

Fusion-io MySQL multi-instances report

Percona Inc.,
Vadim Tkachenko

Oct-2011

Contents

0	Introduction	3
1	Testing methodology	4
2	Results	6
2.1	2400W, 120GB memory	6
2.2	2400W, 64GB memory	7
2.3	1200W, 120GB memory	8
2.4	1200W, 64GB memory	9
3	Conclusions	10
A	Systems Specifications	11
A.1	Server configuration	11
A.2	Client configuration	11
B	MySQL configuration	12

0 Introduction

The goal of this research is get most performance from Fusion-io ioDrive card, used to run MySQL database server.

It is known that MySQL due internal limitations is not able to utilize all CPU and IO resources available on modern hardware. Idea is to run multiple instances of MySQL to gain better performance.

The research is sponsored by Fusion-io, Inc

1 Testing methodology

For tests we used `tpcc-mysql` package, which generates TPCC-like workload on MySQL systems.

1. Server hardware: Dell PowerEdge R815, details in Appendix [A.1](#)
2. Storage: Fusion-io ioDrive Duo 640GB MLC. Fusion-io driver version: 2.3.1 build 123; Firmware v5.0.7, rev 101971
3. Software: Percona Server 5.5.15, configuration in Appendix [B](#)
4. Client hardware: IBM x3650, details in Appendix [A.2](#)

`tpcc-mysql` tests were run for following combinations:

- 2400W, big buffer pool
 - 1 MySQL instance, 2400 warehouses (220GB of data), 120GB buffer pool
 - 2 MySQL instances, 1200 warehouses (110GB of data) each, 60GB buffer pool each
 - 4 MySQL instances, 600 warehouses (55GB of data) each, 30GB buffer pool each
- 2400W, small buffer pool
 - 1 MySQL instance, 2400 warehouses (220GB of data), 64GB buffer pool
 - 2 MySQL instances, 1200 warehouses (110GB of data) each, 32GB buffer pool each
 - 4 MySQL instances, 600 warehouses (55GB of data) each, 16GB buffer pool each
- 1200W, big buffer pool
 - 1 MySQL instance, 1200 warehouses (110GB of data), 120GB buffer pool
 - 2 MySQL instances, 600 warehouses (55GB of data) each, 60GB buffer pool each
- 1200W, small buffer pool
 - 1 MySQL instance, 1200 warehouses (110GB of data), 64GB buffer pool

- 2 MySQL instances, 600 warehouses (55GB of data) each, 32GB buffer pool each

The purpose of different combination of data and memory sizes, was to check how data/memory ratio affects results.

We used 48 user sessions and we performed 2700 sec long run, gathering data for New Order Transaction each 10 seconds.

That is, for each set of user sessions, we take 270 throughput measurements.

Based on this, we constructed the following metric, which we use as final result:

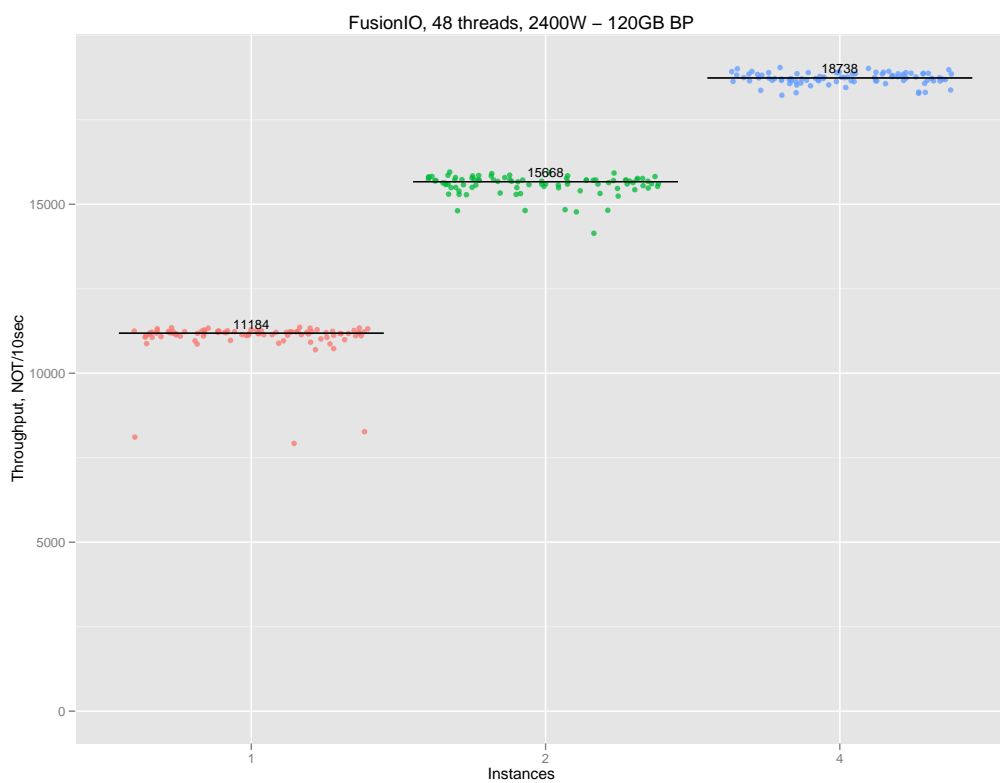
- Median Throughput for last 900 sec, to avoid warm-up influence on results.

2 Results

2.1 2400W, 120GB memory

Table 1: Results for 2400W, 120GB memory

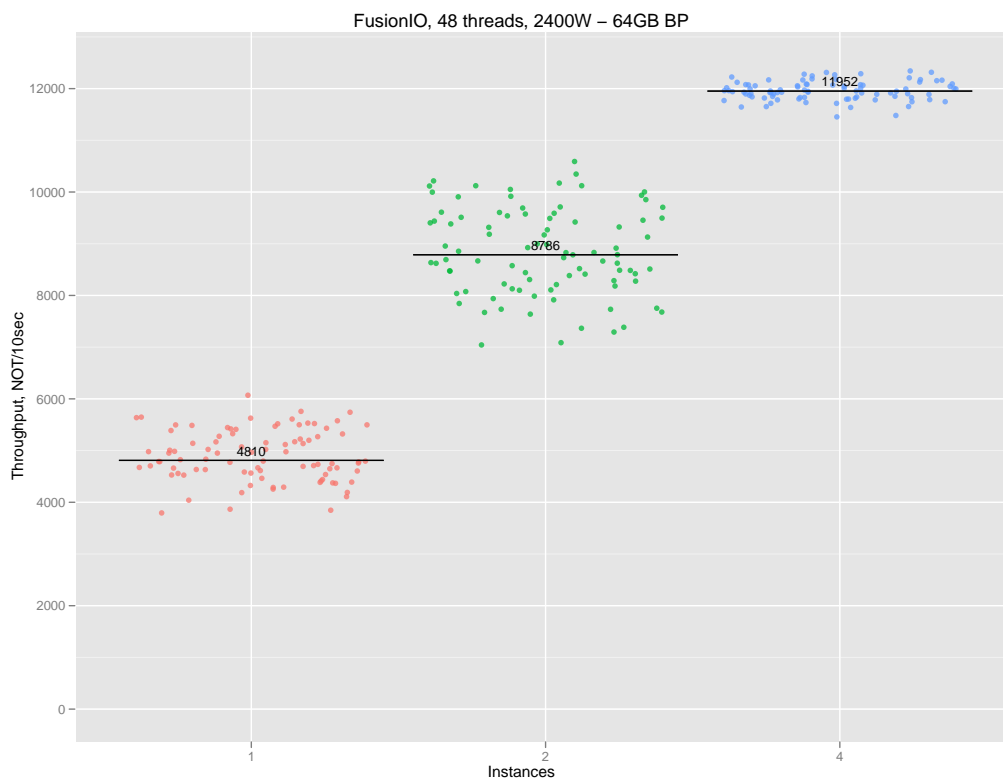
Instances	1	2	4
Median Throughput	11184	15668	18738
Ratio	-	1.4x	1.68x



2.2 2400W, 64GB memory

Table 2: Results for 2400W, 64GB memory

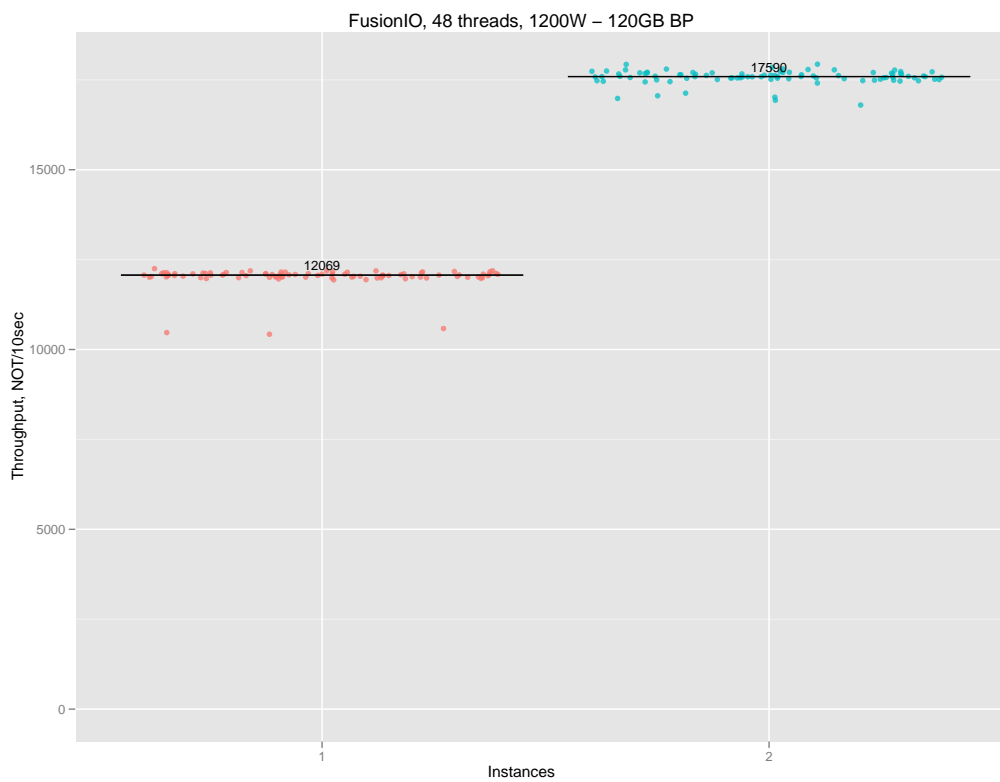
Instances	1	2	4
Median Throughput	4810	8786	11952
Ratio	-	1.83x	2.48x



2.3 1200W, 120GB memory

Table 3: Results for 1200W, 120GB memory

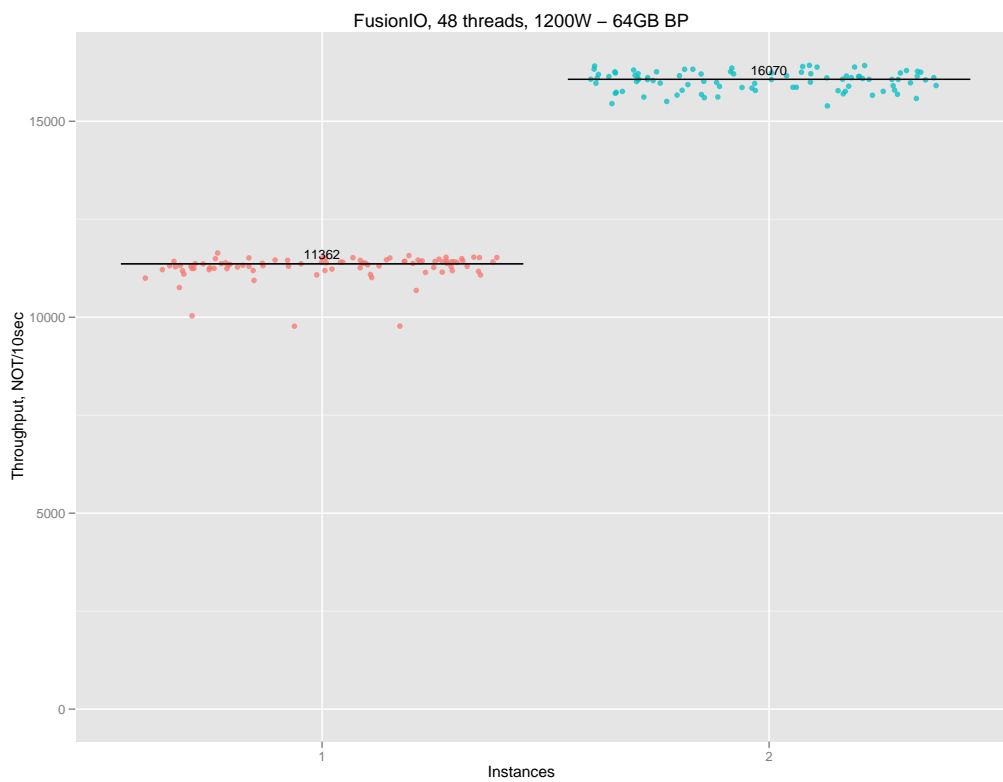
Instances	1	2
Median Throughput	12069	17590
Ratio	-	1.46x



2.4 1200W, 64GB memory

Table 4: Results for 1200W, 64GB memory

Instances	1	2
Median Throughput	11362	16070
Ratio	-	1.41x



3 Conclusions

In conclusion we can highlight:

- Running multiple instances shows good improvement in throughput. 1.4x-1.8x for 2 instances and 1.6-2.4x for 4 instances.
- If you have sharding environment which allows you separate database into multiple instances you may try 2-4 instances setup to get better overall throughput from your MySQL setup

A Systems Specifications

A.1 Server configuration

```
# Percona Toolkit System Summary Report #####
    Date | 2011-10-06 21:01:45 UTC (local TZ: PDT -0700)
    System | Dell Inc.; PowerEdge R815;
    Service Tag | FD9M6Q1
    Platform | Linux
    Release | Red Hat Enterprise Linux Server release 6.1 (Santiago)
    Kernel | 2.6.32-100.34.1.el6uek.x86_64
    Architecture | CPU = 64-bit , OS = 64-bit
    Threading | NPTL 2.12
    SELinux | Disabled
    Virtualized | No virtualization detected
# Processor #####
    Processors | physical = 4, cores = 48, virtual = 48,
               | hyperthreading = no
    Speeds | 48x1900.007
    Models | 48xAMD Opteron(tm) Processor 6168
    Caches | 48x512 KB
# Memory #####
    Total | 157.71G
```

A.2 Client configuration

```
# Percona Toolkit System Summary Report #####
    Date | 2011-10-06 21:06:53 UTC (local TZ: EDT -0400)
    System | IBM; System x3650 M2 -[7947AC1]-;
    Platform | Linux
    Release | CentOS release 5.6 (Final)
    Kernel | 2.6.18-238.12.1.el5
    Architecture | CPU = 64-bit , OS = 64-bit
    Threading | NPTL 2.5
    Compiler | GNU CC version 4.1.2 20080704 (Red Hat 4.1.2-50).
    SELinux | Permissive
    Virtualized | No virtualization detected
# Processor #####
    Processors | physical = 2, cores = 8, virtual = 16,
               | hyperthreading = yes
    Speeds | 16x1596.000
    Models | 16xIntel(R) Xeon(R) CPU X5570 @ 2.93GHz
```

```
Caches | 16x8192 KB
# Memory #####
Total | 94.39G
```

B MySQL configuration

```
[mysqld]
gdb

datadir=/mlc/data

#for SSD
#innodb_read_ahead = none
innodb_flush_neighbor_pages = 0
innodb_adaptive_flushing_method = keep_average
innodb_buffer_pool_restore_at_startup=300

#innodb_adaptive_hash_index_num = 16

#####fixed innodb options
innodb_file_per_table = true
innodb_data_file_path = ibdata1:10M:autoextend
innodb_flush_log_at_trx_commit = 2
innodb_flush_method = O_DIRECT
innodb_log_buffer_size = 256M

#innodb_max_dirty_pages_pct=30

innodb_buffer_pool_size = 110G

innodb_log_file_size = 4G
innodb_log_files_in_group = 2
innodb_log_block_size=4096

#innodb_doublewrite = true
innodb_doublewrite = 0

#####plugin options
innodb_read_io_threads = 16
innodb_write_io_threads = 16
innodb_io_capacity = 10000
```

```
#not innodb options (fixed)

port = 3306
back_log = 50
max_connections = 400
max_connect_errors = 10
table_cache = 2048
max_allowed_packet = 16M
binlog_cache_size = 16M
max_heap_table_size = 64M
sort_buffer_size = 4M
join_buffer_size = 4M
thread_cache_size = 1000
query_cache_size = 0
query_cache_type = 0
query_cache_limit = 2M
ft_min_word_len = 4
memlock
#default_table_type = InnoDB
thread_stack = 192K
tmp_table_size = 64M

server-id = 10

**** MyISAM Specific options
key_buffer_size = 8M
read_buffer_size = 1M
read_rnd_buffer_size = 4M
bulk_insert_buffer_size = 8M
myisam_sort_buffer_size = 8M
myisam_max_sort_file_size = 10G
#myisam_max_extra_sort_file_size = 10G
myisam_repair_threads = 1
myisam_recover

socket=/var/lib/mysql/mysql.sock
user=root
skip-grant-tables

[mysql]
```

```
no-auto-rehash  
socket=/var/lib/mysql/mysql.sock
```

```
[client]  
socket=/var/lib/mysql/mysql.sock
```