Optimizing MySQL for Solid State Storage

Vadim Tkachenko
Percona Inc, co-founder, CTO
Percona Live NYC 2011
May 26th, 2011
This talk

Flash technologies

- Server usage
- Not USB/digital camera flash cards

PCI-E and SATA cards

MySQL application
Revolutionary changes

- From spinning to solid state
- No mechanical moving parts
- Jump in performance
- Requires changes in applications
- In 5-10 years SSD will replace hard disks totally

Optimizing MySQL for Solid State Storage
Physics behind

“floating gate transistors”

• Non-volatile memory
• (more details)

One state – Single Level Cell (SLC)

• Faster, more reliable, more expensive

Many states – Multi Level Cell (MLC)

• Usually 4 states (2bits)
• 3bits (8 states)
• Slower, less reliable, cheaper
Types

NOR

- Random read access (bit granularity)
  - Speed compared with DRAM
- Slow write and erase
- Firmware storage

NAND (this talk)

- Faster writes
- Only block-level read access (4K)
- Idea is to compact many cells in limited space
  - Make competition with Hard Disk Drives
Erase (rewrite) challenges

Erase is to set all bits to “1111…”

- Erasing process is similar to “flash” in photocameras – there where name FLASH comes from
- Erase is slow, done in batch operation (up to 1MB)

Change “1”->”0” is fast

Change “0”->”1” is possible only by erasing

- 1st write: “1111” -> “1110”. Block marked as “written”
- 2nd write: even “1110” -> “1010” is not possible
Erase challenges

Erase is slow

- You want to erase many blocks in single flash
- Block management

When you write – card never writes the same block

Background process to run garbage collector
Erase lifetime

**SLC**
- 100,000 times per cell (may vary)
- ~20 years lifetime

**MLC**
- 10,000 times per cell (2cell)
- 5,000 times (3 cell)
- ~5 years lifetime

Big capacity and even distribution (wear leveling) prolongs lifetime
## SSD types

### SATA
- 200-500MB/sec
- Intel X25-M/E, OCZ, Unigen

### PCI-E
- Over 1GB/sec, 70,000 req/sec, under 1ms response time
- FusionIO, Virident

### SAN
- Violin memory
PCI-E cards

Fast. Very fast.

PCI-E, closest to CPU

Shares host memory / CPU

Most complex part – firmware

Space reservation for heavy writes
PCI-E cards drawbacks

Expensive. Very expensive.

- $18,999.00 / 640GB = 30$ / GB
- On level with memory prices

Performance depends on space

Not hot-swap
SATA SSD

Good performance

Much cheaper, but

Requires engineering work

- How to attach
- What RAID controller
- Not all models are equal
SSD for MySQL / Percona Server

IO performance: 1GB/sec – 70,000 req/sec

• A lot, but MySQL can’t use that all
MySQL basic setups

- Everything on SSD (ibdata, ib_logfiles)
  - 5-7x difference
MySQL basic recommendations

XFS, better with 4k blocks

- Mkfs.xfs -s size=4096
- Mount –o nobarrier

Multiple threads

- Percona Server or InnoDB-plugin or MySQL 5.5

Still uses about 5,000 req/sec, ~200MB/sec
MySQL what can be improved

Single threaded sequential stuff

- InnoDB transactional logs with fsyncs
- Binary logs
- Doublewrite buffer (with whole ibdata)

RAID with BBU good place for them

- Up to 45% improvement
Percona Server tunings

- `innodb_flush_neighbor_pages` = ON | OFF
- `innodb_log_block_size` = 512 | 4096
- `innodb_page_size` = 4K | 8K | 16K
  - Use carefully
- `innodb_doublewrite_file`
- `Innodb_adaptive_checkpoint=keep_average`
- `innodb_log_file_size > 4GB`
Percona Server results

tpcc-mysql, 1000W, Virident, 144GB BP
Still not enough utilization

Single MySQL instance not able to utilize all IO

- Solution: several MySQL instances

Experiment

- Dell PowerEdge R815
- 4 physical AMD CPUs / 48 cores
- 144GB of RAM
- Virident tachIOn 200GB card
- Tpcc-mysql workload
  - 48 user connections
- Whitepaper “Scaling MySQL With Virident Flash Drives and Multiple Instances of Percona Server” on percona.com/about-us/mysql-white-papers/
Results

tpcc-like, 100GB datasize

<table>
<thead>
<tr>
<th>Memory Size</th>
<th>1 inst</th>
<th>2 inst</th>
<th>4 inst</th>
</tr>
</thead>
<tbody>
<tr>
<td>26GB</td>
<td>28157</td>
<td>47747</td>
<td>54574</td>
</tr>
<tr>
<td>52G</td>
<td>47747</td>
<td>86654</td>
<td>103170</td>
</tr>
<tr>
<td>120G</td>
<td>25298</td>
<td>151204</td>
<td>163773</td>
</tr>
</tbody>
</table>
Results conclusions

With 120GB memory single instance result worse then with 26GB
  • InnoDB contentions problems again

Two instances allows to improve 1.5x-6x times

I do not like multi-instance, but
  • Management complexity
  • Good scripts solve it
  • 2-3 instances seems reasonable
Dealing with space problems

Hot tables on SSD / Cold tables on disks

- MySQL does not have proper tablespace management
- Symlinks are pain to maintain

Use SSD as cache

- ZFS
- FlashCache
FlashCache

Developed and maintained in production by Facebook

OpenSource

Shows good/stable results in production

Drawbacks

• Not user friendly
• Kernel module – manual compilation
FlashCache details

- Write-through and write-back modes
- FIFO and LRU block management
- Configurable % of dirty pages
- Cache survive server reboot
- You need to compile kernel module by yourself
- ibdata1/ib_logs layout
  - Keep on non-FlashCache partition
Why you may want Flash

Performance

Scale up instead of scale out

Expensive one time investment, but saving on

• Amount of server
• Power consumption
• Datacenter space
More talks

Today 4:15pm

• “Tuning For Speed: Percona Server and Fusion-io” by Torben Mathiasen from FusionIO
Thank you!

Flash technologies are evolving

• A lot of research ahead

We can discuss more today

• Mickey Mantle open bar 6-8pm

vadim@percona.com