



## Configuring Percona Server with XtraDB for Fast Restarts

*A Percona White Paper*

Baron Schwartz and Vadim Tkachenko

### Abstract

InnoDB requires a long time to restart on servers with large amounts of memory. It can take many hours to warm InnoDB up enough to serve queries quickly. The database's "working set" of data is not in memory after a restart, so queries must perform random disk I/O to read pages, and are thus too slow for production use. Percona Server with XtraDB greatly lowers database warmup time by quickly restoring the buffer pool to its state before shutdown. This enables higher uptime, and removes the need for elaborate methods of warming up servers.

Percona Server with XtraDB offers a unique feature that can reduce downtime during restarts to a matter of minutes, even on servers with very large amounts of memory. This removes many severe operational hurdles and greatly reduces the cost of a server restart, permitting MySQL to scale to much larger hardware and to be used more flexibly. Percona Server's fast restart capabilities provide significant operational and cost advantages in scenarios such as cloud computing, high availability, dynamic scaling, and restores from backups.



---

With over 14 million monthly active users as of July 2010, [mixi](#) is one of the largest social networking service platforms in Japan. To provide billions of pages every month, we deploy hundreds of MySQL servers running the XtraDB storage engine. It's a challenge to keep up with the never-ending demand for expansion of servers, and as the number of servers increases, the operational costs increase exponentially. Without XtraDB, a lot of care was needed to warm up the buffer pool, but XtraDB's easy dump and restore feature makes it possible to add new replicas with much less effort. Also, setting up new replica servers is much easier with the XtraBackup tool. Thanks for making our life easier!

— mixi, Inc. Operations Engineering Team

---

## How Long Is InnoDB's Warmup Time?

Exactly how much time does InnoDB<sup>1</sup> need to warm up on high-end commodity hardware? To find out, we stress-tested the most powerful server in Percona's laboratories, the Cisco UCS C250 server. This server has dual Intel Xeon X5670 (Nehalem) CPUs at 2.93GHz, with 6 cores and 12 hardware threads on each, which is visible to the operating system as 24 CPUs. It has 346GB of memory installed, and can be configured with up to 384GB. The storage was an 8-disk RAID10 array of 15K RPM hard disks. This Cisco hardware configuration is at the leading edge of today's commodity-class servers, but will be more commonplace in the near future.

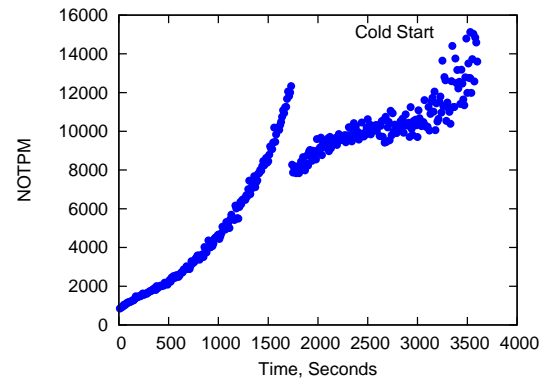
We ran a benchmark that we designed to be very similar to the industry-standard TPC-C benchmark. This benchmark is also called the "order-entry" benchmark. It is a complex OLTP (online transaction processing) workload, and the performance metric is new-order transactions per minute (NOTPM). It is similar to the workload many web applications will demand of a high-end server such as our Cisco unit. The source code is available from [Launchpad](#).

We ran the benchmark with 1000 warehouses and 10 concurrent connections for one hour. The total data size is 100GB, and we sized the buffer pool at 140GB, so the data fits entirely in memory. We benchmarked Percona Server version 5.1.50-rel12.1, which is MySQL 5.1.51 with version 12.1 of the Percona enhancements.

The following figure is a chart of the results (full results are available on the [Percona benchmark wiki](#)). After about 1700 seconds, the throughput peaks and dips<sup>2</sup>, then begins to stabilize after about 2000 seconds as InnoDB's internal structures and activities reach a steady state, which we call "warmed up." Note that the initial throughput is less than ten percent of the throughput when the server is warmed up. This server cannot be put into production use until it is warmed up. Although we did not measure the response time, it clearly would not satisfy users initially.

<sup>1</sup>When we mention InnoDB, we are sometimes writing about XtraDB, Percona's enhanced version of InnoDB. To keep the language clear, we will simply refer to InnoDB in most instances where either could be used interchangeably.

<sup>2</sup>An area of ongoing research; we will speculate later on the reason for the dip.



The above result is for a synthetic benchmark. In production usage, we commonly see servers with a warmup period of more than two hours, and sometimes more than a day. During that time, the server is offline and not able to serve any queries—not even a reduced workload. Even under partial load, the server's response times would be unacceptably long.

The more memory you have, the longer the warmup period. That is why servers such as our Cisco machine show the effects of warmup so clearly.

## The Impact of Slow Warmup

InnoDB's slow warmup time on large-memory servers imposes a surprising number of operational limitations onto the database server. Slow warmup is equivalent to slow restart, which creates problems such as the following:

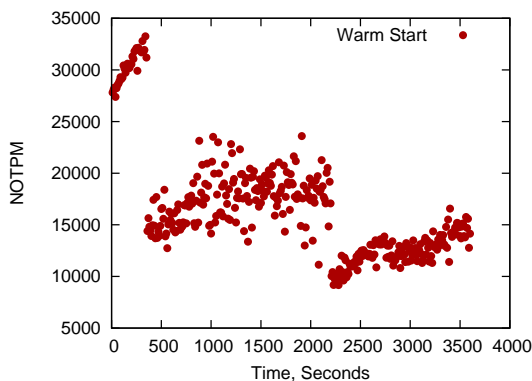
- It can be impossible to meet uptime SLAs without redundant hardware that is otherwise not necessary
- It is more difficult to tune the server properly
- It is more expensive to upgrade the server
- It is difficult to perform intrusive debugging or routine profiling

Anything that requires a server restart becomes difficult to justify in a high-uptime environment, and many of the server’s most valuable features require a restart to enable or configure. For example, in MySQL 5.5 the new Performance Schema helps users profile the server internals, but it requires a restart to enable or disable. New versions of the server deliver bug fixes, and add performance and feature enhancements, but if an upgrade requires many hours of effective downtime, users might elect to skip upgrading and run with a buggy or insecure server version instead.

This is not a complete list of the impacts of slow warmup. We could mention many other scenarios where we wished for faster restarts. For example, after setting up replication, the cold server might replicate so slowly that it never catches up to its master, and the master might purge the binary logs the replica needs, causing replication to fail.

### Fast Warmup with XtraDB

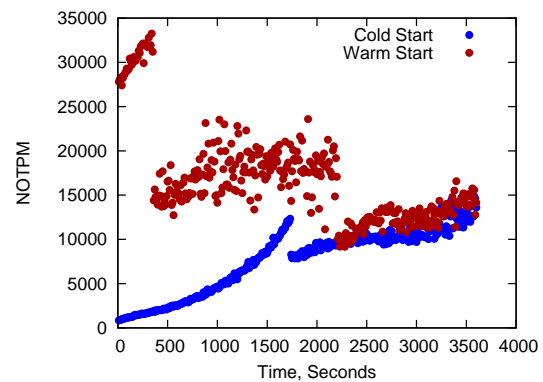
XtraDB’s warmup requires only minutes, even with a 140GB buffer pool. We repeated the benchmark shown previously, but before beginning the benchmark, we restarted the database server and waited five minutes for the LRU load to execute.



As you can see, the benchmark starts off with extremely high throughput—in fact, rather than slowly increasing as time passes, it actually decreases shortly after the start. We think that this is because other processes inside of InnoDB, such as the dirty-buffer flush process, have no work to do

immediately after restart. After a time, they begin to work, and this causes throughput to drop. The variations in performance, and the three distinct stages of performance shown on the chart, are areas of ongoing research. There is much to learn about the intricate inner workings of InnoDB on complex benchmarks such as this.

The before-and-after performance difference is not clear when the charts are viewed separately, but when the charts are rendered together it becomes obvious:



### The InnoDB Buffer Pool

What are we really warming up when we run the server for a while? It is the buffer pool, which is usually the largest memory buffer in a MySQL server using InnoDB (or XtraDB). On today’s hardware, it is not uncommon for servers to allocate more than 100GB of memory to the buffer pool, and we know of users who configure the buffer pool many times larger than that. InnoDB uses the buffer pool to hold pages of data, which are typically 16 kilobytes in size. The buffer pool is managed through a complex series of data structures and algorithms to perform tasks such as flushing changed pages to permanent storage, fetching pages from disk into the buffer pool so that InnoDB can do its work, and removing pages to make room for new pages when the pool is full.

If the dataset is larger than memory, InnoDB uses a least-recently-used (LRU) page replacement algorithm to make room for pages that are required to execute queries. We can simplify by saying that InnoDB removes the oldest page from the buffer pool,

and replaces it with the new page from disk; there are subtleties, but that is enough detail for this paper. InnoDB uses a data structure called an LRU list to record how recently each page has been used. This list contains one entry for every page, which means that it is a complete list of the buffer pool's contents.

### How XtraDB's Fast Warmup Works

XtraDB can warm up the buffer pool almost immediately by reading into the LRU list the pages that resided there just before the server was shut down. It reads from a file that contains a list of page identifiers, which are 8 bytes each. It sorts the page identifiers by tablespace and then by physical location within the tablespace, so they can be read sequentially for fastest loading. Then it reads the pages from disk and places them into the LRU list in the proper place.

The file with the page identifiers is generated by a background thread in XtraDB, which writes out the file at intervals as specified by the `innodb_auto_lru_dump` configuration variable. This variable specifies the period between writing out the LRU list's contents, in seconds.

All that is required to enable saving and reloading of the LRU list is to set the `innodb_auto_lru_dump` variable to some positive value. The process is fully automatic and does not impact performance or cause any locks or stalls. We ran our benchmarks with the variable set to 300, so there was an LRU save operation every five minutes. Upon server restart, the LRU is automatically loaded from the saved file.

Full documentation on the functionality to save and reload the LRU is available from the [Percona documentation wiki](#).

### Other Ways to Warm Up the Buffer Pool

Many solutions to the cold buffer pool have been proposed, but those that appear to be easy fixes simply do not work. These include the following:

#### Full table and index scans

A common response to complaints about slow warmup is advice to use the server's `--init-file` option to execute queries that will scan tables and indexes, loading them from disk into memory. But this loads portions of the tables that are never accessed by the real workload, and if the data is larger than the buffer pool, it discards data that will be needed again soon, thereby forcing it to be read from disk all over again. In addition, this strategy doesn't guarantee sequential I/O access, and with newer versions of InnoDB that have a two-part LRU list to avoid abuse due to one-pass scans, it doesn't even fill the buffer pool because pages are never promoted to the longer-lived portion of the LRU.

#### Avoiding the O\_DIRECT option

High-memory InnoDB installations are usually configured to open the data files with the `O_DIRECT` option, so that the operating system does not cache the data in memory, and the only in-memory copy of the data is in the buffer pool. Without this option, the operating system will cache the data in its page cache, avoiding reading them from disk. But this causes double-buffering of the data, which is not only wasteful, but causes virtual memory pressure and swapping even at moderate buffer pool sizes. `O_DIRECT` avoids double-buffering and swap pressure, at the cost of forcing the data to be read from disk again after a MySQL restart. But even if `O_DIRECT` is not used, the operating system's cache does not persist across a server reboot, so there are still cases where warmup will be slow.

Fundamentally, the easy fixes fail because they do not leave the buffer pool in the complex state that it reaches after running a real production workload for a long time. The LRU list must settle into that steady state for performance to settle, too. Without the features in XtraDB, the best way to warm a server up is to send it the `SELECT` portion of a production server's workload. We have built tools to do this for clients in the past, and such tools are useful for many tasks. But even though this will warm the server up correctly, it does not shorten the warmup period.

## What Are High-Leverage Use Cases?

Where are users most likely to see dramatic benefits from using the fast-warmup capabilities of Percona Server with XtraDB? Based on our experience with consulting clients, we have identified the following use cases.

### Running Databases in the Cloud

Users of cloud computing platforms will appreciate the fast warmup features in Percona Server with XtraDB. Cloud providers are now offering servers with large amounts of memory, which is one condition that causes slow warmup. The other is slow I/O performance, which is generally a problem with cloud computing. Even the fastest I/O offered in the cloud is slower than locally attached disks such as those used in our benchmark. As a result, a large database in the cloud usually runs in-memory, and is generally very slow to warm up. Percona Server with XtraDB offers a significant benefit for typical large databases in the cloud.

### Achieving High Availability

High Availability (HA) requires fast failover time, and fast failover is impossible with a long warmup period. For example, a popular HA solution is DRBD (Distributed Replicated Block Device), which mirrors a server's disk over the network to another disk. If the primary fails, the secondary takes over. This generally requires the secondary to perform a filesystem recovery, InnoDB crash recovery, and server warmup. The server warmup is by far the slowest part of this process in servers with large amounts of memory.

Percona Server with XtraDB solves this problem. The periodic automatic dump of the LRU list is a file like any other, and is replicated to the secondary DRBD server as usual. After failover, the filesystem journal replay and InnoDB crash recovery will proceed as normal, but then instead of a long warmup period, the database server will be warmed up and ready for production traffic almost immediately.

## Accelerating Dynamic Scaling

One reason MySQL became popular on large-scale websites is the ability to use a “scale-out” strategy, with many replicas to serve read queries and offload the master. Applications that experience unpredictable spikes in demand often require new replicas to be added as needed. Indeed, some cloud computing platforms can even spin up new replicas automatically, and we are likely to see this approach adopted more widely in the future. This is called “dynamic scaling.”

Percona Server with XtraDB offers a compelling advantage in dynamic scaling scenarios. The time required to put a server into production decreases dramatically—and becomes more predictable—when the server can be warmed up rapidly. The only requirement is to copy the LRU dump file to the new server.

## Accelerating Restores from Backups

Predictable recovery time is invaluable for restoring backups. We hope that you are testing your backups regularly, including measuring the time and resources required for the restore, so that you do not face a surprise when you least desire it. If you are, then you know your true recovery time, because with Percona Server with XtraDB, you can back up the dumped LRU list to avoid the warmup period after a recovery.

## Shortening Cycle Time

We have experienced a variety of painful situations where a long restart time prevented a valuable activity. These include upgrades, security fixes, configuration changes, intrusive debugging, testing, and benchmarking. Percona Server with XtraDB shortens the restart time enough to make many of these activities practical.

## Acknowledgments

Thanks to [Jeremy Cole](#), for the original idea and initial implementation of the LRU dump. Thanks to [Mark Callaghan \(Facebook\)](#), [Ewen Fortune \(Percona\)](#), and [Morgan Tocker \(Percona\)](#) for reviewing this paper.

## About Percona Software

Percona is committed to producing open-source software for Percona Server, MySQL, and MariaDB users. We offer a range of our own software, and also participate actively in many non-Percona software projects. All of our software is open-source and free of charge.

**Percona Server** is an enhanced 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 XtraDB** is an enhanced version of the InnoDB storage engine. Storage engines are a unique feature of the MySQL database architecture. They are the software that stores and retrieves the data, and executes queries at the lowest level. MySQL supports a large variety of storage engines with differing characteristics. The user can choose which storage engine is best suited for each table, based on features such as ACID compliance, full-text indexing, and clustering. InnoDB is the most popular general-purpose OLTP storage engine. It is transactional and ACID compliant, with foreign keys, row-level locking, and an advanced MVCC (multi-version concurrency control) architecture. It is stable and mature, and has been in production use for many years. Percona XtraDB builds on this foundation with improved performance and scalability, and adds a number of useful features.

**Percona Server with XtraDB** is the combination of Percona Server and the XtraDB storage engine. It includes a companion hot-backup tool, **Percona XtraBackup**, which can also back up standard MySQL and InnoDB data. XtraBackup can make non-blocking backups while the server is running, without interrupting the database's normal operation.

## About Percona

Percona provides commercial support, consulting, training, and engineering services for MySQL databases and the LAMP stack. If you would like help with your database servers, we invite you to 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.

*Percona, XtraDB, and XtraBackup are trademarks of Percona Inc. InnoDB and MySQL are trademarks of Oracle Corp.*