Benchmarking a Scalable and Highly Available Architecture for Virtual Desktops

Ziya Aral
Jonathan Ely
DataCore Software

Summary

This paper reports on a configuration for Virtual Desktops (VDs) which reduces the total hardware cost to approximately $32.41 per desktop, including the storage infrastructure. This number is achieved using a configuration with dual node, cross-mirrored, High Availability storage. In comparison to previously published reports, which tout the storage infrastructure costs alone of VDI at from fifty to several hundred dollars per virtual machine, the significance of the data becomes self evident. In this report, storage hardware costs become inconsequential.

Even more significantly, the reported configuration achieves this result with 220 VDs running on a simple pair of low cost servers. The primary innovation consists in co-locating the redundant storage virtualization systems onto the same hardware platforms as the VDs, thus eliminating the need to amortize high-cost, stand alone storage controllers against many Virtual Server platforms and thousands of VDs. Previous publications have reported on configurations which use thousands of VDs to defray the cost of these controllers. Reading between the lines, it becomes immediately apparent that per VD hardware costs rise very sharply as such configurations are scaled downward. Yet, it is precisely these smaller VDI configurations which are the more important from most practical standpoints.

On the other hand, this configuration may also be scaled upwards, in a linear fashion, to thousands of Virtual Desktops, thus eliminating distended configurations created by the search for artificial "sweet spots" at which costs are optimized.

Finally, this configuration uses the VSI benchmark and is based on DataCore's SANmelody software and the Microsoft Hyper-V virtualization platform. The use of Hyper-V in such benchmarking is unusual as the VMware ESX platform is typically used for VDI sizing. DataCore expects similar results with ESX and will publish those results when they become available.

The Problem

Modern Hypervisors use Storage Area Networks (SANs) to provide the storage infrastructure for networks of virtual machines and hosts. Such shared infrastructures are used to provide portability, provisioning capability, and flexibility to VMs. Moreover, the need for such networks grows proportionally as the numbers of VMs are multiplied. As the number of eggs in “the basket” grows, so too does the need for High Availability configurations to keep them running and to prevent outages which encompass, not one or two, but perhaps dozens of machines. As VMs proliferate, routine management tasks, such as backup and migration, increasingly depend on the storage management functionality of SANs. Multiply these requirements by a factor of 10 or 100, and we arrive at the synergy between Virtual Desktops and SANs.

The problem for Virtual Desktop implementations is that SANs are often implemented with large and costly storage controllers and complex external storage networks. While these have the advantage of achieving reasonable scalability, they introduce a very large threshold cost to VDI implementations. To overcome this burden, hardware vendors typically benchmark with one to several thousand VDs.

Virtual Desktops are at their introductory stage. Many companies, while understanding the potential benefits of the technology, are introducing pilot programs or attempting to fit initial VDI implementations into existing structures. If the granularity of VDI implementations is to be in the thousands, then the user is forced to consume, not just the “whole loaf”, but an entire bakery truck full of loaves at one sitting … and this before even knowing whether the bread tastes good.

The alternative is equally unappetizing. The user “bites the bullet”, and accepts the high threshold cost and complexity of a full-blown SAN while running far less than the optimal number of VDs. Now, the per-desktop cost of the implementation becomes much larger than it would have been if the “old scheme” of discrete desktops had remained. This is quite an introduction to a new, “cost-saving” technology… as an increasing number of ever-practical bloggers have noted.

DataCore

This benchmark is run using the DataCore’s software-based storage virtualization environment on the same platform as the hypervisors hosting the Virtual Desktops. The DataCore software is a full featured implementation of a High-Availability SAN, with all of the functions necessary for a VM hosting infrastructure and unusually high performance characteristics. In addition to that, the software has several characteristics which recommend it for this application. Because the software is portable, it can run as a VM under Hyper-V or ESX and may also run on Windows Server 2008 at OS-level, alongside Hyper-V. While this means that it is a consumer of a common hardware resource, it also means that many of the most important SAN interconnects and storage structures (such as block cache) are implemented locally and virtually – on top of a common backplane. Such a scheme thus gains more from its locality to its clients than it "costs" by virtue of being a consumer for common hardware resources, as will be elaborated in the Performance Discussion section below,

In addition, the scheme described simplifies the overall configuration by scaling transparently – each time a new VDI host is added, so are the additional storage resources needed to feed it. Moreover, the scaling occurs in both directions, eliminating what is otherwise a difficult problem in appropriately sizing infrastructure.

None of this should be interpreted as a “tossing of the gauntlet” by software based controllers against hardware ones. DataCore often runs in conjunction with such hardware controllers. If the user has a need for such devices, then there is no fault. But, if there is no absolute requirement for such devices at the foundation of VDI architectures – if neither cost nor capability argues for their initial inclusion – then, hardware controllers serve only to skew the VDI architecture.

The Benchmark

The benchmark used is the Virtual Session Indexer (VSI) Pro 2.1 from Login Consultants. VSI is becoming the standard for this type of workload.

In comparison to the “roll-your-own” benchmarks used by several of the storage vendors, and, even in comparison to trace data, VSI appears to be just a shade more I/O heavy (and especially WRITE-weighted).

VSI also creates a working set which is somewhat larger than might be expected from discrete desktops. The explanation for this is that VSI runs every element of its emulated desktop environment on every individual VD. In addition, application synchronization across VDs is expressly minimized.

On the whole, this is easy to justify as simple workload normalization, necessary for the creation of a repeatable benchmarking environment. VSI, though, errs on the conservative side and is a bit more challenging than many proprietary workloads.

Configuration

The configuration used in this benchmark consists of two “white box” server nodes, each with an ASUS Z8PE-D18 Dual LGA motherboard, 2 Intel Xeon E5640 Westmere 2.66GHz Quad Core CPUs and an

80GB SATA boot drive. The total cost is $2114.14 per server, exclusive of DRAM and application storage. In addition, each node is configured with 64GB of DDR3 PC3-10600 DRAM (at a cost of $1056.24) and 4 application drives: 2 SAMSUNG HD103SJ SATA disk drives and 2 Western Digital VelociRaptor WD3000HLFS SATA disk drives ($393.60 for all 4). Total server hardware costs are thus $3564.68 per machine and $7,129.36 for the full configuration.

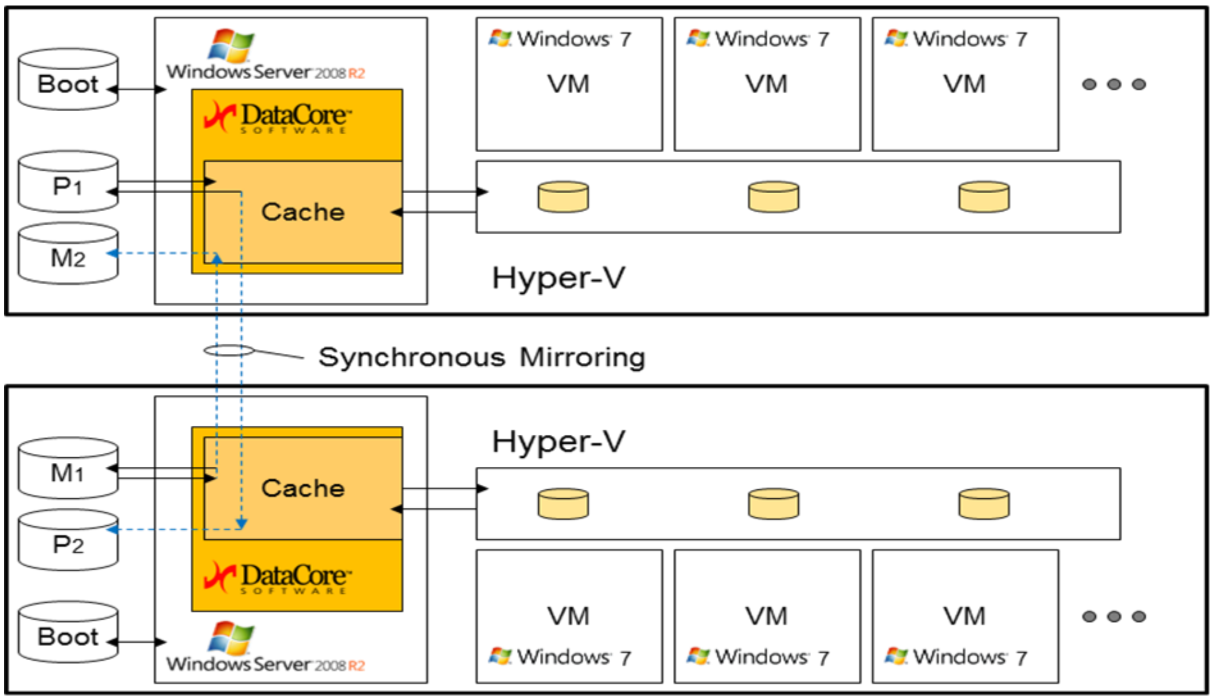
Two SATA drives are configured as a Storage Pool, belonging to SANmelody. SANmelody runs at OS level in this configuration. Whether SANmelody runs at this level, or as a Virtual Machine itself, makes little difference. In this case, the former was chosen for benchmark visibility.

The Storage Pool, once configured, is used to host a “golden image” of the VDs. It is then “snapped” using the DataCore Snapshot facility, and each of the writable “snaps” is presented as a boot LUN to the VDs. Experiments were tried to determine if multiple golden images might improve performance. Not only did they not help but the existence of multiple source images actually increased the amount of memory required to accommodate the working set in DataCore’s block cache.

Finally, the original image and its snapshots are mirrored to the second server (using DataCore synchronous mirrors over iSCSI) to achieve High Availability. Not only is the configuration thus proof against storage failure but the VDs may be restarted on any other server if a more catastrophic failure should occur. Two SATA drives are configured as a second storage pool to receive mirror traffic from the other node.

The configuration described above is implemented with a few simple scripts which then launch 110 Virtual Desktops per server and hand the entire configuration off to the VSI benchmark.

We live in an era in which the primary characteristics of computing machinery of any class are etched in silicon and are thus nearly identical for any member of that class. This configuration was chosen in order to create a baseline reference at the “lowest common denominator”. The choice of server platforms and their costs will vary by the substitution of everything from specific brand names to optimizations for footprint, power consumption, and certain enterprise “features”. Still, none of these variations will impact performance significantly, and may be seen as simple cost deltas to the reference presented here.



Results

The configuration above was able to host 220 Desktops using the VSI 2.0.1 benchmark. The disclosure and benchmark configuration report is reproduced in full as an addendum to this paper.

Performance Discussion:

Co-residency – The principle innovation in this benchmark result is the use of DataCore's storage virtualization system on the same hardware platforms used to host Virtual Desktops. While conventional wisdom might suggest that the DataCore software would thus become a competitor for the same scarce resources of the hardware platform, such as memory and CPU, numerous experiments proved the opposite to be true. In each case, such co-located configurations easily outperformed configurations with external storage.

The reasons are easy to isolate, retrospectively. The Virtual Desktop application is not particularly I/O intensive and proves to be easily served by DataCore at the cost of very few CPU cycles. The elimination of the lion's share of external channel traffic serves to further reduce those demands. In return, block-cache latencies become nearly nonexistent, channel overheads disappear, and I/O latencies are eliminated. The resident SANmelody "pays" for itself, without losing anything in the way of capability or portability.

From the standpoint of memory, a similar dynamic also holds. Virtual Desktops provide for relatively small working sets which tend to be READ heavy. Since all READs go to the single source volume, which in turn is the basis for the differential system volume for each Virtual Desktop, the entire grouping caches very well in a relatively small cache size. Larger caches produce diminishing returns.

Finally, one predictable advantage to these types of configurations is that they are inherently "self-tuning". Each time a group of Virtual Desktops are added, so too is the storage infrastructure necessary to support them. In contrast to generic SANs which are open ended in their requirements and often organized as independent, and sometimes complex, storage networks, in this case the known requirements of the VDI application may be used to make the storage infrastructure largely transparent.

Disks – Given efficient WRITE caching, required disk performance may be significantly minimized. In this benchmark, 4 SATA drives were used, configured into two separate DataCore 2-disk pools. The first pool was used for primary storage and the second was used for mirroring the alternate node. The use of a mix of 7,200 and 10,000 rpm devices is not significant. Any four (and perhaps even 3) 7,200 rpm SATA devices of similar characteristics to the Samsungs produced similar results.

The importance of this distinction for us is contained in a separate discussion on creating a baseline configuration. We were loath to use anything but the most standard, easily configured SATA components. SAS, Fibre Channel spindles, fast devices, hybrid disks, and SSDs all have their place in the real world and are well understood to deliver a multiple of ordinary SATA performance. Still, the incorporation of such devices in baseline benchmarking has the same effect as building VDI architectures around this or that storage array. Such optimizations may become the dozen different tails attempting to wag an increasingly confused dog. Not only do they make comparisons difficult, but they lead to the subtle intrusion of a specific hardware architecture into the realms in which the VDI requirements themselves should be paramount.

Memory – The cost of DRAM comprises a significant portion of the total platform costs in this configuration. Decreasing the size of the memory for each VD is thus a natural temptation for optimizing the number of VDs supported by each platform. The difficulty is that the one constant in our industry is the continuously changing density and cost structure of DRAM... a fact which argues for the avoidance of DRAM optimization, rather than arguing for it.

This difficulty is compounded by the fact that both Microsoft and VMware offer hypervisor features which allow the "over-loading" of the memory allocated to individual virtual machines. In effect, a kind of global virtual memory is created across "machines", though the actual implementation mechanisms may differ.

Finally, it was discovered during the course of DataCore's VSI benchmarking that VDs with very small memory allocations – so small that they would have prevented a discrete PC from booting - nevertheless ran normally in this configuration. On examination, it appeared that the performance of the DataCore block cache created a paging hierarchy which did nothing to prevent "thrashing" but did practically absorb it when a match with working set sizes was achieved.

In the end, neither "overloading" nor "fast thrashing" were used in the implementation described here. Not only did this allow for fair comparisons to benchmark data which had already been published, but it avoided optimization for schemes which are largely proprietary and independent of the VDI sizing exercise. In a word, the conventional allocation scheme used here avoided "diversions".

The actual DataCore cache size employed was 4 GBs. While larger caches produced somewhat better results, this was the point at which "diminishing returns" could be clearly established. Subtracting this usage and the memory allocation for Windows Server 2008, the remaining DRAM was simply divided by the number of resident VDs in a conventional fixed allocation.

Software Costs – It might be protested that since DataCore has used a software-based Storage Virtualization system in place of a hardware controller, that the cost of the software must be reported in order to create a true “apples-to-apples” comparison. This brings with it its own challenges. Most of the hardware controllers which have been benchmarked are fairly minimalist. Much of the feature set bundled into SANmelody is either not available or it is available at additional cost to the reported configurations.

With that qualification noted, the total cost of the software was $3848 per node or $34.98 per Virtual Desktop. This includes both the cost of Microsoft Server 2008 R2 Operating System, which hosts the SANmelody environment and the Hyper-V based VDs, and DataCore’s SANmelody itself.

The fully-burdened cost of the benchmarked configuration, including all of its infrastructural software and hardware, but excluding the OS and applications of the VDs, is thus $67.39 per Desktop.

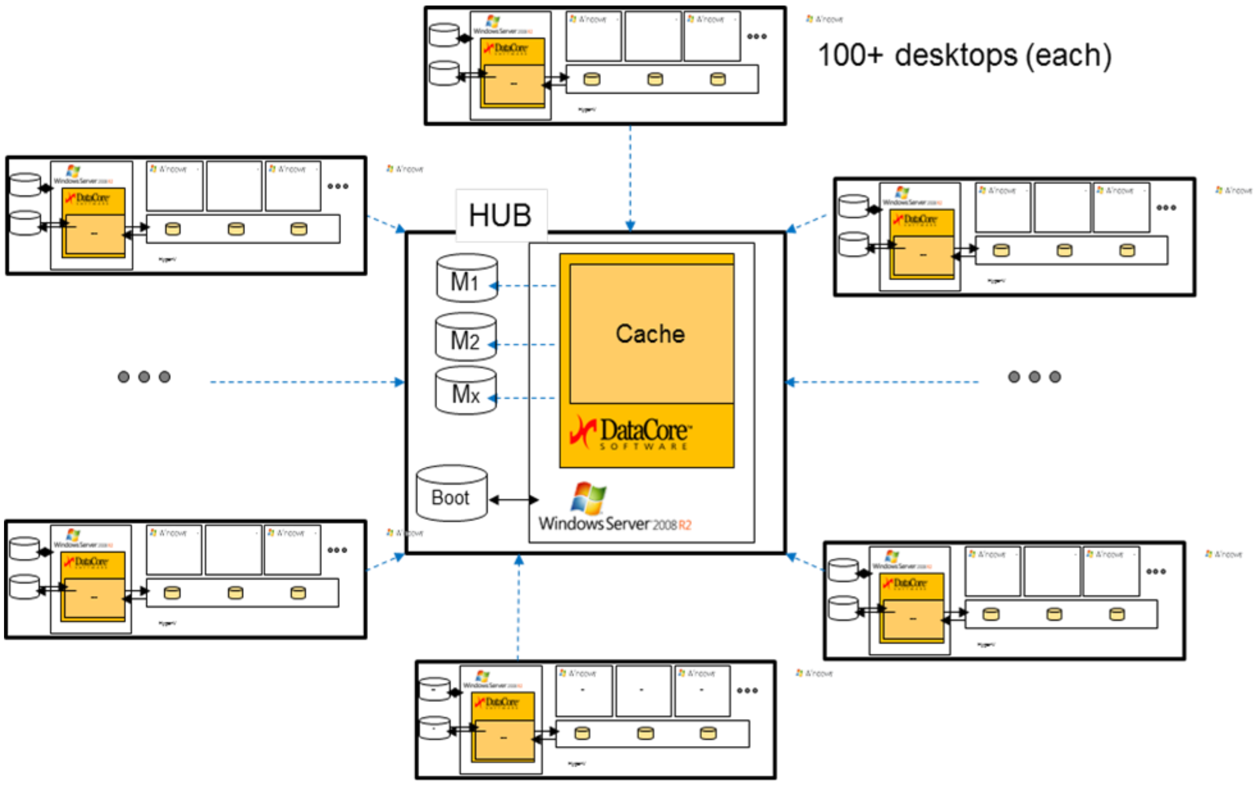
Scaling – As has been discussed, the real problem with VDs has been, not in scaling them up to “thousands” of Virtual Desktops but in scaling them down to practical configurations. It barely needs mentioning that this must occur without radically spiking costs at the low end and also without forgoing the SAN feature set which assures portability, availability, and data redundancy. Otherwise, the very benefits of VDI are compromised. The DataCore configuration, described herein, does a good job of maintaining low infrastructure costs at 220 Desktops by creating a paired server approach which simultaneously hosts both the VDs and the storage infrastructure.

But, what about scaling upward? How does one arrive at “thousands” of Desktops, if that is what is required, and more importantly, is it possible to arrive at such numbers without wildly varying cost increments and complex reconfiguration?

To achieve larger configurations, a DataCore topology known as a “Star” is employed. With the Star, the addition of a third node is used to create a Star center or hub. Additional nodes are then added to the Star. Each of the nodes of the Star acts as before, but they mirror their contents to the hub node rather than to each other. In turn, the hub becomes the single point of control for the entire configuration.

With this scheme, each additional server added as a point to the star brings with it its own storage devices and infrastructure as well as its VDI environment. The scheme is self tuning until the hub is no longer able to accommodate secondary mirror traffic. As an added benefit, most degenerative VDI phenomena, such as “boot storms”, remain contained in quite small and manageable modules without dominating the entire topology.

Simulation and initial experimentation indicates that a simple Star will scale performance (and per VD costs) to several thousand Desktops. DataCore will publish results to bolster these initial findings when they become available.



Conclusion and Future Directions

This paper reports on an architecture for VDI based on DataCore’s SANmelody, Microsoft’s Server 2008/Hyper-V, and standard server hardware. Login Consultant’s Virtual Session Indexer (VSI) is used as the benchmarking vehicle.

On the whole, the benchmarked architecture differs from most of the results published by other storage vendors. Instead of constructing a VDI configuration around a storage controller or pre-existing SAN architecture, it builds the SAN around the VDI servers themselves. In particular, the VDI servers also act as virtualization servers. The resulting locality produces results with as much as ten times the price/performance of other published results, while vastly reducing the number of VDs (220) at which such economics become available. The whole is accomplished on the simplest, “lowest common denominator” hardware. Nevertheless, the environment appears to be “self-tuning” in that it is resistant to major reevaluation as it scales upward, while it is also modular enough to be resilient in the face of hardware failures, reconfigurations and VDI anomalies, such as “boot storms”.

Where to from here?

Various optimizations immediately come to mind. In DataCore’s lab, we have successfully increased the density of VDs by more than 50%, to upwards of 175 Virtual desktops, on the same server platforms. Memory tweaks, enhanced hardware, and various tuning “tricks” produce the expected results. Still, increases in costs and complexity produce a system which is little better in price/performance than the one tested here. Sadly, for a group of “old tuners”, we have had to conclude that the benchmarked configuration is optimized enough to cross the threshold for practical VDI configurations and that additional low-level optimizations make only small, incremental changes.

What about high-level optimizations? Well, perhaps… The co-location of the host OS, the Hypervisor, and the virtualization server – alongside the Virtual Desktops – may provide opportunities for tighter integration. From our vantage point, however, that these remain largely independent today offers up an entirely different class of “optimization”.

VDI has one element in common with the other major computing movement of our day: cloud computing. Both technologies promise to aggregate very large numbers of “machines” of the same class… that is, they promise large numbers of platforms, the resource requirements of which are similar and which may be predicted before-hand. Instead of heterogeneous and unknown requirements, the promise is of homogeneous and known ones. In our world, what is “known” is also pre-configurable.

This creates the opportunity to present one additional level of virtualization and to “divide and conquer” the one overwhelming fact which immediately strikes anyone attempting to work with VDI: the sheer scale of discretely managing many component machines which are, nevertheless, irritatingly similar.

What if, instead of attempting to manage hundreds or thousands of virtual machines discretely, one could divide them into arbitrary groups - sub-units, “virtual datacenters”, or in DataCore parlance, “hives” – and then treat the entire hive as one unit? In the this case, a “hive” would consist of group of VDs (perhaps 50 or 75 of them), all grouped around a DataCore virtualization server and perhaps other servers needed to provide local services… all implemented as virtual machines, all configured to interact with each other and all without regard to any detailed knowledge about the underlying hardware environment. The entire hive would then be administered, moved, and managed as one unit with the virtualization server acting as a single “port” to the outside world. From the standpoint of the environment, all that would be needed would be a rough approximation of capacity: two “hives” over here and four of them, there…

This will be the future direction of our work.

Addendum

Benchmark Testing and Configuration

The performance benchmark used to determine the maximum number of Virtual Desktops in this configuration was Virtual Session Indexer (VSI) rev 2.1.2 by Login Consultants. Login VSI creates a simulated user workload (Outlook, Internet Explorer, Excel, etc…) on each of the desktops. The desktops report performance information to a common share folder which is analyzed after the completion of the benchmark run.

The simulated user workload is initiated by opening a Remote Desktop session to the virtual desktop. In this case, the VSI Launcher program was used in its default configuration to open the sessions, one desktop every 60 seconds. The user workload runs in a continuous 12 minute loop until all of the desktop sessions have been opened and the last one has run through a loop.

After all virtual desktops have run their simulated user workload simultaneously and reported their data to a share folder; the VSI Analysis tool was run. The tool compiles the data in Excel and produces a graph and tables showing response times indicating satisfactory or unsatisfactory user experience.

Configuration Information:

Virtual Desktops -

Microsoft Windows 7 Enterprise x86 – Configured with the default VSI optimizations and the optimizations recommended in Virtual Reality Check's whitepaper "Project VRC: Phase III" http://www.projectvrc.nl/

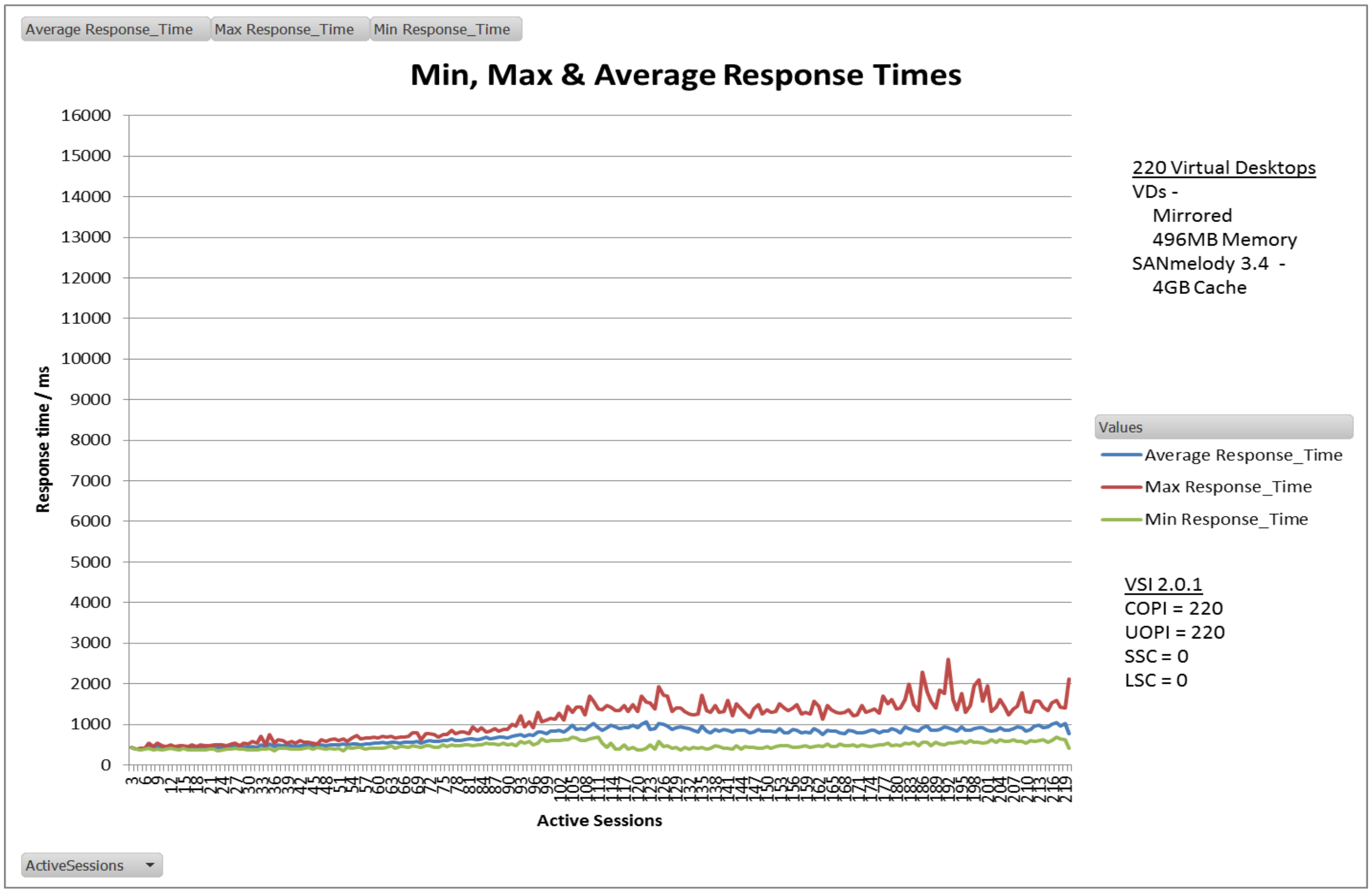
Benchmark –

Login Consultants VSI 2.1.2 http://www.loginconsultants.com/

Per VSI recommendation, seven VSI Launchers were used. All Launchers were running as VMs on a Hyper-V host independent from the hosts running the virtual desktops. The OS of these VMs was Microsoft Server 2008 R2.

DataCore SANmelody 3.4 http://www.datacore.com/

SANmelody was run with two bug fixes which will be included in a maintenance update scheduled for this quarter.



VSI Benchmark Analysis showing a successful run of 220 desktops.