

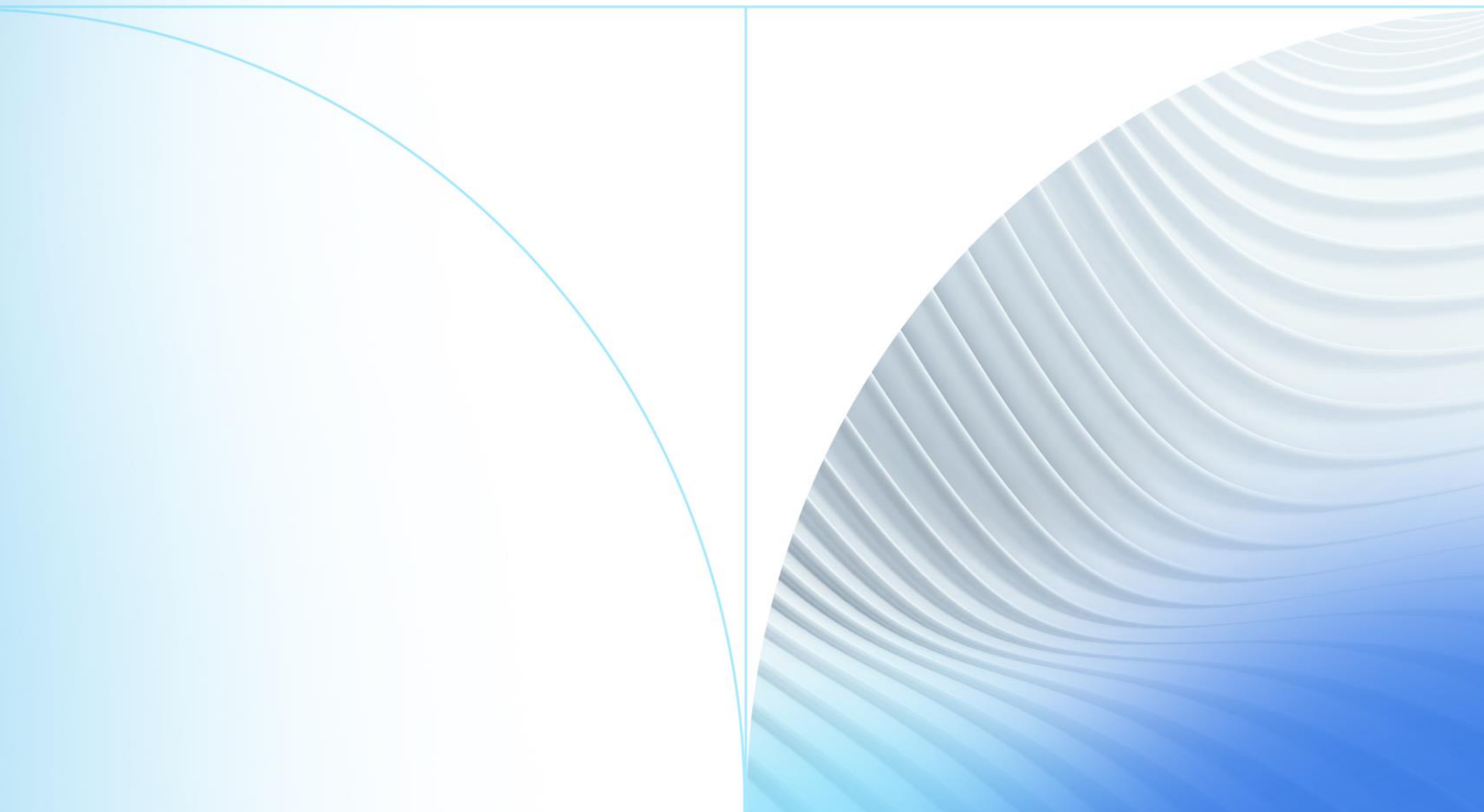


Financial Performance

PROFITstar & PROFITability

Release 2018

Hardware Sizing Recommendations For Optimal Performance



© 1999–2022 Jack Henry & Associates, Inc.

All rights reserved. Information in this document is subject to change without notice. Dates contained in this document are provided as estimates only and can be changed at any time at the sole discretion of Jack Henry & Associates, Inc.

Printed in the United States of America.

No part of this document may be copied, reproduced, stored in a retrieval system, displayed, distributed or transmitted in any form or any means whatsoever (electronic, mechanical or otherwise), including by photocopying or recording for any purpose, without the prior written permission of Jack Henry & Associates, Inc. Making unauthorized copies of this document for any purpose other than your own personal use is a violation of United States copyright laws.

Any unauthorized use of Jack Henry & Associates, Inc.'s, trademarks and service marks is strictly prohibited. A list of registered and unregistered marks owned or licensed by Jack Henry & Associates, Inc. is located at: <https://www.jackhenry.com/more-from-jha/pages/trademark-notice.aspx>.

Various other trademarks and service marks used or referenced in this document are the property of their respective owners.

Navigating the Hardware Sizing Process – FAQs.....	1
How Do We Know If We Should Upgrade Our Hardware?	1
We Want to Upgrade the Hardware, but Cannot Do It All at Once.....	1
We're Large but Only Have One Box. Do We Really Need Three?	2
Summary	2
Key Takeaways:.....	3
Key Things to Consider:	4
Approach.....	4
Background.....	4
Our Take.....	5
Our Findings.....	6
Conclusion.....	7
Appendix 1 – Hardware	8
Appendix 2 – PROFITstar Deployment Architecture.....	9
Appendix 3 – Small.....	10
Appendix 4 – Medium.....	11
Appendix 5 – Large.....	12
Appendix 6 – VM Setup Guide	13
Appendix 7 – SQL Best Practices for the Application	15

Navigating the Hardware Sizing Process – FAQs

You can load 2018 on your existing hardware and it will be faster than 2017, however, you will not see the full effect of the performance improvements without the hardware changes recommended in this guide.

How Do We Know If We Should Upgrade Our Hardware?

Please see where your FI fits in our Small / Medium / Large size-based categories, listed on the next page. Then, if you perform your own timings utilizing our use cases and compare the results to those in Appendix [3](#), [4](#), or [5](#), you can get an idea of how much you might benefit from making the recommended hardware changes. Generally, the larger the institution the more benefit you will see.

We Want to Upgrade the Hardware but Cannot Do It All at Once.

Regardless of size, we recommend prioritizing your hardware upgrades as follows:

1. SSD for your SQL Server MDF
2. SSD for your SQL Server LDF
3. Follow [Appendix 7](#) to configure your SQL box. The first three recommendations will give you the most bang for your buck, while items 4, 5, and 6 are best practices.

What comes next depends on your size. We suggest that after each change you redo the timing tests and see how close you are.

Typically, the two outlier tests are User Defined Quick Projections and Balance Nested Units. Increasing the CPU speed and count on the PROFITstar server will improve the time required for each. Fast CPUs are key for this box.

We're Large but Only Have One Box. Do We Really Need Three?

No, you don't need to buy one or two more boxes if you are happy with the performance.

See [Appendix 5](#). While the highlighted version is the three-box model, you can see the one-box results all the way to the right. The 1tier_HW TOR 7 version used 8 cores and 16 GB of ram. If we had increased the cores, you would see the timings fall even more for Balance Nested Orgs and User Defined Quick Projections.

Summary

To enhance the performance of PROFITstar® and PROFITability®, we worked with product management to identify five key slow areas in our programs and have focused on improving the performance of these areas. We then took this code and ran it on different hardware configurations to determine the optimal configuration for our clients, based on sizing, as defined below (Figure 1).

□	Asset-Size□	#-of-Orgs□	#-BS-Accts□	#-IE-Accts□	Db-Size-(Gig)□	
Large□	1b+□	80+□	900+□	900+□	10+□	
Medium□	500-1.b□	20-80□	500-900□	500-900□	3-10□	
Small□	0-500m□	0-20□	0-500□	0-500□	0-3□	

Figure 1

This led us to the following conclusions (Figure 2):

Key Takeaways:

- Most of our customers will see a significant improvement in performance by going to SSDs for the SQL drives. These gains will be multiplied once they are on the 2018 code base.
- Regardless of size, financial institutions that extensively use formulas will benefit from having an increased number of fast CPUs for the Profitstar Server box, also known as our Middle tier.

Use Cases	Physical Machine		
	Small (Min)	Medium (Min)	Large (Min)
Hardware Configuration	SQL - LCPU: 8 Memory: 16 PServer - LCPU: 8 Memory: 16 Client - LCPU: 4 Memory: 8	SQL - LCPU: 16 Memory: 32 PServer - LCPU: 24 Memory: 72 Client - LCPU: 8 Memory: 16	SQL - LCPU: 16 Memory: 32 PServer - LCPU: 24 Memory: 72 Client - LCPU: 8 Memory: 16

Use Cases	Virtual Machines		
	Small (Min)	Medium (Min)	Large (Min)
Hardware Configuration	All In 1 LCPU: 8 Memory: 16	All In 1 LCPU: 8 Memory: 16	SQL - LCPU: 8 Memory: 16 PServer - LCPU: 12 Memory: 16 Client - LCPU: 4 Memory: 8

Figure 2

- In Figure 2 above, 72 GB of memory in the Physical solution is an overstatement. We did not see any memory pressure at 16 GB; the key is the 24 LCPUs.
- The VM Hardware we were testing on could be purchased today for around \$11K, per Microsoft's Dell representative. This would meet the needs of our largest customers.

Key Things to Consider:

- Do not compare Physical and Virtual times. The underlying hardware is very different and makes such a comparison invalid.
- These times were all done on dedicated hardware. The recommendations above may not be optimal for your situation. See our appendices for detailed results.
- Timings will vary from customer to customer, but you should be in the ballpark.
- We are following the SQL Server best practices for our application—found in [Appendix 7](#).
- In the virtual world, we are running on a Hyper-V setup—found in [Appendix 6](#).

Approach

Background

We took our 2017 release candidate to the Microsoft testing facilities, for a two-week engagement. This engagement was split into two phases: Phase 1 – further optimize the application’s performance. Phase 2 – determine optimal cost-effective hardware recommendations. Our customer base is very diverse, from tiny mom and pop shops to large \$7B+ behemoth institutions. In the past, we offered guidance on minimum requirements that would typically yield just minimal performance to all but the smallest of our customers. Phase 2 of this engagement corrects this shortcoming in our approach and allows us to recommend hardware to our customers, sized in a way that fits them.

NOTE We reran our performance tests on the 2018 version to verify that all performance improvements were maintained.

Our Take

To accomplish our goal, we paired with Customer Service to reach out to six of our customers—two in each category: Small, Medium, and Large, as defined in [Figure 1](#). We asked these customers if they would be willing to participate, and then obtained backups of their data so that this sampling was run on real customer data. We installed and configured our app on a 3-tier system. Then, using P2V, virtualized the system on our Hyper-V setup. We ran the optimized code on wide-open hardware. (See [Appendix 1 – Hardware](#) for a breakdown of the hardware requirements.) We recorded the findings as a baseline. We would run the tests on our physical and virtual boxes in an automated parallel fashion. Next, we would tweak CPUs / Memory or both on a single tier of the system at a time—SQL, PServer, or Client—and then run our tests and record our results.

NOTE We did this in the Physical and Virtual environments at the same time.

We called each test run a TOR (Test of Record), and you can see the details of our tests in [Appendix 3 – Small](#), [Appendix 4 – Medium](#), and [Appendix 5 – Large](#).

Our Findings

For the Large tier, we found our tests were very repeatable, and as such, we went with the lowest sum of all the tests per TOR and picked the column with the lowest sum.

For the Medium tier, we did the same as we did with the Large tier, which worked great for the Virtual Boxes but not so well for the Physical ones, where the wide open boxes we used for the baseline had the lowest time, but that hardware would cost quite a bit more for a gain of 40 seconds. The other issue here was the variability of the test runs, on the Medium and Small. On the Balance Nested Units and the User-Defined Quick Projections use cases, it was common to have a 5 to 20-second variability.

Therefore, we chose TOR 1 to represent the Medium physical hardware recommendation. For the Small tier, the virtual TOR 1 and TOR 4 were equal, based on our +/- error factor, so we went with the more cost-effective TOR 4. For the virtual, TOR 7 was the clear winner.

Conclusion

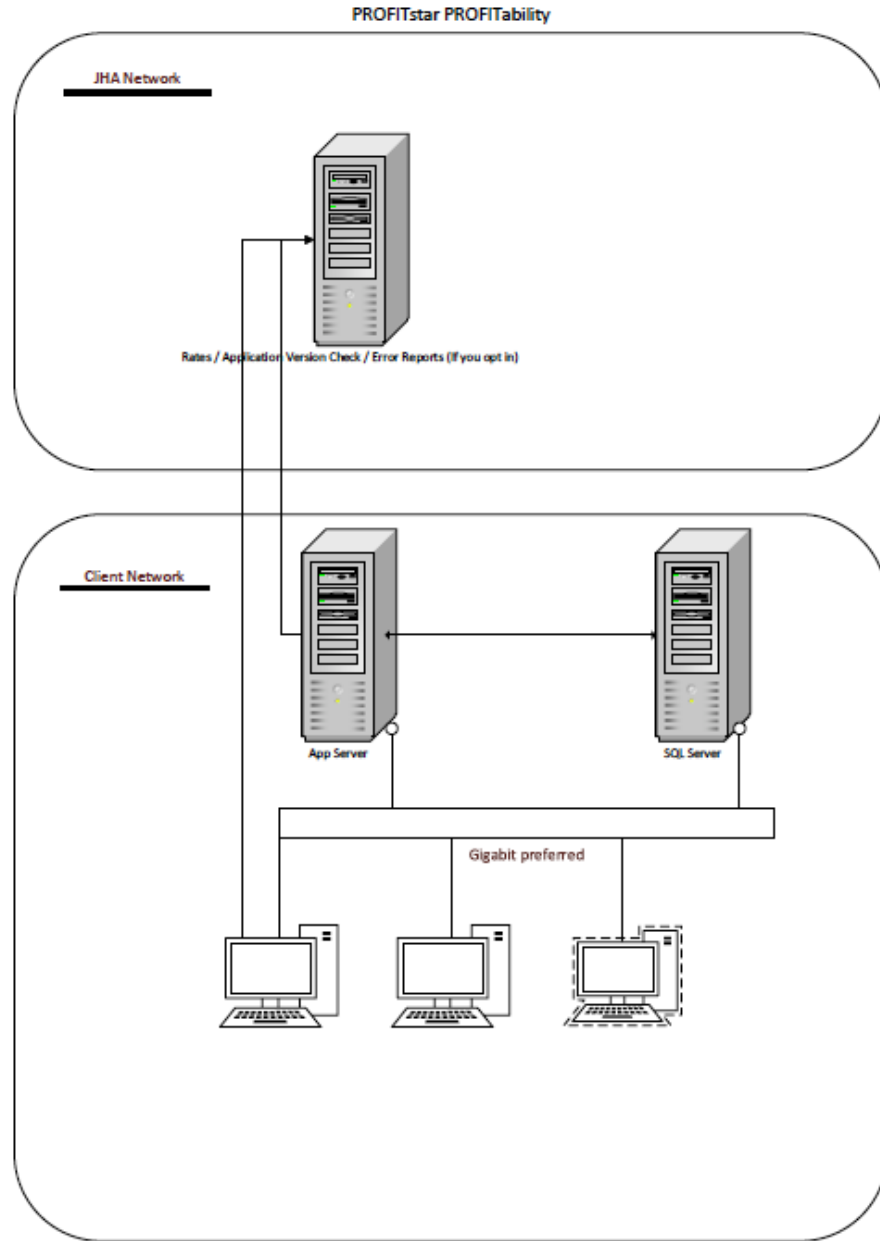
The 2018 release of PROFITstar® and PROFITability® has some key performance enhancements that allow us to leverage more CPUs to do our processing. This makes it possible to do more time-consuming work in parallel, cutting down on the overall duration of long-running tasks. As a result, customers that use a lot of formulas will get a huge speed improvement by giving the PServer / App Server more CPUs—the faster and more plentiful, the better. Most of our processes read and write large amounts of data at once, so the faster the SQL server drives, the better, especially when you follow the best practice guide for installing SQL, which includes some of these key points for our app:

- Separate MDFs and LDFs onto separate physical SSDs.
- Format MDFs and LDFs with a 64K block size.
- Put Temp DBs MDFs on the OS partition, and LDF on the LDF drive.
- Set MDP and Max CPUs—8 if you have more. See [Appendix 7](#) for details about what to set these settings to.

Appendix 1 – Hardware

Machine	CPU	Ram Gig	OS	Drives - SSD
VM Host	Dell R720 12 core, 48L Core	128	Windows server 2012 64-bit data center	200 / 400 / 400 / 300
SQL	Dell R710 6 core, 24 LCore	72	windows server 2012 64-bit	400/400 /400
PServer	Dell R710 6 core, 24 LCore	72	windows server 2012 64-bit	200/200/200
Client	i7 8 LCore	16	Win 10 server 64 bit	127

Appendix 2 – PROFITstar Deployment Architecture



Appendix 3 – Small

Physical Machine										
Small										
Use Cases	Goal (Mln)	Stretch Goal (Mln)	Baseline: SQL - LCPU: 24 Memory: 72 PServer - LCPU: 24 Memory: 72 Client - LCPU: 8 Memory: 16	HW_TOR1: SQL - LCPU: 16 Memory: 32 PServer - LCPU: 24 Memory: 72 Client - LCPU: 8 Memory: 16	HW_TOR2: SQL - LCPU: 8 Memory: 16 PServer - LCPU: 24 Memory: 32 Client - LCPU: 4 Memory: 8	HW_TOR3: SQL - LCPU: 8 Memory: 16 PServer - LCPU: 12 Memory: 16 Client - LCPU: 4 Memory: 8	HW_TOR4: SQL - LCPU: 8 Memory: 16 Memory: 16 Client - LCPU: 4 Memory: 8	HW_TOR5: SQL - LCPU: 8 Memory: 16 PServer - LCPU: 4 Memory: 8 Client - LCPU: 4 Memory: 8	HW_TOR 6: SQL - LCPU: 4 Memory: 8 PServer - LCPU: 4 Memory: 8 Client - LCPU: 4 Memory: 8	1tier_HW TOR 7: LCPU: 8 Memory: 16
Balance Nested Orgs	2	2	0.53	0.79	0.95	0.79	0.87	0.79	0.87	0.96
User-Defined Quick Projections	2	2	0.62	0.87	0.78	0.95	0.79	1.20	1.21	1.14
Move Month Forward	1	1	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
Create New Model	1	1	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20
Copy Model	1	1	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.34

Virtual Machines										
Small										
Use Cases	Goal (Mln)	Stretch Goal (Mln)	Baseline: SQL - LCPU: 24 Memory: 32 PServer - LCPU: 24 Memory: 32 Client - LCPU: 8 Memory: 16	HW_TOR1: SQL - LCPU: 16 Memory: 32 PServer - LCPU: 24 Memory: 32 Client - LCPU: 8 Memory: 16	HW_TOR2: SQL - LCPU: 8 Memory: 16 PServer - LCPU: 24 Memory: 32 Client - LCPU: 4 Memory: 8	HW_TOR 3: SQL - LCPU: 8 Memory: 16 PServer - LCPU: 12 Memory: 16 Client - LCPU: 4 Memory: 8	HW_TOR 4: SQL - LCPU: 8 Memory: 16 PServer - LCPU: 8 Memory: 16 Client - LCPU: 4 Memory: 8	HW_TOR 5: SQL - LCPU: 8 Memory: 16 PServer - LCPU: 4 Memory: 8 Client - LCPU: 4 Memory: 8	HW_TOR 6: SQL - LCPU: 4 Memory: 8 PServer - L CPU: 4 Memory: 8 Client - CPU: 4 Memory: 8	1tier_HW_TOR 7: LCPU: 8 Memory: 16
Balance Nested Orgs	2	2	0.79	0.95	0.95	0.87	0.62	0.70	0.79	0.71
User-Defined Quick Projections	2	2	1.04	1.13	0.95	0.87	1.12	1.12	1.12	0.78
Move Month Forward	1	1	0.07	0.06	0.06	0.06	0.06	0.06	0.06	0.07
Create New Model	1	1	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20
Copy Model	1	1	0.26	0.26	0.26	0.25	0.26	0.34	0.25	0.26

Appendix 4 – Medium

Physical Machine										
Medium										
Use Cases	Goal (Mln)	Stretch Goal (Mln)	Baseline: SQL - LCPU: 24 Memory: 72 PServer - LCPU: 24 Memory: 72 Client - LCPU: 8 Memory: 16	HW_TOR1: SQL - LCPU: 16 Memory: 32 PServer - LCPU: 24 Memory: 72 Client - LCPU: 8 Memory: 16	HW_TOR2: SQL - LCPU: 8 Memory: 16 PServer - LCPU: 24 Memory: 32 Client - LCPU: 4 Memory: 8	HW_TOR3: SQL - LCPU: 8 Memory: 16 PServer - LCPU: 12 Memory: 16 Client - LCPU: 4 Memory: 8	HW_TOR4: SQL - LCPU: 8 Memory: 16 PServer - LCPU: 8 Memory: 16 Client - LCPU: 4 Memory: 8	HW_TOR5: SQL - LCPU: 8 Memory: 16 PServer - LCPU: 4 Memory: 8 Client - LCPU: 4 Memory: 8	HW_TOR 6: SQL - LCPU: 4 Memory: 8 PServer - LCPU: 4 Memory: 8 Client - LCPU: 4 Memory: 8	1tier_HW TOR 7: LCPU: 8 Memory: 16
Balance Nested Orgs	5	5	1.45	2.20	2.28	2.80	2.38	2.38	2.47	2.72
User-Defined Quick Projections	5	5	2.55	2.20	2.38	1.87	2.38	4.55	4.55	3.05
Move Month Forward	1	1	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
Create New Model	1	1	0.20	0.20	0.29	0.20	0.20	0.28	0.28	0.29
Copy Model	1	1	0.28	0.28	0.37	0.28	0.28	0.37	0.37	0.37

Virtual Machines										
Medium										
Use Cases	Goal (Mln)	Stretch Goal (Mln)	Baseline: SQL - LCPU: 24 Memory: 32 PServer - LCPU: 24 Memory: 32 Client - LCPU: 8 Memory: 16	HW_TOR1: SQL - LCPU: 16 Memory: 32 PServer - LCPU: 24 Memory: 32 Client - LCPU: 8 Memory: 16	HW_TOR2: SQL - LCPU: 8 Memory: 16 PServer - LCPU: 24 Memory: 32 Client - LCPU: 4 Memory: 8	HW_TOR 3: SQL - LCPU: 8 Memory: 16 PServer - LCPU: 12 Memory: 16 Client - LCPU: 4 Memory: 8	HW_TOR 4: SQL - LCPU: 8 Memory: 16 PServer - LCPU: 8 Memory: 16 Client - LCPU: 4 Memory: 8	HW_TOR 5: SQL - LCPU: 8 Memory: 16 PServer - LCPU: 4 Memory: 8 Client - LCPU: 4 Memory: 8	HW_TOR 6: SQL - LCPU: 4 Memory: 8 PServer - L CPU: 4 Memory: 8 Client - CPU: 4 Memory: 8	1tier_HW TOR 7: LCPU: 8 Memory: 16
Balance Nested Orgs	5	5	1.78	3.30	3.12	2.43	1.54	2.28	2.37	1.45
User-Defined Quick Projections	5	5	3.20	3.27	3.53	2.62	2.12	4.20	4.28	1.95
Move Month Forward	1	1	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.50
Create New Model	1	1	0.20	0.20	0.20	0.20	0.20	0.28	0.20	0.20
Copy Model	1	1	0.37	0.29	0.37	0.37	0.37	0.37	0.37	0.28

Appendix 5 – Large

Physical Machine										
Large										
Use Cases	Goal (Mln)	Stretch Goal (Mln)	Baseline: SQL - LCPU: 24 Memory: 72 PServer - LCPU: 24 Memory: 72 Client - LCPU: 8 Memory: 16	HW_TOR1: SQL - LCPU: 16 Memory: 32 PServer - LCPU: 24 Memory: 72 Client - LCPU: 8 Memory: 16	HW_TOR2: SQL - LCPU: 8 Memory: 16 PServer - LCPU: 24 Memory: 32 Client - LCPU: 4 Memory: 8	HW_TOR3: SQL - LCPU: 8 Memory: 16 PServer - LCPU: 12 Memory: 16 Client - LCPU: 4 Memory: 8	HW_TOR4: SQL - LCPU: 8 Memory: 16 PServer - LCPU: 8 Memory: 16 Client - LCPU: 4 Memory: 8	HW_TOR5: SQL - LCPU: 8 Memory: 16 PServer - LCPU: 4 Memory: 8 Client - LCPU: 4 Memory: 8	HW_TOR 6: SQL - LCPU: 4 Memory: 8 PServer - LCPU: 4 Memory: 8 Client - LCPU: 4 Memory: 8	1tier_HW TOR 7: LCPU: 8 Memory: 16
Balance Nested Orgs	150	60	25.70	16.12	17.87	25.07	19.23	35.42	36.28	25.25
User-Defined Quick Projections	300	120	35.40	26.18	31.03	45.05	65.78	126.88	122.82	94.93
Move Month Forward	1	1	0.37	0.36	0.36	0.37	0.36	0.36	0.45	0.36
Create New Model	2	1	0.79	0.78	0.70	0.78	0.70	1.04	1.20	0.96
Copy Model	5	1	1.62	1.72	1.78	1.87	1.78	1.70	2.03	1.78

Virtual Machines										
Large										
Use Cases	Goal	Stretch Goal	Baseline: SQL - LCPU: 24 Memory: 32 PServer - LCPU: 24 Memory: 32 Client - LCPU: 8 Memory: 16	HW_TOR1: SQL - LCPU: 16 Memory: 32 PServer - LCPU: 24 Memory: 32 Client - LCPU: 8 Memory: 16	HW_TOR2: SQL - LCPU: 8 Memory: 16 PServer - LCPU: 24 Memory: 32 Client - LCPU: 4 Memory: 8	HW_TOR 3: SQL - LCPU: 8 Memory: 16 PServer - LCPU: 12 Memory: 16 Client - LCPU: 4 Memory: 8	HW_TOR 4: SQL - LCPU: 8 Memory: 16 PServer - LCPU: 8 Memory: 16 Client - LCPU: 4 Memory: 8	HW_TOR 5: SQL - LCPU: 8 Memory: 16 PServer - LCPU: 4 Memory: 8 Client - LCPU: 4 Memory: 8	HW_TOR 6: SQL - LCPU: 4 Memory: 8 PServer - L CPU: 4 Memory: 8 Client - CPU: 4 Memory: 8	1tier_HW TOR 7: LCPU: 8 Memory: 16
Balance Nested Orgs	150	60	31.42	30.22	27.52	23.70	14.95	30.55	30.53	15.22
User-Defined Quick Projections	300	120	42.07	37.23	33.88	28.85	41.65	104.10	103.83	54.80
Move Month Forward	1	1	0.37	0.36	0.36	0.37	0.37	0.36	0.45	0.37
Create New Model	2	1	0.37	0.45	0.54	0.53	0.54	0.54	0.79	0.45
Copy Model	5	1	1.87	1.70	1.87	1.87	1.78	1.87	1.87	1.87

Appendix 6 – VM Setup Guide

Below is some automation to install the Hyper-V role and deploy 4 servers using a base image vhd file.

Assumptions for this script are as follows:

1. windows 2012 R2 server
2. Base windows Server 2012 R2 vhd file located at the root of L:\
3. Each server needs a secondary data drive of 150GB

This script does not include joining the virtual machines to any sort of domain. Further effort would need to be done by JH if that is required.

```
#Install Hyper-V role and tools. This will require a reboot, which is set with the -
Restart switch below.
```

```
Install-WindowsFeature -Name Hyper-V -IncludeAllSubFeature -IncludeManagementTools -
Restart
```

```
#Run to get list of network adapters so you can choose which one you want to set up
and let Hyper-V use in the next step.
```

```
netsh interface ipv4 show interfaces
```

```
#Creates the network switch that the VMs will use using one of your NICs found in the
command above. In my case, the NIC I am using is "NIC4" which you would need to change
for your env.
```

```
New-VmSwitch -Name External -NetAdapterName "NIC4" -AllowManagementOS $false
```

```
#Sets the Path to use for newly created VMs.
```

```
Set-VMHost -VirtualHardDiskPath 'L:VirtualMachines'
```

```
Set-VMHost -VirtualMachinePath 'L:VirtualMachines'
```

```
#This creates a folder for SERVER01-04 and copies the base vhd in each folder. This
part must be done from a command prompt not powershell. And, to see the folders that
are created you will need to uncheck the "Hide protected operating system files" in
file explorer.
```

```
for /l %A in (1,1,4) do robocopy L:\ L:\VirtualMachines\SERVER0%A WINSRV2012R2.VHDX
```

```
#This will deploy SERVER01-04 using the VHDX file copied in the step above.
```

```
$VMName = "SERVER0"
```

```
$VMFirstNumber = 1
```



```

$VMTotal = 4

#Loop for incremental changes
do {

    new-vm -Name $VMName$VMFirstNumber -VHDPath
L:\VirtualMachines\$VMName$VMFirstNumber\WINSRV2012R2.VHDX -SwitchName "External" -
Path L:\VirtualMachines\ -MemoryStartupBytes 16GB -Generation 1

    Set-VMProcessor $VMName$VMFirstNumber -Count 2

#This adds a secondary drive with the size of 150GB. This drive will need to be
onlined and initialized from within disk management inside the virtual machine OS
before it is usable

    New-VHD -SizeBytes 150GB -Path
L:\VirtualMachines\$VMName$VMFirstNumber\$VMName$VMFirstNumber.VHDX
    Add-VMHardDiskDrive -VMName $VMName$VMFirstNumber -Path
L:\VirtualMachines\$VMName$VMFirstNumber\$VMName$VMFirstNumber.VHDX
    start-vm $VMName$VMFirstNumber

    #Increment the counter
    $VMFirstNumber++

}

until ($VMFirstNumber -gt $VMTotal)

```

#Upon completion, you should now have deployed SERVER01-04, each with 2 vCPUs, 16GB RAM, and a secondary drive of 150GB attached.

Appendix 7 – SQL Best Practices for the Application

- 1) Have separate physical SSD drives dedicated for the MDFs and LDFs—due to the difference in the nature of the data access: Random vs Sequential. SSD is required for speed as we do many bulk copy operations.
- 2) Format these drives with a 64K block size, for improved speed for IO operations.
- 3) Most of our customers will not need any SQL recovery mode more than simple due to the nature of the application, so that is our recommendation.
- 4) Take a year's worth of estimated growth and use that as a growth factor.
- 5) Set the Max Server memory, so SQL does not starve the OS.

```
sp_configure 'max server memory (MB)', <Memory in MB>
```

Server Memory for a SQL Server Instance =

(Total RAM available to the OS) –

{
(Memory needed by Operating System which gets allocated to memory pool, filesystem cache, PTE, desktop heap, Driver Images, etc.) +

(Memory needed by Non-buffer Pool region of SQL Server which gets allocated to Multi Page Allocators, Worker Threads, COM, Extended SPs, Backup Buffers, CLR, Linked Server) +

(Memory required for SQL Server Agent, Replication Agents, Bulk Copy, SSRS, SSAS, SSIS, and Full Text) +

(Memory required for Log shipping file copy depending on the size of log backups, if LS is configured) +

(Memory required for other SQL Server instances running in the box) +

(Memory required for other applications running in the box: Antivirus, Monitoring software, Compression software, etc.)
}

6) Set MAXDOP to number of cores – 2; max of 8.

```
EXEC sp_configure 'show advanced options', 1;
GO
RECONFIGURE WITH OVERRIDE;
GO
EXEC sp_configure 'max degree of parallelism', 8;
GO
RECONFIGURE WITH OVERRIDE;
GO
```