PERCONA
Performance Consulting Experts

MySQL Performance for Rails

Baron Schwartz, Percona Inc.

Recent Trends in Web Apps

- Lots of memory
- Cloud computing
- Database sharding
- Agile development techniques
- Many-CPU hardware
- Flash and hybrid storage (SSD, FusionIO)
- Non-ACID databases (NoSQL)
- Sophisticated ORMs and frameworks

Hardware Cost-to-Performance

- Power and cooling
- Rack space
- Procurement and sticker price
- Operations: app complexity, failures

Causes of Downtime

- 20% Operating environment
  - Init scripts, DNS, control panels, misconfiguration
- 20% Change control
  - Upgrading, not upgrading
- 20% Data loss
  - Mistakes, hardware failure, disk too full
- 15% Performance problems
  - Bad queries, bad indexing, lack of capacity planning
- 15% Replication
  - Broken, delayed, data differences

Causes of the Causes

- System complexity
- Inexperienced staff
- Poor leadership
- Accepting defaults
- Wrong application architecture

Performance & Reliability Tip #1

Make things as simple as possible, but no simpler.

- Einstein

Add exactly as much redundancy as needed.

- Your Humble Servant

PERCONA
Performance Consulting Experts

What does a Database Need?

- Memory
- CPU
- Storage
- Network

Old Performance Truths

- As of early 2008, with 5.0 stable and 5.1 beta,
  - MySQL (InnoDB) couldn't use many CPUs
  - MySQL (InnoDB) couldn't use much I/O capacity
  - MySQL (InnoDB) couldn't use much memory
- That's 3 of 4 resources you couldn't scale up!
- The key bottleneck was InnoDB
  - Mutex contention
  - Hard-coded defaults from the olden days
  - Inefficient internal algorithms

Even Worse...

- You couldn't measure what MySQL was doing.
- You cannot improve what you cannot measure!

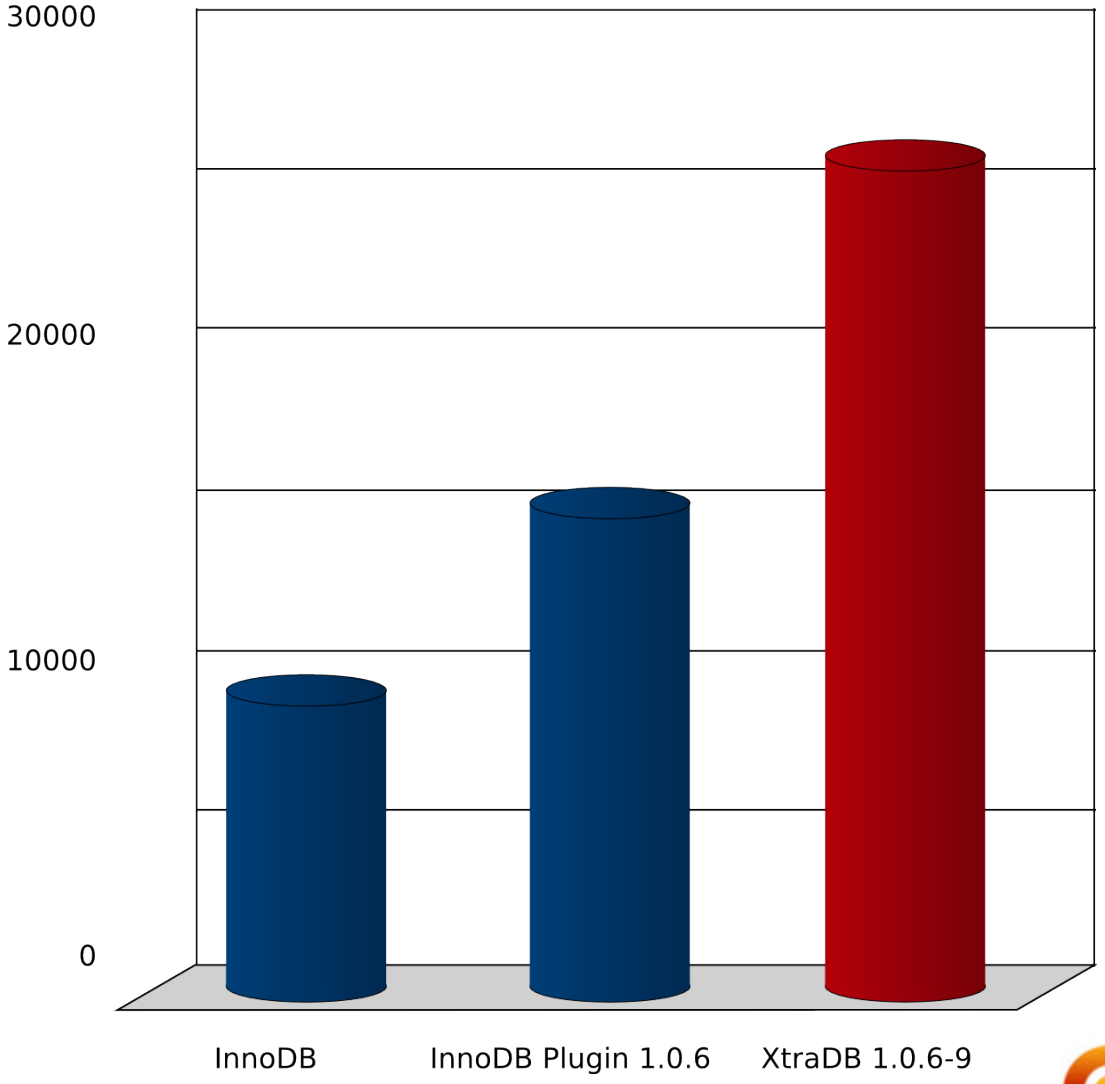
Three Major Problems

- Scalability limitations
- Single-node performance limits
- Lack of instrumentation

What's Changed?

- InnoDB Plugin for MySQL 5.1 (long since GA)
- Google's enhancements
- Percona's enhancements
- Improvements in MySQL 5.5 (not yet GA)

XtraDB (in January!)



PERCONA
Performance Consulting Experts

Better Performance in a Nutshell

- Use InnoDB Plugin or XtraDB
- Set the buffer pool BIG
  - Reserve some memory for the OS, do not use a ratio
- Set innodb_flush_method=O_DIRECT
  - You must have a good RAID controller with BBU
  - Also set swappiness to 0 and block scheduler to deadline
- Enable multiple read and write IO threads
  - innodb_[read|write]_io_threads = <number of disks>
- Increase innodb_io_capacity to ~150 * num disks

Better Performance, Cont'd

- Set the innodb_log_file_size to at least an hour's worth of writes
  - Watch Innodb_os_log_written
- Use a separate purge thread (XtraDB only)
  - Set innodb_use_purge_thread = 1
- Use adaptive checkpointing
  - Set innodb_adaptive_checkpoint = estimate
- Set skip_name_resolve
- Disable the query cache
  - Set query_cache_type = 0, set query_cache_size = 0

Performance Analysis

- Performance = Response Time.
  - The key is to understand, measure, and optimize this
- Use the “slow query log”
  - Set long_query_time=0
  - Analyze the results with mk-query-digest from Maatkit
- Advanced techniques
  - Use after application optimization is exhausted
  - Learn to use strace, oprofile, and gdb
  - Check out poormansprofiler.org, it is not a toy

What's Characteristic of Rails?

- Rather simple queries, easy to optimize and cache
- The N+1 query pattern
  - Find & destroy with mk-query-digest
  - Write as JOIN instead
- The ping-are-you-alive pattern
  - Someone just needs to fix this, really; http://tinyurl.com/379all9
- Running SHOW FIELDS constantly
- Storing sessions in the database

Current Scaling Possibilities

- Today we can scale *much* bigger than in 2008
  - Hundreds of GB of RAM
  - Dozens of CPUs
  - Tens of thousands of IOPS

* This is as far as we've pushed with commodity hardware. We can probably go a lot further.

What's Still Not Great?

- Friends don't let friends use the query cache
  - It is not scalable
  - It is a square peg in a round hole for web apps anyway
- Replication
  - It is single-threaded on the slave
  - A busy master can easily outpace a powerful slave
- Lots of server-level things
  - InnoDB used to be the bad guy
  - Now InnoDB is much better, and other things are hot spots

New in MySQL 5.1

- Partitioning
  - Really just N tables under the hood
  - This changes things in interesting ways
  - Poor server instrumentation: hard to analyze & tune
- Row-based replication
  - Black-box implementation
  - When it is slow, it's hard to find out why
  - Poor server instrumentation: hard to analyze & tune

Summary

- Use InnoDB Plugin
- Consolidate and use more powerful hardware
- Don't just use defaults
- Simplify for better performance and reliability
- Be scientific