Real-World Virtualization Testing on the HP ProLiant DL785 G5

A network and systems architecture consultant's review of HP's x86 8-socket server

By Paul Venezia
Independent Writer, Senior *contributing editor of the InfoWorld Test Center*
2/17/2009

Abstract
The basis for this paper is to detail the suite of tests run on the HP ProLiant DL785 G5 as a single-entity server and in a virtualized infrastructure, and also to detail some potential virtualized use case scenarios for placing the ProLiant DL785 G5 in production.

The virtualization platform focus for the purposes of this paper is on VMware's Virtual Infrastructure 3 and Linux-based virtual servers. However, much of the information contained herein has analogues in Windows services as well.

Overview

The history of symmetric multi-processing is long and varied. Well before multiple core CPUs were available, multiple socket servers were the only way to gather more processing power into a single server. The largest general-use SMP servers were eight CPU systems, with eight single-core CPUs, such as the HP ProLiant DL760. These were generally dedicated to a single task, such as large databases.

The processing power of those systems pales in comparison to modern dual-CPU, multi-core servers, but advances in SMP technology has brought about the possibility of an eight-CPU, multi-core server – the ProLiant DL785 G5.

HP ProLiant DL785 G5 is currently #1 for application performance in virtualized environments with VMmark, the industry's first virtualization benchmark for x86-based computers. [1]

Built on the AMD Opteron platform, the ProLiant DL785 G5 can run with up to eight 2.7GHz quad-core CPUs and up to 512GB of RAM. Even by modern standards, this is a hugely powerful system. In fact, it may be so powerful that many modern operating systems and applications cannot take full advantage of the capabilities of the DL785 G5.

Fortunately, this is where virtualization can be leveraged to get the most out of the hardware while providing for significant ease of management and lower power and cooling costs.

By breaking the raw power of the DL785 G5 into suitable sizes via the use of multiple virtual servers running above a single hypervisor, it's possible to get much more out of this server than would otherwise be possible by running the system as a single entity.

The testing conducted to support these statements is grounded in real-world scenarios involving Web applications, single-threaded processes and database performance. This provides a better picture of the potential placement of the ProLiant DL785 G5 with common workloads.

1 VMmark result as of 2/17/2009. VMMark result can be found at http://www.vmware.com/products/vmmark/results.html.

Background
It's important to understand that in many cases, application and service performance does not increase when more CPUs are added to the physical server hosting the service. If applications and services are single-threaded, or cannot make use of more than a few threads at any time, adding more CPUs to the server cannot improve the speed at which these applications and services operate.

However, adding more CPUs to the server can enable more services and applications to run simultaneously. This means that the more CPUs, cores, and threads present in a single physical server, the more distinct services and applications can be run at one time without negatively impacting one another.

The potential issue here is that when built as a single entity, these applications and services must still run from a single kernel and are built on a single operating system, which may be problematic from a configuration and delivery standpoint.

For instance, it may be possible to run several dozen high-demand Web applications on a single operating system running on the DL785 G5, but the requirements of those applications may overlap or be otherwise incompatible with each other, reducing the overall effectiveness of the solution, and potentially becoming significant problems when issues caused by one application or service negatively impacts another. One example of this might be a highly-visible legacy Web application that requires a specific Oracle client installation that conflicts with newer applications.

Thus, in those instances a better use of the hardware is to break it into discrete blocks of resources dedicated to specific instances via virtualization. In this way, it's not only easier to make full use of the available resources, but by segmenting each application, you remove the potential for a problem or issue in one application instance from corrupting another.

The reality is that adept planning and understanding of the technology can result in a ProLiant DL785 G5 running dozens of high-powered virtual servers on a single hardware platform with all the benefits of individual servers in addition to the benefits of virtualization. Snapshotting, ease of backups and restoration, templates, and VM automation are a few of the features available in VMware Virtual Infrastructure 3, and can be of significant benefit in any organization.

Methods
The testing platform was a ProLiant DL785 G5 with eight AMD Opteron 8360 SE quad-core CPUs running at 2.5GHz per core, 128GB RAM, two 72GB 10k RPM 2.5" SAS drives in a RAID1+0 array driven by an HP SmartArray P400i RAID controller with 256MB RAM and a battery-backed write cache, and an HP MSA70 disk array driven by an HP SmartArray P800 RAID controller with 512MB RAM and a battery-backed write cache. Network connectivity was provided by the two built-in NC371i gigabit NICs. No other hardware was used in testing.

The virtualization platform used was VMware's Virtual Infrastructure 3, with VMware ESX 3.5 Update 3 and Virtual Center 2.5 Update 2. All virtual machines were built on Red Hat Enterprise Linux 5.

The database tests were conducted using MySQL 5.0.45, the version that is included with RHEL5. The database server was configured commensurate to the resources available to the virtual machine, and all database performance tested used InnoDB databases exclusively. All database tests were driven by the sql-bench SQL benchmarking tools.

The Web tests were conducted using a WordPress 2.7 installation that was artificially populated with over 10,000 posts, 30,000 comments, dozens of categories, and other trappings of a popular site. WordPress was chosen due to its popularity and the fact that it's a good example of a modern LAMP application that might be found in any number of companies.

All other tests were conducted using tools within the RHEL5 operating system itself, such as bzip2, gzip, tar, and md5, and some third-party tools such as LAME. These tests were designed to determine the benefits – if any – to using the ProLiant DL785 G5 as a virtualization platform rather than as a single-entity server.

The virtual servers were built as either single- dual- or quad-CPU systems with at least 2GB RAM. The Web servers were built with two CPUs and 2GB RAM. The database servers were built with four CPUs each and 8 or 16GB RAM.

The standalone single-thread task servers were each built with 8GB RAM and four CPUs.

Test overviews and results

Test Section 1 - MySQL Database Testing

The first series of tests focused on MySQL performance. Specfically, MySQL's InnoDB transactional storage engine. MySQL is generally regarded as the best open-source database server available today, and while it lacks certain features found in many high-end database servers such as Oracle and Microsoft SQL Server, it forms the foundation of thousands of popular applications and tools.

To this end, MySQL InnoDB performance was measured using the sql-bench innotest1 methods, which generate a very large number of concurrent mixed operations on the database server. Performance was measured in time to completion of all three concurrent passes.

Two sets of tests were conducted, the first with the ProLiant DL785 G5 built as a single entity server running Red Hat Enterprise Linux 5, and the stock MySQL 5.0.45 package included with the distribution. These tests were conducted on an otherwise quiescent server, with all 32 cores and 128GB RAM available to the MySQL process.

The goals of this testing were to determine the impact that virtualizing the identical workload had on performance. The theory was that by breaking the same workload down into multiple MySQL processes running on discrete virtual servers, the performance per instance would be increased over a single entity build running the same number of concurrent transactions.

There were significant modifications made to the MySQL configuration, specifically boosting the performance of InnoDB transactions. InnoDB was the preferred engine, and key parameters were modified, such as increasing the values of query, buffer and thread cache sizes.

Indeed, further tweaking and adjustments of the MySQL configuration may result in some performance increases, and may reduce these times even further, but they are sufficient for the purposes of this paper.

With MySQL running, the innotest tests were run concurrently, and were modified to record the time required to complete each pass. In all, six full runs of each test iteration were conducted with this hardware configuration, and the best times were used for this portion of the test. See Table 1.1 for the results of this testing.

Table 1.1 – Non-virtualized MySQL InnoDB test results, ProLiant DL785 G5

Test	Time
Innotest1	350s
Innotest1a	156s
Innotest1b	93s

These results constitute a baseline for all subsequent tests, but do not equal the same workload as presented to the virtual servers in later testing.

The next series of tests were run against the single-entity server, and attempted to replicate the same workload that would be seen by the virtualized tests. This workload is identical to the first test series, but eight tests were run concurrently. This is roughly the same workload that would be seen by the ProLiant DL785 G5 in the virtualized tests, but against the server as a single entity, and a single MySQL process. See Table 1.2 for the results of this testing.

Table 1.2 Eight concurrent MySQL InnoDB test results, ProLiant DL785 G5

Test	Time
Innotest1	1177s
Innotest1a	885s
Innotest1b	371s

As you can see, the results of this test show significantly worse performance than the initial baseline results, likely due to the single MySQL server process attempting to handle a vast number of operations on multiple tables simultaneously.

Following these tests, the ProLiant DL785 G5 was built with VMware ESX Server 3.5 U3, and eight Red Hat Enterprise Linux 5 virtual machines were built from a template, each with four CPUs and 16GB of RAM. This provides essentially a one-to-one comparison to the ProLiant DL785 G5 single-entity configuration, utilizing all 32 cores and all 128GB of RAM.

A test harness was constructed to ensure that each test VM was configured exactly the same way, and that all tests would run simultaneously. The MySQL configuration used for each VM was identical to that used for the single-entity tests.

As with the prior tests, six passes were run, and the best numbers from all tests were used for comparison. See Table 1.3 for the results of this testing.

Table 1.3 Virtualized MySQL InnoDB test results, eight VMs, ProLiant DL785 G5

Test	Time
Innotest1	380s
Innotest1a	284s
Innotest1b	156s

The results of these tests show that even though the time required to complete the tests was higher than the first non-virtualized tests, the server actually conducted eight times the number of operations in the virtualized testing compared to that initial test. The apples-to-apples comparison to the results in Table 1.2 show that virtualizing this workload resulted in much better performance overall.

Some elaboration may be required to fully understand this concept. The non-virtualized test referenced in Table 1.1 ran the InnoDB tests against a single database process, exactly like the virtualized tests. However, the virtualized tests actually ran eight sets of the same test concurrently, against eight separate MySQL servers on eight separate virtual machines. Thus, the total workload of the virtualized tests was eight times that of the initial non-virtualized tests.

This shows that by virtualizing this workload, the server is able to handle more raw work than it can as a single entity. Much of this disparity is likely to be due to the fact that scaling a single service to 32 cores is not as efficient as breaking that service into eight equal parts and running them as individual entities on the same hardware.

The next series of tests were conducted using a single four CPU, 16GB RAM virtual machine, and the same tests were run. This test was used to gauge the impact of the other virtual machines on the performance of a single virtual machine. This virtual machine was functionally identical to the single-entity configuration, albeit with one-eighth the resources. See Table 1.4 for the results of this testing.

Table 1.4 Virtualized MySQL InnoDB test results, single VM, ProLiant DL785 G5

Test	Time
Innotest1	384
Innotest1a	283
Innotest1b	154

These test results are essentially identical to the results of the eight-VM run. This would appear to show that the performance impact of other virtual machines running concurrently with no oversubscription is negligible at best. In short, having seven other virtual machines running at capacity did not significantly modify the performance of the eighth virtual machine.

Following these series of tests, two more virtual machines were cloned from the database test image and all ten identical virtual machines were booted. The purpose of this test was to determine the effects of significant oversubscription of resources on the same test scenario. With the original virtualized tests, each of the four CPUs assigned to the eight virtual machines corresponded to a single physical CPU core. By adding two more four-CPU virtual machines, the ProLiant DL785 G5 was oversubscribed by 25%, and thus there was no one-to-one core matchup between the physical and virtual hardware.

With all ten virtual machines running, the same InnoDB tests were run concurrently, with six test passes each. With this test, the breadth of the results was such that they should be presented as best and worst. Thus, the results shown are the absolute best and worst of a single run. See Table 1.5 for the results of this testing.

Table 1.5 Oversubscribed virtualized MySQL InnoDB test results, ProLiant DL785 G5

Test	Best Time	Worst Time
Innotest1	418s	651s
Innotest1a	317s	519s
Innotest1b	168s	271s

These results show a server under significant load. With the 25% oversubscription, some virtual machines were able to produce times only 8 to 10 percent slower than the single virtual machine tests, while others battled for resources and produced times nearly 100% slower than the single VM tests. However, these tests do show that it's possible to significantly oversubscribe a virtualized workload on the ProLiant DL785 G5 and still achieve quite reasonable results from several of the systems, even if all virtual machines are running at capacity. In the real world, it's highly unlikely that ten four-CPU virtual machines would all be tasked at 100% capacity simultaneously, and if so, it would then be obvious that more capacity was required.

Test Section 2 - Web Application Testing

As previously described, the Web application testing utilized a WordPress 2.7 installation that had been populated with over 10,000 posts, 30,000 comments, tags, and other trappings of a high-volume, highly popular Web application. The tests conducted were request based, using a variety of load generators to place load on dynamic and static aspects of the site.

In order to test Web delivery of the ProLiant DL785 G5 in a virtualized environment, a complete Web application module was constructed, consisting solely of virtual machines. The servers used were a single MySQL server with four CPUs and 8GB RAM. There was a two-CPU, 2GB RAM NFS server, a single-CPU 1GB RAM load balancer, and a number of Web front-end servers, each with two CPUs and 2GB RAM.

The MySQL server was built from the same template as the servers used in the MySQL tests, and scripts were written to populate the WordPress database with posts, comments, tags, categories, and so forth. The source for the content was taken from randomizing entries in the Linux /usr/share/dict/words file. All entries were of random length between 200 and 1200 words, and all comments lengths were randomized between 10 and 300 words. Category and tag associations were also randomized, as were titles for each post.

The data was added to the database with the same structure as actual posts would be added, and thus all associated functions of WordPress were applicable, including calendar, category, and tag searches.

The code for the site itself resided on the NFS server, in an export that was then mounted on each Web server. This allowed the Web servers to be instantly added and subtracted from the front-end pool, and appropriate entries added and subtracted from the load balancer configuration when appropriate.

The Web servers were built using Apache 2.2.3, distributed with RHEL5. There were some modifications of the stock configuration made, specifically pertaining to connection limits, keepalives, and max threads. In addition, eAccelerator, an opportunistic caching engine was installed to increase PHP performance. Otherwise, all was left essentially stock.

The load balancer was built using LVS, utilizing the LVS-DR mechanism and standard round-robin balancing. An externally-generated load was presented to this configuration, designed to hit a random assortment of dynamic pages, essentially mimicking a standard load. For each test run, 100,000 randomized page loads were requested, maintaining at least 36 concurrent requests. This constitutes a very heavy load on a dynamic Web application.

The first set of tests were conducted on the ProLiant DL785 G5 as a single-entity server. Thus, the database and Web server were run on the same RHEL5 installation, and all 32 cores and 128GB RAM were available to these processes. There was no load balancer used, since there was only a single server to test. See Table 2.1 for the results of this testing.

Table 2.1 Single-entity Web application testing, ProLiant DL785 G5

Test	Result
Page requests per second	346.29
Mean time per request	17ms

These results show that Apache scales quite well to the full 32 cores available on the ProLiant DL785 G5. At the culmination of each test run, however, the 5-minute load on the server was nearing 20, which indicates a server under extreme load.

The next series of tests were conducted on the virtualized environment described earlier. The first set of tests used three load-balanced Web servers with two CPUs and 2GB RAM each. See Table 2.2 for the results of this testing.

Table 2.1 Virtualized Web application testing, three Web servers, ProLiant DL785 G5

Test	Result
Page requests per second	131.19
Mean time per request	45.734ms

These results reflect a significant performance reduction versus the single-entity tests, and the main reason was that the front-end Web servers became CPU bound, as they were only using six cores between them. The next series of tests ran against the same application stack, but with four front-end Web servers. See Table 2.3 for the results of this testing.

Table 2.3 Virtualized Web application testing, four Web servers, ProLiant DL785 G5

Test	Result
Page requests per second	161.52
Mean time per request	37.147ms

These results show that by adding another two-CPU front-end server, requests per second increased by over 20%, and the mean time between requests dropped by 20%. That constitutes a significant performance boost by adding another two cores running another virtual server.

The next set of tests were identical, adding yet another front-end server for a total of five two-CPU, 2GB RAM front-end Web servers with the same back-end database server and load balancer. See Table 2.4 for the results of this testing.

Table 2.4 Virtualized Web application testing, five Web servers, ProLiant DL785 G5

Test	Result

Page requests per second	186.36
Mean time per request	33.084ms

By adding another front-end Web server the page requests per second increased by nearly 20%, and the mean time per request decreased by over 10%. This isn't as big of an increase as the four-server test, and reflects the fact that adding servers may not result in a linear performance improvement. Regardless, the performance gains were still quite significant. It is expected that adding another five front-end Web servers would result in performance roughly equal to that of the single-entity tests.

Adding front-end servers to this infrastructure design is extremely simple. With the Web content served via NFS and a basic virtual machine template, it takes only a minute or so to provision and deploy a new Web server into the load-balanced farm. Thus this provides a simple way to adapt to changing load requirements without shutting down virtual servers to modify their resource settings. In production, the act of adding these servers could easily be scripted using the VMware APIs.

As with any Web application of this nature, the dynamic and static content of the application plays a major role in determining sufficient resources to run the application. A higher percentage of truly dynamic requests will necessarily require that more CPU time and disk I/O is used to deliver the pages, while content that is either static or cached from a dynamic call will be delivered faster and require less CPU and disk I/O. The intent of this testing was to test a spread of these parameters in every test, as most Web applications will have a mixture of the two, at least to some degree.

Based on these results, a significant number of medium-load Web applications can be run on a single ProLiant DL785 G5. Should one application require more front-end performance, additional Web servers can be easily added to the load-balanced pool as needed, or the resources granted to those virtual servers could be increased.

Test Section 3 - Single-thread tests
In addition to the database and Web test scenarios, some baseline single-thread tests were run to judge the performance of the ProLiant DL785 G5 under significant loads with many discrete single-thread processes.

These tests included simple audio conversion tests, compression tests, and MD5 calculation tests. The audio portion of these tests used the LAME MP3 conversion utility to produce an MP3 file from a 157MB source WAV file. The compression tests used bzip2 and gzip to compress and decompress the resulting 55MB MP3 file, and the MD5 sum tests were comprised of calculating MD5 sums of an 850MB WAV file.

As with the MySQL and Web tests, each test run was conducted on the ProLiant DL785 G5 running Red Hat Enterprise Linux 5 as a single-entity server, and running VMware ESX Server 3.5 with virtual servers.

Unlike the MySQL and Web tests, these processes are single-threaded and cannot make use of more than one CPU core per process. Thus, the goal was to gauge performance when running a single instance of each test as well as when running 32 concurrent instances, and when running 40 concurrent instances of each test. This measured the single-thread, single-core performance as well as performance when every core was tasked, and then when the number of concurrent processes eclipsed the total number of available CPU cores.

In order to run these tests under VMware VI3, eight four-CPU virtual servers were used for the 32-process tests, and ten four-CPU virtual servers were used for the 40-process tests.

Audio Conversion Tests
These tests were designed to replicate a common task, converting a WAV file into an MP3 file. This process is very CPU intensive and a good indicator of raw processing power.

A test harness was constructed to run a certain number of concurrent LAME conversions, gauging the performance at each test level.

The first test was a single pass for each test. This would stress only one CPU core at a time. See Table 3.1 for the results of this test.

Table 3.1 Single thread, single process, single-entity tests, ProLiant DL785 G5

Test	Time
LAME conversion	90s
Gzip compress	2s
Bzip2 compress	7s
Gzip decompress	2s
Bzip2 decompress	4s

MD5 Sum	0.75s

As noted above, these are the baseline numbers for a single-process run of all tests.

The next test case was to run the same tests 32 times, concurrently. This would stress all 32 cores on the ProLiant DL785 G5 simultaneously. See Table 3.2 for the results of this test.

Table 3.2 Single thread, 32 process, single-entity tests, ProLiant DL785 G5

Test	Time
LAME conversion	92s
Gzip compress	4s
Bzip2 compress	9s
Gzip decompress	8s
Bzip2 decompress	12s
MD5 Sum	6s

These results show that the ProLiant DL785 G5 came reasonably close to the single-process results on those tasks that were predominately CPU bound, and slowed down somewhat on the other tests. This is likely due to the disk I/O required for each test. Faster storage would likely show a performance benefit for the compression and decompression tests, as well as the MD5 tests.

The next single-entity test was to run the same number of tests, but oversubscribed to 40 concurrent processes per test. This would show the performance when there was significant resource contention. See Table 3.3 for results of this test.

Table 3.3 Single thread, 40 process, single-entity tests, ProLiant DL785 G5

Test	Time
LAME conversion	115s
Gzip compress	6s
Bzip2 compress	15s
Gzip decompress	11s
Bzip2 decompress	14s
MD5 Sum	7s

As you can see, the audio compression tests suffered significantly, requiring roughly 20% more time to complete than the 32-process test. The difference was not as marked with the other tests, but they too slowed down considerably due to competition for available resources.

The next set of tests were run with virtual servers. The first set were identical to the single-entity, single-process tests, to provide a comparison to the single-entity baseline. The virtual server used was a 4 CPU, 8GB RAM RHEL5 system. See Table 3.4 for the results of this test.

Table 3.4 Single thread, single process virtualized tests, ProLiant DL785 G5

Test	Time
LAME conversion	96s
Gzip compress	2s
Bzip2 compress	6.5s
Gzip decompress	2s
Bzip2 decompress	4s
MD5 Sum	0.75s

Comparing these results to the single-entity, single-process tests, we see that the results are roughly identical, though slightly slower than the raw tests. This shows that the performance penalty for virtualizing these workloads is negligible.

The next set of tests were run across eight 4-CPU 8GB RAM virtual servers. A test harness was constructed to launch all tests simultaneously, and measure the performance of a one-to-one core to virtual server CPU load, with 32 independent processes run across all eight virtual servers. See Table 3.5 for the results of this test.

Table 3.5 Single thread, 32 process virtualized tests, ProLiant DL785 G5

Test	Time
LAME conversion	100s
Gzip compress	3s
Bzip2 compress	10s
Gzip decompress	2s
Bzip2 decompress	7s
MD5 Sum	1s

These results show that most tests results were affected by a 5-10% performance degradation due to the number of concurrent threads in use at any given time. However, these results are certainly comparable to the 32-process single-entity results, and again show a negligible performance difference between the single-entity and virtualized workloads.

The last series of single-thread tests were the oversubscription of the virtualized servers. This test scenario used ten 4-CPU virtual servers, each running four single-threaded tasks simultaneously. See Table 3.6 for the results of this test.

Table 3.5 Single thread, 40 process virtualized tests, ProLiant DL785 G5

Test	Time
LAME conversion	122s
Gzip compress	6s
Bzip2 compress	14s
Gzip decompress	2s

Bzip2 decompress	10s
MD5 Sum	2s

These results show that the contention for available resources produced a times roughly 10-15% slower than the virtualized 32-process test, but only 5% slower than the non-virtualized tests. This suggests that the ProLiant DL785 G5 performs better as a single-entity server for applications that have a large number of single-threaded tasks. However, the performance penalty for virtualizing that workload is not significant.

Testing Conclusion

Throughout the various tests conducted on the HP ProLiant DL785 G5, it became clear that the raw processing power present in the server was more than most services and applications can actually take advantage of. There are a number of highly threaded applications and HPC frameworks that have been developed to harness large numbers of independent CPU cores, generally in large clusters of smaller servers connected with high-speed interconnects such as Infiniband.

In addition, large numbers of single-threaded tasks favored the single-entity build of the ProLiant DL785 G5 rather than the virtualization build. This is due to the nature of singular single-threaded applications and their resource requirements. Even so, the virtualized testing showed a relatively minor performance penalty in these tests.

Beyond those applications it appears that using the ProLiant DL785 G5 as a virtualization platform provides a much better return on investment in real-world infrastructures, and makes better use of the resources available in this class of server. The ability to expand and contract resources dedicated to any virtual server, the RAM oversubscription and resource sharing present in VMware VI3, and the management tools available in VMware VirtualCenter greatly increase the roles that the ProLiant DL785 G5 can play, as well as the size of the infrastructure that can make use of the ProLiant DL785 G5.

Virtualization Use Case Scenarios for the HP ProLiant DL785 G5

Triple Play

Perhaps one of the more obvious scenarios for the Proliant DL785 G5 is a Web production-dev-test layout. With the raw horsepower available with the DL785 G5, it can easily handle a significant number of high traffic Web-based virtualized applications, and simultaneously run the dev and test environments for those applications.

As an example, perhaps a Web application requires a front-end Web server, a rules engine server, and a back-end database server. This “pod” could easily be built with virtualization, and then cloned to templates for the dev and test pods.

By leveraging the tools available in VMware vCenter Lifecycle Manager, these groups of servers can be easily promoted and demoted throughout the application lifecycle, which would significantly reduce implementation times and simplify testing by ensuring that the test environment moves into production without any modifications. Using snapshots, these pods could also be held at a point in time to facilitate simple debugging operations.

Also, using the authentication and authorization features of VI3, specific users or groups of users can be granted access to only certain virtual servers, such as only the test pod, while other users can access all off the pods.

The number of apps and the performance of those applications is obviously highly dependent on the resources that those applications require. However, the trend in virtualization is to increase the number of virtual CPUs and RAM available to individual virtual machines. We can reasonably expect that virtualization vendors will increase the number of CPU cores that each virtual machine could use to perhaps 8 or 16 virtual CPUs, if not more. By using a large-platform server such as the DL785 G5, adding that number of CPUs does not restrict that VM to a single server at the expense of other virtual machines, such as would occur in a dual-socket blade farm model.

For application deployments, this can be a critical feature, as it would be possible to greatly enhance the resources available to a specific virtual machine, such as a database server, without worrying about the consequences inherent in a large farm of small-platform servers or blades. If used with the “pod” model noted above, stringent testing of virtual machines and code running at more than four CPUs could be conducted on the same physical hardware that would later be responsible for running those applications in production.

Disaster Recovery Hotsite

In a virtualized environment, disaster recovery and hotsite management become much simpler. Rather than requiring an equal number of physical servers to replicate a production environment, that virtualized environment can be replicated at a remote site and be ready for work should problems occur in the production infrastructure.

However, traditional 1U or blade-based virtualization farms still require that enough processing power be available at the hotsite to run the required virtual machines during an event. By using the HP ProLiant DL785 G5 at the hotsite, the costs of maintaining that site drop, as there's much less infrastructure required to maintain the availability.

For instance, to replicate a blade infrastructure at a hotsite might require another blade chassis with some number of blades and the power required to run them. Also it will require significant switch configuration and port utilization, essentially identical to the requirements of the production site itself.

On the other hand, by placing a ProLiant DL785 G5 at the remote site, there's a single server that can be remotely powered up and be ready to handle a full workload within minutes, and the only requirements might be two network connections, depending on the design.

In addition to these benefits, using a ProLiant DL785 G5 in this manner essentially removes the need for farm stabilization and load balancing with VMware VI3 features such as DRS (Dynamic Resource Scheduling). The end result is that virtual servers brought up at the hotsite will not shuffle between physical hosts with VMotion following powerup.

Another modification to this design would be to use a ProLiant DL785 G5 as a VMware HA host at a company's headquarters, configured to host virtual servers that would normally be running at remote sites. In the event of a remote site hardware failure, the resources required to continue operations of that site would be brought up on the DL785 G5 until such time as the on-site hardware could be repaired.

Virtual Desktops

VDI, or Virtual Desktop Infrastructure is one of the newer cost-cutting measures directly related to virtualization. In its simplest form, VDI enables a large number of desktop virtual machines to be run on a physical host, with users connecting via RDP (Remote Desktop Protocol) from thin client systems. For the end user, the end result is much the same as having a physical desktop system under their desk, but the power footprint is greatly reduced, as are resources required to maintain dozens or hundreds of physical desktop systems.

One of the greatest benefits of VDI is found in VMware's resource sharing technology that enables oversubscription of physical server resources in environments where many virtual machines are running the same code. For instance, static read-only libraries commonly used by desktop operating systems and applications are not simply copied into RAM per instance, but can be called from existing instances. This reduces the physical memory footprint of the virtual machine.

The larger the resource pool of any one physical host, the greater the benefit, as more virtual servers can leverage this resource sharing without impinging on the performance or stability of the virtual system.

Although VDI was not part of the tests conducted for this paper, it may be quite possible to run hundreds of VDI virtual machines on a single ProLiant DL785 G5, depending on the resources required per system.

As with any virtualization implementation, the specific needs of the virtual machines will dictate the number of virtual machines that may run on any hypervisor simultaneously.

Infrastructure in a box

Many infrastructures may determine that a single ProLiant DL785 G5 running VMware VI3 may provide the entirety of their server and desktop processing needs. With 32 cores and up to 512GB RAM, there is enough horsepower available to run a full complement of standard network services in addition to a reasonable number of VDI instances.

Obviously this is highly dependent on the business needs, and will vary from company to company. However, some small to mid-size businesses may not need more than a Windows Active Directory domain controller, perhaps a few database and application servers, and a Microsoft Exchange server.

Good practice dictates that the VMware Virtual Center server be run on a separate physical server from the virtualization host, and that a separate physical Active Directory domain controller be maintained, but in many situations, those services can reside on the same physical system.

Thus, a single ProLiant DL785 G5 and perhaps a mid-range 1U server can effectively run an entire business. Coupled with another ProLiant DL785 G5 for redundancy and load-balancing, it may even be feasible to run larger infrastructures from this relatively small footprint.

Looking back to 2003, when Windows Server 2003 was released, all server CPUs were single-core. If the infrastructure required less than 32 servers, or less than 16 dual-CPU servers, then it may be that a single DL785 G5 could supplant all of those physical servers, which would greatly reduce cooling, power, and maintenance costs.

In fact, most medium-size infrastructures that are not virtualized are likely to only use 15 to 20% of their available computing resources on a daily basis, leaving many servers largely idle. Implementing a virtualized environment may result in much greater use of available resources while still reducing costs.

Bio
Paul Venezia is a veteran network and systems architecture consultant as well as a Senior Contributing Editor for InfoWorld. He has been published in dozens of magazines and newspapers across the globe, including PC World, CIO, LinuxWorld, and the New York Times.

He runs an independent test lab where these tests were conducted, and advises corporations in several vertical markets on IT infrastructure development.