VERITAS

Comparison of Write Back Strategies

Mahesh Khatpe, Pradeep Gothoskar

Agenda

- Introduction
- Differentiation
- Applications
- Conclusion

Introduction

Four basic concerns

- **Bandwidth -** how many ops/sec are sustained
- **Resiliency -** how is the failure tolerance
- **Restoration -** what is the promise of recovery
- **Capacity -** how much of Hot/Cold data to be held

Introduction

Motivation

- **The Shift** - of focus from $/GB to $/IOPs
- **The Split** - separation of hot and cold data
- **The Apps** - In new-age workloads such as archival and analytics have large % writes and demand low latencies
- **Relaxation** - strictly Vs eventually consistent.
- **Restoration** - Fast reads important to Applications | Fast writes, consistency, restoration important to designers

This makes write-back strategies more important now

Differentiation

Write Back & Efficiency

Write Back Cache

- Reduction in the **Bandwidth** in order to identify data and decide when to flush data
- **Write Efficiency** depends on cache line and % write.
- **Cache size** depends on how much latency tolerance is needed
- **Write Bursts** slower to absorb due to continuous flush to tier1
- Works better if there is write **localization in time**

Write Back Log

- No **Bandwidth** overhead of managing dirty data.
- **Write Efficiency** doesn't depend on cache line or % write.
- **Cache size** depends on how much space can be given to log
- **Write Bursts** faster to absorb due appending sequential writes
- Works better even when there is little **localization in space**

Differentiation

Write Pattern & Restore

Write Back Cache

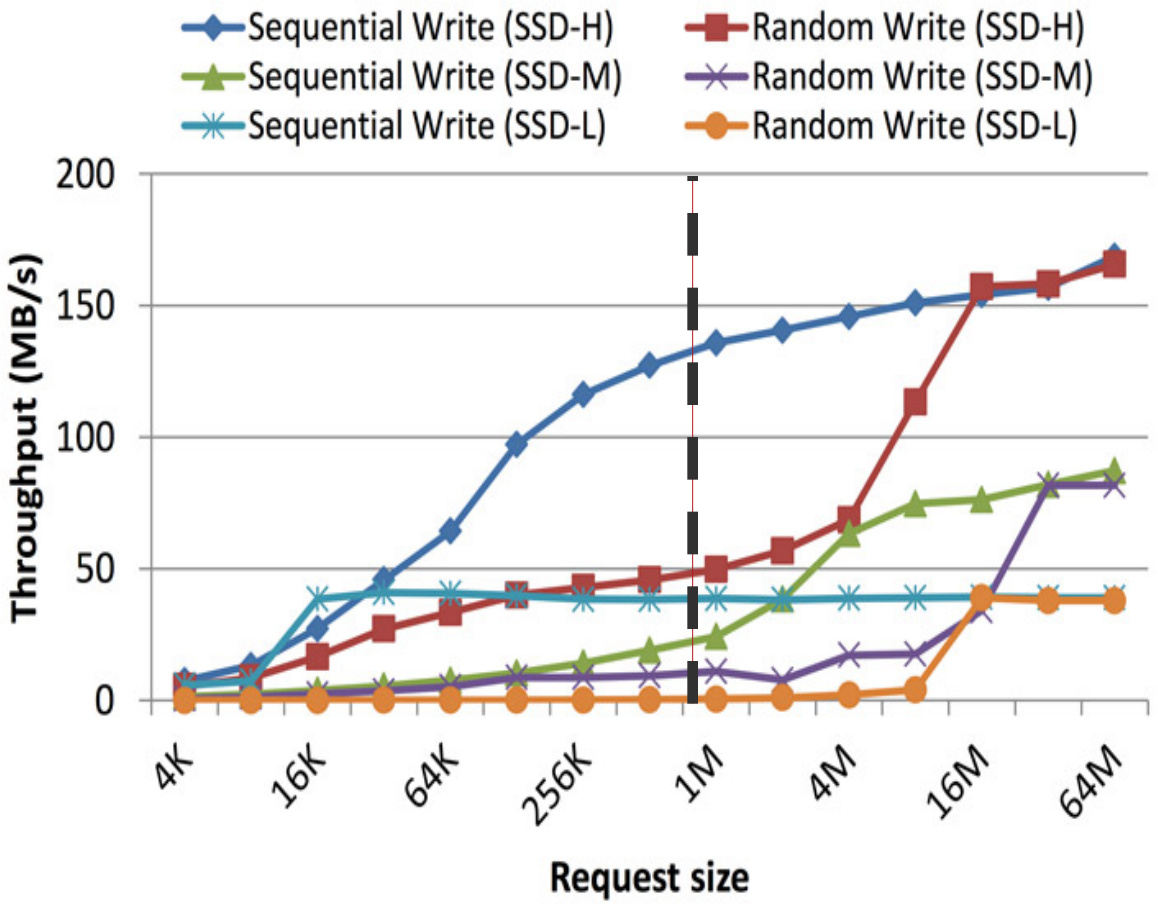
- **Re-read cost** depends on flushing policy
- Better for higher **Hot/Cold ratio**
- **Recovery & Restore** can be costly and hard for a point in time data.
- **In-line processing** difficult due to continuous flush
- **Sustained Performance** difficult to predict and control

Write Back Log

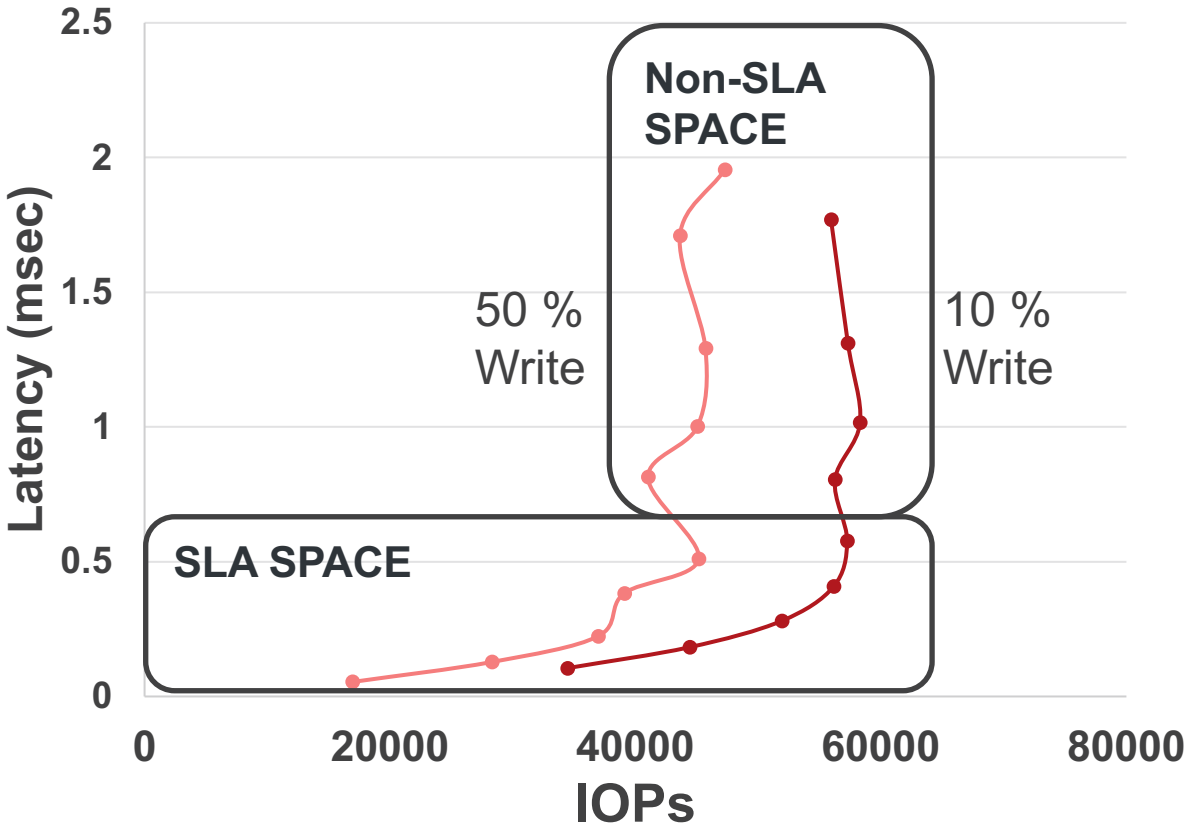
- **Re-read cost** can reduce due to merge factor
- Benefits for smaller **Hot/Cold ratio**
- **Recovery & Restore** less costly and to a point in time data or versions.
- **In-line processing** possible, merge, reorder
- **Sustained Performance** easier to predict and control

Applications

- Random Writes Suffer relative to Sequential writes

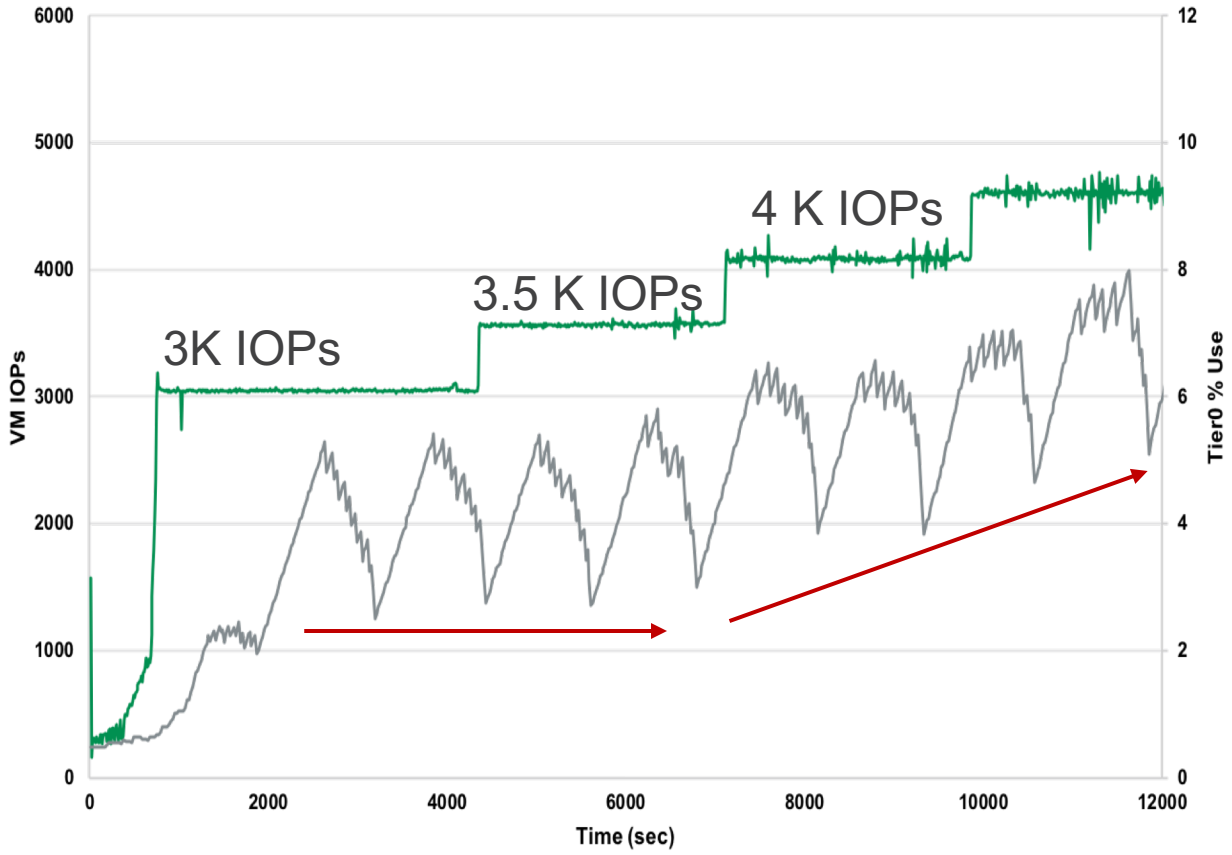


- Write back strategy can expand the SLA Space

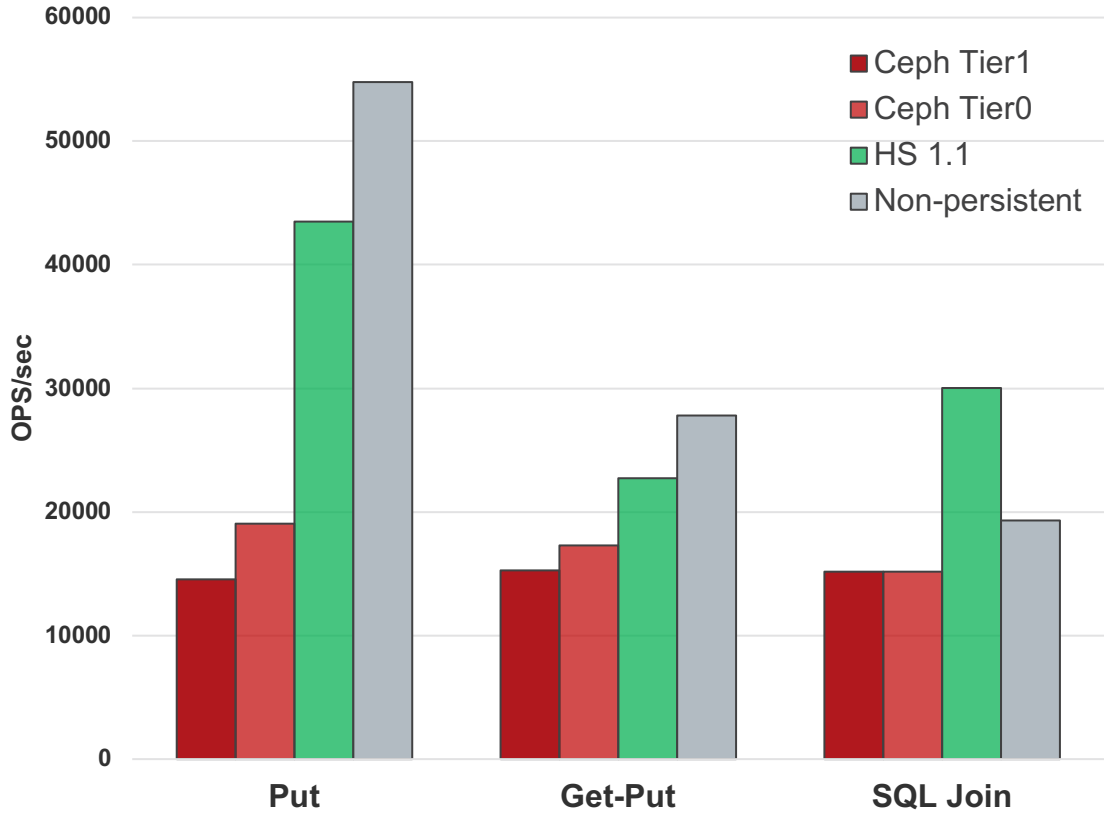


Applications

- Write fill can be predicted and controlled based on Write Flush



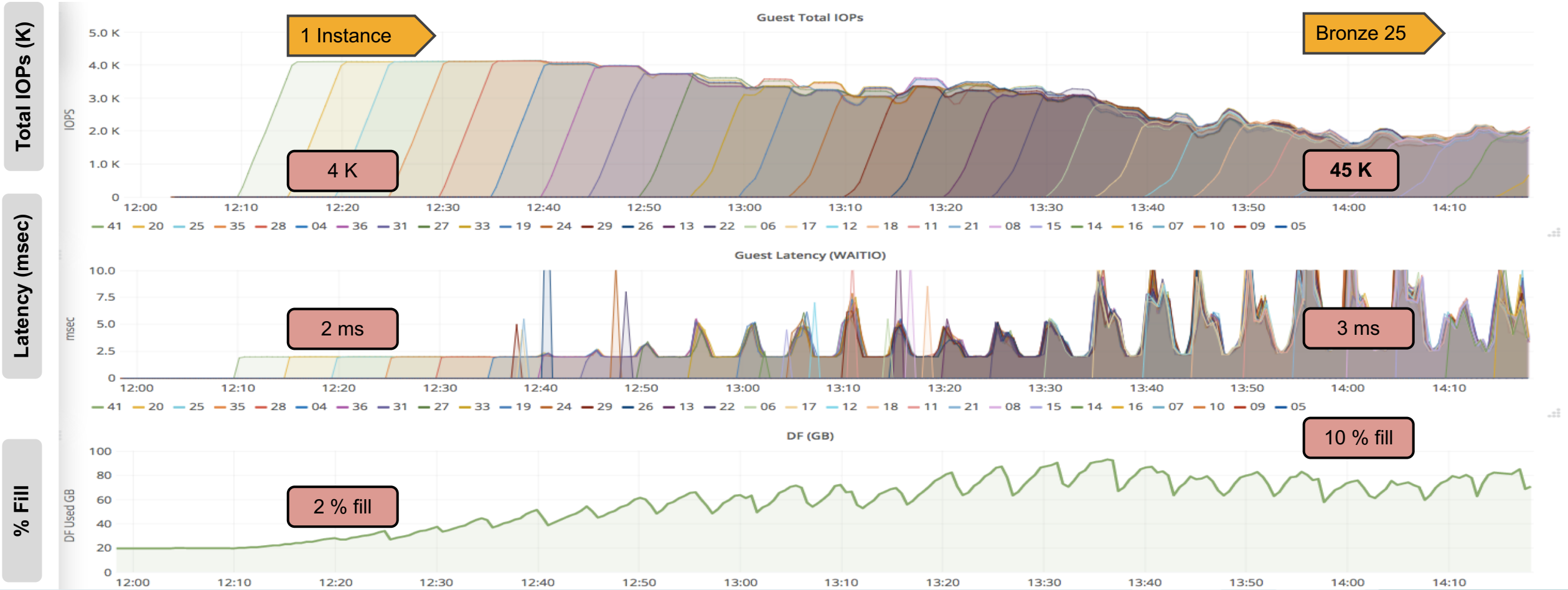
- Where you stage write back affects the bandwidth



Applications

25 Instances
Random 70:30 read:write

Bandwidth 45K IOPs
Latency 3 msec



Applications

Workload	Read/Write	Seqn/Rand	Hot/Cold	iosize	Localization	WB Strategy?
HPC	50:50	90:10	High	Large	High	WBL
VDA	10:90	90:10	Low	Large	High	WBL
Builds	80:20	30:70	Medium	Small	High	-
FileShare	70:30	30:70	Medium	Medium	Moderate	WBC
Database	~ 70:30	10:90	High	Small	Moderate	WBC
Backup	10:90	90:10	Low	Large	High	-
WebServing	90:10	~ 80:20	High	Small	Low	WBC
Search Engines	90:10	~ 40:60	High	~ Large	Low	WBC
VDI	70:30	~ 20:80	Medium	Medium	Moderate	WBL

Conclusion

You may want to try write-back-log -

- if workload has rand writes followed by high re-reads
- if workload benefits from frequent read-aheads
- if workload requires versioning - restore of point-in-time
- if workload is sensitive to write latencies
- if workload has large io-sizes writes and may trigger flushes
- if workload has intense bursts of writes

Conclusion

- **Changing Storage world**
  - Value shifting to $ / IOPs
  - Separation of Cold / Hot Data
  - Eventually consistent
  - Point in time restore
- **Chose Optimal write-back strategy**
  - Better write-bandwidth
  - Higher resiliency
  - Customized restore

Thank you!

Mahesh Khatpe, Pradeep Gothoskar

mahesh.khatpe@veritas.com

pradeep.gothoskar@veritas.com

Q&A