



PERCONA
Performance Consulting Experts

MySQL patches

OpenSQL 2008

Vadim Tkachenko,
Percona Inc,
MySQLPerformanceBlog.com

MySQL patches – why

- Concentrate on customer needs
- MySQL development very slow
 - No new features can be included in 5.0 / 5.1 for now
 - 6.0 will be released 2010 ? 2015 ?
- The same for InnoDB
 - Explosion with InnoDB plugin and then nothing
 - No roadmap, no timeline – what is future ?
- We need solution now – we can do that

Directions

- Diagnostic patches
 - Get from your performance consulting specific
- Performance patches
 - Again from our specific consulting – we see performance problem, we are trying to fix it
- Operations tasks
 - Replication, backup, syslogng etc
- Mainly 5.0 for now
 - But looking for 5.1 actively

Sources

- Google's patches
 - Great resource
 - But all-in-one patch keeps us busy to separate it on small parts
- Our own development
 - Expecting more now when Yasufumi joined us
- Other resources
 - Jeremy Cole's ideas
 - Active cooperation with OpenQuery and OurDelta builds

Diagnostic patches

- Microslow-innodb
- Userstats
- InnoDB IO Pattern
 - Show accesses to InnoDB tablespaces
- innodb_check_frag
 - Checking how fragmented access to InnoDB pages for given query
- [microsec_process](#)
 - Add microsecond granularity to PROCESSLIST
- innodb_show_hashed_memory
 - Memory consumed by InnoDB
- Innodb_show_buffer_pool_content
- [innodb_fsync_source](#)
- [innodb_extra_status](#)

Microslow

- Microsecond resolution for long_query_time and for query timing
 - Log queries into slow log even executed less than 1 sec

```
Query_time: 0.000659 Lock_time: 0.000070 Rows_sent: 0  
Rows_examined: 30 Rows_affected: 0 Rows_read: 30
```

- Show execution plan of query

```
# QC_Hit: No Full_scan: No Full_join: No Tmp_table: Yes  
Disk_tmp_table: No # Filesort: Yes Disk_filesort: No  
Merge_passes: 0
```

- Show InnoDB timing

```
# InnoDB_IO_r_ops: 1 InnoDB_IO_r_bytes: 16384 InnoDB_IO_r_wait: 0.028487  
# InnoDB_rec_lock_wait: 0.000000 InnoDB_queue_wait: 0.000000 #  
InnoDB_pages_distinct: 5
```

Microslow – in additional

- Log queries from slave thread
- Rows_affected / Rows_read - by UPDATE / DELETE / INSERT statements
- Show connection_id and schema

```
# Thread_id: 11167745 Schema: board
```

Userstats

- INFORMATION_SCHEMA.USER_, INDEX_, TABLE_, CLIENT_STATISTICS
 - Incredible useful for hosting providers

```
SELECT * FROM INFORMATION_SCHEMA.USER_STATISTICS WHERE user='mailboxer'\G
User: mailboxer
  Total_connections: 423579
Concurrent_connections: 0
  Connected_time: 1394886
    Busy_time: 114720
    Cpu_time: 114724
  Bytes_received: 5679315884
  Bytes_sent: 16224058396
Binlog_bytes_written: 4330410415
  Rows_fetched: 10426597
  Rows_updated: 7168933
  Table_rows_read: 18038699
  Select_commands: 4203981
  Update_commands: 4722392
  Other_commands: 859838
  Commit_transactions: 9778
  Rollback_transactions: 29
  Denied_connections: 0
  Lost_connections: 536
  Access_denied: 0
  Empty_queries: 1601633
```

Userstats

- We also use to analyze table / index access

```
mysql> select * from information_schema.index_statistics;
```

TABLE_SCHEMA	TABLE_NAME	INDEX_NAME	ROWS_READ
art55	img_out55	from_message_id	8242
art84	img_out84	from_message_id	178777
art119	article119	forum_id_3	10425
art114	forum114	site_id	5198
art108	img_out108	PRIMARY	557656
art84	forum84	site_id	35586
art103	forum_stats103	type	3301
art90	forum90	site_id	11989

- Find unused indexes after some period of work

InnoDB_io_pattern

- Shows what part of tables accessed
- Can be used to determine working set

```
mysql> select INDEX_ID, TABLE_NAME, INDEX_NAME, sum(N_READ), sum(N_WRITE) from  
INFORMATION_SCHEMA.INNODB_ALL_PAGE_IO group  
by INDEX_ID;
```

INDEX_ID	TABLE_NAME	INDEX_NAME	sum(N_READ)	sum(N_WRITE)
30	tpcc/item	PRIMARY	547	0
32	tpcc/district	PRIMARY	1	1
36	tpcc/history	GEN_CLUST_INDEX	11	5
37	tpcc/history	fkey_history_1	166	163
38	tpcc/history	fkey_history_2	37	30
39	tpcc/new_orders	PRIMARY	76	76
43	tpcc/order_line	PRIMARY	218	189
44	tpcc/order_line	fkey_order_line_2	1040	1040
46	tpcc/stock	PRIMARY	3137	1764
47	tpcc/stock	fkey_stock_2	269	0
48	tpcc/customer	PRIMARY	960	580
49	tpcc/customer	idx_customer	171	0
50	tpcc/orders	PRIMARY	94	70
51	tpcc/orders	idx_orders	142	129

InnoDB_show_hash_memory

- Where memory is allocated to

```
-----  
BUFFER POOL AND MEMORY  
-----  
Total memory allocated 15882679960; in additional pool allocated 1048576  
Internal hash tables (constant factor + variable factor)  
  Adaptive hash index 267687368      (203998552 + 63688816)  
  Page hash           12750664  
  Dictionary cache   159487552      (153001160 + 6486392)  
  File system        334072      (82672 + 251400)  
  Lock system        31879928      (31875512 + 4416)  
  Recovery system    0          (0 + 0)  
  Threads            410296      (406936 + 3360)  
Buffer pool size     786432  
Buffer pool size, bytes 12884901888
```

InnoDB_show_buffer_pool

- What is in buffer_pool right now

```
select space,offset, RECORDS, DATASIZE, INDEX_NAME,TABLE_SCHEMA,TABLE_NAME from
information_schema.INNODB_BUFFER_POOL_CONTENT limit 10;
```

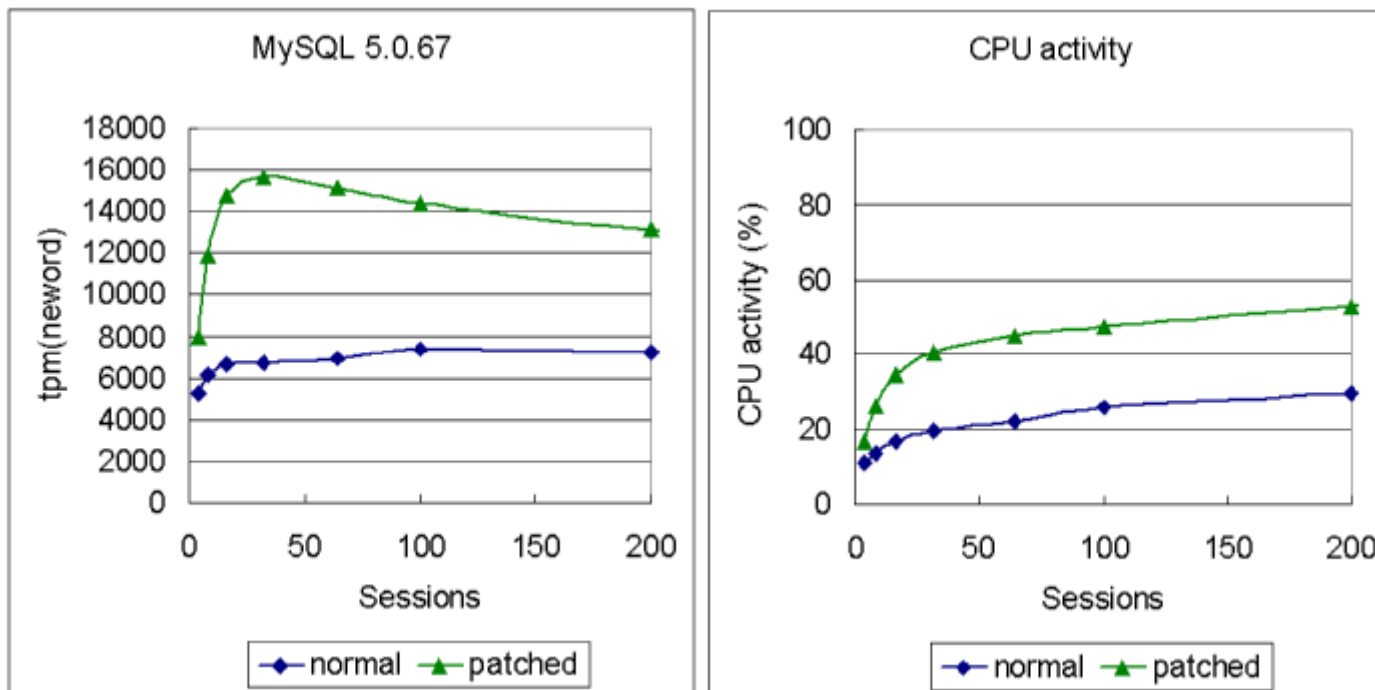
space	offset	RECORDS	DATASIZE	INDEX_NAME	TABLE_SCHEMA	TABLE_NAME
1584	640643	9	10312	PRIMARY	art104	article104
1648	2100	135	15226	PRIMARY	art114	author114
1492	4507	158	15130	PRIMARY	art87	author87
1406	17498	141	16056	img_status	art52	img_out52
1466	47632	49	15140	PRIMARY	art62	img_out62
1470	1395457	24	14769	PRIMARY	art84	article84
1460	16025	62	15174	PRIMARY	art61	img_out61
1458	560956	20	14977	PRIMARY	art61	article61
1466	67953	56	15182	PRIMARY	art62	img_out62
1621	162962	46	15134	PRIMARY	art110	link_out110

Performance patches

MySQL patches

Split_buffer_pool_mutex

- Additionally separate buffer_pool mutex into several
 - (graph for combined split_buffer_pool & io_patches), 8 cores box

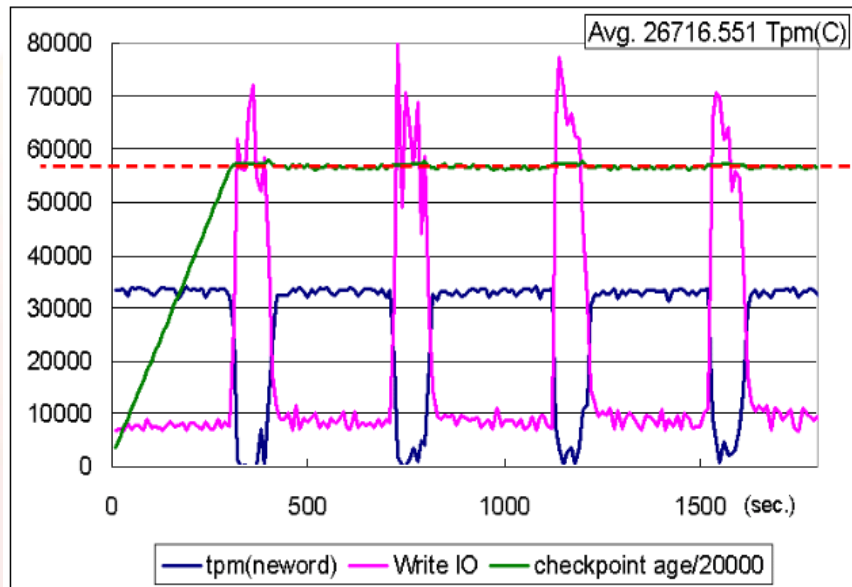


InnoDB_io_patches

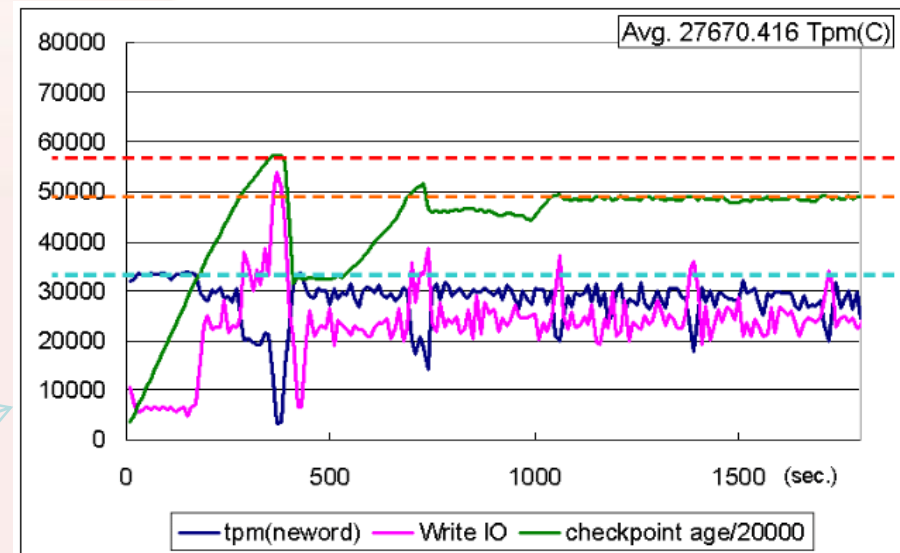
- **innodb_read_io_threads** - the number of background IO threads for read requests.
- **innodb_write_io_threads** - the number of background IO threads for writing dirty pages from the buffer pool.
- **innodb_read_ahead** - control native read-ahead behavior of InnoDB.
 - disable read-ahead,
 - enable read-ahead for random access only,
 - enable read-ahead for sequential access only,
 - enable both of read-ahead feature.
- **innodb_io_capacity** (default **100**) - number of disk IOPs the server can do.
- **innodb_adaptive_checkpoint** (default **0**) - control the added feature *adaptive checkpointing*(*). **0**:disable *adaptive checkpointing*, **1**:enable *adaptive checkpointing*.

InnoDB_adaptive_checkpoint

- Flushing process does not hurt user queries



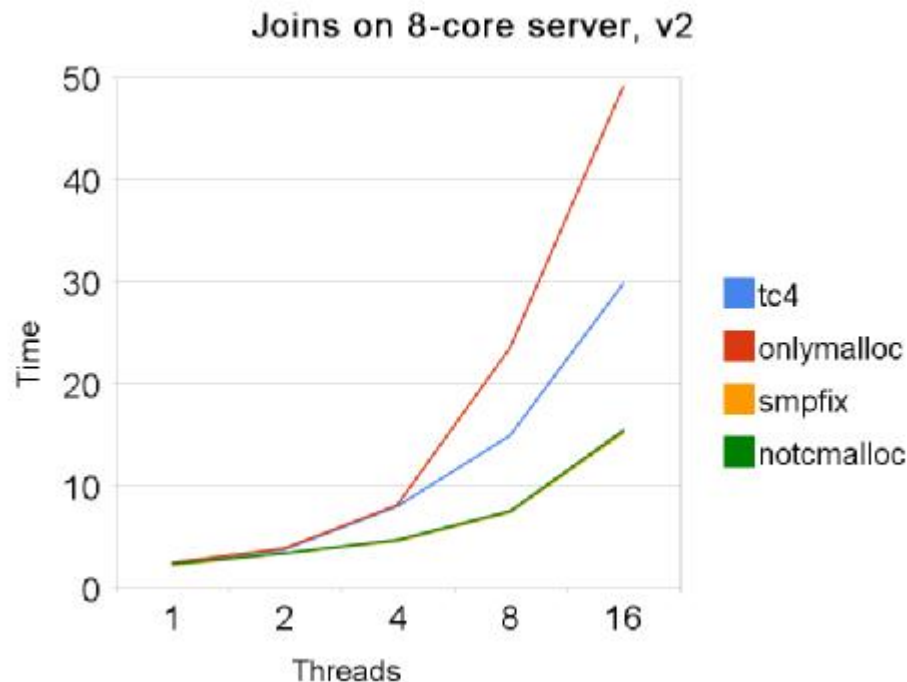
before



after

InnoDB_rwlock

- Similar to Google's smpfix
 - New implementation of InnoDB rw_locks
 - Helps on 8+ cores boxes
 - (graph from <http://code.google.com/p/google-mysql-tools/wiki/SmpPerformance>)

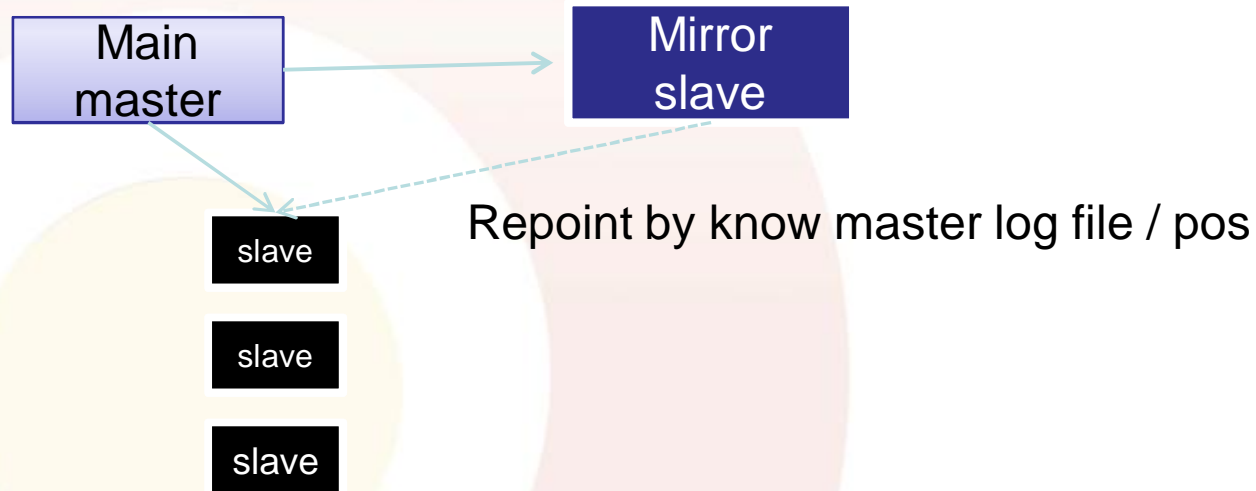


Replications

MySQL patches

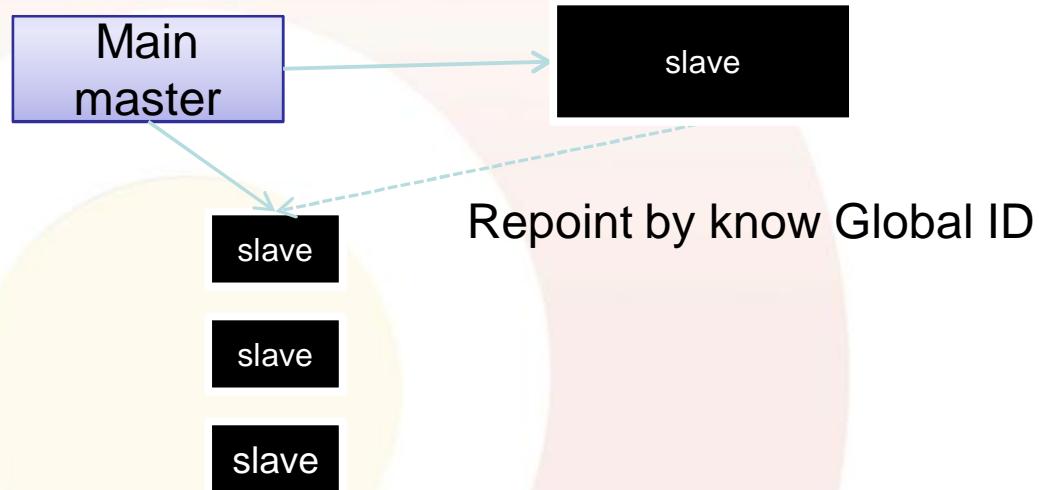
Mirror binlog

- Makes the exact copy of binary logs on slave
 - Easy repoint slaves to new master
 - Backups of binary logs



Global Transaction ID

- Patch is NOT available yet (replacement of mirror binlog)
 - Global ID of event for whole farm



Plans

- Our tasks
 - <http://www.percona.com/docs/wiki/devplan:start>
 - Depends on Customer demands, first priority – customer requests
- Our builds
 - <http://www.percona.com/docs/wiki/release:start>
- Short term plans
 - 5.1 porting
 - Incremental InnoDB backups
- Q & A
- Thank you!