Scaling MySQL Deployments With Percona Server and Viridient tachIOn Drives

A Percona White Paper

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

Abstract

Until recently, MySQL and InnoDB were unable to take advantage of high-performance hardware, especially high-performance storage systems. Large-scale deployments relied on so-called *horizontal scaling* with a technique called *sharding*—partitioning data across many small-to-midsize servers. Sharded architectures arose from the different nature of read and write workloads. If the database's write workload was not too heavy, then a MySQL deployment could be scaled horizontally with replication, using read replicas. Reads could be scaled because replication creates multiple copies of the data, which can serve queries independently of each other. However, writes must be repeated on every replica's copy of the data, so replication does not help scale write traffic. The traditional way to do that with MySQL was through sharding.

Today, MySQL—especially Percona's enhanced version of MySQL, **Percona Server with XtraDB**—is capable of exercising much more powerful servers and storage systems. As a result, server consolidation and vertical scaling is a viable—and often much more economical—path to high performance with MySQL. This white paper explains how to configure servers for high performance with MySQL and the **Viridient *tach*IO*n* PCIe solid-state storage device**. Key benefits of this approach are lower power consumption, decreased rack space for lower CapEx, and decreased architectural complexity to reduce operational and administrative costs.

1 Traditional MySQL Scaling Strategy

The prevailing scaling strategy with MySQL is to "scale out" by making many copies of the data with MySQL replication and distributing them to multiple servers. This scales the **read** workload. However, as the demand on the database grows, this approach can result in a large number of replicas, leading to a management and operational nightmare—not only requiring maintenance of many machines but also managing several copies of data. In addition, replication requires application modification to accommodate its asynchronous nature. More importantly, users often find that this approach does not scale very well as the associated **write** workload grows. The number of replicas that can be added to address the read workload is limited by the number of writes that have to be replicated and the single-threaded nature of MySQL replication. The net result is that scaling of the read workload is not independent of the write workload.

Scaling for writes is much more complex. Some approaches include increasing DRAM sizes, using a SAN or external storage, and sharding. Increasing the server's DRAM capacity looks like an easy fix, but works only up to a certain point. As the write workload and dataset size grows, this solution becomes prohibitively expensive due to the non-linear price-to-density relationship for DRAM, and there is a physical limitation on the amount of memory that can be added per server. Adding memory also does not solve the ultimate need for durable, persistent storage. That requires MySQL to periodically flush the data from memory to the backing store. As a result, the steady-state performance of such a solution is limited by the performance of backing store drives, and hence performance can be poor. Another limitation of this approach is the significantly high "warm-up time," which is the length of time between server startup and the time it can accept high loads.

Although SANs can deliver good I/O throughput, the latency remains a problem. Latency is critical for MySQL performance, especially for replication, long queries, and batch jobs. These operations are run serially; hence their response time is directly impacted by the large latencies of a SAN device. SAN-based solutions are also very expensive.

Sharding is the last option. But this is never an easy

solution, and should be considered only as a last resort. It requires complex data management along with changes to applications. The architectural design must be thought out in advance in detail, and it often leads to limitations on applications. For example, it becomes difficult or impractical to execute some types of queries because data is placed in different locations. Joining data across different servers requires emulated joins in the application code. The data becomes married to one form of architecture and a subsequent change to the architecture, to meet new business needs, could be very difficult.

Sharding is also complex—the application developer has to write more code to be able to handle sharding logic, and operational issues become more difficult (backing up, adding indexes, changing schema). Thus, scaling for read and write workloads using traditional methods leads to a more complex infrastructure and higher costs.

2 Scaling Up with Flash Storage

Flash-based storage, such as solid-state drives (SSDs), has created a paradigm shift in the way in which data is stored, managed, and accessed. Both read and write performance issues can be significantly alleviated, and many applications can see instant improvement in I/O performance. Of the various types of SSDs, the highest performance (highest bandwidth and lowest latencies) is delivered by PCIe-based SSDs. SATA-based SSDs are inherently limited by bandwidth and have higher latency. This is not due to the underlying storage technology, but rather to the physical interconnect technology and topology.

Virident's *tach*IO*n* drive is a flash storage device that works well for scaling MySQL servers without the need for sharding or replication. Virident *tach*IO*n* drives can deliver over 330,000 4kB read IOPS (I/O operations per second) and 200,000 IOPS in a mixed, random, read-write OLTP-like workload at 100% capacity utilization. This is orders of magnitude higher than spindle-based disk drives. The *tach*IO*n* drive's I/O performance is alse exremely consistent and predictable, helping produce high quality of service to the database's end user. The following are some of the characteristics of the *tach*IO*n* drive:

Performance

: Benchmarking and customer deployments have shown that, depending on the server's workload and working set of data, the *tach*IO*n* drive can provide up to 15x performance gain compared to HDDs. The Virident device lets companies scale their MySQL infrastructure more effectively and less expensively than buying more DRAM, purchasing a SAN, or having to shard the database, with no further changes to the system (e.g. no need for a proprietary variant of MySQL or a closed-source storage engine).

Latency

: The *tach*IO*n* drive connects directly with the CPU using PCIe, giving latencies in the tens of of μs (microseconds). This is better, by orders of magnitude, than a SAN or even SATA- or SAS-based SSDs.

Capacity

: The *tach*IO*n* drive is available in usable capacities ranging from 300GB to 800 GB.

Form Factor

: The Virident *tach*IO*n* low profile PCIe SSDs can be installed into any server chassis without the need for additional power.

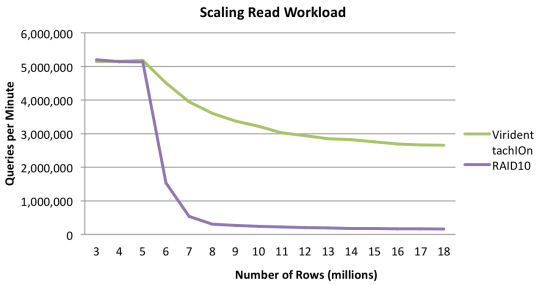
Modularity and Fault Tolerance

: The *tach*IO*n* drive design is modular, consisting of a base card with field replaceable flash modules. It has an onboard flash-aware RAID5, which ensures high data availability at all times, even in the event of a flash module failure. This is in addition to the ECC implemented at the flash level.

3 Scaling Read Workloads

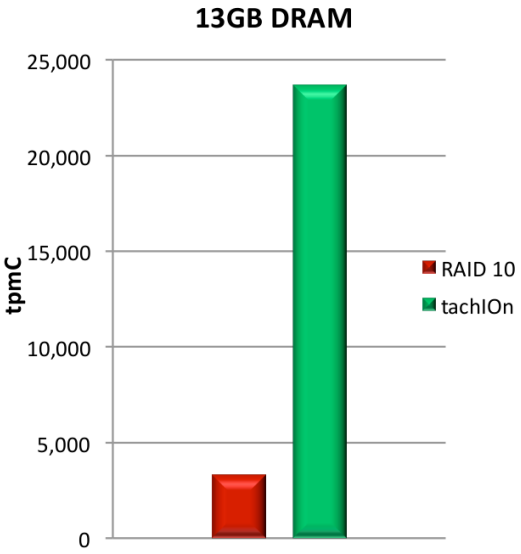
We benchmarked a read-only workload on both *tach*IO*n* and traditional RAID hard drives. The results show that the traditional phenomenon of performance dropping dramatically as data size exceeds RAM size is significantly less of a problem with the *tach*IO*n* drive. With fast storage such as the *tach*IO*n* , it is possible to get very good performance

on datasets that are much larger than memory. This permits consolidating many servers into one.



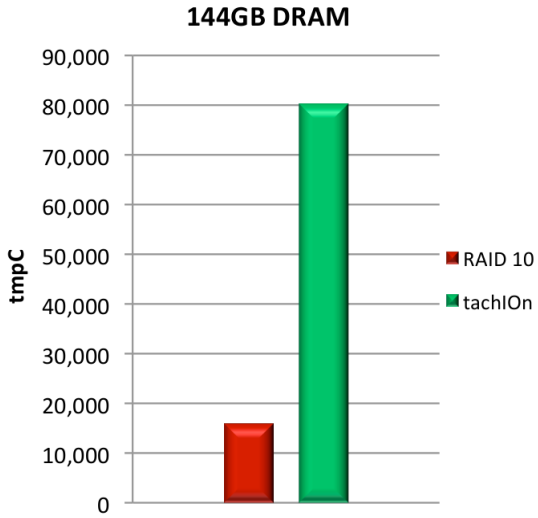
4 Scaling Write Workloads

The *tach*IO*n* drive is not only good at handling reads—it is very efficient in handling write traffic as well. The following benchmark is a read-write benchmark called tpcc-mysql, which is designed to mimic the industry-standard TPCC-C benchmark.



As the benchmark shows, a single *tach*IO*n* could increase a server's capacity by a factor of almost 10. This is a significant improvement. This enables the *tach*IO*n* drives to improve MySQL replication performance, alleviating the single-threaded replication bottleneck and making it possible to scale reads and writes more independently. More importantly, using *tach*IO*n* drives on a master can eliminate the need to shard the database. Many sharded environments we have worked with could be consolidated to a single server with this performance increase.

The preceding benchmark illustrates the situation when the RAM is limited, and the dataset is much larger than memory. However, even when the entire data set can fit into DRAM, using a *tach*IO*n* drive can deliver a 5X improvement in the ability of the MySQL master server to handle write workloads:



5 Scaling When Data Exceeds Memory

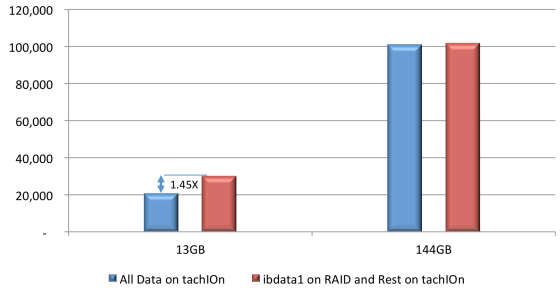
If the entire data set fits on a single card (up to 800GB usable capacity), the previously discussed performance improvements can be achieved by moving all the data into the *tach*IO*n* card. No additional changes are required. If the dataset size is too large, there are three ways to arrange the data.

1. Stripe: Stripe the data across multiple *tach*IO*n* cards on a single server
2. Tier: Locate the most IO-intensive files on the *tach*IO*n* drive
3. Cache: Implement a flash-memory-friendly caching solution such as FlashCache

You may also consider separating the files in the following manner: place the transaction logs and binary logs on a RAID 10 SAS HDD and the entire remaining index and data files on the *tach*IO*n* drive. This conserves valuable space on the PCIe SSD. A

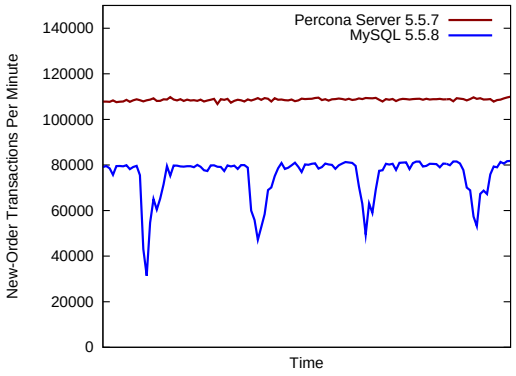
RAID 10 HDD array with a battery-backup unit is a good choice. The RAID array can also be used for other logs such as the slow query log and error log.

Performance can be further improved by putting system tablespace (ibdata1) on separate HDD storage, because I/O patterns for this tablespace are different from I/O patterns of data and index files. The additional improvement can be as much as 1.45x. This improvement is over and above all the performance improvement obtained by using *tach*IO*n* drives for write intensive workloads where the size of the hot data is significantly larger than the DRAM size.



6 Using Percona Server with XtraDB

An important additional measure for improving the overall performance is to use Percona Server with XtraDB, which has much higher performance than the standard MySQL server, due to its more scalable design on modern hardware. Percona Server with XtraDB provides significant throughput increases, reduced response times, and much more consistent performance. It is particularly important on extremely fast I/O devices such as the *tach*IO*n* drive, because the standard MySQL server simply cannot take advantage of all the I/O capacity available to it. The standard MySQL server will reach its throughput ceiling while there is still idle CPU and I/O capacity. In addition, the standard InnoDB's checkpoint algorithm must be tuned precisely for a particular workload an I/O device in order to avoid periodic server stalls—and this is possible only in an artificial setting such as a laboratory benchmark. A real workload has variations, and no single combination of server settings will produce good performance all the time under a varying workload with standard MySQL and InnoDB.



7 Conclusion

The sharding approach that has been advocated for the last five years or so is becoming increasingly questionable advice in some environments. Traditionally, IT departments have adopted a multi-pronged strategy to address the scaling problem. They typically separate out the read and write traffic. Masters absorb the write traffic, while the majority of the read traffic is directed to the replicas. Today's solid-state PCIe hardware offers extremely high-bandwidth, low-latency I/O performance, exemplified by the Virident *tach*IO*n* drive. And today's MySQL server, especially MySQL 5.5 and even more so Percona Server with XtraDB, is capable of utilizing much more of that hardware's available capacity effectively. "Scaling up" is once again a viable and economical strategy for MySQL, and "scaling out" need no longer be the default database architecture.

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**.