HybridStore: A Cost Efficient, High Performance Storage System Combining SSDs and HDDs

July 26th 2011

Youngjae Kim, Aayush Gupta, Bhuvan Urgaonkar, Piotr Berman, and Anand Sivasubramaniam

Oak Ridge National Laboratory
Pennsylvania State University
Enterprise-scale Storage Systems

Enterprise-scale Hard Disk Drive

- **Enterprise-scale Storage Systems**
  - Information technology focusing on storage, protection, retrieval of data in LARGE-SCALE environments

- **Data-centric services**
  - File, web & media servers, transaction processing servers

Google's massive server farms
A Persistent Hurdle in Enterprise Computing

Huge Performance Discrepancy Between CPU and HDD

- Normalized CPU Performance and Media Access Time

- HDD Performance (Random Read IOPs) has been stagnant for decades.
- I/O bottleneck has become increasingly worse over time.

- Flash SSD Potential!!

Measured CPU Performance scaling = 175X

Measured HDD Performance scaling = 1.3X since Jan’96

Source: Intel Measurements

4KB Random Read:
- 35,000 IOPs
- 4,000 IOPs

Intel® X25-E
Seagate Cheetah 15K.6
Emergence of NAND Flash

Embedded, Desktop, and Enterprise

- **Embedded Storage**
  - PDAs, mobile phones, digital cameras

- **Desktop storage**
  - MacBook Air, One Laptop Per Child (OLPC), game consoles, Intel’s X25-E Extreme SATA Solid-State Drive

- **Enterprise scale storage**
  - Fusion-io’s ioDrive, Texas Memory System’s RamSan-500, Symmetrix DMX-4 from EMC

---

**Unknown**

Violin Memory Inc – Violin 1010

- Scalable Memory Architecture (VXM)
- 84 VIMMs (Violin Intelligent memory Modules)
- (1M random IOPs, PCIe x4/x8 I/F, DRAM/Flash SSD)

---

**Fusion-IO’s ioDrive Duo (MLC)**
- (100KIOPS for Reads, 141KIOPS for Writes)

---

Intel 320 MLC Series
- (38K IOPS for Reads, 1.4K IOPS for Writes)

---

**Price**

- $8,335 / 320GB
- $219 / 120GB
Contents

○ Introduction

○ Background
  ● NAND Flash based SSDs versus HDD
  ● Motivation for HybridStore and Related Works

○ Overview of HybridStore
  ● Capacity Planner
    – Workload Analyzer
    – Storage Optimization Solver

○ Experimental Results

○ Conclusion
Emergence of NAND Flash based SSD

NAND Flash vs. Hard Disk Drives

• Pros:
  – Semi-conductor technology, no mechanical parts
  – Offers lower and more predictable access latencies
    ✦ Microseconds (45us Reads / 200us Writes) vs. Milliseconds for Hard Disks
  – Lower power consumption
  – Higher robustness to vibrations and temperature

• Cons:
  – Limited lifetime
    ✦ 10K - 1M erases per block
  – High cost
    ✦ About 8X more expensive than current hard disks
  – Random writes can be sometimes slow
NAND Flash based SSD

System Architecture

fwrite (file, data) → Process → Process → Process → Application

File System (FAT, Ext2, NTFS...) → Block Device Driver

Block Interface (SATA, SCSI, etc) → Memory

Write Buffer, Read Cache → CPU (FTL) → Garbage Collector, Wear-Leveler, Address Translation

Control Signal → Page write (Bank, Block, Page) → Block write (LBA, size)
Existing Storage Server Platform

- Examples of Storage Server Platform
  - Various network interface
    - Fibre Channel, SAS etc
  - Various types of hard disk drives
    - 2.5” SAS drive, 3.5” SATA drive, etc
Can SSDs replace HDDs?

- **Challenges**
  - Unique performance characteristics of SSD
    - SSD may become worse than HDD due to GC.
  - Reliability Concerns
    - Lifetime of SSDs is limited by the write rates.
  - Cost Concerns
    - NAND Flash is still expensive over HDD.

- **HybridStore**
  - Hybrid storage systems that combine HDDs and SSDs.
Existing Proposals in Enterprise

- **Hybrid Hard Disk**
  - NAND Flash is on-board cache in HDD.

- **Intel Turbo Memory (ITM) [ACM TOS’08]**
  - Support for the ReadyBoost and Ready Drive of Microsoft

- **Two-tier Architecture from Microsoft [Eurosys’09]**
  - Use SSDs as Long-Term read Cache and Short Write Buffer

- **ZFS (designed by Sun)**
  - ReadZilla & LogZilla (Implementation of read cache and write buffer)
Overview of HybridStore

Capacity Planner and Dynamic Controller

Capacity Planner

Admin

Requirement Lists
- Applications
- Available Budget
- Performance Requirement, etc

Optimal SSD, HDD Configuration

Capacity Planner

Estimator
- Performance
- Lifetime

Dynamic Planning

Re-Capacity Planning

Device Driver

Dynamic Controller

Dynamic Controller
- Performance Predictor
- Fragmentation Buster (Front)
- Adaptive Wear-leveler (Front)
- Write Regulator

SSD Array

Adaptive Wear-leveler (Back)

Fragmentation Buster (Back)

HDD Array

Client’s requests

Monitor

Dispatcher

Data Partition Table

Long-Term Per. Predictor

Write Regulator

Frag. Busting

Perf. Predictor

Time

12

Short-Term
# Capacity Planning

## Problem Formulation: Goal and Constraints?

<table>
<thead>
<tr>
<th>Goal</th>
<th>Minimize Cost of HybridStore</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constraints</td>
<td>1) Perf. of HybridStore &gt; Perf. Budget</td>
</tr>
<tr>
<td></td>
<td>2) Lifetime of HybridStore &gt; Lifetime Budget</td>
</tr>
</tbody>
</table>

\[
\text{Cost of HybridStore} = \text{Cost SSDs} + \text{Cost HDDs} + \text{Cost Recur}
\]

**Inputs**
1. Workload Characteristics
2. Hardware Properties (SSD and HDD)

**Constraints**
1. Performance requirement
2. Lifetime requirement

**Capacity Planner**
1. Capacity of SSD
2. Workload Partitioning
Capacity Planning

HybridStore Hardware Model

- **Provisioning SSDs**
  - Find storage capacity of SSDs and HDDs
  - Find out the amount of data partition sent to SSDs for a given workload

- **Storage Model & Data Partitioning**
I/O workloads can be characterized by

- Hot (highly accessed) and cold (rarely accessed) data, Read/write ratio, Sequentiality, Request arrival rate, etc

**Data Classification**
- A methodology to partition a workload into smaller subsets.

**Finding workload attributes**
- The entire logical address space of the workload is divided into fixed-size chunks (or records), then, mapped to different data classes.
  - 1MB record size is used because 1MB roughly corresponds to the granularity of data prefetching done by HDDs/SSDs.
- Each data record is represented by the following workload attributes
  - Temporality (frequency of accesses per unit time)
  - Read/write ratio
  - Request size (spatial locality) – sequential, partially sequential, partially random, and random.
  - Request arrival rate
Hierarchical Data Classification

- Tuples (Hot or cold, Read ratio, Sequentiality (request size), Arrival rate)

Partitioned logical address by records

0 1 2 3 .... 0 1 2 3 .... 0 1 2 3 ....

Cold (0) or Hot (1)?

Read (0) or Write (1)?

Sequentiality (Request Size)?
Eg. <16KB, <32KB, <64KB, others

Intensity of Arrival Rate?
Eg. Lower (25th perc)
  Middle (50th perc)
  Upper (75th perc)
Capacity Planning

Capacity Planner: Problem Formulation

- **Declaration of Variables**
  - Properties of *device type* $i$
    
    $C_i = \text{Capacity of device type } i$
    $U_i = \text{Utilization of device type } i$
    $B_i = \text{Maximum Bandwidth of device type } i$
  
  - Properties of *data class* $j$
    
    $S_j = \text{Size of data class } j$
    $F_i = \text{Frequency of data class } j$
    $W_{ij} = \text{Weight factor for bandwidth of data class } j \text{ on } y_i \text{ devices of device type } i$

- **Decision Variables**
  
    $x_{ij} = \text{data of class } j \text{ on } y_i \text{ devices of device type } i$
    $y_i = \text{number of devices of device type } i$

  Integer variable
Capacity Planning

Mixed Integer Linear Programming

- **Objective Function**

\[
Cost_{HybridStore} = Cost_{Installation} + Cost_{Recurring}
\]

\[
= \left( \sum_{i=1}^{I} y(i) \times D^S(i) \times C_i \right) + \left( K^S \times \sum_{i=1}^{I} y(i) \times \int P(t) dt \right)
\]

- **Constraints**

\[
\sum_{i} x_{ij} = S_j, \quad (\forall j \in J)
\]

\[
\sum_{j} x_{ij} \leq (U_i \times C_i) \times y_i, \quad (\forall i \in I)
\]

\[
F_j \times \frac{x_{ij}}{S_j} \leq B_{ij} \times y_i, \quad (\forall i \in I, \forall j \in J)
\]

Expected lifetime = \( \frac{\text{Size of NAND flash} \times \# \text{ of erase cycles}}{\text{bytes written per day}} \)

Lifetime\((i,x)\) \leq \text{Useful Lifetime of HDD} \quad (i \in \text{Flash based SSDs})
Evaluating HybridPlan

- **Solver development**
  - Developed a trace analyzer (lines of codes less than 500)
  - Developed the solver of HybridPlan using CPLEX
    - CPLEX, a well-regarded Integer Linear Programming (ILP) solver

- **Workloads**
  - Synthetic workloads
  - Realistic workloads
    - MSR Cambridge traces, and Microsoft Exchange server Traces

- **Devices**

<table>
<thead>
<tr>
<th>Device</th>
<th>Type</th>
<th>Capacity (GB)</th>
<th>Per-GB ($)</th>
<th>Utilization</th>
<th>Read (MB/s)</th>
<th>Write (MB/s)</th>
<th>Latency (ms)</th>
<th>Erase (#)</th>
<th>Power (W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seagate Cheetah</td>
<td>15K HDD</td>
<td>146</td>
<td>1.80</td>
<td>0.8</td>
<td>171</td>
<td>171</td>
<td>3.6</td>
<td>-</td>
<td>12.92</td>
</tr>
<tr>
<td>Seagate Barracuda</td>
<td>7.2K HDD</td>
<td>750</td>
<td>0.17</td>
<td>0.8</td>
<td>125</td>
<td>125</td>
<td>4.2</td>
<td>-</td>
<td>9.4</td>
</tr>
<tr>
<td>Intel X 25-E</td>
<td>SLC SSD</td>
<td>32</td>
<td>11.96</td>
<td>0.5</td>
<td>230</td>
<td>200</td>
<td>0.125</td>
<td>100K</td>
<td>2</td>
</tr>
<tr>
<td>Intel X-25-M</td>
<td>MLC SSD</td>
<td>80</td>
<td>3.22</td>
<td>0.5</td>
<td>220</td>
<td>80</td>
<td>0.25</td>
<td>10K</td>
<td>2</td>
</tr>
</tbody>
</table>
## Synthetic Workloads

### Description of Synthetic Workloads

<table>
<thead>
<tr>
<th>Workloads</th>
<th>Index</th>
<th>Read (%)</th>
<th>Size (KB)</th>
<th>Inter-Arrival</th>
<th>I/O Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Time (ms)</td>
<td>MB/s</td>
</tr>
<tr>
<td>Sequential Read</td>
<td>SR1</td>
<td>80</td>
<td>128</td>
<td>100 (L)</td>
<td>1.25</td>
</tr>
<tr>
<td></td>
<td>SR2</td>
<td>80</td>
<td>128</td>
<td>2 (M)</td>
<td>62.5</td>
</tr>
<tr>
<td></td>
<td>SR3</td>
<td>80</td>
<td>128</td>
<td>0.2 (H)</td>
<td>1,250</td>
</tr>
<tr>
<td>Random Read</td>
<td>RR1</td>
<td>80</td>
<td>4</td>
<td>100 (L)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>RR2</td>
<td>80</td>
<td>4</td>
<td>2 (M)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>RR3</td>
<td>80</td>
<td>4</td>
<td>0.2 (H)</td>
<td>-</td>
</tr>
<tr>
<td>Sequential Write</td>
<td>SW1</td>
<td>20</td>
<td>128</td>
<td>100 (L)</td>
<td>1.25</td>
</tr>
<tr>
<td></td>
<td>SW2</td>
<td>20</td>
<td>128</td>
<td>2 (M)</td>
<td>62.5</td>
</tr>
<tr>
<td></td>
<td>SW3</td>
<td>20</td>
<td>128</td>
<td>0.2 (H)</td>
<td>1,250</td>
</tr>
<tr>
<td>Random Write</td>
<td>RW1</td>
<td>20</td>
<td>4</td>
<td>100 (L)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>RW2</td>
<td>20</td>
<td>4</td>
<td>2 (M)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>RW3</td>
<td>20</td>
<td>4</td>
<td>0.2 (H)</td>
<td>-</td>
</tr>
</tbody>
</table>
Impact of I/O Intensity

- **Results for Sequential Read Dominant Workloads**
  - SR1, SR2, SR3

- SR1 only requires 2 slow 7.2K RPM HDDs whereas it requires 9 fast 15K RPM HDDs.
- Our solver determines the right devices to meet the capacity needs.
- As the arrival rate increases, we observe the need for MLC SSDs (considering $/GB for SLC SSD, it is not efficient to use compared to using MLC SSD).
- Recurring cost (Electricity cost) are quite small compared to device installation cost.
Impact of I/O Intensity

- **Results for Sequential Write Dominant Workloads**
  - SW1, SW2, SW3

  ![Graph showing number of devices vs. total cost for SW1, SW2, and SW3 workloads.](image)

  - For write dominant SW3, unlike observation from SR workloads, the solver suggests to use one SLC SSD instead of the MLC ones for its read-intensive counterpart (SR3). It’s because SLC SSD that we use is 2.5 times faster than the MLC one.
  - Also it needs a sharp increase in the number of slow HDDs because of the vast $/GB difference between SLC SSDs and slow HDDs.
Impact of Sequentiality

Results for Sequential and Random Workloads

- SR, SW

• We clearly see the needs of the larger number of SSDs as the workloads are random.
• For RW3, we observe the needs of SLC-SSDs to meet the high IOPS requirement.
• As a storage administrator, it is highly advisable to increase the sequentiality of incoming workloads so as not to employ expensive SSDs.
Impact of Lifetime Constraint

- Results for without and with lifetime constraints
  - denoted as (A) and (B) respectively

- Lifetime constraint is an important metric in capacity provisioning.
- Without lifetime constraint, we see a greater portion of SSDs being used than with the lifetime constraint.
- For SW3, without lifetime constraint, we may have lower number of devices as well as the overall cost compared to when the lifetime constraint is forced, however, the storage administrator needs to re-provision prematurely, eventually increase the overall costs over the initial estimated period.
Realistic Workloads

- **Description of Realistic Workloads**

<table>
<thead>
<tr>
<th>Workload</th>
<th>Size (TB)</th>
<th>Read (%)</th>
<th>Request Size (KB)</th>
<th>IOPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSR Trace</td>
<td>5.7 TB</td>
<td>68.1</td>
<td>23.32</td>
<td>823</td>
</tr>
<tr>
<td>Exchange Server</td>
<td>750GB</td>
<td>38.3</td>
<td>16.54</td>
<td>3,692</td>
</tr>
</tbody>
</table>


Can SSDs replace HDDs?

- **Results for MSR Traces**

![Graph showing the number of devices and total cost for different storage options.]

- Employing 7.2K RPM HDDs is more economically efficient than employing 15K RPM HDDs.
- In case of SSD systems, it requires several hundreds of SSDs to satisfy the capacity requirement.
- The bounding factor for decision-making of HybridPlan is not I/O bandwidth but storage capacity requirement.
Efficacy of HybridStore

- **Results for MSR Traces**

- HybridPlan can find the most economic storage composition.
- HybridPlan suggests 2 x 7.2K RPM HDDs and 1 MLC SSD for MSR Trace.
- Total cost saving of HybridStore is about 85% compared to high-end HDD only system.
- 99% data are classified into C32, a data class storing data rarely accessed.
Lessons Learned

I. We developed a capacity planner that finds the most economically efficient storage configurations while meeting the performance and lifetime requirements of devices.

II. We provided a general form of comprehensive methodology using a well-known technique for optimization problems, Mixed Integer Linear Programming (LP).

III. Experiments showed that our capacity planner is able to identify close to minimum SSD capacity needed to meet a specified performance goal for realistic workloads while ensuring similar performance as compared to a comparatively more over-provisioned system.