From 512 to 4K

A case study in supporting large sector size SSDs in Solaris

Bo Zhou

Sun Microsystems

Agenda

- Introduction
- Background
- Design & Implementation
- Future Work
- Performance Comparison
- Conclusion

Introduction – Need for Speed

- Two trends of storage devices
  - Craving for Speed
    - Hard disk drives bring down the system performance because of the rotational traits
    - The advent of Solid-State Drive (SSD) represents a sea change in storage system
      - Exceptional bandwidth
      - Excellent random I/O performance
      - Save power and improve reliability
    - The optimal sector size of SSD is typically 4KB

Solid State drive

Introduction – Need for Capacity

- Craving for Capacity
  - more platters, heads and parts introduce more heat, more vibration and more opportunity for errors
  - Large sector size brings benefits
    - High areal of density while maintaining data integrity
    - Reliability
  - Sample 4K HDD drives are available now

Background – Problem Statement

- Solaris can not use SSDs with sector sizes other than 512 bytes.
  - Many modules and drivers hardcode disk sector/block size as 512 bytes.
  - Many applications are sector/block size sensitive
  - Read-Modify-Write (RMW) can apply to disk driver, but sacrifices performance
- The SSD/Flash drives are already in use now
- The performance of SSD can be increased if I/O is sector size aligned

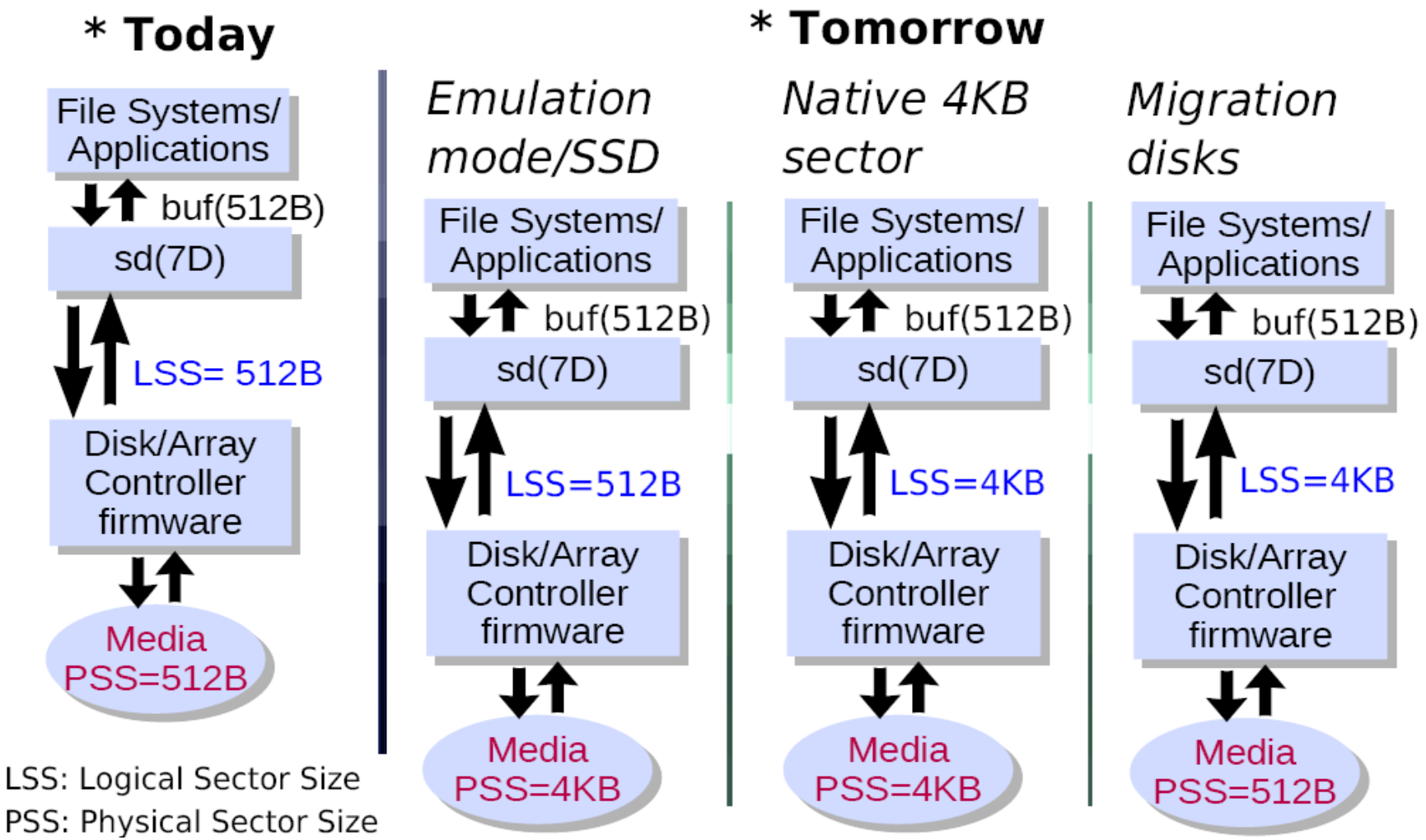
Background – Key Concept

- Physical sector/block
  - The physical unit of storage on the surface of the disk
  - The smallest unit of data which can be physically written to or read from the disk
- Logical sector/block
  - The disk drive presents itself to outside world as a linear address space of logical blocks
  - The size may be in principle different from that of the physical blocks

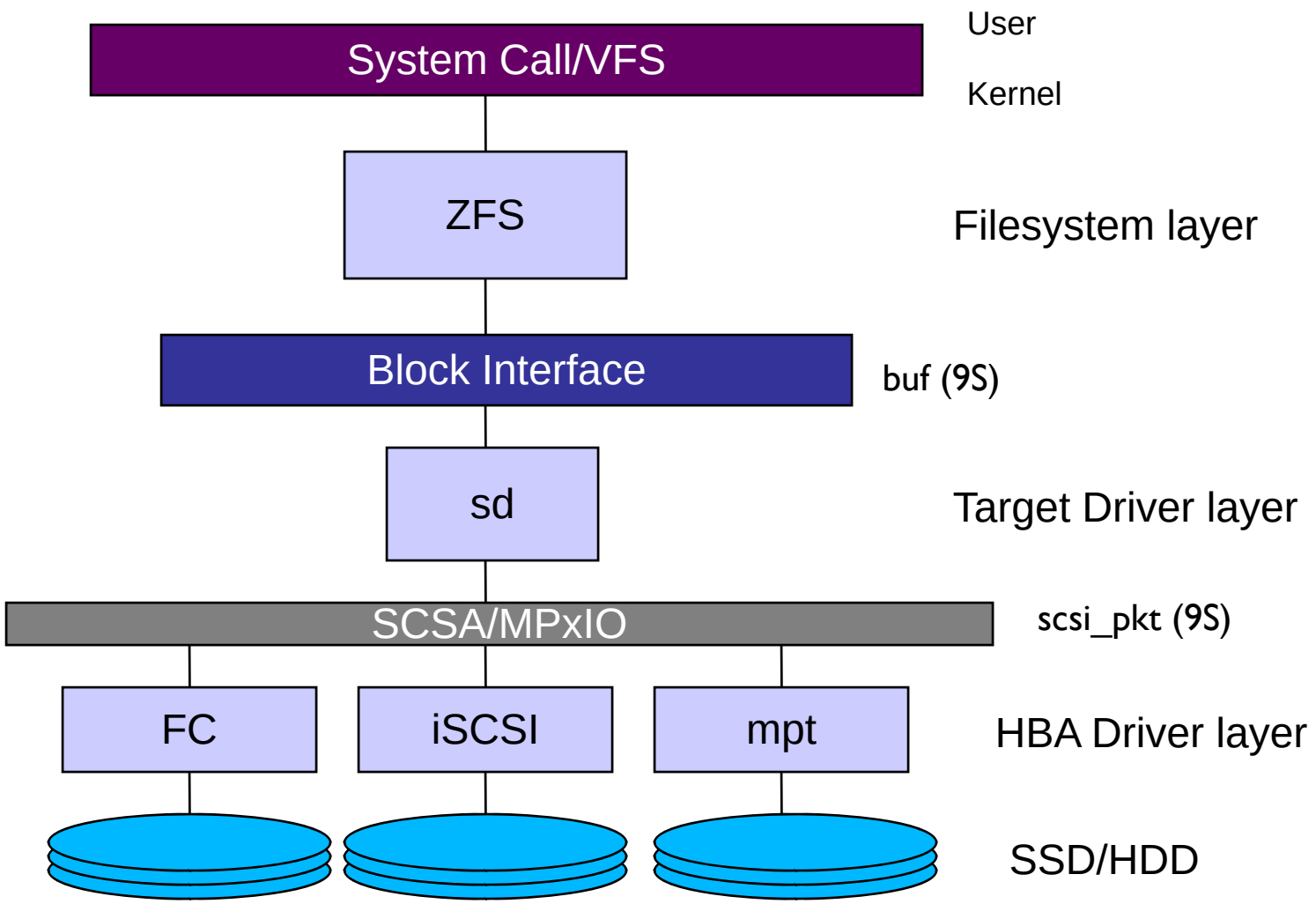
Background – Key Concept (Cont.)

- Emulation Mode
  - The physical sector size is 512 bytes, the disk firmware presents the size as 4K bytes for the upper layers
  - The physical sector size is 4K bytes, the disk firmware presents the size as 512 bytes for the upper layers, such as SSD.
  - Disk drivers can also do the emulation
    - Export large sector size to the upper modules and apps
    - Handle the block size and address tranlation inside driver

Design & Implementation - Overview



Design & Implementation - Overview



Solaris Host Driver Stack

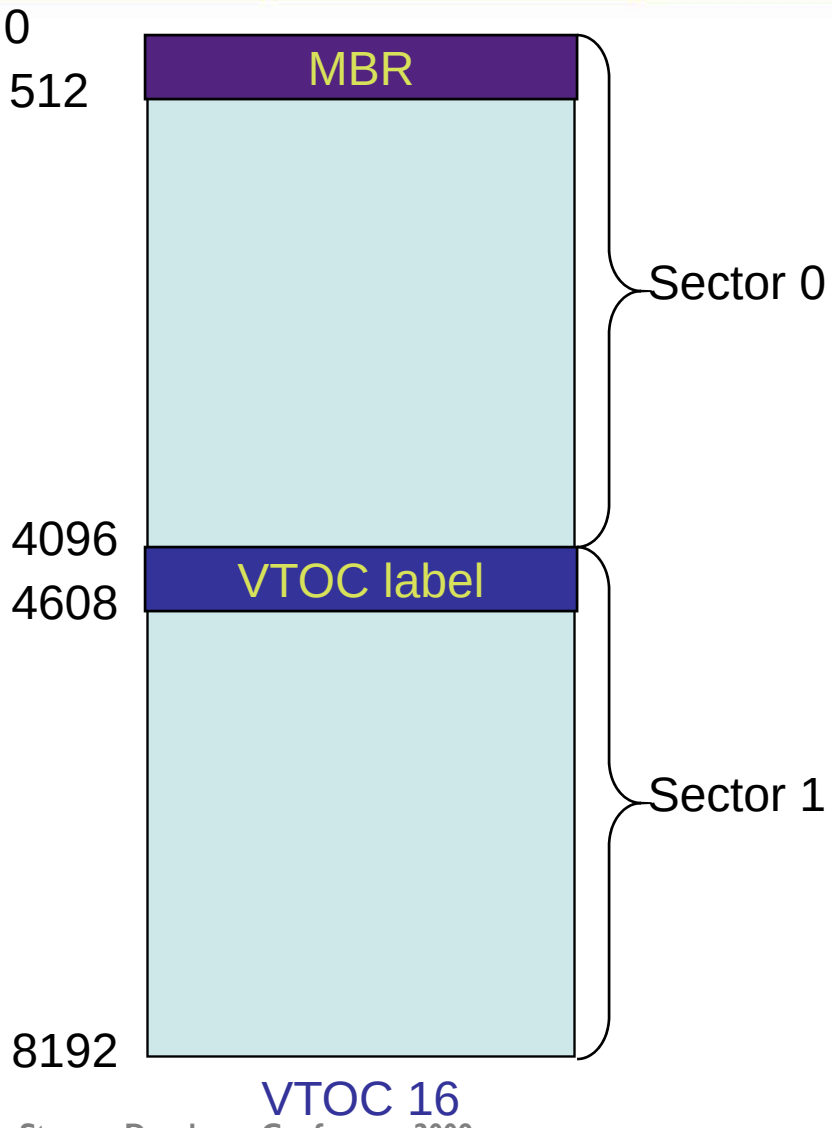
Design & Implementation - Overview

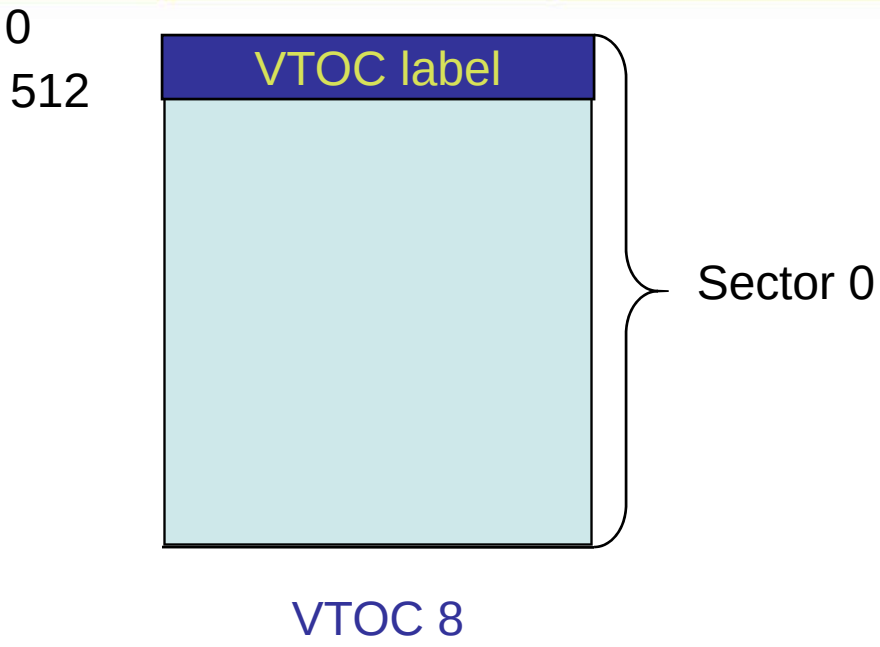
- Buf(9S) is kept intact
  - Backward compatible with most existing modules
  - Logical block size is fixed at 512B/block
  - SCSI disk driver (sd) is responsible for translating between physical sector size and logical block size
- Query disk sector size
  - Applications will not be automatically supported on large sector size disks: for physical sector size/address sensitive applications, use ioctl to query the size

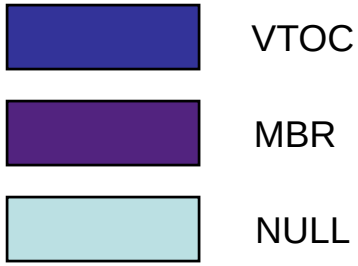
Design & Implementation – Label

- VTOC label
  - Solaris VTOC 16 - x86 platform
    - MBR is at the first 512 bytes of sector 0, sector size can be 512B or larger ones
    - VTOC label is at the first 512 bytes of sector 1 of Solaris partition
  - Solaris VTOC 8 – SPARC platform
    - VTOC label is at the first 512 bytes of sector 0

Design & Implementation - Label



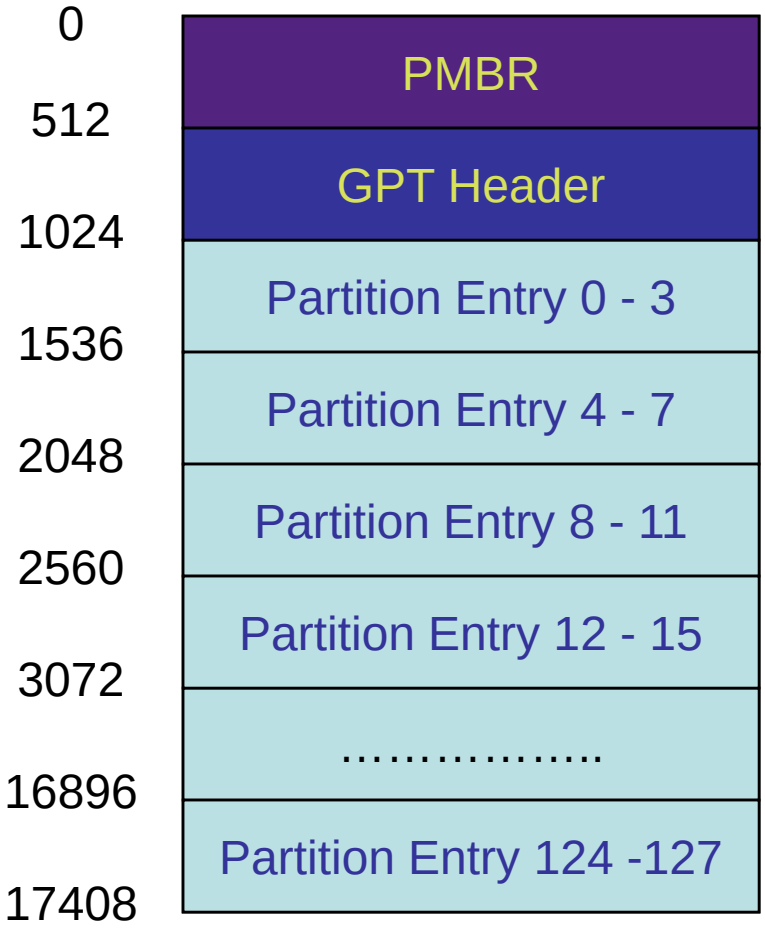




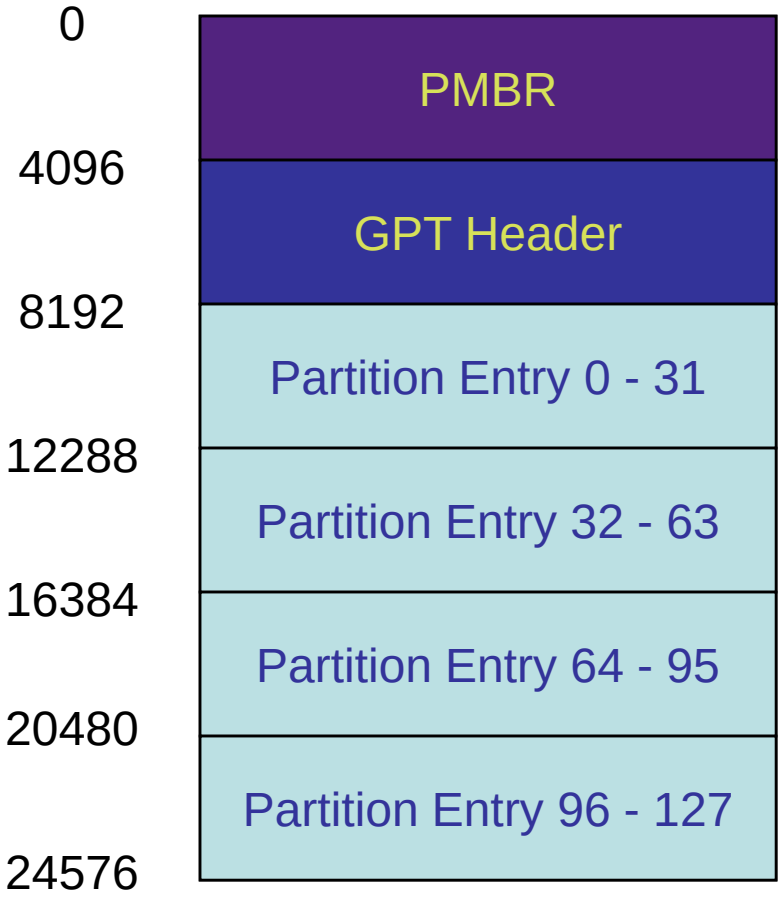
Design & Implementation - Label

- EFI (Extensible Firmware Interface) label
  - PMBR is at the first 512 bytes of sector 0, the remaining bytes are set to 0
  - GPT header is at the first 512 bytes of sector, the remaining bytes are set to 0
  - Partition entries are at the sectors after GPT header

Design & Implementation - Label



512-byte sector size drive



4KB sector size drive

Design & Implementation - Apps

- Disk Utilities
  - Sector/Block size is hardcoded as 512 bytes
  - The EFI label layout changes
  - Fdisk layout changes
  - Partition tools
  - All applications which are sector/block size sensitive needs be to changed by the application developers

Design & Implementation – sd

- SCSI Disk Driver (sd)
  - Sd queries the drive's physical sector size during attach process
  - For non-USCSI I/O request, sd checks whether the size is aligned with physical sector size, if it is, sd sends SCSI READ/WRITE commands to the disk drive
  - If the I/O is misaligned:
    - RMW is enabled for hard disk drive, transfer is allowed, performance penalty
    - RMW is disabled for hard disk drive, transfer is forbidden, error is returned

Design & Implementation - RMW

- RMW applies to large sector size SSDs and HDDs
  - For transition from 512 to 4K
  - Sacrifices the performance to gain backward compatibility
  - Enable/Disable RMW is configurable by end users
  - Misalign message prompts to warn user that the performance penality

Design & Implementation - ZFS

- ZFS
  - ZFS supports large sector size SSDs without any changes
  - Fit various of sector sizes dynamically
  - The I/O sizes are typically larger than the sector/block size
  - ZFS enables hybrid storage pools using high performance SSDs and low cost HDDs

Design & Implementation - Others

- Virtual Machine
  - Xen
    - Can use large sector size SSDs as data disk
    - Host OS provides the drive's sector size to guest OS
  - LDOM
    - Large sector size SSDs could be exported to guest domain as data disk
    - Host domain provides sector size information to guest domain

Future Work - BIOS

- BIOS Firmware
  - BIOS is responsible for detecting, identifying, and configuring the hardware devices and peripherals
  - Two methods to deal with large sector size disk drives
    - Maintain 512 bytes logical sector size backward compatibility by emulating the logical to physical translation within the drive
    - Utilize a logical sector size which is greater than 512 bytes, the same as the physical sector size, this requires the change for BIOS

Future Work - Installation

- OS installation
  - The installer must determine what kind of disk drives it is installing on
  - Current installer hardcode the sector size to 512 bytes
  - Installer needs to adjust any internal buffer sizes to accommodate the larger LBA transfers

Future Work - Applications

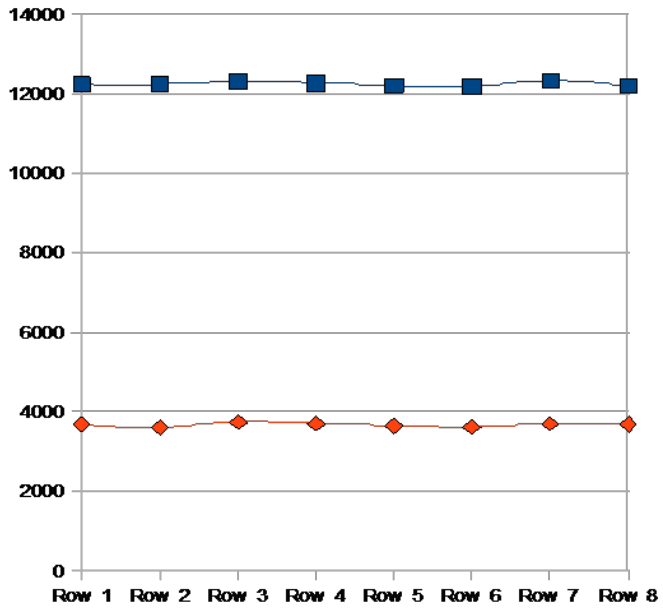
- Applications will not automatically support large sector size SSDs
- It's applications' responsibility to detect, identify, configure and access the installed drive (SSDs or HDDs), using the native sector size
- ISVs should update their applications
- Sector size aligned I/Os from applications accelerate the performance.

Performance Comparison

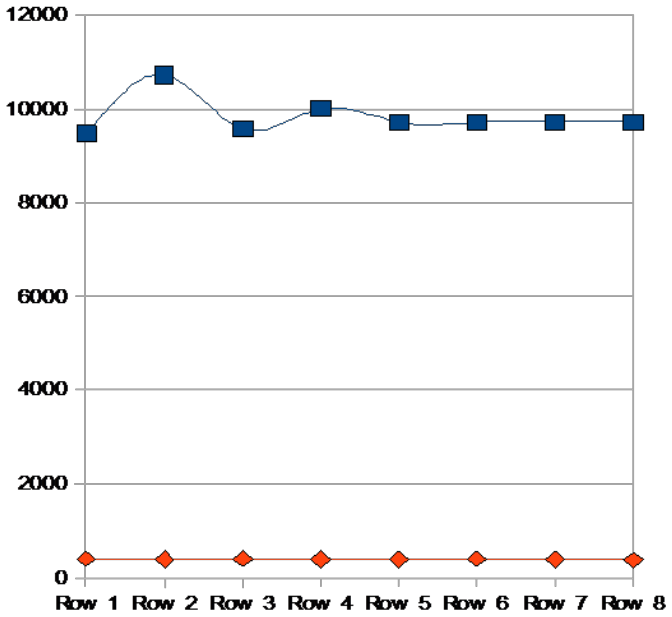
- Impact on performance
  - Partition alignment issue
    - The starting address of partition is not aligned with 4KB
    - The misaligned I/Os introduces serious performance downgrade
    - Can prove 4K aligned I/Os are better than the misaligned ones
  - Other impacts from both software and hardware

Performance Comparison

- Comparison results from SSD/Flash drive performance team



READ IOPS



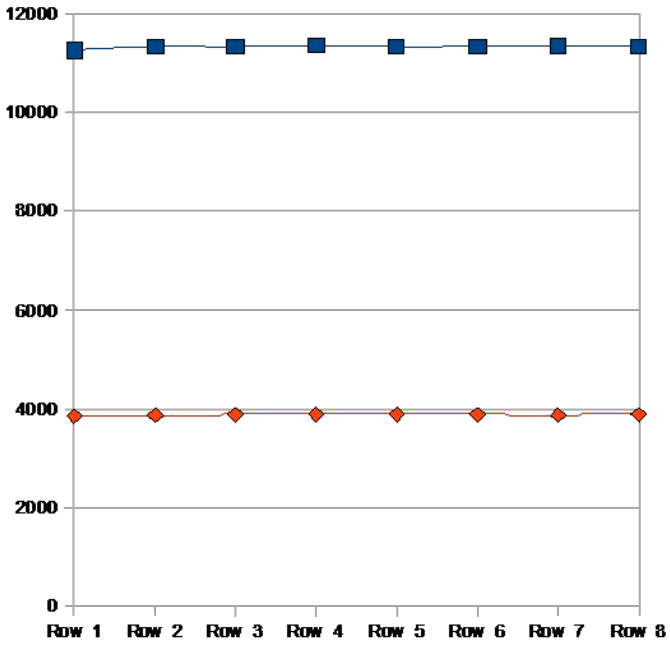
WRITE IOPS

4K misaligned

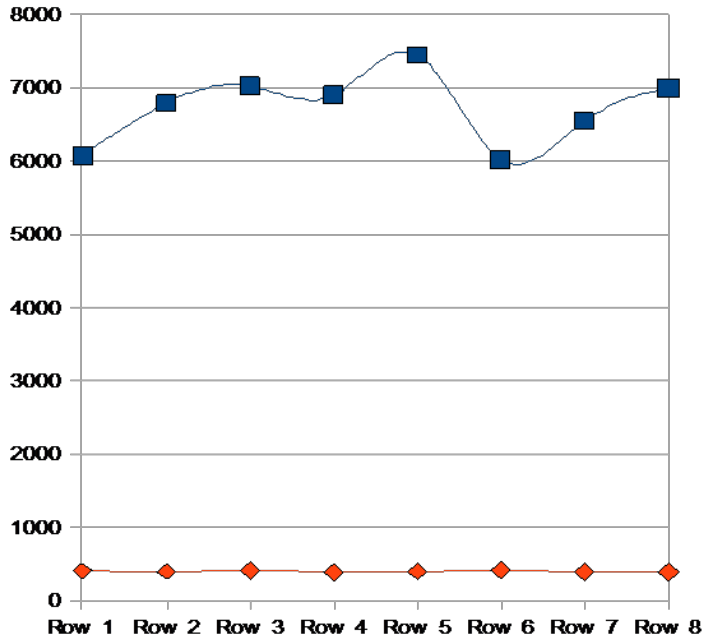
4K aligned

Performance Comparison

- Comparison of experimental results with our testbed



READ IOPS



WRITE IOPS

4K misaligned

4K aligned

Conclusion

- Large sector size SSD and HDD represents the future storage device
- The chasing for speed and capacity never ends
- The transition from 512B to 4KB sector size disk drive should last for a relative long time
- The entire industry food chain needs to prepare for the transition

Questions?