Towards SLO Complying SSDs Through *OPS Isolation*

13th USENIX Conference on FAST 2015

**Jaeho Kim** (University of Seoul, Korea)
Donghee Lee (University of Seoul)
Sam H. Noh (Hongik University)
Applications of Flash Memory

- Area of flash storage
  - From embedded to server storage
Introduction & Motivation

• Virtualization system
  – Need to satisfy Service Level Objective (SLO) for each VM
  – SLO is provided through hardware resource isolation

• Existing solutions for isolating CPU and memory
  – Distributed resource scheduler [VMware inc.]
  – Memory resource management in VMware ESX server [SIGOPS OSR 2002]
Do SSDs provide decent performance isolation?

- Does each VM proportionally consume I/O bandwidth of shared SSD among VMs?

- How does proportionality vary as state of SSD is varied?
Initial Experiments on Commercial SSD

- Linux kernel-based virtual machine (KVM) on 4 VMs
- Proportional I/O weight (by Cgroups feature in Linux kernel 3.13.x)
  - \textbf{VM-x}: x is I/O weight value (Higher value $\rightarrow$ Allocate higher throughput)
- SSD as shared storage
  - 128GB capacity, SATA3 interface, MLC Flash
  - \textbf{clean SSD}: empty SSD
  - \textbf{aged SSD}: full SSD (busy performing garbage collection)
    - Aging is conducted by issuing 4KB $\sim$ 32KB sized random writes for a total write that exceeds the SSD capacity
- Each VM runs the same workload concurrently
  - Financial, MSN, and Exchange

\begin{center}
\begin{tabular}{c|c|c|c}
\hline
Hypervisor (KVM) & \textbf{I/O weight} & Two state: Clean & Aged \\
\hline
\hline
\end{tabular}
\end{center}
Results: Proportionality of I/O Bandwidth

- For all workloads, on HDD, proportionality is close to I/O weight except for VM-10
- Proportionality deviation is worse for aged SSD than clean SSD
Monitor Internal Workings of SSD

- Commercial SSD: Proprietary, black box SSDs
- Monitor using Simulator
  - SSD simulator: DiskSim SSD Extension
  - Workloads: Financial, MSN, and Exchange
    - Traces are captured as VMs run concurrently on real system
Analysis #1: Mixture of Data

- Within block (GC unit): mixture of data from all VMs

Data of all VMs are mixed into a block

Over-Provisioned Space (OPS)
- reserved space for write reqs.
- used for garbage collection (GC)

Data layout of conventional SSD
Analysis #2: Interference among VMs during GC

- Movement of data: live pages of workloads other than the one invoking GC

Data layout of conventional SSD:
- Over-Provisioned Space (OPS)
  - reserved space for write reqs.
  - used for garbage collection (GC)

1) Victim block for GC
2) Pages moved to OPS
Analysis #3: **Work induced by other VMs**

- From one VM’s viewpoint: **doing unnecessary work** induced by other workloads.

While executing the VM-F (Financial) workload, only 30% of them are its own pages.

Number of pages moved for each workloads during GC.
More Closely

- GC leads to interference problem among VMs
- GC operation employed by one VM is burdened with other VM’s pages

Employed by VM-F
"Why do I have to clean others?"

GC operation in conventional SSD
Avoiding Interference

- Cost of GC is major factor in SSD I/O performance
- Each VM should pay only for its own GC operation
Proposed scheme: **OPS isolation**

- Dedicate flash memory blocks, including OPS, to each VM separately when allocating pages to VMs
  ➔ Prevent interference during GC
VM OPS Allocation

• How much OPS for each VMs to satisfy SLO?
### IOPS of SSD

#### Parameter | Meaning
--- | ---
$tGC$ | Time to GC (depends on utilization ($u$) of victim block at GC)
$tPROG$ | Time for programming a page (constant value)
$tXfer$ | Time for transferring a page (constant value)

\[
IOPS = \frac{1}{tGC + tPROG + tXfer}
\]

Variable value (Crucial factor for IOPS)

Determined by **OPS size**
How to Meet SLO (IOPS) of each VM?

: Dynamically adjusting OPS

SSD – state #1

Data Space

SSD – state #2

Data Space

IOPS of state #2

IOPS of VM1 = Prev. IOPS + Δ
IOPS of VM2 = Prev. IOPS
IOPS of VM3 = Prev. IOPS – Δ
Evaluation of **OPS isolation**

- **Evaluation environment**
  - **SSD simulator**: DiskSim SSD Extension
    - FTL: Page-mapped FTL
    - GC: Greedy policy
    - Aged state SSD
  - **Workloads**:
    - Financial, MSN, and Exchange
      - Traces are captured as VMs run concurrently on real system
  - **Host interface**
    - Tags of VM ID are informed to SSD

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page size</td>
<td>4KB</td>
</tr>
<tr>
<td>Block size</td>
<td>512KB</td>
</tr>
<tr>
<td>Page read</td>
<td>60us</td>
</tr>
<tr>
<td>Page write</td>
<td>800us</td>
</tr>
<tr>
<td>Block erase</td>
<td>1.5ms</td>
</tr>
<tr>
<td>Xfer latency (Page unit)</td>
<td>102us</td>
</tr>
<tr>
<td>OPS</td>
<td>5%</td>
</tr>
</tbody>
</table>
Results

- **x-axis:** groups of VMs that are executed concurrently
- **y-axis:** proportionality of I/O bandwidth relative to smallest weight

### SLO satisfied by OPS isolation

- **(Ideal)**
- VM-F (Financial)
- VM-M (MSN)
- VM-E (Exchange)

### Weights allotted to VMs

<table>
<thead>
<tr>
<th>Group of VMs</th>
<th>Proportionality of I/O bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:5:10</td>
<td>12</td>
</tr>
<tr>
<td>10:2:1</td>
<td>10</td>
</tr>
<tr>
<td>1:2:5</td>
<td>8</td>
</tr>
</tbody>
</table>
Conclusion

• Performance SLOs can not be satisfied with current commercial SSDs
  – Garbage collection interference among VMs

• Propose *OPS isolation*, allocates flash memory blocks so that VM is isolated from other VMs
  – Do not allow mix of pages in same block
  – Size of OPS is dynamically adjusted per VM

• Evaluation showed that OPS isolation is an effective way for SSDs to provide performance SLOs to competing VMs
Thank you! & Questions?

Towards SLO Complying SSDs Through *OPS Isolation*

Please visit our poster at tonight.

Jaeho Kim (kjihnet@gmail.com, University of Seoul, Korea)
Donghee Lee (University of Seoul)
Sam H. Noh (Hongik University)