VMWARE HORIZON SOLUTION REFERENCE IMPLEMENTATION CASE STUDY FOR RETAIL FINANCIAL SERVICES

Including Reference Architecture for Geographically Remote Sites in a Multi-Data Center Implementation
# Table of Contents

About VMware Reference Implementation Case Studies .................................................. 7
Executive Summary ........................................................................................................... 8
Getting Started: Team Leads and Managers .................................................................... 10
Collecting Metrics That Matter: Engineers ....................................................................... 11
Different Approaches to Capacity Planning ..................................................................... 11
Utilizing a Local or Regional Professional Services Consultant ...................................... 11
Utilizing a Third-Party Application .................................................................................. 11
Utilizing Existing Monitoring Tools ................................................................................. 11
The Do I.T. Yourself Paradigm ......................................................................................... 11
Manually Gathered Performance Counters ....................................................................... 12
Analyzing and Forecasting Needs: Engineers ................................................................. 15
Compiled Structured Data ............................................................................................... 15
Understood the Peaks and Averages ................................................................................ 16
Capacity Planning ........................................................................................................... 16
Summary ......................................................................................................................... 16
Evaluating Solutions: Engineers and Team Leads ........................................................... 17
Application Compatibility ............................................................................................... 17
Application Supportability .............................................................................................. 17
Protocols ......................................................................................................................... 17
Video and Multimedia Performance .................................................................................. 17
Streaming Audio and Video ............................................................................................. 17
Hypervisor Selection ........................................................................................................ 18
The Toaster Test™ ............................................................................................................ 18
Performance .................................................................................................................... 18
Ease of Use ..................................................................................................................... 18
Cost ................................................................................................................................. 18
Private vs. Public Clouds .................................................................................................. 18
VDI Client Selection ....................................................................................................... 19
Choosing a Client Platform .............................................................................................. 19
Serial USB and Network Peripherals ............................................................................... 19
Mobility ........................................................................................................................... 19
Local and Remote Data Protection .................................................................................... 19
Shared Storage ................................................................................................................ 20
Spinning Drives and RAID .............................................................................................. 20
Hybrid Arrays .................................................................................................................. 20
All-Flash Arrays .............................................................................................................. 20
SAN Accelerators ........................................................................................................... 20
Hyper-Converged Devices .............................................................................................. 21
Server Powered SANs ..................................................................................................... 21
Host Servers .................................................................................................................... 23
Monolithic vs. Blade Servers ......................................................................................... 23
Sockets vs. Cores vs. Clock-Speed ................................................................................. 23
Failure Domains ............................................................................................................. 24
Memory .......................................................................................................................... 25
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Storage</td>
<td>25</td>
</tr>
<tr>
<td>Out of Band Management</td>
<td>28</td>
</tr>
<tr>
<td>Vendor Warranty and Support Strategy</td>
<td>28</td>
</tr>
<tr>
<td>Networking</td>
<td>28</td>
</tr>
<tr>
<td>Virtual Networking</td>
<td>28</td>
</tr>
<tr>
<td>Host Networking</td>
<td>29</td>
</tr>
<tr>
<td>Layer-2 Switching</td>
<td>29</td>
</tr>
<tr>
<td>Layer-3 Routing</td>
<td>29</td>
</tr>
<tr>
<td>Virtual Local Area Networks (VLANs)</td>
<td>30</td>
</tr>
<tr>
<td>Virtual Routing and Forwarding (VRFs)</td>
<td>30</td>
</tr>
<tr>
<td>WAN Acceleration</td>
<td>31</td>
</tr>
<tr>
<td>WAN Redundancy</td>
<td>31</td>
</tr>
<tr>
<td>Licensing</td>
<td>32</td>
</tr>
<tr>
<td>VMware</td>
<td>32</td>
</tr>
<tr>
<td>Microsoft</td>
<td>32</td>
</tr>
<tr>
<td>Security</td>
<td>32</td>
</tr>
<tr>
<td>Two-Factor Authentication</td>
<td>32</td>
</tr>
<tr>
<td>Firewalls</td>
<td>32</td>
</tr>
<tr>
<td>Administration</td>
<td>32</td>
</tr>
<tr>
<td>Selling the Solution: Managers</td>
<td>33</td>
</tr>
<tr>
<td>Architecture Overview: Engineers</td>
<td>35</td>
</tr>
<tr>
<td>Branch Office Topology</td>
<td>35</td>
</tr>
<tr>
<td>Data Center Topology</td>
<td>36</td>
</tr>
<tr>
<td>Network Topology</td>
<td>37</td>
</tr>
<tr>
<td>Core Switching</td>
<td>37</td>
</tr>
<tr>
<td>Core Routing</td>
<td>38</td>
</tr>
<tr>
<td>Business Continuity</td>
<td>39</td>
</tr>
<tr>
<td>Lessons Learned</td>
<td>40</td>
</tr>
<tr>
<td>Conclusions</td>
<td>40</td>
</tr>
<tr>
<td>About the Author</td>
<td>41</td>
</tr>
<tr>
<td>About VMware</td>
<td>42</td>
</tr>
<tr>
<td>References</td>
<td>42</td>
</tr>
<tr>
<td>Appendixes</td>
<td>43</td>
</tr>
<tr>
<td>Appendix A: Analysis and Forecasting Worksheet Example</td>
<td>43</td>
</tr>
<tr>
<td>Appendix B: Reference Configurations</td>
<td>45</td>
</tr>
<tr>
<td>Cisco 802.3ad Link Aggregation Control Protocol (LACP)</td>
<td>45</td>
</tr>
<tr>
<td>Cisco Nexus Switch Virtual Port Channels</td>
<td>46</td>
</tr>
<tr>
<td>Cisco Quality of Service (QoS) Framework</td>
<td>48</td>
</tr>
<tr>
<td>Cisco Virtual Route Forwarding</td>
<td>52</td>
</tr>
<tr>
<td>Cisco Dynamic Multipoint VPN Tunnels (DMVPN)</td>
<td>53</td>
</tr>
<tr>
<td>Cisco Layer 2 Tunneling Protocol Version 3 (Pseudo-Wire)</td>
<td>66</td>
</tr>
<tr>
<td>Appendix C: Bill of Materials</td>
<td>68</td>
</tr>
<tr>
<td>Branch Access Switches</td>
<td>68</td>
</tr>
<tr>
<td>Branch Out of Band Management Consoles</td>
<td>68</td>
</tr>
<tr>
<td>Branch Routers</td>
<td>68</td>
</tr>
<tr>
<td>Branch Host Servers: Infrastructure Class</td>
<td>69</td>
</tr>
<tr>
<td>Branch Host Servers: Domain Class</td>
<td>70</td>
</tr>
<tr>
<td>Data Center Core Distribution Switches</td>
<td>71</td>
</tr>
</tbody>
</table>
Legal Disclaimer
This white paper is for information and illustrative purposes only and is provided “as is” with no warranties whatsoever. It is not, and should not be regarded as advice or as a recommendation regarding any particular technology or course of action. Opinions expressed herein are current opinions as of the date appearing in this material only and are subject to change without notice. Reasonable people may disagree about the opinions expressed herein. In the event any of the assumptions used herein do not prove to be true, results are likely to vary substantially. This information is provided with the understanding that with respect to the material provided herein, that you will make your own independent decision with respect to any course of action in connection herewith and as to whether such course of action is appropriate or proper based on your own judgment, and that you are capable of understanding and assessing the merits of a course of action. You may not rely on the statements contained herein. VMware, Northrim Bank, and their employees, officers, shareholders, directors, vendors, sub-contractors, and affiliated parties make no representations or warranties of any kind as to the white paper’s completeness, accuracy, reliability, and compliance with applicable laws, nor will they or any of the parties involved in creating, producing or delivering this whitepaper be liable for any damages, lost opportunities, lost profit, or injury whatsoever — including but not limited to damages arising out of, related to or resulting from reading this white paper, or reliance on any information provided herein, to include any errors or omissions. This white paper references many vendors; it is not meant to endorse or promote products, but rather illustrate the solutions Northrim Bank used to overcome the specific challenges unique to their industry, geography, and environment. You should consult your advisors with respect to these areas. This white paper was written by a VMware customer, Northrim Bank’s Chief Information Officer: Benjamin Craig. Its intended audience includes: Engineers, Project Leads, IT Managers, and C-Level Management. As such, many of the topics discussed here begin with foundational knowledge and build up to complex concepts — apologies in advance to all Engineers for not just getting to the point. The musings, opinions, and poor attempts at humor contained herein are solely those of the author. For instance, if in the course of this paper you read that the author hates the VMware vSphere® Web Client with a vitriolic passion and firmly believes VMware should bring back the C# client, that would be his opinion, and not the opinion of VMware. By continuing to read this white paper, you acknowledge your understanding and acceptance of the foregoing.
Introduction
Northrim Bank is a geographically diverse community bank serving 90 percent of Alaska’s population with branches and offices from Fairbanks, Alaska to Bellevue, Washington. It is one of three publicly traded, SEC registered, publicly owned companies based in the state of Alaska, and supports a total of 34 locations to include those of their wholly owned subsidiary, Residential Mortgage. Between the two companies, Northrim’s IT Department supports approximately 400 employees serving more than 100,000 customers: businesses, professionals, and individual Alaskans who value personal service and competitive products.

Northrim Bank believes that customers deserve the very highest level of service. VMware Horizon® is helping the bank meet its “Customer First Service” philosophy by enabling workforce mobility between headquarters and branch offices, and serving as a critical component of the Bank’s secure and highly-available infrastructure. Horizon helps Northrim Bank reduce costs, stay in compliance, and enable always-on access to applications for business-critical processes performed by management, bank tellers, and loan originators.

About VMware Reference Implementation Case Studies
A reference implementation case study shows how specific customers in a variety of locations and industry verticals have deployed and benefited from VMware Horizon. A reference implementation describes in detail the project approach, architecture, and business benefits at a real customer site, and provides lessons learned for that particular deployment. These implementations are often built on reference architectures validated by VMware.

While a reference implementation is often built on best practices, sometimes tradeoffs are made to meet project requirements or constraints. When a reader references these implementations, it is critical to understand where the architecture has deviated from best practices, and where improvements can be made. This is easily achieved by comparing a reference implementation to reference architecture documentation provided by VMware. For further information regarding this and other technical reference architectures, visit VMware Horizon Design resources on the VMware Web site.

This case study is intended to help customers—IT architects, consultants, and administrators—in the early phases of planning, design, and deployment of VMware Horizon solutions. The purpose is to provide an example of a successful architecture that represents specific industry vertical challenges met, and the benefits gained from the solution.
Executive Summary

Northrim Bank (NASDAQ: NRIM) was founded in 1990 out of the ashes of the late 1980’s recession. Its first branch was housed in a construction trailer in the parking lot of the building that now serves as its Corporate Headquarters. At that time, employees pioneered the concept of BYOC (Bring Your Own Computing) by bringing their home desktop computers to work each day because the bank only had a couple of workstations.

Less than ten years later Northrim Bank more than doubled in size by acquiring the eight Alaska branches of Bank of America. In the years to come, Northrim continued to invest in the communities that it serves: acquiring other banks and expanding their portfolio of services, ultimately growing to well over a billion dollars in assets. By 2009 Northrim’s aging technology infrastructure was due for a major overhaul, prompting the Bank’s leadership to initiate a three year project to upgrade its security, network, server, storage, and client computing infrastructure. That project was completed on time and under budget just two short months before Northrim faced its first real-world disaster event in 2012: a core banking system failure requiring implementation of emergency business continuity protocols, failing over services to its brand new data center in Fairbanks from the primary data center in Anchorage.

Thanks in large part to VMware, the bank was able to recover its core banking system 360 miles away in less than fifteen minutes with less than three minutes of manually-validated processes resulting in zero lost transactions. This successful real-world business continuity event was a textbook example of the power of private cloud computing, but it was not without lessons learned. The day after the disaster, the branches worked flawlessly throughout the day, but as with all financial institutions the real work happens at night when the core Batch Operators process all the calculations that roll one business day into the next. That night Northrim discovered how heavily dependent its core database is on low-latency connections to the batch workstations. Fortunately, the IT Department was already testing virtual workstations and was able to move the Batch Operator’s workstations to the secondary data center within a few minutes. The Bank ran flawlessly out of their secondary data center for a week before returning operations to their primary data center during a fifteen minute planned maintenance window.

A scant six months later, the Bank encountered its second business continuity event: a massive wind storm in Anchorage, coupled with a water main break (thus disabling the water-cooled backup generators at the corporate office) that took out power for well over six hours during critical batch processing. However, applying lessons learned from their first event, the IT Department had already virtualized all of the most mission-critical workstations in the organization — batch operations were still running with no interruption at the primary data center, and the Batch Operators were able to access them from one of the branch locations in Anchorage that still had power. The next morning, Northrim was one of the few Anchorage banks with branches that opened on time.

In 2014 Northrim bought Alaska Pacific Bank and with it five branch locations in Sitka, Ketchikan, and Juneau. To make the acquisition as seamless as possible, Northrim’s IT Department deployed virtual desktops at these new branches in southeast Alaska. This allowed the former Alaska Pacific Bank employees to continue using their legacy physical workstations while training on Northrim’s new systems from the first day of the acquisition until they were converted to Northrim’s core systems six months later. Leveraging virtual desktops in the branches allowed for a seamless migration, doubled performance at the teller line, improved security, simplified remote support, and extended business continuity from the corporate office to the branch.

With a little help from VMware, by its 25th Anniversary, Northrim and its subsidiaries total more than 400 employees and 35 locations spanning well over 2,000 miles.
About VMware Horizon: Overview

VMware Horizon is the VMware virtual desktop infrastructure (VDI) solution; Horizon is designed to make VDI easier to manage and control while delivering a robust end-user experience across a myriad of devices and networks. The VMware Horizon solution has tangibly helped some organizations automate desktop and application management, reduce costs, and increase data security by centralizing their desktop environment. This centralization can result in greater end-user freedom and increased control for IT organizations. By encapsulating the operating systems, applications, and user data into isolated layers, IT resources have used Horizon to deliver dynamic, elastic desktop cloud services such as applications, unified communications, and 3D graphics for real-world productivity and greater business agility.

Unlike other desktop virtualization products, VMware Horizon is built on and tightly integrated with VMware vSphere®, the industry-leading virtualization platform, allowing existing VMware customers to extend the value of their virtual infrastructure and its enterprise-class features such as high availability, disaster recovery, and business continuity.

For additional details and features available in VMware Horizon, see the release notes. Typical VMware Horizon deployments consist of several common components, shown in Figure 1, which represent a typical architecture.

![Figure 1: Typical Horizon Components](image-url)
Getting Started: Team Leads and Managers

“Where to start?” These three words are spoken out loud or silently at the beginning of every major implementation. In most cases the answer lies with understanding what drives your organization’s business needs. Most organizations have a need to improve some aspect of their technology, to include: security, performance, high-availability, ease of use, efficiency, functionality, or disaster recovery/business continuity response. A properly implemented virtual desktop infrastructure can frequently improve any combination of these areas. The key is to find what combination of needs motivate your leadership, and build a proposal that solves their perceived problems. At Northrim Bank, our virtual desktop initiative was motivated by enhanced security, performance, and business continuity capabilities.

We used the following steps to get started:

1. **We based the infrastructure design on empirical data.** When we asked for resources, we provided factual and measurable data that supported our conclusions.

2. **We factored the organization’s tolerance for business risk into our infrastructure design.** For example, we allowed that commoditized private clouds might be well suited for organizations that can’t or choose not to maintain the technical talent to administer a private-cloud or local server infrastructure. But, because we are in a highly secure industry, we knew a “global” public-cloud offering was not right for us. (U.S. banks and credit unions are directed by regulators to not use public-cloud services that could be under foreign control.)

3. **We performed a sanity check on the technical solution.** At least every other year we send our engineers and managers to VMworld®, giving them exposure to each vendor’s solution contemporaneously — not a single year has passed where that investment failed to pay for itself.

   **Tip:** For those who have never attended a VMworld conference, it’s easy to fill up your Schedule Builder with back-to-back courses — don’t do that. Most of the courses are available online a few weeks after the European VMworld (at least by November), so if you miss any courses because they are full or you have scheduling conflicts, you can usually catch them online later. Instead focus on attending the things you can’t get online later: Ask-the-Expert Sessions, peer networking, and the Solutions Exchange are all great places to validate your technical solution.

4. **We presented a unified plan and budget in a way leadership could understand.** We tied our technology initiatives to our organization’s goals, clearly illustrating how we could contribute to security, efficiency, profitability, and growth.

5. **We implemented in phases.** We work in IT: We know our staff will have plenty of opportunities to work late nights, sacrifice personal time, and dine on vending machine food: they don’t need any extra practice. When building tactical and budget plans, we always tried to convert Engineer-Time (the theoretical time it would take us to perform a task if completely left to it) to Real-World-Time (the reality of unforeseen calamities, poorly communicated surprise projects, and frequent visits from the good-idea-fairy).
Collecting Metrics That Matter: Engineers

Different Approaches to Capacity Planning
There are several schools of thought when it comes to collecting the data needed to correctly design, size, and scope a virtual desktop infrastructure deployment. Regardless of what storage, server, or network vendor we chose, whether we deployed into a private or public cloud, whether we elected to hyper-converge our infrastructure or opt to use a hybrid combination of local or shared storage, we felt it would be a really good idea to understand how our servers and workstations actually perform. Knowing these metrics allowed us to fairly compare different vendors’ solutions, ensuring they met our current and forecasted needs. Here are four popular paradigms we evaluated before collecting the data we needed for such an analysis:

Utilizing a Local or Regional Professional Services Consultant
Organizations that have no full-time technical staff and choose to outsource their Information Technology functions generally opt to stick with the providers they know and hold accountable on a regular basis. VMware itself has a Lighthouse Program that connects customers with VMware Professional Services Organization.

Utilizing a Third-Party Application
Organizations that have technical staff who understand processor, memory, storage, and network capacity planning, but either don’t have the time to perform a manual analysis, or have economies of scale requiring frequent analysis, often consider using an application capable of measuring current capacity and forecasting future growth.

Utilizing Existing Monitoring Tools
Organizations that have already embraced virtual infrastructure and/or shared storage have a significant advantage when performing capacity planning because those technologies have existing monitoring and diagnostic tools built into the hypervisor and storage arrays that report on processor, memory, network and storage utilization. For example, VMware vSphere has the native ability to report on performance metrics both across aggregated infrastructure and individual virtual machines, giving us all the metrics needed to perform an analysis.

The Do I.T. Yourself Paradigm
Organizations that are not already virtualized, who have a finite number of servers (fewer than 250) and workstation “classes” (fewer than 25, where a “class” is defined as a group of workstations that generally run identical applications and workloads: e.g. IT, Accounting, or Marketing workstations) may find it advantageous to perform their own analysis either to save on cost or to better understand their infrastructure. This type of analysis requires technical staff who can use Windows Performance Monitor or Linux TOP commands to collect data that can be later imported into a spreadsheet and analyzed. This is the methodology we initially used at Northrim Bank, and because it illustrates the logic and calculus behind capacity planning, the following section will explain it in detail.
Manually Gathered Performance Counters

To ensure the most accurate results, we performed the following procedure from the same workstation, at the same time, for the same duration — during a regularly reoccurring peak processing time (mid-morning during a month’s end cycle). For servers, this procedure was run on every Server that accessed shared infrastructure. For workstations, this procedure was run on two workstations of each “class” (see above) and the results were averaged per class.

**Note**: The following procedures apply to a Windows Server, for Linux we used TOP.

1. Launched Performance Monitor (Perfmon.exe) from a dedicated workstation with at least a 1Gb network connection to our target servers or workstations. This ensured we didn't inadvertently add overhead to the servers or workstations that were being monitored.

![Figure 2: Creating a New Data Collector Set Within Performance Monitor](image)

3. Within the “Create new Data Collector Set” dialog box: gave the Data Collector Set a meaningful name (Virtualization Capacity Stats) and clicked Next. Accepted the default location for the data to be saved instead of choosing a different location and clicked Finish.
4. Within Performance Monitor: expanded Data Collector Sets > expanded User Defined > right-clicked on Virtualization Capacity Stats (or whatever we called them in Step 3 above), selected New > Data Collector.

![Figure 3: Creating a New Data Collector Within a Data Collector Set](image)

5. Within the “Create new Data Collector” dialog box: gave the Data Collector a meaningful name (ServerXX) and ensured the default Data Collector Type “Performance Counter Data Collector” was selected, then clicked Next.
6. Within the “Create new Data Collector” dialog box: clicked on the Add… button to start creating performance counters.

![Figure 4: Adding Performance Counters to a New Data Collector](image)

7. Added the Performance Counters from the list below using the default “Instances of Selected Object.”

   **Note**: We browsed for the target server or workstation that we wanted to collect counters from, otherwise it would have pulled counters from the local machine we were currently using!

   When finished, we clicked OK.
   a. Processor: % Processor Time
   b. Memory: % Committed Bytes In Use
   c. Physical Disk: Disk Read Bytes/sec
   d. Physical Disk: Disk Reads/sec
   e. Physical Disk: Disk Write Bytes/sec
   f. Physical Disk: Avg. Disk Write Queue Length
   g. Physical Disk: Avg. Disk Read Queue Length
   h. Network Adapter: Bytes Received/sec
   i. Network Adapter: Bytes Sent/sec
   j. Network Adapter: Packets Received/sec
   k. Network Adapter: Packets Sent/sec

   **Note**: For servers and workstations with multiple disks (e.g. separated operating system, database, transaction logs, etc.) we made a note of what each disk was being used for.

8. Once all Counters were added, we clicked OK to return to the “Create New Data Collector” dialog box, accepted the default **Sample Interval** value of 15 seconds, and selected Next.

9. Checked the “Open Properties for this Data Collector” checkbox and clicked Finish. The Properties for the data collector we just created were displayed.
10. Within the “[Data Collector] Properties” dialog box, changed the Log format to “Comma Separated” and clicked OK.

![Figure 5: Configuring Data Collector Properties](image)

11. Repeated Steps 4 – 10 for each server and workstation we were monitoring.

12. Within Performance Monitor: expanded Data Collector Sets > expanded User Defined > right-clicked on Virtualization Capacity Stats (or whatever we called them in Step 3 above) and selected Properties.

13. Within the Data Collector Set Properties dialog:
   - On the Schedule tab: clicked Add. Chose the Active Range Beginning and Expiration Date. (We ran this at peak processing time: three days at the end of our business quarter starting on a Wednesday at midnight, ending Friday at 11:59pm.)
   - **Note**: The files generated over a 24hr period were fairly large, we found that a 72 or 96 hour report taken during peak processing times yielded the most insight for us.
   - On the Stop Condition tab: checked the Overall Duration checkbox and chose when we wanted to stop the analysis. (Using the example above, we selected “3 Days.”) When finished we clicked OK.

![Figure 6: Configuring Data Collector Set Monitoring Start and Stop Times](image)

14. After validating the performance counter log files were created in the folder in step 3 above, we waited for the schedule to complete, then retrieved the output file.
Analyzing and Forecasting Needs: Engineers

A consultant can make a pretty good living off of sizing and scoping virtual infrastructure, and there are undoubtedly many patented methodologies that shroud server, network, and storage assessments in mystery: delivering reports intended to drive purchases. Being a curious fellow, I’ve always aspired to understand the math behind the numbers — so with the help of some really smart people and a few beverages of choice, we developed a forecasting spreadsheet that has proven effective and accurate for our needs over the last fourteen years...

Compiled Structured Data

We started by dumping all the metrics collected from the previous Section into a spreadsheet, whether they were obtained from existing hypervisors and shared storage, or manually collected from performance monitoring tools.

It helped that the data was already formatted using Comma Separated Values (CSV), so we could use a wizard to import the data into unique fields as seen below.

<table>
<thead>
<tr>
<th>Time</th>
<th>Memory % Bytes In Use</th>
<th>Disk Reads/sec</th>
<th>Disk Read Bytes/sec</th>
<th>Disk Writes/sec</th>
<th>Disk Write Bytes/sec</th>
<th>% Processor Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>3/12/08 4:42 PM</td>
<td>37.57993680</td>
<td>1.1685959009</td>
<td>4914.624673</td>
<td>24.79709714</td>
<td>692554.9148</td>
<td>10.625</td>
</tr>
<tr>
<td>3/12/08 4:43 PM</td>
<td>37.57813421</td>
<td>1.266516883</td>
<td>5187.652339</td>
<td>4.599455323</td>
<td>105996.0284</td>
<td>1.25</td>
</tr>
<tr>
<td>3/12/08 4:43 PM</td>
<td>37.57973994</td>
<td>0.133317775</td>
<td>546.0696066</td>
<td>2.066425513</td>
<td>3208.158939</td>
<td>2.760416667</td>
</tr>
</tbody>
</table>

Figure 7: Formatting Imported Performance Data into a Spreadsheet for Analysis

**Note:** By collecting performance monitoring data every fifteen seconds over a period of three days, we had a lot of rows in our spreadsheet. For this reason we used separate tabs for each server’s performance metrics.

Now that the data was structured, we determined four relevant figures for each column of data:

- **average workload**: obviously higher than mean, but relevant for baseline performance
- **high-percentile workload**: accounted for the majority of performance spikes
- **peak workload**: relevant for boot-storms, peak-processing times, etc.
- **graphical representation of workload**: visual representation of performance trends

Rather than size for a peak workload, we (used to) size based on the 95th percentile average. The reason for this was simple: we didn’t have access to unlimited financial resources, and sizing for peak workload was like using a Bugatti Veyron as a daily driver. (Sure it will get you there fast: but at eight miles per gallon and a $1.7M price-tag, it’s not the most practical commuter car.) So, to build infrastructure that could handle most of the workload spikes, rather than use the Peak or Average, we used an average based on 95% of the recorded data, leaving the remaining 5% as outliers. As the cost of memory, processing power, solid-state storage, and NVRAM have dropped while their capacity increases, our calculus for deriving the optimal performance metrics has changed: Today we generally size for a 97th percentile average on servers and a 98th percentile average for virtual desktop infrastructure — because its cost effective and why be slow if you don’t have to?

Charting these metrics in a spreadsheet not only helped us visualize thousands of metrics at once, it also allowed us to see when and how long peak workloads lasted to determine if the peaks need to be accounted for in our design.
Understood the Peaks and Averages
The following figure illustrates the deltas between these values. This figure depicts the “Total Disk Writes per Second” in pink and the “Total Disk Reads per Second” in blue (together they would equal the total Input-Output per Second or IOPS).

![Figure 8: Calculating Average, High-Percentile Average, and Peak Needs](image)

Note the following conclusions for the Disk Writes per Second (pink) values:

- **Average Writes per Second:** 111
- **97th Percentile Average Writes per Second (excluding 3% of outlying results):** 267
- **98th Percentile Average Writes per Second (excluding 2% of outlying results):** 1172
- **Peak Writes per Second:** 2143

The reason we got away with using the High-Percentile Averages (97%–98%) instead of Cumulative Peak Workloads (100%) is because not all of our virtual machines consume the same resources at the same time; therefore, there was no need to over-provision infrastructure to accommodate Cumulative Peak Workloads. In case you’re wondering, the reason our workstations got 98% while our servers only got 97% is because our virtual desktop operating systems tend to have more performance spikes, resulting in perceived impact to end-users, and are more frequently rebooted and taxed than our server operating systems.

Capacity Planning
The metrics we collected in the previous section were also used to calculate the Read/Write Ratio and the consumption rate over time, allowing us to project storage growth and requirements for business continuity. For instance, because we knew exactly how much storage space we were currently consuming, by taking another data sample one month later, we were able to determine our monthly consumption rate, then multiply that by the number of months we wanted to forecast. And because we had the total Bytes Written per Second, we could calculate our daily change rate which helped us determine how much network throughput was required to replicate our data from one site to another, and how much space was required for SAN snapshots and replicas to protect our data.

Summary
These calculations gave us empirical data that we applied to every server, network, and storage vendors’ solution to make sure we were properly sized. They also gave us accurate numbers needed to quantifiably justify our expenditures and plan for our future investments. An example worksheet containing the steps and calculations referenced herein is provided as Appendix A of this reference guide.
Evaluating Solutions: Engineers and Team Leads

After compiling an analysis of compute, network, and storage needs, the next step was to get familiar with the technology available at the time. We realized that whatever we knew by heart as of two years ago subtly changed thanks to the constant evolution of progress and the bane of all technologists: marketing departments who like to frequently change product names. As of Q2 2016 here were some of the several technical considerations we evaluated for our virtual desktop infrastructure deployment:

Application Compatibility

Before we did anything else, we ran a pilot program. It cost us nothing and took little time to load up a virtual machine with our standard desktop image and use the native remote desktop functionality built into our guest OS (e.g. Microsoft Remote Desktop Protocol). This allowed us to test all our organization’s proprietary and specialized software up front. While evaluating our applications, we took the following into consideration:

Application Supportability

Most reputable commercial software developers support their applications in a virtual environment. However, when dealing with proprietary industry-specific software (banking, healthcare, etc.), we expected to hit a roadblock when we asked if the software vendor will support running their application in a virtual environment. We reached an agreement with our software vendors allowing us to take ownership for reproducing any errors on physical hardware if we mutually suspected the error was related to our virtual infrastructure. That said, in over six years of running all our banking software in a virtual environment, we have never had to actually reproduce an error on a physical server or workstation. Looking to the future, we asked three large core-banking providers (Fidelity Information Systems (FIS), Jack Henry & Associates, and Fiserv) to support their applications in a virtual infrastructure.

Protocols

Not all remote access protocols are equal. We found that some applications refused to run in Microsoft RDP or Remote FX, but worked fine using Citrix HDX, Teradici PCoIP and the new VMware Blast protocol. In order for us to truly understand how our applications performed, we tested all available protocols over a variety of network connections during our pilot phase. We are currently running Teradici PCoIP.

Video and Multimedia Performance

Multimedia works in our virtual environment, but it can be choppy in comparison to a bare-metal installation. The built in Horizon video functionality has proven fully sufficient for our organization. But if our users should ever need enhanced multimedia we would consider using a vGPU like the NVIDIA GRID to improve their graphics experience.

Streaming Audio and Video

We use webcams, mics, and speakers/headsets within our virtual machines. The audio is really good, but the video quality is understandably not quite as good as using a bare-metal installation. We have been very successfully using the following audio/video communication software within our virtual machines across the organization for multiple years now: (Cisco WebEx, Cisco VoIP Soft Phone Client, Cisco Jabber, Microsoft Skype, Slack).

Note: Because our organization uses VoIP softphones, VDI was a great way to enhance our DR/BC plan. We successfully use Cisco VoIP Softphones running in our virtual desktops to allow our employees to seamlessly communicate with internal and external callers from remote locations. Many of our telecommuters only use a softphone running inside their VM as their primary line of communication.
Hypervisor Selection

There are three mainstream virtual desktop infrastructure providers today: VMware, Citrix, and Microsoft. In our testing we found each had its own strengths and weaknesses that influenced our selection. Based on the fact this is a VMware Horizon Resource, you can guess which hypervisor we selected. That said, rather than rely solely on third-party analysis, we performed a comprehensive in-house comparison between Citrix XenDesktop and VMware Horizon using the criteria below (Microsoft’s Remote FX was still in its infancy at the time we performed our analysis):

The Toaster Test™

If given a loaf of bread and a toaster, we feel we should be able to make toast without having to read a manual. Likewise, if given the installation files, demo licenses, and a high-level implementation guide, we feel we should be able to install a product without having to call support. Between my Systems & Network Manager and myself, we were able to get both XenDesktop and View working, but the experience was very different between the two platforms...

The Citrix offering took far more steps to install, there were multiple interdependencies on add-on modules, the installation required some obscure command-line configuration, the implementation guide was vague and we found ourselves having to use Google several times to get past some of the installation steps for a basic deployment.

VMware Horizon took no time to implement: it took far fewer steps, had excellent documentation, and was simple to integrate with existing infrastructure. It was equally simple to leverage some of the advanced features in Horizon: In an afternoon, we configured the reporting engine, two-factor authentication, and secure Internet-facing remote access in our DMZ on the first try.

Performance

Although there were nuances between the ways each virtual desktop solution rendered sessions from an end-user's perspective, the difference between the two solutions was not significant enough to sway our decision. That said, from an administrator's perspective, it was clear that we were able to support far more (almost 2:1) virtual machines in our VMware lab than our Citrix lab using identical hardware resources.

Ease of Use

At the end of the day, VMware Horizon was clearly easier to install, maintain, support, and troubleshoot in our environment. And since its implementation, both its client-side and administrative interfaces have remained consistently simple and intuitive.

Cost

We believe we could have gotten both vendors to a near identical price point; however, there is a soft-cost to maintaining two disparate virtual infrastructure solutions. And if we had chosen Citrix, we would still need to license some VMware components to support our virtual infrastructure. Because we already had vSphere infrastructure, we didn’t have to train and maintain expertise on a completely different system.

Private vs. Public Clouds

Every vendor’s definition of “The Cloud” is different — here’s mine: Cloud computing is the proper application of networking within a multi-site virtual infrastructure. Private Clouds are hosted by you and Public Clouds are hosted by someone else. In my experience, there is little cost difference between using a private versus a public cloud. With a private cloud you are paying for the capital expense of infrastructure and in-house talent to support your systems, and with a public cloud you are paying a recurring operational expense and for outsourced engineering talent to support your systems. As a bank with high security standards connected to the rest of the United States via transoceanic cables, the decision to use a private cloud was one of the easiest made in a long time. We couldn’t take the chance that our security, availability, or business continuity, would be negatively impacted by a provider over 2000 miles away.
VDI Client Selection
Once we had a handle on whether our applications would function in a virtual environment, we tested our organization’s peripherals for compatibility with our intended clients. As a bank, we have a heavy dependency on proprietary hardware peripherals such as: check scanners/validators, receipt printers, signature pads, driver’s license scanners, cash recyclers, and high-volume document scanners (in addition to the usual keyboard and mouse devices). Like their software counterparts, industry-specific hardware manufacturers have been slow to develop hardware that is optimized for a virtual desktop infrastructure, and frequently require that any errors be reproduced in a physical environment before they will troubleshoot a problem. Here are some items we considered when we selected our VDI clients:

Choosing a Client Platform
We have had consistent success with the VMware Horizon client for Microsoft Windows running on Windows 7 and Windows 10, as well as Apple’s Mac client running on OS X. We have evaluated the Dell® Wyse® thin-clients for some time now, but we have consistently opted for full desktop workstations because our price-point of the thin-clients was too high and the thin-client OS required periodic updating and maintenance, adding to our administrative burden. That said, we have had consistent success with LG and ViewSonic zero-clients. Looking forward to our next desktop refresh in 2018, it is highly probable we will adopt zero-clients across the enterprise.

Serial USB and Network Peripherals
Most of the devices available to banks today support USB, although there are still a few vendors shipping devices with legacy serial connections. Naturally, our preference for device connectivity is always network based — the only caveat being that the network connectivity must work across a WAN, not just within the same broadcast domain. We feel that if the device can’t support a network connection, USB tends to be the next safest bet. When all else failed, VMware Horizon does still support serial connections, but they can be problematic when logging in across multiple desktops.

Mobility
We use a Mobile Device Manager (MDM) to consistently deploy the latest client applications, ensure that mobile clients are not rooted or jailbroken, and assure the client configuration, client-side certificates, and any proprietary settings can be wiped should the phone or tablet become inaccessible. To better understand each MDM’s nuances, we evaluated several leading (at the time) MDM providers, to include: VMware AirWatch® (just prior to the VMware acquisition), Amtel, Good Technology, Maas360, McAfee, MobileIron, SimpleMDM, Sophos, Soti, and Symantec. After ruling out the rest of the pack for various reasons, we chose AirWatch and MobileIron to participate in an on-site proof-of-concept deployment. In the end, we selected MobileIron as our preferred MDM solution based on the fact it was the most stable, feature-rich, and easiest on-premise mobile device manager to work with on a daily basis. We also standardized on Apple iPads and iPhones as our preferred mobile devices. We do not support Android devices because of security concerns.

Local and Remote Data Protection
When designing our virtual server infrastructure, we established Data Protection Tiers (Gold, Silver, Bronze, Iron, Rust, etc.) for both local and remote copies of our virtual infrastructure. For our “Gold Class” Servers, We elected to retain local (snapshot) and remote (replica) copies of our data every five minutes for 25 hours, every hour for four days, every day for a quarter year, and quarterly for eight years. For our “Iron Class” workstations, we elected to retain snapshots and replicas daily, six days a week. We are currently evaluating both Zerto and Veeam for our relatively demanding replication and archival needs.
Shared Storage

Storage Area Networks (SANs) used to only be found in large organizations who had a need for massive pooled storage that could be accessed by multiple servers. Shortly after the turn of the century, VMware became mainstream and introduced the concept of vMotion® (migrating a virtual machine from one host to another), which allowed for all sorts of interesting possibilities. In my opinion, this more than anything led to the evolution of the storage industry over the last decade.

**Types of Shared Storage**

*Spinning Drives and RAID*

In the early 2000s most SANs still used spinning drives (10,000 RPM SAS or 7,200 RPM SATA) and depended on some form of RAID for performance and protection. These SANs typically used dedicated Controllers that often ran either a proprietary or highly customized operating system designed to act as a Target for hardware or software based Initiators coming from servers. These Controllers would often be configured for redundancy and were sometimes integral to the same enclosure that housed the drives (like EqualLogic) and sometimes were independent, controlling multiple shelves of disks (like some NetApp or EMC SANs). Physics dictated that increasing the number of spinning drives would proportionally increase the speed of reads and writes until the Controllers reached their maximum processing capabilities, at which point the Controllers (and frequently the Enclosures) would have to be upgraded.

*Hybrid Arrays*

In time, as Solid-State Drives (SSDs) became less costly and more prolific, they were successfully incorporated into SANs, acting as a layer of cache, often sandwiched between high-speed NVRAM and spinning disks. One of the first SAN vendors to successfully leverage this “hybrid” architecture was Nimble Storage, whose intelligent algorithms and CASL architecture made good use of each layer of storage. By migrating our production data from two EqualLogic PS6510 arrays (48 drives in a RAID 10 set) to a single Nimble CS240G, we saw a 5X increase in performance and 17X reduction in space utilization. Shortly thereafter many SAN Vendors (established and new) started replacing their existing spinning drives with SSDs, claiming their solutions were also part of this hybrid generation, but without significantly optimizing their algorithms for solid-state memory. In my opinion, their claims were technically true; however, it would also be fair to call those SANs “Hybrid” if their SSDs were attached onto the side of the SAN enclosure with wood screws.

*All-Flash Arrays*

Eventually, the cost of high-capacity SSDs started to drop, and most SAN vendors began shipping “all flash” arrays (like Pure Storage). These arrays were screaming fast in comparison to the arrays of the 2000s, but were still dependent on the Controllers and the underlying network protocols used to transmit and receive data: iSCSI, NFS, Fiber Channel, or Fiber Channel over Ethernet.

*SAN Accelerators*

Once most of the performance barriers were removed within the SAN enclosure, manufacturers started to attack the elephant in the room (not Hadoop): Remote storage is always going to be measured in milliseconds while local storage can be measured in microseconds or even nanoseconds. It doesn’t matter how fast your array is, if it is at the other end of a network connection it is slow compared to the speeds attainable on the server itself. To tackle this physics problem, a few clever vendors created SAN Acceleration Software (like Pernix Data and Atlantis Computing) that cached the most frequently used reads (and sometimes writes) on high-speed local storage. Although some of these third-party solutions have write-caching limitations, and are not integrated with the back-end storage (which complicates troubleshooting and upgrading), they have been successfully used to improve storage performance on legacy SANs.
Hyper-Converged Devices

One way storage vendors got around the lack of integration between local storage and the SAN was to merge the host and SAN into a single unit, known as a hyper-converged device. Nutanix, Simplivity, and VMware were some of the first vendors to introduce the concept of a SAN-less (or if you prefer, a virtual SAN) data center. Hyper-converged solutions contain at least three hosts, and protect their data by “striping” (my words) the data across each host to ensure that a single host failure doesn’t destroy the entire Datastore. Although both Nutanix and Simplivity require proprietary hardware, VMware Virtual SAN™ can be installed on most any host that meets their published hardware compatibility requirements. We found hyper-converged solutions can work very well with predictable workloads (think hundreds of identical web-application servers). Conversely, it was a challenge to support elastic workloads that require frequent independent changes in compute and storage resources. While this technology is innovative, we opted against a hyper-converged solution for two reasons: steep linear cost and because our host-side patching, upgrades, and maintenance windows are all too frequent to feel comfortable using our servers as a “distributed SAN.” We recently had a host down for over a week as we worked through a bug in the iDRAC remote access card, if that host were part of a hyper-converged solution the data on that host would have had to be evacuated or rebuilt.

Server Powered SANs

The latest paradigm in the world of shared storage comes from Datrium, and because it is so different, it’s hard to classify — here’s my take on it:

Step 1: Start with a highly reliable traditional network storage appliance.

Step 2: Migrate all the intelligence and functionality typically found in a SAN Controller along with some intense compression and deduplication algorithms to the vSphere Host as a software “Hyperdriver.”

Step 3: Allocate a small portion of the vSphere Host’s processing power and memory to run the Hyperdriver at the hypervisor level.

Step 4: Allow customers to use whatever non-proprietary solid-state storage they want for performance caching (knowing all the writes are safe on the network storage appliance).

From our perspective, Datrium’s solution incorporates both the High-Availability features of traditional network storage and High-Performance features of Host-based caching to create a new genre of SAN. One of the most undersold and awesome features of Datrium’s SAN is that it presents its entire storage capacity as a single NFS Datastore within vSphere (it’s not actually NFS, it just presents that way). So, rather than carve up storage into legacy SAN artifacts like LUNs and Volumes in order to provide granular performance and protection, with Datrium we store all virtual machines in the same logical volume, recouping all the wasted space that would have been allocated for thin-provisioning in each of our legacy volumes. And because all of the persistent writes are protected on the network storage appliance, we don’t worry about the risk of data loss in the event of a host failure. As early adopters of Datrium’s DVX platform, we were left speechless by the fact we were pushing between 2X and 5X the performance, and 4X the compression/deduplication of our production Nimble Storage SANs using only a single Solid-State Drive we bought off Amazon for host-side caching. From a sheer architectural standpoint, we came to the conclusion that Datrium’s “Server Powered SAN” was the best fit for our virtual desktop infrastructure. Here’s why...
Server Powered SANs (continued)

**Performance:** Our Datrium is fast; really fast. Our maximum production Oracle, Microsoft SQL and Exchange workloads could not come close to hitting the Datrium DVX performance ceiling. Because we needed to know where the SAN starts to struggle, we used several benchmarking tools (such as Iometer and Login VSI®) to test both IOPS and throughput. Using “Insane Mode” (Datrium’s maximum performance setting) with only two $400 Samsung 850 1TB solid-state drives in each of our four vSphere test hosts, we were able to push just over 163,000 IOPS using 80% random/75% reads, at a little over 1.3GB/sec with only 2.94ms of latency, using Iometer (See Appendix D). Based on our current production IO spikes and capacity, we are able to easily consolidate two EqualLogic PS6510 arrays (90+ drives in a single RAID 10 Storage Pool), a Nimble Storage® CS-440G-X2, and a Nimble Storage CS-420G-X8 into a single 2U Datrium DVX at each data center with significant capacity and performance to spare.

**Capacity:** About 25% of our data is not compressible because it’s made up of compressed images (.tiff files of loan documents, statements, etc.), the rest of our data is pretty normal (databases, file and applications servers, desktops, etc.). At the time of writing, we observe a little over 7X compression across all data residing on our DVX. We’ve never seen compression this high with any other solution. Even using the most conservative math, we believe we can store at least 100TB on a single Datrium 2U Netshelf. Also, because the Datrium DVX presents its entire capacity to our hosts as a single large NFS Datastore, we don’t have to allocate between 20-40% of unusable “overhead” on each of dozens of Volumes/LUNs to account for thin-provisioning — that is a huge differentiator in comparison to alternative solutions.

**Resiliency:** Hyper-converged solutions are a cool concept, but we couldn’t bring ourselves to store our most valued assets across our compute resources. Don’t get me wrong, I’m a big believer in the software-defined data center. Virtualizing network assets, servers, desktops, and applications makes complete sense, but when we factored in our maintenance windows, software upgrades, and statistical host failures, spanning our data across our hosts didn’t seem as good an idea as placing it in an appliance that was purpose built to provide stability. That said, the notion of using high-speed host-based cache to increase SAN performance is brilliant, and Datrium does just that.

**Cost:** The Datrium DVX gives us ample capacity, high performance, and the sound data protection, but it also has the most affordable performance upgrade path — if we need more performance, we simply purchase more commoditized local storage for our vSphere hosts — in less than five minutes we can increase our performance without having to buy a new Controller or Enclosure.

There are a lot of good storage vendors out there (and a few duds). That said, we would never buy a SAN without kicking the tires first. Every SAN vendor I’ve ever met has offered a demo unit. Some SAN vendors have tried to make us sign an NDA so we can’t talk about their performance, others have agreed to a demo only if we sign a Purchase Order and commitment to “try and buy” — to them, I say “Nuts.” The way I see it, if I have an approved budget and have narrowed down my selection to two vendors, they need to earn my business by putting some skin in the game — that means going the extra mile to bring a SAN on site.
Host Servers
For many, server manufacturer preference is as contentious a topic as religion and politics. We have used Dell, Cisco, and HP servers in the past; however, at the time of writing we just went through this process, and as always, learned some interesting things. We felt that Cisco had one of the most innovative and well-designed compute solutions in the market, but their annual support costs were out of line with our expectations and, like HP, they only sell through a channel partnership. HP has always developed well-designed hardware with performance and scalability in mind, but many of their consumable parts (memory, hard drives, etc.) are proprietary contributing to their relatively high initial cost and very high annual maintenance. Several years back, Dell made a foray into the channel partner space like HP and Cisco, but they also maintained their direct customer relationships, something that I always appreciated. Dell also has been on an innovation spree: recently introducing their new FX2(s) 2U enclosure capable of housing their full-width quad-socket FC830, half-width dual-socket FC630, and quarter-width single-socket FC430 blades, among other interesting things. At the time we made our host server purchase, Dell offered Intel’s new fourth generation E5 processors, putting them in the lead for our data center Host Servers.

Monolithic vs. Blade Servers
We generally found that blade form-factors are still more expensive than their monolithic rack-mounted counterparts, and rack-mounted servers tend to offer more flexibility for Local Storage, networking options, and expansion-cards. We ultimately selected three classes of host servers:

Data Center Enterprise Class: Dell’s R630 1U rack-mounted server because it was large enough to give us the expansion-card and local storage options we were looking for, but small enough to cut our current rack utilization by 50% while increasing our memory and compute resources.

Infrastructure Class: Cisco’s UCS-E160D-M2 blade server that fits in Cisco’s Integrated Service Routers because it allowed us to provision all our critical VoIP infrastructure (virtual instances of our Cisco Unified Communications Manager and our Cisco® Unity Express voicemail) inside our redundant core routers.

Branch Office Domain Class: Dell’s R730xd 2U rack-mounted server because it allowed us to fit up to twelve front-facing 3.5” high-capacity (10TB) drives for surveillance and file-storage, and two 2.5” high-performance drives in the rear flex-bay for operating systems and databases.

Sockets vs. Cores vs. Clock-Speed
Here’s a quiz: Which of the following processors is the best solution for VDI?

- Intel Xeon E5-2603 v3 1.6GHz,15M Cache,6.40GT/s QPI,8 Turbo,8 No HT,6C/6T (85W) Max Mem 1600MHz
- Intel Xeon E5-2667 v4 3.2GHz,25M Cache,9.60GT/s QPI,Turbo,HT,3C/6T (135W) Max Mem 2400MHz [add $1,759.22]
- Intel Xeon E5-2637 v4 3.5GHz,15M Cache,9.60GT/s QPI,Turbo,HT,4C/8T (135W) Max Mem 2400MHz [add $758.88]
- Intel Xeon E5-2698 v4 2.2GHz,50M Cache,9.60GT/s QPI,Turbo,HT,20C/40T (135W) Max Mem 2400MHz [add $3,187.29]
- Intel Xeon E5-2609 v4 1.7GHz,20M Cache,6.4GT/s QPI,8C/8T (85W) Max Mem 1866MHz [add $93.82]
- Intel Xeon E5-2699 v4 2.2GHz,55M Cache,9.60GT/s QPI,Turbo,HT,22C/44T (145W) Max Mem 2400MHz [add $4,153.14]
- Intel Xeon E5-2643 v4 3.4GHz,20M Cache,9.60GT/s QPI,Turbo,HT,6C/12T (135W) Max Mem 2400MHz [add $1,317.69]
- Intel Xeon E5-2690 v4 2.6GHz,35M Cache,9.60GT/s QPI,Turbo,HT,14C/28T (135W) Max Mem 2400MHz [add $1,773.02]
- Intel Xeon E5-2660 v4 2.0GHz,35M Cache,9.60GT/s QPI,Turbo,HT,14C/28T (105W) Max Mem 2400MHz [add $1,076.23]
- Intel Xeon E5-2620 v4 2.10GHz,20M Cache,8.0GT/s QPI,Turbo,HT,8C/16T (85W) Max Mem 2133MHz [add $220.76]
- Intel Xeon E5-2697A v4 2.6GHz,40M Cache,9.60GT/s QPI,Turbo,HT,32C/64T (145W) Max Mem 2400MHz [add $2,704.37]
- Intel Xeon E5-2630Lv 4 1.8GHz,25M Cache,8.0GT/s QPI,Turbo,HT,10C/20T (55W) Max Mem 2133MHz [add $358.74]
- Intel Xeon E5-2697 v4 2.3GHz,45M Cache,9.60GT/s QPI,Turbo,HT,18C/36T (145W) Max Mem 2400MHz [add $2,476.70]
- Intel Xeon E5-2630 v4 2.2GHz,25M Cache,8.0 GT/s QPI,Turbo,HT,10C/20T (85W) Max Mem 2133MHz [add $420.83]

Figure 9: Examples of Processor Choices at Time of Writing
Sockets vs. Cores vs. Clock-Speed (continued)
The answer? It depends, but we felt the 2.3GHz 18 core, 2.2GHz 20 core, and 2.2GHz 22 core processors stood out. Generally, we found that intense processing or heavy writes were better served by a processor with a faster clock-speed. And that concurrent processing (like a virtual workload), worked better with more cores. And if our cumulative cores were overwhelmed and we need more aggregated resources, we should look to more sockets. For active VDI users, we found that an average of five virtual workstations per core seemed to work best for us.

For our Dell R630 Datacenter Enterprise Class Host Servers, we selected dual-socket Intel E5-2698 v4 2.2GHz 20 core 2400MHz processors as they had the highest core-density (we were willing to pay for) to support our virtual desktop infrastructure.

For our Cisco UCS-E160D-M2 Infrastructure Class Host Servers, we selected single-socket Intel E5-2400 2GHz 6 core processors as they were more than enough to support our VoIP virtual machines.

For our Branch Office Domain Class Host Servers we selected dual-socket Intel E5-2630 v4 2.2GHz 10 core 2133MHz processors as they had just enough cores for our virtual Microsoft Active Directory Read-Only Domain Controllers, Network Access Control servers, and Surveillance servers.

Failure Domains
We have found that all hardware fails eventually and when it does, it’s nice to limit the damage to as small a footprint as possible. As previously mentioned, blade enclosures are starting to shrink, which is a very good thing. In the past server vendors would do their best to squeeze as many blades as possible into a single enclosure, then count on the built-in redundancies of the enclosure to provide redundancy. As consumer demand grew for more intelligent and manageable enclosures, so grew the complexity and feature-set of the firmware on those enclosures, increasing the risk of an enclosure failure. Having seen the calamitous results of a 12-blade enclosure failure first hand, I prefer to limit the failure domain to a more manageable four (vSphere) servers per enclosure.

Another consideration we found worth calculating is the number of virtual machines we planned to run on a single host. This decision dictated whether we opted to select a dual-socket or quad-socket configuration and how much memory and storage we needed on that host.

Through trial and error we determined we can comfortably run 120 Virtual Desktops (and a handful of required VDI servers) on a dual-socket host server. If we used quad sockets, we found that we would need to fully populate all the memory slots, which slowed down the system clock speed and created cooling problems on some platforms. We typically run about four non-VDI and four VDI production servers on dual-socket host servers. We allowed for N+1 redundancy in both our VDI and Server clusters, ensuring we have the ability to perform maintenance and absorb unplanned growth.

Because our Voice over IP infrastructure is critical to our business, we utilize two Cisco UCS-E160D-M2 blade host servers at each branch office, one in each of our Cisco Integrated Service Routers, for maximum redundancy.

We only use one Branch Domain Class host server at remote locations because it only supports a Read-Only stub of our Microsoft Active Directory domain, and our surveillance servers have the ability to replicate their digital imagery to our data centers.
Memory

It used to be memory was the one simple thing that we never needed to spend too much time on when configuring a host — we just used the time-tested adage: more is better. While that’s still not the worst approach, we found a few nuances that crept up when we configured a fully populated host.

Take another look at the list of processors in figure 9 on the previous page. Note they advertise both a Processor Speed (1.6-3.5GHz) and a System Clock Frequency (1600MHz, 2133MHz, or 2400MHz). The System Clock Frequency is the speed of the Memory Bus running on the motherboard — think of it as the speed between the memory and CPU, and as you might guess: faster is better. Here’s the catch, in order for us to achieve maximum Memory Bus throughput, we must have:

1. A motherboard capable of supporting the maximum speed, and
2. CPU(s) capable of supporting the maximum speed, and
3. RAM capable of supporting the maximum speed AND
   a. The amount of RAM slots in use will dictate the maximum speed.

Yes, even if we paid extra for the fancy processor, motherboard, and RAM, if we fully populated every slot in the motherboard it will slow the System Clock Frequency down. Let’s use an example of the memory available with the latest 4th generation Intel E-5 Processors:

- 32GB RDIMM, 2133MT/s, Dual Rank, x4 Data Width [$482.23]
- 32GB RDIMM, 2400MT/s, Dual Rank, x4 Data Width [$496.03]
- 64GB LRDIMM, 2133MT/s, Quad Rank, x4 Data Width [$1,310.78]
- 64GB LRDIMM, 2400MT/s, Quad Rank, x4 Data Width [$1,379.09]

Figure 10: Examples of Processor Choices at Time of Writing

Assuming that we selected a processor that is capable of running at a System Clock Frequency of 2400MHz, if we populate all 24 of the server’s slots with 32GB 2400MHz Registered Dual Inline Memory Modules (RDIMMs), the maximum clock speed would only be **1600MHz**. However, if we populated the exact same server with only 12 of the 64GB 2400MHz Load Reduced Dual Inline Memory Modules (LRDIMMs), then the System Clock Speed would still be **2400MHz (but at 3X the price)**. We populated twelve slots of 64GB 2400MHz Quad Rank x4 LRDIMMs in our Datacenter Enterprise Class Host Servers. This allowed us to provision 6GB of RAM for each virtual desktop, averaging 120 desktops per host server, which left us a little under 48GB of RAM to support VMware Horizon Connection Brokers and **VMware vCenter™ servers**. Currently we only use two slots of 32GB 2133MHz RDIMMs in our Branch Office Domain Class and two slots of 8GB RDIMMs in our Infrastructure Class Host Servers, which is plenty for our applications.

Local Storage

We felt Local Storage was one of the most important considerations for our physical hosts because processors, network adapters, and memory have become so fast and ample that even with a full virtual workload they are rarely a limitation. Storage is the last performance bottleneck for virtual environments, and high-speed local storage can be the key to removing that bottleneck. At a minimum we wanted some local storage to install the vSphere hypervisor; such as a local SD Card or internal USB — it didn’t have to be screaming fast. But we also wanted separate high-speed Local Storage to give us the following additional functionality on our hosts:

- **Local Datastores**: There are some situations where it is better to run a virtual machine using local storage, rather than taxing shared storage; a frequent use case is Monitoring Servers because they generate a tremendous amount of IO that would otherwise have to go across the network to a SAN.
Local Storage (continued)

Enhanced Storage Performance: Software-based storage accelerators such as Pernix Data and Atlantis Computing make use of local high-speed Solid State Drives (SSDs) to cache reads and/or writes locally in order to vastly improve performance. We’ve observed these technologies frequently used to extend the life of existing SANs by augmenting their performance, but less commonly deployed in green-field implementations.

Hyper-Converged Storage: Technologies like Virtual SAN, Nutanix, and Simplivity all leverage local storage to build “clusters” that are distributed across three or more vSphere hosts in order to create a redundant pool of storage.

Server Powered Storage: Datrium uses local high-speed storage to cache reads and writes as close to the hypervisor as possible, before compressing and de-duplicating any data sent to their network-based appliance.

When configuring our new servers, we found it best to plan ahead for our future local storage needs up front. We knew we could retrofit our servers with local storage, but we found some manufactures make it more difficult by leaving out key components (such as the drive backplane, cables between the controller and the backplane, and drive caddies) if we didn’t order at least some local storage.

To connect local storage we found we had three options on most server motherboards:

ACHI on-board SATA: This is the built-in SATA controller, typically used for optical drives.

On-board or PCIe RAID: These controllers typically support both SAS (12Gb/sec) and SATA (6Gb/sec) drives and can be used in either Pass-Through HBA mode (presenting the drives directly to the operating system) or varying levels of RAID (aggregating and/or protecting the local drives).

SATA and SAS drives come in two variants: legacy high-capacity spinning drives (up to 10TB) and Solid-State Drives (up to 4TB). SATA is usually half the speed and price of SAS.

Note: We found some server manufactures tried to incent us to only use their branded drives (I’m looking at you HP). Technically, a third-party drive works on our HP servers, but our onboard diagnostics show the drive in an error state and more annoyingly, the on-board cooling fans run at full speed because of that error state, where normally they are at idle. That said, some manufacturers’ (Dell) servers worked with whatever third-party drives we put in them.

PCIe SSD: For high-performance applications, nothing is faster than slapping some solid-state memory directly on a PCIe card, skipping the inefficient legacy SATA and SAS protocols entirely. SanDisk’s proprietary Fusion IO cards were some of the first to take advantage of PCIe SSDs, today a new open standard, NVMe, is becoming popular amongst server manufacturers.

Warning: Science ahead!: How SSDs Work
Not all SSDs are created equal. SSDs write data in units called Pages (made up of multiple Cells), but erase data in Blocks (made up of multiple Pages). Over time some of the Pages in a Block are no longer needed and are flagged as “Stale,” but the entire Block can’t be erased and reused as long as there are still Pages in-use in that Block. If left unchecked, this would lead to a SSD filling up quickly, and not being able to write any more data because there are no more free Blocks. To account for this, drive manufacturers created a process known as Garbage Collection that copies all the In-use Pages out of a Block with Stale Pages into a brand new Block, then once all the Pages in the old Block are no longer in use, the Block can be erased and reused. Some SSD vendors deal with Garbage Collection by overprovisioning the capacity of the drive, but only advertising a fraction of that capacity to the operating system (e.g. creating a 1.3TB drive, that is seen as a 1TB drive); these drives are often advertised as “Enterprise” or “Professional” Class, but because there is no governing standard, those words are just marketing. So, when we selected our drives, we reviewed published benchmarks from reputable third-party sources to compared drive performance.
Local Storage (continued)

Given all this information, we used the following criteria to select the local storage on our VDI hosts:

- We made sure our controller supports the drive type (SAS/SATA) and the number of drives we planned to use. We also made sure the controller supports passing TRIM (SATA) and UNMAP (SAS) commands from the operating system (these help with the Garbage Collection mentioned in the note above).
- We avoided all "spinning disks" in favor of Solid-State Disks (SSDs) because spinning disks are mechanical; physics dictates they will always be slower, and moving parts tend to fail faster than non-moving parts.
- We found many vendors are now selling Read Intensive, Write Intensive, and Mixed-Use Solid-State Drives. Generally speaking, we found Reads are always faster than Writes, so we leaned towards Write-Intensive drives.
- In our testing with multiple controllers, we found two SATA SSDs were significantly faster than one SATA SSD, but three or more SSDs were not appreciably faster (only a single digit percentage improvement) than two SSDs.
- We found SAS SSDs are about twice as fast (and expensive) as SATA SSDs.
- We discovered proprietary PCIe SSD cards like Sandisk’s® Fusion IO are significantly faster and frequently less expensive than even SAS SSDs (because they don’t need to rely on the SATA/SAS standards that were originally created for mechanical spinning disks). But that if we used a proprietary PCIe SSD, we should make sure it’s supported according to the VMware Hardware Compatibility List (HCL).
- We learned the new NVMe PCIe SSD Standard has the benefit of proprietary PCIe SSDs, but is also more likely to be supported by multiple manufacturers in the future. NVMe drives come in two form-factors: internal PCIe cards and 2.5” monolithic units that can be hot-swapped like a traditional drive. At a price point less than a comparable SATA SSD drive, and performance well in excess of any SAS drive available, PCIe SSDs have the added benefit of not requiring a RAID controller.

In all our Host Servers we use dual 16GB internal SD Cards solely dedicated for the vSphere hypervisor, this allows us to keep the hypervisor on a separate medium from any other local storage.

In our Data Center Enterprise Class Host Servers we also use two Intel 2TB P3700 PCIe NVMe drives per host, this ensures that we have the ability to leverage the next generation of high-performance SANs that use local cache.

Whereas in our Branch Office Domain Class Host Servers we use dual 1TB SATA SSDs in RAID 1 for our operating systems and databases, and nine 10TB SATA 7200RPM drives in RAID 5 for high-capacity storage. Most of our Branch Office Domain Class Host Servers are configured to use one 10TB drive as a hot-spare, leaving approximately 70TB of available space in a RAID 5 set, which is just slightly more than the maximum vSphere VMFS Datastore size of 64TB. For surveillance servers that need more space, the Dell 730xd can accommodate up to sixteen 3.5” drives.

Lastly, in our Infrastructure Class Host Servers we used dual 900GB SAS drives in RAID 1 for our hypervisor Datastore.
Out of Band Management
Whether we chose to buy a “white-box” server from somewhere like Server’s Direct or from a mainstream manufacturer, we knew we would want an Out of Band Management Card (such as Dell’s iDRAC or HP’s iLO). At some point we knew we would need to remotely manage the console of our hypervisor host, at times like that it paid dividends to have an alternate means to manage the Host Server. On our Data Center Enterprise Class and Branch Office Domain Class hosts, we selected Dell’s latest iDRAC Enterprise card with 16GB of onboard cache so that we could perform firmware updates without having to boot into a Microsoft Windows operating system. Cisco has similar IMC Out of Band Management functionality on our UCS-E Infrastructure Class host servers. We also felt it would be a very good idea to have a console server like the Lantronix SLC 8000 to allow secure access to our network devices’ consoles.

Vendor Warranty and Support Strategy
Most server manufacturers have graded levels of support, from 24/7 (Gold), to 24 hours a day Monday-Friday (Silver), to 8:00a-5:00p Monday-Friday Next Business Day (Bronze). Because we generally build N+1 redundancy into our virtual infrastructure design, we opt for the “Iron” support package (or if they have it, “Rust”). That said, we tend to extend our hardware support out to at least five years (frequently seven), allowing for a longer refresh cycle in case we have a business reason to not upgrade at the typical three year mark.

Networking
In my opinion, regardless of size, regulated and mission-critical organizations (such as banks, hospitals, and government services) should have at least two data centers for business continuity; beware of anyone who rationalizes otherwise. That said, a secondary data center doesn’t have to be fancy, or even corporeal — it may be more cost effective to run a primary data center in-house and a secondary site using a public cloud provider, or both data centers in-house, or neither. In any case, by virtue of having at least two sites, by (my) definition you have a Cloud, a.k.a.: The proper application of networking within a multi-site virtual infrastructure. Private Clouds are hosted by you and Public Clouds are hosted by someone else. What follows are some of the key networking concepts that we took into consideration when designing our infrastructure to be Cloud ready.

Virtual Networking
Standard vSwitches support VLANs and VLAN Trunking (the segmentation of a single Layer-2 network into multiple broadcast domains), Cisco Discovery Protocol (a feature that allows Cisco devices to identify what’s on the other side of a network link), Jumbo Frames (the ability to send data packets larger than the default Maximum Transmission Unit (MTU) size of 1500 — up to 9000+), and can be configured to work with 802.3ad Link Aggregation Control Protocol (LACP: the ability to aggregate multiple network connections for increased throughput and stateful failover). Other than that, Standard vSwitches are pretty basic. A Standard vSwitch works within one vSphere host only, where a Distributed vSwitch can be used across multiple vSphere hosts, as long as they exist within the same host cluster. Although Distributed vSwitches offer significant features and capabilities, we found that they added a cumbersome layer of complexity to our disaster recovery procedures — specifically, after recovering from a full power loss event, the Distributed vSwitches were unmanageable without our vCenter, Active Directory Domain Controllers, Certificate Servers, Two-factor Authentication Servers, and 802.1x Network Access Control Servers — an unwelcome hindrance to bringing up our infrastructure quickly and with as little drama as possible. That said, we are considering a mixed use of Standard vSwitches for Management, Storage, and mission-critical traffic, and Distributed vSwitches for all other production traffic so that we can leverage the new features of VMware NSX® (see below).
**Distributed vSwitches** support all of the features found in a Standard vSwitch, but also can be accessed via SSH and managed with most if not more of the functionality found in an enterprise-class physical switch. At the time of writing, Distributed vSwitches are only available in the “Enterprise Plus” Edition of vSphere (not Standard Edition). Distributed vSwitches can also support 40Gb network interfaces, Private VLANs, Port Mirroring and Netflow (the ability to provide packet-capturing diagnostics to a monitoring server), and VxLAN (the ability to span Layer-2 traffic across a Layer-3 wide area network without having to use a Layer-2 Tunneling Protocol). In addition to their own Distributed vSwitch, VMware now supports third-party virtual distributed switches, each with their own unique feature-sets.

**NSX** is the latest VMware addition to the software-defined data center. NSX expands the networking features available in the vSphere hypervisor to include, routing, bridging, firewalling, VPNs, load balancing, and integration with third-party network devices. We are currently considering NSX integration with Palo Alto Networks virtual firewalls.

**Host Networking**

For now, at some point we all have to leave the virtual world to enter the physical world. When we were a small business we could easily get by with multiple 1Gb network interfaces on our hosts, though because the price per port of 10Gb switches has dropped so dramatically, most consumers seem to be using at least a 10Gb fabric between their hosts and storage. Because we selected a monolithic (as opposed to blade) architecture, our choice was easy. For our Data Center Enterprise Class host servers we have had great success using a total of four 10Gb Intel Network Interface Cards (NICs). This allows us to route one 10Gb connection from each card to four separate physical distribution switches (two for Storage and two for Production traffic) — allowing us to withstand a switch failure (or planned upgrade) without dropping traffic. For the two 10Gb Production switches we used Cisco’s Virtual Port Channel feature, in combination with 802.3ad LACP on each of the host uplinks — this allowed us to leverage both links for a combined 20Gb throughput and stateful failover. Because most all iSCSI SANs support link aggregation at the application layer, we did not use 802.3ad on the 10Gb Storage switches. For a blade enclosure, the uplink to physical switches is manufacturer dependent. Cisco enclosures use Fabric Extenders (FEXs) whereas Dell’s FX2 chassis uses FN-IO Aggregators. Both our Infrastructure and Branch Office Domain Class host servers use 802.3ad LACP uplinks to diverse Layer-2 access switches for maximum redundancy.

**Layer-2 Switching**

For many years we successfully leveraged Cisco’s Layer-2 Nexus 5020 10Gb switches: two for Production traffic and two for Storage traffic. We used Cisco’s Virtual Port Channels (VPCs) to logically connect each of the two pairs, allowing us to lose both a storage and a production switch without dropping traffic. We recently upgraded each of our legacy Layer-2 10Gb data center switches to Layer-3 Cisco Nexus 93180 switches, each supporting 48 10Gb ports. In our branch offices we use Cisco’s SG300-52MP 52-port Gigabit Power-over-Ethernet access switches.

**Layer-3 Routing**

Because Cisco’s Nexus 5020 switches had no Layer-3 routing capabilities, we used a pair of Cisco 3750X Layer-3 switches for all our core routing needs — the 3750X conveniently had 48 1Gb ports for all our 1Gb network devices in the data center and four 10Gb ports that we used to provide redundant routing for our Layer-2 10Gb switches. Cisco’s new Nexus 93180 Layer 3 switch allows us to route 10Gb data center traffic on the switches connected directly to the vSphere hosts, avoiding the extra (albeit small) latency of leaving the device as we had done in the past. We use Cisco 2951 routers with Security and Unified Communication License Bundles (to include CUBE and 32 DSP channels) in our branch offices.
VLAN/VRF Segmentation

Virtual Local Area Networks (VLANs)
VLANs are intended to segment a single Layer-2 network into multiple broadcast domains. We use VLANs extensively; here's a simple explanation of why for anyone unfamiliar with VLANs:

Let's say you have a 48 port switch, and on that switch you have several hosts: 6 web-servers, 8 SANs, 20 workstations, 4 routers, and 2 firewalls. VLANs allow you to segment each of those different hosts into 5 isolated groups that can't communicate with each other without using a router, even though they are all on the same switch. Once the five groups are segmented we can allow only the access needed between each group by using a router or firewall Access Control List (ACL) — for example: members of the workstation users group only need TCP port 80 & 443 to access the web-servers group, and don't need access to the SANs group at all — by placing a router ACL on these groups we limit anyone in the workstations group from accessing restricted resources. We can also use VLAN segmentation to prioritize different traffic — using the example above, we can create a Quality of Service (QoS) policy that ensures all router traffic has the highest priority, followed by SANs, Servers, Firewalls, and finally workstations. Thus, the servers will never have to compete for traffic over end-users who are surfing the Internet.

Virtual Routing and Forwarding (VRFs)
VRFs are to Layer-3 what VLANs are to Layer-2. VRFs allow us to create multiple isolated routing domains on a single router. We also use VRFs extensively; here are two use cases for VRFs:

Scenario 1: You have a vendor that requires remote access to their devices located at each branch office. You could provision a completely separate dedicated Internet connection, like a cable modem, and a dedicated firewall at each branch office, or you could use a VRF to span a dedicated and isolated network across your existing branch routers back to your data center and dump the vendor into a single DMZ network. Using VRFs would save you the cost of dedicated firewalls and internet access at each site and allow you to centrally manage the vendor’s access without ever letting the vendor’s traffic communicate with hosts in your production VRF. We did this.

Scenario 2: You want to create a thoroughly realistic lab/test network that spans multiple data centers across your Wide Area Network — you even want to use the exact same IP Addresses used in your production live network, and you know that if any of the test/lab traffic touches production you will be brushing the dust off your resume. You could use a VRF to segment your lab/test traffic across the WAN, because each VRF is completely isolated from other VRFs, it doesn't matter if you are using the same IP Addresses in separate VRFs — they can never conflict with each other. Using VRFs in this manner allows our IT staff to create a complete mirror of any and all of our production servers, and by placing them on the Lab/Test VLAN (that is part of the VRF), they can test the effect of real-world latency, multicasting, routing, etc. across our WAN with no impact to production.

To ensure the security and isolation of these VRFs, we encrypt them in General Routing Encapsulation (GRE) Tunnels. That way they are both logically and cryptographically isolated. We discuss this topic more in the Designing Infrastructure Section later in this white paper. Most reputable router manufacturers support VRF; our Cisco 2951 Integrated Service Routers support VRF and GRE tunnels provided you have the Security Licensing Bundle.
WAN Acceleration

Our data centers are over 350 miles away — great for Disaster Recovery, but challenging for data replication. However, we are fortunate enough to have a 100Mb fiber-optic VPLS connection between our primary and secondary data centers resulting in about 12ms of latency. Because we replicate our entire primary data center to our secondary site, as frequently as every five minutes, we require a WAN Accelerator. We opted to use Riverbed’s 6050 Steelhead WAN Accelerator, which allows us to squeeze the equivalent of about 800Mb of peak throughput through a 100Mb link. Without Riverbed’s WAN Acceleration, there is no way we would be able to replicate as much data over that 100Mb connection.

WAN Redundancy

We have discovered many advantages of virtualizing our desktop infrastructure, to include:

- Significantly increased performance (because the workstations are running in the same data center as the servers they use)
- Significantly increased security (because ultra-hardened client workstations or zero clients at branch locations have a lower attack surface than full desktops, and because we can use a secure remote access methodology that doesn’t rely on gratuitous VPN access)
- Significantly increased efficiency (because we can now quickly replace a “pooched” desktop by spinning up a clone from a template in minutes, and because we can roll-back failed test deployments to a known working state in seconds)
- Significantly increased business continuity (because we can replicate our entire desktop infrastructure along with our servers between processing sites, ensuring that our end-users will be able to function in the event of a disaster as effectively as our servers)

That said, we felt it was important to identify how we process transactions in the event of a WAN outage. For this scenario we developed two solutions:

Branch in a Box (Infrastructure Host Servers)

We worked with Cisco to develop the UCS-E modular server that fits in the back of an Integrated Service Router. With two of Cisco’s 2951 Integrated Service Routers and UCS-E160D-M2 Servers running vSphere, we have fully redundant core routing, DHCP, VoIP Call Control and Voicemail in 4U of rack space. Using this solution, our branch office can continue to use their VoIP services and access voicemail in the event of a total WAN failure. And when coupled with SAN Acceleration software like Pernix Data or Atlantis Computing this single socket server has enough resources to handle approximately 12 virtual desktops using only two onboard SAS SSDs; unfortunately, our current core banking application has unreliable “offline-mode” functionality, so while this solution works perfectly for network and VoIP infrastructure, we opted for the strategy below to ensure maximum VDI reliability at remote branches in the event of a total WAN failure.

Very Redundant WAN Connections

Even though we have branches all over Alaska, all of our locations are on a highly-reliable, fiber-optic, diverse, VPLS backbone. That isn’t to say we don’t have WAN outages — Murphy the intern, formerly of Murphy’s Law Firm, frequently likes to trip over cables and pull power cords, but generally speaking, our WAN is very reliable. To account for Mr. Murphy, we use out-of-band alternative network paths (usually 1Gb/sec downstream, 50Mb/sec upstream cable-modems, but sometimes 100Mb point-to-point 25GHz wireless radios) that use a completely diverse core network — it’s incredibly rare for both our fiber-optic and alternate paths to be out at the same time. We leverage always on VPN tunnels from our alternate paths to our external firewalls at each data center. That said, if both the Primary and Secondary WAN connections fail, we simply close the branch until one of them is restored — that’s happened once in over six years.
**Licensing**

There are a couple of licensing considerations that we took into account when designing our virtual desktop infrastructure.

**VMware**

At the time of writing, VMware has the option to license their virtual desktop infrastructure based on the number of concurrent Horizon sessions. That means if we have 500 employees, and only 400 of them are logged in (inside or outside our network), than we only need 400 concurrent licenses. It also means that we can use as many hosts, sockets, cores, etc. as needed to support our virtual desktop infrastructure. However, we can only run the virtual servers needed to support our virtual desktop infrastructure on those hosts — any other virtual server must be run on separate vSphere hosts.

For hosts running non-VDI workloads, VMware is typically sold by the socket — and that dictated how we designed our physical hosts. For us, a dual-socket high-core-density server is best for non-VDI workloads, because in order to make the memory-to-processor ratio work for today’s 22 core quad-socket processors, we would need an exorbitant amount of RAM.

**Microsoft**

At the time of writing, for virtual desktop workloads we are required to use either a Microsoft Virtual Desktop Access license (about $90/seat or ~$90/user) or an Enterprise License agreement with active Software Assurance (which comes out to (~$80/seat). If we didn’t have either of those, we could not legally run our Windows desktops in our virtual environment.

As of Q3 2016, Microsoft is in the process of converting their data center licensing from per socket to per core. This change required us to be fully aware of the impact on our processor, socket, and core selection.

**Security**

**Two-Factor Authentication**

At a minimum, as a financial institution we felt we should implement two-factor token authentication for all remote access. We selected Dell Defender’s two-factor token solution because it seamlessly integrates with Active Directory, can be installed in a matter of minutes, is easy to maintain and upgrade, and ultimately passed the Toaster Test™ (see above). We found other solutions, such as RSA and Symantec were exponentially more difficult for us to install and support.

**Firewalls**

As previously mentioned in the Networking section, VMware has put significant development into its NSX platform. We are currently evaluating Palo Alto Networks for our East-West (client-server) traffic as well as North-South (external and B2B security contexts) traffic.

**Administration**

We have used Dell KACE, a traditional inventory and package-based application deployment solution, to manage our virtual desktops for several years. That said, we are in the early stages of a promising Unidesk proof of concept.
Selling the Solution: Managers

Anyone not interested in building strategic and tactical plans, capital and operational expense forecasts, and tables of organization and equipment, feel free to skip this section. For everyone else (Hi mom), here is how I approached these topics.

Most organizations are fueled by a mission: profitability, charity, compliance, security… something. When technology initiatives can be leveraged to fulfill that mission, leadership will be instantly more receptive to those initiatives. We have found investing the time to focus on our organization’s mission and using our knowledge of technology to directly support its needs has resulted in obtaining the resources that allow us to efficiently support the mission, instead of being run over by it.

Building Trust

What follows is a list of practices we’ve successfully used to gain the trust and support of leadership:

1. **We discovered the organization’s business pain points.** We interviewed every power-user, line-supervisor, and manager to find out what technology works well and what could be improved upon. By understanding our customers’ needs and building a business case, we knew we were more likely to get funding. And along the way we found some easily fixable technology problems that we were able to quickly address, elevating our hero status and giving us a better understanding of our customers. Our Board and Executive Officers had a clear list of goals and directives — we looked for ways that our technology initiatives could support those goals.

2. **We built consensus within the team.** Our managers and team leads annually organize a planning session (preferably offsite with premium food and beverages) to think of ways to meet organizational and departmental goals. Because engineers and team leads are usually the ones who are pushing for capital investments, they forecast their own completion dates and timelines – then we use The Scotty Formula™, multiplying their aggressive forecasts by 2.25X to get close to real-world timelines.

3. **We made an effort to educate leadership about the technology challenges within the organization in a format they could relate to.** Each year we presented a summary of technology risks and challenges at least a month before the annual planning and budgeting process. We included pictures, graphs, and metaphors: whatever it took to resonate with our audience. Rather than solve one problem at a time, such as “how to provide secure remote access” or “how to reduce WAN expense,” we identified strengths and weaknesses across our entire organization: networking, storage, servers, workstations, mobility, security, disaster recovery, licensing, operations, etc. We invested the time to explain and document these technical challenges to build trust with our leadership and demonstrate our full understanding of our organization’s technology challenges and how to address them.

4. **We forecasted a long-term (3-5 year) strategic plan with clear goals and accurate expenditures for the upcoming year, and educated cost estimates for subsequent years.** We started with our most important assets, people, addressing proposed staffing changes. Then we enumerated current organizational challenges and risks identified by leadership, peer-management, and regulators — illustrating how we will solve those challenges. Then we highlighted both recent accomplishments and opportunities for improvement (with lessons learned). Then we wrote a summary justification for any major expenditure that exceeded a dollar or man-hour threshold ($25,000 or 160 man-hours). This justification identified our challenges, the proposed solutions, their cost, their expected return on investment, and how success would be measured. If we couldn’t justify the expenditure (either in hard dollars/man-hours saved, or significantly increased security, performance, or availability) we omitted it until we could justify it. This demonstrated our alignment with the business, understanding of our resource limitations, and our accountability.
5. **We built a corresponding budget.** Any good strategic plan should be married to realistic budget numbers, and what Board doesn’t appreciate hyper-accurate numbers? We identified all our operational expenses: recurring software maintenance, telecommunications, Full-Time-Employees (FTE), leased equipment, etc. We found a good way to start this list was to enumerate our entire software inventory — the logic being that most software has an initial and maintenance cost. Then we enumerated all our planned capital expenditures: any investments that would likely add to our organizations assets and be depreciated over multiple years (e.g: SANs, servers, networking gear, new licensing, etc.).

We created a working budget in a spreadsheet and included:

a. Project line-item-identifiers: to correlate future invoices to what we forecasted
b. Internal and external descriptions: both for our team and in laymen terms for other departments like Accounting
c. Projected cost: based on publicly available pricing and shipping from large retailers like CDW (not MSRP or fully negotiated pricing)
d. Actual cost: every single future invoice, down to the penny
e. Cost details: to include the date, purveyor, exact cost, invoice number, and description of every invoice
f. Projected man-hours: to determine how many personnel are allocated per project
g. Recommended depreciation: 3-5 years for most equipment, 10 on in-wall cabling
h. 5 year expenditure forecast: the month the expenditure will be made, the length of time it will be depreciated, and the projected maintenance renewal cost and frequency, if applicable.

6. **We presented with flair.** When we initially proposed the concept of a virtualized desktop infrastructure, I had my team roll three carts into the Boardroom on queue. The first cart was overloaded with six traditional PCs and six retail boxes of Windows 7. The second was struggling to hold a variety of ten legacy 2U rack-mounted servers with ten retail boxes of Windows Server. The third cart had only three 2U servers, ten retail boxes of Windows Server, and six retail boxes of Windows 7. By visually illustrating the consolidation of ten servers and six desktops onto three servers, we were able to connect with every member of the Board. The visualization of hand-carrying the retail boxes of Windows from one host to another perfectly illustrated the concept of vMotion and Dynamic Resource Scheduling, and it illustrated the concept of how virtual infrastructure can seamlessly migrate across disparate hardware to augment DR/BC plans.

**Tip:** I advise against smoke machines and lasers, but taking the time to relate to your leadership and communicate technical solutions in a way that resonates with them will always pay dividends.

7. **We consistently delivered.** The key to building trust is consistent execution and delivery on promises made. At my very first Board Meeting I made the bold claim that if I didn’t deliver everything I projected on time and under budget, they wouldn’t have to fire me, I would resign. One of the Board Members then asked, “Well what good does that do us, we will have already spent the money!” To which I replied, “Touché... That being the case, you simply have my word: I will not fail.” Eight months later, the project was completed, under budget, and I was still employed. Every year thereafter, I have met my goals and demonstrated realized returns on our technology investments, and in doing so earned the trust of my leadership, transforming the perception of IT from a drain on revenue into a resource that empowers our organization — something we never take for granted.
Architecture Overview: Engineers

Dear Engineers: I intentionally excluded detailed engineering diagrams from the “Evaluating Solutions” Section because the truth is, there was no point in getting you excited until you’re reasonably sure you have Budget and Approval (Layers 8 and 9 of the OSI Network Model). From this point on let’s assume you’re funded.

Branch Office Topology

Figure 11 below illustrates the infrastructure found in our standard branch office. Each branch uses three vSphere Host Servers: two Infrastructure Class Cisco UCS-E160D-M2 blades that fit into Cisco 2951 Integrated Service Routers and one Branch Office Domain Class Dell R730xd rack-mounted Host Server. The Cisco UCS-E Host Servers are used to run redundant virtual instances of our 802.1x Network Access Control, Cisco Unified Communications Manager (VoIP Call Manager) and our Cisco Unity Express (VoIP voicemail) servers. This configuration allows System and Network Engineers to perform maintenance on critical network infrastructure without causing a network or phone outage, and greatly enhances our business continuity capabilities. All Microsoft Windows servers are hosted on the Dell R730xd Host Servers, allowing us to consolidate and maximize our Windows Server licensing. The Dell R730xd is used to host about 64TB of “casino quality” digital surveillance and a read-only stub of our Active Directory Domain. We selected Cisco SG-500 series access switches because they have Full-Power 802.3af capability (to support our phones and IP-based cameras) and a decent backplane at a very reasonable price-point. All of our servers, routers, and even our out-of-band console and power management device are dual homed to each switch so we can withstand a full switch failure. As previously mentioned, we utilize Cisco VoIP infrastructure, to include Cisco 8865 video phones and Cisco Tandberg Codecs along with Clear One’s ConvergePro and BeamForm Microphone Array in our larger branches for whole-room video conferencing.

Each branch has a dedicated 5-20Mbps Fiber-optic VPLS connection terminating on a primary core router, and a diverse medium (usually cable modem) Internet-facing connection terminating on a secondary core router. Each branch secondary router’s Internet connection is used for two purposes:

- An always-on VPN connection back to our data centers for backup connectivity.
- Wireless access, segregated from all internal Bank networks, so that employees can use their mobile devices without consuming higher-cost VPLS bandwidth.
Data Center Topology

Like our branches, our Data Centers were designed with redundancy in mind. As you can see in Figure 12, our data centers have redundant core routers, each connected to dedicated VPLS Fiber-optic circuits. Our data centers have two classes of Host Servers: Infrastructure and Enterprise. Like our branch Infrastructure Class Host Servers, the Cisco UCS-E blades in our Cisco 4451 routers run redundant VoIP infrastructure on the VMware vSphere hypervisor. This ensures that even in the event of SAN maintenance or a partial data center failure that our company has access to phone and voicemail services. The Enterprise Class Host Servers are Dell PowerEdge Servers running vSphere that supports our virtual server and desktop infrastructure. All of our Enterprise Host Servers have redundant 10Gb network connections to our Layer-3 Core Distribution Switches: two for production traffic and two for storage traffic.

Datacenter 1 (dc1)

Our Datrium and Nimble Storage SANs are also redundantly connected into our 10Gb Core Distribution Storage Switches. We also have redundant 1Gb Layer-3 Core Access Switches for management devices such as UPS, power distribution units, environmental monitoring, and IP-based cameras. We use a Riverbed Steelhead WAN Optimizer between our core routers and core switches to increase our data center throughput by a little over 8X — should that device malfunction, it fails to a pass-through state, allowing traffic to flow even while it’s powered off. Lastly, we have diverse connections between our Core Switches and our redundant stateful-firewalls, allowing us to seamlessly fail-over from one firewall to another without dropping our current sessions.

Figure 12: Data Center Topology

Our Datrium and Nimble Storage SANs are also redundantly connected into our 10Gb Core Distribution Storage Switches. We also have redundant 1Gb Layer-3 Core Access Switches for management devices such as UPS, power distribution units, environmental monitoring, and IP-based cameras. We use a Riverbed Steelhead WAN Optimizer between our core routers and core switches to increase our data center throughput by a little over 8X — should that device malfunction, it fails to a pass-through state, allowing traffic to flow even while it’s powered off. Lastly, we have diverse connections between our Core Switches and our redundant stateful-firewalls, allowing us to seamlessly fail-over from one firewall to another without dropping our current sessions.
Network Topology

Core Switching

At our branch offices, we generally use Layer-2 access switches, leaving all Layer-3 routing to the redundant Cisco 2951 routers. This allows us to maintain a low enough price point to afford redundant 48-port Full-Power-over-Ethernet switches. At the data centers we use access switches in our DMZ and border networks to provide redundant connectivity to external networks, but the heavy lifting is done by our Layer-3 Core Distribution and Core Access Switches. Our Core Distribution Switches are made up of four Cisco Nexus 93180 switches: two for production and two for storage. Each vSphere Host Server, SAN appliance, and Archive appliance has redundant 10Gb connections to each of their respective pairs of Core Distribution Switches. Each of the Core Distribution Switches are interconnected via Virtual Port Channels using 40Gb twinax cables.

Note: We generally use copper twinax cables for high-speed low-latency connections. While fiber-optic can achieve longer distances, copper electrical connections will always have lower latency because they don’t have to convert the optical signal into an electrical signal.

Our Core Access Switches are made up of two Cisco Catalyst 3850 switches configured as a single logical switch stack using Cisco Stackwise+ cables across their backplane. This redundant configuration ensures high-availability for our 1Gb data center devices, to include: Infrastructure Class vSphere Host Servers, Power Distribution Units, UPS, Console Servers, Environmental Monitoring Servers, IP Cameras, and other management hosts.

By leveraging Virtual Port Channels and Stackwise+ cables, we have the ability to configure multi-homed production hosts for 802.3ad Link Aggregation Control Protocol (LACP). When using LACP in a virtualized environment it is most effective to load-balance using an IP addresses instead of a MAC addresses. Generally, we don’t enable LACP on switch ports that connect to Storage Area Networks because most SAN vendors use their own algorithms to aggregate multiple connections.

We opted for Layer-3 Core Distribution and Access switches in our data centers because we wanted to keep the routing close to the devices that use it, allow for redundant failover of our routed traffic across multiple devices, and implement a robust Quality of Service framework to prioritize business services.

All Layer 2 traffic is segregated into functional VLANs to maximize security, performance, and traffic optimization. We further segregate our Layer-3 network traffic by leveraging Virtual Router Forwarding (VRF). Much like a Layer 2 VLAN works on a switch, VRF allows us to create multiple “contexts” on our physical routers, each with its own isolated routing tables. These isolated VRFs are then secured in Generic Routing Encapsulating (GRE) tunnels that span our private VPLS network and IPSEC tunnels that traverse the public Internet. This data encapsulation is illustrated in Figure 13 below.

Figure 13: Data Encapsulation
Core Routing

Our routing strategy includes the ability for our Logical Primary and Secondary Data Centers to operate from either Physical Data Center in an active-active configuration. The active-active routing configuration is possible because each Logical Data Center is implemented inside of a unique Virtual Route Forwarding (VRF) context at both physical locations. We leverage the ability of Border Gateway Protocol (BGP) to redistribute each Logical Data Center’s routing tables into the alternate Logical Data Center’s VRF. In the event of a total Physical Data Center loss, BGP redistribution will automatically ensure the alternate Physical Data Center assumes routing responsibility for both Logical Data Centers. We then used Hot Standby Redundancy Protocol (HSRP) to smoothly transition the Layer-3 interfaces for both Logical Data Centers over to the active physical location.

As you can see in Figure 14 above, each VRF is secured by leveraging the Cisco Dynamic Multipath Virtual Private Networking (DMVPN) technology on both our public and private service-provider circuits between Data centers and Branch locations. All DMVPN tunnels are encrypted between all locations across all service providers. By combining VRF and DMVPN functionality we effectively created a fully-meshed, fully-redundant, multi-hub, encrypted network. This also ensured that when we have branch-to-branch traffic, tunnels are dynamically created to provide a secured session regardless of the transport medium. This allowed us to provide secure enterprise services over low-cost Internet connections as easily as over dedicated VPLS circuits.

**Figure 14: High Availability High Security WAN**
Business Continuity

In addition to our production Data center VRFs, we leverage VRF functionality to virtualize routing contexts for different security zones to include a unique VRF for production, wireless, internet, kiosk, vendor-specific data, and laboratory testing. Each VRF ensures total separation between other VRF contexts, allowing us to fully leverage our hardware and telecommunications investments for multiple purposes. Put simply, we are running multiple virtual routers on a single hardware device, each encrypted and isolated from the other.

By combining our BGP failover functionality with a Layer 2 Tunneling Protocol version 3 (L2TPv3) Pseudo-wire Bridge, we introduced the ability to span traffic from our server, workstation, and management VLANs across both physical Data centers at the same time. This concept is illustrated in Figure 15 below.

**Note:** This design was based on the technology of the time, it is not vendor specific, and there are emerging technologies that accomplish the same thing, such as: Cisco OTV/LISP, VTEP, VXLAN, NSX, etc.

![Figure 15: Data Center Bridging](image)

While it may seem like a small thing, having our server, workstation and management VLANs both routed and bridged across data centers, allowed us to do some pretty interesting things, such as not having to change IP addresses in the event of a partial or complete failure event. This greatly reduces the complexity and time required to promote a Logical Data center at an alternate physical location. This core routing topology also allows us to accurately simulate testing across our WAN in a Laboratory VRF context. Because the Laboratory context houses a perfect duplicate of production networks with the same IP addresses, we can easily test and validate our documented Business Continuity procedures without impacting production.
Lessons Learned
The concepts and methodologies illustrated in this whitepaper are based on our fourth generation reference design. For us, there were many lessons learned over the course of deploying this infrastructure, to include:

• The first time we migrated a data center from a corporate office (where 50% of our personnel work) to a collocated facility was scary. Even though our calculations indicated it would work, we had never bifurcated so many users from their servers across a WAN. However, the migration turned out better than expected for our physical desktops thanks in large part to our Riverbed Steelhead WAN Optimization. That said, virtualizing our desktops and leveraging VMware Horizon dramatically increased our performance, and had the unforeseen added benefit of incenting our employees to rapidly abandon their physical workstations and excessive VPN access (only available on old bank-issued laptops) for our more secure remote access using the VMware Horizon Security Server (allowing them to use a wider array of client devices).

• We spent significant time testing two-factor authentication solutions and ultimately selected Dell Quest Defender. We found that deploying two-factor authentication early in the beta cycle prevented us from having to reconfigure a lot of clients.

• Many of our employees who use multiple monitors initially experienced display errors when using Horizon. We quickly found the solution was to increase the default “Max number of monitors” to 4 (the highest setting), and maximize the available video memory for each Desktop Pool within the View Administrator web-client. After making those changes even users with large monitors (single 55” 4K or four 24” 1080p monitors) had had no issues.

• The VMware Horizon Client for Microsoft Windows uses the local desktop’s Default Playback Device and Default Recording Devices for the virtual machine’s audio output and input respectively. To easily switch between these devices we found a command-line application, NirCmd, that allowed us to change the default Playback/Recording devices via the “setdefaultsounddevice” argument. After creating a batch file that switched devices, swapping between headset and speakers is as easy as toggling a hotkey.

Conclusions
VMware Horizon is one of the few technologies our Systems & Network Engineers don’t lose a lot of sleep over: it just works. From start to finish, we had the Horizon Connection Brokers and Security Servers online in an afternoon, including its advanced features, like two-factor authentication. VMware Horizon has become a key component of our infrastructure, enhancing our security, performance, efficiency, business continuity, and manageability — and the biggest impartial indicator of its success is the fact that all of our employees want to be virtualized. We have successfully migrated a little under 70% of our entire workforce (about 400 employees) to virtual desktops, and as we just took possession of our newest Host Servers, we will have the remainder virtualized by year’s end. Most of our management, critical support-staff, and lenders have remote access to their virtual desktops via two-factor authentication, a feature that allowed us to also hire telecommuters for specialized positions — a new paradigm for us that has proven very successful. By virtualizing our tellers, even our most remote branches experience stellar application performance because their workstations are running in the same data center as the servers they depend on — this translates into a more positive customer experience and makes our employees feel more connected. Because our loan officers and wealth management advisors have the ability to securely access their virtual machines using their tablet or laptop of choice, they can provide on-site service at our customers’ convenience and make decisions in the field faster. In short, VMware Horizon has delivered a solid return on our investment and we’re glad we have it.
About the Author

Benjamin Craig is the Executive Vice President and Chief Information Officer of Northrim Bank, one of three publicly traded, publicly owned companies based in the state of Alaska. Mr. Craig began his Information Technology career in 1995 with the United States Air Force, where his primary duties included managing the server, network, and security infrastructure of Misawa Air Base, Japan. After honorably serving overseas, Mr. Craig provided interim network management for 3NF Corporation, a technology services provider specializing in military applications until 2001. From 2001 until 2009, Mr. Craig served as the Vice President, Information Technology Manager for River City Bank in Sacramento, California. Mr. Craig joined Northrim Bank in 2009 and was promoted to Chief Technology Officer in 2010. Mr. Craig co-authored “Microsoft Exchange Server 2007: Implementation and Administration” and is an active leader of the Anchorage VMware User Group.

The author would like to acknowledge the following individuals for their contributions to this project, as well as for providing information relating to the architecture, design, and implementation:

Erick Stoeckle provided all the example reference configurations referenced herein. Erick is Northrim Bank’s Vice President, Architecture & Cybersecurity Manager. Erick joined the Company in 2012 and accepted his current position in May 2016. Mr. Stoeckle’s information technology career began in 1999 when he was employed by Acclaim Technology, a security consulting firm in San Jose, California. In 2004, Mr. Stoeckle transitioned to Verizon Business where he managed mission-critical networks for the U.S. Department of Justice, Environmental Protection Agency, and Fish and Wildlife Service. In 2007, Mr. Stoeckle left Verizon to work for River City Bank as the Senior Network Engineer. In 2009, he joined Delta Dental Insurance Company as a Lead Network Engineer. Mr. Stoeckle obtained the highly coveted Cisco Certified Internetwork Expert (CCIE) certification in 2005 and later obtained the specialized CCIE Voice certification in April 2012.

Joshua Rabe performed and documented all the benchmark tests referenced in this white paper. Joshua is one of Northrim Bank’s Senior Systems & Network Engineers. Joshua started his Information Technology career in 2002 as a Systems Technician for CRW Engineering. In 2007 Joshua joined First American Corporation, where he developed custom .NET applications to streamline business processes. In 2008 Joshua joined Action Property Management where he was responsible for implementing all new enterprise systems and virtualizing the majority of their data center. Joshua joined Coulee Medical Center in 2012 as a Systems Architect, where he designed and implemented an enterprise-wide network and virtual infrastructure. Joshua first joined Northrim Bank in 2013, as a Systems Engineer responsible for upgrading Exchange and SharePoint servers. He left to take a challenging role at North Bend Medical Center in Coos Bay Oregon where he overhauled the campus-wide physical and virtual infrastructure, then returned to Northrim Bank in 2016 as a Systems & Network Engineer responsible for the Bank’s virtual infrastructure. Joshua is a Microsoft Certified Technology Specialist and Linux Certified Professional, he has completed the ITIL Foundations 2 Service Level Management Certification, and earned his Bachelor of Science in Computer Information Systems from the University of Phoenix in 2013.
About VMware

VMware, the global leader in virtualization and cloud infrastructure, delivers customer-proven solutions that accelerate IT by reducing complexity and enabling more flexible, agile service delivery. VMware enables enterprises to adopt a cloud model that addresses their unique business challenges. VMware’s approach accelerates the transition to cloud computing while preserving existing investments and improving security and control. With more than 250,000 customers and 25,000 partners, VMware solutions help organizations of all sizes lower costs, increase business agility and ensure freedom of choice.

References

VMware Always On Design Guide for 24/7 Branch Anywhere Solution
VMware Horizon
VMware vSphere
VMware Horizon Technical Resources
VMware Horizon Documentation
VMware vSphere Documentation

For more information or to purchase VMware products, call 1-877-4VMWARE (outside North America dial +1-650-427-5000), or visit www.vmware.com/products, or search online for an authorized reseller. For detailed product specifications and system requirements, please refer to the VMware Horizon documentation.
Appendixes

Appendix A: Analysis and Forecasting Worksheet Example

This worksheet is intended to illustrate the concepts discussed herein, and provide the reader with a foundational understanding of how performance metrics can be analyzed. It is not intended to be used as decision point in your infrastructure design, deployment, or configuration. That said, here are the steps we took to compile the accompanying Virtualization Estimator Worksheet:

Step 1: Open the Virtualization Estimator Worksheet. There you will find an example spreadsheet (tab) in that workbook called “BS-DC1-MAIL1.” Make a copy of that example spreadsheet (tab).

![Figure 16: Example Performance Monitoring Data](image)

Step 2: Replace the existing data starting on Row 5 on the newly duplicated spreadsheet with the performance monitoring data you collected (as discussed in the “Collecting Metrics That Matter” Section of this whitepaper). Create a new spreadsheet for every server and “workstation class” that you want to analyze. The final product should look identical in format to the example spreadsheet.

Note: The formulas in rows 2 and 3 are the only “magic” on this spreadsheet. They calculate the Peak and 97th or 98th percentile averages (depending on whether you are analyzing servers or desktops). The formulas are quite simple, here is their syntax:

\[
\text{PERCENTILE}(X5:XY,Z)
\]

where X5 is the column, row 5, of data you want to average,
where XY is the column, and Y is the last row of data you want to average,
and where Z is the percentile (expressed in one-hundredths) that you want to average to.

For example: \(\text{PERCENTILE}(B5:B5609,0.97)\) would yield the 97th percentile average of all values within column B between rows 5 and 5609.

Step 3: You will find three more example spreadsheets (tabs) in that worksheet called “Disk Reads & Writes per second,” “Bytes Read & Written per second,” and “Processor Utilization.” If you so desire, you can alter the data sources for those charts to point to your newly imported data. The charts are intended to visualize performance trends, you can choose to create a separate chart for each server you are analyzing, or just reuse the same chart as needed — it’s totally optional.

Step 4: Enter the output from the formulas in columns B-G of row 3 from each of the spreadsheets (tabs) that you created in Step 2 into the appropriate dark green fields on the “Virtualization Estimator” spreadsheet (tab). Fill in the remaining green fields on that same spreadsheet for every Server and Workstation Class you want to include in your analysis.

Once all the green fields are filed in, the blue and red fields will automatically populate and render the totals.
Appendix A: Analysis and Forecasting Worksheet Example (continued)

The following formulas were used to compile the accompanying Virtualization Estimator Spreadsheet:

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>FORMULA</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAM in Use</td>
<td>=TOTAL_RAM * (&quot;Memory: % Committed Bytes In Use&quot; / 100)</td>
</tr>
<tr>
<td>Total Processor Available</td>
<td>=&quot;NUMBER OF PROCESSOR CORES&quot; * &quot;PROCESSOR SPEED&quot;</td>
</tr>
<tr>
<td>Processor in Use</td>
<td>=&quot;Total Processor Available&quot; * (&quot;Processor: % Processor Time&quot; / 100)</td>
</tr>
<tr>
<td>Net Increase/Decrease</td>
<td>=(&quot;USED DISK SPACE SAMPLE 1&quot;) - &quot;USED DISK SPACE SAMPLE 2&quot;) /</td>
</tr>
<tr>
<td></td>
<td>-(&quot;USED DISK SPACE SAMPLE 1&quot;)</td>
</tr>
<tr>
<td>Projected Grown in 1 year</td>
<td>=&quot;USED DISK SPACE SAMPLE 2&quot; + (&quot;USED DISK SPACE SAMPLE 2&quot; * &quot;Net</td>
</tr>
<tr>
<td></td>
<td>Increase/Decrease&quot;)</td>
</tr>
<tr>
<td>Projected Grown in 2 years</td>
<td>=&quot;Projected Grown in 1 year&quot; + (&quot;USED DISK SPACE SAMPLE 2&quot; * &quot;Net</td>
</tr>
<tr>
<td></td>
<td>Increase/Decrease&quot;)</td>
</tr>
<tr>
<td>Projected Grown in 3 years</td>
<td>=&quot;Projected Grown in 2 years&quot; + (&quot;USED DISK SPACE SAMPLE 2&quot; * &quot;Net</td>
</tr>
<tr>
<td></td>
<td>Increase/Decrease&quot;)</td>
</tr>
<tr>
<td>MB Read/s</td>
<td>=&quot;Physical Disk: Disk Read Bytes/sec&quot; / 1024 / 1024</td>
</tr>
<tr>
<td>MB Written/s</td>
<td>=&quot;Physical Disk: Disk Write Bytes/sec&quot; / 1024 / 1024</td>
</tr>
<tr>
<td>Total MB/s Read + Writes</td>
<td>=&quot;MB Read/s&quot; + &quot;MB Written/s&quot;</td>
</tr>
<tr>
<td>IO/s (IOPS)</td>
<td>=&quot;Physical Disk: Disk Reads/sec&quot; + &quot;Physical Disk: Disk Writes/sec&quot;</td>
</tr>
<tr>
<td>Write Activity change rate/</td>
<td>=&quot;MB Written/s&quot; / (&quot;USED DISK SPACE SAMPLE 2&quot; * 1024)</td>
</tr>
<tr>
<td>sec</td>
<td></td>
</tr>
<tr>
<td>MB/day</td>
<td>=&quot;MB Written/s&quot; * 60 * 60 * 24</td>
</tr>
<tr>
<td>Write Activity change rate/day</td>
<td>=&quot;MB/day&quot; / (&quot;USED DISK SPACE SAMPLE 2&quot; * 1024)</td>
</tr>
<tr>
<td>Snap Space Required</td>
<td>=&quot;USED DISK SPACE SAMPLE 2&quot; * &quot;Write Activity change rate/day&quot; *</td>
</tr>
<tr>
<td></td>
<td>&quot;DESIRED DAILY SNAPS&quot;</td>
</tr>
<tr>
<td>Snap Reserve Setting</td>
<td>=&quot;Snap Space Required&quot; / &quot;USED DISK SPACE SAMPLE 2&quot;</td>
</tr>
<tr>
<td>Replica Reserve</td>
<td>=&quot;USED DISK SPACE SAMPLE 2&quot; + (&quot;USED DISK SPACE SAMPLE 2&quot; * &quot;Write</td>
</tr>
<tr>
<td></td>
<td>Activity change rate/day&quot;)</td>
</tr>
<tr>
<td>Replica Reserve Setting</td>
<td>=&quot;Replica Reserve&quot; / &quot;USED DISK SPACE SAMPLE 2&quot;</td>
</tr>
<tr>
<td>Read/Write Ratio</td>
<td>=&quot;Physical Disk: Disk Reads/sec&quot; / (&quot;Physical Disk: Disk Reads/sec&quot; +</td>
</tr>
<tr>
<td></td>
<td>&quot;Physical Disk: Disk Writes/sec&quot;)</td>
</tr>
</tbody>
</table>

Figure 17: Analysis Calculus

Click here to download the Analysis & Forecasting Worksheet
Appendix B: Reference Configurations

Cisco 802.3ad Link Aggregation Control Protocol (LACP)

This is an example configuration used on a switch port that connects to another switch. This example also applies to a switch port that connects to a Host Server’s network adapter that is being used as an uplink to a virtual switch.

Switch1#
!
! This command changes the load balance method to calculate a hash for every Source and Destination IP address Pair.
port-channel load-balance src-dst-ip
!
! This command creates the LACP Etherchannel virtual interface. The interfaces in the group all need to match, speed, encapsulation, VLANs allowed, and the native VLAN.
! When making changes to the allowed VLANs it needs to be changed on this virtual interface.
interface Port-channel1
description Po1 | Trunk-port to <DEVICE_NAME>_〈PORT-NUMBER_1,PORT-NUMBER_2>
switchport access vlan <MANAGEMENT_VLAN>
switchport trunk encapsulation dot1q
switchport trunk native vlan <MANAGEMENT_VLAN>
switchport trunk allowed vlan <LIST_OF_VLANS_NECESSARY>
switchport mode trunk
!
! This command configures the interface to be part of the LACP group. The interfaces in the group all need to match, speed, encapsulation, VLANs allowed, and the native VLAN.
interface GigabitEthernet1/0/x
description gi1/0/x | <VLAN_NAME> | Trunk-port to <DEVICE_NAME>_〈PORT-NUMBER_1>
switchport access vlan <MANAGEMENT_VLAN>
switchport trunk native vlan <MANAGEMENT_VLAN>
switchport trunk allowed vlan <LIST_OF_VLANS_NECESSARY>
switchport mode trunk
logging event trunk-status
logging event status
flowcontrol receive desired
spanning-tree portfast trunk
channel-group 1 mode active
!
! This command configures the interface to be part of the LACP group. The interfaces in the group all need to match, speed, encapsulation, VLANs allowed, and the native VLAN.
interface GigabitEthernet2/0/x
description gi2/0/x | <VLAN_NAME> | Trunk-port to <DEVICE_NAME>_〈PORT-NUMBER_2>
switchport access vlan <MANAGEMENT_VLAN>
switchport trunk native vlan <MANAGEMENT_VLAN>
switchport trunk allowed vlan <LIST_OF_VLANS_NECESSARY>
switchport mode trunk
logging event trunk-status
logging event status
flowcontrol receive desired
spanning-tree portfast trunk
channel-group 1 mode active
!
-----END-----
Appendix B: Reference Configurations (continued)

Cisco Nexus Switch Virtual Port Channels

This is an example configuration used to create Virtual Port Channels (VPCs) across two separate physical switching devices. In this example we used Cisco Nexus switches; however, most switching vendors have a similar command structure.

Switch_1#

! This command turns on the VPC and the Link Aggregation Control Protocol (LACP)
! functionality
feature vpc
feature lacp
!
! The following configuration enables Jumbo frames via the Nexus QoS framework on the ! default traffic class
class-map type qos class-fcoe
policy-map type network-qos jumbo
class type network-qos class-fcoe
    pause no-drop
    mtu 2158
class type network-qos class-default
    mtu 9216
system qos
    service-policy type network-qos jumbo
!
! The following configuration creates the first Virtual Port Channel, domain “1” and ! settings. Each switch in the VPC domain will need the peer configuration settings to ! negotiate the VPC cross traffic.
vpc domain 1
    role priority 1
    system-priority 1
    peer-keepalive destination <SWITCH_2_IP_ADDRESS> source <SWITCH_1_IP_ADDRESS>
!
! The following command allows the primary switch in the VPC configuration to bring up the ! VPC interfaces when the secondary is offline.
    peer-config-check-bypass
!
! The following command allows for devices that load-balance return traffic, this command ! forwards the frame over to the correct peer.
    peer-gateway
!
! The following command allows the switch to bring up interfaces in the event of a reboot.
    auto-recovery reload-delay 300
!
! The following configurations create a virtual interface that ties to the logical Virtual ! Port Channel.
! This configuration will need to match the physical interfaces assigned to the Virtual ! Port Channel.
interface port-channel1
    description Po1 | VPC to <DEVICE_INTERFACE_1,DEVICE_INTERFACE_2>
switchport mode trunk
switchport trunk native vlan <MANAGEMENT_VLAN>
switchport trunk allowed vlan <LIST_OF_VLANS_NECESSARY>
spanning-tree port type normal
    speed 10000
vpc 1
!
Appendix B: Reference Configurations (continued)

Cisco Nexus Switch Virtual Port Channels (continued)

The following configurations configure the physical interface and joins it to the port channel.

```
interface Ethernet1/x
    description eth1/x | PO 1 | VPC-Trunk-port to <DEVICE_INTERFACE_1>
    switchport mode trunk
    switchport trunk native vlan <MANAGEMENT_VLAN>
    switchport trunk allowed vlan <LIST_OF_VLANS_NECCESSARY>
    spanning-tree port type normal
    channel-group 1 mode active

Switch_2#
    feature vpc
    feature lacp
    class-map type qos class-fcoe
    policy-map type network-qos jumbo
        class type network-qos class-fcoe
            pause no-drop
            mtu 2158
        class type network-qos class-default
            mtu 9216
    system qos
        service-policy type network-qos jumbo
    vpc domain 1
        The following command sets the priority of this switch to “2” which makes it the secondary
        role priority 2
        system-priority 1
        peer-keepalive destination <SWITCH_1_IP_ADDRESS> source <SWITCH_2_IP_ADDRESS>
        peer-config-check-bypass
        peer-gateway
        auto-recovery reload-delay 300
    interface port-channel1
        description Po1 | VPC to <DEVICE_INTERFACE_1,DEVICE_INTERFACE_2>
        switchport mode trunk
        switchport trunk native vlan <MANAGEMENT_VLAN>
        switchport trunk allowed vlan <LIST_OF_VLANS_NECCESSARY>
        spanning-tree port type normal
        speed 10000
        vpc 1
    interface Ethernet1/x
        description eth1/x | PO 1 | VPC-Trunk-port to <DEVICE_INTERFACE>
        switchport mode trunk
        switchport trunk native vlan <MANAGEMENT_VLAN>
        switchport trunk allowed vlan <LIST_OF_VLANS_NECCESSARY>
        spanning-tree port type normal
        channel-group 1 mode active
```

-----END-----

-----END-----
Appendix B: Reference Configurations (continued)

Cisco Quality of Service (QoS) Framework

This is an example configuration that was used to create a Quality of Service framework. In this template we identified common protocols and allocated them to one of four priority QoS Queues. This template does not alter the default QoS class. The template allows for matching traffic based on existing QoS tags as well as re-tagging specific traffic via ACLs.

Warning: QoS can negatively impact traffic, if incorrectly implemented. The QoS tags, matching protocols, and Access Control Lists will be unique to every environment. Do not attempt to use this example configuration in production.

Router#
! The following configurations create a RealTime QoS class to be used for voice traffic.
class-map match-any RealTime
   match ip dscp ef
   match protocol sip
   match protocol rtp
   match access-group name Queue_RealTime
!
! The following configurations create a High QoS class to be used for video, dns, snmp, ! ldap, icmp, and other important business traffic.
class-map match-any High
   match dscp af41
   match dscp af42
   match dscp af43
   match protocol dns
   match protocol ntp
   match protocol snmp
   match protocol ldap
   match protocol icmp
   match access-group name Queue_High
!
! The following configurations create a Medium QoS class to be used for the bulk of ! business traffic. The Medium ACL deny statements to push that traffic to the Low Queue.
class-map match-any Medium
   match dscp af31
   match dscp af32
   match dscp af33
   match access-group name Queue_Medium
!
! The following configurations create a Low QoS class to be used for bulk file transfer ! traffic. Backup traffic and large file transfers go in this queue.
class-map match-any Low
   match dscp default
   match dscp af11
   match dscp af12
   match dscp af13
   match access-group name Queue_Low
!
Appendix B: Reference Configurations (continued)
Cisco Quality of Service (QoS) Framework (continued)

! The following configurations create a policy that allocates percentages to each QoS queue. The percentages and the tags can be changed to fit business requirements.

```
policy-map C-PRIO_10M
  class RealTime
    priority percent 10
    set dscp ef
  class High
    bandwidth remaining percent 30
    set dscp af41
  class Medium
    bandwidth remaining percent 40
    set dscp af31
  class Low
    bandwidth remaining percent 29
    set dscp af21
  class class-default
    set dscp af11
```

! The following configurations shape all traffic to 10 Megabits and during congestion will allocate traffic bandwidth based on the percentages configured above.

```
policy-map 10M-shape
  class class-default
    shape average 10000000
    service-policy C-PRIO_10M
```

! The following configurations shape all traffic to 20 Megabits. Each branch will need a corresponding policy that matches its actual available bandwidth.

```
policy-map 20M-shape
  class class-default
    shape average 20000000
    service-policy C-PRIO_20M
```

! This access list allows specific traffic to be included into the RealTime QoS queue if it was not previously tagged.
```
ip access-list extended Queue_RealTime
  remark "permit DNS-U traffic"
  permit udp any any eq 5060
  permit udp any any eq 5061
  permit tcp any any eq 5060
  permit tcp any any eq 5061
```

!
Appendix B: Reference Configurations (continued)
Cisco Quality of Service (QoS) Framework (continued)

This access list allows specific traffic to be included into the High QoS queue if it was not previously tagged.

```
ip access-list extended Queue_High
remark ** UDP Services **
remark **permit DNS-U traffic**
permit udp any any eq domain
permit udp any eq domain any
remark **permit NTP traffic**
permit udp any any eq ntp
permit udp any eq ntp any
remark **permit management traffic**
permit udp any any eq snmp
permit udp any eq snmp any
permit udp any any eq snmptrap
permit udp any eq snmptrap any
permit udp any any eq syslog
permit udp any eq syslog any
remark ** TCP ** Services
remark **permit SSH Management traffic**
permit tcp any any eq 22
permit tcp any eq 22 any
remark **permit DNS-T traffic**
permit tcp any any eq domain
permit tcp any eq domain any
remark **permit LDAP and LDAPS traffic**
permit tcp any any eq 389
permit tcp any eq 389 any
permit tcp any any eq 636
permit tcp any eq 636 any
remark **permit RDP traffic**
permit tcp any any eq 3389
permit tcp any eq 3389 any
remark **permit PCoIP traffic**
permit tcp any eq 4172 any
permit tcp any eq 4172 any
permit udp any eq 4172 any
permit udp any eq 4172 any
```

This access list allows specific traffic to be included into the Medium QoS queue if it was not previously tagged.

```
ip access-list extended Queue_Medium
remark ** Deny heavy traffic and push down to low queue **
deny ip <BACKUP_SOURCE_NETWORK> <WILDCARD_MASK> <BACKUP_DESTINATION_NETWORK> <WILDCARD_MASK>
remark ** Deny CIFS traffic push to low queue **
deny tcp any any eq 139
deny tcp any eq 139 any
deny tcp any any eq 445
deny tcp any eq 445 any
remark ** Allow everything else **
permit ip any any
```


Appendix B: Reference Configurations (continued)
Cisco Quality of Service (QoS) Framework (continued)

! This access list allows the denied traffic from the Medium Queue to be included into the ! Low QoS queue if it was not previously tagged. If traffic is denied in the Low queue it ! will drop to the class default queue.
ip access-list extended Queue_Low
remark **Allow everything else**
permit ip any any

! The following configuration applies the 20M-shape QoS policy in the egress direction on ! a physical interface.
interface GigabitEthernet0/0/x
service-policy output 20M-shape

! The following configuration applies a QoS policy to a multipoint tunnel interface. This ! is important for a larger hub bandwidth location (Datacenter) to shape traffic to a ! smaller spoke bandwidth location (Branch). The spoke branch tunnel configuration will ! indicate the GROUP_NAME when it joins into the multipoint tunnel.
interface Tunnel<NUMBER_1>
ip nhrp map group <GROUP_NAME> service-policy output 10M-shape

-----END-----
Cisco Virtual Route Forwarding

This is an example configuration that was used to enable Virtual Route Forwarding (VRF). VRF is to Layer-3 what VLANs are to Layer-2. VRF allows us to create multiple isolated routing domains on a single router. Before we could leverage VRF we had to verify we had the appropriate hardware, software, and licensing. VRF capable devices have a native context called the “Global” or “Default” VRF. The Global VRF is commonly used in flat layer 3 environments that operate in the default Routing Information Base (RIB) instance. This is an example configuration of two VRF instances that cannot directly communicate with each other or the Global VRF; however, there are ways to allow full or restricted communication between VRFs instances. Understanding VRFs can allow a Network Architect to create scaled and segmented environments while utilizing the same physical hardware.

Layer3Switch#

! This command enables the routing table on layer3 switches.
ip routing

! This command creates the first VRF on top of the Global configuration
ip vrf <INSTANCE_NAME_1>
description <VRF_FUNCTION>

! The following command sets the Route Descriptor which is essentially a tag, similar to VLAN tags. In this sample we utilize an IP address followed by a signifier “yy”.
rd x.x.x.x:yy
route-target export x.x.x.x:yy
route-target import x.x.x.x:yy

! The following command sets the Route Descriptor which is essentially a tag, similar to VLAN tags. In this sample we utilize an IP address followed by a signifier “zz”.
ip vrf <INSTANCE_NAME_2>
description <VRF_FUNCTION>
rd x.x.x.x:zz
route-target export x.x.x.x:zz
route-target import x.x.x.x:zz

! The following configuration creates a VLAN layer 3 interface and assigns the interface to VRF instance “yy”
interface Vlan<NUMBER_1>
description VLAN<NUMBER_1> | <VLAN_NAME> | <IP_ADDRESS/MASK>
ip vrf forwarding <INSTANCE_NAME_1>
ip address <IP_ADDRESS> <NETMASK>

! The following configuration creates a VLAN layer 3 interface and assigns the interface to VRF instance “zz”
interface Vlan<NUMBER_2>
description VLAN<NUMBER_2> | <VLAN_NAME> | <IP_ADDRESS/MASK>
ip vrf forwarding <INSTANCE_NAME_2>
ip address <IP_ADDRESS> <NETMASK>

-----END-----
Appendix B: Reference Configurations (continued)

Cisco Dynamic Multipoint VPN Tunnels (DMVPN)

This is an example configuration that was used to enable Dynamic Multipoint VPN (DMVPN) tunnels. The configurations below use variables that are unique for each environment and these configurations will require adjustment and forethought prior to implementation. DMVPNs can be utilized on either public or private service provider connections. The primary data center tunnel(s) of this sample DMVPN configuration stay active while dynamic branch-to-branch VPN tunnels are created only when there is spoke-to-spoke traffic. This functionality provides a full-mesh topology without the need to build unique VPN tunnels for every branch-to-branch permutation. In this example a Class C network was used for the tunnel network interfaces to allow multipoint communication between Datacenter and Branch as well as Branch-to-Branch communication using the same tunnel interfaces.

```
Datacenter_1_Router_1#
! The configuration below is for the Primary router at Datacenter 1
!
! This loopback interface is the primary datacenter tunnel source interface.
interface Loopback<LOOPBACK_NUMBER_1>
  description Lo<LOOPBACK_NUMBER_1>| source for <TUNNEL_NUMBER_1>| VRF:<VRF_INSTANCE_NAME_1> | <IP_ADDRESS/CIDRMASK>
  ip address <LOOPBACK_1_IP_ADDRESS NETMASK>
!
! The following template is a preshared key-ring for DMVPN.
crypto keyring keys-<TUNNEL_NUMBER_1>
  local-address Loopback<LOOPBACK_NUMBER_1>
  pre-shared-key address <BRANCH_LOOPBACK_NETWORK> key <PRESHARED_KEY>
!
! The following commands provide the phase 1 crypto policy.
crypto isakmp policy 10
  encr aes 256
  authentication pre-share
  group 2
  crypto isakmp invalid-spi-recovery
  crypto isakmp keepalive 10
!
! The following is an isakmp profile to associate a remote address with a keyring
crypto isakmp profile wan-<TUNNEL_NUMBER_1>
  description wan crypto profile for <TUNNEL_NUMBER_1>
  keyring keys-<TUNNEL_NUMBER_1>
  match identity address <REMOTE_LOOPBACK_NETWORK NETMASK>
  local-address <LOOPBACK_NUMBER_1>
!
! The following are AES and SHA 256 transport layer and tunnel encryption sets for phase 2
crypto ipsec transform-set tunnel_AES_SHA2 esp-aes 256 esp-sha256-hmac
mode tunnel
crypto ipsec transform-set transport_AES_SHA2 esp-aes 256 esp-sha256-hmac
mode transport
!
! The following ipsec profile brings all the ipsec crypto policies together
crypto ipsec profile wan-<TUNNEL_NUMBER_1>
  set transform-set transport_AES_SHA2 tunnel_AES_SHA2
  set isakmp-profile wan-<TUNNEL_NUMBER_1>
```

Appendix B: Reference Configurations (continued)
Cisco Dynamic Multipoint VPN Tunnels (DMVPN) (continued)

! The following configuration template creates a Datacenter_1 primary hub DMVPN tunnel.
! The command arguments below will need to be verified prior to implementation.
interface Tunnel<TUNNEL_NUMBER_1>
  description <TUNNEL_NUMBER_1> | DC1 tunnel 1 | VRF:<INSTANCE_NAME_1> | <IP_ADDRESS/MASK>
  bandwidth 100000
  ip vrf forwarding <VRF_INSTANCE_NAME_1>
  ip address <IP_ADDRESS NETMASK>
  no ip redirects
  ip mtu 1400
  no ip split-horizon eigrp <EIGRP_INSTANCE_1>
  ip pim nbma-mode
  ip pim sparse-dense-mode
  ip nhrp map multicast dynamic
  ip nhrp map group 100M service-policy output 100M-shape
  ip nhrp map group 10M service-policy output 10M-shape
  ip nhrp map group 20M service-policy output 20M-shape
  ip nhrp network-id <NETWORK_ID_1>
  ip nhrp holdtime 600
  ip nhrp shortcut
  ip nhrp redirect
  ip tcp adjust-mss 1360
  load-interval 30
  delay 2000
  qos pre-classify
  tunnel source Loopback<LOOPBACK_NUMBER_1>
  tunnel mode gre multipoint
  tunnel key <NETWORK_ID_1>
  tunnel path-mtu-discovery
  tunnel bandwidth transmit 100000
  tunnel bandwidth receive 100000
  tunnel protection ipsec profile wan--<TUNNEL_NUMBER_1> shared

-----END-----
Appendix B: Reference Configurations (continued)
Cisco Dynamic Multipoint VPN Tunnels (DMVPN) (continued)

```
Datacenter_1_Router_2#  
! The configuration below is for the Secondary router at Datacenter 1  
! This loopback interface is the primary datacenter secondary tunnel source interface.  
interface Loopback<LOOPBACK_NUMBER_2>  
  description Lo<LOOPBACK_NUMBER_2>| source for <TUNNEL_NUMBER_2> | VRF:<VRF_INSTANCE_NAME_1> | <IP_ADDRESS/CIDRMASK>  
  ip address <LOOPBACK_2_IP_ADDRESS NETMASK>  
! The following template is a preshared key-ring for DMVPN.  
  crypto keyring keys-<TUNNEL_NUMBER_2>  
    local-address Loopback<LOOPBACK_NUMBER_2>  
    pre-shared-key address <BRANCH_LOOPBACK_NETWORK> key <PRESHARED_KEY>  
! The following commands provide the phase 1 crypto policy.  
  crypto isakmp policy 10  
    encr aes 256  
    authentication pre-share  
    group 2  
    crypto isakmp invalid-spi-recovery  
    crypto isakmp keepalive 10  
! The following is an isakmp profile to associate a remote address with a keyring  
  crypto isakmp profile wan-<TUNNEL_NUMBER_2>  
    description wan crypto profile for <TUNNEL_NUMBER_2>  
    keyring keys-<TUNNEL_NUMBER_2>  
    match identity address <REMOTE_LOOPBACK_NETWORK NETMASK>  
    local-address <LOOPBACK_NUMBER_2>  
! The following are AES and SHA 256 transport layer and tunnel encryption sets for phase 2  
  crypto ipsec transform-set tunnel_AES_SHA2 esp-aes 256 esp-sha256-hmac  
    mode tunnel  
  crypto ipsec transform-set transport_AES_SHA2 esp-aes 256 esp-sha256-hmac  
    mode transport  
! The following ipsec profile brings all the ipsec crypto policies together  
  crypto ipsec profile wan-<TUNNEL_NUMBER_2>  
    set transform-set transport_AES_SHA2 tunnel_AES_SHA2  
    set isakmp-profile wan-<TUNNEL_NUMBER_2>  
! 
```
Appendix B: Reference Configurations (continued)
Cisco Dynamic Multipoint VPN Tunnels (DMVPN) (continued)

! The following configuration template creates a Datacenter_1 secondary hub DMVPN tunnel. ! The command arguments below will need to be verified prior to implementation.

```
interface Tunnel<TUNNEL_NUMBER_2>
  description <TUNNEL_NUMBER_2> | DC1 tunnel 1 | VRF:<INSTANCE_NAME_1> | <IP_ADDRESS/MASK>
  bandwidth 100000
  ip vrf forwarding <VRF_INSTANCE_NAME_1>
  ip address <IP_ADDRESS NETMASK>
  no ip redirects
  ip mtu 1400
  no ip split-horizon eigrp <EIGRP_INSTANCE_1>
  ip pim nbma-mode
  ip pim sparse-dense-mode
  ip nhrp map multicast dynamic
  ip nhrp map group 100M service-policy output 100M-shape
  ip nhrp map group 10M service-policy output 10M-shape
  ip nhrp map group 20M service-policy output 20M-shape
  ip nhrp network-id <NETWORK_ID_2>
  ip nhrp holdtime 600
  ip nhrp shortcut
  ip nhrp redirect
  ip tcp adjust-mss 1360
  load-interval 30
  delay 2000
  qos pre-classify
  tunnel source Loopback<LOOPBACK_NUMBER_2>
  tunnel mode gre multipoint
  tunnel key <NETWORK_ID_2>
  tunnel path-mtu-discovery
  tunnel bandwidth transmit 100000
  tunnel bandwidth receive 100000
  tunnel protection ipsec profile wan-<TUNNEL_NUMBER_2> shared
!
-----END-----
Appendix B: Reference Configurations (continued)
Cisco Dynamic Multipoint VPN Tunnels (DMVPN) (continued)

**Datacenter_2_Router_1#**

! The configuration below is for the Primary router at Datacenter 2

! This loopback interface is the primary datacenter tunnel source interface.

```
interface Loopback<LOOPBACK_NUMBER_3>
   description Lo<LOOPBACK_NUMBER_3> | source for <TUNNEL_NUMBER_3> | VRF:<VRF_INSTANCE_NAME_1> | <IP_ADDRESS/CIDRMASK>
   ip address <LOOPBACK_3_IP_ADDRESS NETMASK>
```

! The following template is a preshared key-ring for DMVPN.

```
crypto keyring keys-<TUNNEL_NUMBER_3>
   local-address Loopback<LOOPBACK_NUMBER_3>
   pre-shared-key address <BRANCH_LOOPBACK_NETWORK> key <PRESHARED_KEY>
```

! The following commands provide the phase 1 crypto policy.

```
crypto isakmp policy 10
   encr aes 256
   authentication pre-share
   group 2
   crypto isakmp invalid-spi-recovery
   crypto isakmp keepalive 10
```

! The following is an isakmp profile to associate a remote address with a keyring

```
crypto isakmp profile wan-<TUNNEL_NUMBER_3>
   description wan crypto profile for <TUNNEL_NUMBER_3>
   keyring keys-<TUNNEL_NUMBER_3>
   match identity address <REMOTE_LOOPBACK_NETWORK NETMASK>
   local-address <LOOPBACK_NUMBER_3>
```

! The following are AES and SHA 256 transport layer and tunnel encryption sets for phase 2

```
crypto ipsec transform-set tunnel_AES_SHA2 esp-aes 256 esp-sha256-hmac
   mode tunnel
ncrypto ipsec transform-set transport_AES_SHA2 esp-aes 256 esp-sha256-hmac
   mode transport
```

! The following ipsec profile brings all the ipsec crypto policies together

```
crypto ipsec profile wan-<TUNNEL_NUMBER_3>
   set transform-set transport_AES_SHA2 tunnel_AES_SHA2
```

!
Appendix B: Reference Configurations (continued)
Cisco Dynamic Multipoint VPN Tunnels (DMVPN) (continued)

! The following configuration template creates a Datacenter_2 primary hub DMVPN tunnel.
! The command arguments below will need to be verified prior to implementation.

interface Tunnel<TUNNEL_NUMBER_3>
description <TUNNEL_NUMBER_3> | DC2 tunnel 1 | VRF:<INSTANCE_NAME_1> | <IP_ADDRESS/MASK>
bandwidth 100000
ip vrf forwarding <VRF_INSTANCE_NAME_1>
ip address <IP_ADDRESS NETMASK>
no ip redirects
ip mtu 1400
no ip split-horizon eigrp <EIGRP_INSTANCE_1>
ip pim nbma-mode
ip pim sparse-dense-mode
ip nhrp map multicast dynamic
ip nhrp map group 100M service-policy output 100M-shape
ip nhrp map group 10M service-policy output 10M-shape
ip nhrp map group 20M service-policy output 20M-shape
ip nhrp network-id <NETWORK_ID_3>
ip nhrp holdtime 600
ip nhrp shortcut
ip nhrp redirect
ip tcp adjust-mss 1360
load-interval 30
delay 2000
qos pre-classify
tunnel source Loopback<LOOPBACK_NUMBER_3>
tunnel mode gre multipoint
tunnel key <NETWORK_ID_3>
tunnel path-mtu-discovery
tunnel bandwidth transmit 100000
tunnel bandwidth receive 100000
tunnel protection ipsec profile wan-<TUNNEL_NUMBER_3> shared
!
-----END-----
Appendix B: Reference Configurations (continued)
Cisco Dynamic Multipoint VPN Tunnels (DMVPN) (continued)

Datacenter_2_Router_2# 
! The configuration below is for the Secondary router at Datacenter 2 
! This loopback interface is the primary datacenter tunnel source interface. 
interface Loopback<LOOPBACK_NUMBER_4> 
   description Lo<LOOPBACK_NUMBER_4> | source for <TUNNEL_NUMBER_4> | VRF:<VRF_INSTANCE_NAME_1> | <IP_ADDRESS/CIDRMASK> 
      ip address <LOOPBACK_4_IP_ADDRESS NETMASK> 
! 
! The following template is a preshared key-ring for DMVPN. 
crypto keyring keys-<TUNNEL_NUMBER_4> 
    local-address Loopback<LOOPBACK_NUMBER_4> 
    pre-shared-key address <BRANCH_LOOPBACK_NETWORK> key <PRESHARED_KEY> 
! 
! The following commands provide the phase 1 crypto policy. 
crypto isakmp policy 10 
   encr aes 256 
      authentication pre-share 
      group 2 
   crypto isakmp invalid-spi-recovery 
   crypto isakmp keepalive 10 
! 
! The following is an isakmp profile to associate a remote address with a keyring 
crypto isakmp profile wan-<TUNNEL_NUMBER_4> 
   description wan crypto profile for <TUNNEL_NUMBER_4> 
   keyring keys-<TUNNEL_NUMBER_4> 
   match identity address <REMOTE_LOOPBACK_NETWORK NETMASK> 
      local-address <LOOPBACK_NUMBER_4> 
! 
! The following are AES and SHA 256 transport layer and tunnel encryption sets for phase 2 
crypto ipsec transform-set tunnel_AES_SHA2 esp-aes 256 esp-sha256-hmac 
   mode tunnel 
crypto ipsec transform-set transport_AES_SHA2 esp-aes 256 esp-sha256-hmac 
   mode transport 
! 
! The following ipsec profile brings all the ipsec crypto policies together 
crypto ipsec profile wan-<TUNNEL_NUMBER_4> 
   set transform-set transport_AES_SHA2 tunnel_AES_SHA2 
   set isakmp-profile wan-<TUNNEL_NUMBER_4> 
!
Appendix B: Reference Configurations (continued)

Cisco Dynamic Multipoint VPN Tunnels (DMVPN) (continued)

! The following configuration template creates a Datacenter_2 secondary hub DMVPN tunnel. ! The command arguments below will need to be verified prior to implementation.

```
interface Tunnel<TUNNEL_NUMBER_4>
  description <TUNNEL_NUMBER_4> | DC2 tunnel 2 | VRF:<INSTANCE_NAME_1> | <IP_ADDRESS/MASK>
  bandwidth 100000
  ip vrf forwarding <VRF_INSTANCE_NAME_1>
  ip address <IP_ADDRESS_NETMASK>
  no ip redirects
  ip mtu 1400
  no ip split-horizon eigrp <EIGRP_INSTANCE_1>
  ip pim nbma-mode
  ip pim sparse-dense-mode
  ip nhrp map multicast dynamic
  ip nhrp map group 100M service-policy output 100M-shape
  ip nhrp map group 10M service-policy output 10M-shape
  ip nhrp map group 20M service-policy output 20M-shape
  ip nhrp network-id <NETWORK_ID_4>
  ip nhrp holdtime 600
  ip nhrp redirect
  ip tcp adjust-mss 1360
  load-interval 30
  delay 2000
  qos pre-classify
  tunnel source Loopback<LOOPBACK_NUMBER_4>
  tunnel mode gre multipoint
  tunnel key <NETWORK_ID_4>
  tunnel path-mtu-discovery
  tunnel bandwidth transmit 100000
  tunnel bandwidth receive 100000
  tunnel protection ipsec profile wan-<TUNNEL_NUMBER_4> shared

-----END-----
```
Appendix B: Reference Configurations (continued)
Cisco Dynamic Multipoint VPN Tunnels (DMVPN) (continued)

Branch_Router_1#
! The following DMVPN template is for a branch "spoke" router
! This loopback interface is the primary datacenter primary tunnel source interface.
interface Loopback<LOOPBACK_NUMBER_1>
   description Lo<LOOPBACK_NUMBER_1> | source for <TUNNEL_NUMBER_1> | VRF:<VRF_INSTANCE_NAME_1> | <IP_ADDRESS/CIDRMASK>
   ip address <LOOPBACK_1_IP_ADDRESS NETMASK>
!
! The following template is a preshared key-ring for DMVPN.
crypto keyring keys-<TUNNEL_NUMBER_1>
   local-address Loopback<LOOPBACK_NUMBER_1>
   pre-shared-key address <BRANCH_LOOPBACK_NETWORK> key <PRESHARED_KEY>
!
! The following commands provide the phase 1 crypto policy.
crypto isakmp policy 10
   encr aes 256
   authentication pre-share
   group 2
   crypto isakmp invalid-spi-recovery
   crypto isakmp keepalive 10
!
! The following is an isakmp profile to associate a remote address with a keyring
crypto isakmp profile wan-<TUNNEL_NUMBER_1>
   description wan crypto profile for <TUNNEL_NUMBER_1>
   keyring keys-<TUNNEL_NUMBER_1>
   match identity address <REMOTE_LOOPBACK_NETWORK NETMASK>
   local-address <LOOPBACK_NUMBER_1>
!
! The following are AES and SHA 256 transport layer and tunnel encryption sets for phase 2
crypto ipsec transform-set tunnel_AES_SHA2 esp-aes 256 esp-sha256-hmac
   mode tunnel
crypto ipsec transform-set transport_AES_SHA2 esp-aes 256 esp-sha256-hmac
   mode transport
!
! The following ipsec profile brings all the ipsec crypto policies together
crypto ipsec profile wan-<TUNNEL_NUMBER_1>
   set transform-set transport_AES_SHA2 tunnel_AES_SHA2
   set isakmp-profile wan-<TUNNEL_NUMBER_1>
!
! This loopback interface is the primary datacenter secondary tunnel source interface.
interface Loopback<LOOPBACK_NUMBER_2>
   description Lo<LOOPBACK_NUMBER_2> | source for <TUNNEL_NUMBER_2> | VRF:<VRF_INSTANCE_NAME_1> | <IP_ADDRESS/CIDRMASK>
   ip address <LOOPBACK_2_IP_ADDRESS NETMASK>
!
! The following template is a preshared key-ring for DMVPN.
crypto keyring keys-<TUNNEL_NUMBER_2>
   local-address Loopback<LOOPBACK_NUMBER_2>
   pre-shared-key address <BRANCH_LOOPBACK_NETWORK> key <PRESHARED_KEY>
!
Appendix B: Reference Configurations (continued)
Cisco Dynamic Multipoint VPN Tunnels (DMVPN) (continued)

! The following commands provide the phase 1 crypto policy.
crypto isakmp policy 10
  encr aes 256
  authentication pre-share
  group 2
crypto isakmp invalid-spi-recovery
crypto isakmp keepalive 10
!
! The following is an isakmp profile to associate a remote address with a keyring
crypto isakmp profile wan-<TUNNEL_NUMBER_2>
  description wan crypto profile for <TUNNEL_NUMBER_2>
  keyring keys-<TUNNEL_NUMBER_2>
  match identity address <REMOTE_LOOPBACK_NETWORK NETMASK>
  local-address <LOOPBACK_NUMBER_2>
!
! The following are AES and SHA 256 transport layer and tunnel encryption sets for phase 2
crypto ipsec transform-set tunnel_AES_SHA2 esp-aes 256 esp-sha256-hmac
  mode tunnel
crypto ipsec transform-set transport_AES_SHA2 esp-aes 256 esp-sha256-hmac
  mode transport
!
! The following ipsec profile brings all the ipsec crypto policies together
crypto ipsec profile wan-<TUNNEL_NUMBER_2>
  set transform-set transport_AES_SHA2 tunnel_AES_SHA2
  set isakmp-profile wan-<TUNNEL_NUMBER_2>
!
! This loopback interface is the secondary datacenter primary tunnel source interface.
interface Loopback<LOOPBACK_NUMBER_3>
  description Lo<LOOPBACK_NUMBER_3> | source for <TUNNEL_NUMBER_3> | VRF:<VRF_INSTANCE_NAME_1> | <IP_ADDRESS/CIDRMASK>
  ip address <LOOPBACK_3_IP_ADDRESS NETMASK>
!
! The following template is a preshared key-ring for DMVPN.
crypto keyring keys-<TUNNEL_NUMBER_3>
  local-address Loopback<LOOPBACK_NUMBER_3>
    pre-shared-key address <BRANCH_LOOPBACK_NETWORK> key <PRESHARED_KEY>
!
! The following commands provide the phase 1 crypto policy.
crypto isakmp policy 10
  encr aes 256
  authentication pre-share
  group 2
crypto isakmp invalid-spi-recovery
crypto isakmp keepalive 10
!
! The following is an isakmp profile to associate a remote address with a keyring
crypto isakmp profile wan-<TUNNEL_NUMBER_3>
  description wan crypto profile for <TUNNEL_NUMBER_3>
  keyring keys-<TUNNEL_NUMBER_3>
  match identity address <REMOTE_LOOPBACK_NETWORK NETMASK>
  local-address <LOOPBACK_NUMBER_3>
!
Cisco Dynamic Multipoint VPN Tunnels (DMVPN) (continued)

! The following are AES and SHA 256 transport layer and tunnel encryption sets for phase 2
crypto ipsec transform-set tunnel_AES_SHA2 esp-aes 256 esp-sha256-hmac
  mode tunnel
crypto ipsec transform-set transport_AES_SHA2 esp-aes 256 esp-sha256-hmac
  mode transport
|
! The following ipsec profile brings all the ipsec crypto policies together
crypto ipsec profile wan-<TUNNEL_NUMBER_3>
  set transform-set transport_AES_SHA2 tunnel_AES_SHA2
  set isakmp-profile wan-<TUNNEL_NUMBER_3>
|
! This loopback interface is the secondary datacenter secondary tunnel source interface.
interface Loopback<LOOPBACK_NUMBER_4>
  description Lo<LOOPBACK_NUMBER_4> | source for <TUNNEL_NUMBER_4> | VRF:<VRF_INSTANCE_NAME_1> | <IP_ADDRESS/CIDR MASK>
  ip address <LOOPBACK_4_IP_ADDRESS> <NETMASK>
|
! The following template is a preshared key-ring for DMVPN.
crypto keyring keys-<TUNNEL_NUMBER_4>
  local-address Loopback<LOOPBACK_NUMBER_4>
    pre-shared-key address <BRANCH_LOOPBACK_NETWORK> key <PRESHARED_KEY>
|
! The following commands provide the phase 1 crypto policy.
crypto isakmp policy 10
  encr aes 256
  authentication pre-share
  group 2
crypto isakmp invalid-spi-recovery
crypto isakmp keepalive 10
|
! The following is an isakmp profile to associate a remote address with a keyring
crypto isakmp profile wan-<TUNNEL_NUMBER_4>
  description wan crypto profile for <TUNNEL_NUMBER_4>
  keyring keys-<TUNNEL_NUMBER_4>
    match identity address <REMOTE_LOOPBACK_NETWORK> <NETMASK>
    local-address <LOOPBACK_NUMBER_4>
|
! The following are AES and SHA 256 transport layer and tunnel encryption sets for phase 2
crypto ipsec transform-set tunnel_AES_SHA2 esp-aes 256 esp-sha256-hmac
  mode tunnel
crypto ipsec transform-set transport_AES_SHA2 esp-aes 256 esp-sha256-hmac
  mode transport
|
! The following ipsec profile brings all the ipsec crypto policies together
crypto ipsec profile wan-<TUNNEL_NUMBER_4>
  set transform-set transport_AES_SHA2 tunnel_AES_SHA2
  set isakmp-profile wan-<TUNNEL_NUMBER_4>
|
Appendix B: Reference Configurations (continued)
Cisco Dynamic Multipoint VPN Tunnels (DMVPN) (continued)

interface Tunnel<TUNNEL_NUMBER_1>
description <TUNNEL_NUMBER_1> | DC1 tunnel 1 | VRF:<INSTANCE_NAME_1> | <IP_ADDRESS/MASK>
bandwidth 10000
ip vrf forwarding <VRF_INSTANCE_NAME_1>
ip address <TUNNEL_1_IP_ADDRESS NETMASK>
no ip redirects
ip mtu 1400
ip flow ingress
ip nat enable
ip nhrp group 10M
ip nhrp map multicast <DC1_ROUTER_1 LOOPBACK_NUMBER_1>
ip nhrp map <DC1_ROUTER_1_TUNNEL_1_IP_ADDRESS> <DC1_ROUTER_1 LOOPBACK_NUMBER_1>
ip nhrp network-id <NETWORK_ID_1>
ip nhrp holdtime 600
ip nhrp nhs <DC1_ROUTER_1_TUNNEL_1_IP_ADDRESS>
ip nhrp shortcut
ip nhrp redirect
ip virtual-reassembly in
ip tcp adjust-mss 1360
load-interval 30
delay 1000
qos pre-classify
tunnel source <LOOPBACK_NUMBER_1>
tunnel mode gre multipoint
tunnel key <NETWORK_ID_1>
tunnel path-mtu-discovery
tunnel protection ipsec profile wan-<TUNNEL_NUMBER_1> shared
!
interface Tunnel<TUNNEL_NUMBER_2>
description <TUNNEL_NUMBER_2> | DC1 tunnel 2 | VRF:<INSTANCE_NAME_1> | <IP_ADDRESS/MASK>
bandwidth 10000
ip vrf forwarding <VRF_INSTANCE_NAME_1>
ip address <TUNNEL_2_IP_ADDRESS NETMASK>
no ip redirects
ip mtu 1400
ip flow ingress
ip nat enable
ip nhrp group 10M
ip nhrp map multicast <DC1_ROUTER_2 LOOPBACK_NUMBER_1>
ip nhrp map <DC1_ROUTER_2_TUNNEL_2_IP_ADDRESS> <DC1_ROUTER_2 LOOPBACK_NUMBER_1>
ip nhrp network-id <NETWORK_ID_2>
ip nhrp holdtime 600
ip nhrp nhs <DC1_ROUTER_2_TUNNEL_2_IP_ADDRESS>
ip nhrp shortcut
ip nhrp redirect
ip virtual-reassembly in
ip tcp adjust-mss 1360
load-interval 30
delay 4000
qos pre-classify
tunnel source <LOOPBACK_NUMBER_2>
tunnel mode gre multipoint
tunnel key <NETWORK_ID_2>
tunnel path-mtu-discovery
tunnel protection ipsec profile wan-<TUNNEL_NUMBER_2> shared
!
Appendix B: Reference Configurations (continued)
Cisco Dynamic Multipoint VPN Tunnels (DMVPN) (continued)

```
interface Tunnel<TUNNEL_NUMBER_3>
description <TUNNEL_NUMBER_3> | DC2 tunnel 1 | VRF:<INSTANCE_NAME_1> |<IP_ADDRESS/MASK>
bandwidth 10000
ip vrf forwarding <VRF_INSTANCE_NAME_1>
ip address <TUNNEL_3_IP_ADDRESS NETMASK>
no ip redirects
ip mtu 1400
ip flow ingress
ip nat enable
ip nhrp group 10M
ip nhrp map multicast <DC2_ROUTER_1_LOOPBACK_NUMBER_3>
ip nhrp map <DC2_ROUTER_1_TUNNEL_3_IP_ADDRESS> <DC2_ROUTER_1_LOOPBACK_NUMBER_3>
ip nhrp network-id <NETWORK_ID_3>
ip nhrp holdtime 600
ip nhrp nhs <DC2_ROUTER_1_TUNNEL_3_IP_ADDRESS>
ip nhrp shortcut
ip nhrp redirect
ip virtual-reassembly in
ip tcp adjust-mss 1360
load-interval 30
delay 1000
qos pre-classify
tunnel source <LOOPBACK_NUMBER_3>
tunnel mode gre multipoint
tunnel key <NETWORK_ID_3>
tunnel path-mtu-discovery
    tunnel protection ipsec profile wan-<TUNNEL_NUMBER_3> shared
!
interface Tunnel<TUNNEL_NUMBER_4>
description <TUNNEL_NUMBER_4> | DC2 tunnel 2 | VRF:<INSTANCE_NAME_1> |<IP_ADDRESS/MASK>
bandwidth 10000
ip vrf forwarding <VRF_INSTANCE_NAME_1>
ip address <TUNNEL_4_IP_ADDRESS NETMASK>
no ip redirects
ip mtu 1400
ip flow ingress
ip nat enable
ip nhrp group 10M
ip nhrp map multicast <DC2_ROUTER_2_LOOPBACK_NUMBER_4>
ip nhrp map <DC2_ROUTER_2_TUNNEL_4_IP_ADDRESS> <DC2_ROUTER_2_LOOPBACK_NUMBER_4>
ip nhrp network-id <NETWORK_ID_4>
ip nhrp holdtime 600
ip nhrp nhs <DC2_ROUTER_2_TUNNEL_4_IP_ADDRESS>
ip nhrp shortcut
ip nhrp redirect
ip virtual-reassembly in
ip tcp adjust-mss 1360
load-interval 30
delay 4000
qos pre-classify
tunnel source <LOOPBACK_NUMBER_4>
tunnel mode gre multipoint
tunnel key <NETWORK_ID_4>
tunnel path-mtu-discovery
tunnel protection ipsec profile wan-<TUNNEL_NUMBER_4> shared
!
------END------
```
Appendix B: Reference Configurations (continued)

Cisco Layer 2 Tunneling Protocol Version 3 (L2TPv3, Also Known as Pseudo-Wire)

This is an example configuration used to illustrate how L2TPv3 can bridge traffic between two physical locations. L2TPv3 predates newer technologies such as VXLAN, OTV, and NVGRE, and is useful for devices that do not support these newer technologies. It is important to understand that L2TPv3 will cause Layer 2 loops if not deployed correctly. Typically, L2TPv3 bridge interfaces on a router are connected to a switch with Cisco’s LoopGuard feature enabled on the corresponding switch. The Cisco command for Loop Guard is ‘spanning-tree guard loop’ and it is applied to the connected interface.

**Note:** IP fragmentation and performance degradation can occur on traffic that traverses the bridge if the path between the data center routers does not support the Maximum Transmission Unit (MTU) packet size for the corresponding VLAN. Having a WAN path that supports large MTU (Jumbo) size frames can help prevent this fragmentation.

```
Datacenter_1_Router_1#
! The configuration below is for the Primary router at Datacenter 1
!
! This loopback interface is the primary datacenter primary bridge source interface.
interface Loopback<BRIDGE_LOOPBACK_1>
description Lo<BRIDGE_LOOPBACK_1>| source for <BRIDGE_LOOPBACK_1>| <IP_ADDRESS/CIDRMASK>
   ip address <BRIDGE_LOOPBACK_1_IP_ADDRESS NETMASK>
!
! The following template creates an L2TPv3 class and maps it to a Loopback
   pseudowire-class l2tpv3
   encapsulation l2tpv3
   sequencing both
   ip local interface Loopback<BRIDGE_LOOPBACK_1>
!
! The following commands apply the pseudo wire class to the physical interface. This
! essentially turns this interface into one side of a virtual cable.
interface GigabitEthernet0/0/x
description gi0/0/x | DC1 WAN Primary L2TP link | Access-port to <DEVICE_NAME>_<PORT_NUMBER>
   no ip address
   negotiation auto
   xconnect <BRIDGE_LOOPBACK_3_IP_ADDRESS> <BRIDGE_INSTANCE_1> encapsulation l2tpv3 pw-class l2tpv3
!
----END----
```

```
Datacenter_1_Router_2#
! This loopback interface is the primary datacenter secondary bridge source interface.
interface Loopback<BRIDGE_LOOPBACK_2>
description Lo<BRIDGE_LOOPBACK_2>| source for <BRIDGE_LOOPBACK_2>| <IP_ADDRESS/CIDRMASK>
   ip address <BRIDGE_LOOPBACK_2_IP_ADDRESS NETMASK>
!
! The following template creates an L2TPv3 class and maps it to a Loopback
   pseudowire-class l2tpv3
   encapsulation l2tpv3
   sequencing both
   ip local interface Loopback<BRIDGE_LOOPBACK_2>
!
! The following commands apply the pseudo wire class to the physical interface. This
! essentially turns this interface into one side of a virtual cable.
interface GigabitEthernet0/0/x
description gi0/0/x | DC1 WAN Secondary L2TP link | Access-port to <DEVICE_NAME>_<PORT_NUMBER>
   no ip address
   negotiation auto
   xconnect <BRIDGE_LOOPBACK_4_IP_ADDRESS> <BRIDGE_INSTANCE_2> encapsulation l2tpv3 pw-class l2tpv3
!
----END----
```
Cisco Layer 2 Tunneling Protocol Version 3 (L2TPv3, Also Known as Pseudo-Wire) (continued)

**Datacenter_2_Router_1#**

```
! The configuration below is for the Primary router at Datacenter 2
!
! This loopback interface is the Secondary datacenter primary bridge source interface.
interface Loopback<BRIDGE_LOOPBACK_3>
  description Lo<BRIDGE_LOOPBACK_3>| source for <BRIDGE_LOOPBACK_3>| <IP_ADDRESS/CIDR_MASK>
  ip address <BRIDGE_LOOPBACK_3_IP_ADDRESS NETMASK>
!
! The following template creates an L2TPv3 class and maps it to a Loopback
pseudowire-class l2tpv3
  encapsulation l2tpv3
  sequencing both
  ip local interface Loopback<BRIDGE_LOOPBACK_3>
!
! The following commands apply the pseudo wire class to the physical interface. This
! essentially turns this interface into one side of a virtual cable.
interface GigabitEthernet0/0/x
  description gi0/0/x | DC2 WAN Primary L2TP link | Access-port to <DEVICE_NAME>_<PORT_NUMBER>
  no ip address
  negotiation auto
  xconnect <BRIDGE_LOOPBACK_1_IP_ADDRESS> <BRIDGE_INSTANCE_1> encapsulation l2tpv3 pw-class l2tpv3
!
-----END-----
```

**Datacenter_2_Router_2#**

```
! This loopback interface is the primary datacenter secondary bridge source interface.
interface Loopback<BRIDGE_LOOPBACK_4>
  description Lo<BRIDGE_LOOPBACK_4>| source for <BRIDGE_LOOPBACK_4>| <IP_ADDRESS/CIDR_MASK>
  ip address <BRIDGE_LOOPBACK_4_IP_ADDRESS NETMASK>
!
! The following template creates an L2TPv3 class and maps it to a Loopback
pseudowire-class l2tpv3
  encapsulation l2tpv3
  sequencing both
  ip local interface Loopback<BRIDGE_LOOPBACK_4>
!
! The following commands apply the pseudo wire class to the physical interface. This
! essentially turns this interface into one side of a virtual cable.
interface GigabitEthernet0/0/x
  description gi0/0/x | DC2 WAN Secondary L2TP link | Access-port to <DEVICE_NAME>_<PORT_NUMBER>
  no ip address
  negotiation auto
  xconnect <BRIDGE_LOOPBACK_2_IP_ADDRESS> <BRIDGE_INSTANCE_2> encapsulation l2tpv3 pw-class l2tpv3
!
----END----
```
Appendix C: Bill of Materials

Branch Access Switches
Two Cisco SG-Series 52 port Gigabit Access Switches with Full-Power over Ethernet (per branch)

<table>
<thead>
<tr>
<th>QUANTITY</th>
<th>MFR PART #</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>SG500-52MP-K9-NA</td>
<td>52PORT GIGABIT MAX POE+ STACKABLE MANAGED SWITCH</td>
</tr>
</tbody>
</table>

Branch Out of Band Management Consoles
One Lantronix SLB Branch Office Manager (per branch)

<table>
<thead>
<tr>
<th>QUANTITY</th>
<th>MFR PART #</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SLB882KIT-15P</td>
<td>8 RJ45 serial console connections, 8 110v switched-receptacle outputs</td>
</tr>
</tbody>
</table>

Branch Routers
Two Redundant Cisco 2951 Integrated Services Routers (per branch)

Note: the following Bill of Materials came as a Bundle, which included Voice over IP and CUBE Licensing as well as Digital Signal Processors (DSPs); it also included a Primary Rate Interface (PRI) trunk interface. If we did not have a Cisco VoIP infrastructure the bottom four line items would be omitted from the BoM.

<table>
<thead>
<tr>
<th>QUANTITY</th>
<th>MFR PART #</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>C2951-VSEC/K9</td>
<td>CISCO Cisco 2951 Voice Sec. Bundle, PVDM3-32, UC&amp;SEC Lic,FL-CUBE10</td>
</tr>
<tr>
<td>2</td>
<td>PWR-2921-51-AC</td>
<td>CISCO Cisco 2921/2951 AC Power Supply</td>
</tr>
<tr>
<td>2</td>
<td>CAB-AC</td>
<td>CISCO AC Power Cord (North America), CI3, NEMA 5-15P, 2.1m</td>
</tr>
<tr>
<td>2</td>
<td>SL-29-SEC-K9</td>
<td>CISCO Security License for Cisco 2901-2951</td>
</tr>
<tr>
<td>2</td>
<td>MEM-2951-512MB-DEF</td>
<td>CISCO 512MB DRAM (1 512MB DIMM) for Cisco 2951 ISR (Default)</td>
</tr>
<tr>
<td>2</td>
<td>ISR-CCP-EXP</td>
<td>CISCO Cisco Config Pro Express on Router Flash</td>
</tr>
<tr>
<td>2</td>
<td>SL-29-IPB-K9</td>
<td>CISCO IP Base License for Cisco 2901-2951</td>
</tr>
<tr>
<td>2</td>
<td>MEM-CF-256MB</td>
<td>CISCO 256MB Compact Flash for Cisco 1900, 2900, 3900 ISR</td>
</tr>
<tr>
<td>6</td>
<td>HWIC-BLANK</td>
<td>CISCO Blank faceplate for HWIC slot on Cisco ISR</td>
</tr>
<tr>
<td>2</td>
<td>S2951UK9-15503M</td>
<td>CISCO Cisco 2951 IOS UNIVERSAL</td>
</tr>
<tr>
<td>2</td>
<td>PVDM3-32</td>
<td>CISCO 32-channel high-density voice DSP module</td>
</tr>
<tr>
<td>2</td>
<td>SL-29-UC-K9</td>
<td>CISCO Unified Communication License for Cisco 2901-2951</td>
</tr>
<tr>
<td>8</td>
<td>FL-CUBEE-5</td>
<td>CISCO Unified Border Element Enterprise License - 5 sessions</td>
</tr>
<tr>
<td>2</td>
<td>HWIC-ICE1T1-PRI</td>
<td>CISCO 1 port channelized T/EI and PRI HWIC (data only)</td>
</tr>
</tbody>
</table>
Appendix C: Bill of Materials (continued)

**Branch Host Servers: Infrastructure Class**

Two Redundant Cisco UCS-E160D-M2 blades (per branch) that fit inside our Cisco 2951 Integrated Services Routers, each running vSphere. This “branch in a box” gives us a fully redundant network and server infrastructure at each branch office supporting: Routing, DHCP, Network Access Control, and virtual instances of our Cisco Voice over IP Unified Communications Manager (call manager) and our Cisco Unity Express (voicemail).

<table>
<thead>
<tr>
<th>QUANTITY</th>
<th>MFR PART #</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>UCS-E160D-M2/K9</td>
<td>CISCO UCS-E, DoubleWide, 6 Core CPU 2.0, 2x8G SD, 1x8G RDIMM, 1-3 HDD</td>
</tr>
<tr>
<td>2</td>
<td>E100D-HDD-SAS900G</td>
<td>CISCO 900 GB, SAS hard disk drive for DoubleWide UCS-E</td>
</tr>
<tr>
<td>2</td>
<td>E100D-MEM-RDIMM8G</td>
<td>CISCO 8GB, 667MHz RDIMM/PC3-10600 2R for DoubleWide UCS-E</td>
</tr>
<tr>
<td>4</td>
<td>E100-FCPLT-BRKT</td>
<td>CISCO Bracket Used To Attach Faceplate to E100 Series Module</td>
</tr>
</tbody>
</table>
Appendix C: Bill of Materials (continued)

**Branch Host Servers: Domain Class**

One Dell R730xd rack-mounted server (per branch) used for Microsoft servers, to include: Surveillance servers and a Read-only, Stub, Active Directory Domain Controller.

<table>
<thead>
<tr>
<th>QUANTITY</th>
<th>MFR PART #</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>210-ADBC</td>
<td>PowerEdge R730xd Server</td>
</tr>
<tr>
<td>1</td>
<td>329-BCZK</td>
<td>PE R730/xd Motherboard MLK</td>
</tr>
<tr>
<td>1</td>
<td>330-BBCO</td>
<td>R730/xd PCIe Riser 2, Center</td>
</tr>
<tr>
<td>1</td>
<td>330-BBCR</td>
<td>R730/xd PCIe Riser 1, Right</td>
</tr>
<tr>
<td>1</td>
<td>540-BBBCB</td>
<td>Intel Ethernet i350 QP 1Gb Network Daughter Card</td>
</tr>
<tr>
<td>1</td>
<td>385-BBBCB</td>
<td>VFlash, 8GB SD Card for iDRAC Enterprise</td>
</tr>
<tr>
<td>1</td>
<td>385-BBHO</td>
<td>iDRAC8 Enterprise, integrated Dell Remote Access Controller, Enterprise</td>
</tr>
<tr>
<td>1</td>
<td>350-BBEX</td>
<td>Chassis with up to 12 + 4 Internal, 3.5” and 2, 2.5” Flex Bay Hard Drives</td>
</tr>
<tr>
<td>1</td>
<td>350-BBEJ</td>
<td>Bezel</td>
</tr>
<tr>
<td>1</td>
<td>384-BBBL</td>
<td>Performance BIOS Settings</td>
</tr>
<tr>
<td>1</td>
<td>800-BBDM</td>
<td>UEFI BIOS</td>
</tr>
<tr>
<td>1</td>
<td>780-BBLH</td>
<td>No RAID for H330/H730/H730P (1-24 HDDs or SSDs)</td>
</tr>
<tr>
<td>1</td>
<td>405-AAEH</td>
<td>PERC H730P Integrated RAID Controller, 2GB Cache</td>
</tr>
<tr>
<td>1</td>
<td>338-BFFO</td>
<td>Intel Xeon E5-2640 v3 2.6GHz,20M Cache,8.00GT/s QPI,Turbo,HT,8C/16T (90W) Max Mem 1866MHz</td>
</tr>
<tr>
<td>1</td>
<td>374-BBGV</td>
<td>Upgrade to Two Intel Xeon E5-2640 v3 2.6GHz,20M Cache,8.00GT/s QPI,Turbo,HT,8C/16T (90W)</td>
</tr>
<tr>
<td>6</td>
<td>370-ABUG</td>
<td>16GB RDIMM, 2133 MT/s, Dual Rank, x4 Data Width</td>
</tr>
<tr>
<td>1</td>
<td>370-ABUF</td>
<td>2133MT/s RDIMMs</td>
</tr>
<tr>
<td>1</td>
<td>370-AAAP</td>
<td>Performance Optimized</td>
</tr>
<tr>
<td>10</td>
<td>400-ANXG</td>
<td>10TB 7,2K RPM SATA 6Gbps 512e 3.5in Hot-plug Hard Drive</td>
</tr>
<tr>
<td>2</td>
<td>400-AMNG</td>
<td>800GB SSD SATA Read Intensive MLC 6Gbps 2.5in Flex Bay Drive, S3510</td>
</tr>
<tr>
<td>1</td>
<td>770-BBBQ</td>
<td>ReadyRails Sliding Rails Without Cable Management Arm</td>
</tr>
<tr>
<td>1</td>
<td>450-ADWS</td>
<td>Dual, Hot-plug, Redundant Power Supply (1+1), 750W</td>
</tr>
<tr>
<td>2</td>
<td>450-AALV</td>
<td>NEMA 5-15P to C13 Wall Plug, 125 Volt, 15 AMP, Power Cord</td>
</tr>
<tr>
<td>1</td>
<td>330-BBCL</td>
<td>Internal Dual SD Module</td>
</tr>
<tr>
<td>1</td>
<td>385-BBCF</td>
<td>Redundant SD Cards Enabled</td>
</tr>
<tr>
<td>2</td>
<td>385-BBIF</td>
<td>8GB SD Card For IDSDM</td>
</tr>
<tr>
<td>1</td>
<td>370-ABWE</td>
<td>DIMM Blanks for System with 2 Processors</td>
</tr>
<tr>
<td>2</td>
<td>374-BBHR</td>
<td>Heatsink for 12 + 4 Chassis PowerEdge R730xd</td>
</tr>
</tbody>
</table>
Appendix C: Bill of Materials (continued)

Data Center Core Distribution Switches
Four Cisco Nexus 93180YC-EX 10/25/40/50/100GB Switches (per data center), two reserved for storage traffic and two reserved for production traffic.

Note: The following Bill of Materials came as a Bundle, so while we didn’t need the sixteen “QSFP-40G-SR-BD” Optical SFP Modules, we got them for free. This BoM also includes eight Twinax-copper SFP cables used to interconnect the switches. (Twinax has less latency than optic).

<table>
<thead>
<tr>
<th>QUANTITY</th>
<th>MFR PART #</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>N9K-C93180YCEXB1BQ</td>
<td>CISCO 2 Nexus 93180YC-EX with 8 QSFP-40G-SR-BD</td>
</tr>
<tr>
<td>4</td>
<td>N9K-C93180YC-EX-B</td>
<td>CISCO Nexus 93180YC-EX bundle PID</td>
</tr>
<tr>
<td>4</td>
<td>N3K-C3064-ACC-KIT</td>
<td>CISCO Nexus 3K/9K Fixed Accessory Kit</td>
</tr>
<tr>
<td>4</td>
<td>NXOS-703I4.2</td>
<td>CISCO Nexus 9500, 9300, 3000 Base NX-OS Software Rel 7.0(3)4(2)</td>
</tr>
<tr>
<td>8</td>
<td>CAB-9K12A-NA</td>
<td>CISCO Power Cord, 125VAC 13A NEMA 5-15 Plug, North America</td>
</tr>
<tr>
<td>8</td>
<td>NXA-PAC-650W-PE</td>
<td>CISCO Nexus NEBs AC 650W PSU - Port Side Exhaust</td>
</tr>
<tr>
<td>16</td>
<td>NXA-FAN-30CFM-F</td>
<td>CISCO Nexus 2K/3K/9K Single Fan, port side exhaust airflow</td>
</tr>
<tr>
<td>4</td>
<td>N93-LIC-PAK</td>
<td>CISCO N9300 License PAK Expansion</td>
</tr>
<tr>
<td>4</td>
<td>N93-LAN1K9</td>
<td>CISCO LAN Enterprise License for Nexus 9300 Platform</td>
</tr>
<tr>
<td>8</td>
<td>QSFP-H40G-CU5M=</td>
<td>CISCO 40GBASE-CR4 Passive Copper Cable, 5m</td>
</tr>
<tr>
<td>16</td>
<td>QSFP-40G-SR-BD</td>
<td>CISCO QSFP40G BiDi Short-reach Transceiver</td>
</tr>
</tbody>
</table>

Data Center Core Access Switches
Two Cisco Catalyst 3850 Switches (per data center). These switches connect all 1Gb data center hosts.

<table>
<thead>
<tr>
<th>QUANTITY</th>
<th>MFR PART #</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>WS-C3850-48P-E</td>
<td>Cisco Catalyst 3850 48 Port PoE IP Services</td>
</tr>
<tr>
<td>4</td>
<td>CAB-C15-CBN</td>
<td>Cabinet Jumper Power Cord, 250 VAC 13A, C14-C15 Connectors</td>
</tr>
<tr>
<td>2</td>
<td>PWR-C1-715WAC/2</td>
<td>715W AC Config 1 Secondary Power Supply</td>
</tr>
<tr>
<td>2</td>
<td>C3850-NM-4-1G</td>
<td>Cisco Catalyst 3850 4 x 1GE Network Module</td>
</tr>
<tr>
<td>2</td>
<td>S3850U9K9-32-0SE</td>
<td>CAT3850 UNIVERSAL</td>
</tr>
<tr>
<td>2</td>
<td>STACK-T1-50CM</td>
<td>50CM Type 1 Stacking Cable</td>
</tr>
<tr>
<td>2</td>
<td>CAB-5PWR-30CM</td>
<td>Catalyst 3750X Stack Power Cable 30 CM</td>
</tr>
<tr>
<td>2</td>
<td>PWR-C1-715WAC</td>
<td>715W AC Config 1 Power Supply</td>
</tr>
</tbody>
</table>
Appendix C: Bill of Materials (continued)

Data Center Core Routers

Two redundant Cisco 4451 Integrated Service Routers (per data center)

**Note:** The following Bill of Materials came as a Bundle, which included Voice over IP and CUBE Licensing as well as Digital Signal Processors (DSPs). The bottom three line items only apply because we have a Cisco VoIP infrastructure.

<table>
<thead>
<tr>
<th>QUANTITY</th>
<th>MFR PART #</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>ISR4451-X-VSEC/K9</td>
<td>Cisco ISR 4451 VSEC Bundle PVDM4-64 w/ UC SEC Lic CUBE-25</td>
</tr>
<tr>
<td>4</td>
<td>PWR-4450-AC</td>
<td>450W AC Power Supply for Cisco ISR 4450</td>
</tr>
<tr>
<td>4</td>
<td>PWR-4450-AC/2</td>
<td>450W AC Power Supply (Secondary PS) for Cisco ISR 4450</td>
</tr>
<tr>
<td>8</td>
<td>CAB-C13-C14-2M</td>
<td>Power Cord Jumper C13-C14 Connectors 2 Meter Length</td>
</tr>
<tr>
<td>4</td>
<td>MEM-4400-DP-2G</td>
<td>2G DRAM (1 DIMM) for Cisco ISR 4400 Data Plane</td>
</tr>
<tr>
<td>4</td>
<td>SL-44-IPB-K9</td>
<td>IP Base License for Cisco ISR 4400 Series</td>
</tr>
<tr>
<td>4</td>
<td>MEM-4400-4G</td>
<td>4G DRAM (2G+2G) for Cisco ISR 4400</td>
</tr>
<tr>
<td>4</td>
<td>MEM-FLASH-8G</td>
<td>8G Flash Memory for Cisco ISR 4400</td>
</tr>
<tr>
<td>12</td>
<td>NIM-BLANK</td>
<td>Blank faceplate for NIM slot on Cisco ISR 4400</td>
</tr>
<tr>
<td>8</td>
<td>POE-COVER-4450</td>
<td>Cover for empty POE slot on Cisco ISR 4450</td>
</tr>
<tr>
<td>4</td>
<td>SL-44-SEC-K9</td>
<td>Security License for Cisco ISR 4400 Series</td>
</tr>
<tr>
<td></td>
<td>SISR4400UK9-311S</td>
<td>Cisco ISR 4451-X IOS XE UNIVERSAL</td>
</tr>
<tr>
<td>4</td>
<td>PVDM4-64</td>
<td>64-channel DSP module</td>
</tr>
<tr>
<td>4</td>
<td>SL-44-UC-K9</td>
<td>Unified Communication License for Cisco ISR 4400 Series</td>
</tr>
<tr>
<td>4</td>
<td>FL-CUBEE-25</td>
<td>Unified Border Element Enterprise License – 25 sessions</td>
</tr>
</tbody>
</table>
Appendix C: Bill of Materials (continued)

Data Center Host Servers: Infrastructure Class

Two Redundant Cisco UCS-E160D-M1 blades (per data center) that fit inside our Cisco 4451 Integrated Services Routers, each running vSphere. This gives us a fully redundant network and server infrastructure, to include: Routing, DHCP, Network Access Control, and virtual instances of our Cisco Voice over IP Unified Communications Manager (call manager), our Cisco Unity Express (voicemail), and our Cisco Contact Center Express (call center).

<table>
<thead>
<tr>
<th>QUANTITY</th>
<th>MFR PART #</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>E100D-HDD-SAS900G</td>
<td>900 GB SAS hard disk drive for DbleWide UCS-E</td>
</tr>
<tr>
<td>8</td>
<td>E100D-MEM-RDIM16G</td>
<td>16GB 1333MHz RDIMM/PC3-10600 2R for DoubleWide UCS-E</td>
</tr>
<tr>
<td>4</td>
<td>E100-8-16-MEM-UPG</td>
<td>Upgrades E140/160 First memory dimm from 8 to 16GB</td>
</tr>
<tr>
<td>4</td>
<td>E100-FCPLT-BRKT</td>
<td>Bracket Used To Attach Faceplate to E100 Series Module</td>
</tr>
<tr>
<td>4</td>
<td>E100-SD-8G</td>
<td>8 GB SD Card for SingleWide and DoubleWide UCS-E</td>
</tr>
<tr>
<td>4</td>
<td>UCS-E160D-M1/K9</td>
<td>UCS-EDoubleWide6CoreCPU2x8G SD1x8G RDIMM1-3 HDD</td>
</tr>
<tr>
<td>4</td>
<td>DISK-MODE-RAID-1</td>
<td>Configure hard drives as RAID 1 (Mirror)</td>
</tr>
</tbody>
</table>

Data Center WAN Optimizer

One Riverbed SteelHead CX7070-M WAN Optimizer (per data center)

<table>
<thead>
<tr>
<th>QUANTITY</th>
<th>MFR PART #</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SHA-7070-M</td>
<td>SteelHead CX7070-M (our 6050-H models have been replaced by the 7070)</td>
</tr>
</tbody>
</table>

Data Center VDI SAN

One Datrium 10GB DVX Server Powered SAN (per data center)

<table>
<thead>
<tr>
<th>QUANTITY</th>
<th>MFR PART #</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>D12x4</td>
<td>Datrium DVX NetShelf, 12 Bay 4TB</td>
</tr>
</tbody>
</table>

Data Center Server SAN

One Nimble Storage Hybrid SAN (per data center)

<table>
<thead>
<tr>
<th>QUANTITY</th>
<th>MFR PART #</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CS500-4G-36T-3200F</td>
<td>CS500, 2x1 GigE, Dual 10GbE SFP+, 12 x 3TB HDD, 4 x 800GB SSDs</td>
</tr>
</tbody>
</table>
Appendix C: Bill of Materials (continued)

**Note:** Depending on space and power requirements, we could have selected either a medium density rack-mounted solution or a high-density blade enclosure solution for our Enterprise Class Host Server servers. From a compute level, these are comparable systems, so we included both options here.

**Data Center Host Servers: Enterprise Class (Medium Density: 1U per Server 110V or 200V AC)**

Twelve Dell R630 Servers (per data center).

<table>
<thead>
<tr>
<th>QUANTITY</th>
<th>MFR PART #</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>210-ACXS</td>
<td>PowerEdge R630 Server</td>
</tr>
<tr>
<td>12</td>
<td>329-BCZI</td>
<td>PowerEdge R630 Motherboard MLK</td>
</tr>
<tr>
<td>12</td>
<td>340-AKPR</td>
<td>PowerEdge R630 Shipping - 10/24 Drive Chassis</td>
</tr>
<tr>
<td>12</td>
<td>555-BCKP</td>
<td>Intel X710 Quad Port 10Gb DA/SFP+ Ethernet, Network Daughter Card</td>
</tr>
<tr>
<td>12</td>
<td>385-BGCC</td>
<td>VFlash, 16GB SD Card for iDRAC Enterprise</td>
</tr>
<tr>
<td>12</td>
<td>385-BBHO</td>
<td>iDRAC8 Enterprise, integrated Dell Remote Access Controller, Enterprise</td>
</tr>
<tr>
<td>12</td>
<td>321-BBKN</td>
<td>Chassis with up to 6, 2.5” Hard Drives, 4 PCIe SSDs, 3 PCIe Slots</td>
</tr>
<tr>
<td>12</td>
<td>325-BBBI</td>
<td>Bezel 10/24 Drive Chassis</td>
</tr>
<tr>
<td>12</td>
<td>338-BJDX</td>
<td>Intel Xeon E5-2698 v4 2.2GHz,50M Cache,9.60GT/s QPI,Turbo,HT,20C/40T (135W) Max Mem 2400MHz</td>
</tr>
<tr>
<td>12</td>
<td>338-BJED</td>
<td>Intel Xeon E5-2698 v4 2.2GHz,50M Cache,9.60GT/s QPI,Turbo,HT,20C/40T (135W) Max Mem 2400MHz</td>
</tr>
<tr>
<td>144</td>
<td>370-ACNV</td>
<td>64GB LRDIMM, 2400MT/s, Quad Rank, x4 Data Width</td>
</tr>
<tr>
<td>12</td>
<td>370-ACPG</td>
<td>2400MT/s LRDIMMs</td>
</tr>
<tr>
<td>12</td>
<td>370-AAIP</td>
<td>Performance Optimized</td>
</tr>
<tr>
<td>12</td>
<td>461-AADP</td>
<td>PE Server FIPS TPM 1.2,CC</td>
</tr>
<tr>
<td>12</td>
<td>400-ADTI</td>
<td>PowerEdge Express Flash PCIeSSD Ready Configuration</td>
</tr>
<tr>
<td>12</td>
<td>770-BBBL</td>
<td>ReadyRails Sliding Rails With Cable Management Arm</td>
</tr>
<tr>
<td>12</td>
<td>450-ADWS</td>
<td>Dual, Hot-plug, Redundant Power Supply (1+1), 750W</td>
</tr>
<tr>
<td>24</td>
<td>450-AALV</td>
<td>NEMA 5-15P to C13 Wall Plug, 125 Volt, 15 AMP, 10 Feet (3m), Power Cord</td>
</tr>
<tr>
<td>12</td>
<td>330-BBCL</td>
<td>Internal Dual SD Module</td>
</tr>
<tr>
<td>12</td>
<td>385-BBCC</td>
<td>Redundant SD Cards Enabled</td>
</tr>
<tr>
<td>24</td>
<td>385-BBII</td>
<td>16GB SD Card For iDSDM</td>
</tr>
<tr>
<td>12</td>
<td>370-ABWE</td>
<td>DIMM Blanks for System with 2 Processors</td>
</tr>
<tr>
<td>24</td>
<td>412-AAEF</td>
<td>160W Heatsink for PowerEdge R630</td>
</tr>
<tr>
<td>24</td>
<td>SSDPEDM0D020T401</td>
<td>Intel® SSD DC P3700 Series (2.0TB, HHHL PCIe3x4, 20nm, MLC) NVMe drives</td>
</tr>
</tbody>
</table>
Appendix C: Bill of Materials (continued)

Data Center Host Servers: Enterprise Class (High Density: ½U per Server 200V AC)

Twelve Dell FC630 Servers (4 per chassis) in three FX2s Chassis (per data center).

<table>
<thead>
<tr>
<th>QUANTITY</th>
<th>MFR PART #</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>210-ABUX</td>
<td>PowerEdge FX2</td>
</tr>
<tr>
<td>3</td>
<td>321-BBFM</td>
<td>PowerEdge FX2 Chassis for up to 4 Half-Width Nodes</td>
</tr>
<tr>
<td>3</td>
<td>540-BBQM</td>
<td>Redundant Ethernet Switch Configuration</td>
</tr>
<tr>
<td>3</td>
<td>321-BBFZ</td>
<td>FX2S Chassis Configuration Label</td>
</tr>
<tr>
<td>3</td>
<td>321-BBGH</td>
<td>PowerEdge FX2S Chassis Configuration with Flexible I/O (up to 8 PCIe Slots)</td>
</tr>
<tr>
<td>3</td>
<td>403-BBEQ</td>
<td>SD Storage for Chassis Management Controller</td>
</tr>
<tr>
<td>3</td>
<td>634-0287</td>
<td>CMC Enterprise for FX2</td>
</tr>
<tr>
<td>3</td>
<td>770-BBER</td>
<td>FX2 ReadyRails Sliding Rails</td>
</tr>
<tr>
<td>3</td>
<td>450-ADTZ</td>
<td>Power Supply, Redundancy Alerting Enabled Configuration</td>
</tr>
<tr>
<td>6</td>
<td>492-BBEI</td>
<td>Power Cord, C20 to C19, PDU Style, 16A, 250V, 2ft (0.6m)</td>
</tr>
<tr>
<td>6</td>
<td>310-9696</td>
<td>Serial I/O Management Cable, for Ethernet Blade Switches</td>
</tr>
<tr>
<td>6</td>
<td>210-AHBX</td>
<td>Dell PowerEdge FN410S I/O Module, 8x Internal to 4x SFP+ external ports</td>
</tr>
<tr>
<td>6</td>
<td>634-BFBL</td>
<td>Software, Rights to use Full-Switch Mode, FN I/O Module</td>
</tr>
<tr>
<td>12</td>
<td>210-ACWK</td>
<td>PowerEdge FC630 Server Node</td>
</tr>
<tr>
<td>12</td>
<td>329-BCZL</td>
<td>PowerEdge FC630 Motherboard MLK</td>
</tr>
<tr>
<td>12</td>
<td>389-BESB</td>
<td>PowerEdge FC630 Regulatory Label, DAO</td>
</tr>
<tr>
<td>12</td>
<td>750-AADI</td>
<td>DSS7500 Consolidated Shipping</td>
</tr>
<tr>
<td>12</td>
<td>492-BBQT</td>
<td>PowerEdge FC PCIe Mezzanine Adapter</td>
</tr>
<tr>
<td>12</td>
<td>542-BCCF</td>
<td>Intel X710 Quad Port, 10Gb KR Blade Network Daughter Card</td>
</tr>
<tr>
<td>12</td>
<td>385-BHHO</td>
<td>iDRAC8 Enterprise, integrated Dell Remote Access Controller, Enterprise</td>
</tr>
<tr>
<td>12</td>
<td>385-BBBY</td>
<td>VFlash, 16GB SD Card for iDRAC Enterprise, V2</td>
</tr>
<tr>
<td>12</td>
<td>384-BBDS</td>
<td>Standard Cooling,FC630</td>
</tr>
<tr>
<td>12</td>
<td>384-BBBL</td>
<td>Performance BIOS Settings</td>
</tr>
<tr>
<td>12</td>
<td>338-BJDX</td>
<td>Intel Xeon E5-2698 v4 2.2GHz,50M Cache,9.60GT/s QPI,Turbo,HT,20C/40T (135W) Max Mem 2400MHz</td>
</tr>
<tr>
<td>12</td>
<td>338-BJED</td>
<td>Intel Xeon E5-2698 v4 2.2GHz,50M Cache,9.60GT/s QPI,Turbo,HT,20C/40T (135W) Max Mem 2400MHz</td>
</tr>
<tr>
<td>144</td>
<td>370-ACNV</td>
<td>64GB LRDIMM, 2400MT/s, Quad Rank, x4 Data Width</td>
</tr>
<tr>
<td>12</td>
<td>370-ACP6</td>
<td>2400MT/s LRDIMMs</td>
</tr>
<tr>
<td>12</td>
<td>370-AAIP</td>
<td>Performance Optimized</td>
</tr>
<tr>
<td>12</td>
<td>461-AADP</td>
<td>PE Server FIPS,TPM 1.2,CC</td>
</tr>
<tr>
<td>12</td>
<td>330-BBCV</td>
<td>Internal Dual SD Module</td>
</tr>
<tr>
<td>12</td>
<td>385-BBCF</td>
<td>Redundant SD Cards Enabled</td>
</tr>
<tr>
<td>24</td>
<td>385-BBII</td>
<td>16GB SD Card For IDSDM</td>
</tr>
<tr>
<td>12</td>
<td>374-BBHL</td>
<td>DIMM Blanks for System with 2 Processors</td>
</tr>
<tr>
<td>12</td>
<td>412-AAEJ</td>
<td>68MM Heatsink for PowerEdge FC630 Processor 1</td>
</tr>
<tr>
<td>12</td>
<td>412-AAEK</td>
<td>68MM Heatsink for PowerEdge FC630 Processor 2</td>
</tr>
<tr>
<td>24</td>
<td>750-AAFG</td>
<td>68MM Processor Heatsink Shroud for PowerEdge FC630</td>
</tr>
<tr>
<td>24</td>
<td>SSDPE2M0D020T401</td>
<td>Intel® SSD DC P3700 (2.0TB, 2.5in PCIe 3.0 x4, 20mm, MLC) NVMe drives</td>
</tr>
</tbody>
</table>
Appendix C: Bill of Materials (continued)

### Data Center Out of Band Management Consoles

One Lantronix SLC Advanced Console Manager (per data center)

<table>
<thead>
<tr>
<th>QUANTITY</th>
<th>MFR PART #</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SLC80482201S</td>
<td>Advanced Console Manager - 48 Ports RJ45, Dual AC Supply</td>
</tr>
</tbody>
</table>

### Data Center Archival Appliance

One EMC Data Domain 640 Archival Appliance (per data center)

<table>
<thead>
<tr>
<th>QUANTITY</th>
<th>MFR PART #</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>C-ES30-15-B</td>
<td>OPTION;ES30 SHELF;15X1TB HDD;2XLC</td>
</tr>
<tr>
<td>1</td>
<td>DD640-1E15</td>
<td>SYSTEM;DD640+1ES15;27TB;NFS;CIFS</td>
</tr>
<tr>
<td>1</td>
<td>DD640-12T-B-B</td>
<td>SYSTEM DD640-12X1;12T;NFS;CIFS</td>
</tr>
<tr>
<td>1</td>
<td>L-REP-640</td>
<td>LICENSE; REPLICATOR; DD640</td>
</tr>
<tr>
<td>1</td>
<td>U-BST-640</td>
<td>LICENSE;BOOST;DD640; UPGRADE</td>
</tr>
</tbody>
</table>

### Horizon Client USB Hardware Accessories

Used with the VMware Horizon Client for Microsoft Windows 7 and 10

<table>
<thead>
<tr>
<th>VENDOR</th>
<th>MFR PART #</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apricom</td>
<td>Secure Key 3.0</td>
<td>Horizon Client friendly encrypted USB mass storage</td>
</tr>
<tr>
<td>Addmaster</td>
<td>IJ7100</td>
<td>Teller Receipt Printer &amp; Validator</td>
</tr>
<tr>
<td>Burroughs</td>
<td>SSP2120100</td>
<td>SmartSource Check Scanner</td>
</tr>
<tr>
<td>Dell</td>
<td>B2360d-dn</td>
<td>Dell Duplex Laser Printer</td>
</tr>
<tr>
<td>Fujitsu</td>
<td>Fi-6160</td>
<td>Document Scanner</td>
</tr>
<tr>
<td>Fujitsu</td>
<td>Fi-6670</td>
<td>Document Scanner</td>
</tr>
<tr>
<td>Jabra</td>
<td>9470</td>
<td>USB/Bluetooth/Hook-Lift Touchscreen Headset</td>
</tr>
<tr>
<td>Logitech</td>
<td>C920</td>
<td>1080p Webcam</td>
</tr>
<tr>
<td>Logitech</td>
<td>CC3000E</td>
<td>1080p Conference Camera/Mic/Speaker</td>
</tr>
<tr>
<td>Magtek</td>
<td>IPAD 30050200</td>
<td>Debit Card PIN Pad</td>
</tr>
</tbody>
</table>
Appendix D: Performance Benchmarking

In our virtual environment, our servers consume the most processor time, but our CPUs are still significantly underutilized thanks to the high-core-density processors available today. In our virtual desktop infrastructure, memory has historically been the biggest barrier to growth, but today’s servers can accommodate terabytes of RAM, and the cost of memory is generally commoditized. And even with the most intensive workloads (backups included) we don’t come close to saturating our network thanks to 10Gb network interface cards. Storage continues to be the biggest barrier to performance, and for that reason we spent significant time benchmarking Shared Storage. To ensure that our testing accurately replicated our production environment, we monitored and averaged all our SAN traffic for a week to generate the realistic IO Profile seen in the table below, which was used to agnostically compare our new Datrium DVX to our existing Nimble Storage CS-420G-X8.

### Northrim IO Profile

<table>
<thead>
<tr>
<th></th>
<th>SEQUENTIAL</th>
<th>RANDOM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read Rate</td>
<td>75.97%</td>
<td>25%</td>
</tr>
<tr>
<td>Write Rate</td>
<td>24.03%</td>
<td>80%</td>
</tr>
</tbody>
</table>

### Testing Environment

For all tests we used the following configuration:

<table>
<thead>
<tr>
<th>TESTING ENVIRONMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>vSphere Hosts</td>
</tr>
<tr>
<td>Dell PowerEdge R820</td>
</tr>
<tr>
<td>Processors</td>
</tr>
<tr>
<td>Quad 2.6GHz 8-core E5-4620 v2 CPUs</td>
</tr>
<tr>
<td>Memory</td>
</tr>
<tr>
<td>512GB 2133 MHz RDIMMs</td>
</tr>
<tr>
<td>Network adapters</td>
</tr>
<tr>
<td>Intel 10Gb (two dedicated for storage IO)</td>
</tr>
<tr>
<td>Local storage</td>
</tr>
<tr>
<td>Dual 1TB Samsung 850Pro SSDs in HBA mode on a PERC 730P RAID card</td>
</tr>
<tr>
<td>vSphere version</td>
</tr>
<tr>
<td>6.0 update 1 Enterprise Plus</td>
</tr>
<tr>
<td>Guest VMs</td>
</tr>
<tr>
<td>Microsoft Windows 2012 r2</td>
</tr>
<tr>
<td>Guest vCPUs</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>Guest vRAM</td>
</tr>
<tr>
<td>8GB</td>
</tr>
<tr>
<td>Guest target drive</td>
</tr>
<tr>
<td>10GB (unformatted)</td>
</tr>
<tr>
<td>Bench-mark</td>
</tr>
<tr>
<td>IOmeter 1.1</td>
</tr>
<tr>
<td>SAN 1</td>
</tr>
<tr>
<td>Nimble Storage CS-420G-X8</td>
</tr>
<tr>
<td>SAN 2</td>
</tr>
<tr>
<td>Datrium DVX</td>
</tr>
</tbody>
</table>
Appendix D: Performance Benchmarking (continued)

Batch Processing Test

Our Batch Operator workstations place the most demanding workload on our shared storage because they read and write huge amounts of data stored in local temp files. For this reason, we have found them to be a good internal indicator of storage performance as seen in the table below:

<table>
<thead>
<tr>
<th></th>
<th>BATCH PROCESSING TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datrium</td>
<td>2 hours 53 minutes</td>
</tr>
<tr>
<td>Nimble Storage</td>
<td>7 hours 32 minutes</td>
</tr>
</tbody>
</table>

IOMeter Test

Based on the information in the Northrim IO Profile section above, IOMeter was configured to run with four Workers and thirty Outstanding IOps per target against an empty 10GB .vmdk on a vSphere 6 VMFS v5 volume using the following parameters:

- **Percent Random/Sequential Distribution Ratio**: 80% Random : 20% Sequential
- **Reply Size**: No Reply
- **Percent of Access Specification**: 100%
- **Burst Length**: 1 IOps
- **Burst Transfer Delay**: 0ms
- **Percent Read/Write Distribution Ratio**: 75% Read : 25% Write
- **Align IOps**: Request Size Boundaries
Appendix D: Performance Benchmarking (continued)

### TEST 1: NORTHRIM IO TEST RESULTS SINGLE HOST

<table>
<thead>
<tr>
<th>Transfer Request Size</th>
<th>Total IOps</th>
<th>Total MB/sec</th>
<th>Average I/O Response time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datrium 64KB</td>
<td>13,934</td>
<td>913</td>
<td>2.87</td>
</tr>
<tr>
<td>Nimble Storage 64KB</td>
<td>3,191</td>
<td>209</td>
<td>25.07</td>
</tr>
<tr>
<td>Datrium 8KB</td>
<td>26,258</td>
<td>215</td>
<td>1.52</td>
</tr>
<tr>
<td>Nimble Storage 8KB</td>
<td>21,373</td>
<td>175</td>
<td>3.74</td>
</tr>
</tbody>
</table>

### TEST 2: NORTHRIM IO TEST RESULTS TWO HOSTS

<table>
<thead>
<tr>
<th>Transfer Request Size</th>
<th>Total IOps</th>
<th>Total MB/sec</th>
<th>Average I/O Response time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datrium 64KB</td>
<td>24,553</td>
<td>1,609</td>
<td>9.77</td>
</tr>
<tr>
<td>Nimble Storage 64KB</td>
<td>16,197</td>
<td>1,012</td>
<td>29.63</td>
</tr>
<tr>
<td>Datrium 8KB</td>
<td>60,998</td>
<td>500</td>
<td>3.93</td>
</tr>
<tr>
<td>Nimble Storage 8KB</td>
<td>28,321</td>
<td>232</td>
<td>12.39</td>
</tr>
</tbody>
</table>

### TEST 3: NORTHRIM IO TEST RESULTS FOUR HOSTS

<table>
<thead>
<tr>
<th>Transfer Request Size</th>
<th>Total IOps</th>
<th>Total MB/sec</th>
<th>Average I/O Response time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datrium 64KB</td>
<td>61,851</td>
<td>4,054</td>
<td>7.76</td>
</tr>
<tr>
<td>Nimble Storage 64KB</td>
<td>12,480</td>
<td>780</td>
<td>19.23</td>
</tr>
<tr>
<td>Datrium 8KB</td>
<td>163,346</td>
<td>1,338</td>
<td>2.94</td>
</tr>
<tr>
<td>Nimble Storage 8KB</td>
<td>38,712</td>
<td>302</td>
<td>43.25</td>
</tr>
</tbody>
</table>

Our Testing Conclusions

In our initial test, it was immediately clear that the Datrium was significantly faster than the Nimble Storage when performing large (64K) block sequential writes, as opposed to smaller (8K) block writes. However, as we added more vSphere Hosts, it became clear that the Datrium impressively scaled in both large and small block sizes for, IOps, Throughput, and Latency. About half-way through our real-world testing we had this ah-ha moment of realization that we were focusing on the IOps and Throughput throughout our initial tests because that’s primarily what we used to compare array-based SAN performance. SAN latency wasn’t part of the conversation because it’s so often beyond the control of the SAN since it depends on the underlying network topology. However, our applications are particularly latency sensitive, and because the Datrium keeps active data cached on the host, our latency drops by an order of magnitude.
Appendix E: Network Optimization for PCoIP
VMware recommends that the network equipment used for traffic from the virtual machine to the endpoint is configured to use RED/WRED, as opposed to Tail Drop. Tail Drop can have a negative impact on PCoIP.

Refer to the network manufacturer’s guide on how to configure for RED/WRED, and to verify that Tail Drop is not currently being used.

The table below contains a summary of PCoIP latency and bandwidth requirements. Note the above-average bandwidth and below-average latency numbers.

<table>
<thead>
<tr>
<th></th>
<th>VMWARE VIEW 4.6</th>
<th>VMWARE VIEW 5.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latency requirements</td>
<td>Tolerate up to 50ms of latency before session degradation becomes noticeable.</td>
<td></td>
</tr>
<tr>
<td>Significant performance degradation for network conditions at 55ms.</td>
<td>Tolerate up to 55ms of latency before session degradation becomes noticeable.</td>
<td></td>
</tr>
<tr>
<td>Significant performance degradation for network conditions at 60ms.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bandwidth requirements</td>
<td>Requires a minimum bandwidth of 220Kbps per desktop session.</td>
<td>Requires a minimum bandwidth of 210Kbps per desktop session.</td>
</tr>
</tbody>
</table>
Appendix F: View Connection Server Architecture Redundancy

In each data center, a number of VMware Horizon Connection Servers are installed to provide scalability and availability. In the event of a data center outage, alternate Connection Servers are capable of taking on extra loads. The DR solution in this design is active-active, where all data center sites are used concurrently to provide virtual desktops.

Potential failure points and measures for redundancy are shown in the table below.

<table>
<thead>
<tr>
<th>FAILURE POINT</th>
<th>REDUNDANCY</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMware Horizon Manager™ Connection Server</td>
<td>At least two Horizon Manager Connection Servers are required for redundancy. These are load balanced using Cisco ACE. In the event of a failed Connection Server, users will not be disconnected from their session.</td>
</tr>
<tr>
<td>Horizon desktop</td>
<td>If a desktop fails then the user must report the fault to Support. A new desktop can be provisioned if the current desktop cannot be fixed. Users might lose work in this scenario.</td>
</tr>
<tr>
<td>VMware vCenter Server®</td>
<td>If vCenter Server fails, Horizon will not be affected. Virtual desktops can still be connected; however, new desktops cannot be provisioned. Workloads will not be balanced across clustered hosts. Desktops cannot be powered on or off. IT recommends that VMware vCenter Server Heartbeat™ is used for making vCenter server roles redundant.</td>
</tr>
<tr>
<td>VMware ESXi host</td>
<td>If a virtual desktop host fails, then a user will lose connection to their desktop. The virtual desktop will be automatically migrated to another host in the cluster and started. Users will be able to connect to their desktop within minutes. Users might lose work in this scenario. In this case, the virtual desktops will be powered off and deltas will be lost. If a Horizon Manager Server host fails, then the user’s session will be disconnected. A user can log back in and no work will be lost on their virtual desktop. IT recommends that all Horizon Manager Servers in the same instance are not placed on a single host. Anti-affinity rules for two Horizon Connection Servers are recommended.</td>
</tr>
<tr>
<td>VMware management cluster failure</td>
<td>If all hosts (hosting the VMware Horizon Manager Servers) in a VMware cluster lose connectivity or fail, then availability will be lost. Users that are currently connected will not be disconnected; however, users trying to authenticate to the Horizon Servers at time of failure will not succeed.</td>
</tr>
<tr>
<td>Horizon desktop cluster failure</td>
<td>If all hosts in a Horizon desktop cluster lose connectivity or fail, then users assigned to the desktop pools hosted on the affected cluster will be unable to access a virtual desktop until the cluster is restored.</td>
</tr>
</tbody>
</table>
### Appendix G: Client Hardware Specifications

Client hardware specifications will vary based on the amount of computing resources required by the end user. Typical specifications shown in the table below.

<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>SPECIFICATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware</td>
<td>VMware virtual hardware version 7</td>
</tr>
<tr>
<td>vCPU</td>
<td>2</td>
</tr>
<tr>
<td>vMemory</td>
<td>6GB</td>
</tr>
<tr>
<td>vNICs (VMXNET 3)</td>
<td>1</td>
</tr>
<tr>
<td>Virtual Network Adapter 1</td>
<td>Template-designated VLAN</td>
</tr>
<tr>
<td>Virtual SCSI Controller 0</td>
<td>LSI Logic</td>
</tr>
<tr>
<td>Virtual Disk OS: VMDK</td>
<td>50GB</td>
</tr>
<tr>
<td>Floppy, COM, and LPT</td>
<td>Disabled in BIOS</td>
</tr>
<tr>
<td>Virtual CD/DVD Drive 1</td>
<td>Enabled</td>
</tr>
</tbody>
</table>