



Scaling MySQL With Virident Flash Drives and Multiple Instances of Percona Server

A Percona White Paper

By Vadim Tkachenko (Percona), Shridar Subramanian (Virident), and Baron Schwartz (Percona)

Abstract

Scaling a MySQL database server has traditionally meant deploying in a so-called *scale-out* configuration, with many MySQL instances running on inexpensive machines. This strategy was also called *horizontal scaling*. In recent years, however, the rapid increases in hardware capacity and decreases in price made scale-out less economically desirable, and there was considerable pressure to make the software able to run on larger, more powerful servers. This *scale-up* or *vertical scaling* architecture remained out of reach until dramatic improvements in database performance, led by Percona Server with XtraDB, made vertical scaling a practical and economical strategy. However, hardware continues to outpace software, and even Percona Server cannot today utilize the full power of commodity hardware. This is especially true when Virident's fast PCI-E *tachION* storage device is used, because database technology has been optimized for spindle-based disks for decades. With the large amounts of memory, many CPU cores, and enormously powerful *tachION* drives readily available today, a hybrid strategy of multiple database servers per physical server is necessary to better utilize the hardware's capacity. Such a configuration can enable a 5x to 15x consolidation factor, and improve performance and hardware utilization many-fold.

1 Scaling Up Versus Scaling Out

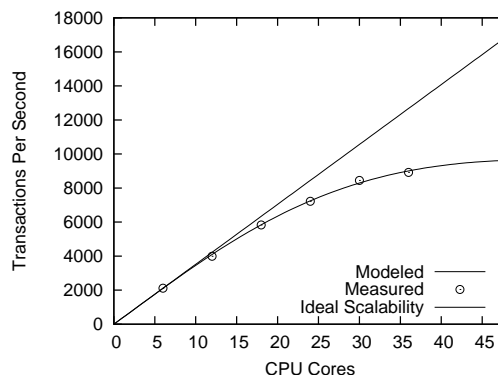
Our previous white paper, *Scaling MySQL Deployments With Percona Server and Virident tachION Drives*, demonstrated clearly that scaling up is preferable to scaling out on today's high-performance hardware. In our opinion, commodity hardware today brings the following configuration well within the average MySQL user's reach: 48 CPU cores, up to half a terabyte of memory, and up to a million IOPS (I/O operations per second) with Virident *tachION* drives.

This is more than an entire cluster of servers boasted a few years ago. Obtaining a comparable total amount of computing power in 2006 would require approximately 12 servers with 48GB of memory each. However, even that cluster of machines would not provide as many IOPS as today's single server can obtain; typical RAID arrays of 15k RPM disks would provide only a few thousand IOPS at best, making the entire cluster's total IOPS less than a tenth of what the Virident devices can sustain.

Put another way, although Moore's Law has been driving CPU power and memory capacities up continually, the recently increase in I/O power outstrips all other improvements in commodity servers by a wide margin.

2 How MySQL Utilizes CPU Capacity

As recently as version 5.0 and early 5.1, MySQL was famously incapable of scaling to more than about 4 CPU cores before performance would actually regress to lower than the level achievable on 4 cores. Today, however, the situation is very different. Recent versions of MySQL with the InnoDB Plugin, and especially Percona Server with XtraDB, scale up to many more CPU cores. For example, see the following model of CPU scalability, based on a [sysbench benchmark](#) from Mat Keep (Oracle):



The model, which is based on Dr. Neil J. Gunther's Universal Scalability Law, predicts that InnoDB should scale to 54 cores. However, note that even though MySQL continues to provide more

throughput at all measured core counts, the scaling is not perfect, and the returns diminish more and more quickly as the core count rises. The performance at 48 cores will not be double that of 24 cores.

The salient point of this section is that commodity servers have much more CPU power than MySQL and InnoDB can use effectively, and at lower core counts, MySQL delivers more performance per core than at higher counts.

3 How MySQL Utilizes Memory

In addition to imperfect scaling as the CPU count climbs, MySQL tends to deliver diminishing returns as the InnoDB buffer pool size increases. The interaction between the buffer pool size, the workload, and the hardware is difficult to isolate precisely in benchmarks, so this is an area where more research is needed. However, we know that the combination of more memory and more CPUs further inhibits MySQL's scalability.

Based on our knowledge of InnoDB internals, and observations such as wait analysis during benchmarks, we know that the primary bottleneck is the InnoDB kernel mutex, and the secondary bottleneck is probably the transaction log mutex. Future versions of InnoDB will be greatly different—in fact, the kernel mutex is being removed entirely—and that will certainly change the scalability bottlenecks.

4 How MySQL Utilizes I/O Capacity

Making MySQL use large amounts of I/O capacity is no simple task. InnoDB, the default transactional storage engine in MySQL, is heavily optimized for traditional disks. In fact, it used to be hard-coded to assume a fixed capacity of 100 IOPS from its storage. Removing this limit was the first step towards making it scale vertically on modern systems. Beyond that simple configuration parameter, however, lie a variety of challenges. Percona has solved many of them, but some internal algorithms are still not ready for the number of IOPS that a Virident *tachION* card can provide. In addition, internal locking strategies cause mutex contention that reduces InnoDB's scalability on extremely fast storage.

Many of these roadblocks have been significantly eased in the last several years, especially in Percona Server with XtraDB, which is arguably faster version-for-version than any other MySQL-based database server. But some of the problems remain difficult to solve, enmeshed as they are in InnoDB's intricate inner workings. We believe that this will be an ongoing project for quite some time.

5 Scaling Through Multiple Instances

Our previous white paper enumerated the Virident *tachION* device's features: extremely high performance, low latency, high capacity, small form factor, modularity, and fault tolerance. Given that no immediate solution is available to the software's limitations, and that there is adequate CPU and memory capacity to handle much more work than MySQL can perform, how can we more fully use the raw power available from the Virident *tachION* drive?

We have seen that each instance of MySQL delivers more performance per core at lower core counts and smaller memory sizes. This means that it is still more economical per CPU and per GB of memory to run smaller MySQL instances than larger ones, to a point. However, it is actually more economical to purchase CPUs and memory in larger quantities than MySQL excels at using. This is especially true when one considers the relatively fixed overhead cost of the other components in the server, and the operating expenses of running each server. In other words, the optimum server configuration in terms of cost is different from the optimum in terms of efficiency for MySQL.

Therefore, even though it is more economical today to scale vertically than it used to be, from a purely server-centric viewpoint, it is still less expensive to partition the work onto multiple servers. This is known as *sharding*.

In theory, then, the best of both worlds is to buy powerful, high-capacity servers and run several small database server instances on them. To test our theory, we created a sharded benchmark.

6 The Sharded Benchmark

Our benchmark is called *tpcc-mysql*, and is designed to be similar to the industry-standard TPC-C benchmark, which emulates an OLTP system processing transactions. We executed the benchmark with the following three total buffer pool sizes: 26GB, 52 GB, and 120GB. We created sample data sets that fit into these memory sizes, and divided each set into four shards.

For each buffer pool size, we executed three benchmarks. The first was with all four of the shards in a single MySQL instance with one large buffer pool. For the second test, we created two MySQL instances, each with a buffer pool one-half the size of the data, and with two shards per instance. And finally, we created four instances at one-quarter the size, and loaded one shard into each instance.

We used the *numactl* command to bind the MySQL server process to specific NUMA modules, isolating it onto physical processors to emulate running on a machine with fewer CPUs. The single-instance benchmark ran on all 48 cores, the benchmark with two instances ran each instance on 24 cores, and the four-instance benchmark allocated 12 cores to each instance.

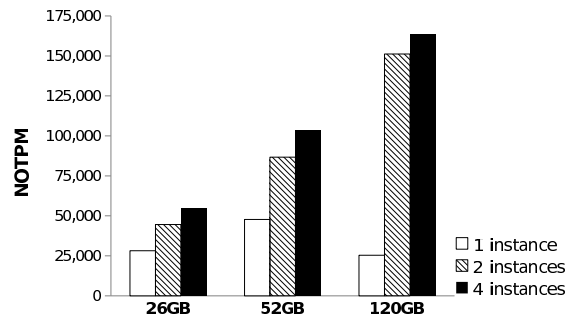
The benchmark machine was a Dell R815 with 48 cores total, and 160GB of memory. The data was placed on a single *tachiOn* card with 200GB of capacity¹. In all cases, we exercised the servers with 12 user connections per shard from another server running four instances of the *tpcc-mysql* benchmark driver program, for 48 connections total. The MySQL system under test was Percona Server version 5.5.9, a release candidate of the 5.5 series of Percona Server.

7 Benchmark Results

The following table summarizes the results of the benchmark. The numbers in the cells are New Order Transactions Per Minute (NOTPM).

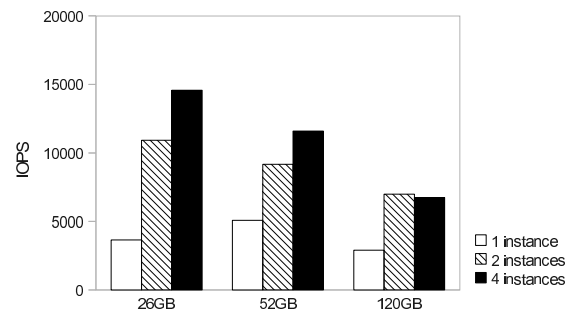
Total Size	1 Instance	2 Instances	4 Instances
26GB	28157	44530	54573
52GB	47747	86654	103169
120GB	25297	151204	163773

As you can see, splitting the single instance into two instances produces a very good improvement in throughput: 1.6x, 1.8x, and 5.9x for the 26GB, 52GB, and 120GB cases, respectively. This is despite the decreased memory available to each instance. Dividing into four instances further improves the performance, but not as dramatically as splitting into two instances. This is to be expected, because we know that MySQL scales quite well up to 24 CPU cores. The following chart shows the table visually:



The contention problem is especially bad for a single instance at 120GB of buffer pool, which matches our expectation that the combination of large amounts of memory and many CPU cores causes the most scalability problems.

Sharding the database and using the server to host multiple instances lowered the internal contention inside MySQL enough that MySQL could drive significantly more I/O operations per second to the Virident card, making much better use of the device. The following chart shows the number of I/O operations per second for each of the benchmarked configurations:



¹The 200GB capacity card is no longer produced; capacities range from 300GB to 800GB at the time of writing.

The chart might look counter-intuitive, but remember that all of the tests actually have the same total amount of data. It is divided up differently amongst the servers in the sharded benchmark, but we held the amount as a constant while varying the amount of memory and number of server instances (and CPU cores assigned to each instance). So it is expected that there is less I/O in the benchmark tests with larger amounts of memory—that is what normally happens when the buffer pool size is increased. The important thing to note is the improvement in the amount of I/O the database is able to push to the Virident *tachION* card, as we move from one instance to two instances, and in some of the cases even up to four instances.

8 Benefits and Drawbacks

The most obvious benefit of the approach we used in this benchmark is the improved performance of the database server. Improving performance approximately six-fold by simply breaking the data in half and running two server instances is quite an important result. Our experience and benchmark results lead us to believe that this will be even more pronounced at higher memory sizes, and a 120GB buffer pool is not very large in today's commodity hardware. We expect, as a rule of thumb, that this technique can enable users to consolidate between 5 and 15 machines onto a single physical server, depending on the number of CPU cores and the amount of memory available. A single Virident *tachION* drive should be, in our experience, sufficient to handle the IO needs of such a configuration. Note that Virident makes larger capacity drives, which might be needed at higher consolidation factors, simply because of the volume of data.

The secondary benefit is much better resource utilization, which leads to improved total cost of ownership (TCO). A single MySQL instance cannot exercise the full I/O capacity of the Virident *tachION* drive at present.

Our approach adds some complexity and risk over

the standard sharding practice. Sharding is normally performed at the level of physical servers. Running multiple instances of MySQL on a single host is less popular, for several reasons. For example, it requires additional MySQL configuration, and there are no good-quality and widely-tested startup scripts for multi-instance deployment.

These objections can be addressed, but without a SSD as powerful as the Virident *tachION*, one problem always remained: there was no practical way to scale I/O. Today, I/O capacity need no longer be the limiting factor. As a result, for those who are or soon will be sharding, the benefits of scaling through multiple instances can outweigh the costs.

9 Conclusion

In the future, we expect MySQL and InnoDB to continue to improve and become more scalable, which should make it more economical to run monolithic server instances. But we also expect servers with many more CPU cores and much more memory to become popular, and we believe that flash-based storage devices such as the *tachION* drive will also become much faster. It remains to be seen whether MySQL and InnoDB's scalability will improve fast enough to keep up with the hardware improvements we expect. We think this is unlikely, and so we expect that there will continue to be some compelling benefit from sharding well into the future.

Although we dislike sharding, the configuration we have discussed in this paper still has at least this virtue: rather than sharding over many machines, it is possible to shard on a single physical server. This can help reduce cost and improve utilization significantly, which is enough to make it worthwhile in many cases. Our research suggests that collocating two MySQL server instances on fast storage represents a good balance between cost, complexity, and performance on today's commodity server hardware.

About Percona

Percona is the oldest and largest independent provider of commercial support, consulting, training, and engineering services for MySQL databases and the LAMP stack. You can contact us through our website at <http://www.percona.com/>, or to call us. In the USA, you can reach us during business hours in Pacific (California) Time, toll-free at 1-888-316-9775. Outside the USA, please dial +1-208-473-2904. You can reach us during business hours in the UK at +44-208-133-0309.



About Virident Systems

Virident Systems builds enterprise-class solutions based on Flash and other storage-class memories (SCM). These disruptive technologies will revolutionize the data center and cloud computing by solving performance, reliability, and serviceability problems that further compound in large-scale deployment of SSDs in current environments. Visit <http://www.virident.com> for more information, or call us at (408) 503-0100 during business hours in Pacific (California) Time.



About Percona Server

Percona Server is an enhanced, high-performance version of the world's most popular open-source database, MySQL. MySQL is used by many of the world's largest websites, including Facebook, Flickr, and YouTube. MySQL is also deployed widely in industries such as financial services, government, education, pharmaceuticals, and telecommunications. Its simplicity, reliability, and ease of use make it cost-effective to manage, and because it is open-source, it can be used without license fees. Percona Server is derived from the MySQL database, to which it adds features such as enhanced monitoring and configurability. Percona Server offers much faster and more consistent performance than the standard MySQL server. Percona also provides a free hot-backup program, **Percona XtraBackup**.

Percona, XtraDB, and XtraBackup are trademarks of Percona Inc. InnoDB and MySQL are trademarks of Oracle Corp. Virident and tachION are trademarks of Virident Systems Inc.