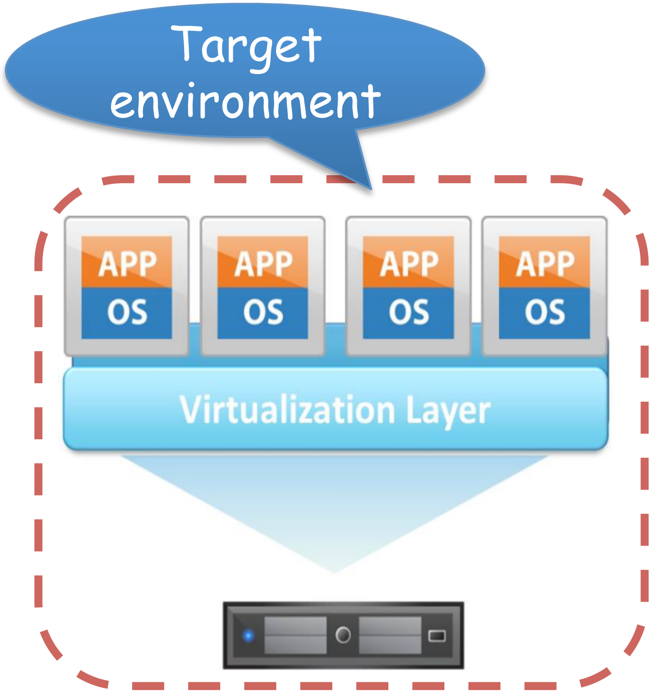
Towards SLO Complying SSDs Through *OPS Isolation*

13th USENIX Conference on FAST 2015

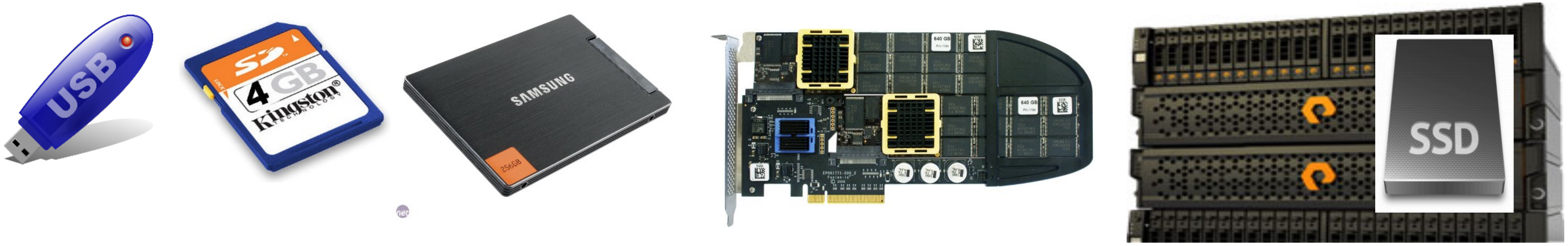
Jaeho Kim (University of Seoul, Korea)
Donghee Lee (University of Seoul)
Sam H. Noh (Hongik University)

Applications of Flash Memory

- Area of flash storage
  - From embedded to server storage



Target environment

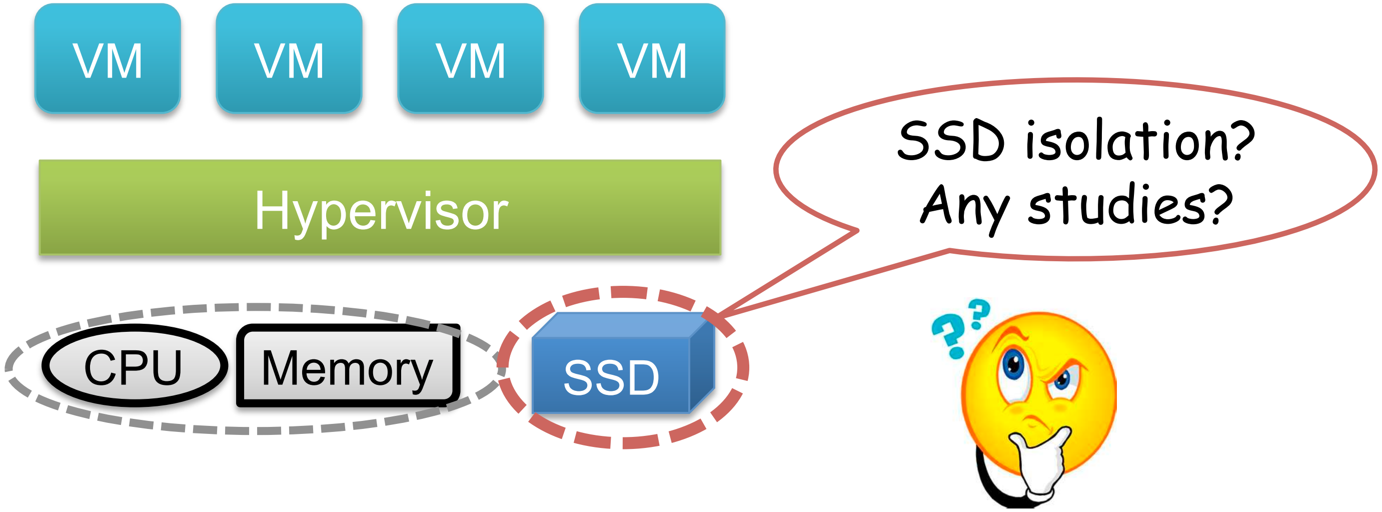


Application field

Introduction & Motivation

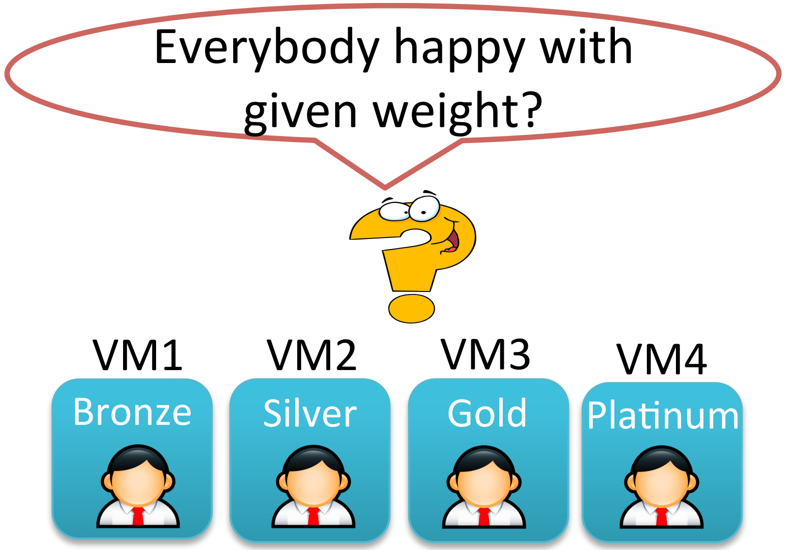
- Virtualization system
  - Need to satisfy Service Level Objective (SLO) for each VM
  - SLO is provided through hardware resource isolation

- 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?



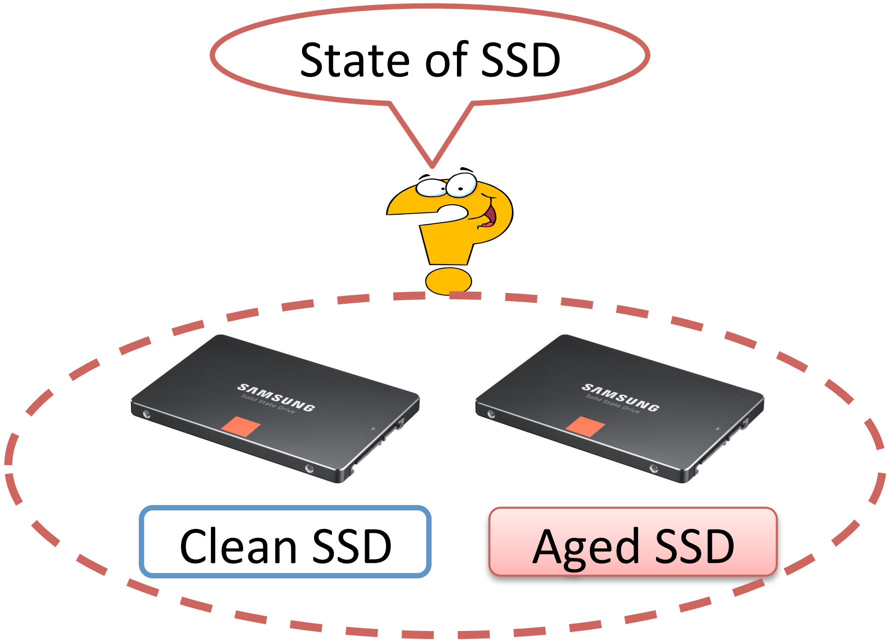
Everybody happy with given weight?

VM1 Bronze

VM2 Silver

VM3 Gold

VM4 Platinum



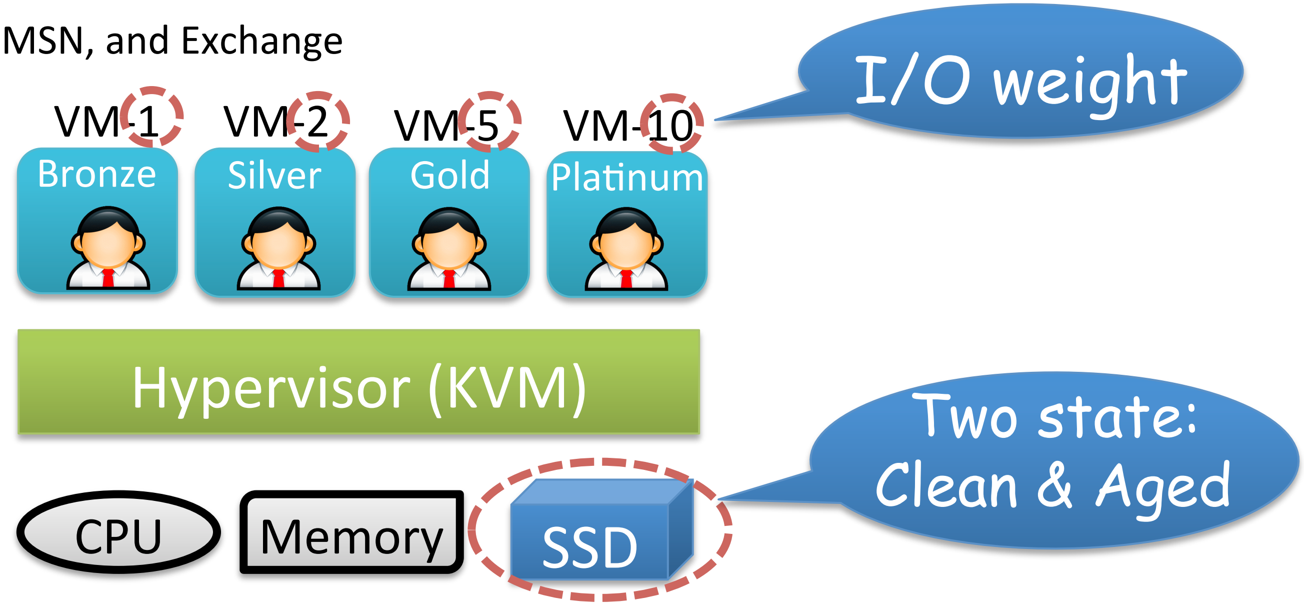
State of SSD

Clean SSD

Aged SSD

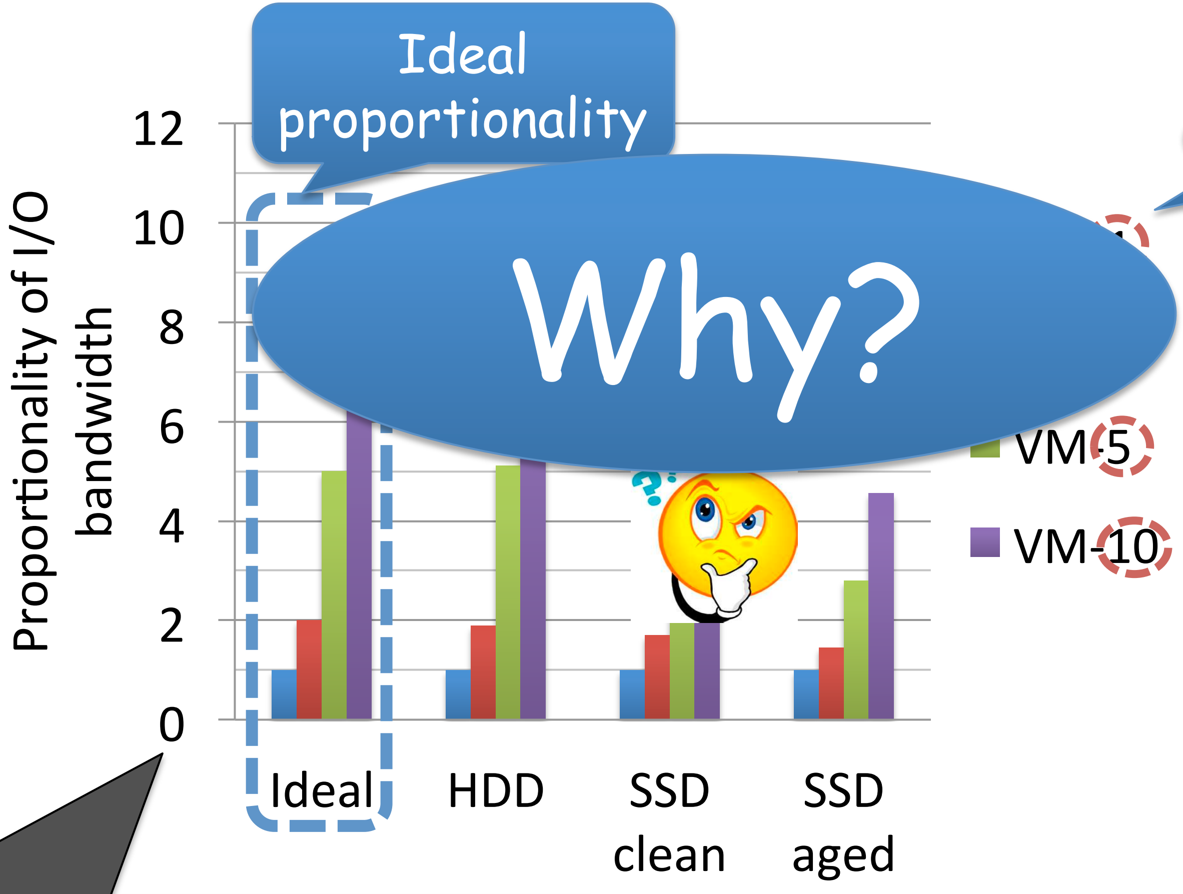
Initial Experiments on Commercial SSD

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



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



Ideal proportionality

I/O weight

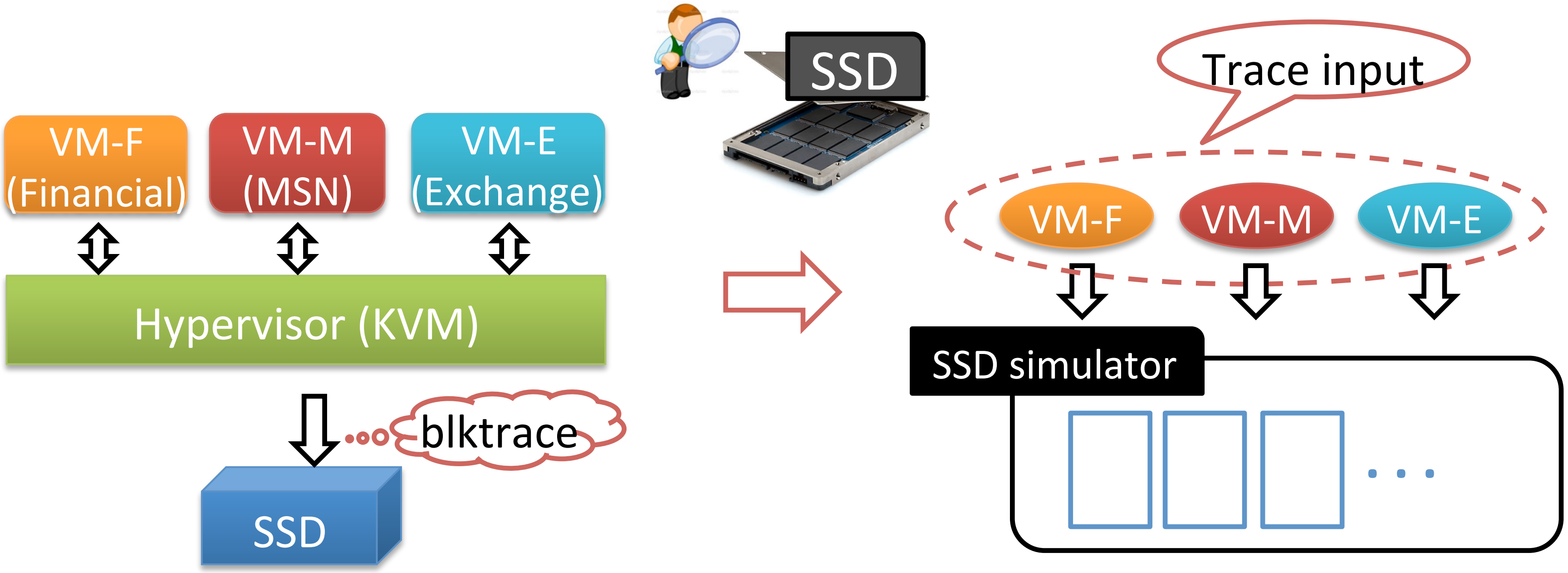
Why?

I/O bandwidth relative to VM-1

Financial

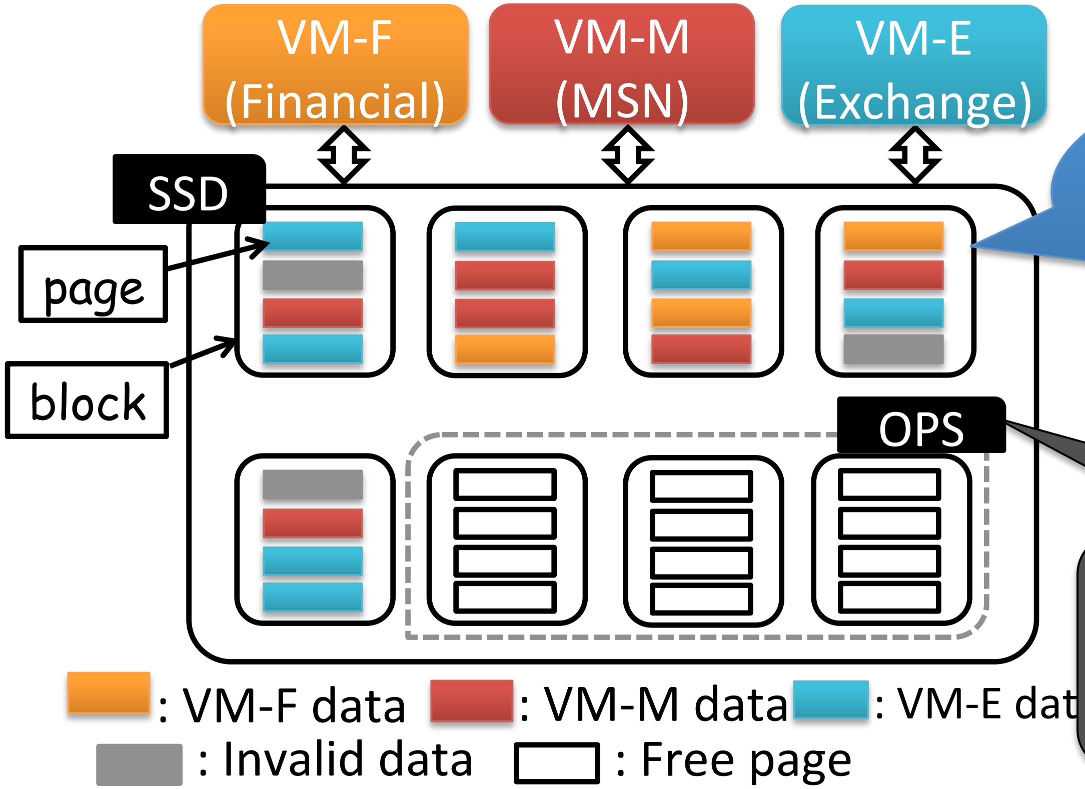
Monitor Internal Workings of SSD

- 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



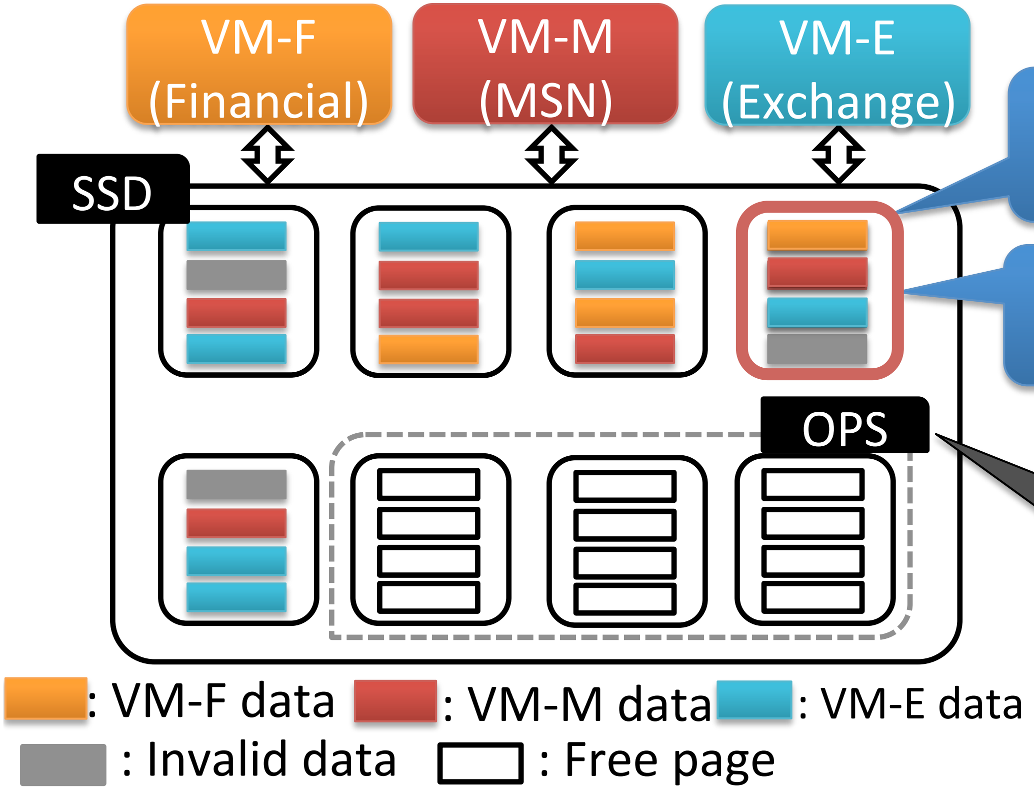
Data of all VMs are mixed into a block

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



1) Victim block for GC

2) Pages moved to OPS

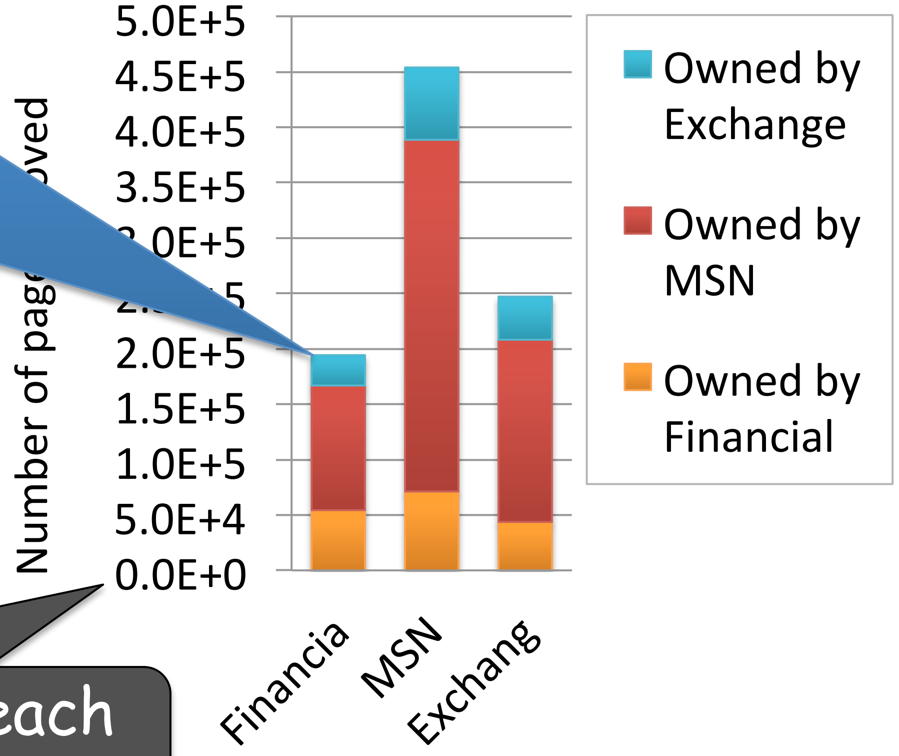
Over-Provisioned Space (OPS)
- reserved space for write reqs.
- used for garbage collection (GC)

Data layout of conventional SSD

Analysis #3: Work induced by other VMs

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

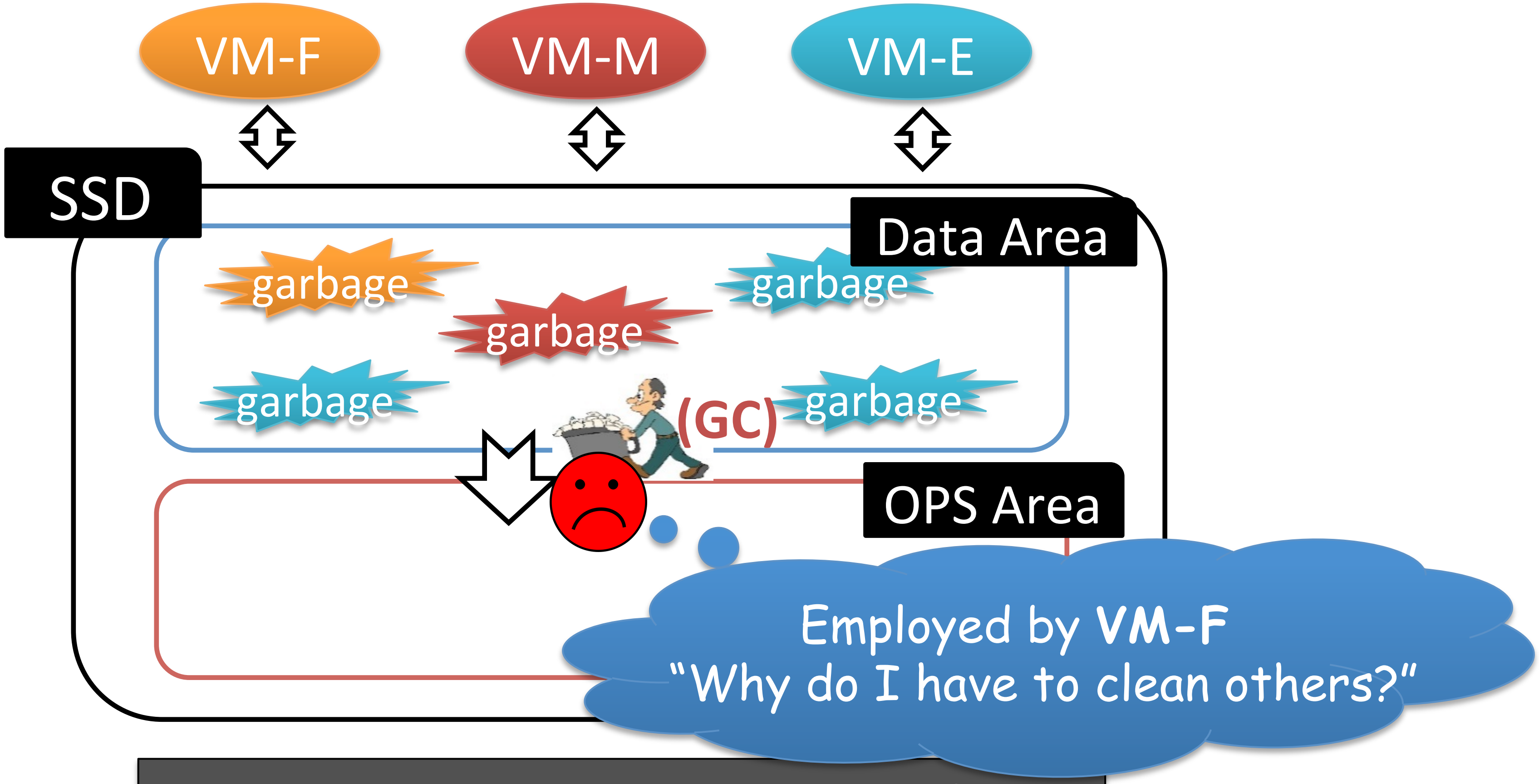
While executing the VM-F (Financial) workload, only 30% of them are its own pages



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

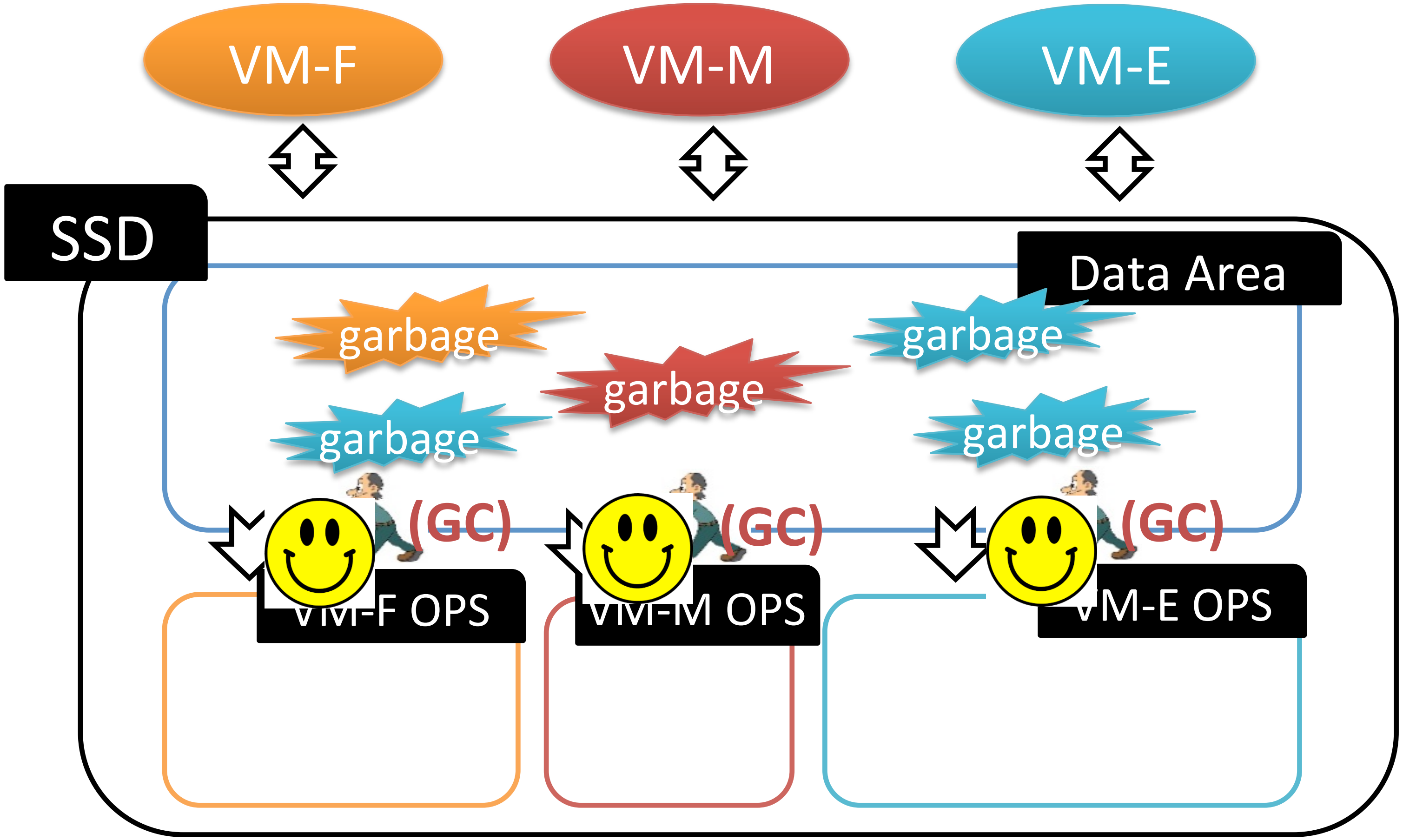


Employed by **VM-F**
"Why do I have to clean others?"

GC operation in conventional SSD

Avoiding Interference

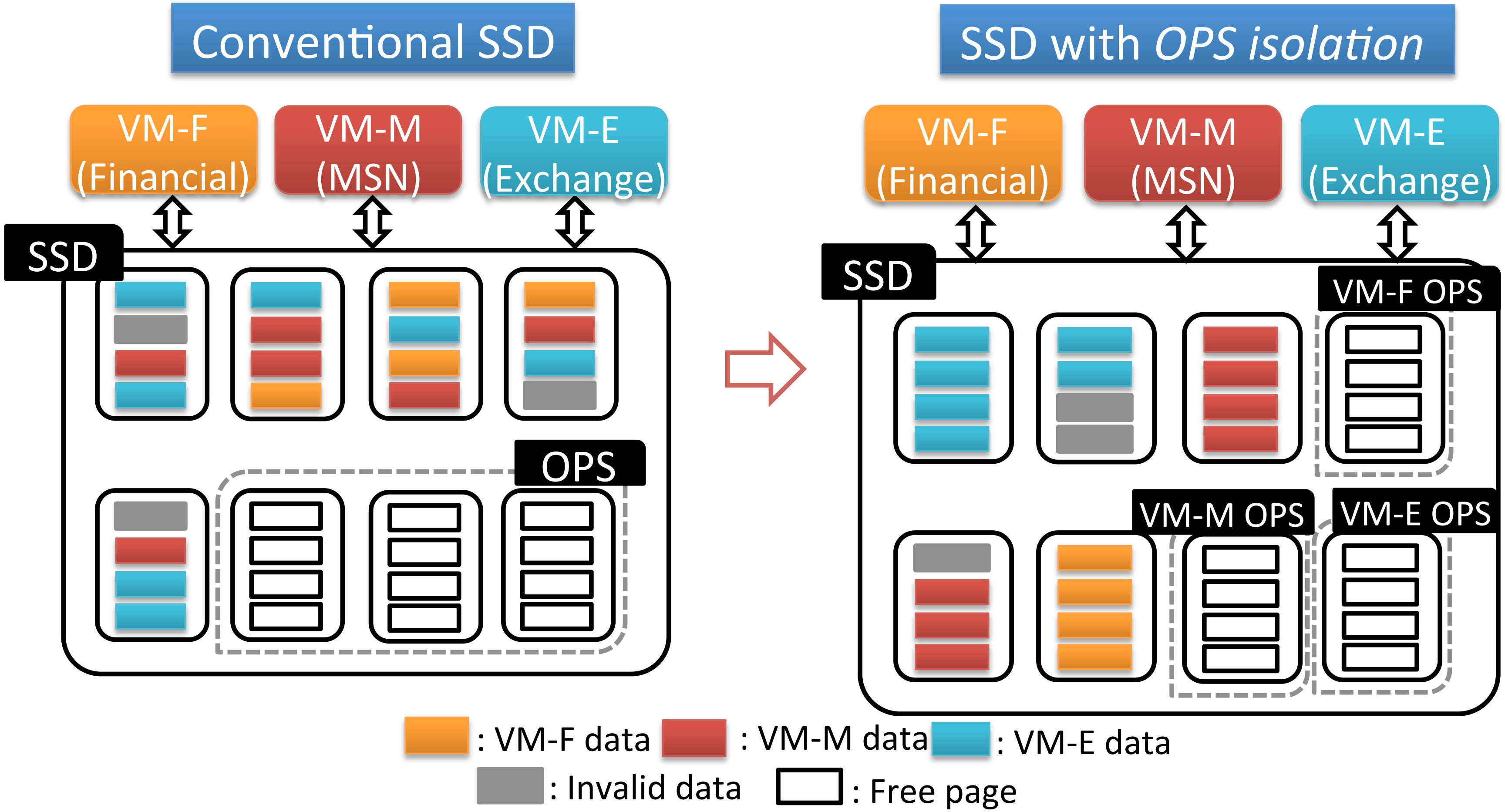
- Cost of GC is major factor in SSD I/O performance
- Each VM should pay only for its own GC operation



Proposed scheme: *OPS isolation*

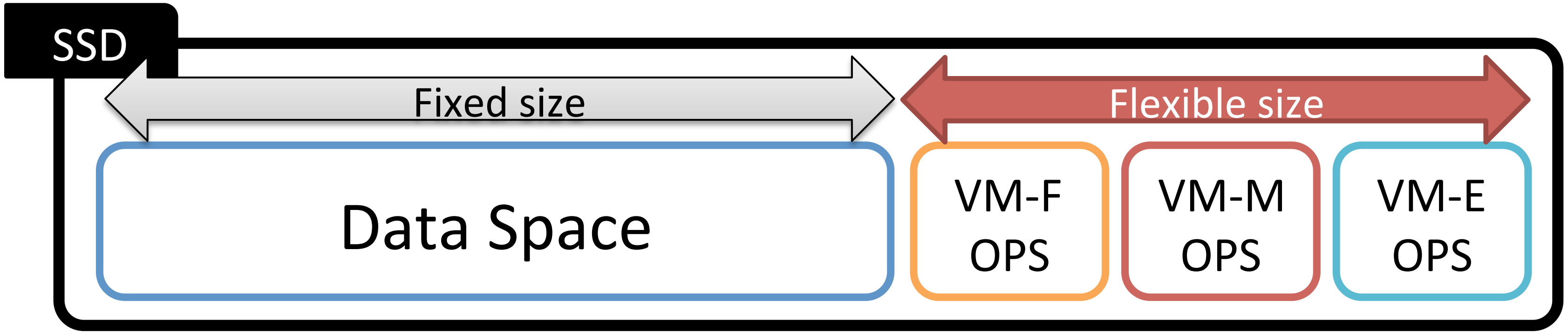
- Dedicate flash memory blocks, including OPS, to each VM separately when allocating pages to VMs

➔ Prevent interference during GC

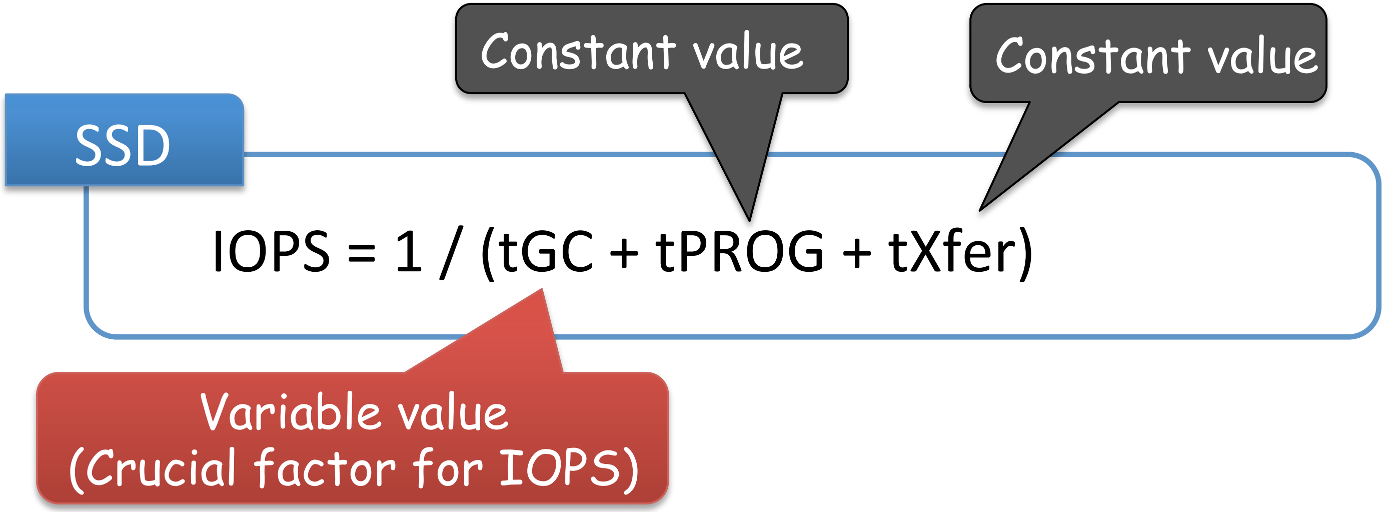


VM OPS Allocation

- How much OPS for each VMs to satisfy SLO?



IOPS of SSD



SSD

$$\text{IOPS} = 1 / (\text{tGC} + \text{tPROG} + \text{tXfer})$$

Constant value

Constant value

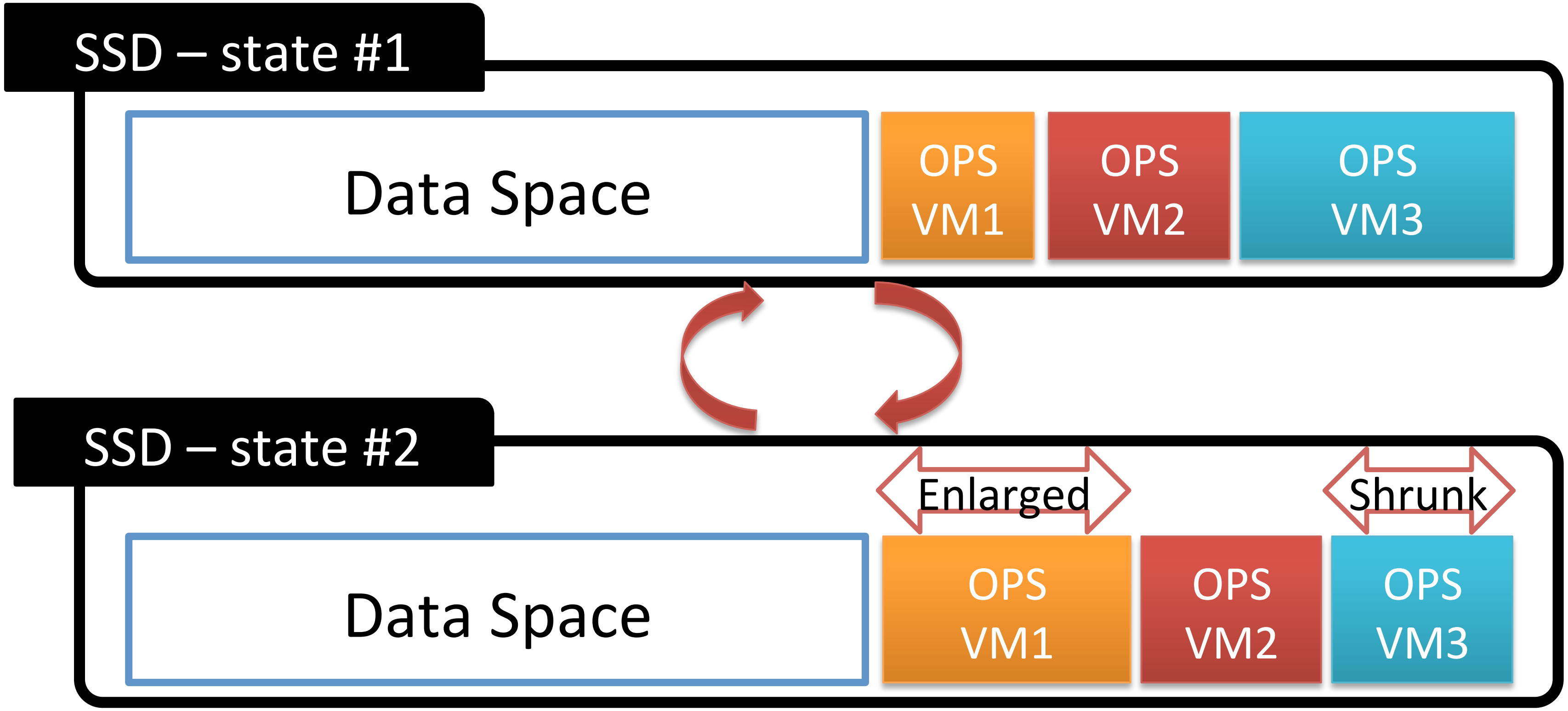
Variable value
(Crucial factor for IOPS)

Determined by **OPS size**

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)

How to Meet SLO (IOPS) of each VM?

: Dynamically adjusting OPS



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*

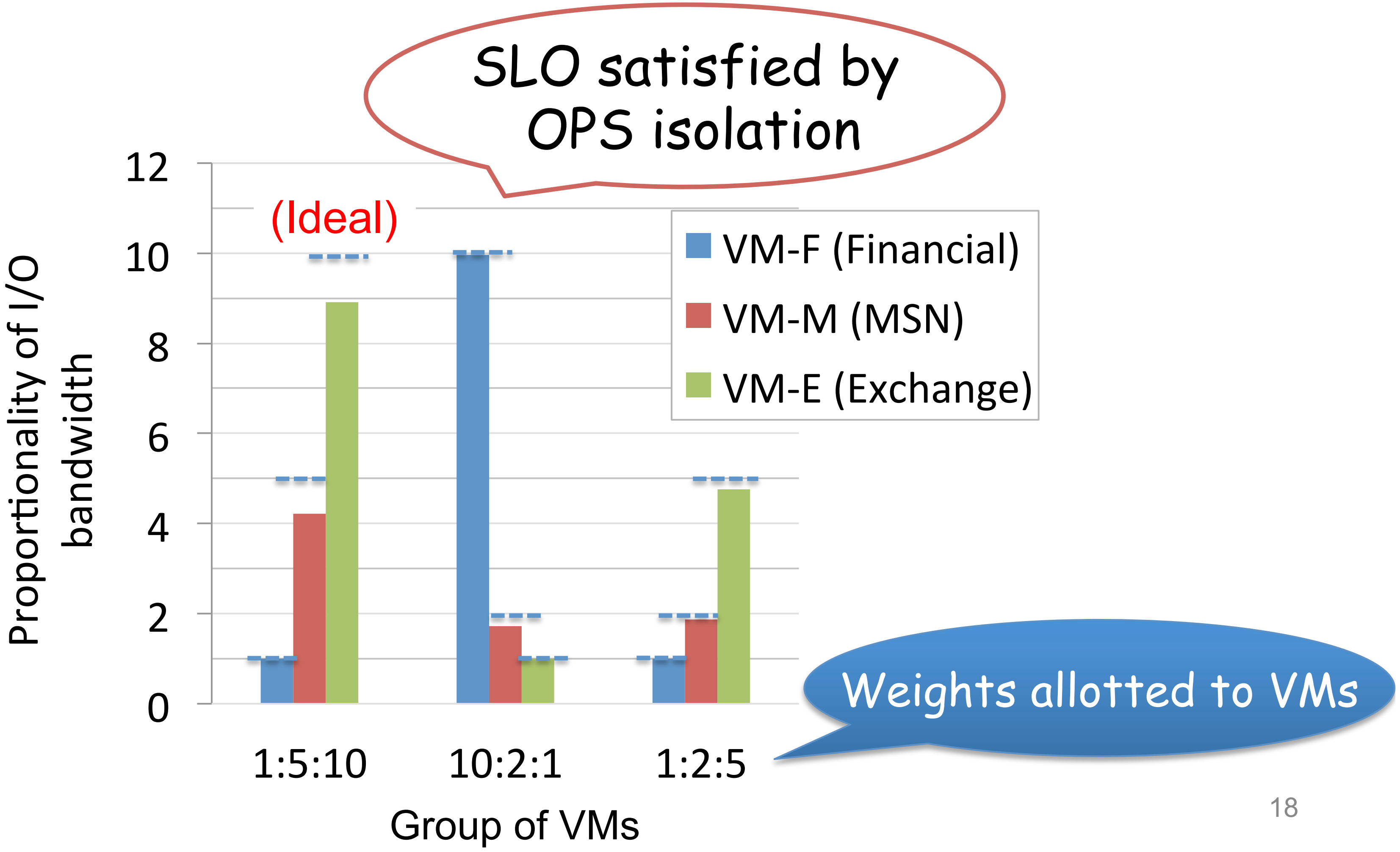
- Evaluation environment
  - SSD simulator: DiskSim SSD Extension
    - FTL: Page-mapped FTL
    - GC: Greedy policy
    - Aged state SSD

Parameter	Description
Page size	4KB
Block size	512KB
Page read	60us
Page write	800us
Block erase	1.5ms
Xfer latency (Page unit)	102us
OPS	5%

  - 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

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

Weights allotted to VMs

Conclusion

- Performance SLOs can not be satisfied with current commer cial 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 (kjhnet@gmail.com, University of Seoul, Korea)
Donghee Lee (University of Seoul)
Sam H. Noh (Hongik University)