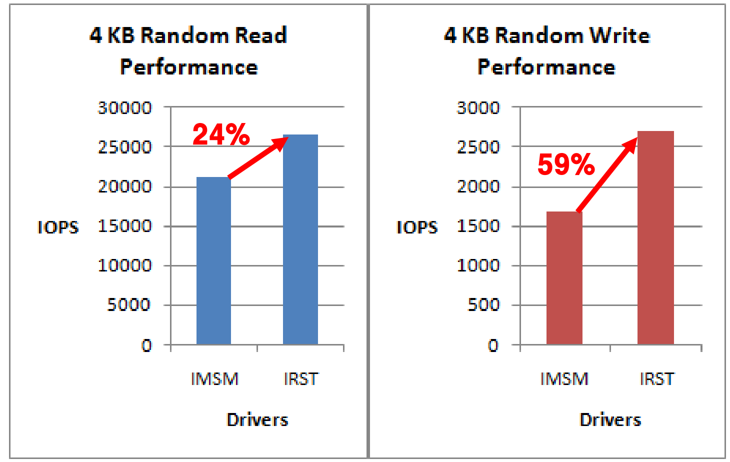
System-wide Issues for Efficient Use of E-SSD

김경호
Samsung Electronics

HBA Performance Improvement
By Chipset Driver Update

HBA – Case1 : Delay Reduction by Driver Update

- **Background**
  - The delay in the HBA affects the SSD performance.
  - The delay can be reduced just by the proper chip set driver.
- **Test Environment**
  - Intel core i7 920
  - Intel X58 Chipset
  - Windows 7
  - SSD
  - Driver
    - IMSM 8.9 vs. IRST 9.0



Store #	Timestamp	Frame Type
3,322	30.856 us	FIS 27 - Cmd: 0x60=RD FPDMA QUEUED
3,354	3.088 us	FIS 34 - Status: 0x40 - DRDY
3,459	27.944 us	FIS 41 - DMA Setup - A: 0 I: 0 D: 1
3,485	908 ns	FIS 46 - Payload Data
3,487	[illegible]	FIS A1 - Set Device Bit - I: 1 Err: 0x00
3,594	32.032 us	FIS 27 - Cmd: 0x60=RD FPDMA QUEUED
3,626	[illegible]	FIS 34 - Status: 0x40 - DRDY
3,745	35.288 us	FIS 41 - DMA Setup - A: 0 I: 0 D: 1
3,774	892 ns	FIS 46 - Payload Data
3,776	16.672 us	FIS A1 - Set Device Bit - I: 1 Err: [illegible]
3,884	31.032 us	FIS 27 - Cmd: 0x60=RD FPDMA QUEUE
3,912	3.056 us	FIS 34 - Status: 0x40 - DRDY
4,011	27.080 us	FIS 41 - DMA Setup - A: 0 I: 0 D: 1
4,041	904 ns	FIS 46 - Payload Data
4,043	16.684 us	FIS A1 - Set Device Bit - I: 1 Err: 0x00
4,148	31.596 us	FIS 27 - Cmd: 0x60=RD FPDMA QUEUE

IMSM

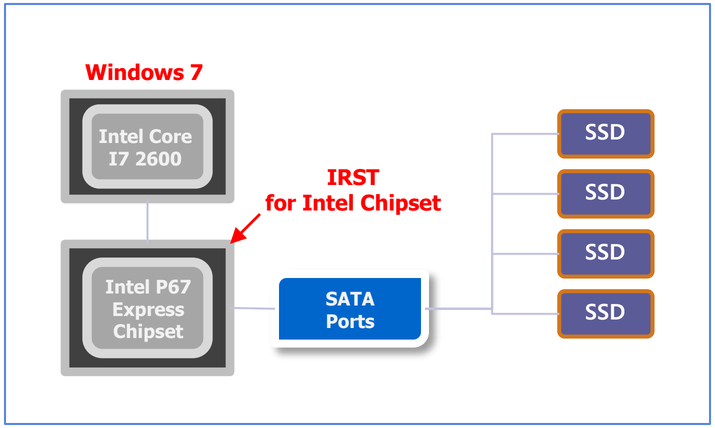
~10 ms reduction between Set Device Bit and Next Command

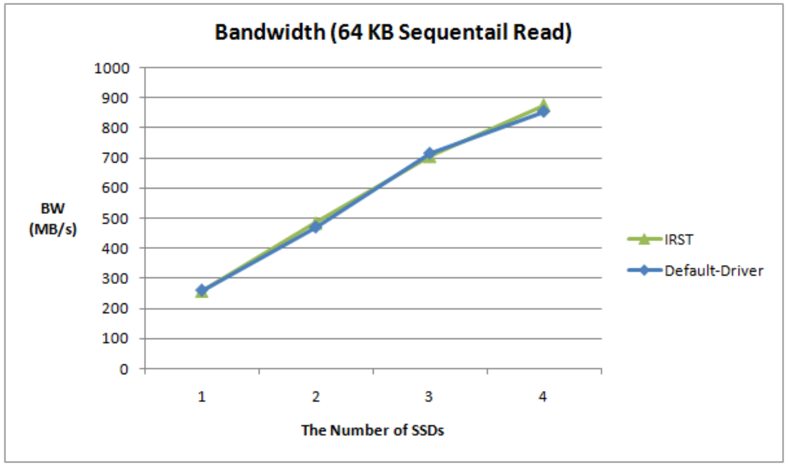
Store #	Timestamp	Frame Type
84	23.540 us	FIS 27 - Cmd: 0x60=RD FPDMA QUEUED
116	3.072 us	FIS 34 - Status: 0x40 - DRDY
217	27.052 us	FIS 41 - DMA Setup - A: 0 I: 0 D: 1
243	920 ns	FIS 46 - Payload Data
245	[illegible]	FIS A1 - Set Device Bit - I: 1 Err: 0x00
334	24.104 us	FIS 27 - Cmd: 0x60=RD FPDMA QUEUED
366	[illegible]	FIS 34 - Status: 0x40 - DRDY
462	27.172 us	FIS 41 - DMA Setup - A: 0 I: 0 D: 1
489	920 ns	FIS 46 - Payload Data
491	16.656 us	FIS A1 - Set Device Bit - I: 1 Err: 0x00
575	23.520 us	FIS 27 - Cmd: 0x60=RD FPDMA QUEUED
605	3.092 us	FIS 34 - Status: 0x40 - DRDY
703	25.896 us	FIS 41 - DMA Setup - A: 0 I: 0 D: 1
729	904 ns	FIS 46 - Payload Data
731	16.644 us	FIS A1 - Set Device Bit - I: 1 Err: 0x00
820	24.548 us	FIS 27 - Cmd: 0x60=RD FPDMA QUEUED
852	3.092 us	FIS 34 - Status: 0x40 - DRDY

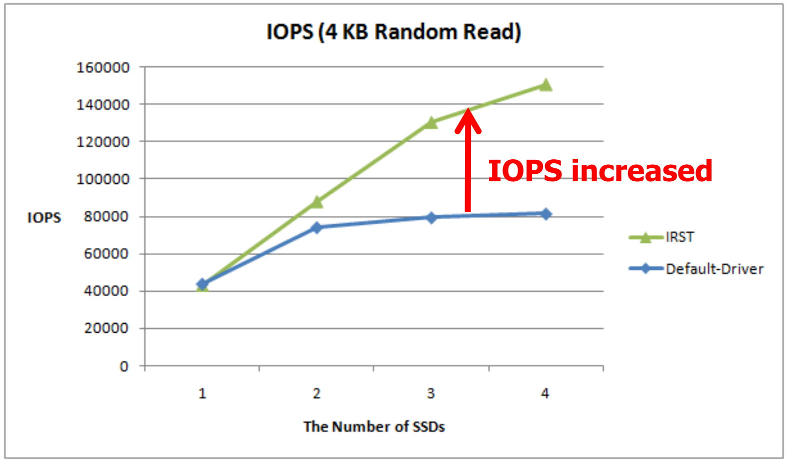
IRST

HBA – Case2 : IOPS enhancement by Driver Update

- **Background**
  - System can get the full IOPS by Driver update.
- **Test Environment**
  - Intel core i7 2600 @ 3.7 GHz (Quad-Core)
  - Intel P67 Express Chipset
  - Windows 7
  - Driver
    - Windows 7 Default Driver vs. IRST 9.0





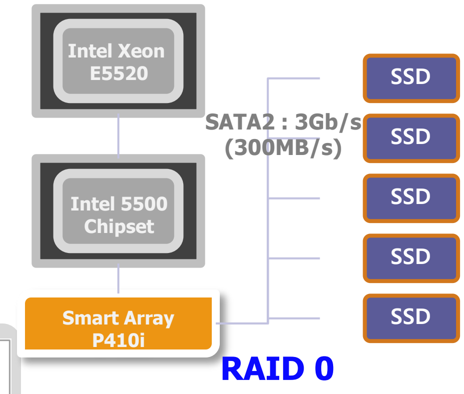


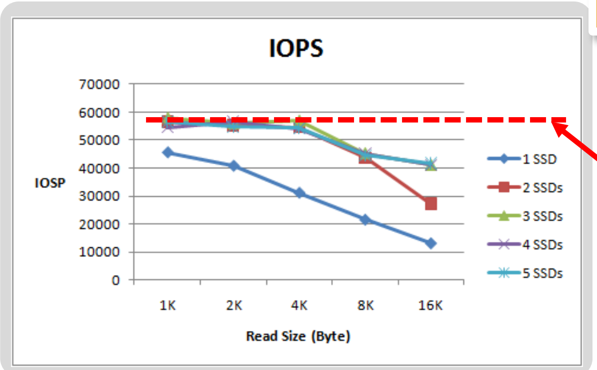
For large requests (64 KB), the bandwidth scales up with SSDs.
For small requests (4 KB), the IOPS is saturated at 80K for Windows 7 default driver.
Just driver upgrade to IRST makes the IOPS scalable for the small requests.

Issues in RAID

IOPS Saturation in Server RAID – HP-ML370 (1)

- **Background**
  - The performance of RAID systems seems to be saturated by IOPS.
- **Experiment Environment**
  - HP-ML370 Server System
  - RAID Controller : Smart Array P410i
  - RAID 0 Configuration
  - IOMeter

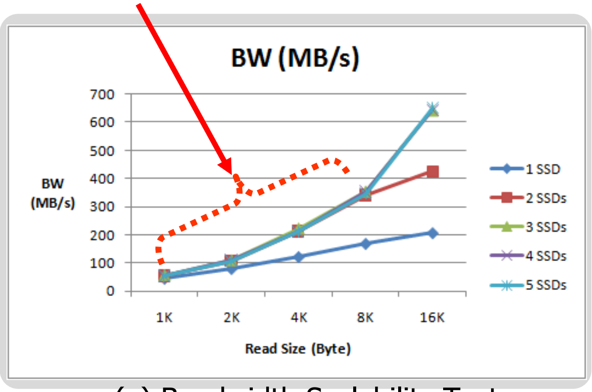




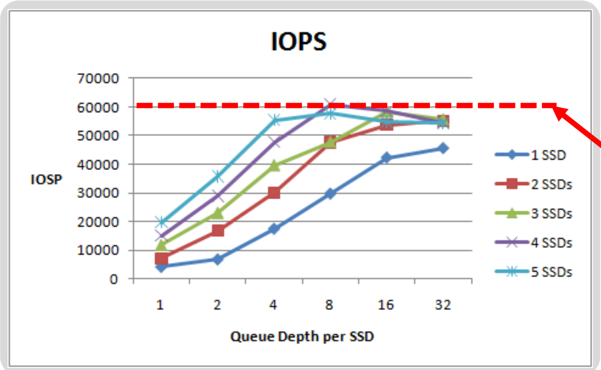
(B) IOPS Test – Queue Depth(32), Various sizes

IOPS saturation at ~60K

BW is saturated at 2 SSDs in small size.



(a) Bandwidth Scalability Test

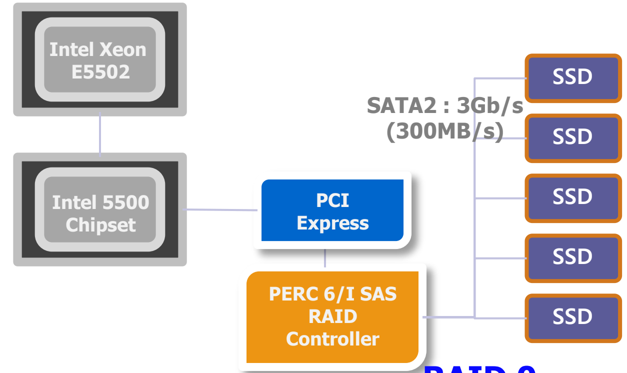


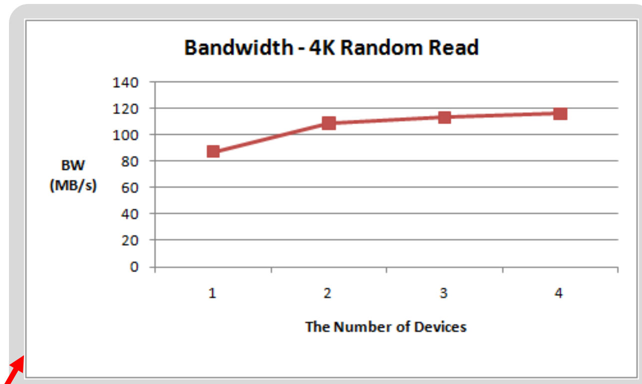
IOPS saturation at ~60K

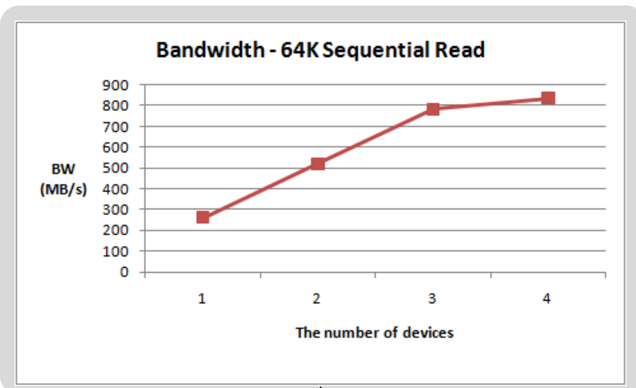
(C) IOPS Test – 1KB, Variable Queue Depth

IOPS Saturation in Server RAID – Dell-T410 (2)

- Background
  - The performance of RAID systems seems to be saturated by IOPS.
- Experiment Environment
  - Dell-T410 Server System
  - RAID Controller : Dell PERC 6/I Adapter Raid Controller
  - RAID 0 Configuration
  - IOMeter



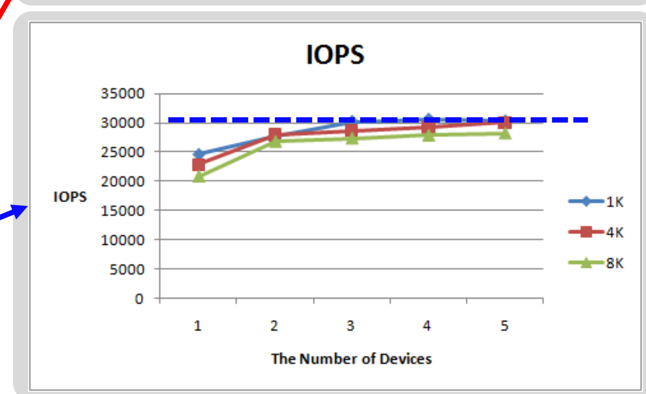




Bandwidth is Scalable for Large Request,

but, not for Small Request.

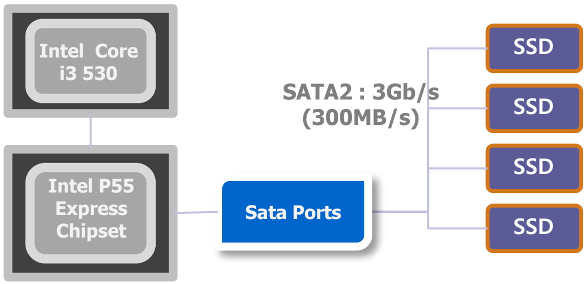
IOPS seem to be saturated at ~30K.

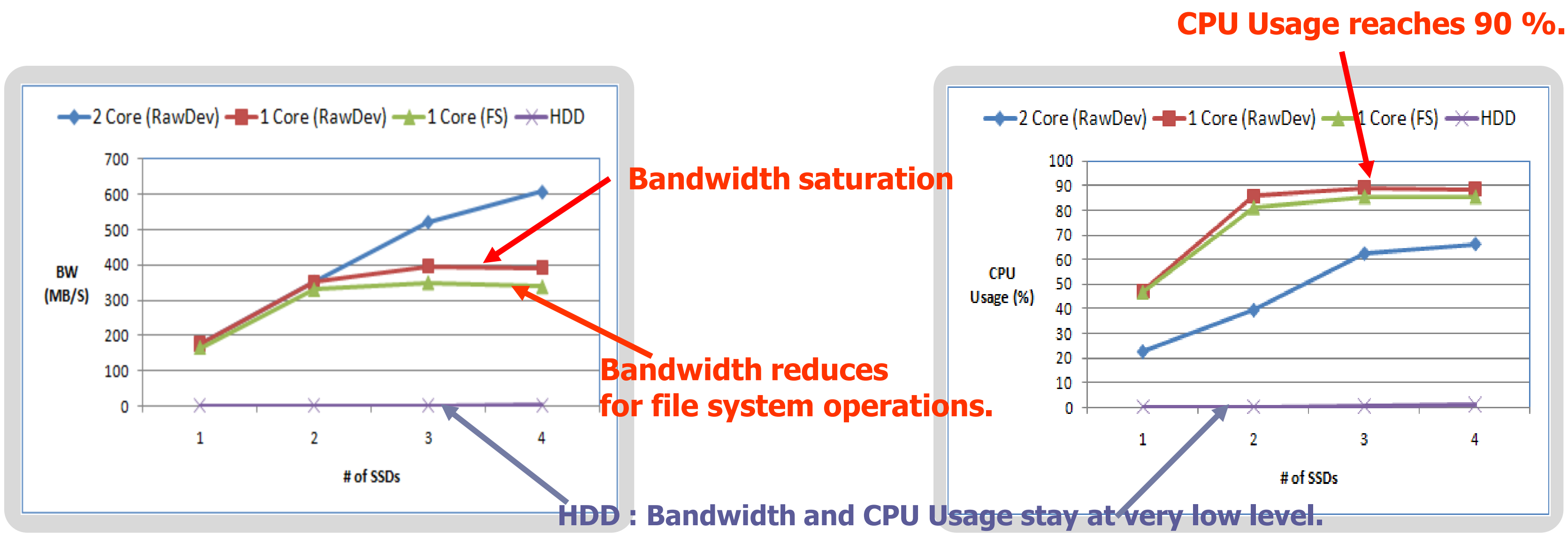


Operating System Optimization

Problem : CPU Usage and SSD Bandwidth

- **Background**
  - Only I/O Treatment consumes the CPU resources.
  - This slide shows the capability of each CPU-Core.
- **Experiment Environment**
  - Intel Core i3 530 @2.97 GHz, [Dual-Core]
  - Windows 7, Intel Driver is installed.
  - IOMeter : 4 KB Random Read



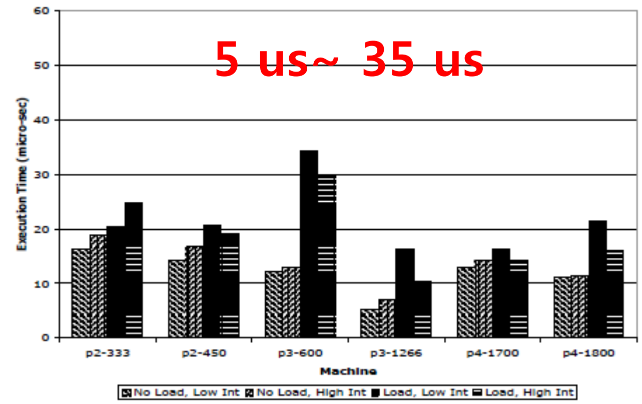


• **When a core is used, the bandwidth is not scaled up with more than 2 SSDs.**

Improvement Point[1] : Interrupt Handling

- **Background**
  - **Disk Interrupt Overhead are about 5 us ~ 35 us[1]**

    [1] Branden Moore Thomas , En Moore , Thomas Slabach , Lambert Schaelicke, "**Profiling Interrupt Handler Performance through Kernel Instrumentation**", Proceedings of the 21 st IEEE International Conference on Computer Design, 2003



(b) Disk Interrupts

  - **Interrupt handling can give burden to CPU for SSD of High IOPS.**
  - **Experiment**
    - Windows 7
    - Measuring Tool : IOMeter
    - SSD (43K IOPS @ 4KB, Random Read, QD=32)
    - Read Latency @ 4KB, QD=1
      - **220 us (SSD latency) + 60 us (Host latency : Intr. Handling + etc)**
  - **IOPS * Interrupt Service Time per IO**
    - Assume that Interrupt handling overhead is **10 us,**
    - Interrupt Handling Overhead per Second is 43K (IOPS) x 10 us = **0.43s.**

- **Idea**
  - **Interrupt handling for group of commands[2]**
  - **Process/Processor-aware interrupt handling[3]**

[2] Salah, K., El-Badawi, K., and Haidari, F., "*Performance Analysis and Comparison of Interrupt-Handling Schemes in **Gigabit Networks***", *International Journal of Computer Communications, Elsevier Science, Vol. 30(17) (2007), pp.* 3425-3441.

[3] Moore Thomas , En Moore , Thomas Slabach , Lambert Schaelicke, "**Process-Aware Interrupt Scheduling and Accounting**", RTSS '06 Proceedings of the 27th IEEE International Real-Time Systems Symposium, 2006

Improvement Point[2] : Kernel Storage Stack

Background

- Kernel Storage stack is designed based on the HDD rather than SSD. The characteristics are changed like this:

HDD

- Extremely Slow Access Time
- Seek Time proportional to LBA distance
- Read/Write Symmetric

→

SSD

- Even Faster than HDD
- Independent to LBA
- Read/Write Asymmetric
- Fast Read, Slow Write with Variation (GC)

Ex) Read : 0.28 ms
Write (QD=1) : 0.1 ms
Write (QD=32) : 1 ms

Ex1) SLC R/P/E : 25 us/200 us/1.5 ms
MLC R/P/E : 60 us/800 us/ 2.5 ms

Storage Kernel Stack Improvement Part

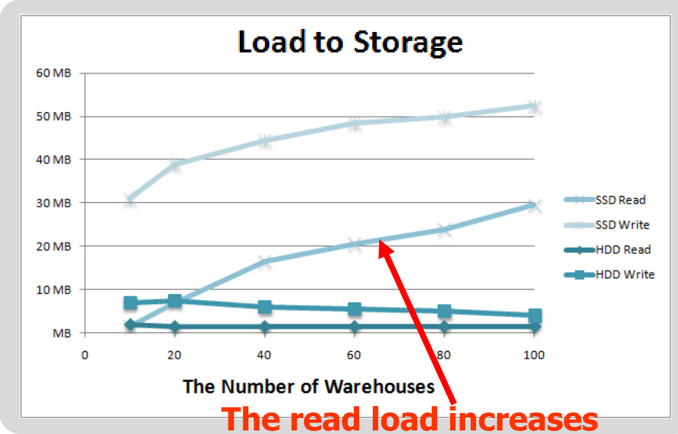
- **Disk Scheduler** * J Kim, Y Oh, E Kim, J Choi, D Lee, "*Disk Scheduler for Solid State Drives*", Proceedings of the seventh ACM international conference on Embedded software, 2009
- **Buffer Replacement** * S Park, D Jung, J Kang, J Kim, "*CFLRU: A Replacement Algorithm for Flash Memory*", Proceedings of the 2006 international conference on Compilers, architecture and synthesis for embedded systems
- **Lightening Block Device Driver Layer** * Matthew T. O'keefe , David J. Lilja , " *High performance solid state storage under linux*" in Proceedings of the 30th IEEE Symposium on Mass Storage Systems, 2010
- **Prefetching off**
- **Swapping** * Mohit Saxena, Michael M. Swift, "*FlashVM: revisiting the virtual memory hierarchy*", Proceedings of the 12th conference on Hot topics in operating systems, 2009

TPC-C Analysis

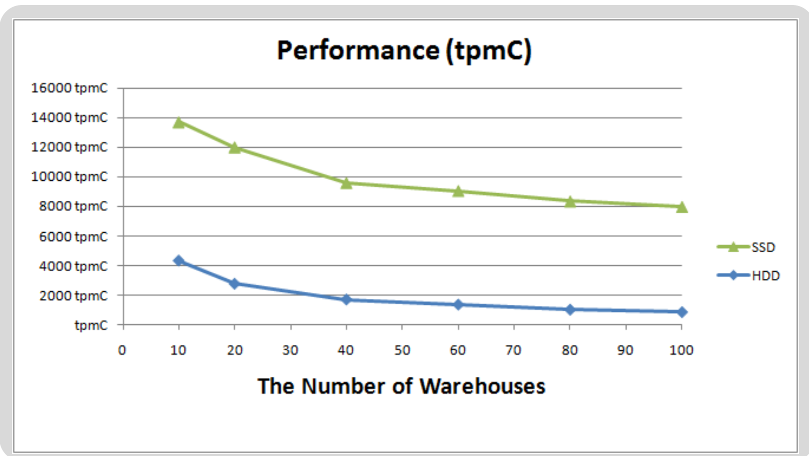
SAMSUNG

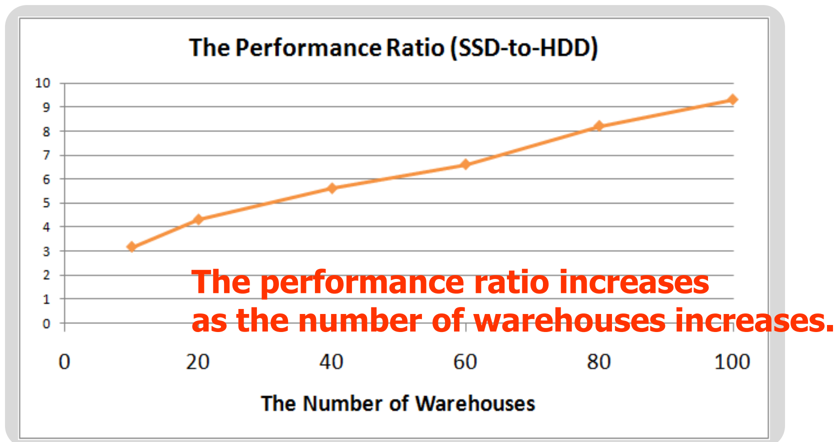
Performance Comparison (SSD vs HDD)

- Background
  - TPC-C data sizes are various.
  - In small data size (Small Warehouses), most read data can be hit by server-side cache.
    In this case, SSD shows performance similar to that of HDD.
- Experiment Environment
  - Client : Benchmark Factory
  - Server
    - DELL T710 (Intel XEON Quad), MySQL, Windows Server 2008
    - SSD, HDD(WD5000AAKS)
    - 10 ~ 100 warehouses (700MB ~ 7 GB)
    - 100 users, no delay
    - 3 GB RAM (Size Fixed)



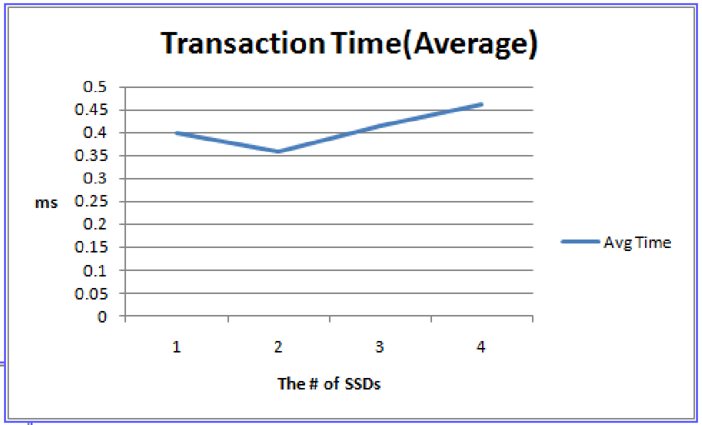
The read load increases
as the number of warehouses increases.

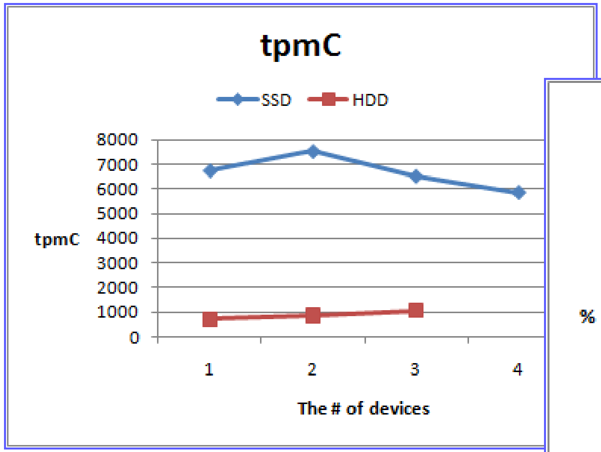


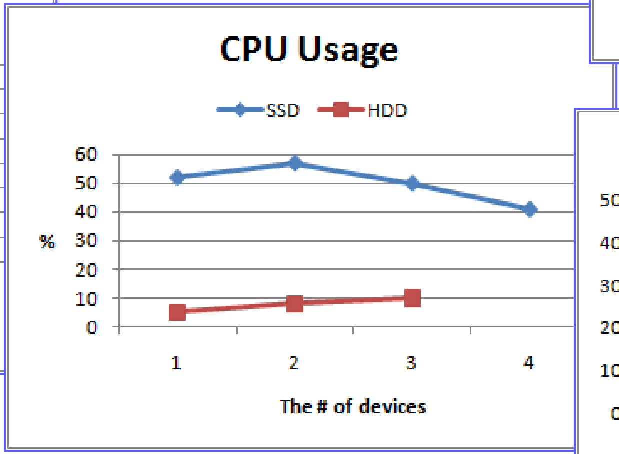


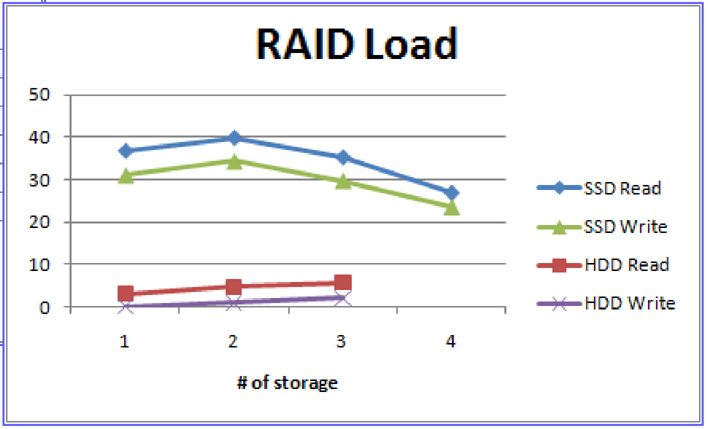
TPC-C Performance in RAID : PC

- Background
  - SSD RAID does not shows the TPC-C performance improvement.
- Text Environment
  - Intel i3 core
    - Windows XP 2008
    - MySQL
    - TPC-C by BM Factory (100 users, 100 warehouse, no delay)
  - RAID0 within Intel Chipset P55 express



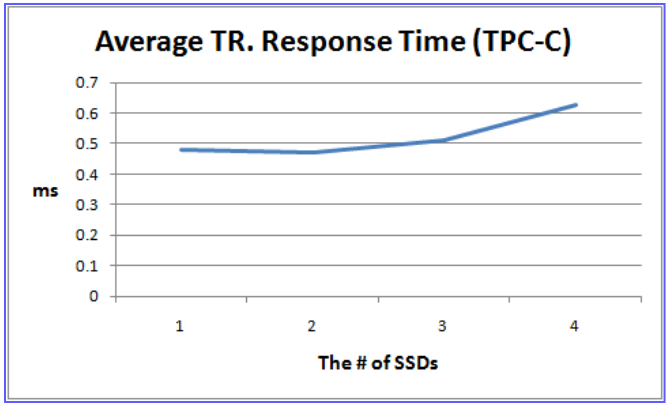


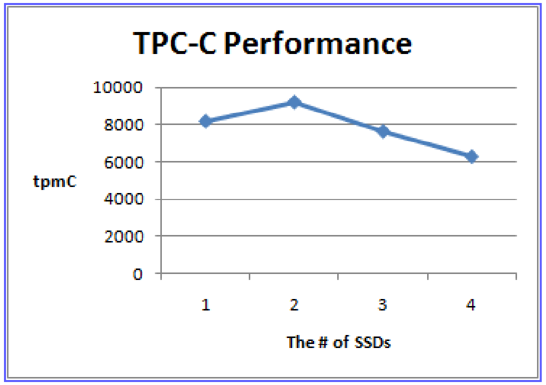


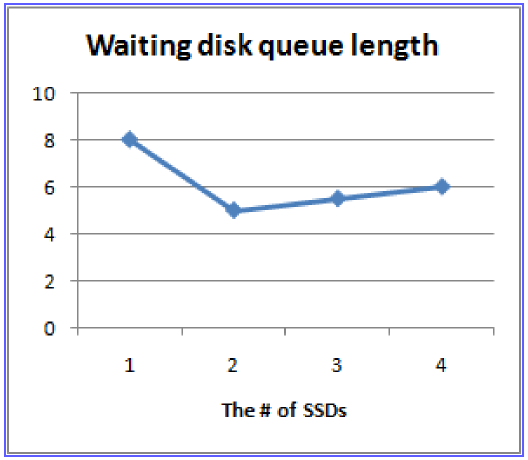


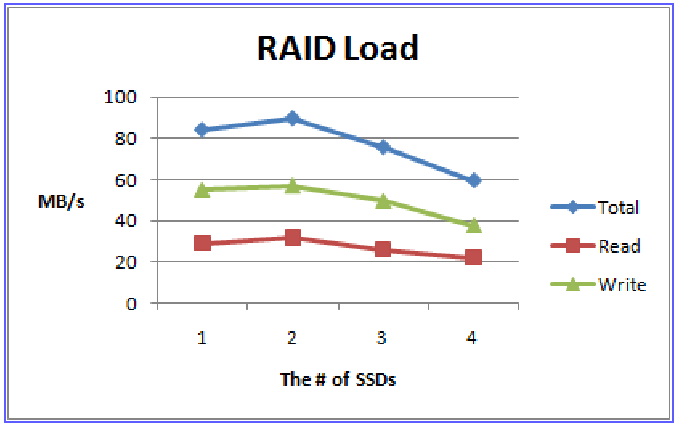
TPC-C Performance in RAID : HPML370 G6

- Background
  - SSD RAID does not shows the TPC-C performance improvement.
- Text Environment
  - HP ML370 G6
    - Intel Xeon Quad Core, Windows Server 2008
    - MySQL
    - TPC-C by BM Factory (100 users, 100 warehouse, no delay)
  - RAID0 within SMART Array P410i



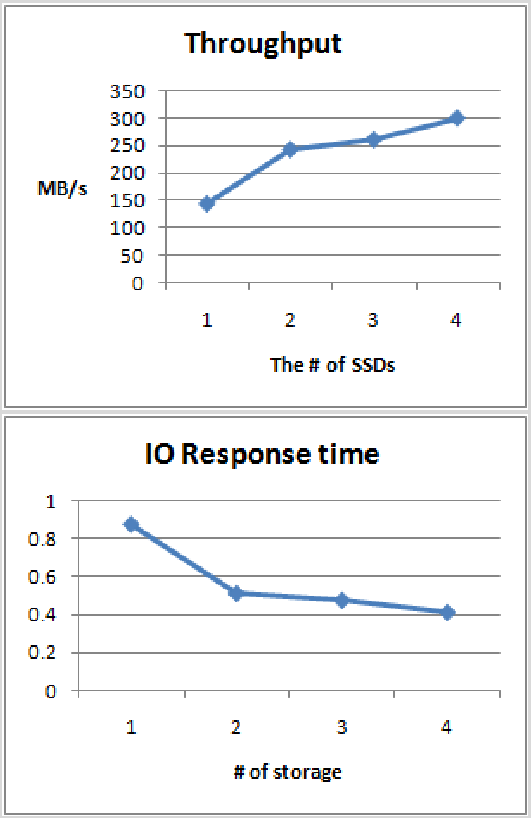




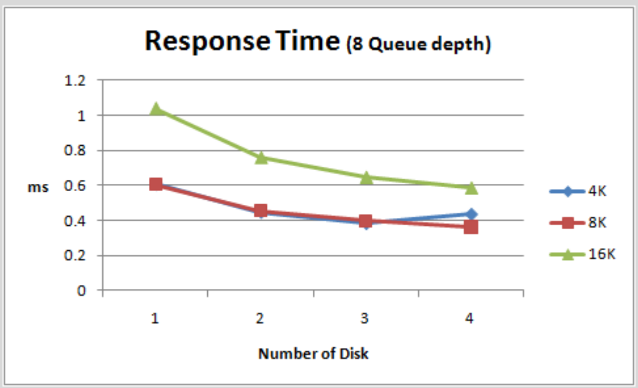


TPC-C Performance in RAID : PC

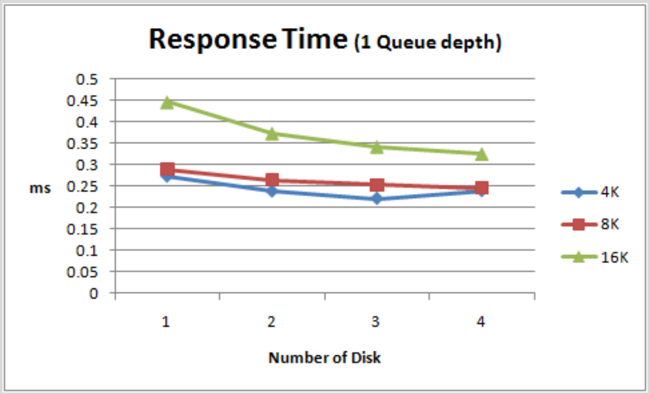
- **Background**
  - SSD RAID does not shows the TPC-C performance improvement.
- **Text Environment**
  - RAID0 within Intel Chipset P55 express – 128 KB Stripe Unit Size
  - IO Meter Test



1 Thread-QD32, 4K RR



1 Thread-QD8, Random Read/Write



1 Thread-QD1, Random Read/Write

2011.4.18