Tuning Linux for Your Database

What you need to know about the hardware/cloud, I/O, file systems, memory, CPU, network, and resources to improve database performance on Linux.

Author: Colin Charles
Presenters: Alexey Zhebel and Nickolay Ihalainen

PERCONA

Author

Colin Charles
Chief Evangelist at Percona

Worked at:

- MariaDB
- MySQL AB
- Fedora
- OpenOffice

MySQL Community Contributor of the Year in 2014

Presenters

Alexey Zhebel
Technical Writer at Percona

Nickolay Ihalainen
Senior Technical Services Engineer at Percona

PERCONA

Agenda

- Hardware or cloud
- I/O
- File systems
- Memory
- CPU
- Network
- Resources

Operating Systems

- Focus of this talk: Linux
- Distributions are different:
  RHEL 6 has `tuned`, while Ubuntu doesn't
- Versions are different:
  - RHEL 5: `ktune`
  - RHEL 6: `tuned`
  - RHEL 7: `tuned throughput-performance`

Hardware Considerations

1. CPU
2. Memory
3. **I/O**
   - Queries: from large table scans to index scans and row changes
   - Durability requires each transaction commit to be flushed to disk with `fsync()`

I/O

Type of storage:

- SAS or SATA
- Ethernet (NAS or DRBD)
- SSD
- Fusion IO (NVMe)

I/O Schedulers / Elevators

- CFQ: default, good for slow storage (SATA)
- noop: low latency storage (SSD)
- **deadline: multi-process, disk-intensive apps**
  - Most database workloads benefit from this scheduler (goal is to prioritize process I/O)
  - boot parameter `elevator=deadline`
  - `echo "deadline" > /sys/class/block/sda/queue/scheduler`
- blk-mq (Multi-Queue Block IO Queueing Mechanism) - NEW
- http://dimitrik.free.fr/blog/archives/2012/01/mysql-performance-linux-io.html
- https://www.percona.com/blog/2009/01/30/linux-schedulers-in-tpcc-like-benchmark/

Comparing I/O Schedulers

	1 thread tps	1 thread ioutil%	8 threads tps	8 threads ioutils%
deadline*	15000	<1%	19100	<1%
deadline	6850	32%	15250	45%
cfq	5880	45%	1240	94%

* `innodb_flush_log_at_trx_commit=0`

File Systems

- **ext4**: rw, data=ordered, nosuid, noatime, nobarrier
- **XFS**: nobarrier

http://lwn.net/Articles/283161/

- Turn off I/O barriers
- Separate files (/data/logs/undo) or not?
- Flash-backed write cache (5-7k fsync() per sec)
- Battery-backed write cache (3-15k fsync() per sec)
- `iostat -dmx <interval>`

SSD / Flash

SSD:

- Much more IOPS than traditional disks
- Low latency
- Make sure you have a new RAID controller

“Flash”

- More IOPS than SSD
- Lower latency
- You can disable InnoDB doublewrite buffer (MariaDB 10.0+, Percona Server 5.6+)
- Parallel doublewrite buffer in PS 5.7

RAID

- RAID0: fast writes, reads, no redundancy
- RAID1: slower writes, fast reads
- RAID5: slow for random writes, fast for sequential writes, fast reads and slow recovery
- RAID10: fast reads and writes

LVM

Pros:

- Easily expand disk
- Snapshot backup

Cons:

- 2-3% performance overhead
- Snapshot penalties

Memory

- More RAM -> less I/O requirements
- Use ECC RAM: log errors to dmesg/mcelog
- “Swap insanity” on NUMA architectures
  - `numactl --interleave=all mysqld &`
  - Twitter patch / Percona Server 5.6+
    - `numa_interleave` option
    - `innodb_buffer_pool_populate`
      (NUMA decisions when buffer cache is clean)

https://blog.jcole.us/2010/09/28/mysql-swap-insanity-and-the-numa-architecture/

Why does MySQL need memory?

- Session buffers
  - Sort buffer, temp tables
- Metadata / locking
  - Index statistics, table definitions
- Caching data
  - Decrease reads: faster response time
  - Decrease random writes: queues write in cache, perform more sequential writes

Where else is memory used?

- File system cache
  - Binary / relay logs
  - MyISAM data relies on file system cache
  - PostgreSQL and MyRocks uses the concept of write-ahead logging (WAL)
- MySQL cache
  - `innodb_buffer_pool_size` (pages)
  - MyISAM's `key_buffer_size` (indexes)
- Per-session buffers
  - `sort_buffer_size`, `join_buffer_size`, `read_buffer_size`, `tmp_table_size`
- `free -m` / `SHOW ENGINE INNODB STATUS`

Transparent huge pages

- 2M (or even 4M) instead of standard 4K pages in Linux
- vm.nr_hugepages=8192
- MySQL benefits from huge pages, MongoDB and Redis does not
- You should turn off huge pages on latest Percona Server because of TokuDB

Swappiness

- Controls how aggressively the system reclaims “mapped” memory
- Default is 60
  (vm.swappiness=0 is fine, but not for newer kernels where =1 is better)
  - Decrease for more aggressive reclaiming of unmapped pagecache memory
  - Increase for more aggressive swapping of mapped memory
- 3.5 kernels change behavior
  (this was backported to RHEL 2.6.32-303)
- https://www.phoronix.com/forums/forum/hardware/processors-memory/31339-sandy-bridge-performance-hit-w-pthread-condition-variables-contended-mutexes

CPU

- Default power scheme is `ondemand`.
  It is generally better to use `performance` governor:

```
echo “performance” >
/sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
```

- Do not activate the `powersave` scheme
- Use tools such as `cpufrequtils` and `cpupower` to configure
- Check BIOS settings that might limit CPU frequency
  (turbo mode, C-state, OS power control)

Network

- Tools: `tc`, `dropwatch`
- Use gigabit Ethernet (more bandwidth = less latency)
- Can also trunk/bond for HA
- `net.ipv4.tcp_no_metrics_save=1`
- Enable ARP (address resolution protocol) filter to prevent ARP flux
  `echo 1 > /proc/sys/net/ipv4/conf/all/arp_filter`
- Set MTU (max transmission unit) size to 9000 (jumbo frames)
- `sar -n DEV`
  (network statistics for eth0, eth1, etc)

Network continued

- Settings in `/etc/sysctl.conf`
  - Allow more connections to queue up:
    `net.ipv4.tcp_max_syn_backlog = 4096`
  - Increase kernel packet buffers:
    `net.core.netdev_max_backlog = 4096`
  - Increase socket connection wait queue:
    `net.core.somaxconn = 4096`
  - Reduce TCP timeout that comes after closing socket (default 60 = 1min):
    `net.ipv4.tcp_fin_timeout = 30`
- Runtime: `echo 4096 > /proc/sys/net/ipv4/tcp_max_syn_backlog`

Network final thoughts

- Fully synchronous replication (Percona XtraDB Cluster or NDBCluster): **As fast as your slowest node**
- Leave the first NIC for transactions and route gcomm traffic to the 2nd NIC
- `evs.send_window` + `evs.user_send_window` should be more than 512 (number of data packets in replication at a time)
  https://www.percona.com/blog/2016/03/14/percona-xtradb-cluster-in-a-high-latency-network-environment/
-

Resource limits

- `ulimit` (/etc/security/limits.conf)
  - -f (file size): unlimited
  - -t (cpu time): unlimited
  - -v (virtual memory): unlimited
  - -n (open files): 64000
  - -m (memory size): unlimited
  - -u (processes/threads): 32000
- check real limits with `cat /proc/`pgrep -xn mysqld`/limits`

Tune that database!

- Find slow queries and fix them
  - `pt-query-digest`
  - PMM: https://www.percona.com/software/database-tools/percona-monitoring-and-management
- Reduce locking or time waiting
  (run benchmarks for your app all the time)
- `mysqlslap`
- Configure database server settings! For example `/etc/my.cnf`

KVM / Xen / other virtualization

- Database performance depends mostly on hardware
- For KVM you can turn off caching at device level
- Native AIO used over threaded nowadays
- VMWare crash consistency:
  https://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1008542

Containers

- Kubernetes container manager: vitess.io, CoreOS, Docker
- Differentiate local vs remote benchmarking for apps (networks add overhead)
- Local: up to 25% loss for r/w workload on 1 thread (average is 15%)
- Remote: up to 32% is possible
- http://blog.felter.org/post/91483169880/some-mysql-benchmarking-under-docker
- http://www.davidmkerr.com/2014/06/postgresql-performance-on-docker.html

EC2

- Instance limitations!
- EBS: unpredictable I/O -> use RAID10 or RAID0
- RDS: similar performance between EBS RAID10 MySQL and RDS

Why Percona Server

- Threadpool
- NUMA interleaving
- Numerous performance fixes in XtraDB
  - Fast InnoDB restarts
  - Buffer pool warming
  - Parallel doublewrite buffer
  - etc.
- Can run Percona XtraDB Cluster

Hardware overall

- Test everything!
- Sometimes NICs are a bad batch or the driver is buggy
  (RHEL5 + some Broadcom NICs drop packets under high load)
- Sometimes there is a bad batch of disk

Hadoop / Hbase

- Running in EC2 is problematic
- Make sure name resolution works
- Enable bonding
- networking: hadoop uses big buffers (64MB blocks by default)
  - net.ipv4.tcp_rmem = 32768 436600 4194304
  - net.ipv4.tcp_wmem = 32768 436600 4194304
- http://docs.hortonworks.com/HDPDocuments/HDP2/HDP-2.1.3/bk_cluster-planning-guide/content/ch_hardware-recommendations.html

Other databases

- MongoDB
  https://docs.mongodb.com/manual/administration/production-notes/
- PostgreSQL
  https://wiki.postgresql.org/wiki/Tuning_Your_PostgreSQL_Server
- Neo4j
  https://neo4j.com/docs/operations-manual/current/performance/
- Riak
  http://docs.basho.com/riak/kv/2.1.4/using/performance/

SELinux and other security

- Easy way: disable!
- Better: learn SELinux
- It is like a firewall that you need to configure
- AppArmour on Ubuntu is the same

https://stopdisablingselinux.com/

Tools

- ps
- vmstat
- iostat
- top
- free
- sar -n DEV
- gdb
- tcpdump
- strace
- perf
- htop

Percona Toolkit for MySQL

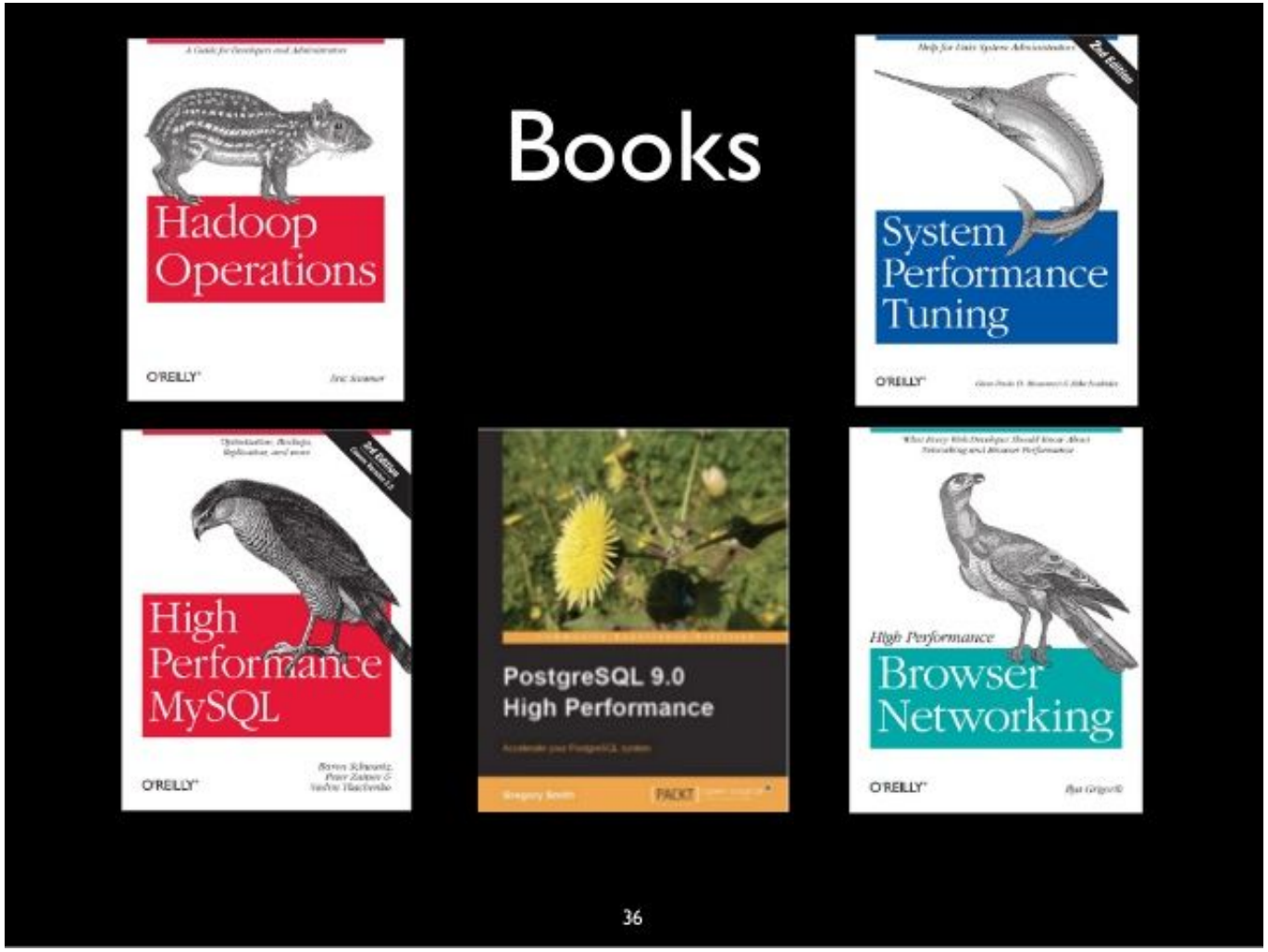
- pt-query-digest
- pt-summary
- pt-diskstats
- etc

Percona Monitoring and Management

Benchmarking

- sysbench
  - oltp.lua
  - Use tables with 20M rows and 20M transactions
  - Check 1-128 threads/run
- LinkBench
- Yahoo! Cloud Serving Benchmark
- Google’s PerfKit Benchmarker

Books



PERCONA

Percona Live 2017: Call for papers deadline is November 13

Percona Live 2017 to take place April 24-27 in Santa Clara, CA.

Submission Guidelines

http://bit.ly/2exss8u

Submission Form

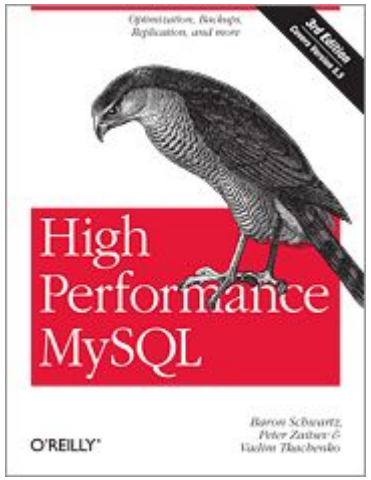
http://bit.ly/2e01oT2

PERCONA

We are hiring!

Get your friend hired at Percona

to receive *1000 USD* and a signed copy of *High Performance MySQL*



https://www.percona.com/about-percona/careers

PERCONA

DATABASE PERFORMANCE MATTERS