PCIe SSDを用いたMySQL 5.6と5.7 のパフォーマンス対決！[+α版]
～ MySQLの性能は、どこまで向上するのか ～

日商エレクトロニクス株式会社
マーケティング本部　SODCグループ
長井 伸次

自己紹介

- 1982年4月に日商エレクトロニクス株式会社入社
- Sybaseを使った銀行系システムの開発・保守を担当
- Oracleデータベースを使ったアプリケーション設計、開発、保守、およびパフォーマンス・チューニングなどのコンサルティング業務を担当
- Oracleデータベースのデータ移行、再編成などを行う製品のサポート、プロジェクトを担当
- MySQL互換の分散スケールアウト・データベース ClustrixDBの検証
- Huawei ES3000のパフォーマンス検証

検証目的

- Fusion-IO ioDrive2 と Huawei ES3000 の基礎体力比較
- MySQL v5.6とv5.7の性能比較
- MySQL v5.7でPCIe SSDカードの性能をどこまで引き出せるか

検証環境

CPU	Intel Xeon CPU E5-2630 v2 @ 2.60GHz（x2）
メモリ	128GB
HDD	300GB x2
PCIe SSD	Fusion-IO MLC PCIe ioDrive2 1.2TB
	Huawei ES3000 1.2TB
OS	CentOS 6.5（2.6.32-431.el6.x86_64）
データベース	MySQL Community Server 5.6.21
	MySQL Community Server 5.7.4 m14
負荷テストツール	fio 2.1.7
	ORION 11.1.0.7.0
	SysBench 0.5

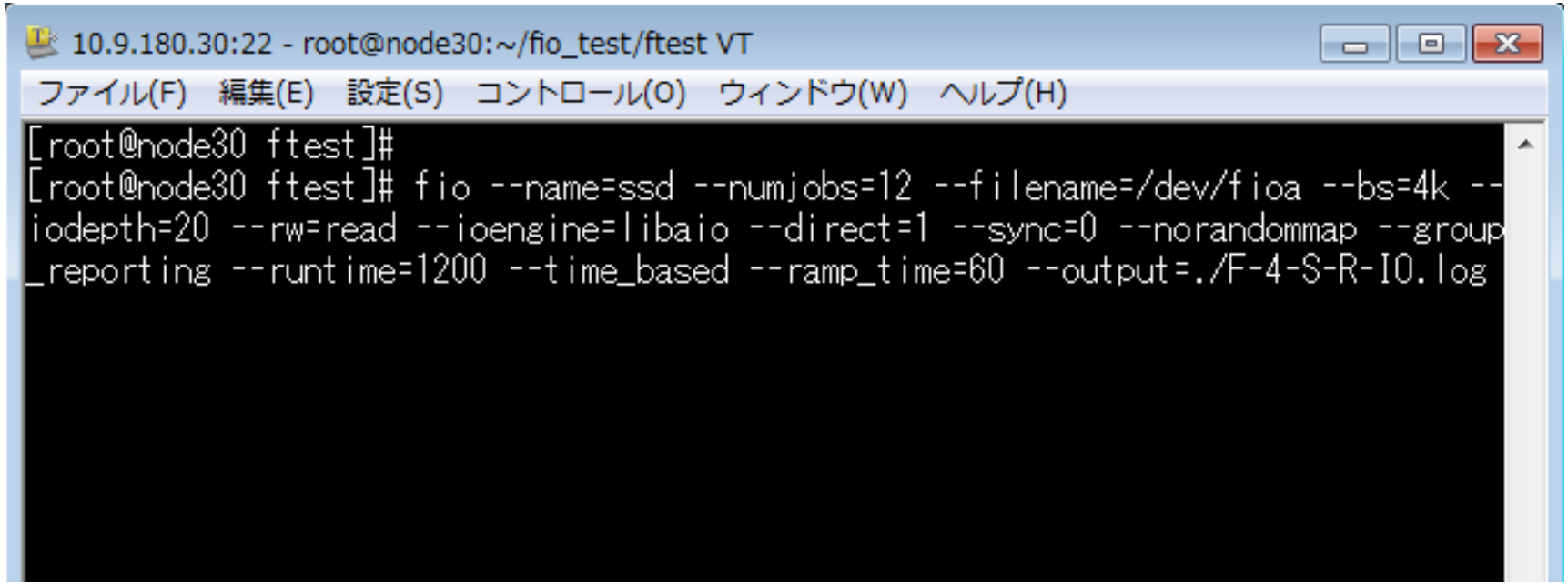
Fusion-IO ioDrive2 と Huawei ES3000 の基礎体力比較

カタログベースの基礎体力比較

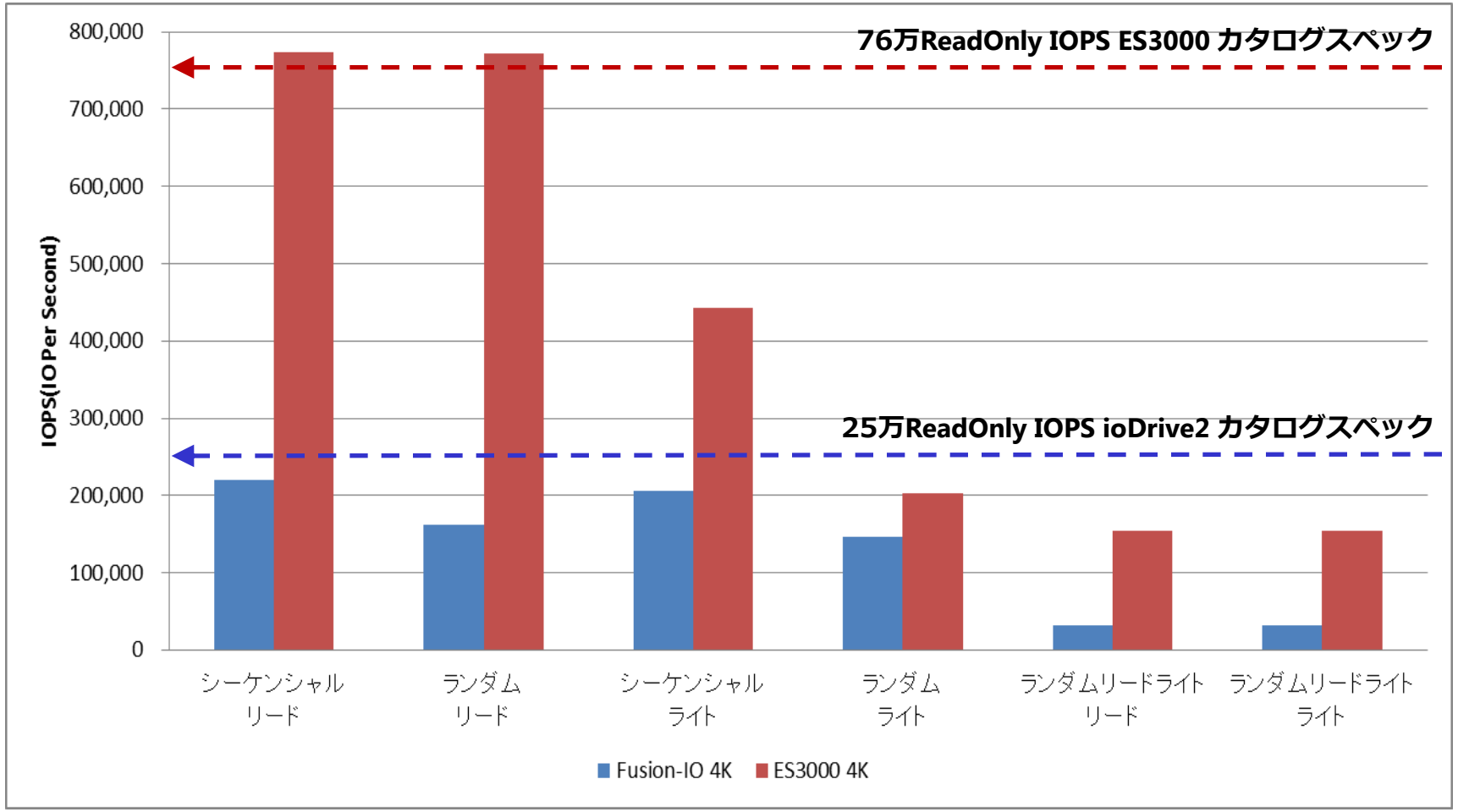
	Fusion-IO ioDrive2 1.2TB	Huawei ES3000 1.2TB
NANDタイプ	MLC	MLC
ランダム読取IOPS（4K）	245,000	760,000
ランダム書込IOPS（4K）	250,000	180,000
読み取り遅延	68μs	49μs
書き込み遅延	15μs	8μs

fioコマンド例

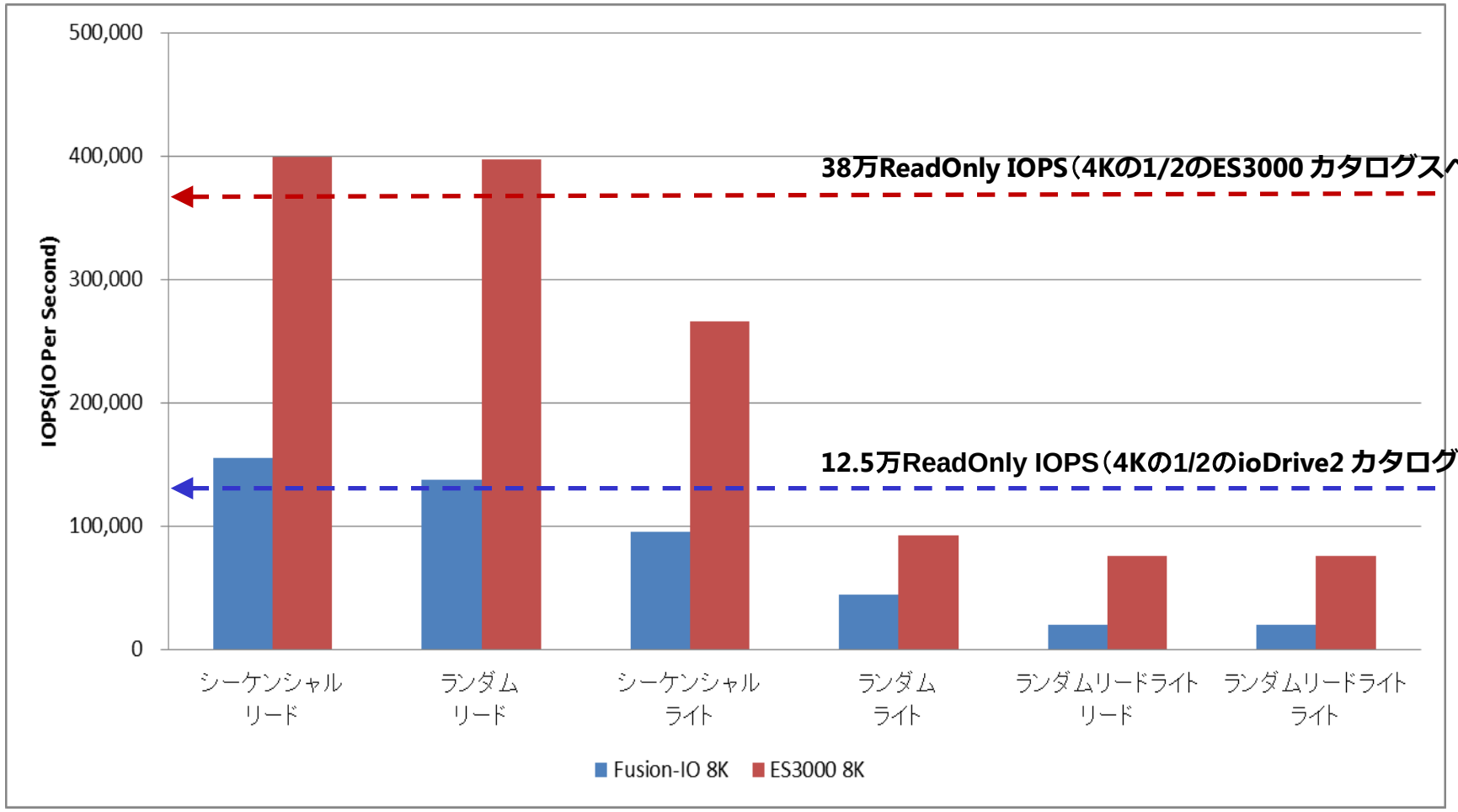
- ioDrive2 シーケンシャルリード ４KBの場合



fioによる基礎体力比較（4K）



fioによる基礎体力比較（8K）

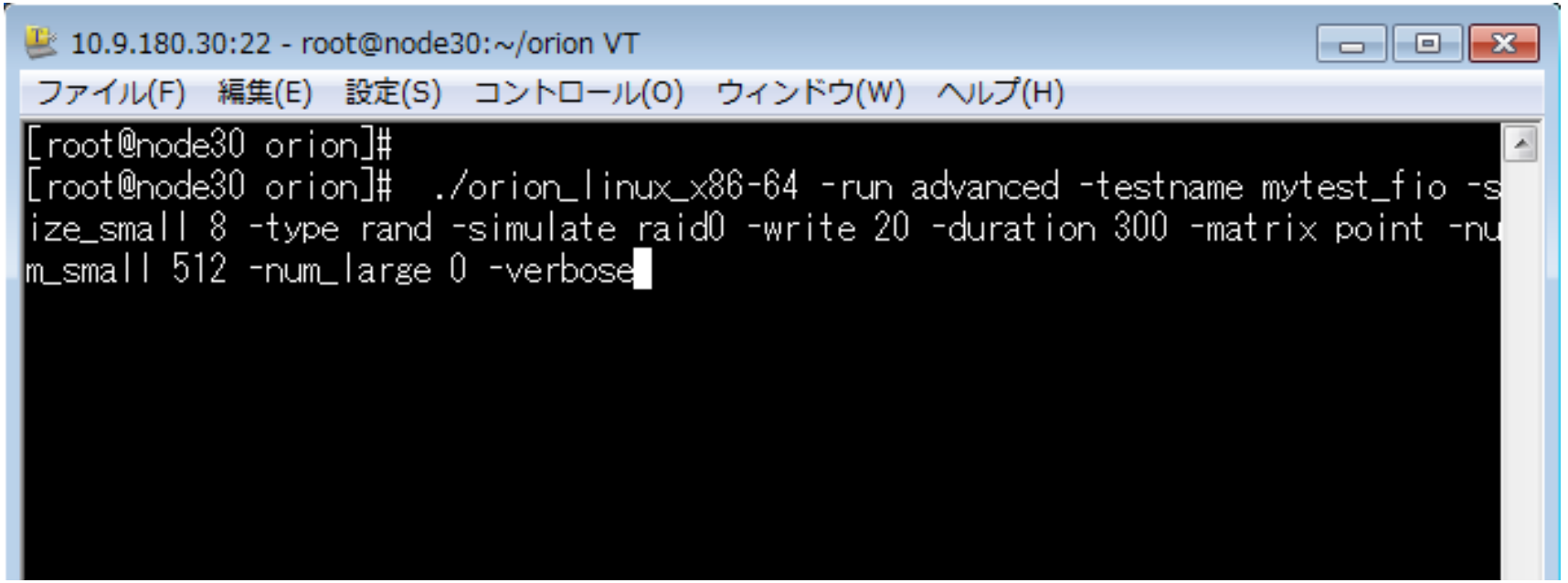


ORION(Oracle IO Numbers)

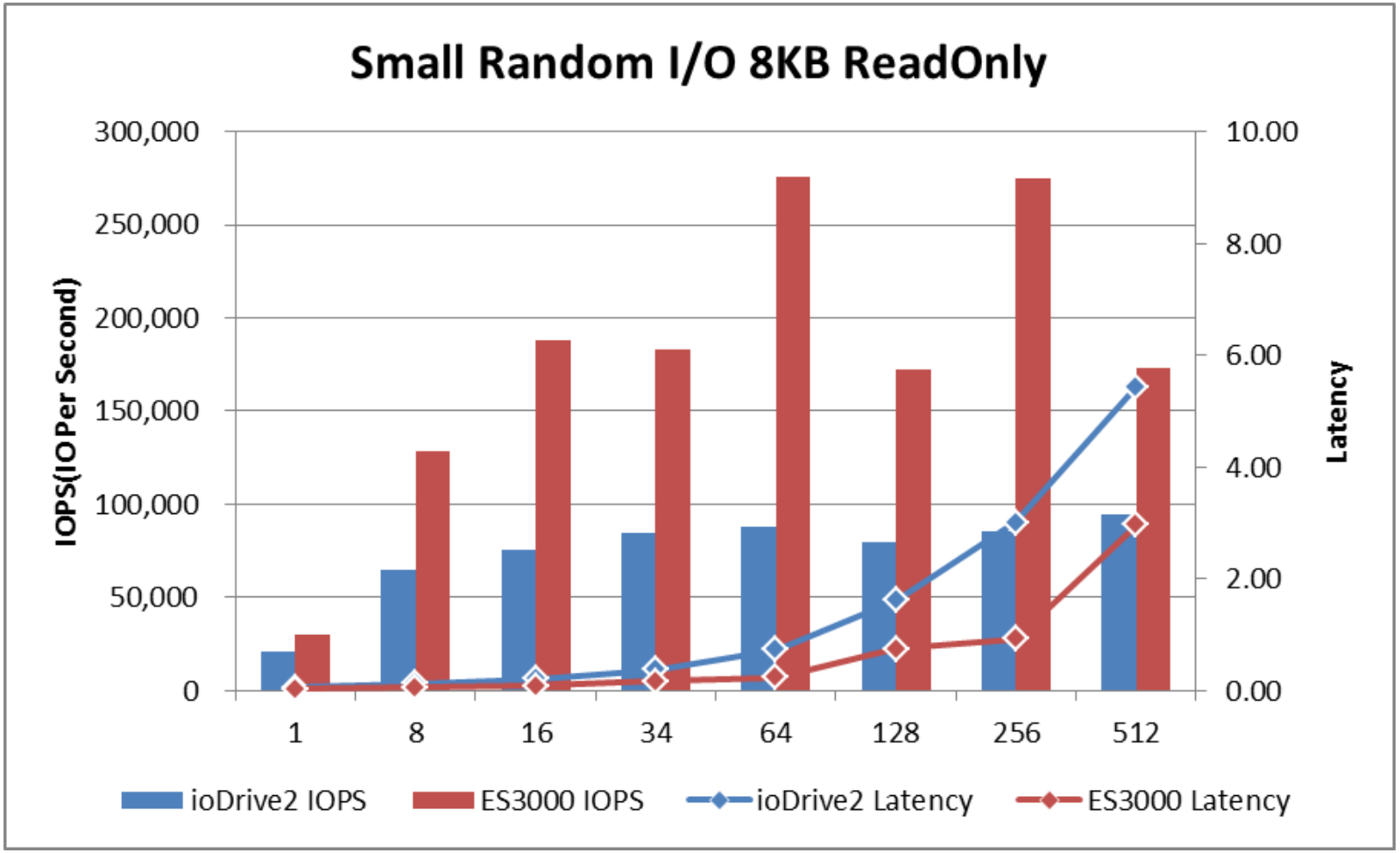
- Oracleのインストールやデータベース作成不要
- Oracleのワークロードをシュミレートして
  ストレージI/O性能を予測するフリーのツール
- Oracleと同じI/Oソフトウェア・スタックを使用
- シュミレートできる主なワークロード
  - OLTPアプリケーション
  - DWHアプリケーション

ORIONコマンド例

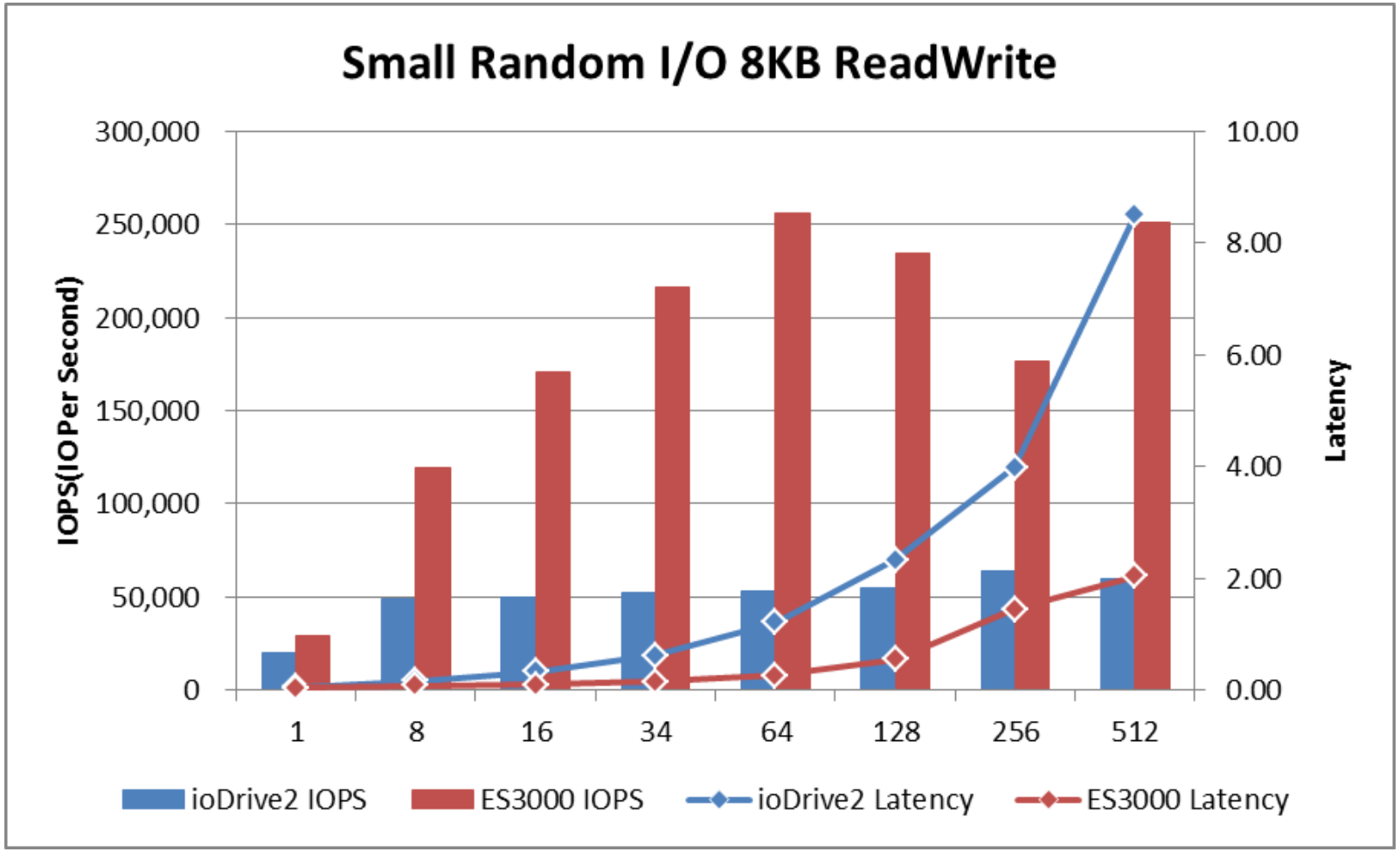
- ioDrive2 Small Random I/O 8KB ReadWrite 未処理I/Oの最大数 = 512の場合



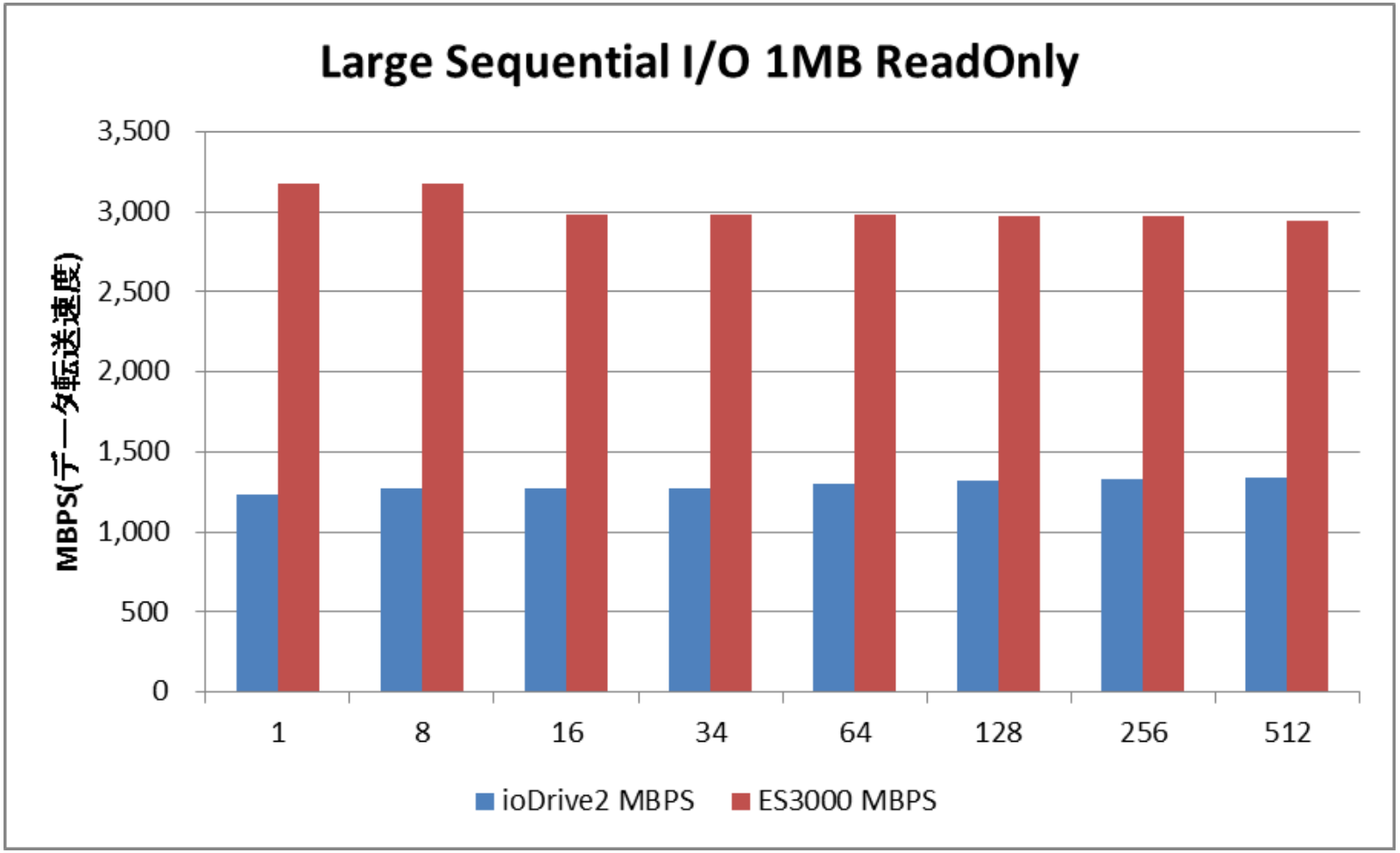
ORIONによる基礎体力比較(1/4)



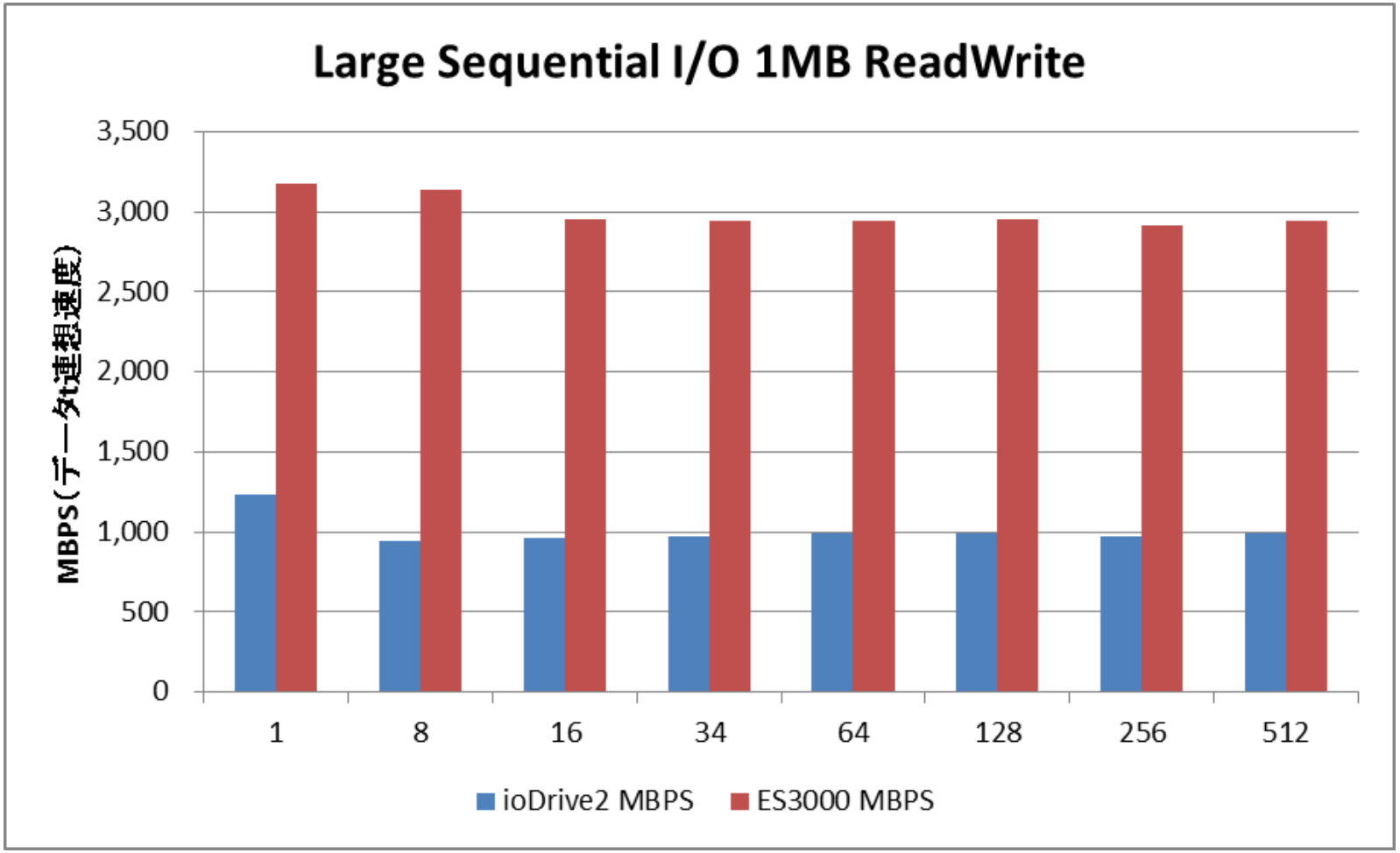
ORIONによる基礎体力比較(2/4)



ORIONによる基礎体力比較(3/4)



ORIONによる基礎体力比較(4/4)



基礎体力比較結果まとめ

- fioベンチマーク
  - 4KのSequential ReadとRandom ReadでES3000が77万IOPSを記録（カタログスペックとおり）
  - 8KのSequential ReadとRandom ReadでioDrive2とES3000が4Kの1/2のIOPSを記録
- ORIONベンチマーク
  - Small Random I/O 8KB ReadWriteモードでES3000が25万IOPSを記録
  - Small Random I/OでLatencyに反比例してIOPSが増減
  - Large Sequential I/OでES3000がioDrive2の２倍以上のデータ転送速度を記録

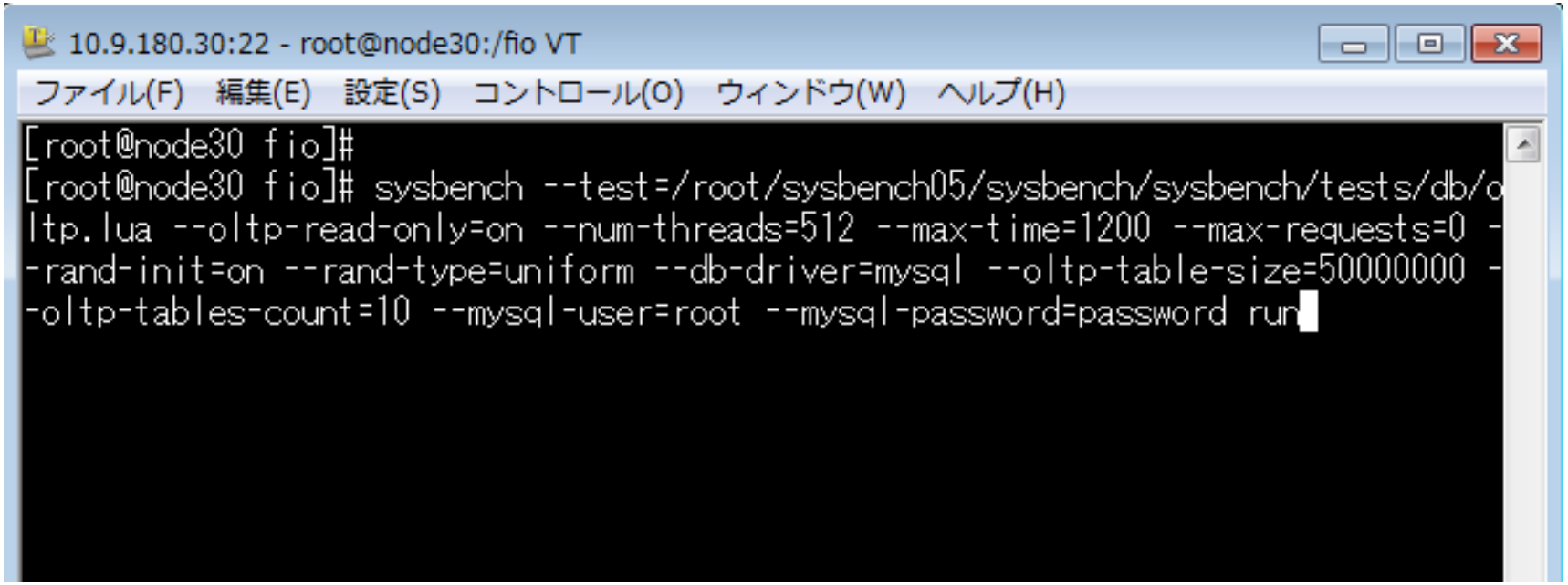
MySQL v5.6とv5.7の性能比較

検証方法

- innodbのページサイズをPCIe SSDの最適サイズに設定
- ファイルシステムのブロックサイズをPCIe SSDの最適サイズに設定
- innodbのキャッシュサイズを実装メモリの半分に設定
- 上記のPCIe SSD向けの簡単な設定で、どの程度性能の差が出るかを確認

Sysbenchコマンド例

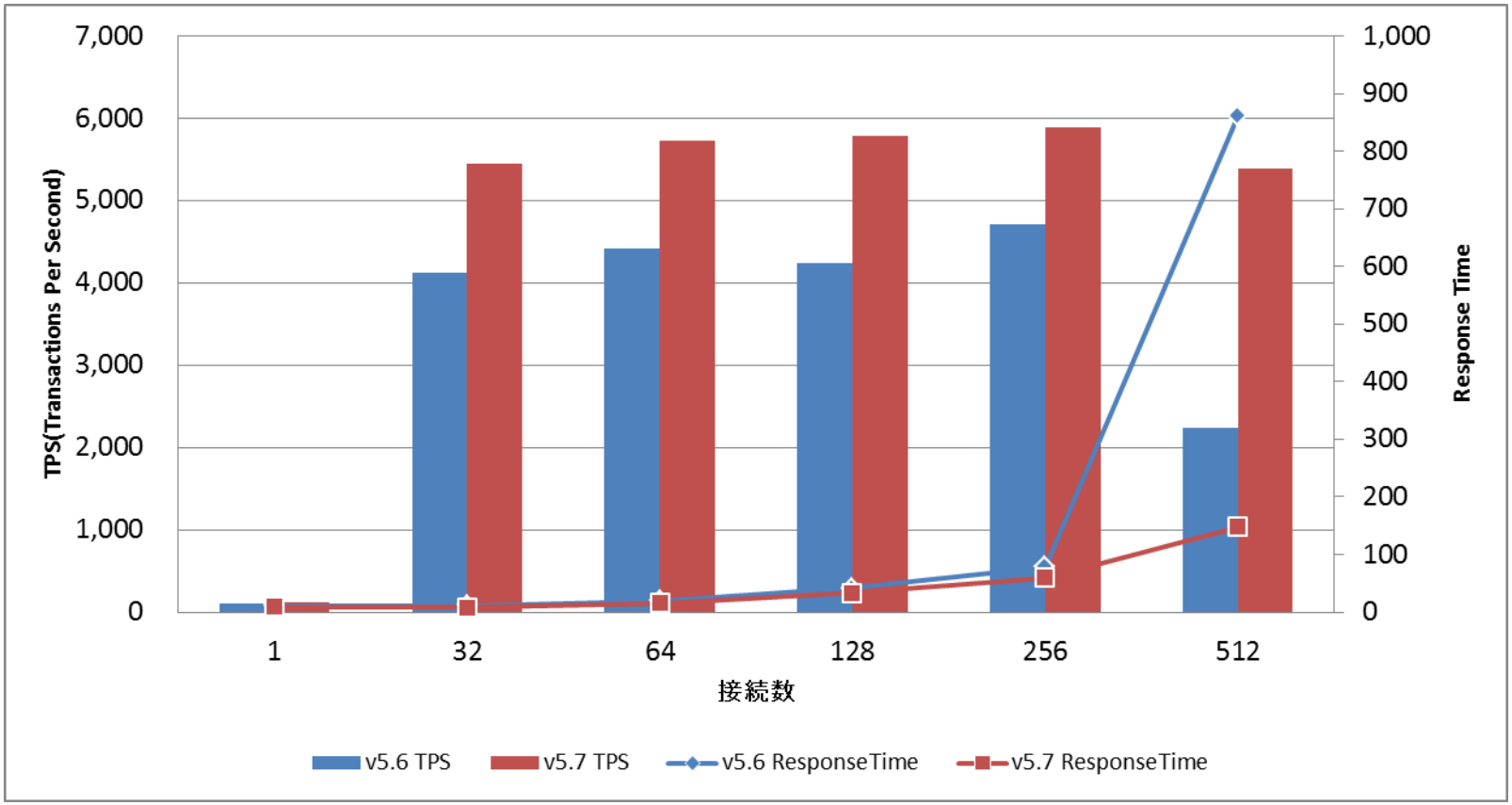
- ReadOnly モード 接続数 = 512 の場合



MySQL v5.6 と v5.7の性能比較のパラメータ設定（my.cnf）

パラメータ名	デフォルト	チューニング
innodb_buffer_pool_size	128MB	64GB
innodb_flush_method	fsync	O_DIRECT
innodb_flush_neighbors	1	0
innodb_page_size	16KB	4KB
innodb_read_ahead_threshold	56	0

MySQL 5.6 vs 5.7 ReadOnly(ioDrive2)

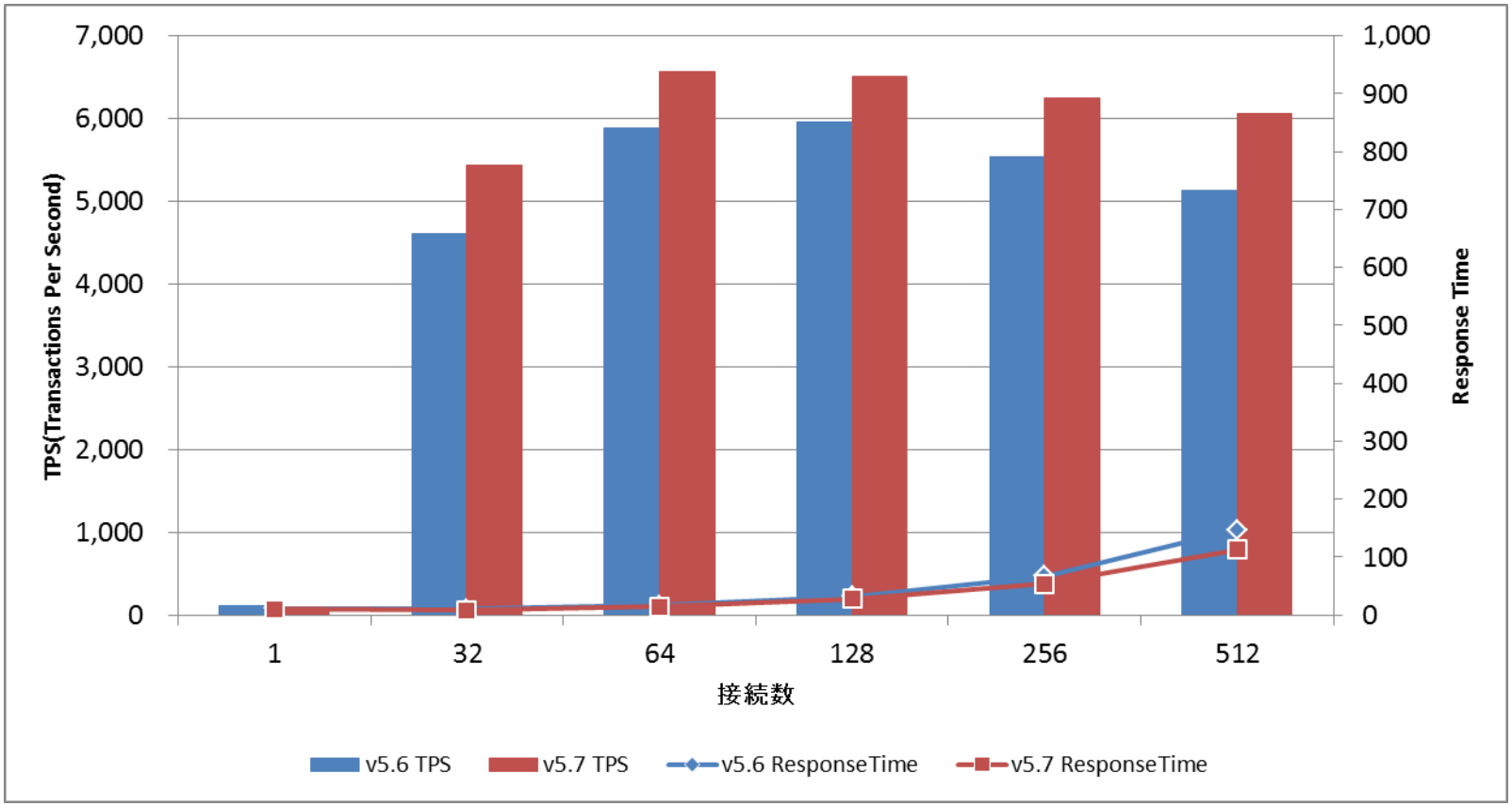


◆ファイルシステム：
・タイプ = EXT4
・ブロックサイズ = 4K

◆Sysbenchのパラメータ：
・oltp-table-size = 50,000,000
・oltp-tables-count = 10
・max-time = 1200

◆データベース：
・全体サイズ = 134GB(最適化なし)

MySQL 5.6 vs 5.7 ReadOnly(ES3000)



◆ファイルシステム：
・タイプ = EXT4
・ブロックサイズ = 4K

◆Sysbenchのパラメータ：
・oltp-table-size = 50,000,000
・oltp-tables-count = 10
・max-time = 1200

◆データベース：
・全体サイズ = 134GB(最適化なし)

MySQL v5.6 と v5.7の性能比較結果

- ioDrive2
  - 22% ～ 141%のパフォーマンスUP
  - v5.6よりもv5.7で高いTPSを記録
  - v5.7における使用を推奨
- ES3000
  - 9% ～ 12%パフォーマンスUP
  - v5.6でもv5.7でもioDrive2より高いTPSを記録
  - v5.6でもv5.7でも安定した高い性能を発揮

MySQL v5.7でPCIe SSDカードの性能をどこまで引き出せるか

検証方法

- ページサイズ、IO Capacity、IOスレッド数をPCIe SSD用に最適化
- バッファプールサイズを、128MBから48GBに変化させて、PCIe SSDの性能変化と特性を確認
- 同時接続数は、128
- イベント毎の待機時間の確認
  - MySQL の Performance Schema を利用
  - events_waits_summary_global_by_event_name テーブルからイベント毎にサマリーされた待機時間を取得
  - 待機時間の大きい順に15秒毎に上位5個をチェック

待機時間確認用PROCEDURE

Query 1

Limit to 1000 rows

```
1  DELIMITER $$
2  CREATE DEFINER=`root`@`localhost` PROCEDURE `dump_events_waits`(p0 VARCHAR(50),p1 INT,p2 INT)
3  BEGIN
4    DECLARE v1 INT DEFAULT 0;
5    DECLARE v2 INT DEFAULT 0;
6
7    WHILE p1 > v1 DO
8
9      TRUNCATE TABLE performance_schema.events_waits_summary_global_by_event_name;
10
11     SET v2 = SLEEP(p2);
12
13     SELECT p0 title,
14            CURRENT_DATE() cdate, CURRENT_TIME() ctime,
15            event_name, count_star,
16            sum_timer_wait/1000000000000 sum_timer_wait_s
17     FROM (SELECT *
18           FROM performance_schema.events_waits_summary_global_by_event_name
19           WHERE count_star > 0
20             AND event_name NOT IN ('idle','wait/synch/cond/sql/Item_func_sleep::cond')) AS a
21     ORDER BY sum_timer_wait_s DESC LIMIT 5;
22
23     SET v1 = v1 + 1;
24   END WHILE;
25 END$$
26 DELIMITER ;
27
```

MySQL v5.7 パラメータチューニング

- ページサイズ
  - PCIe SSDの最適サイズ に設定
  - ファイルシステムのブロックサイズと合わせる
- IO Capacityをfioの検証結果をもとに設定
- IOスレッド数の増加
- ログファイルサイズの拡張
- 読み込み／書き込み動作をPCIe SSD用に設定

MySQLパラメータ設定（my.cnf）

パラメータ名	デフォルト	チューニング
innodb_buffer_pool_size	128MB	128MB、1GB、8GB、16GB、32GB、48GB
innodb_flush_method	fsync	O_DIRECT
innodb_flush_neighbors	1	0
innodb_io_capacity	200	100000
innodb_io_capacity_max	2000	400000
innodb_log_file_size	48M	4GB
innodb_page_size	16KB	4KB
innodb_read_ahead_threshold	56	0
innodb_read_io_threads	4	12
innodb_write_io_threads	4	12
innodb_doublewrite	1	1 （stores all data twice）
innodb_flush_log_at_trx_commit	1	1 （full ACID compliance）

データ整合性重視

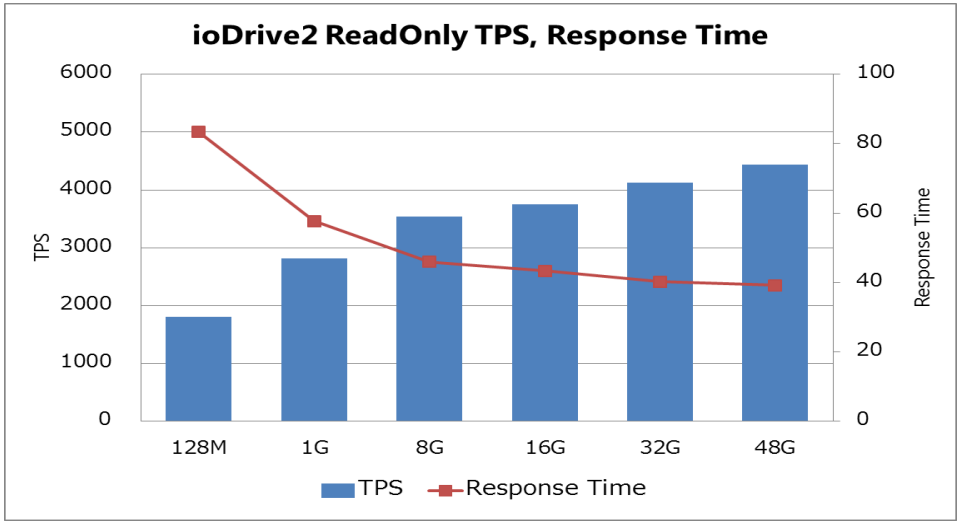
ファイルシステムとデータベース属性

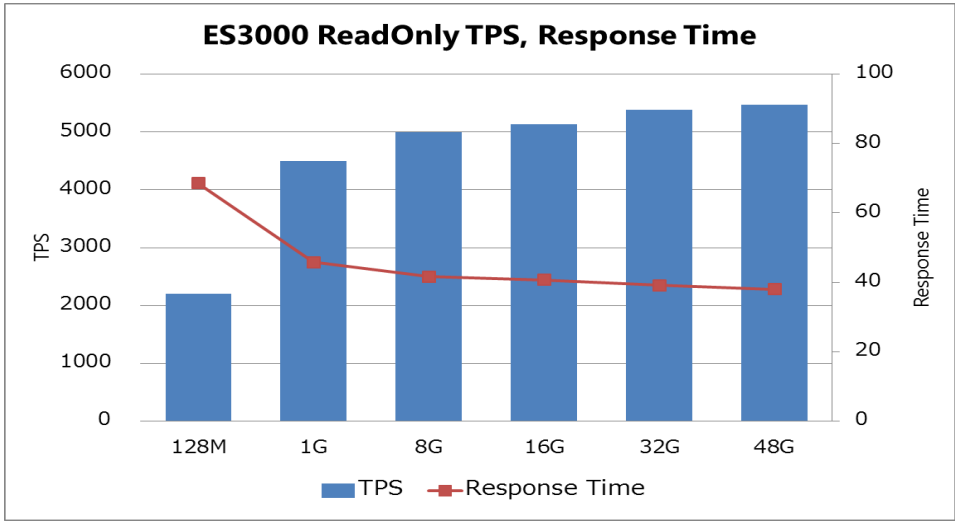
ファイルシステム属性	
タイプ	XFS
ブロックサイズ	4K

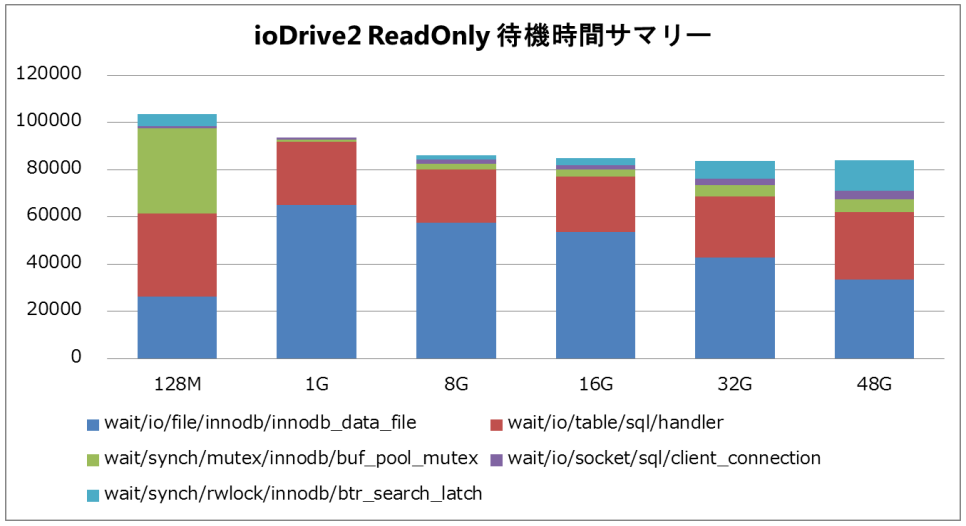
データベース属性	
全体サイズ	127G
データファイルの最適化	実施済

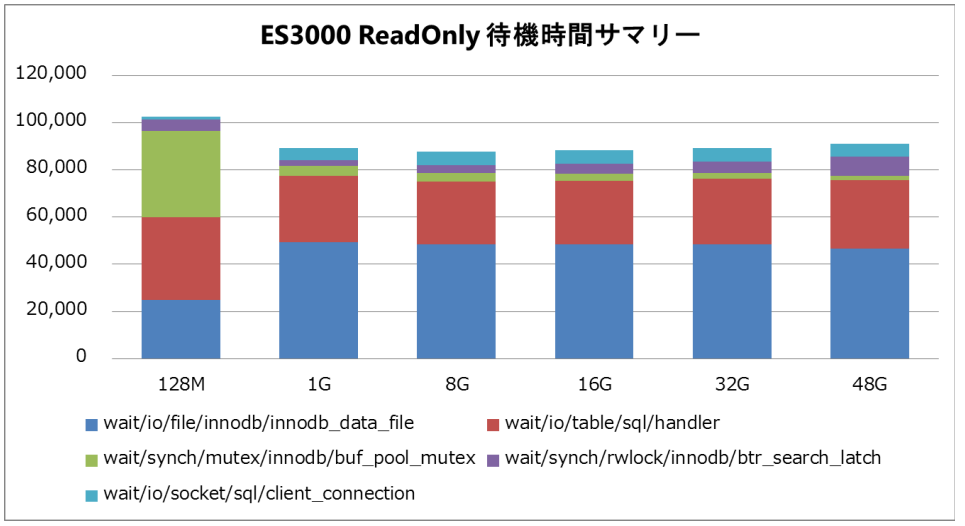
その他	
I/Oスケジューラ	cfq （default）

128 Users ReadOnly モード

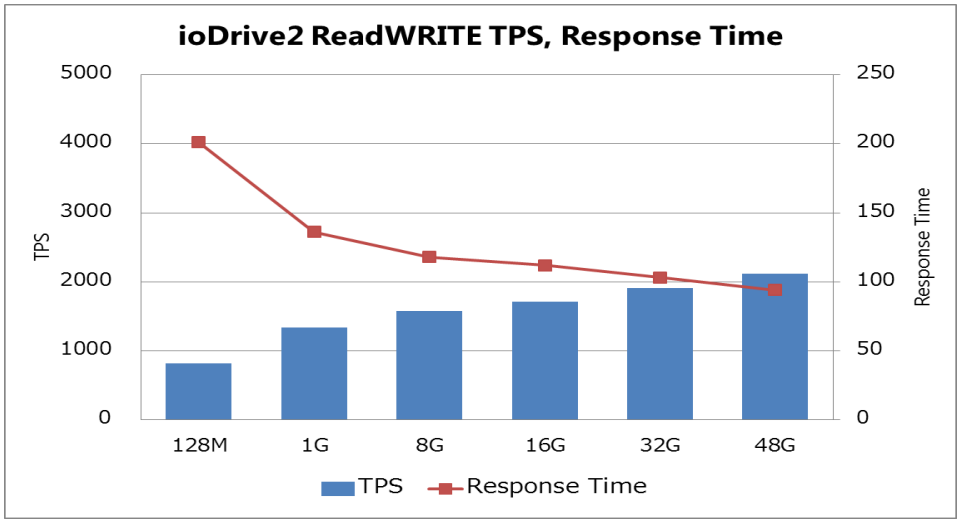


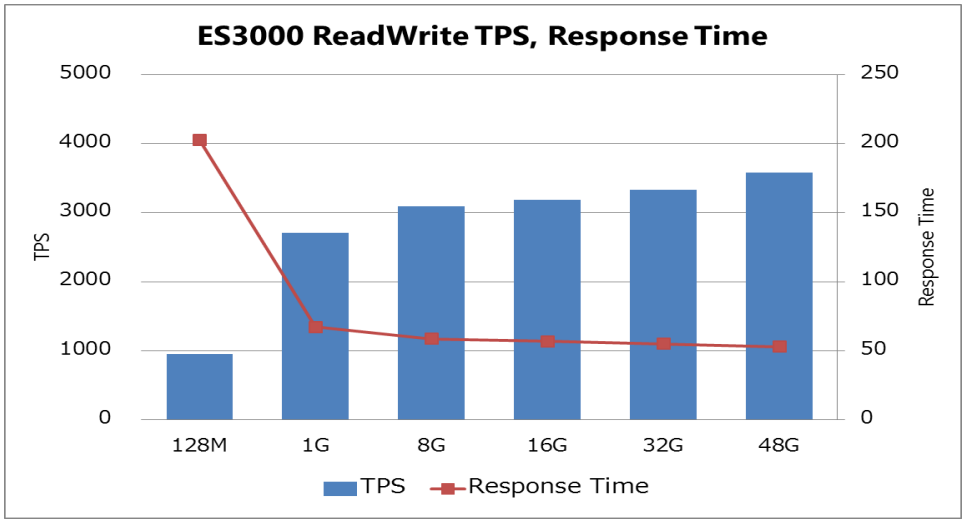


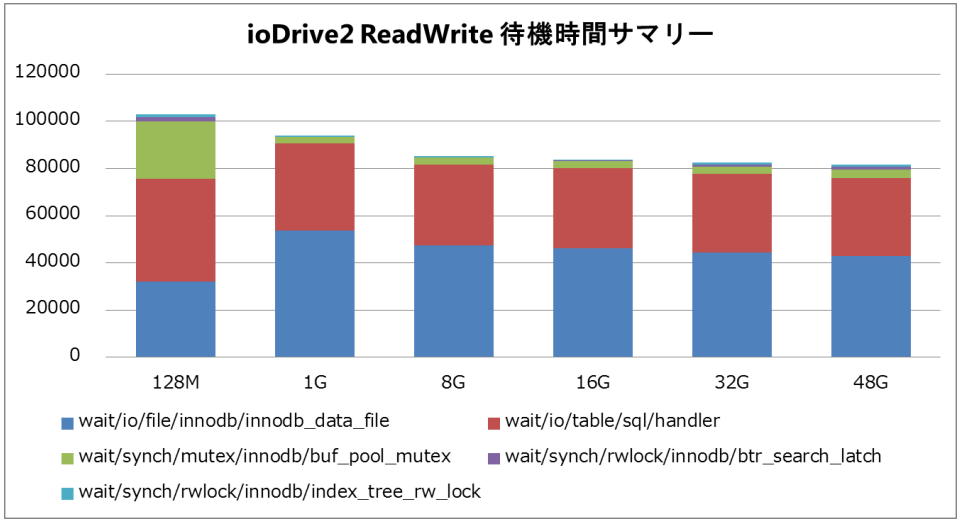


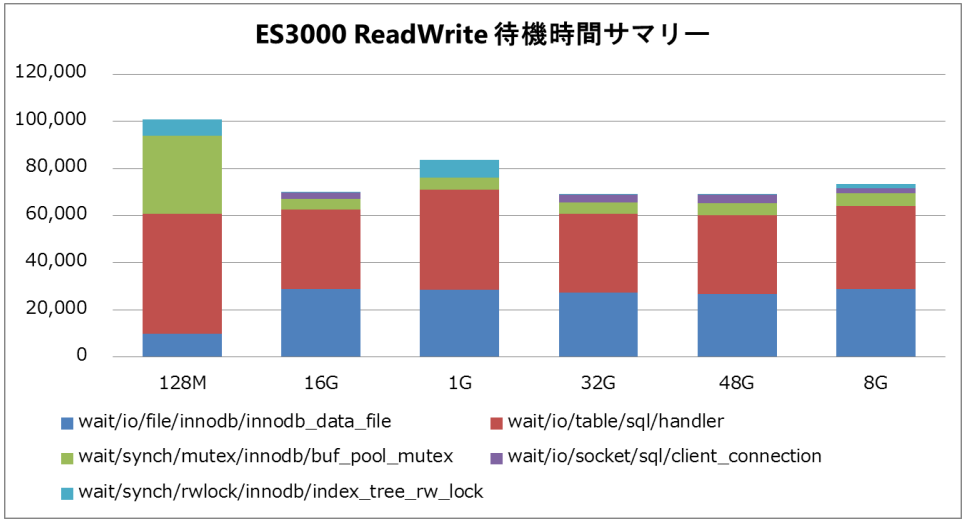


128 Users ReadWriteモード









待機時間の多かったwait events

wait/io/file/innodb/innodb_data_file

- ファイル操作の完了待ち

wait/io/table/sql/handler

- テーブルI/O操作の完了待ち

wait/synch/mutex/innodb/buf_pool_mutex

- 原因：メモリー内のセマフォやオブジェクト競合
- 対策：innodb_buffer_pool_instance を増やす

wait/synch/rwlock/innodb/btr_search_latch

- 原因：メモリー上のHash Indexの競合
- 対策：innodb_adaptive_hash_index をoff

MySQLパラメータ設定（my.cnf）

パラメータ名	デフォルト	チューニング
innodb_buffer_pool_size	128MB	48GB
innodb_flush_method	fsync	O_DIRECT
innodb_flush_neighbors	1	0
innodb_io_capacity	200	100000
innodb_io_capacity_max	2000	400000
innodb_log_file_size	48M	4GB
innodb_page_size	16KB	4KB
innodb_read_ahead_threshold	56	0
innodb_read_io_threads	4	12
innodb_write_io_threads	4	12
innodb_doublewrite	1	1 （stores all data twice）
innodb_flush_log_at_trx_commit	1	1 （full ACID compliance）
innodb_buffer_pool_instances	8	24
innodb_adaptive_hash_index	1	0

ファイルシステムとデータベース属性

ファイルシステム属性	
タイプ	XFS
ブロックサイズ	4K

データベース属性	
全体サイズ	127G
データファイルの最適化	実施済

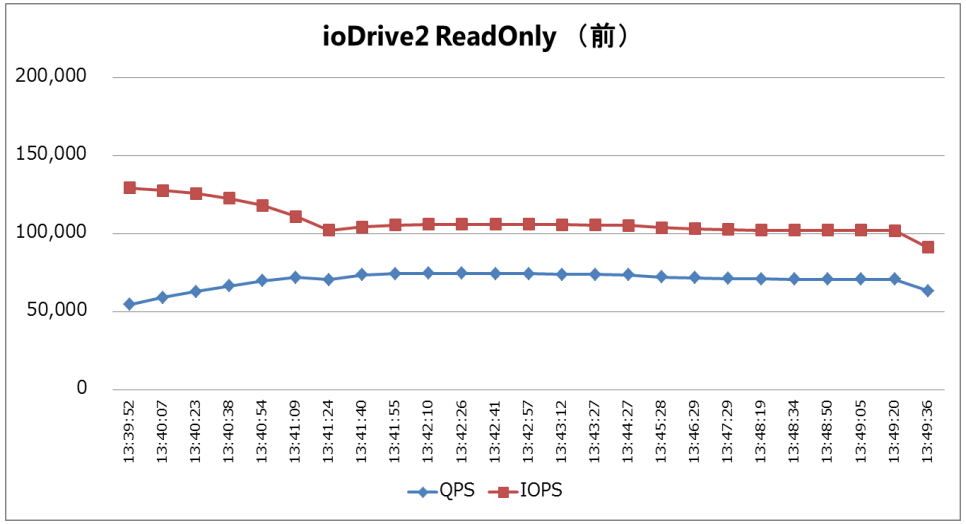
その他	
I/Oスケジューラ	noop

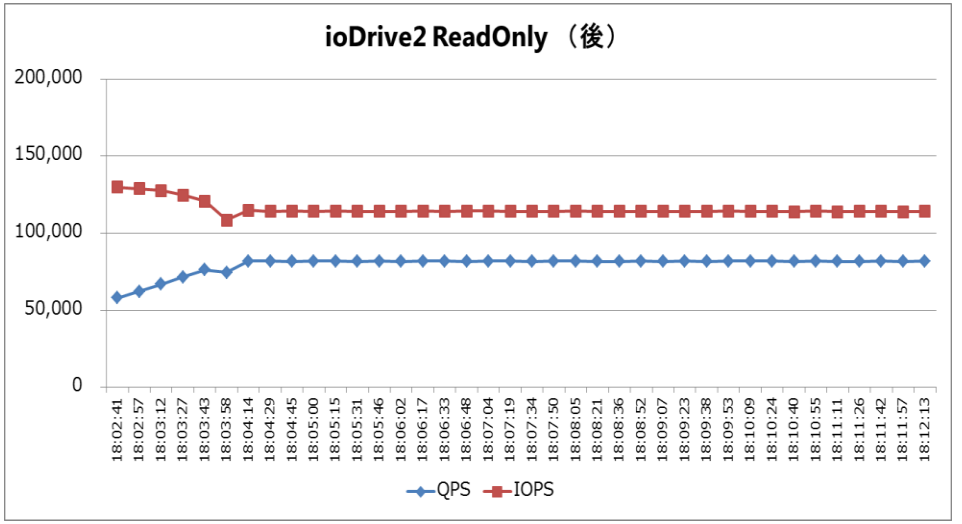
チューニング結果

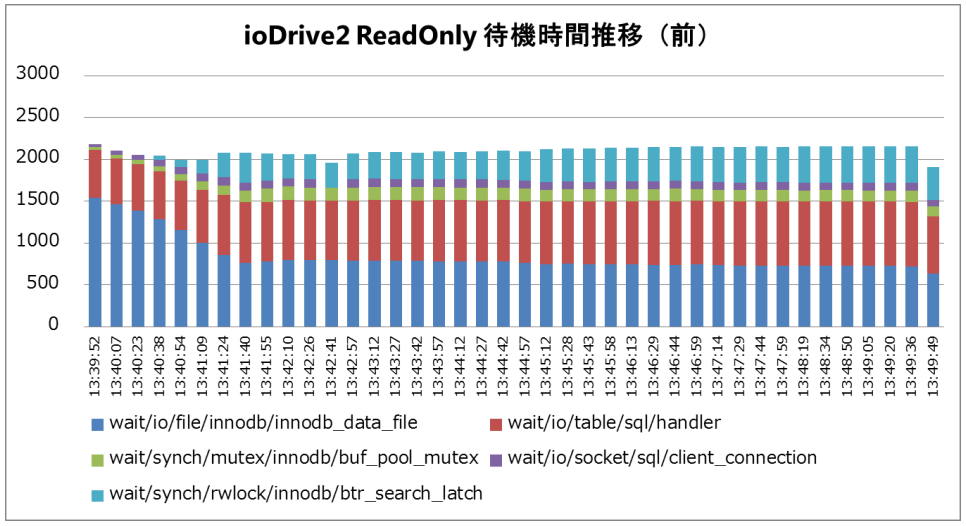
128 Users　innodb_buffer_pool_size = 48GB

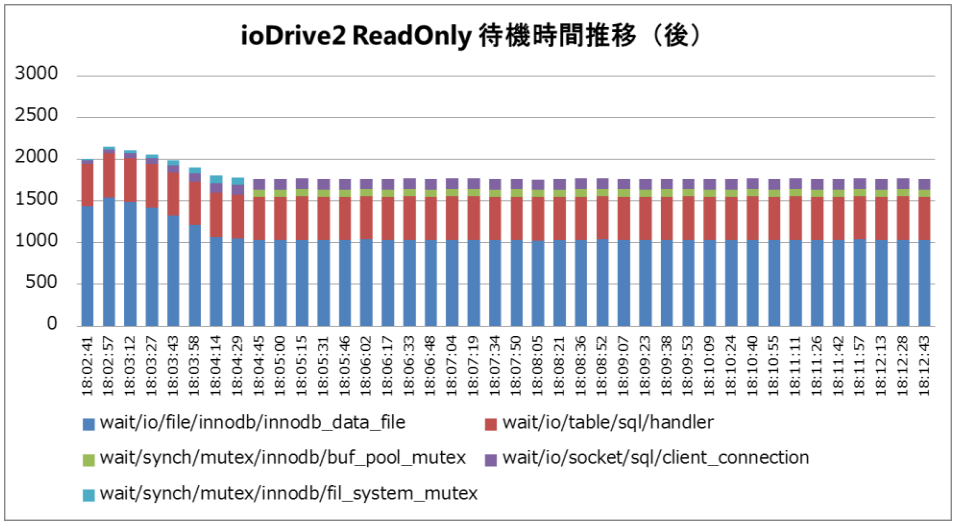
		TPS		Response Time	
		チューニング前	チューニング後	チューニング前	チューニング後
ioDrive2	Read Only	4,437	4,948	39.13	36.29
	Read Write	2,116	2,237	94.02	90.27
ES3000	Read Only	5,479	6,004	38.01	36.47
	Read Write	3,580	3,804	52.73	50.55

ioDrive2 ReadOnly モード

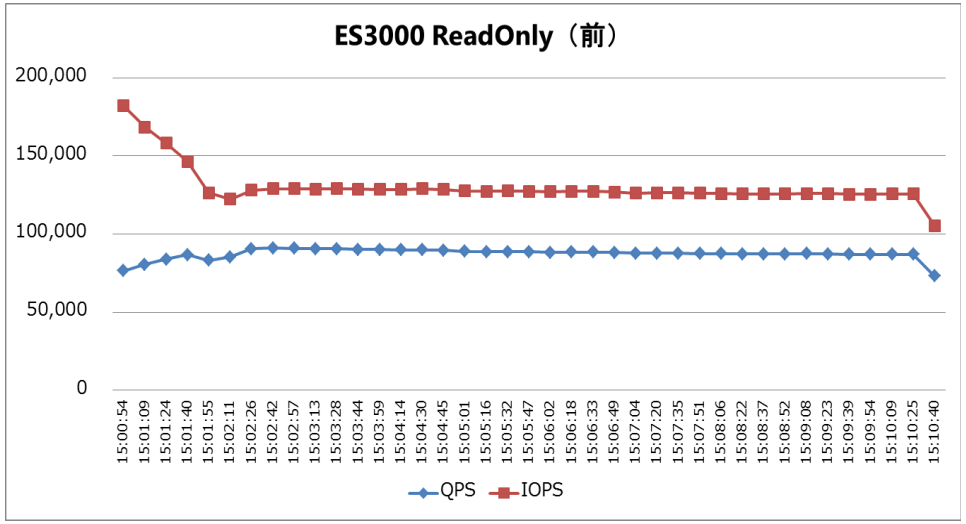


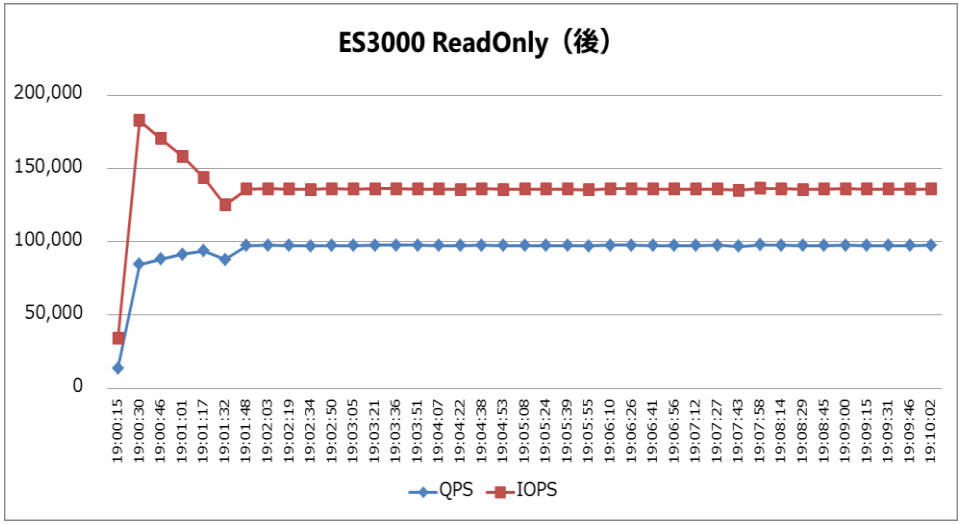


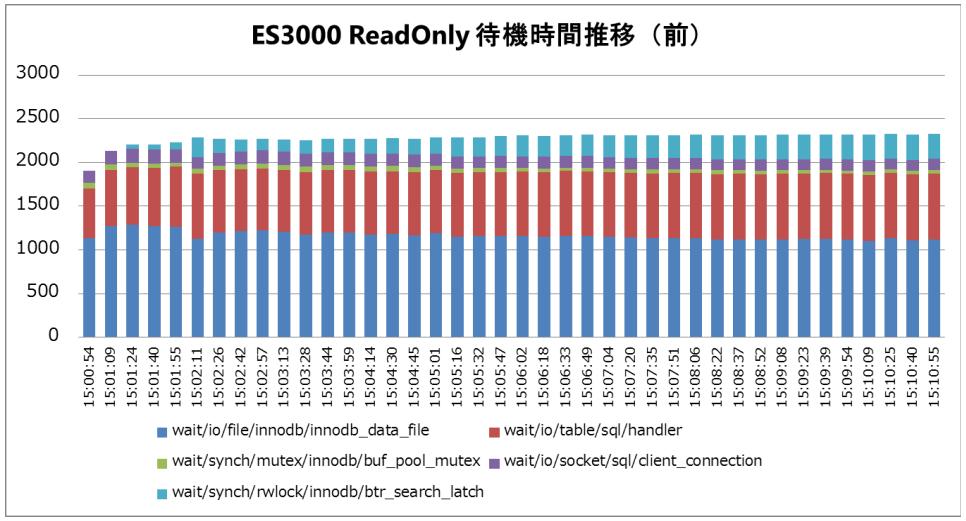


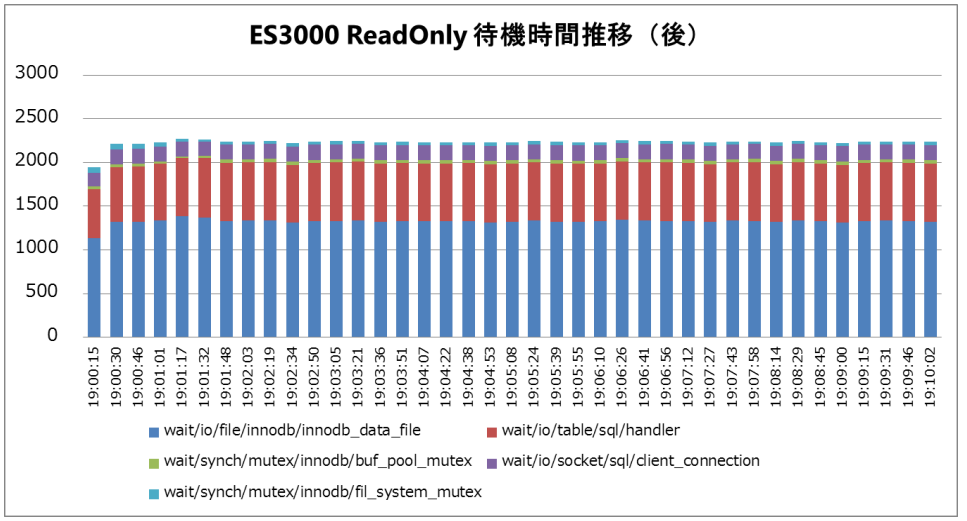


ES3000 ReadOnly モード

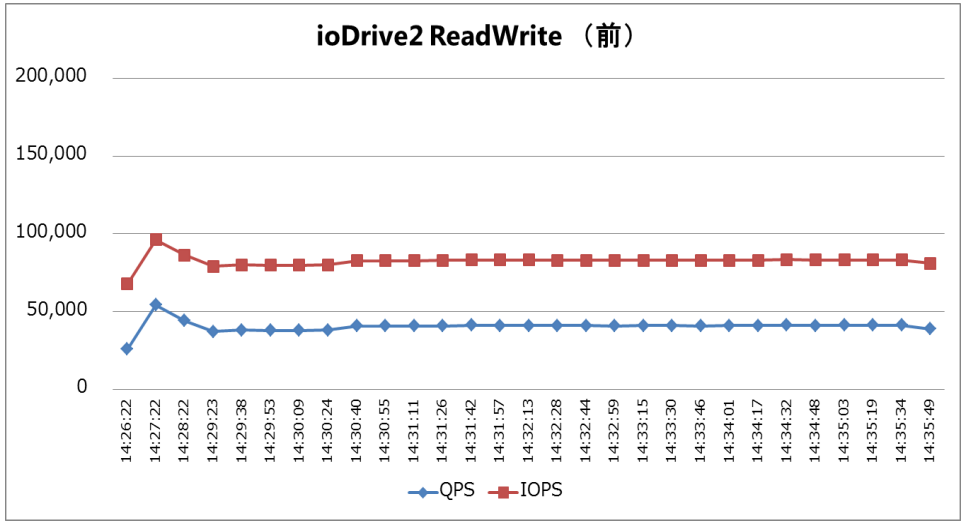


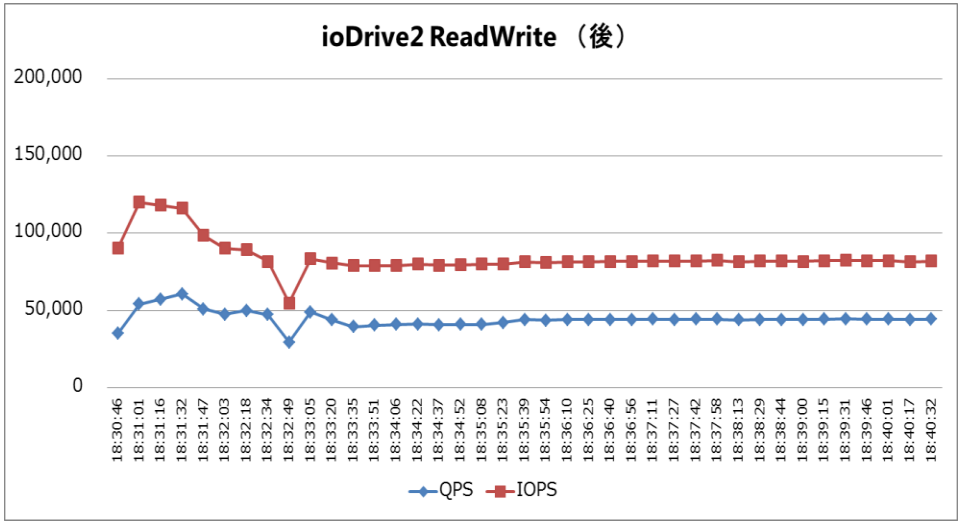


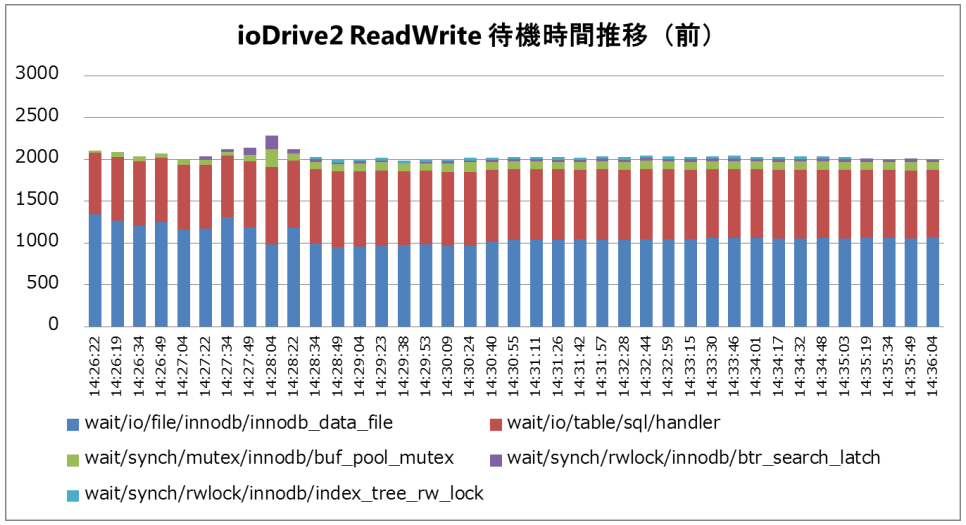


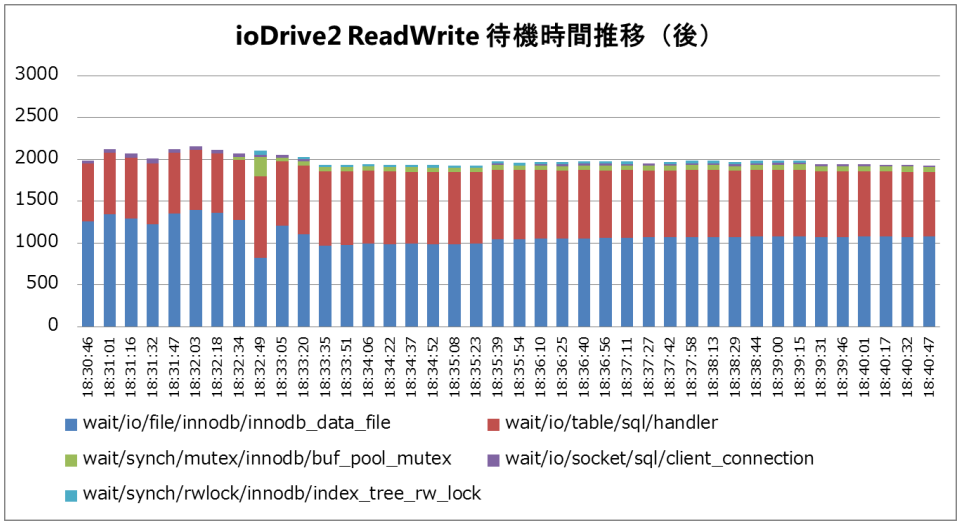


ioDrive2 ReadWrite モード

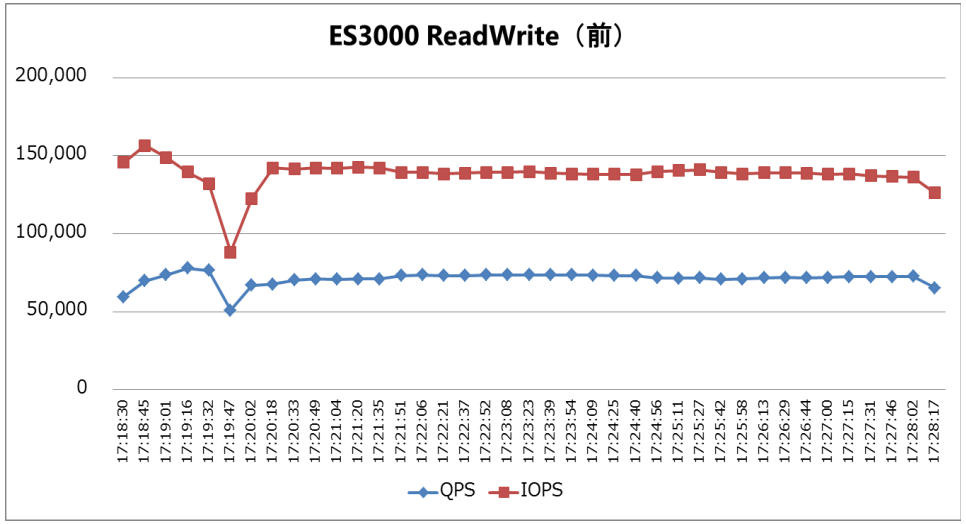


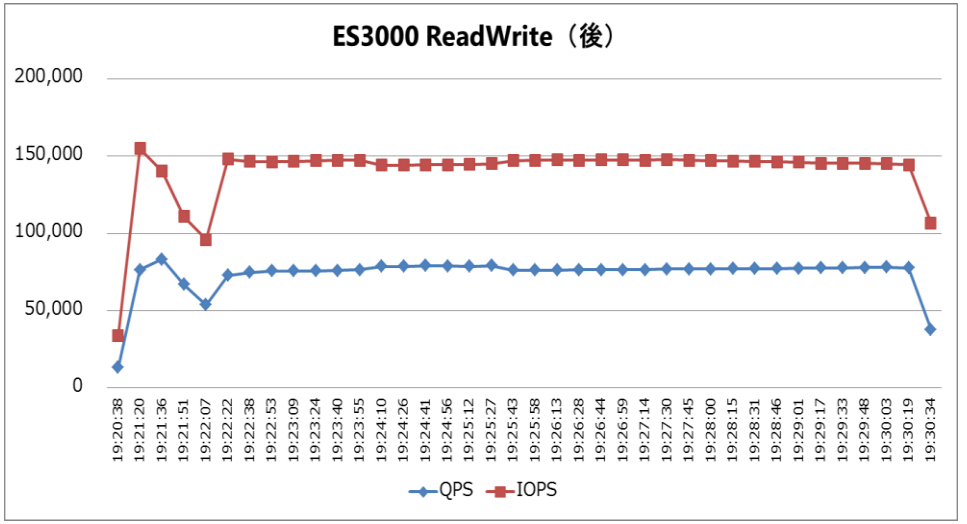


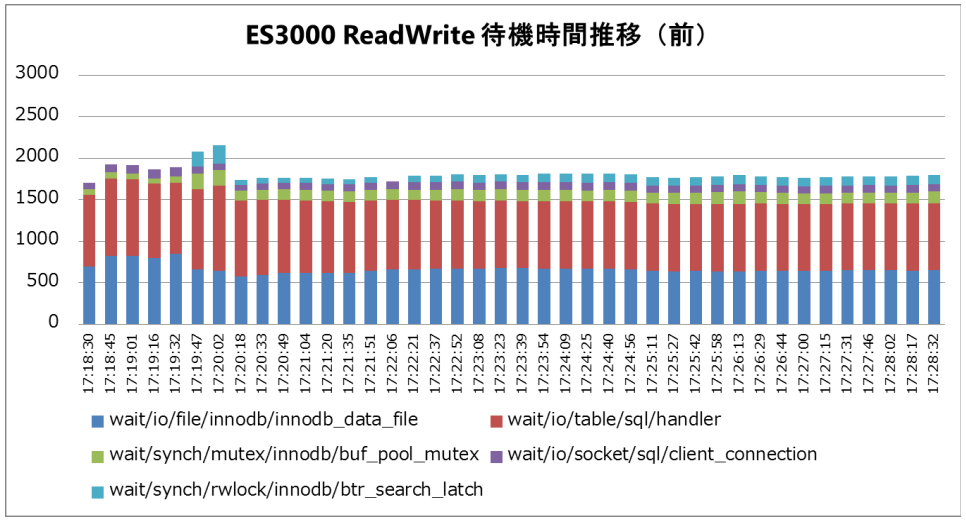


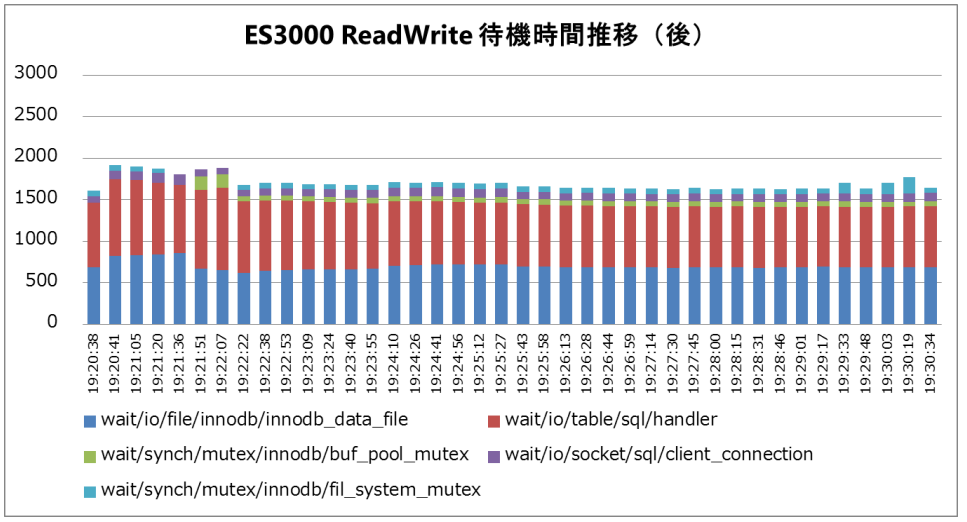


ES3000 ReadWrite モード









チューニング結果

- innodb_buffer_pool_size = 1Gでbuf_pool_mutexが激減、これに比例してTPSが増加、Response Timeが減少
- innodb_buffer_pool_size = 48Gで待機イベント対策により、ReadOnlyモードで511～525TPSアップ、ReadWriteモードで121～224TPSアップ
- innodb_buffer_pool_size = 48Gで待機イベント対策により、ReadOnlyモードの上位5個以内にbtr_search_latchイベントがランキングされなくなった
- 待機時間の減少に比例してQPS、IOPSが向上

チューニング・ポイント

- buf_pool_mutexの回避策としてinnodb_buffer_poolの分割が効果的、innodb_buffer_pool_size = 1Gでinnodb_buffer_poolが自動的に8分割される
- buf_pool_mutexを回避してPCIe SSDの性能を引き出す為にinnodb_buffer_pool_sizeを1G以上にすることが推奨される
- innodb_buffer_pool_sizeが大きくなるとメモリー上のHash Index競合が発生することがある
- Hash Index競合は、 innodb_adaptive_hash_indexをoffにすることで回避できる可能性がある

まとめ

- fioベンチマーク
  - ES3000が4KのSequential ReadとRandom Readで77万IOPSを記録（カタログとおり）
- ORIONベンチマーク
  - Small Random I/O 8KB ReadWriteモードでES3000が25万IOPSを記録
- MySQLベンチマーク
  - ioDrive2は、v5.7における使用を推奨
  - ES3000は、v5.6でもv5.7でも安定した高い性能を発揮
  - 128 Users ReadWriteモードでioDrive2が 9万IOPS、ES3000が17万IOPS (HDDの約800倍)を記録
  - 待機イベントの対策を実施することでパフォーマンス・チューニングの費用対効果を最大化できる可能性がある

キャンペーンのご案内

Huawei ES3000キャンペーン

気持ちいい3つの驚き

1. **驚きの安さ**
   特別価格でご提供
2. **驚きの声**
   CyberAgent様が積極採用！
3. **驚きの速さ**
   本セッションにてご説明！

詳しくは ES3000キャンペーン 検索

驚きの安さ－特別価格でご提供－

800GBモデル

通常提供価格（税別）
768,000円

特別価格（税別）
460,000円

1.2TBモデル

通常提供価格（税別）
1,150,000円

特別価格（税別）
690,000円

2.4TBモデル

通常提供価格（税別）
2,100,000円

特別価格（税別）
1,260,000円

先出センドバック3年ワランティを含む価格です。

驚きの声 – CyberAgent様が積極採用 –

サービス実装されている
『ボーイフレンド（仮）』

PCIe SSDの用途

DBのパフォーマンスを向上するため PCIe SSDの導入を進めています。サーバーに挿すだけで、DBシステムに大きな変更をしないで導入できるのがメリットです。

「Huawei ES3000」導入経緯

コストパフォーマンスの高さに驚きました。導入済みのPCIe SSDと比べ、スペックやパフォーマンスに大差ないのに、導入コストが圧倒的に低いため採用しました。

現在は「ボーイフレンド（仮）」というサービスのカスタマサポート調査用DBと利用者向けマイページシステムのタイムラインDBで利用中です。

"Your Best Partner"

NISSHO ELECTRONICS