worker scenario, but instead of spending time using PowerPoint, the user spends more time typing Word documents, filling Excel spreadsheets, and typing e-mail messages. This scenario is significantly lighter in terms of CPU usage compared to the Knowledge Worker scenario because PowerPoint, while taking only approximately 10% of the total work cycle duration, uses more than half of the processor. This also generates significant variation in the processor usage during the work cycle, with much higher levels of processor usage during the short PowerPoint interaction sequence. There were two reasons to introduce this scenario: PowerPoint usage data shows that it is not as widely used as the other Office applications in the mix and this scenario gives an alternate angle on examining various factors due to its relatively lighter load and smoother variations in resource usage.

## Examples of test results for different scenarios

|  |  |  |  |
| --- | --- | --- | --- |
| Server Configuration | Guest Configuration | Scenario | Max Users |
| CPU configuration | **Cores[[1]](#footnote-1)** | **RAM** | **OS Version** | **RAM** | **Hyper-V/RDVH support limit[[2]](#footnote-2)** | **Performance support limit[[3]](#footnote-3)** |
| 4 x AMD Opteron 8378 Quad-core 2.4 GHz | 16 | 128 | Windows 7 | 1024 | KW v2.1  | 128 | 120 |
| 4 x AMD Opteron 8378 Quad- core 2.4 GHz | 16 | 128 | Windows 7 | 1024 | KW v2.1 w/o PPT | 128 | 120 |
| 4 x AMD Opteron 8378 Quad-core 2.4 GHz | 16 | 128 | Windows XP | 512 | KW v2.1  | 128 | 170 |
| 4 x AMD Opteron 8378 Quad-core 2.4 GHz | 16 | 128 | Windows XP | 512 | KW v2.1 w/o PPT | 128 | 220 |
| 2 x Intel Xeon E5530 Quad-core(non-hyper threaded) 2.4 GHz | 8 | 72 | Windows 7 | 1024 | KW v2.1  | 64 | 65 |
| 2 x Intel Xeon E5530 Quad-core(non-hyper threaded) 2.4 GHz | 8 | 72 | Windows 7 | 1024 | KW v2.1 w/o PPT | 64 | 65 |
| 2 x Intel Xeon E5530 Quad-core(non-hyper threaded) 2.4 GHz | 8 | 72 | Windows XP | 512 | KW v2.1  | 64 | 130 |
| 2 x Intel Xeon E5530 Quad-core(non-hyper threaded) 2.4 GHz | 8 | 72 | Windows XP | 512 | KW v2.1 w/o PPT | 64 | 130 |

**Table 1 - Server capacity by scenario**

Table 1 shows the comparison of server capacity for the two versions of Knowledge Worker scenario (with and without PowerPoint) on 2 differently configured test servers. The capacity numbers are determined by using the criteria outlined above, but these numbers should be treated with caution and should be adjusted for the real deployments. This is the case specifically for processor-limited configurations, where it is advisable to allow for significant spare capacity on the processor (at least 20%) to mitigate temporary peaks and congestion points in the workload.

Both servers had all available DIMM slots fully populated with 4GB DIMMs. The tests used 2 different virtual machine configurations:

* Configuration 1 - 1 Virtual Processor, 1GB of memory, and the Windows 7 operating system
* Configuration 2 - 1 virtual processor, 512 MB of memory, and the Windows XP operating system

As explained further in the document, the memory configuration for Windows 7 was decided based on minimum supported configuration numbers, not performance considerations. The storage for virtual machines was provided by a SAN-based RAID 0 volume with 15, 15K RPM FC disks of 146 GB each.

In all but 2 cases (the Windows 7 guest virtual machines running on the 16 core system), the test limit was determined by the Hyper-V support limits for the Virtual Processor to Logical Processor ratio (8:1). It is worth noting that from a performance point of view the system was able to achieve significantly higher ratios (as high as 13.75:1), which should give confidence that deploying Hyper-V can efficiently handle the 8:1 supported ratio. But a significantly more important factor is the storage throughput. Supporting large number of users on a server requires significant throughput to accommodate the high rate of I/O operations, especially during virtual machine start up. Additionally it requires significant storage space for holding all the virtual machines involved (in this case approximately 10 GB per virtual machine).

# Tuning Your Server to Maximize Capacity

This section of the document will take a closer look at a few hardware configuration aspects that have a significant impact on server capacity. The numbers below are specific to the hardware and scenarios used in our tests and will likely differ for other scenarios/hardware configurations, but they should still be able to give a good sense of the order of magnitude and direction in which such a configuration change could impact a Remote Desktop Services deployment.

There are a few general considerations as to what would be a suitable server for a Remote Desktop Virtualization Host server deployment that would give a reasonable approximation for a good server without taking the scenario in consideration:

* 2-4 processor sockets supporting 8 to 24 cores (32 in the near future when 8 core processors will be available)
* At least 8 to 9 memory DIMM slots per socket which can be populated with RAM by using cost effective 4 GB modules (should result in 32+ GB per socket and 4+GB per core)

The three major hardware configuration factors individually analyzed below are processor, memory and disk throughput.

### Memory

In our test results, although 6 out of the 8 tests stopped due to exhaustion of available memory, they hit that limit at a point where the test already crossed the boundary for the supported Virtual Processor to Logical Processor ratio. The high memory consumption is expected given the significant amount of memory that needs to be allocated to each virtual machine that further gets multiplied by the relatively large number of users expected to be supported on a server. The server memory usage is driven mostly by three main factors:

* Operating system overhead
* Hyper-V service overhead per virtual machine
* Memory allocated to each virtual machine

The virtual machine allocated memory is a fixed quantity specified in the virtual machine configuration and is normally determined as the maximum between the minimum amount of memory required by the operating system running inside the virtual machine and the minimum amount of memory required to execute the user scenario without significant performance degradations. The second number is harder to determine since it depends on the applications involved in the scenario and needs to evaluate the working set size required to keep paging activity to levels that can be effectively serviced by the storage. In the test configuration using Windows 7 as the guest operating system, the Knowledge Worker scenario has a reference set of less than 400 MB, which means that the minimum Windows 7 supported configuration of 1 GB of memory provides ample space to accommodate the Knowledge Worker working set. For Windows XP, since the memory consumption is lower compared to Windows 7 we decided to use 512 MB of memory that accommodates the entire scenario working set without causing any paging. Windows XP would likely work well with even lower amounts of memory, but in the test configurations used for this whitepaper, the difference is not relevant since the test systems have enough memory to saturate the processor before running out of memory.

The tests that used Windows 7 as the guest operating system configured with 1 GB of memory per virtual machine stopped because no more virtual machines could be started due to lack of available memory. The amount of overhead observed in our testing by the operating system and Hyper-V services, although non-negligible, is not too large either: ~6% (or 8 GB) for the 128 GB system and ~10% (7 GB) for the 72GB system.

Memory pressure is solvable in two ways:

* As larger memory DIMMs are released, at an affordable price point, customers can replace the existing ones and gain headroom.
* As Service Pack 1 for Windows Server 2008 R2 is released, customers will be able to utilize Dynamic Memory to enable more effective memory management.

### Storage

A virtualization based solution for a multi-user system is very demanding on the storage subsystem. The more straightforward aspect of the storage requirements is the need for enough space to hold the .VHD files for the virtual machine images. In our tests using differencing disks to reduce space consumptions, each virtual machine still required 8-10 GB of disk space for its VHD file. In order to accommodate 250 virtual machines, the storage needs to have at least 2.5 TB available, which is reasonably easy to accommodate. But the challenge comes when sizing the storage throughput. An initial round of tests that used the same test progression as the RD Session Host server tests (groups of 10 users logging at a rate of 30 seconds between users and 5 minutes between groups for an overall rate of 10 users every 9 minutes and 30 seconds) had frequent failures in the automation infrastructure and resulted in significant strain on the storage system (large disk queue length values and degradation in duration of disk operations). These problems were caused by the amount of disk I/O issued as part of the machine start-up. Increasing the interval between user login from 30 to 60 seconds substantially improved the reliability of the test.

To better assess the impact of virtual machine startup, a test was run that consisted of simultaneously logging in 64 users against the 16 core test server. The 64 user set was chosen because this is the maximum number of users supported per server in a cluster environment, which is the likely configuration for a large capacity deployment. All users were able to successfully log in in 5 minutes from moment when the test started. The I/O activity as measured on the server is summarized in the “Logon” line in Table 3. The I/O operation rate is very high (3500 Ops/sec at the peak), a rate that would require about ~20 physical disk (assuming an I/O rate of ~180 Ops/sec per disk). This level of load explains well the correlation noticed between the virtual machine start rate and the instability of test execution: at higher start rates, the pressure on the disk system is becoming high enough to slow down other operations from the normal scenario execution that depend on disk I/O (file open and saves, for example) and thus cause issues in the UI automation. It is worth noting that the UI automation is able to handle well slow-downs in the order of 2-3 seconds, so the unreliability noticed in test execution would translate to a very poor user experience.

|  |  |  |  |
| --- | --- | --- | --- |
| 64 virtual machines | Read | Write | Read+Write |
|  | **Mbytes/sec** | **Ops/sec** | **Mbytes/sec** | **Ops/sec** | **Mbytes/sec** | **Ops/sec** |
|  | **Avg** | **Peak** | **Avg** | **Peak** | **Avg** | **Peak** | **Avg** | **Peak** | **Avg** | **Peak** | **Avg** | **Peak** |
| Logon | 10 | 220 | 350 | 2500 | 8 | 75 | 350 | 2500 | 18 | 224 | 700 | 3500 |
| Steady State | .8 | 3.6 | 40 | 260 | 3.3 | 10 | 130 | 220 | 4 | 12 | 170 | 400 |

**Table 3. Storage activity data**

For comparison, another test was conducted where 64 users were logged on in the system and then kept working for 1 hour without any other login activity to interfere with the normal scenario execution. The profile of disk activity during this period is listed in Table 3 in the **Steady State** line. The peak values for most counters are lower by almost an order of magnitude across the board. Only the average for write megabytes and write operations per second are somehow closer, but even for this value there is a 3x increase in the logon test.

Some best practices that help improve storage throughput:

1. Use SAN De-Duplication in combination with caching. This helps by caching a significant portion of each user’s VHD so that the SAN only has to go to disk for anything not in the cache. Additionally De-Duplication minimizes the disk footprint of potentially hundreds of virtual machines, each with a significant amount of common file and binary data. This type of architecture also offers the advantage of better existing management solution support within the virtual machine.
2. Use differencing Disks. In a similar way, these help by directing a significant fraction of the read operations during the virtual machine startup to the base image, which allows the storage system to use caching to service those requests faster. In our tests, the SAN controller achieved a very high cache hit rate for reads, in large part because of this configuration. Typically though, you would only use differencing disks in a pooled VDI architecture and the downside is that management becomes more difficult and duplicative of existing management processes.
3. Use a SSD device to hold the base image is another option given SSD’s ability to service tremendous amounts of operations per second.
4. Optimize your physical disk layout to deal with large amounts of write operations. RAID 0+1 is a more suitable configuration for this type of loads compared to RAID 5.
5. Keep the virtual machines in a saved state as opposed to turned off. Starting a virtual machine from a saved state significantly reduces the amount of I/O operations compared to a cold start.

### CPU

One of the most typical questions asked about RD Virtualization Host server capacity planning is how many users (or virtual machines) can be supported per CPU core. The reason why this question has such high importance is the nature of processor utilization that can exhibit quick changes and frequent peaks in usage.

In our tests, the amount of memory required for a supported Windows 7 guest configuration makes it such that the test will fail because of memory exhaustion, before saturating the processor. Since we know that for this scenario even the Windows 7 system will have no performance issues if configured with 512 MB RAM, we re-ran the Windows 7 guest operating system test cases with virtual machines configured to 512 MB RAM in order to allow the test to saturate the processor.

Note: This is not a supported configuration and should not be used in production environments.

The test results using this configuration are summarized in Table 5 and demonstrated that the 8 virtual machines/core support limit is a very real capability and not an upper maximum ceiling of performance. In our testing we saw up to 13.75 virtual machines/core, however this result needs to be tempered that this is a test case and not what we typically see in production which on average is around 5 to 6 virtual machines/core.

We would like to point out that there are considerations besides performance, mainly reliability and functionality related, that are included in the decision to support only 8 virtual machines per core from Hyper-V and we strongly suggest that deployments be limited to this value.

When looking at processor utilization within a Hyper-V machine it is important to consider the use of Second Level Address Translation (SLAT). This hardware feature takes advantage of various processor specific implementations: AMD implements this as Rapid Virtualization Indexing (RVI), also known as Nested Page Tables (NPT), starting with their Phenom generation processors while Intel implements Extended Page Tables in their Nehalem processors. We tested the impact of this feature by using a bcdedit option that allows turning off this feature. The results in Table 5 below show improvements in the 15-20% range of more users per host.

Another interesting observation is that using Windows 7 as the operating system in the guest virtual machines provided generally better performance when compared to using Windows XP. The difference was as high as ~10% and can be explained through better guest-host integration due to enhanced virtualization support in Windows 7.

|  |  |  |  |
| --- | --- | --- | --- |
| Server Configuration | Guest Configuration | Scenario | Performance support limit |
| CPU configuration | **Cores** | **RAM** | **SLAT Enabled** | **OS Version** | **RAM** |
| 4 x AMD Opteron 8378 Quad-core 2.4 GHz | 16 | 128 | Yes | Windows 7 | 512 | KW v2.1  | 190 |
| 4 x AMD Opteron 8378 Quad- core 2.4 GHz | 16 | 128 | No | Windows 7 | 512 | KW v2.1 | 160 |
| 4 x AMD Opteron 8378 Quad-core 2.4 GHz | 16 | 128 | Yes | Windows XP | 512 | KW v2.1  | 170 |
| 4 x AMD Opteron 8378 Quad-core 2.4 GHz | 16 | 128 | No | Windows XP | 512 | KW v2.1 | 140 |
| 4 x AMD Opteron 8378 Quad-core 2.4 GHz | 16 | 128 | Yes | Windows 7 | 512 | KW v2.1 w/o PPT | 220 |
| 4 x AMD Opteron 8378 Quad-core 2.4 GHz | 16 | 128 | No | Windows 7 | 512 | KW v2.1 w/o PPT | 190 |
| 4 x AMD Opteron 8378 Quad-core 2.4 GHz | 16 | 128 | Yes | Windows XP | 512 | KW v2.1 w/o PPT | 220 |

**Table 5 - Server capacity by CPU configuration and scenario**

# Conclusions

Capacity planning for Remote Desktop Virtualization Host deployments is subject to many variables. Based on usage scenario and hardware configuration, the variance in capacity can reach up to two orders of magnitude. If you need an accurate estimate, either deploying a pilot or running a usage specific load simulation are reliable ways to obtain a useful estimation for your specific deployment.

Remote Desktop Virtualization Host server can provide good consolidation for certain scenarios if care is taken when configuring the hardware and software. We expect 2-4 socket configurations using 4 core processors, 4 GB DIMMs and high performance storage solutions to get close to the Hyper-V support limit of 8 virtual machines per core (64-128 users overall).

In order, the sequence of likely capacity related issues to be found in production deployments are:

1. Disk I/O performance issues, especially related to virtual machine startup. We expect this to be the primary concern both for stand-alone and clustered RD Virtualization Host servers, but we recommend special attention to be given to the clustered deployments that use a Cluster Shared Volume.
2. Memory ceilings are likely to play a significant role when targeting very high consolidation ratios per server while using Windows 7 as a guest operating system. This can be mitigated by using higher capacity DIMMS.
3. Processor performance depends on the usage scenario. In general, we expect disk and memory to be more likely determining factors for the capacity limit.

When configuring an RD Virtualization Host server, give special attention to the following:

* Because storage is the most likely performance bottleneck, size your storage to properly handle the I/O load generated by virtual machine state changes. If a pilot or simulation is not feasible, a good guideline is to provision one disk spindle for each 4 active virtual machines. Use disk configurations that have good write performance (like RAID 1+0).
* Use SAN-based disk de-duplication and caching to reduce disk read load and enable your storage solution to speed up performance by caching a significant portion of the image.
* Plan for provisioning enough memory per server to allow taking full advantage of the processing power available on the box. Server configurations with at least 8 DIMMs per socket, populated with 4 GB DIMMs are a great compromise from a price/performance viewpoint.
* Use at least 4 core CPUs to not only increase overall server capacity, but also allow a server to better absorb temporary peaks in processor load like logon bursts or variation in load.
* Make sure the processor supports paging at the hardware level (RVI for AMD, EPT for Intel).

# Appendix A: Test Hardware Details

The following servers were tested for Remote Desktop Services capacity planning data:

* Hard Drives Northwest
	+ 4 x AMD Opteron 8378 2.4 GHz CPUs (Quad-core)
	+ 128 GB DDR2 memory
	+ 500 GB 7.2K RPM SATA disk
	+ 4 Gbps Emulex LPe 1150 HBA
	+ 100/1000 Mbps Intel NIC
* HP ProLiant DL 370 G6
	+ 2 x Intel Xeon E5530 2.4 GHz CPUs (Quad-core)
	+ 72 GB PC3-10600R memory
	+ 2 x 146 GB 15K RPM SAS drives
	+ 4 Gbps Emulex LPe 1150 HBA
	+ 100/1000 Mbps Intel NIC
* SAN
	+ Emulex CX3-20 controller 16 GB cache
	+ 15 x 146 GB, 15K RPM disks, RAID 0

Other components of the test laboratory included:

* RD Connection Broker: Dell GX 620
	+ Intel Pentium D 3.40Ghz, 64bit, EM64T
	+ 2 GB memory
	+ Windows Server 2008 R2 Standard
* Domain Controller and Test Controller: HP Proliant DL145 G2
	+ Dual core AMD Opteron processor 280 2.4GHz
	+ 2 GB memory
	+ Windows Server 2008 Enterprise
	+ This server is the DHCP and DNS server for the domain. It manages the workstations running Windows 7 Ultimate, including script control, software distribution, and remote reset of the workstations.
* E-mail Server and Web Server: HP Proliant DL145 G2
	+ Dual core AMD Opteron processor 280 2.4GHz
	+ 2 GB memory
	+ Windows Server 2008 Enterprise, IIS role installed
	+ Exchange Server 2007
* Workstations: Dell Optiplex GX 620
	+ Intel Pentium D 3.40Ghz, 64bit, EM64T
	+ 2 GB memory
	+ Windows 7 Ultimate

# Appendix B: Testing Tools

Microsoft developed the Remote Desktop Load Simulation Tools to perform scalability testing. Remote Desktop Load Simulation Tools is a suite of tools that assists organizations with capacity planning for Remote Desktop Services running on Windows Server 2008 R2. These tools allow organizations to easily place and manage simulated loads on a server. This in turn can allow an organization to determine whether or not its environment is able to handle the load that the organization expects to place on it. If you’d like to conduct a capacity planning exercise for your specific deployment, you can download the [Remote Desktop Load Simulation Tools](http://go.microsoft.com/fwlink/?LinkId=178956) from the Microsoft Download Center (http://go.microsoft.com/fwlink/?LinkId=178956).

The automation tools included in the suite are described below.

## Test control infrastructure

* **Test Controller - RDLoadSimulationController.exe**

The RDLoadSimulationController tool is the central control point for the load simulation testing. It is typically installed on the test controller computer. RDLoadSimulationController controls all test parameters and defines the progression of the simulated user load. It also controls all custom actions that are executed at any point during the test process. It communicates with RDLoadSimulationClients and RDLoadSimulationServerAgent to synchronize and drive the client-server remote desktop automation. It commands the RDLoadSimulationClients to run scripts that load the RD Session Host server at operator-specified intervals.

* **Client Agent - RDLoadSimulationClient.exe**

The RDLoadSimulationClient tool controls the client side of the load simulation testing. RDLoadSimulationClient is typically installed on the test client computers. RDLoadSimulationClient receives commands from RDLoadSimulationController to run scripts that load the RD Session Host server at operator-specified intervals. It executes custom commands received from the RDLoadSimulationController and also sends the status of the executing scripts to the RDLoadSimulationController. RDLoadSimulationClient also performs desktop management on the test client computers. It creates a new desktop for each script that it launches and provides the means to navigate between all desktops.

* **Server Agent - RDLoadSimulationServerAgent.exe**

The RDLoadSimulationServerAgent tool runs on the target Remote Desktop Virtualization Host server. It runs custom commands that are sent to it by the RDLoadSimulationController. It is also used by RDLoadSimulationController for test synchronization.

* **SwitchDesktop.exe**

The SwitchDesktop tool runs on the test client computers. It runs inside each new desktop that is created on the client. Its only function is to provide a way to switch back to the default desktop where the RDLoadSimulationClient is running.

## Scenario execution tools

* **Script automation tool - RemoteUIControl.dll**

RemoteUIControl.dll is a COM based tool which provides functionality for driving the client side load simulation. It exposes functionality for creating RDP connections to the server, as well as sending keyboard input to the Remote Desktop Services session. It synchronizes executions based on drawing events in the applications that are running inside the Remote Desktop Services session.

* **RUIDCOM.exe**

RUIDCOM is a DCOM tool which is a wrapper around RemoteUIControl.dll. This tool exposes all the functionality of RemoteUIControl.dll. Test scripts use RUIDCOM instead of directly using RemoteUIControl.dll because it provides some extra functionality. RUIDCOM communicates with the RDLoadSimulationClient to report the status of a simulated user.

* **TSAccSessionAgent.exe**

TSAccSessionAgent runs on the virtual machines. One instance of TSAccSessionAgent runs inside every session that is created for a simulated test user. RemoteUIControl.dll on the client side communicates with TSAccSessionAgent to synchronize user input with drawing events in the applications that are running inside the session.

# Appendix C: Test Scenario Definitions and Workflow

## Knowledge Worker v2.1

Typing Speed = 35 words per minute

Definition: the Knowledge Worker scenario includes creating and saving Word documents, printing Excel spreadsheets, communicating by e-mail in Outlook, adding slides to PowerPoint presentations, running slide shows, and browsing Web pages in Internet Explorer. The following workflow details the scenario.

* Connect User “smcxxx”
* **Start (Outlook) -** Send new e-mail messages
* Send a new appointment invitation
* Send a new e-mail message
* Minimize Outlook
* **Start (Word) -** Start and exit Word
* **Start (Microsoft Excel)** - Start and exit Excel
* loop(forever)
* **Start (Word) -** Type a page of text and print
* Open a Word document
* Type a page of text
* Modify and format text
* Check spelling
* Print
* Save
* Exit Word
* **Start (Microsoft Excel)** - Load Excel spreadsheet, modify, and print it
* Load Excel spreadsheet
* Modify data and format
* Print
* Save
* Exit Excel
* **Start (PowerPoint) -** Load presentation and run slide show
* Load a PowerPoint presentation
* Navigate
* Add a new slide
* Format text
* Run slide show
* Save file
* Exit PowerPoint
* **Switch To Process, (Outlook) -** send e-mail,read message, and respond
* Send e-mail to other users
* Read e-mail and respond
* Minimize Outlook
* **Start (Internet Explorer) -** Load presentation and run slide show
* Loop (2)

URL http://tsexchange/tsperf/WindowsServer.htm
URL http://tsexchange/tsperf/Office.htm
URL http://tsexchange/tsperf/MSNMoney.htm

* End of loop
* Exit Internet Explorer
* End of loop

This version of the script differs from the v2 version by a few minor changes:

* Different timing for actions in the “Start Outlook” section. Timing was redistributed to better handle random slow-downs experienced in virtual machines.
* More reliable events used to synchronize some Excel actions
* Timing changes for PowerPoint actions related to saving/opening the file
* Moving send-receive at the end of the Outlook actions inside the repeated loop
* Adding one more user (SMC000) in the “To” list of the mail message sent
* Changed timing in the ”Outlook “ section of the loop to compensate for increases in the PowerPoint section
* Additional actions in the first iteration to handle IE behavior at first start

# Appendix D: Remote Desktop Virtualization Host Settings

* System configuration
* System disk - locally attached SATA disk, formatted using NTFS
* Virtual machine storage disk – SAN mounted LUN, 15 146GB 15K FC disks configured in RAID 0, formatted as NTFS
* Intel Pro/1000 MT network adapters connected to test network
* Windows Server 2008 R2 Enterprise operating system installed on locally attached SATA disk
* Server joined as a member to Windows Server 2008 test domain
* Roles Installed
* Remote Desktop Services/Remote Desktop Virtualization Host
* Hyper-V
* Hyper-V configuration
* Virtual Network:
	+ Connected to Intel Pro/1000 MT network adapter
	+ External connection type
	+ Shared with OS management functions
* Virtual Machine Storage configured to use the virtual machine storage disk

# Appendix E: Connection Broker Settings

* System configuration
* Windows Server 2008 R2 Standard operating system installed on SATA Hard Drive
* Roles Installed
* Remote Desktop Services/Remote Desktop Session Host configured in Redirector mode
* Remote Desktop Services/Remote Desktop Connection Broker
* RD Connection Broker Configuration
* Personalized virtual machine pool including all virtual machines from the test server
* Each virtual machine is associated to the a singles test user

# Appendix F: Guest Virtual Machine Settings

* Virtual machine configuration:
* 1 Virtual processor
* 512 MB or 1024 MB RAM based on test case requirements
* Network Adapter connected to the Virtual Switch, static MAC assignment
* IDE Hard Drive
* Operating system installation
* Hard Drive formatted by using NTFS
* Networking left at default with typical network settings
* Joined as a member to a Windows Server 2008 test domain
* Configured to allow access to RD Virtualization Host role service
* Default page file configuration
* Password change for the machine domain account was disabled.
* RDP protocol client settings
* Disable all redirections (drive, Windows printer, Clipboard, , LPT, COM, audio and video playback, audio recording, Plug and Play devices)
* Color depth is set to 16 bit for Remote Desktop Services connections
* Office 2007 settings
* Office 2007 installed enabling the following features from Office customization
	+ Microsoft Office Excel
	+ Microsoft Office Outlook
	+ Microsoft Office PowerPoint
	+ Microsoft Office Word
	+ Office Shared Features
	+ Office Tools
* Outlook settings
* Mailbox on Exchange server
* E-mail options
* AutoSave of messages disabled
* Automatic name checking disabled
* Do Not Display New Mail Alert for users enabled
* Suggest names while completing To, Cc, and Bcc fields disabled
* Return e-mail alias if it exactly matches the provided e-mail address when searching OAB enabled
* AutoArchive disabled
* Word Settings
	+ Background grammar-checking disabled
	+ Check Grammar With spelling disabled
	+ Background saves disabled
	+ Save AutoRecover information disabled
	+ Always show full menus enabled
	+ Microsoft Office Online disabled
	+ Customer Experience Improvement Program disabled
	+ Automatically receive small updates to improve reliability disabled
* Internet Explorer Settings
* Protected Mode was disabled for all security zones
* Low Rights Mode was disabled
* Information Mode was disabled
* Printer settings
* HP Color LaserJet 9500 PCL 6 created to print to NUL port
* User profiles
* Configuration script executed to pre-create cached profiles, copy template files for applications, configure e-mail accounts, and set home page on Internet Explorer
* Roaming profiles used for all users
* General settings
	+ Disable screen saver for all users through Group Policy
	+ Disable Windows Firewall
	+ Enable Remote Desktop Connections
	+ Delete all office and XPS printers installed at setup
1. The number of cores specified on the Xenon E5530 represents the physical number of cores on the machine and not the number of cores reported by the Operating System. [↑](#footnote-ref-1)
2. This is based on the stated support limits for Hyper-V. Specifically this takes into account that the maximum supported ratio of Virtual Processors to Logical Processors is 8:1. [↑](#footnote-ref-2)
3. This is the maximum number of users the system was able to support without a significant degradation to user experience. This limit was not always determined based on a degradation of user experience; in some cases this was simple the point where no more virtual machines could be started due to lack of memory. [↑](#footnote-ref-3)