Agentless Cloud-wide Monitoring of Virtual Disk State

Wolfgang Richter

wolf@cs.cmu.edu

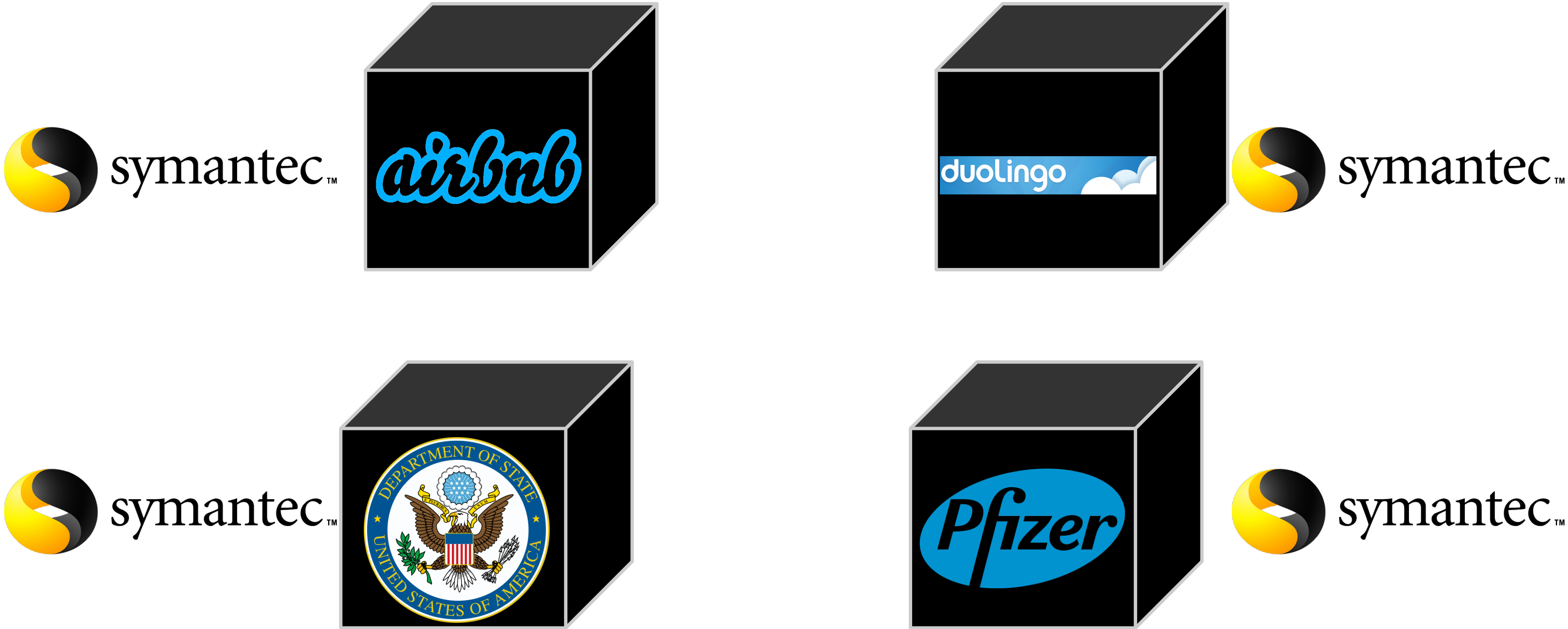
Monitoring is Broken

ec2-start-instance

EventTracker
Logging In Depth

Coupling Policy with Mechanism: CVE-2012-0493

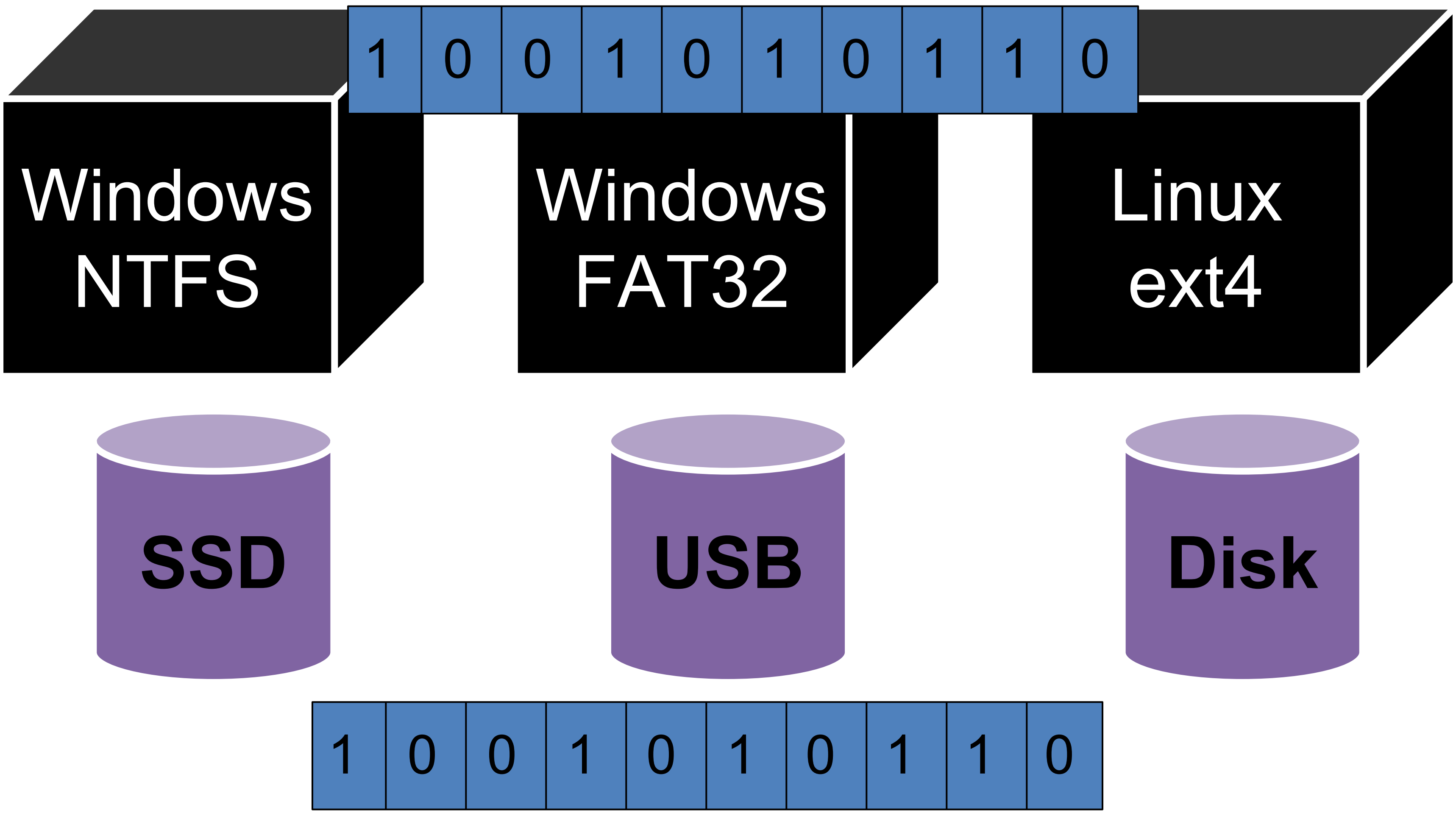
Symantec Endpoint Protection ... does not properly perform bounds checks of the contents of CAB archives, which allows remote attackers to … execute arbitrary code via a crafted file.



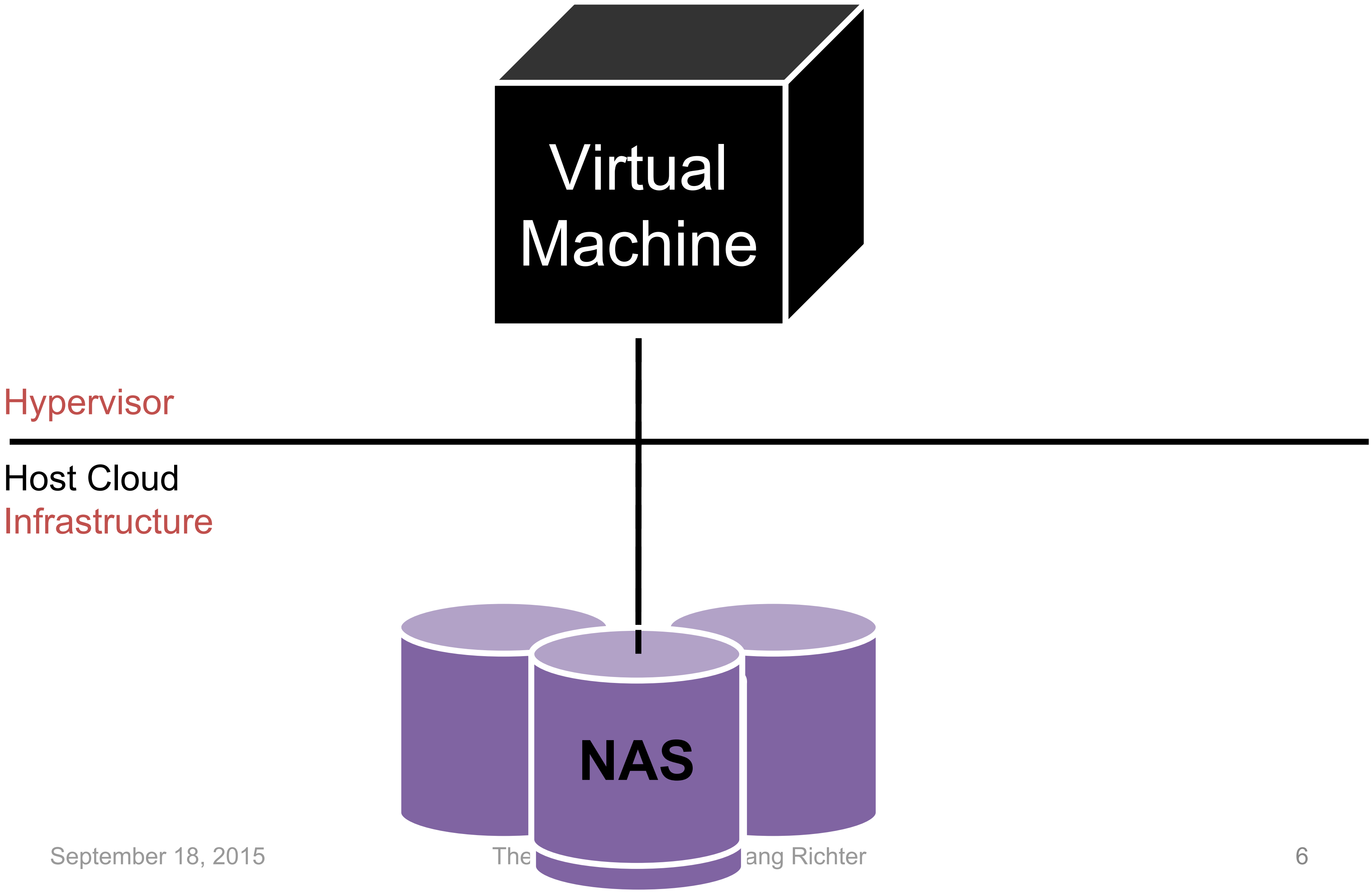
Cloud
Customers

Cloud
Providers

special-talk.pptx



Modern Clouds



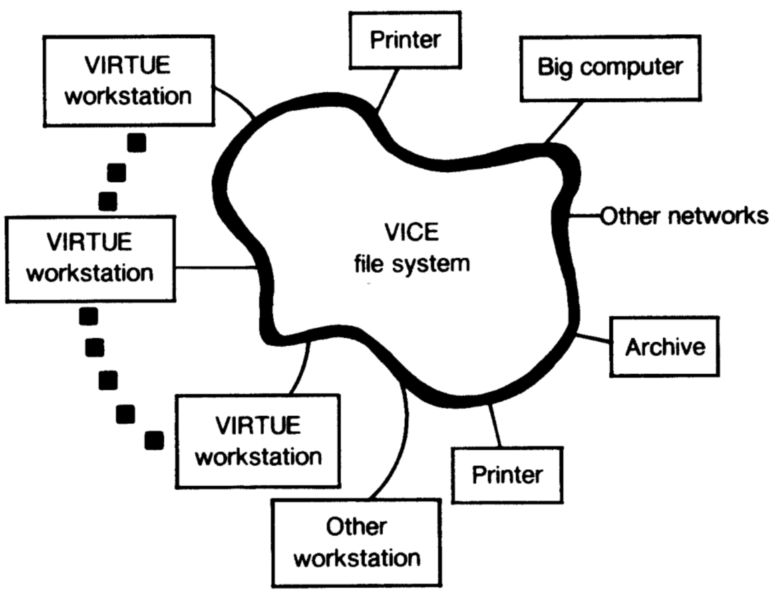
How to fix Monolithic Systems?

Distributed File Systems

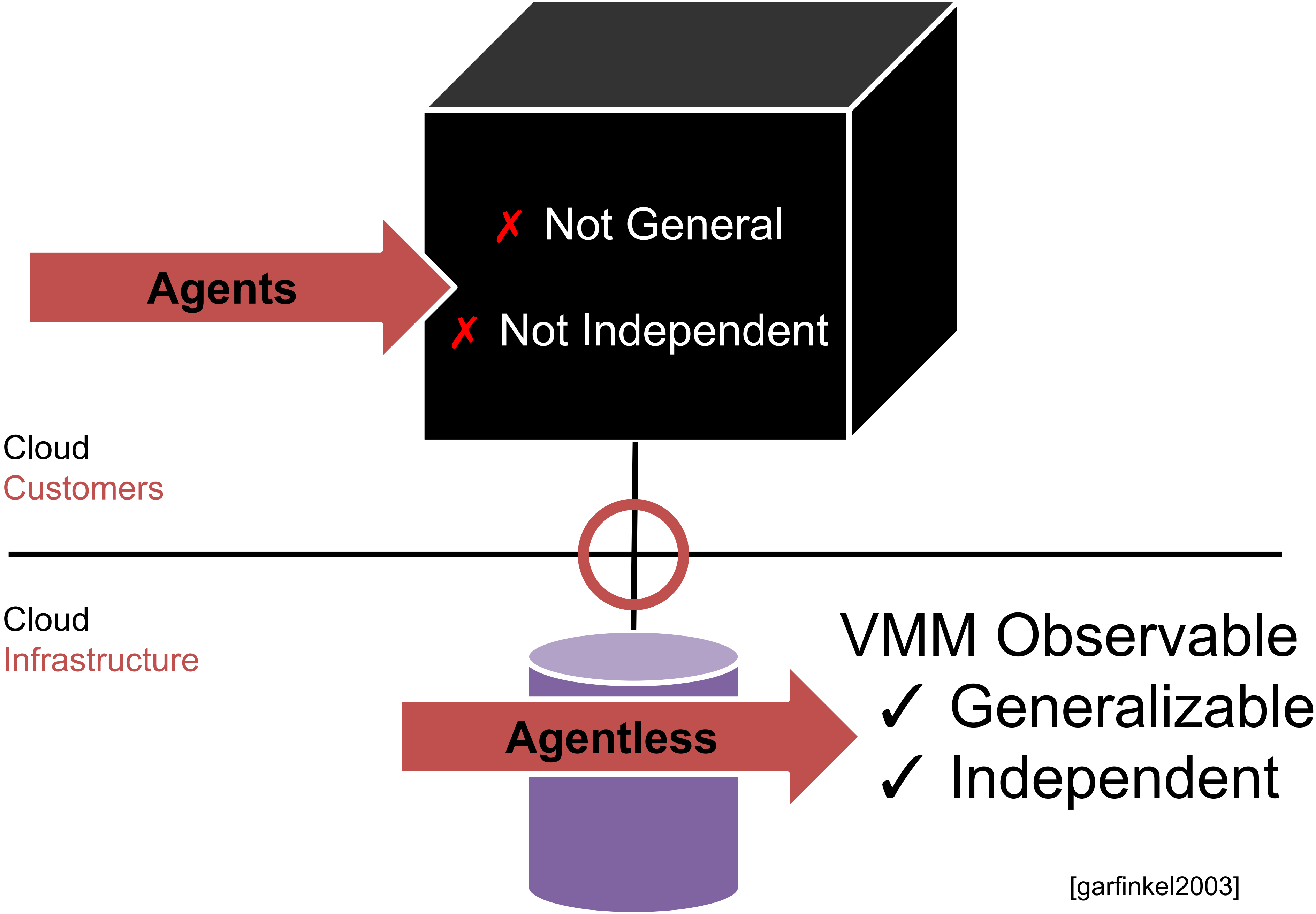
- Guest Support
- Per-OS Implementation
- Tightly Coupled
- Still Monolithic

Smarter Infrastructure

- Zero Configuration
- Generalizable Interface
- Loosely Coupled
- Separates Policy and Mechanism

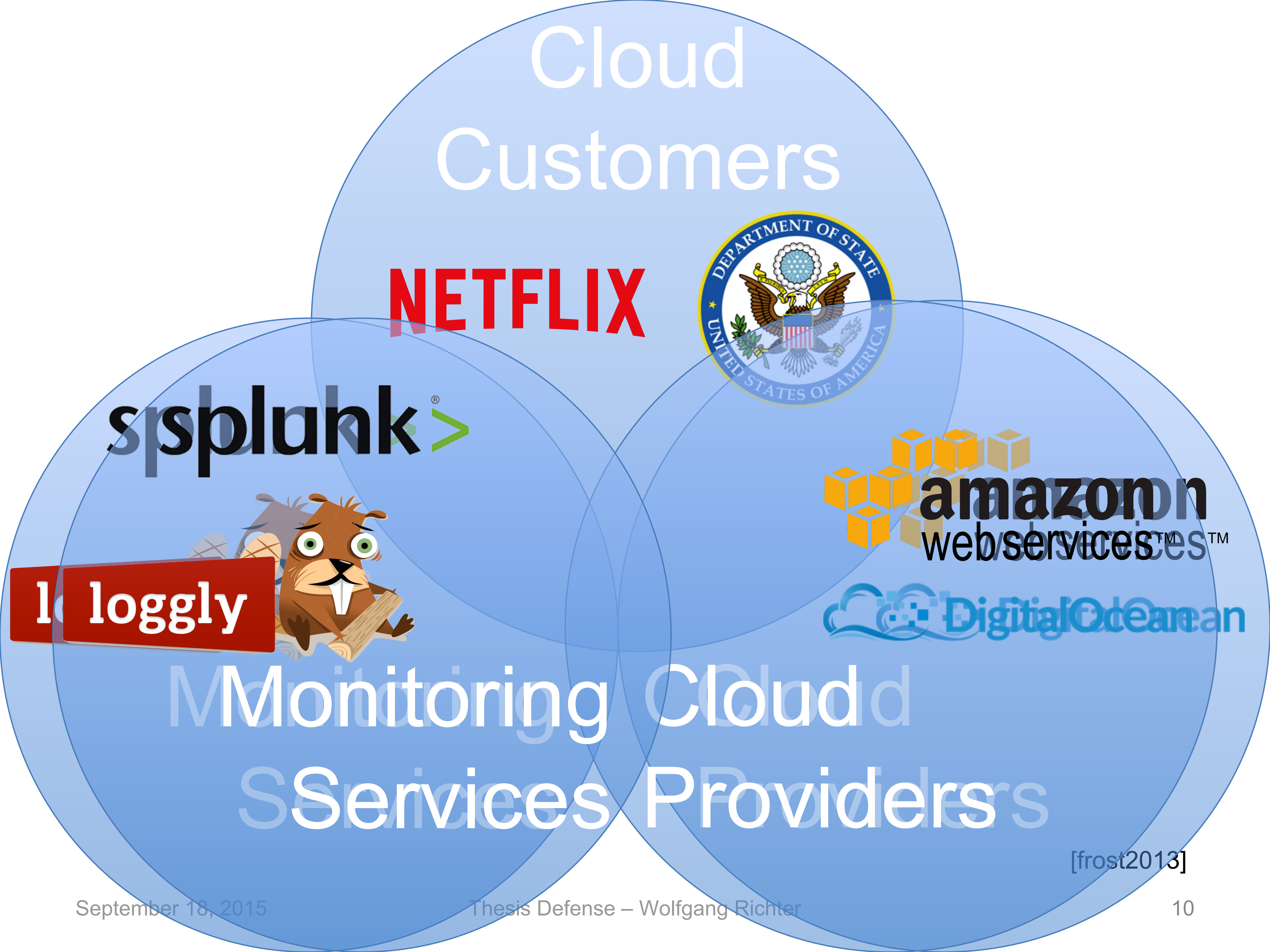


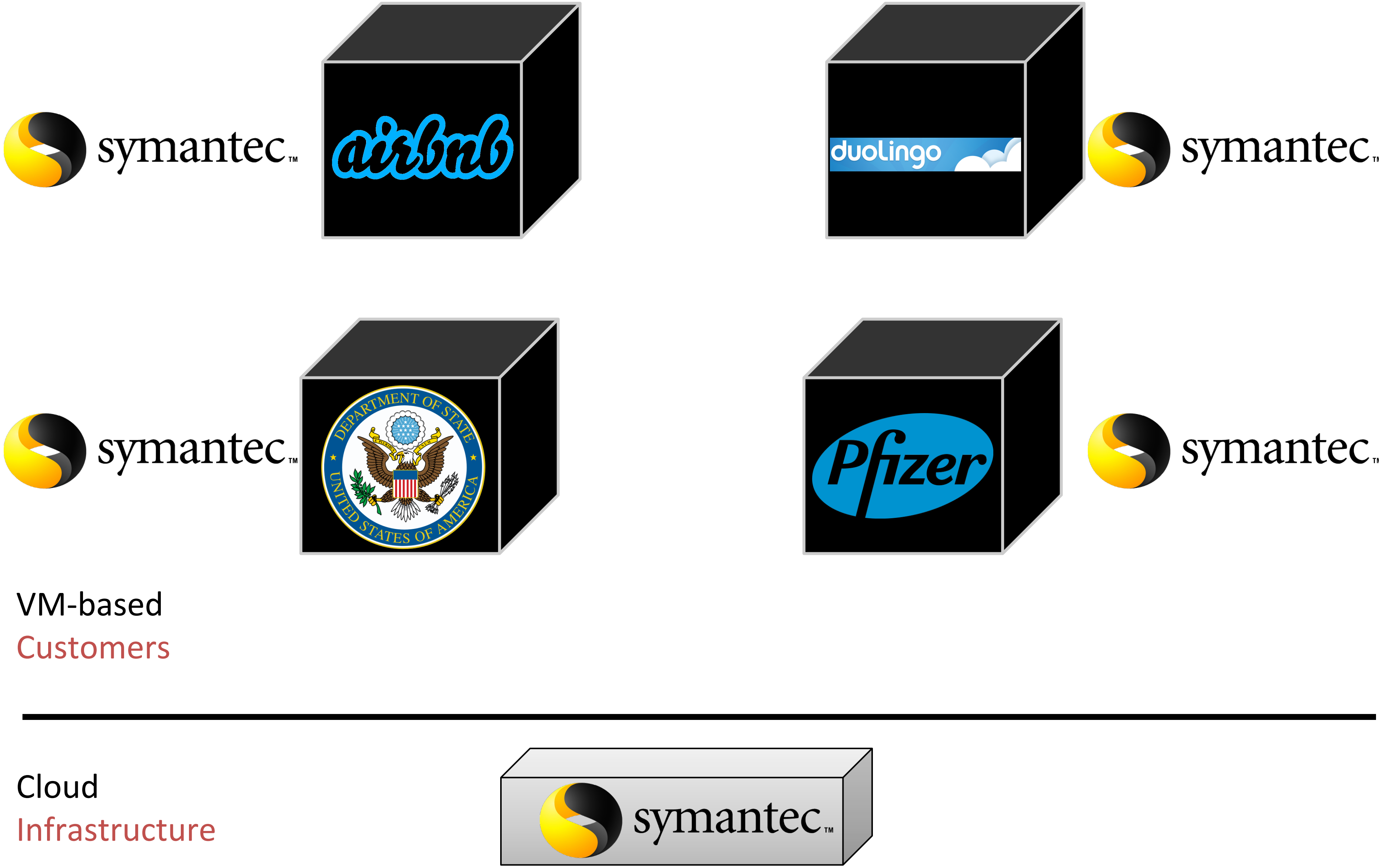
[morris1986]


✗ Not General
✗ Not Independent
Agents
Cloud
Customers
Cloud
Infrastructure
Agentless
VMM Observable
✓ Generalizable
✓ Independent
[garfinkel2003]

Agentless Monitoring of Disk State

- Stronger security guarantees
- Stronger correctness guarantees
- Enables Generalizability Across
  - OS
  - Application
  - Runtime environment (libraries, configuration)
  - Versions (OS, library, application, configuration)
- With modest infrastructure modifications

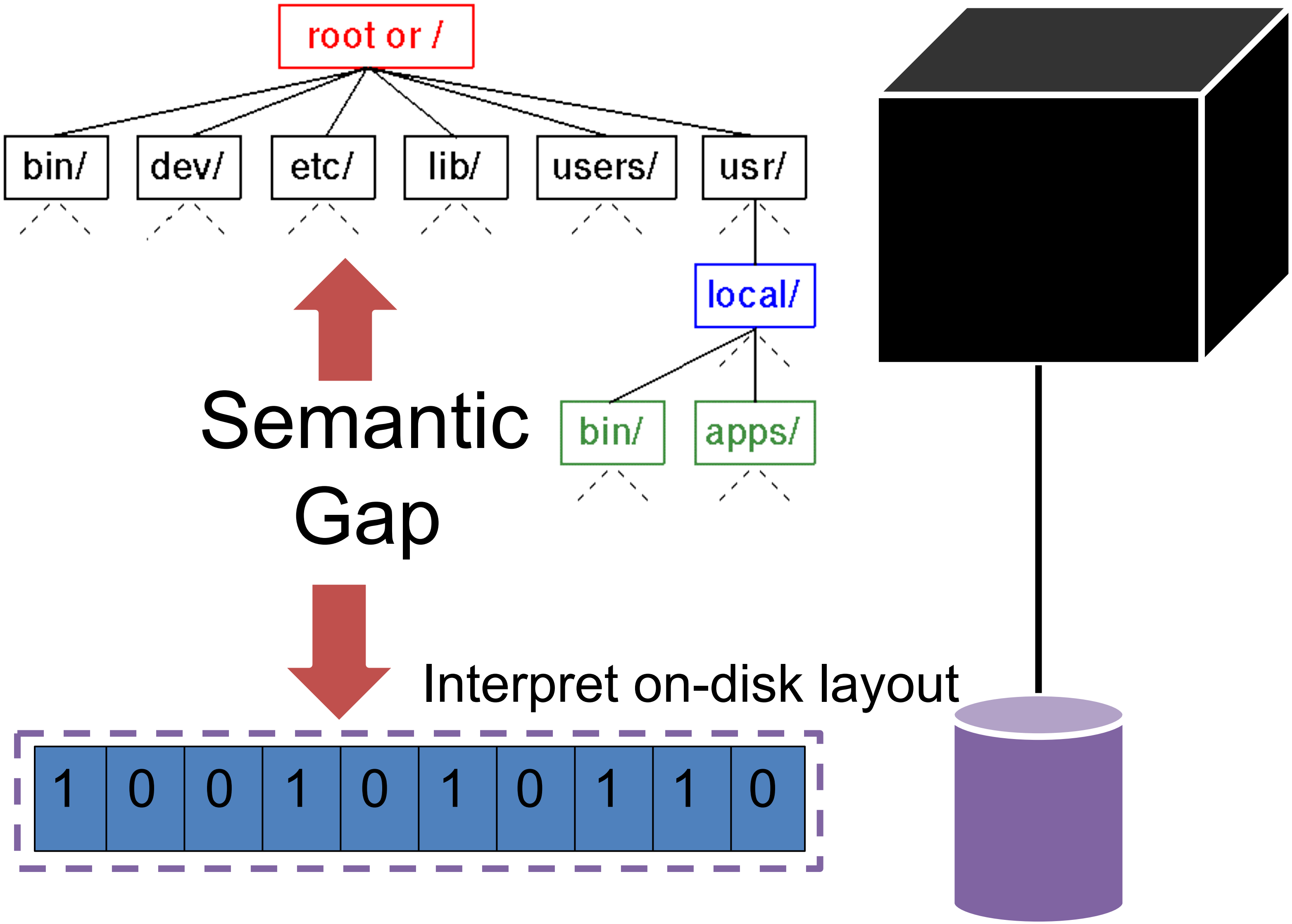

Cloud Customers
NETFLIX
DEPARTMENT OF STATE
UNITED STATES OF AMERICA
splunk
loggly
amazon web services™
DigitalOcean
Monitoring Services
Cloud Providers
[frost2013]


symantec
airbnb
duolingo
symantec
symantec
DEPARTMENT OF STATE
UNITED STATES OF AMERICA
Pfizer
symantec
VM-based
Customers
Cloud
Infrastructure
symantec

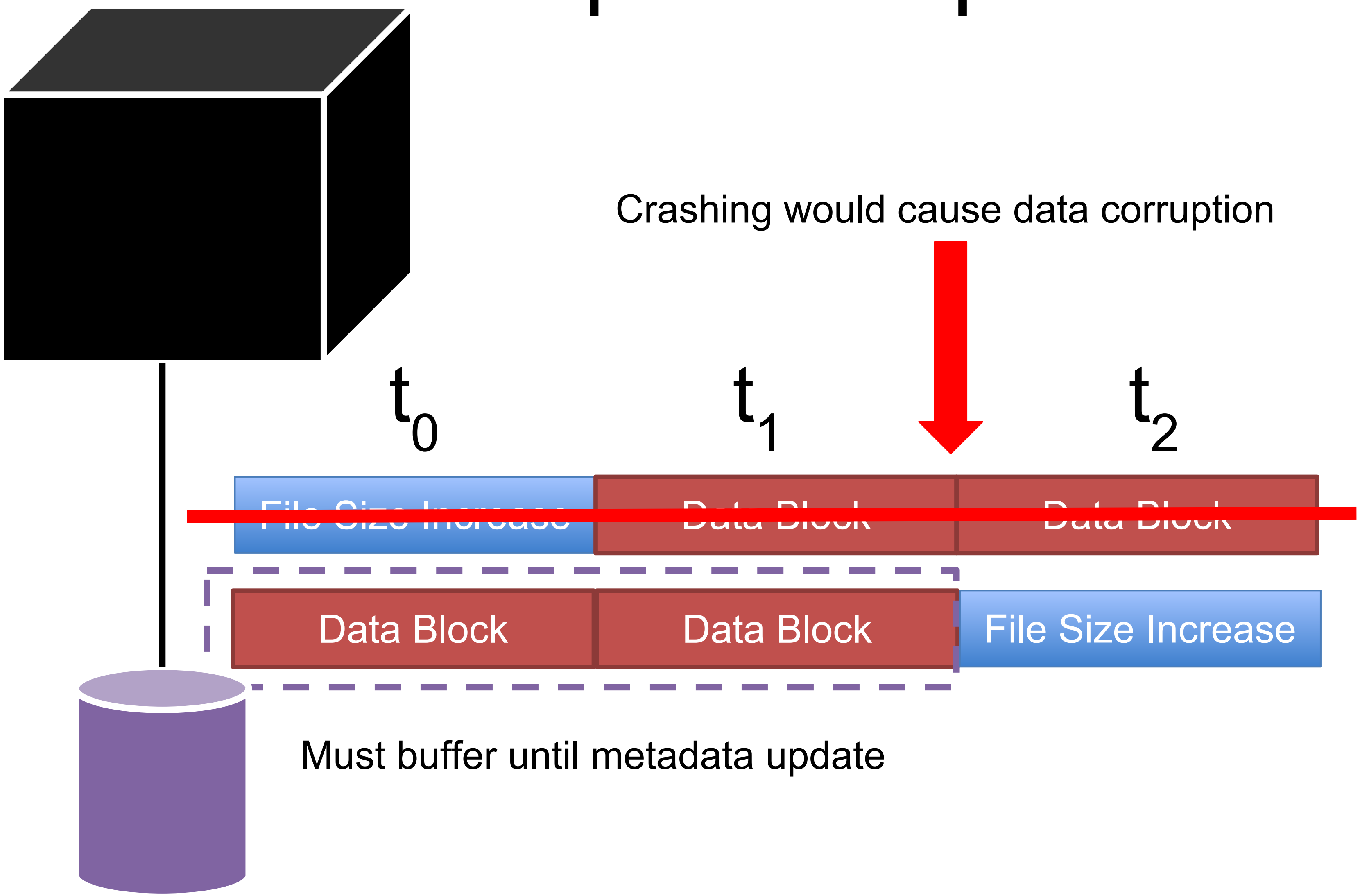
Outline

- Challenges
- Mechanism and Interfaces
  - Distributed Streaming Virtual Machine Introspection
  - `/cloud`
  - `cloud-inotify`
  - `/cloud-history`
- Summary and Conclusion

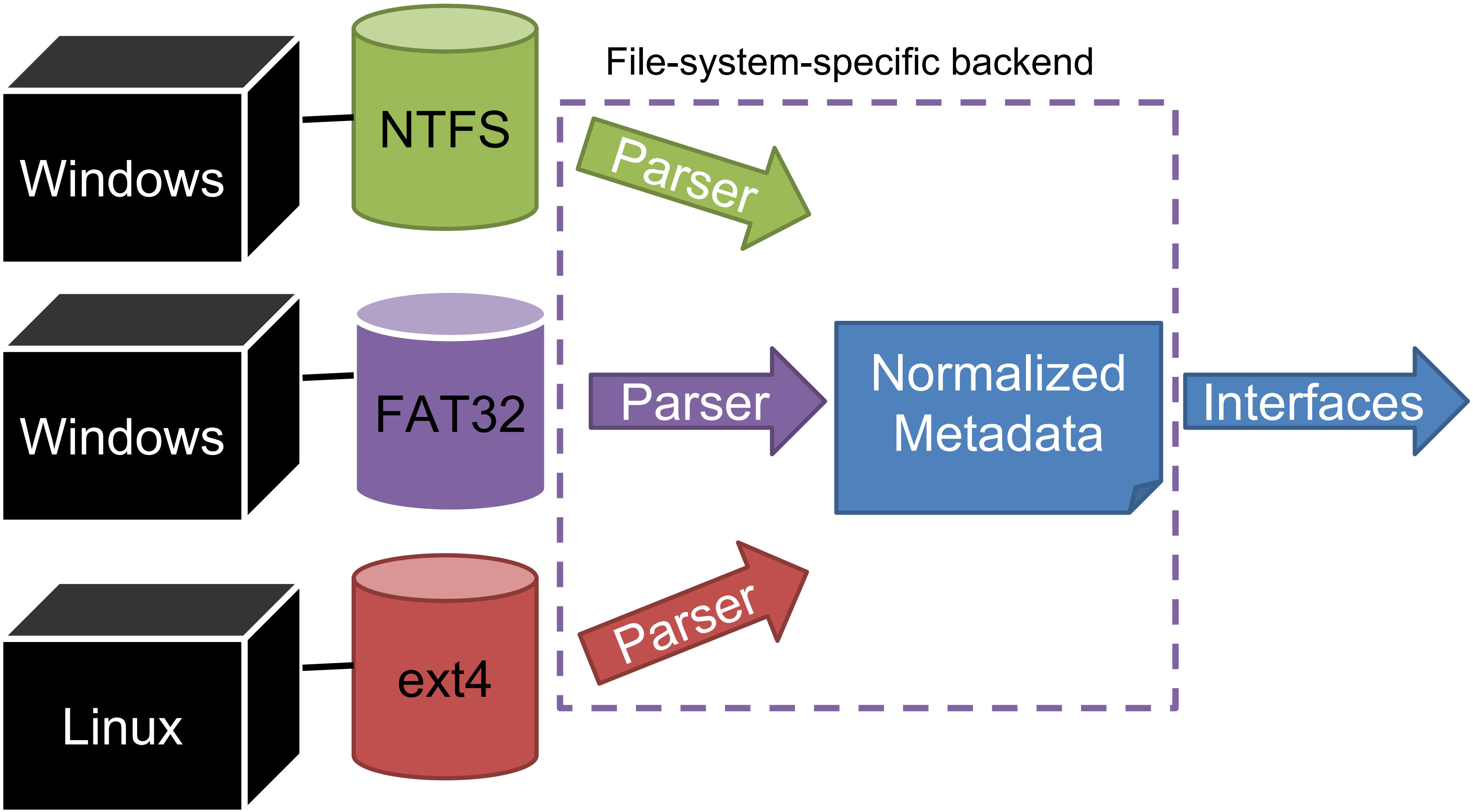
The Semantic Gap



Temporal Gap



Achieving Generality



Bounded Overhead

- Latency-completeness-performance tradeoff
  - Capturing every write is costly
  - Too much buffering hurts latency

- Must tolerate loss of writes
  - Extreme: detaching and re-attaching

Select Related Work

System	Semantic	Temporal	General	Bounded	Scalable
VMI, Garfinkel, 2003	✓	✓	✓	✗	✗
Maitland, Benninger, 2012	✓	✓	✗	✗	✗
File-aBLS, Zhang, 2006	✓	✓	✗	✗	✗
SDS, Sivathanu, 2003	✓	✓	✓	✗	✗

Outline

- Challenges
- Mechanism and Interfaces
  - Distributed Streaming Virtual Machine Introspection
  - `/cloud`
  - `cloud-inotify`
  - `/cloud-history`
- Summary and Conclusion

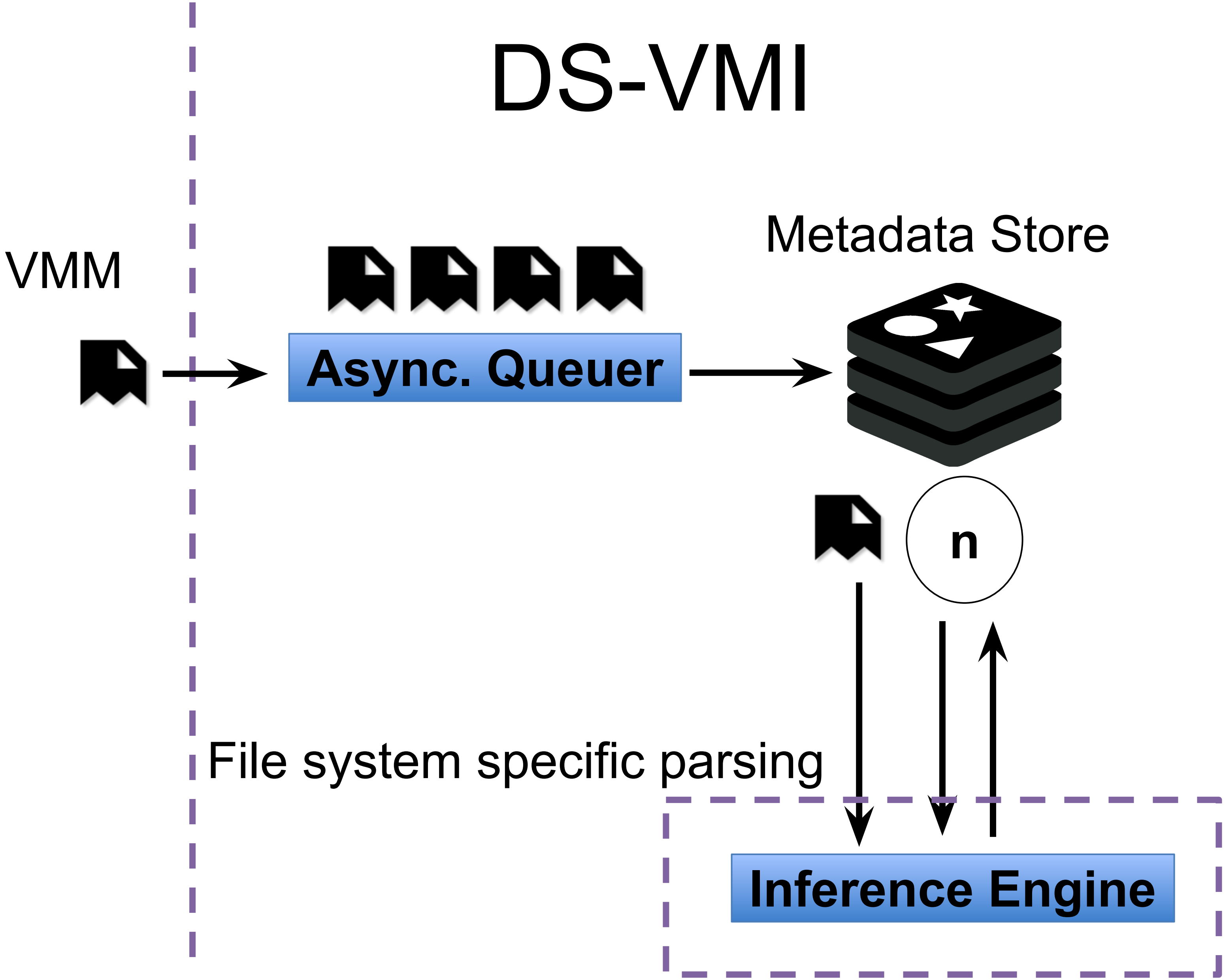
`/cloud` | `cloud-inotify` | `/cloud-history`

Distributed Streaming Virtual Machine Introspection (DS-VMI)

/cloud | cloud-inotify | /cloud-history

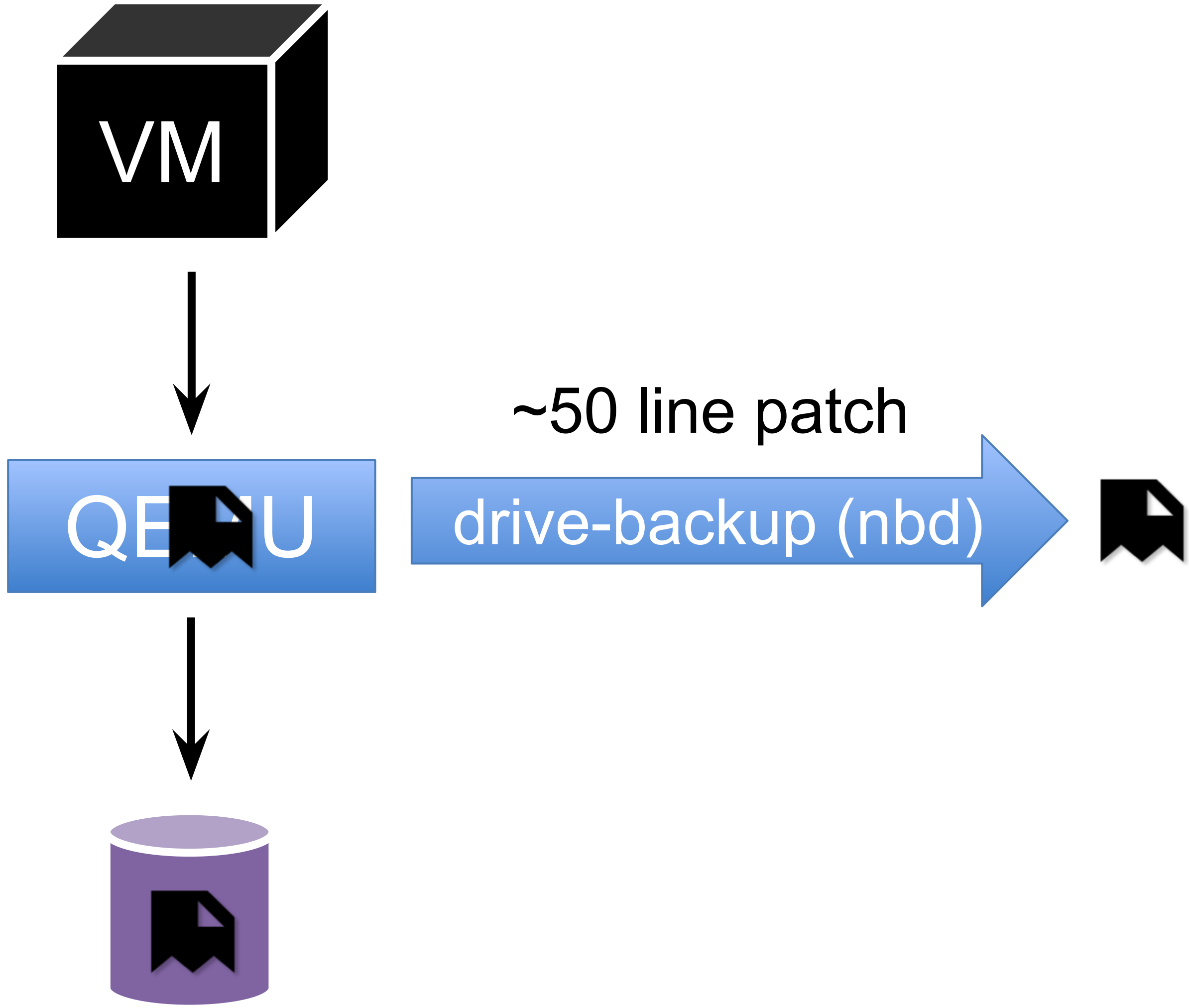
Distributed Streaming Virtual Machine Introspection (DS-VMI)

DS-VMI

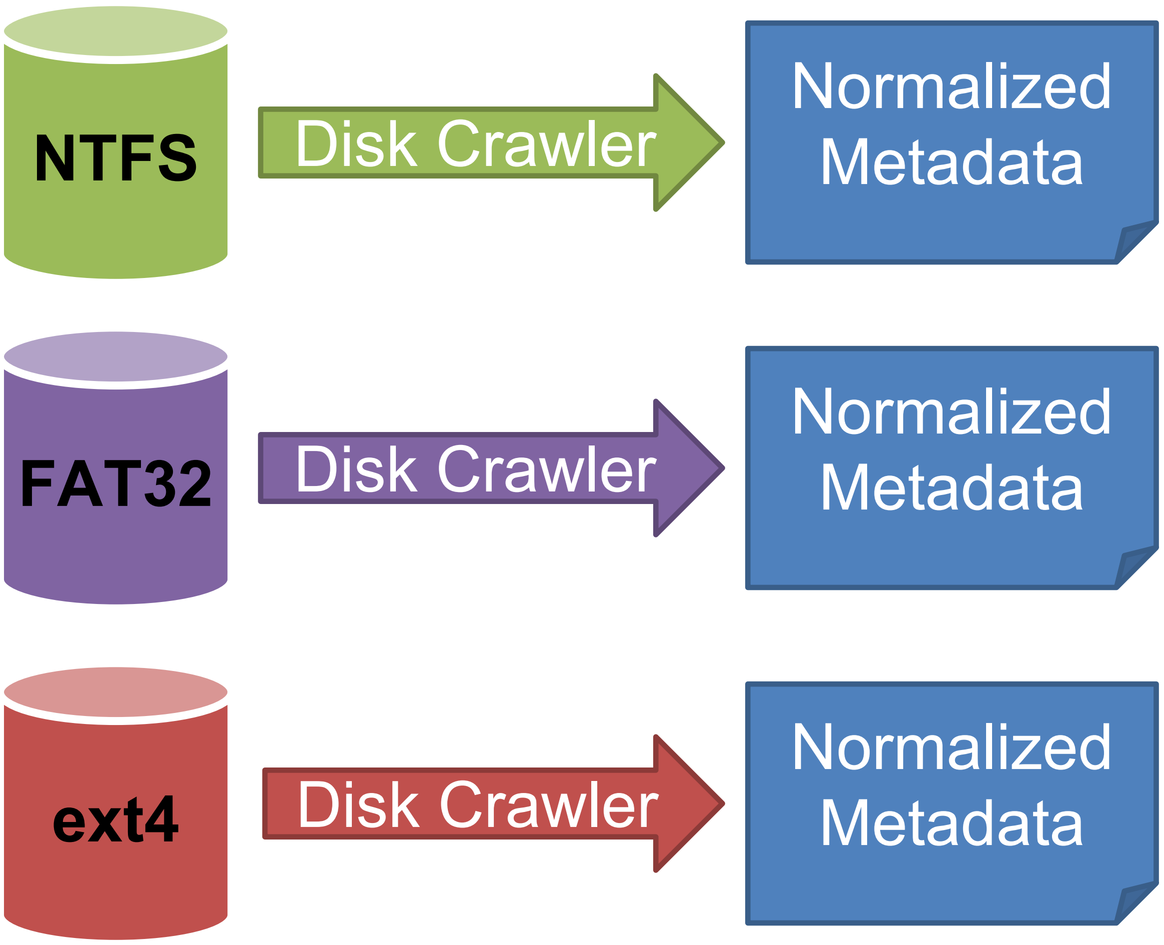


VMM
Metadata Store
Async. Queuer
n
File system specific parsing
Inference Engine

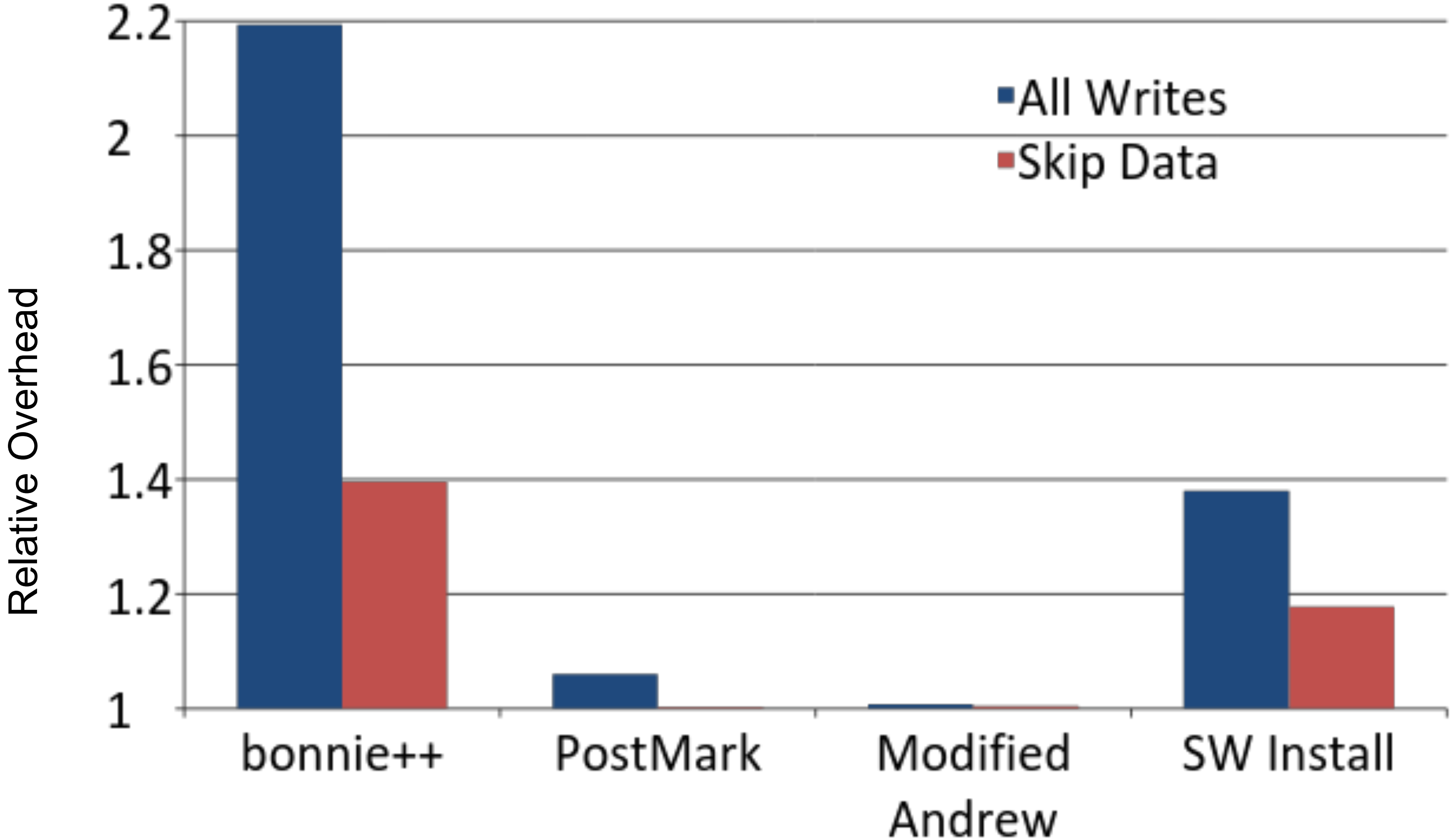
Tapping the Disk Write Stream



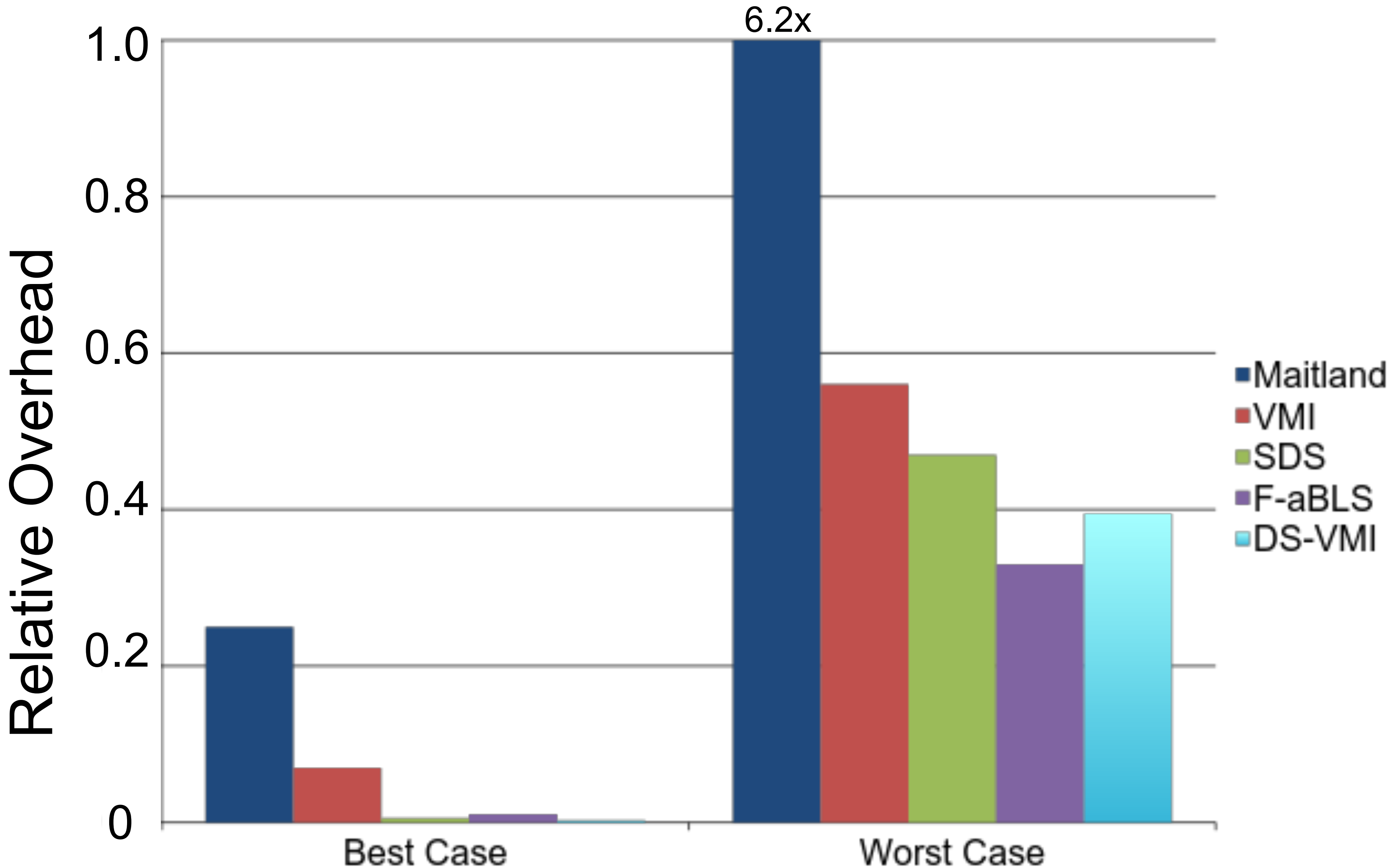
Bootstrapping



DS-VMI Overhead on Running VM



[richter2014]


6.2x
1.0
0.8
0.6
0.4
0.2
0
Relative Overhead
Maitland
VMI
SDS
F-aBLS
DS-VMI
Best Case
Worst Case

/cloud | cloud-inotify | /cloud-history

Distributed Streaming Virtual Machine Introspection (DS-VMI)

/cloud

Eventual consistency

Legacy FS interface

Batch-based

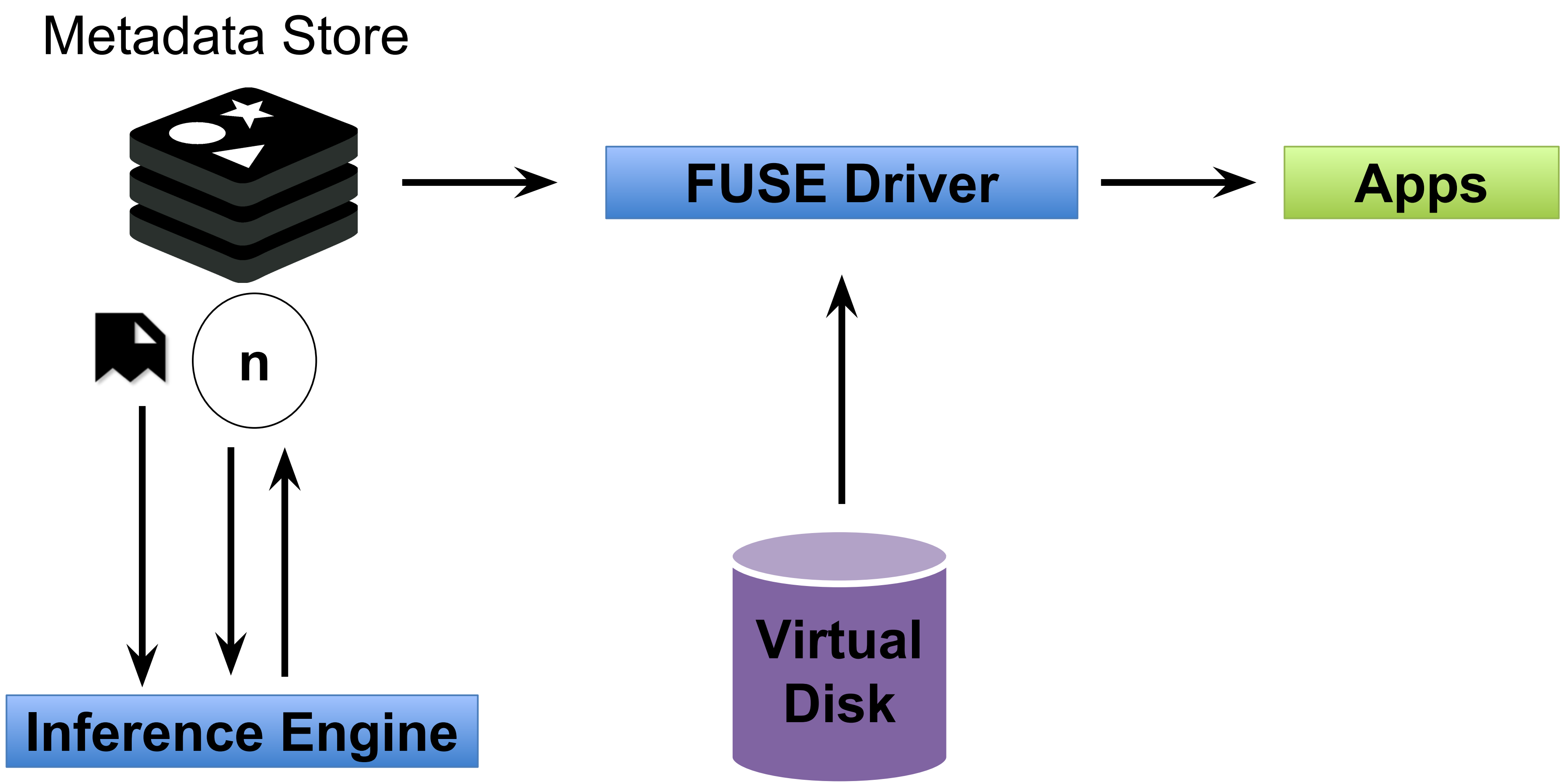
Legacy/batch-based apps: /cloud/host/vm/path

```
find /cloud/*/*/lib  \
      -maxdepth 0 \
      -not \
      -perm 755
```

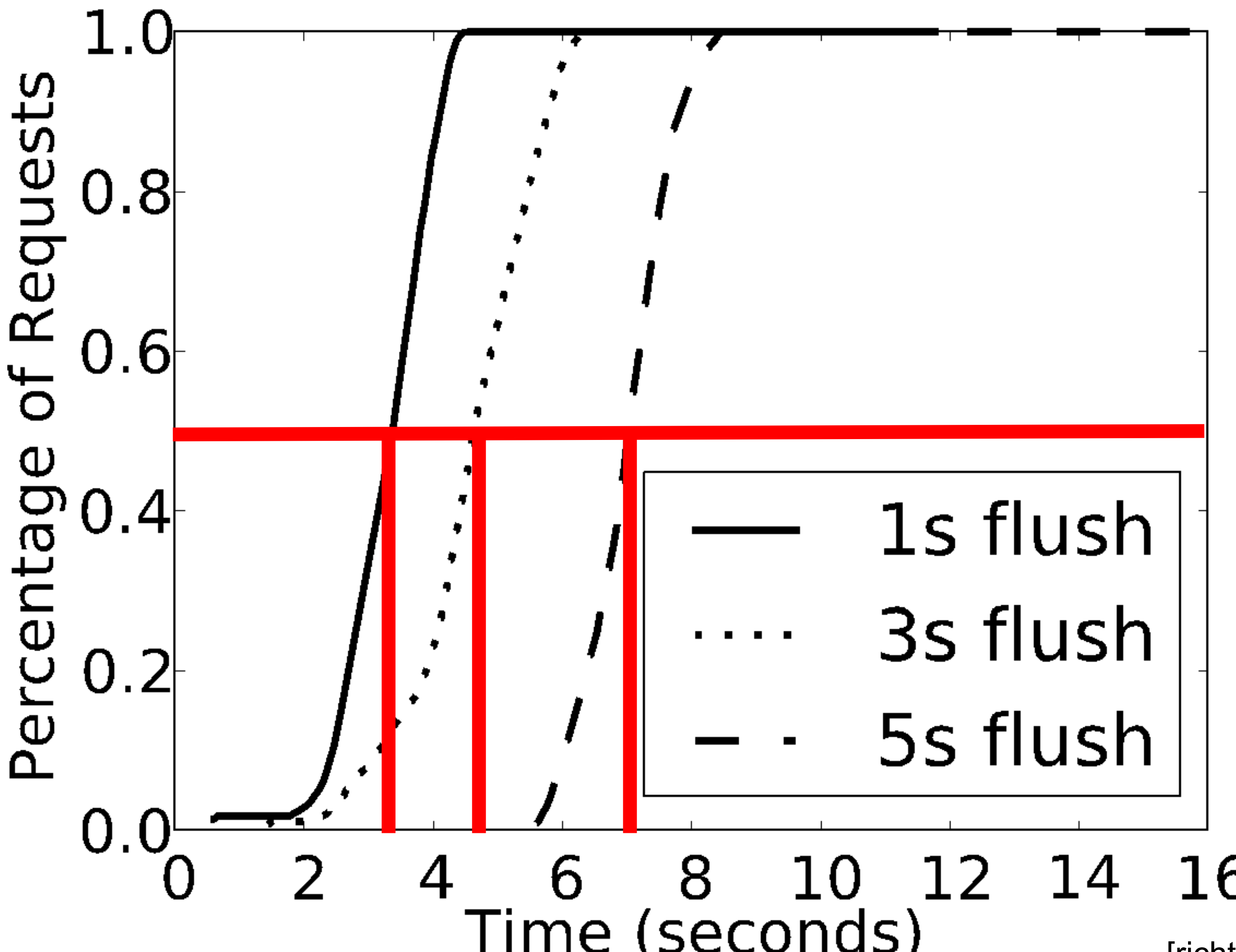
Google

On all hosts check permissions of /lib inside every VM instance.

/cloud Architecture



Latency – Guest Syncs



[richter2014]

/cloud | cloud-inotify | /cloud-history

Distributed Streaming Virtual Machine Introspection (DS-VMI)

`cloud-inotify`

Strong consistency

Publish-subscribe

Event-driven

Subscription format: <host>:<VM>:<path>

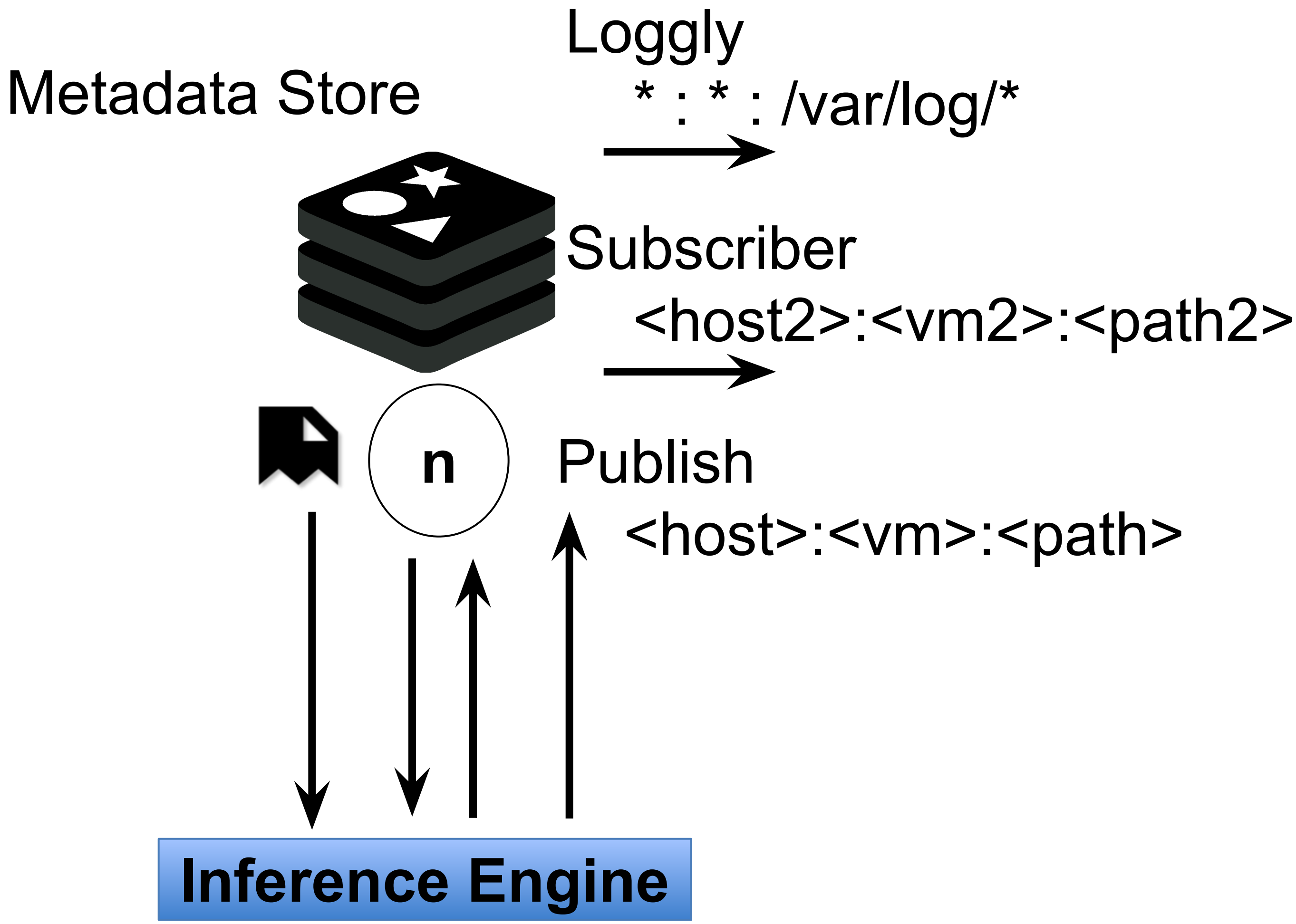
gs9671:bg1:/var/log/*

On host gs9671

monitor all files
under file system
subtree /var/log/

in all VM’s in group bg1.

cloud-inotify Architecture



OpenStack “Live” Demo

Bedford Springs

Internet

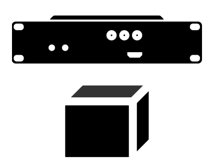
WebSocket Proxy

CMU

OpenStack

`cloud-inotify`

Distributed Streaming Virtual Machine Introspection (DS-VMI)



[pdlretreat2014]

/cloud	cloud-inotify	/cloud-history

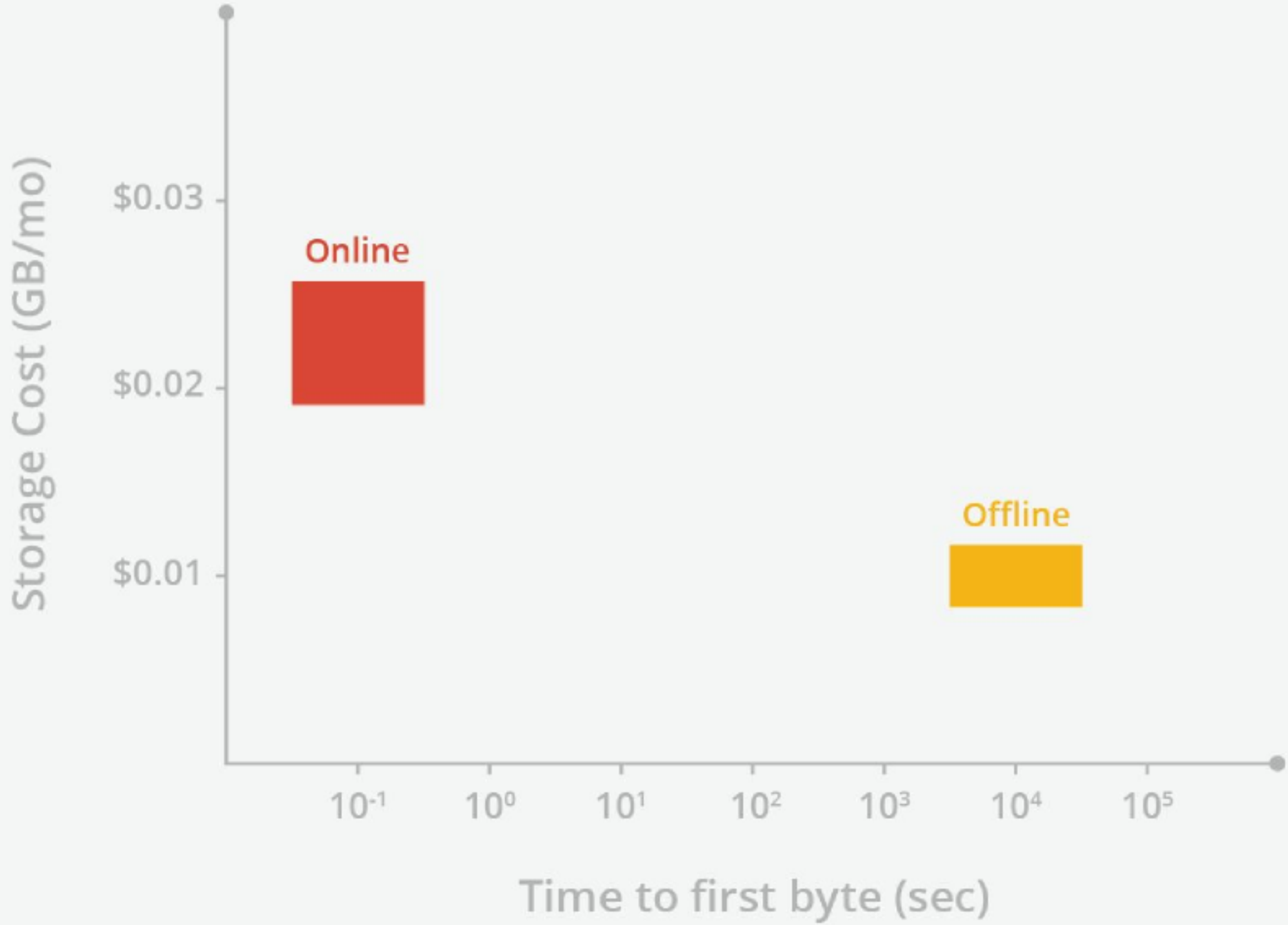
Distributed Streaming Virtual Machine Introspection (DS-VMI)

CVE-2014-0160: Heartbleed

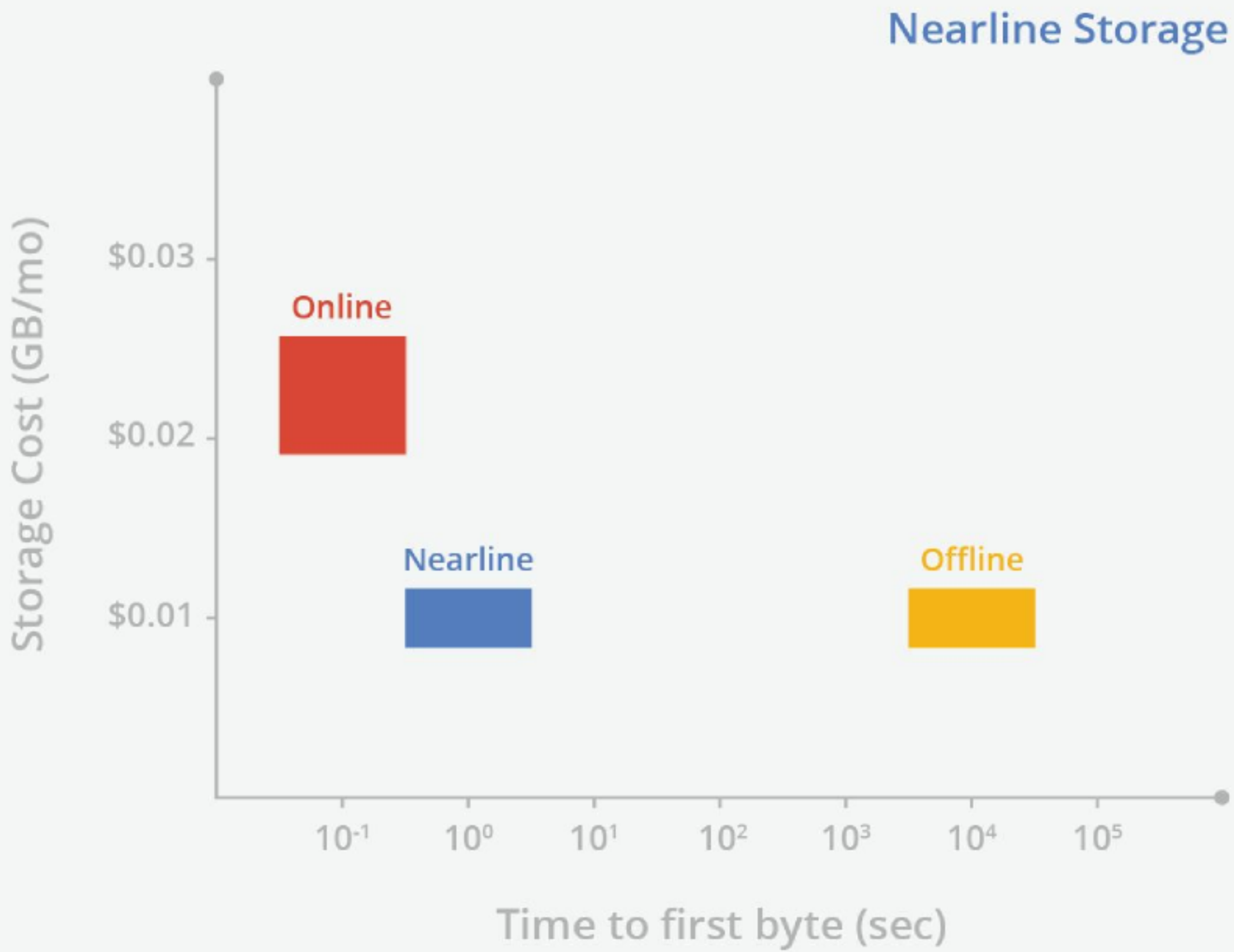
- Untraceable exploit
- In the wild 2 years
  - OpenSSL 1.0.1 - 1.0.1f
  - March 2012 - April 2014
- Leaks server memory

Are my systems vulnerable?
Are any customers affected?

Online vs. Offline



[google2015]


Nearline Storage
Storage Cost (GB/mo)
$0.03
$0.02
$0.01
Online
Nearline
Offline
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
10^{4}
10^{5}
Time to first byte (sec)

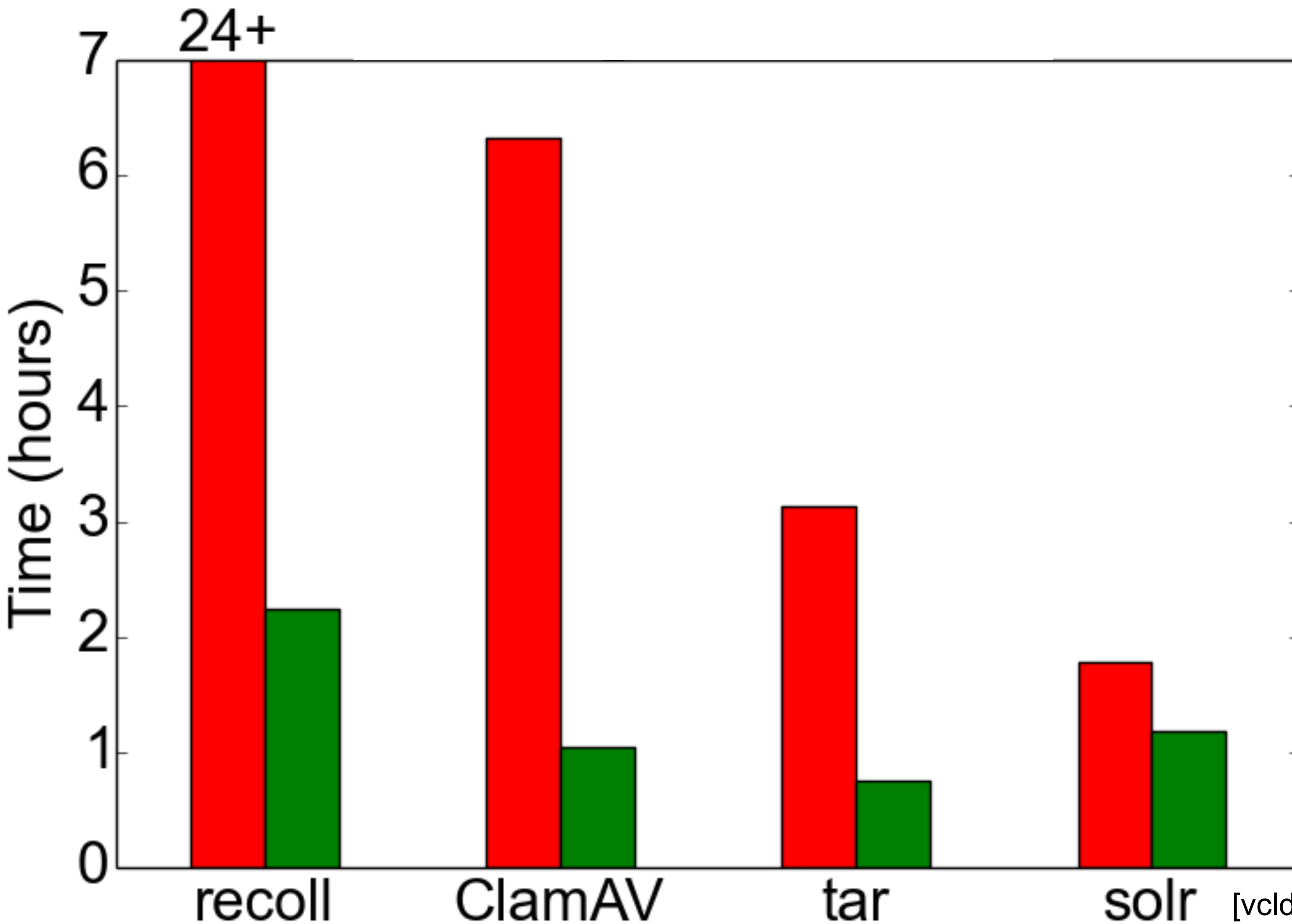
[google2015]

/cloud-history

Indexed Log-structure

`open('f', O_WRONLY) = 3`	MD_{atime}
`write(3, "test", 4) = 4`	w[0]
`lseek(3, 4096, SEEK_SET)` `write(3, "test", 4) = 4`	w[4096]
`close(3) = 0`	MD_{atime} MD_{mtime} MD_{size}

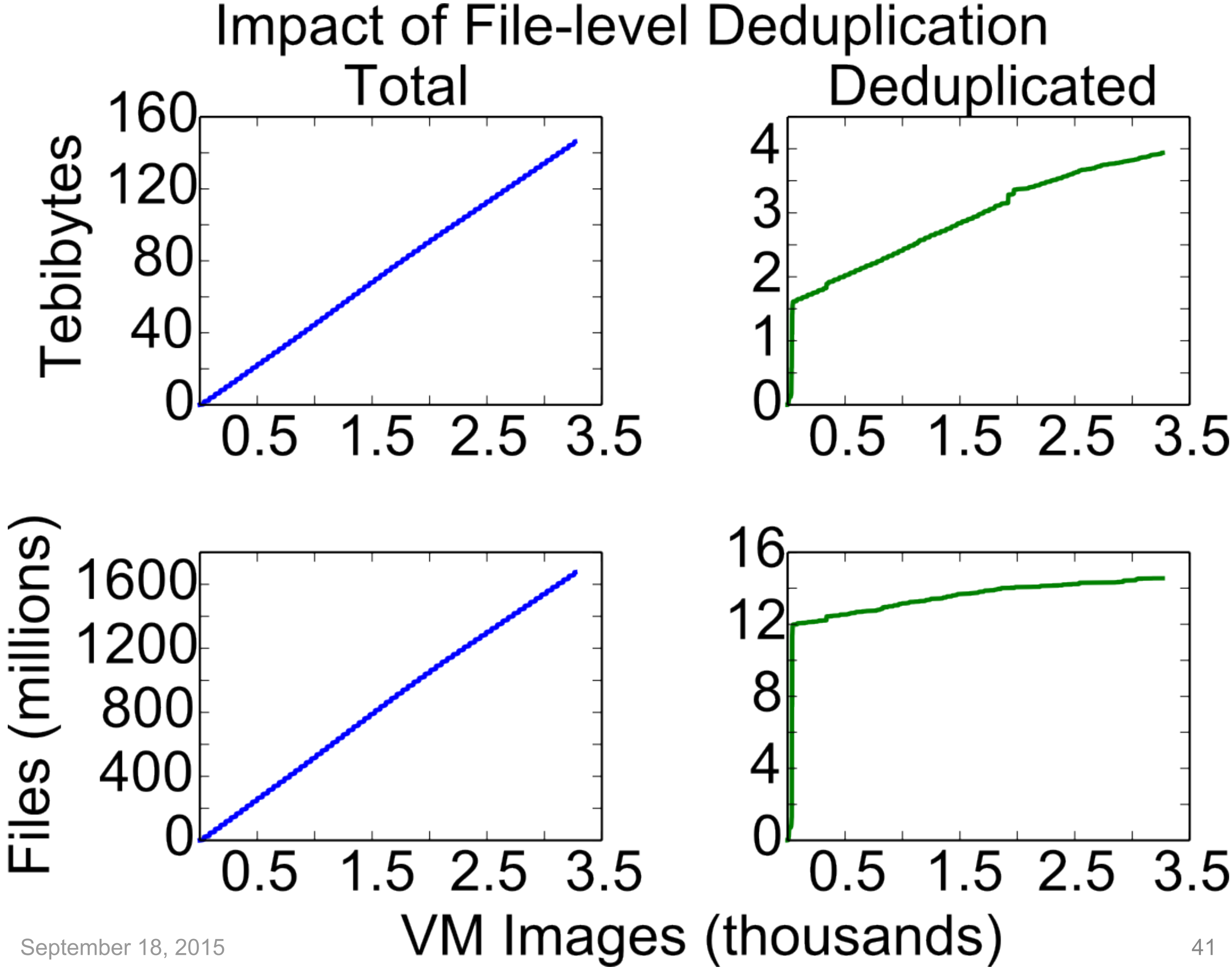
Effect of File-level Deduplication on Indexing



[vcldataset]

Deltaic Backup Study

- 58 hosts, ~1-year timeframe
- 3,267 file system snapshots
- 1.676 billion referenced files
- 146 TiB of crawled bytes


Impact of File-level Deduplication
Total
Deduplicated
Tebibytes
160
120
80
40
0
4
3
2
1
0
Files (millions)
1600
1200
800
400
0
16
12
8
4
0
0.5
1.5
2.5
3.5
VM Images (thousands)

tripwire splunk> Dropbox

ClamAV loggly veeam Modern Data Protection

/cloud	cloud-inotify	/cloud-history

File-level deduplication

Distributed Streaming Virtual Machine Introspection (DS-VMI)

Desired Hash Properties

Quick to re-compute for random writes

DS-VMI works with a stream of writes

No extra bytes from disk required

Can’t rely on virtual disk, or reconstruction

Collision Resistant

For correctness

Compact

Network synchronization

Traditional Hashing?

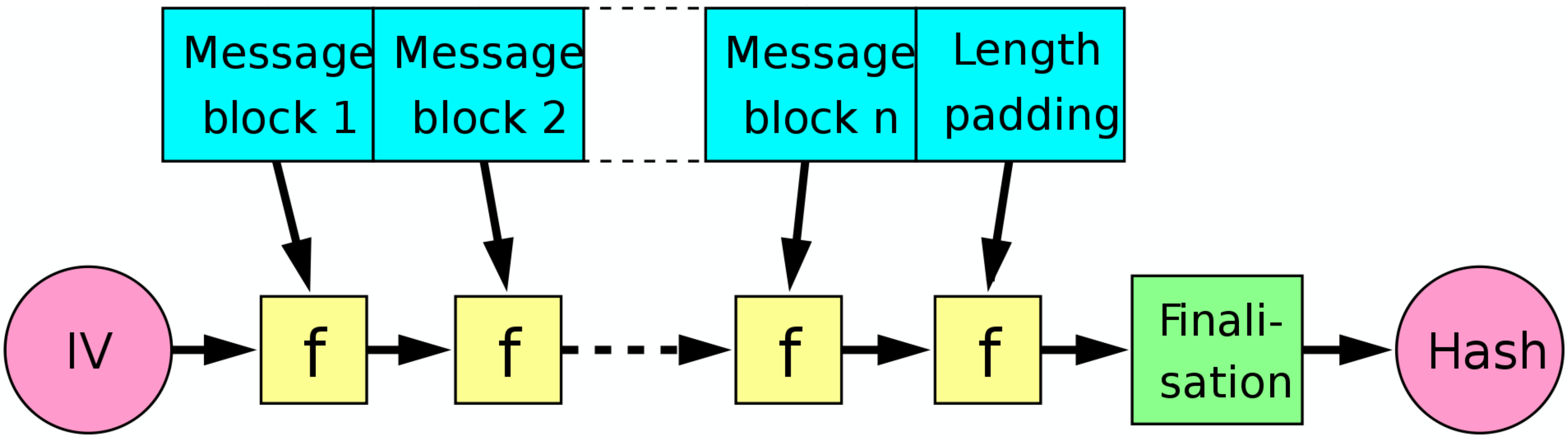
Supports rapid recomputation of whole-file hash for append-only operations

Normal C API (SHA-3, NIST):

```
Update(hashState *state,
       const BitSequence *data,
       DataLength datalen);
```

[nist]

Merkle-Damgård



[wikipedia, damgård1990]

`open('f', O_WRONLY) = 3`	MD_{atime}
`write(3, "test", 4) = 4`	w[0]
`lseek(3, 4096, SEEK_SET)` `write(3, "test", 4) = 4`	w[4096]
`close(3) = 0`	MD_{atime} | MD_{mtime} | MD_{size}

Incremental Hashing

Incremental

Efficient random updates

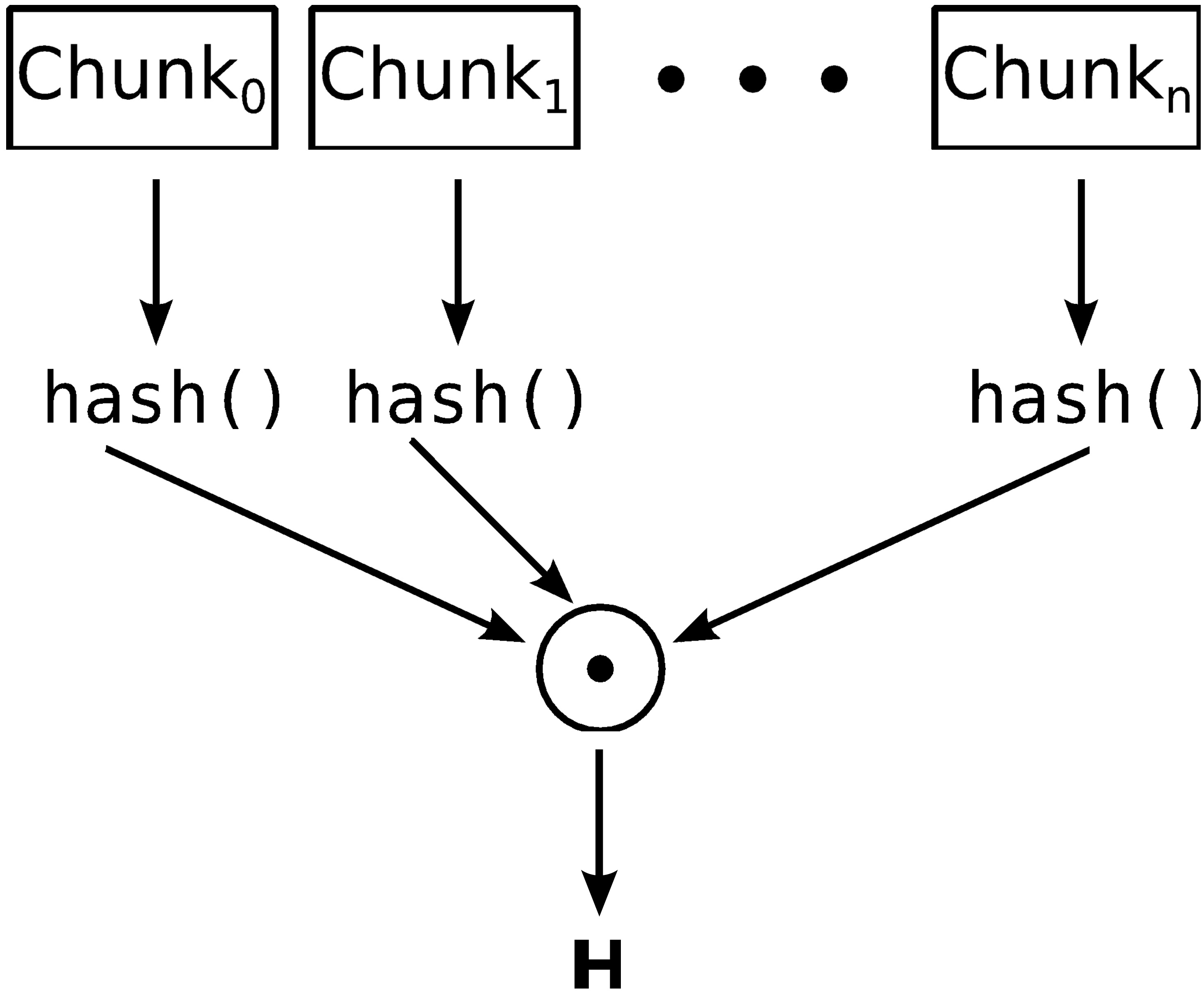
Collision-free

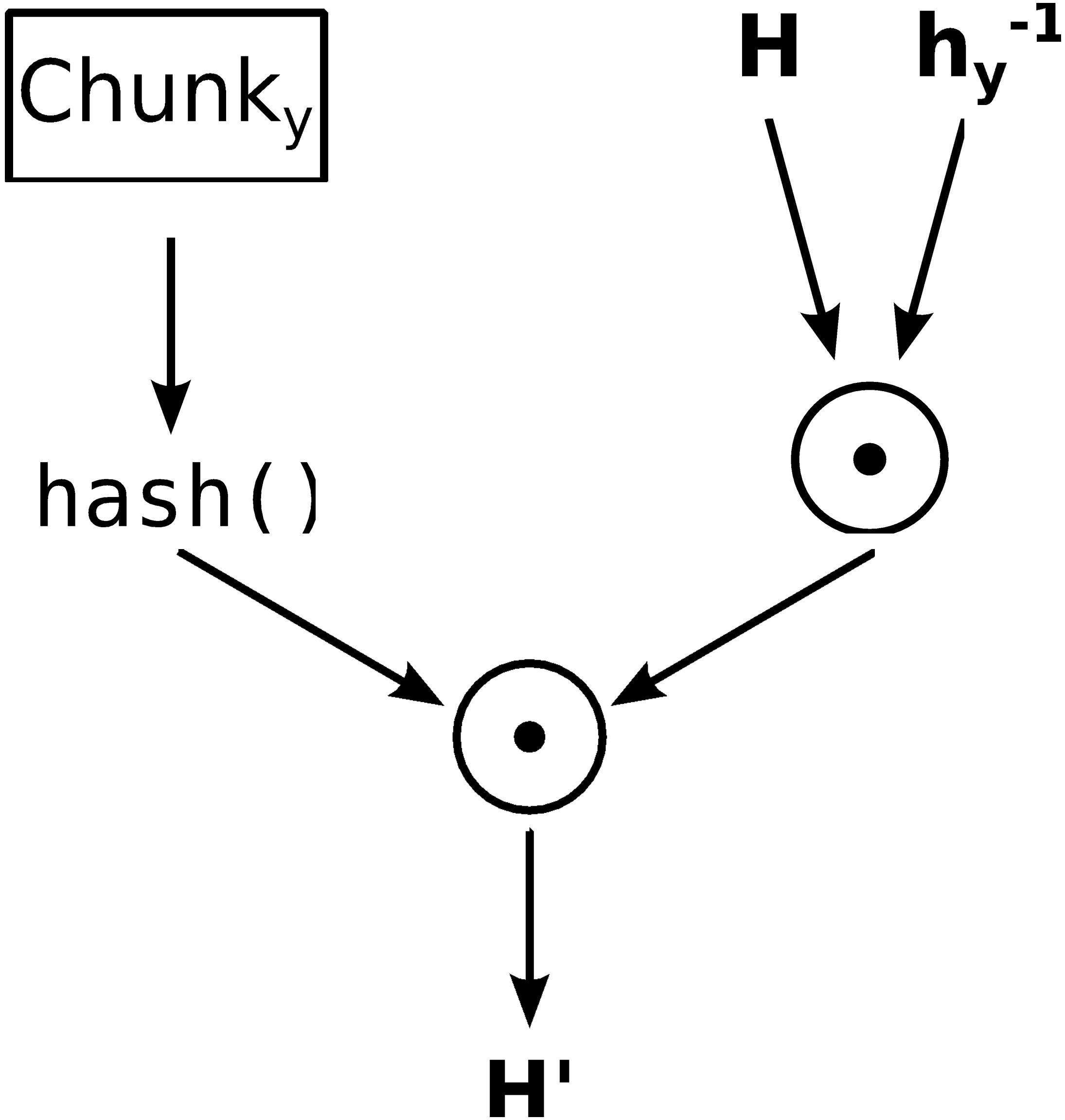
Cryptographically secure

Parallelizable

Faster than sequential

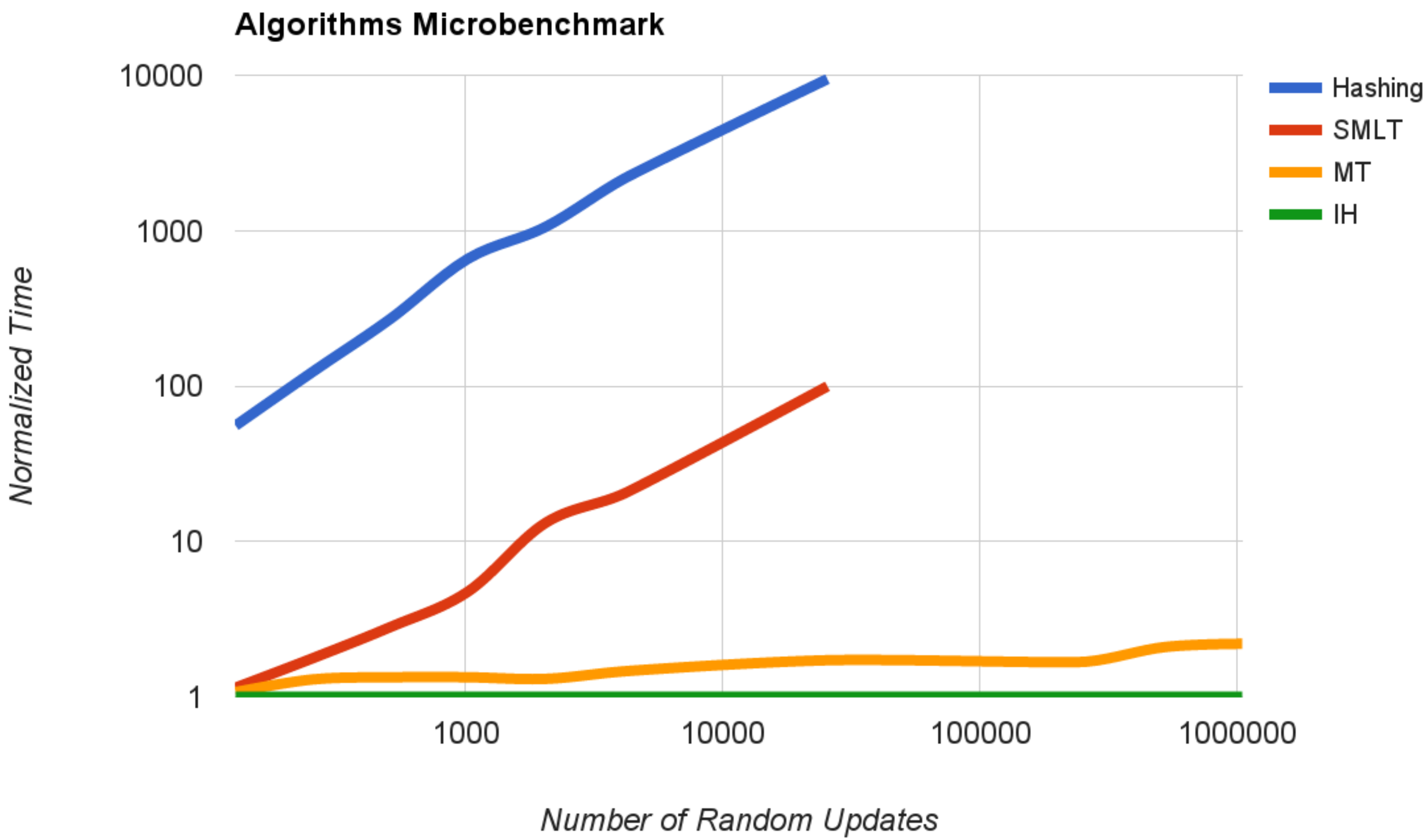
[bellare1997]


Chunk0
Chunk1
Chunkn
hash()
hash()
hash()
H


Chunk$_y$
hash()
H
h_y^{-1}
H'

Hashing Analysis

Operation	H	MT	SLMT	IH
Update (S)	$\mathbf{O}(\mathbf{1})$	$\mathrm{O}(\log_f N + 1)$	$\mathrm{O}(N)$	$\mathbf{O}(\mathbf{1})$
Update (R)	$\mathrm{O}(N)$	$\mathrm{O}(\log_f N + 1)$	$\mathrm{O}(N)$	$\mathbf{O}(\mathbf{1})$
Update (B)	$\mathrm{O}(N)$	$\mathrm{O}\left(\frac{fN'-1}{P(f-1)} + \lceil \log_f N' \rceil\right)$	$\mathrm{O}\left(\frac{N'+1}{P} + N\right)$	$\mathrm{O}\left(\frac{N'}{P} + \lceil \log_2 N' \rceil\right)$
Space	$\mathbf{O}(\mathbf{1})$	$\mathrm{O}\left(\frac{fN-1}{f-1}\right)$	$\mathrm{O}(N+1)$	$\mathrm{O}(N+1)$


Algorithms Microbenchmark
10000
1000
100
10
1
1000
10000
100000
1000000
Normalized Time
Number of Random Updates
Hashing
SMLT
MT
IH

Summary

`/cloud`	`cloud-inotify`	`/cloud-history`
File-level deduplication		
Distributed Streaming Virtual Machine Introspection (DS-VMI)		

Open Source, Apache v2.0 License
https://github.com/cmusatyalab/gammaray

Contact me for backup dataset (250 GiB database)

Citations - 1

[bellare1997] Bellare, Mihir and Micciancio, Daniele. A New Paradigm for Collision-Free Hashing: Incrementality at Reduced Cost. EUROCRYPT' 97.

[benninger2012] Benninger, C. and Neville, S.W. and Yazir, Y.O. and Matthews, C. and Coady, Y. Maitland: Lighter-Weight VM Introspection to Support Cyber-security in the Cloud. CLOUD' 12.

[cohen2010] Cohen, Jeff and Repantis, Thomas and McDermott, Sean and Smith, Scott and Wein, Joel. Keeping track of 70,000+ servers: the Akamai query system. LISA' 10.

[damgård1990] Ivan Bjerre Damgård. A Design Principle for Hash Functions. CRYPTO' 89.

[frost2013] Frost & Sullivan. Analysis of the SIEM and Log Management Market. 2013, http://goo.gl/Vup9ml.

[garfinkel2003] Garfinkel, Tal and Rosenblum, Mendel. A Virtual Machine Introspection Based Architecture for Intrusion Detection. NDSSS' 03.

[kufel2013] Kufel, L. Security Event Monitoring in a Distributed Systems Environment. 2013, IEEE Journal of Security and Privacy.

[nist] NIST. ANSI C Cryptographic API Profile for SHA-3 Candidate Algorithm Submissions. 2009, http://goo.gl/WsFCzp.

Citations - 2

[richter2011] Richter, Wolfgang and Ammons, Glenn and Harkes, Jan and Goode, Adam and Bila, Nilton and de Lara, Eyal and Bala, Vasanth and Satyanarayanan, Mahadev. Privacy-Sensitive VM Retrospection. HotCloud' 11.

[richter2014] Wolfgang Richter and Canturk Isci and Benjamin Gilbert and Jan Harkes and Vasanth Bala and Mahadev Satyanarayanan. Agentless Cloud-wide Streaming of Guest File System Updates. IC2E' 14.

[satya2010] Satyanarayanan, Mahadev and Richter, Wolfgang and Ammons, Glenn and Harkes, Jan and Goode, Adam. The Case for Content Search of VM Clouds. CloudApp' 10.

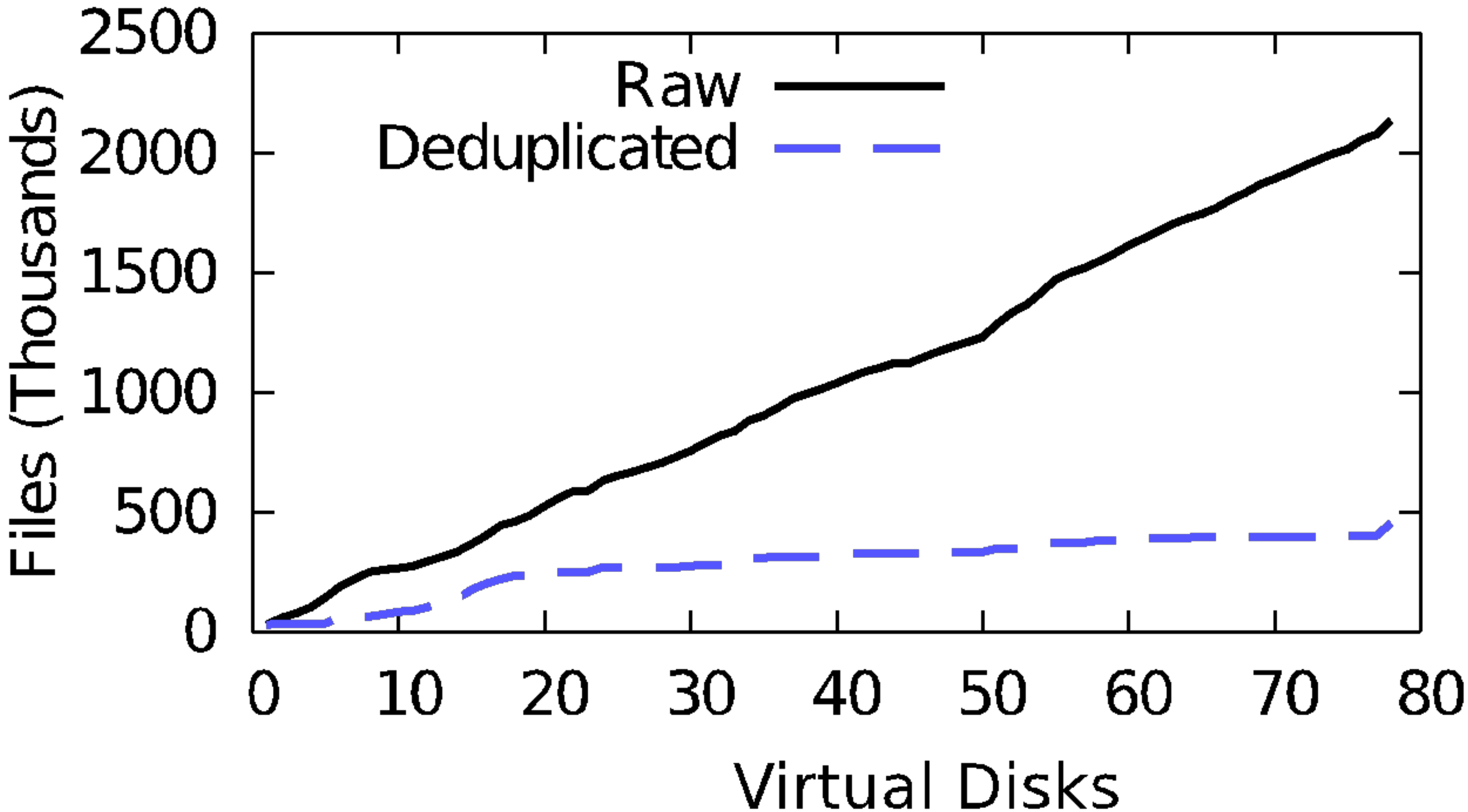
[sivathanu2003] Sivathanu, Muthian and Prabhakaran, Vijayan and Popovici, Florentina I. and Denehy, Timothy E. and Arpaci-Dusseau, Andrea C. and Arpaci-Dusseau, Remzi H. Semantically-Smart Disk Systems. FAST' 03.

[wei2009] Wei, Jinpeng and Zhang, Xiaolan and Ammons, Glenn and Bala, Vasanth and Ning, Peng. Managing Security of Virtual Machine Images in a Cloud Environment. CCSW' 09.

[wikipedia] Wikipedia. Merkle-Damgård Construction. 2014, http://goo.gl/ZUQZFE.

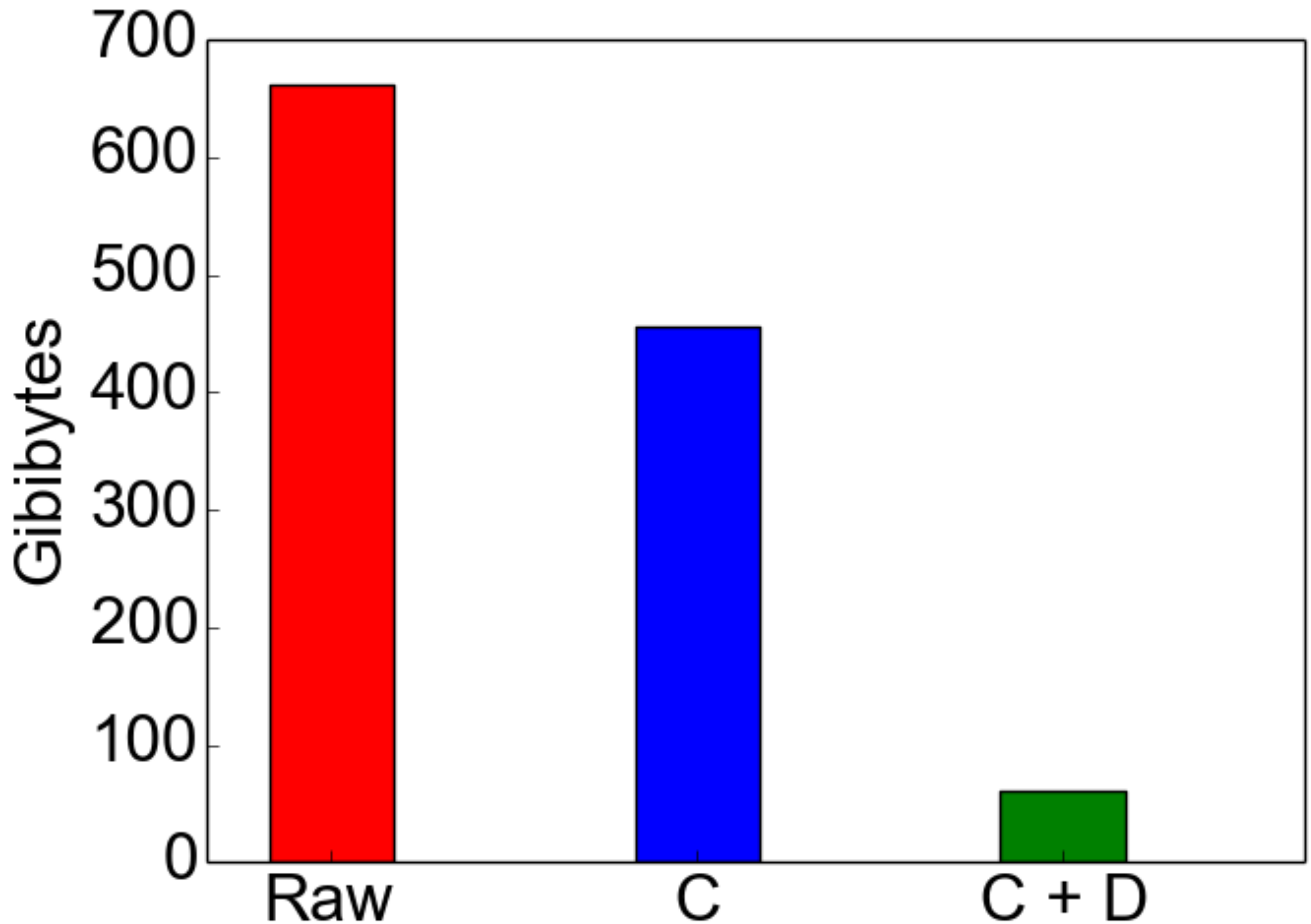
[zhang2006] Youhui Zhang and Yu Gu and Hongyi Wang and Dongsheng Wang. Virtual-Machine-based Intrusion Detection on File-aware Block Level Storage. SBAC-PAD' 06.

File-level Duplication?

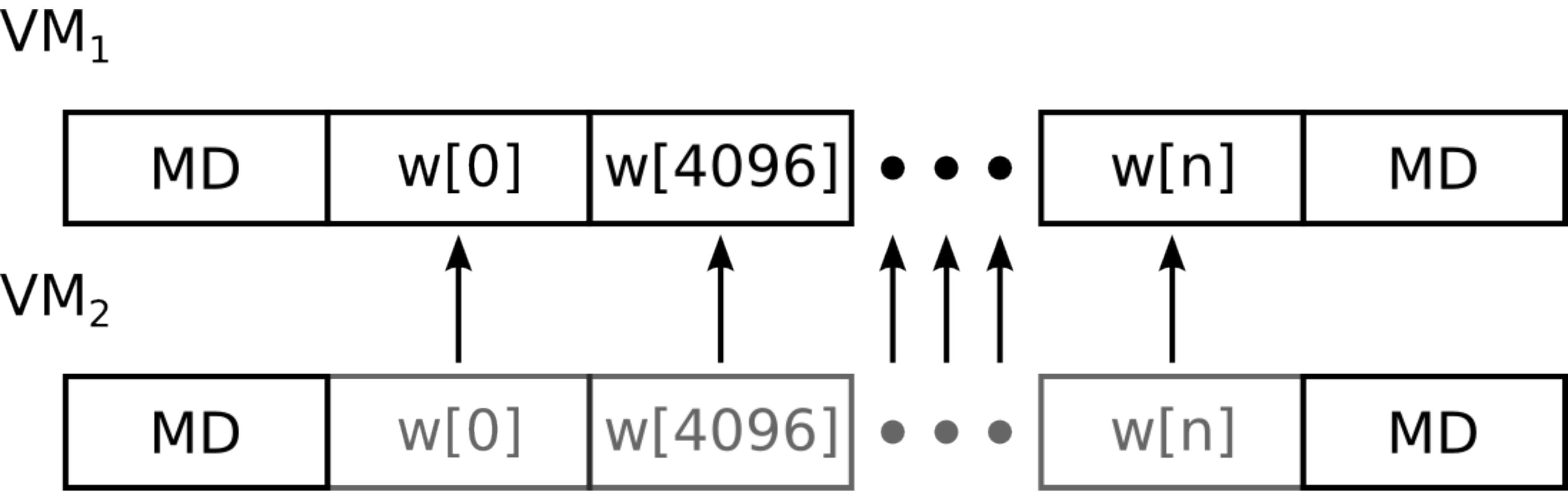


[satya2010]

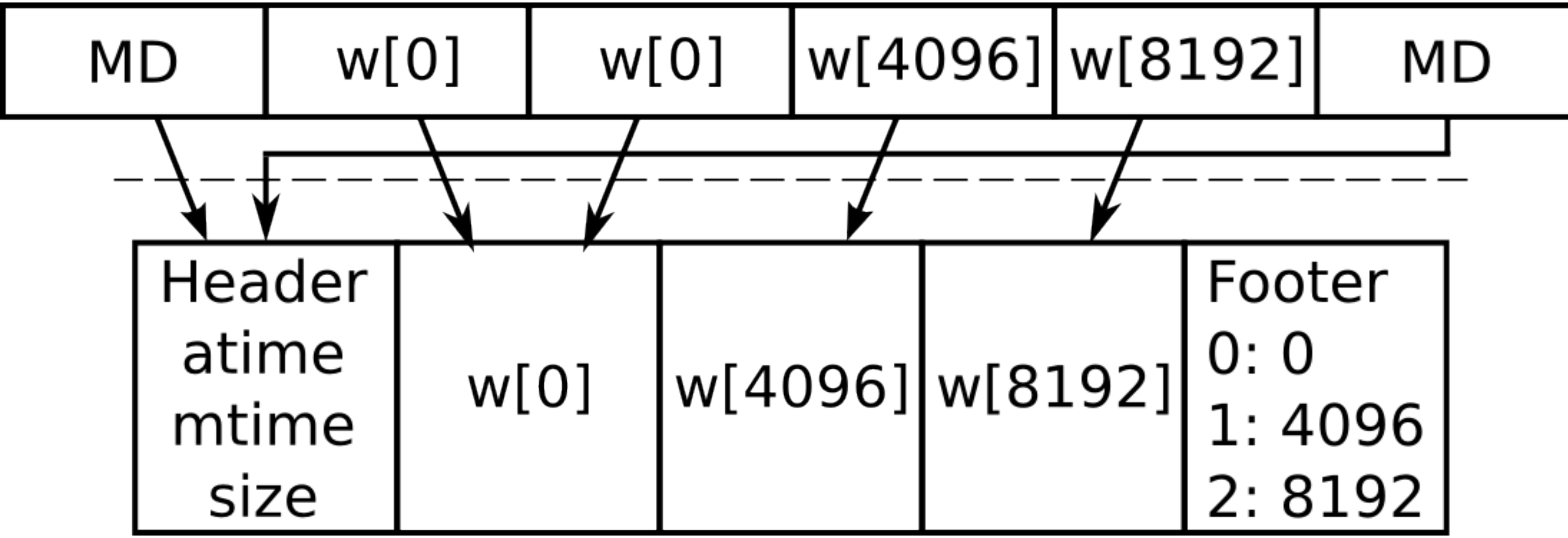
Block-level Compression and Deduplication



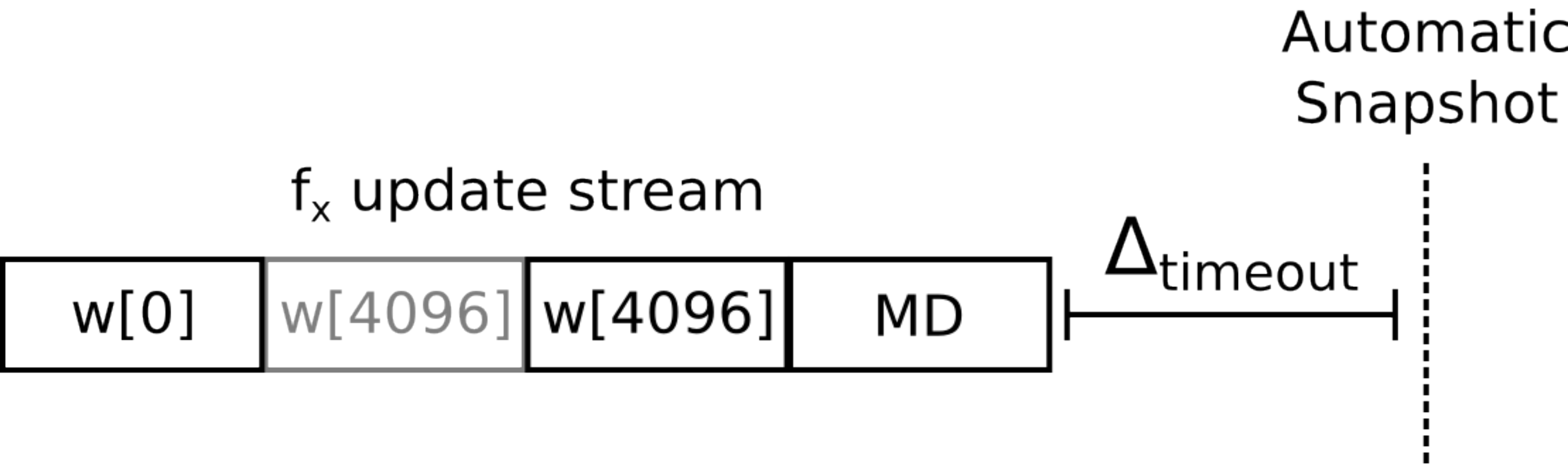
Ensure Block-aligned Data



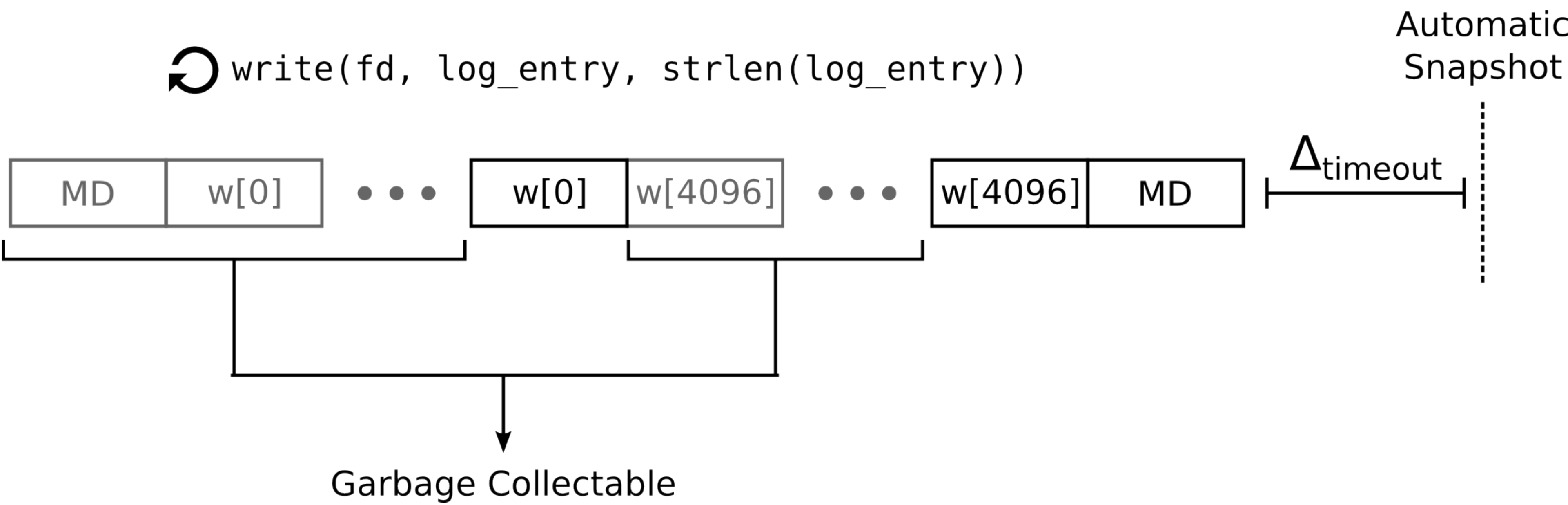
On-disk Log Layout

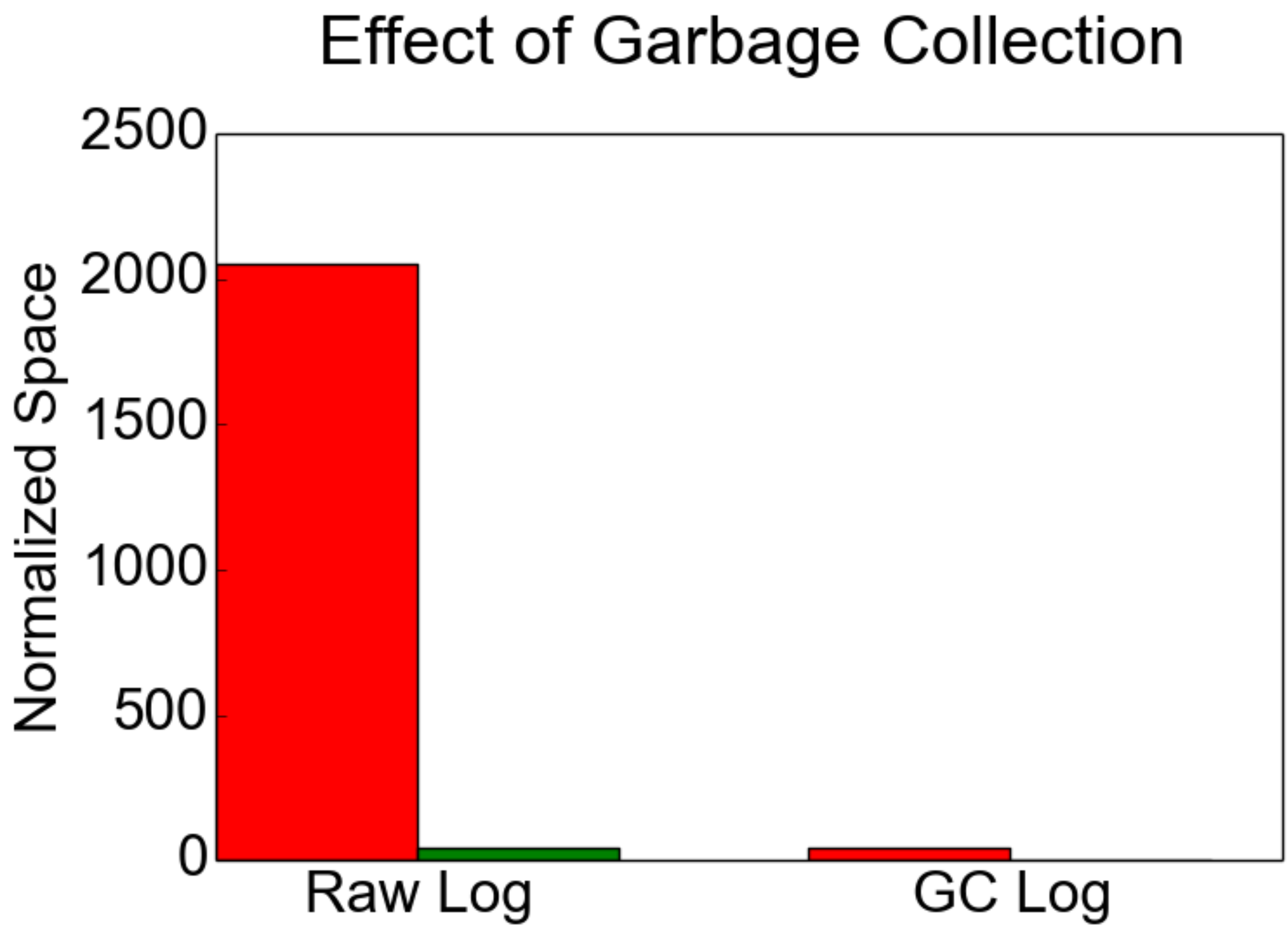


Versioning Heuristic



Garbage Collection




Effect of Garbage Collection
Normalized Space
2500
2000
1500
1000
500
0
Raw Log
GC Log

How Slow is Crawling? (used space)

Used (GB)	MD Raw (MB)	MD gzip (MB)	Crawl (s)	Load (s)
2.6	**109**	**9**	**10.16 (0.89)**	**13.52 (0.41)**
4.6	117	11	10.75 (0.62)	19.27 (1.30)
6.6	123	12	11.47 (0.60)	24.04 (0.14)
8.6	130	13	12.77 (0.65)	29.68 (0.31)
11	136	14	14.20 (0.55)	38.84 (0.34)
13	143	15	18.24 (0.56)	40.08 (0.27)
15	149	17	17.49 (0.81)	42.42 (0.29)
17	156	18	18.47 (0.83)	51.39 (0.33)

Metadata compressed size < 18 MB, crawl time < 20 seconds, load time < 60 seconds.

20 GB Raw disk; single ext4 partition; experiments repeated 20 times; first row stock Ubuntu 12.04 LTS Server

How Slow is Crawling? (used inodes)

inodes	MD Raw (MB)	MD gzip (MB)	Crawl (s)	Load (s)
127,785	**109**	**9**	**10.16 (0.89)**	**13.52 (0.41)**
500,000	243	26	50.81 (1.26)	31.06 (0.23)
1,000,000	421	49	120.73 (1.37)	56.37 (0.51)
1,310,720*	533	65	164.91 (1.73)	76.14 (1.00)

Metadata compressed size < 65 MB, crawl time < 3 minutes, load time < 78 seconds.

20 GB Raw disk; single ext4 partition; experiments repeated 20 times; first row stock Ubuntu 12.04 LTS Server; * means the file system ran out of inodes and could not create more files

What is an agent?

An agent is a process performing administrative tasks that generally runs in the background.

Loggly – log collection and analytics

ClamAV – virus scanning

Dropbox – file backup and synchronization

Windows Update – OS / system update

Tripwire – file-based intrusion detection

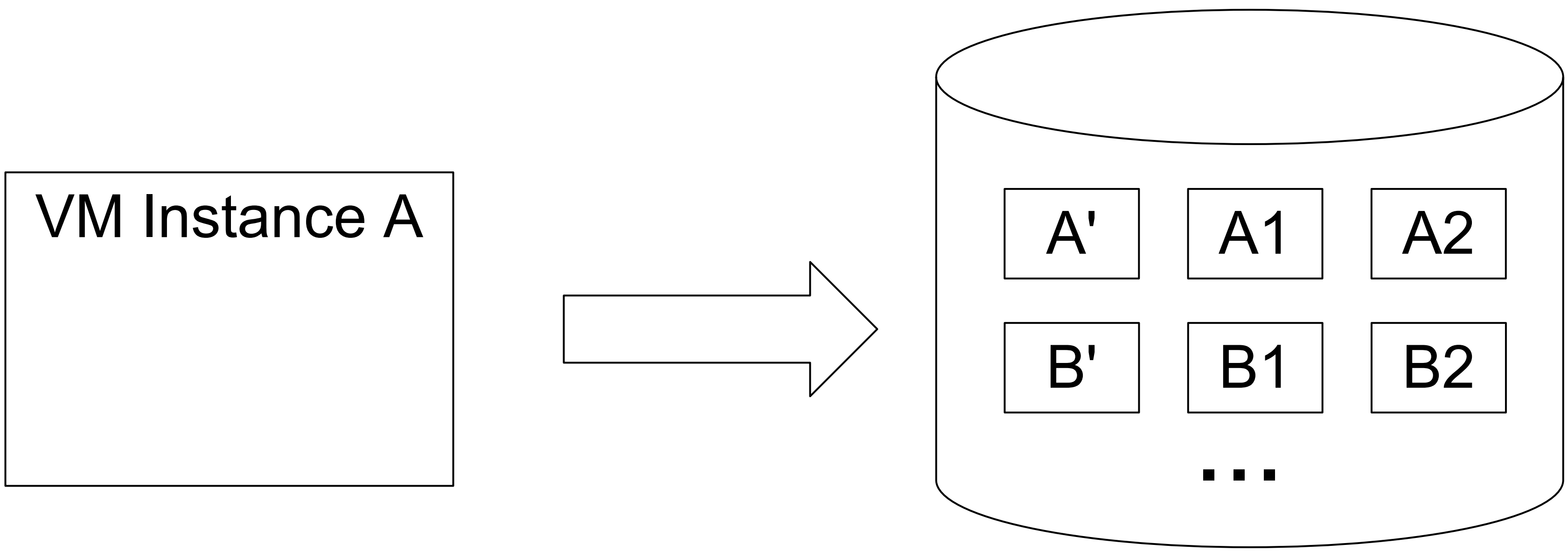
Research Questions

1. What quantitative and qualitative benefits does an agentless approach have over agents?
2. How does agentless monitoring of disk state change the implementation of file-level monitoring?
3. How does agentless monitoring of disk state change the implementation of snapshotting?
4. What properties do interfaces need for scaling file-level monitoring workloads?

Introspection vs. Retrospection

Examine active state of VM during execution

Examine historical state of VMs and their snapshots

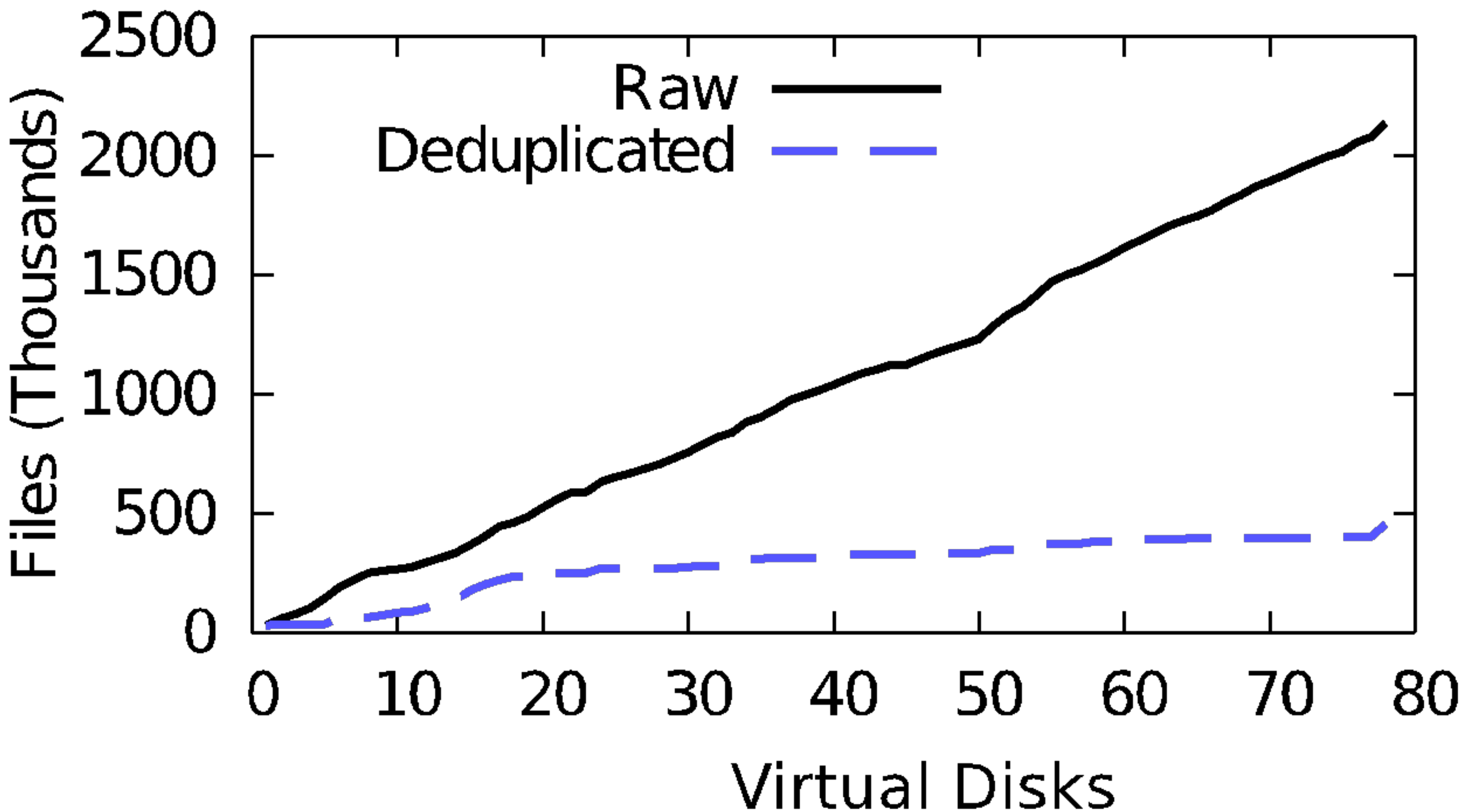


Examine live logs

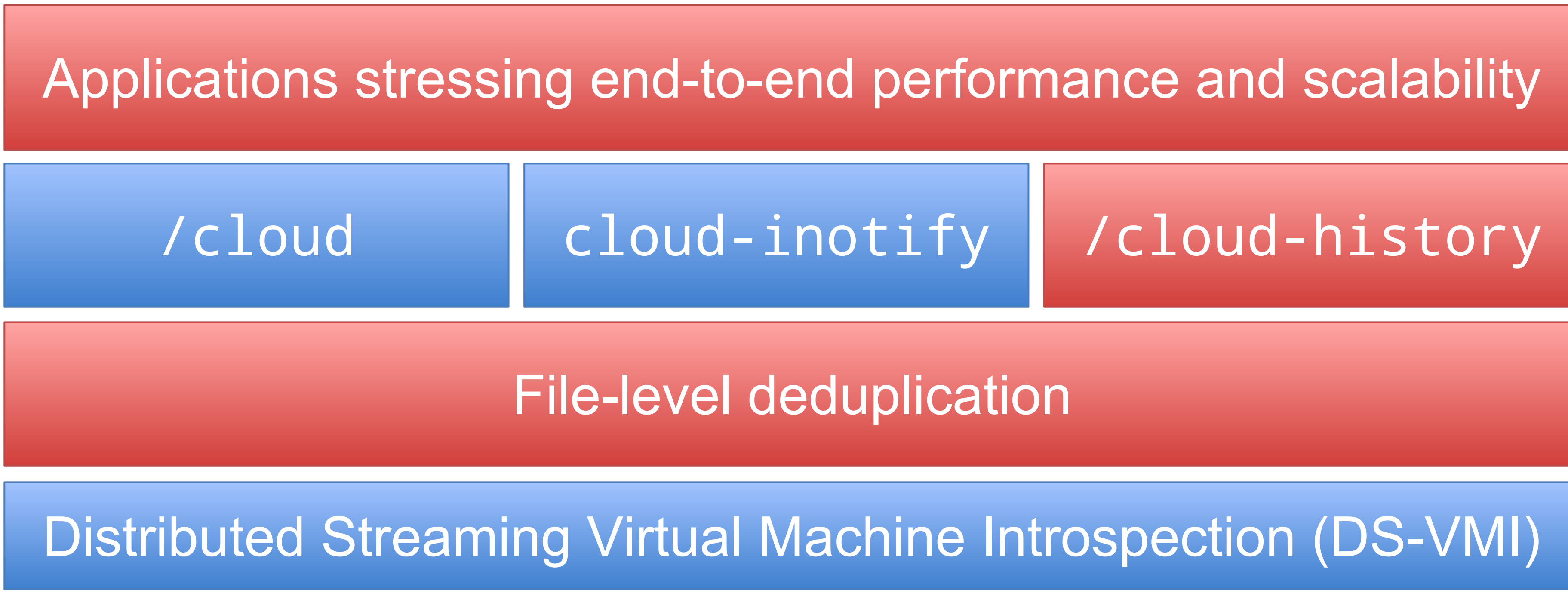
Examine all historic logs A*

[richter2011]

File-level Deduplication



[satya2010]


Applications stressing end-to-end performance and scalability
/cloud
cloud-inotify
/cloud-history
File-level deduplication
Distributed Streaming Virtual Machine Introspection (DS-VMI)

What is a monitoring agent?

A monitoring agent is a process performing administrative tasks that generally runs in the background and can not modify state.

Loggly – log collection and analytics

ClamAV – virus scanning

Dropbox – file backup and sync

Windows Update – OS / system update

Tripwire – file-based intrusion detection

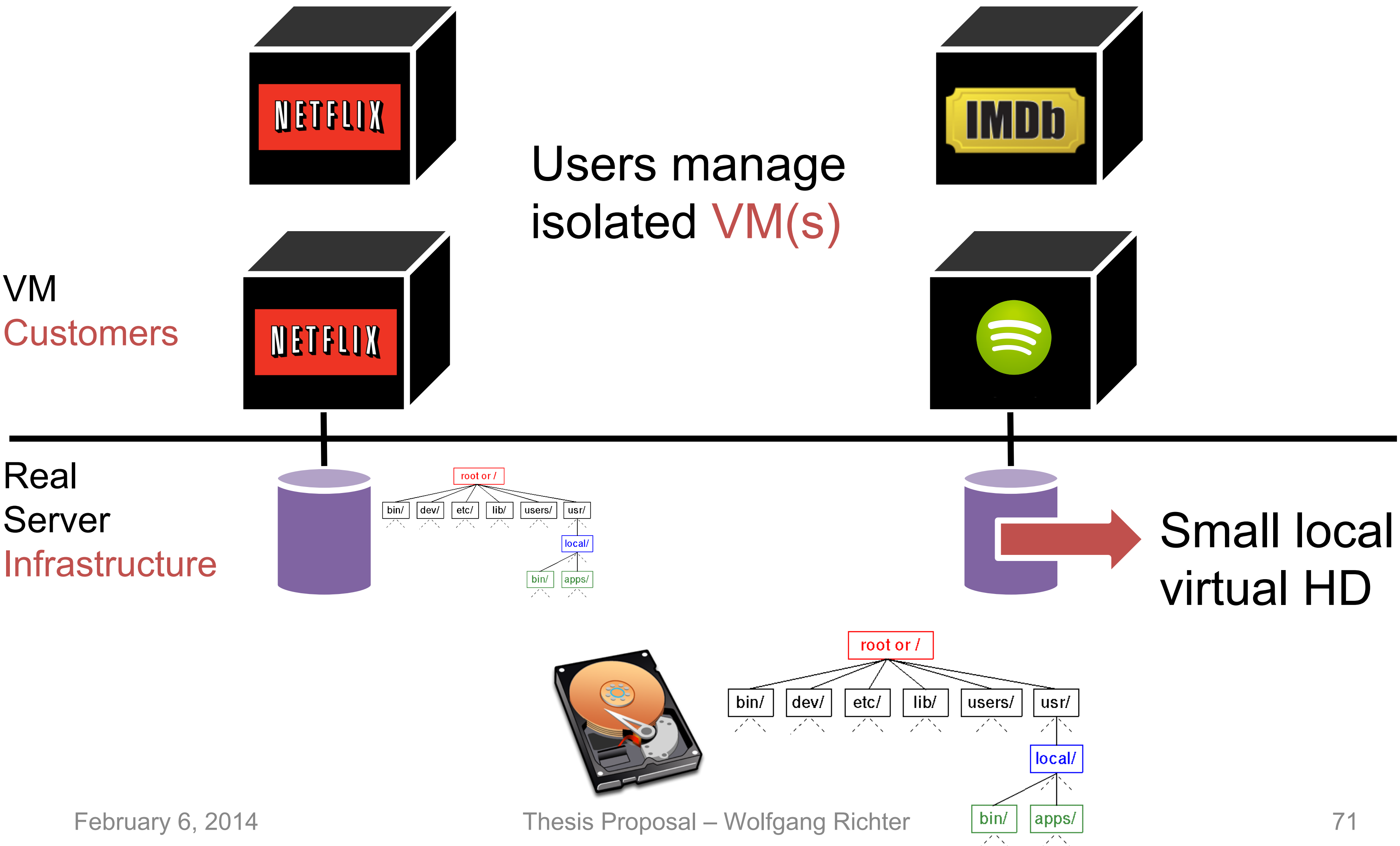
Scalability

- Support 10,000+ monitored systems
  - Overall latency ~10 minutes
  - Reasonable network bandwidth overhead

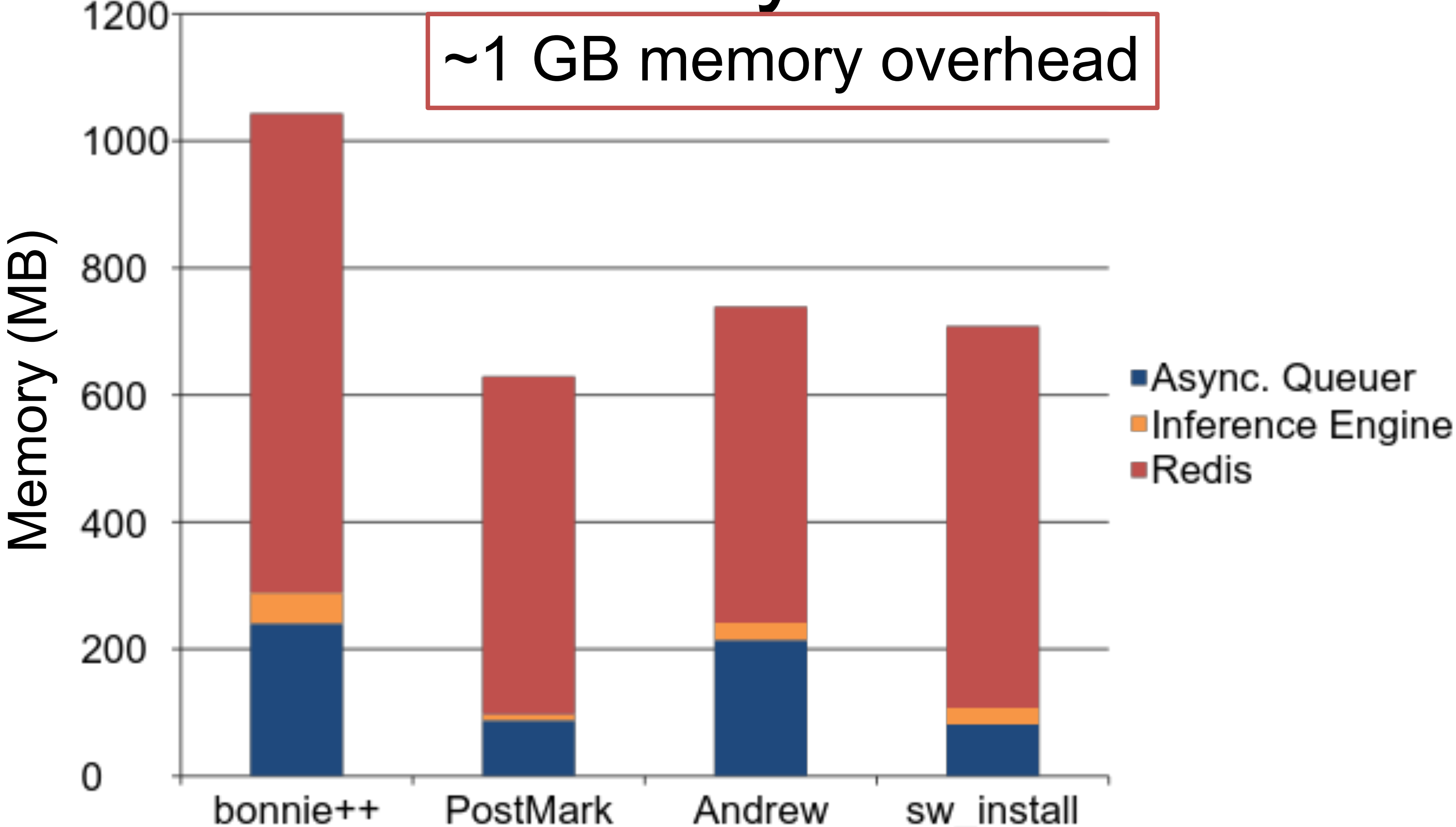
- Maximize monitored VMs per host
  - Minimize decrease in consolidation

[cohen2010]

What is meant by cloud?

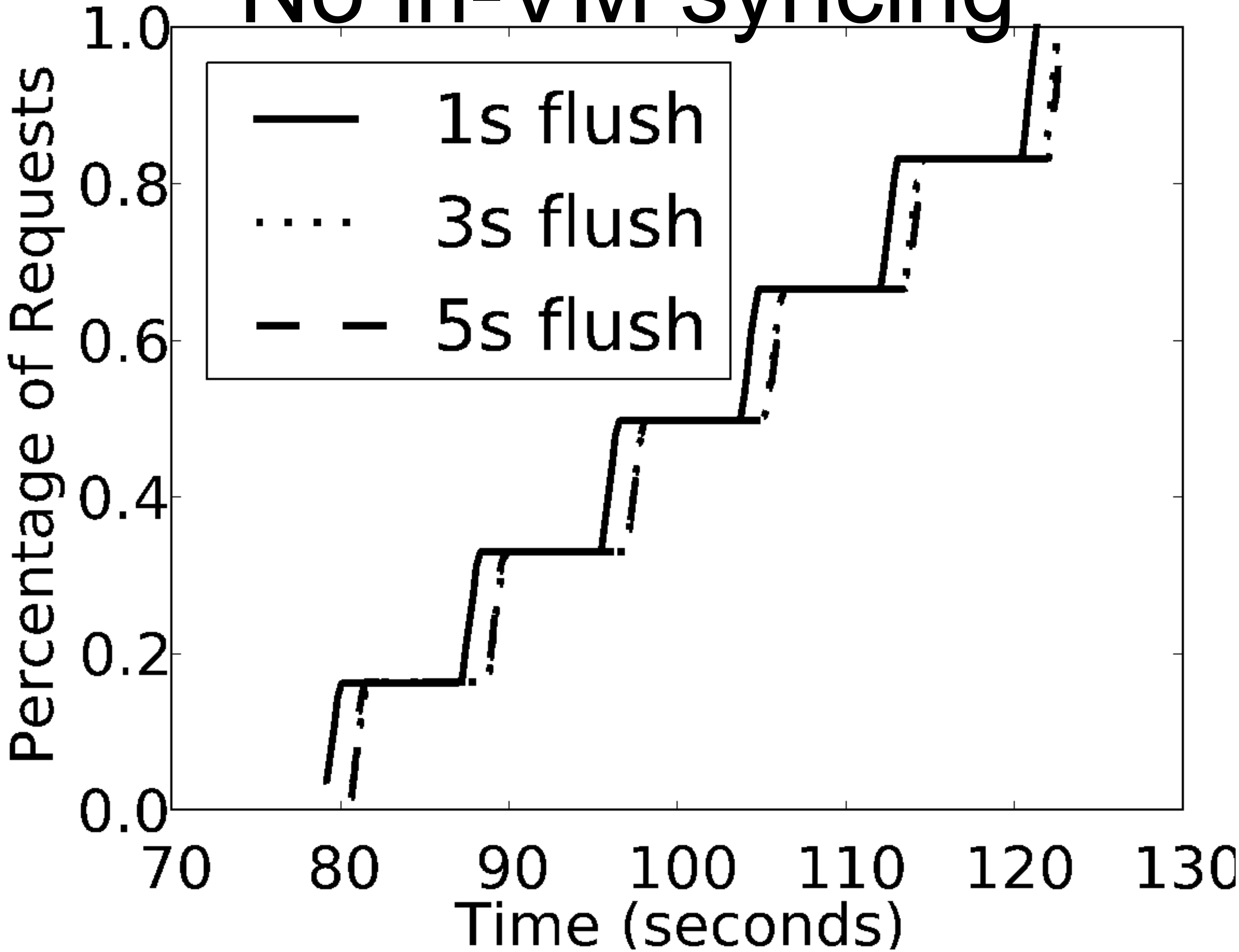


Peak Memory Overhead

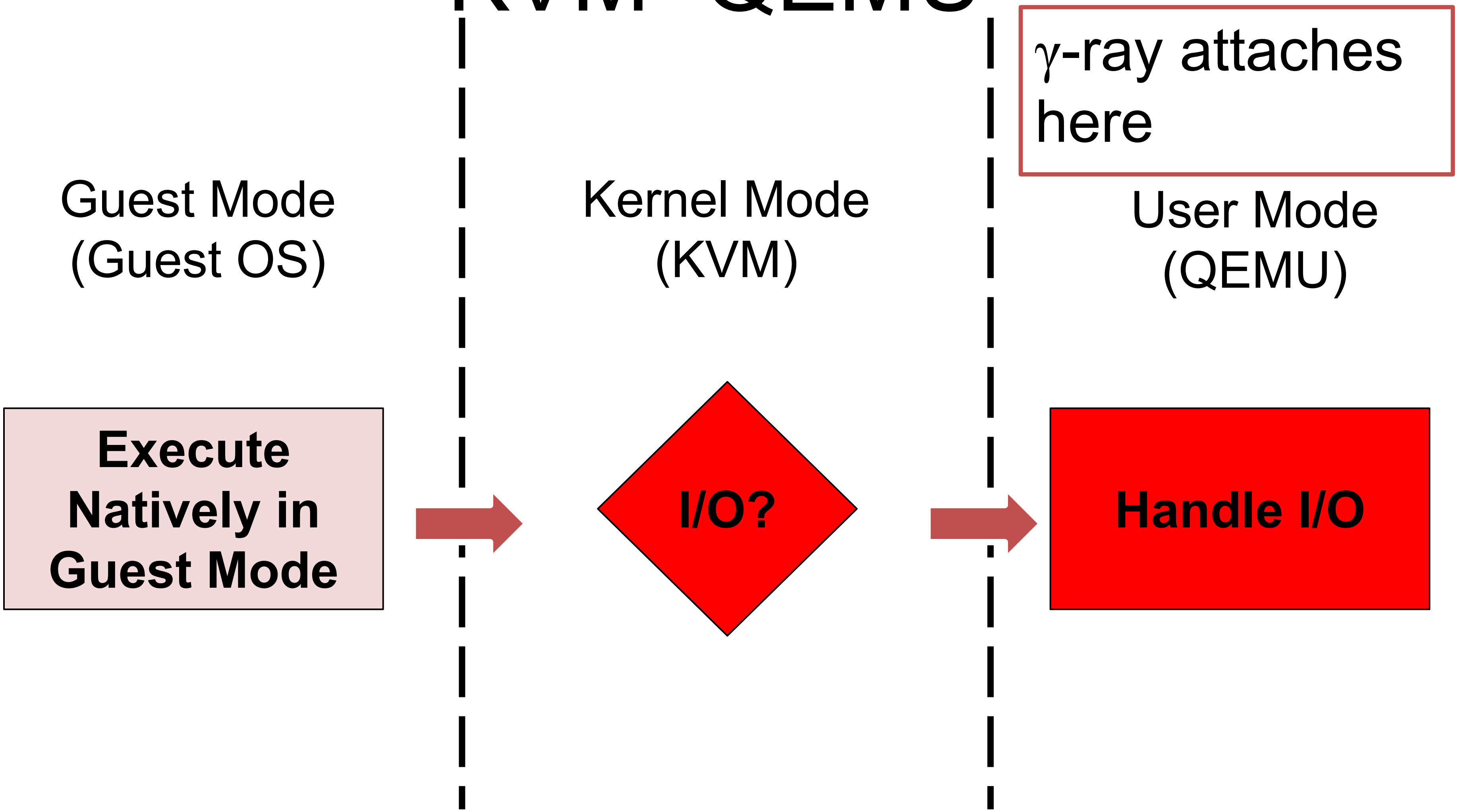


[richter2014]

No in-VM syncing



KVM+QEMU



[kivity2007]

Zero Guest Modifications

- Independent of
  - Guest OS
  - Virtual Machine Monitor (VMM)
  - VM disk format

- Implications
  - Centralize any file-level monitoring task
  - Remove the need for in-VM processes
  - Solve monitoring at an infrastructure-level
  - Maintain compatibility with legacy tools

Teaser: Problem (2)

- TubeMogul suffered cloud storage failure
  - > 50% Fortune 500 use TubeMogul for video ads

Can we take advantage of virtualized infrastructure to complete the puzzle?

- Did TubeMogul corrupt their own file system?

[brousse2011]

Teaser: Potential Win (3)

- Deeper knowledge of application performance
  - Allocate resources more intelligently to VMs
- Coupled with application service level objective

80% reduced mean deviation of response time
100% increase number of hosted VMs

[sangpetch2010]

Bootstrapping: ext4 Example (1)

MBR   Swap   ext4

```
uint8_t   code[440];
uint32_t  disk_sig;
uint16_t  reserved;
pt_table  pt[4];
uint8_t   signature[2];
```

```
uint32_t s_first_data_block;
uint32_t s_inodes_per_group;
uint16_t s_inode_size;
uint8_t  s_last_mounted[64];
...
```

```
uint8_t   status;
uint8_t   start_chs[3];
uint8_t   pt_type;
uint8_t   end_chs[3];
uint32_t  first_sector_lba;
uint32_t  sector_count;
```

Bootstrapping: ext4 Example (2)

ext4

Superblock BGD Table Inode Table Data

```
uint32_t bg_block_bitmap;
uint32_t bg_inode_bitmap;
uint32_t bg_inode_table;
uint16_t bg_flags;
```

```
uint32_t inode;
uint16_t rec_len;
uint8_t  name_len;
uint8_t  file_type;
uint8_t  name[0,255];
```

```
uint16_t i_mode;
uint32_t i_size_lo;
uint16_t i_links_count;
uint32_t i_block[15];
uint32_t i_size_hi;
```

```
uint16_t eh_entries;
uint16_t eh_depth;

uint16_t ee_block;
uint16_t ee_start_hi;
uint32_t ee_start_lo;
```

Keeping Track of 70,000+ Servers: The Akamai Query System

- Scalable: goal of 70,000 monitored VMs
  - > 1,000,000 software components
- Real-Time: flushed file updates < 10 minutes
- File Updates: data write, metadata updates
  - Create, delete, modify permissions, write

[cohen2010]

Tunable Parameters

Tunable	Default
Unknown Write TTL	5 minutes
Async Flush Timeout	5 seconds
Async Queue Size Limit	250 MB
Async Outstanding Write Limit	16,384 writes
Redis Maximum Memory	2 Gigabytes

Problem 1: Monitoring Large VM Deployments

- Monitoring instances is critical for
  - Debugging distributed applications
  - Measuring performance
  - Intrusion detection

- Clouds leave this unsolved for their users
  - Users resort to running agents within VMs
  - Log monitoring (Splunk), anti-virus (ClamAV), etc.

Problem 2: Black Box Metrics Aren't Enough

- Coarse-grained metrics are good detectors
  - Anomaly detection (memory usage suddenly high)
  - Early warning systems (onset of thrashing)
- But what about answering why?
  - Root cause analysis (memory up from DB config)
  - A fundamental issue with black box metrics

[tan2012]

Best Practice Monitoring Today

- Agents run inside the monitored system
  - Per-OS type
  - Per-Application type
  - Per-System configuration
  - Per-System update + patch
  - Sometimes globally aware

[kufel2013]

Reimagining Monitoring

General

OS and application agnostic

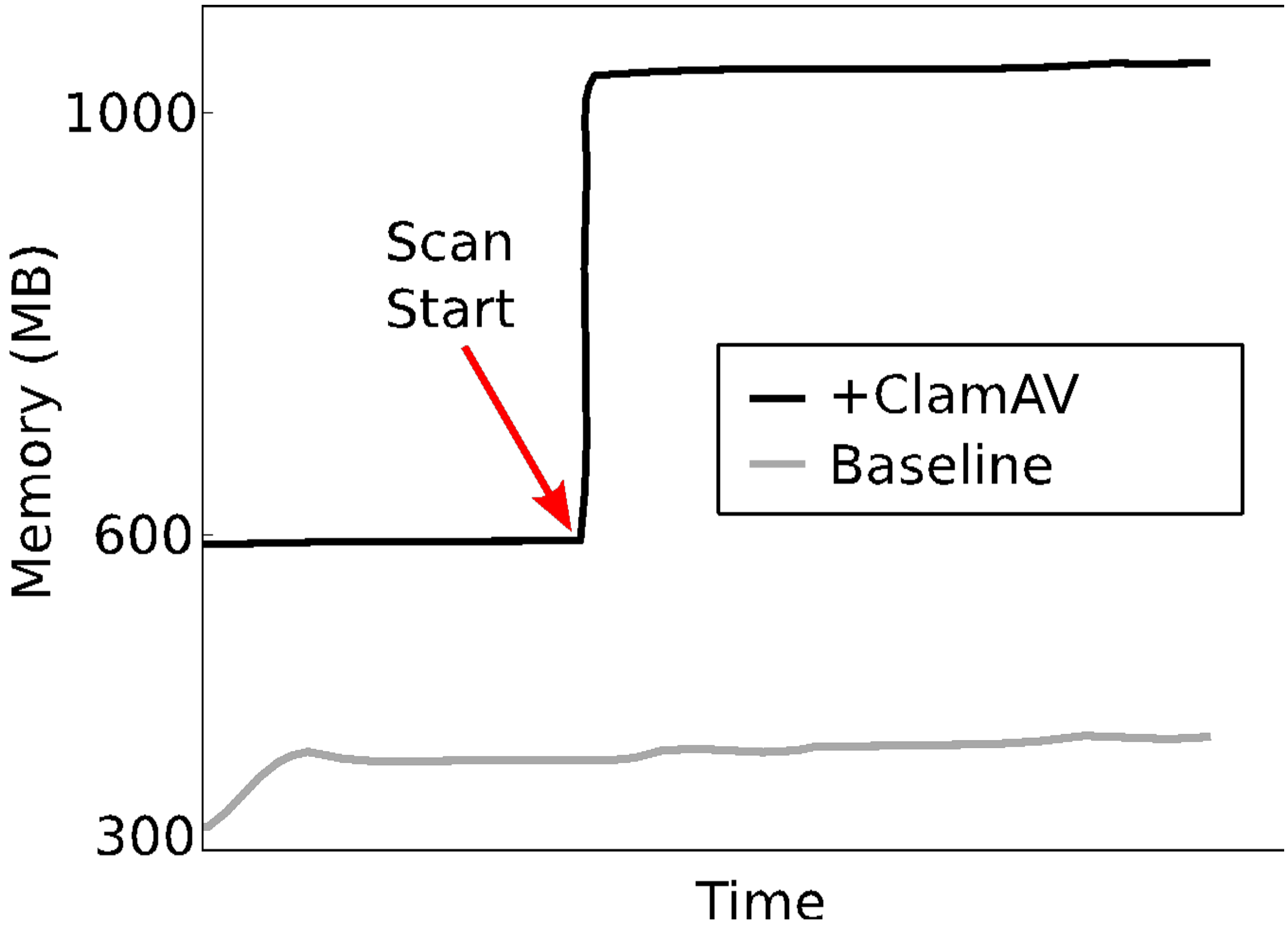
Independent

Misconfiguration and Compromise

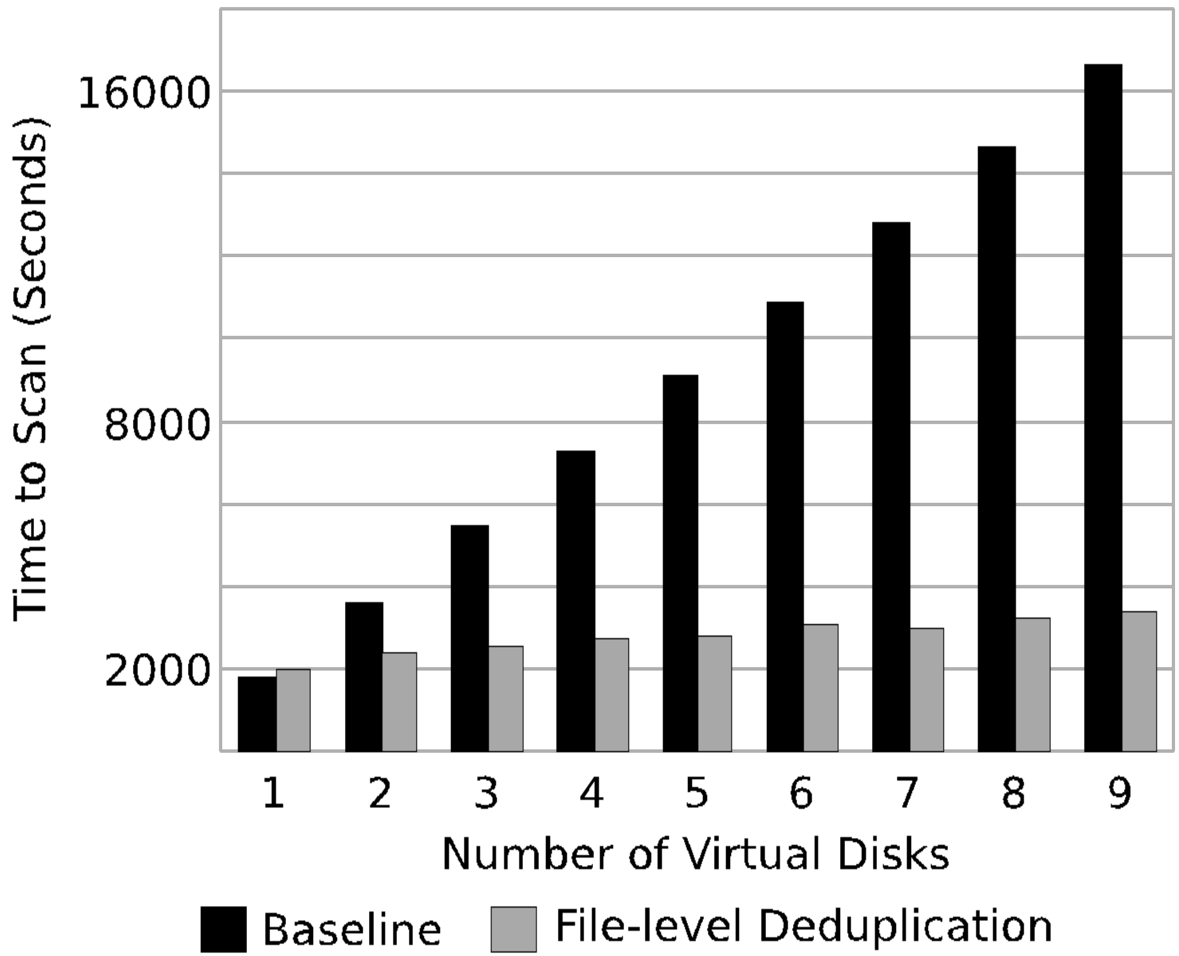
Scalable

Globally aware

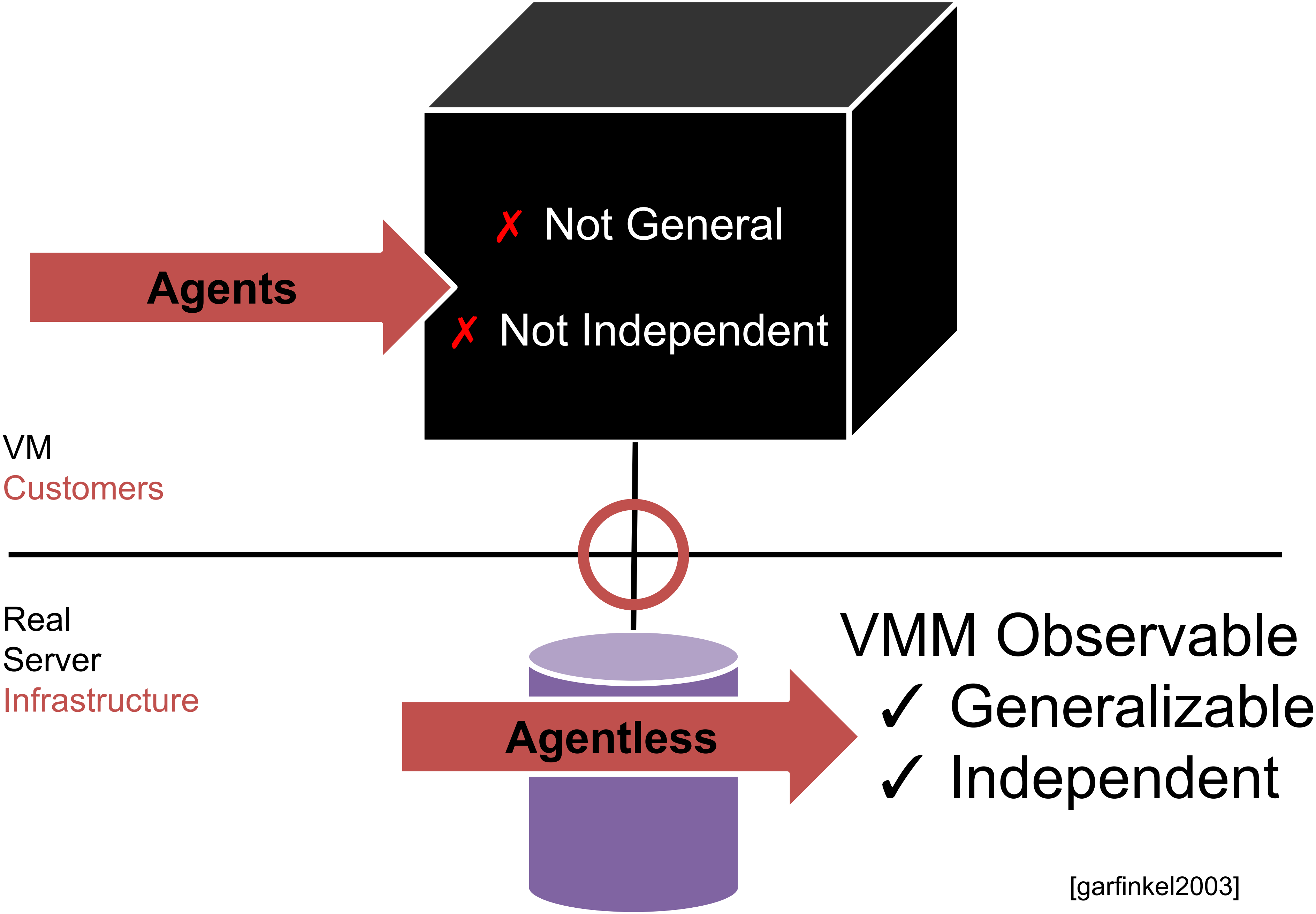
Independent Monitoring Resources



Leverage Global Knowledge



[wei2009]


Agents
✗ Not General
✗ Not Independent
VM
Customers
Real
Server
Infrastructure
Agentless
VMM Observable
✓ Generalizable
✓ Independent
[garfinkel2003]

Applications stressing end-to-end performance and scalability

/cloud

cloud-inotify

/cloud-history

File-level deduplication

Distributed Streaming Virtual Machine Introspection (DS-VMI)

Applications

`/cloud`

Virus Scanning (ClamAV)

Log Collection (Splunk)

`cloud-inotify`

Continuous Compliance Monitoring

`/cloud-history`

File Recovery

Unindexed Search

Planned Measurements

- Latency-completeness-overhead
  - Vary queue sizes and flush parameters
  - Analyze metadata vs data
  - Re-attachment time
- In-VM performance vs Agentless
- Scalability in number of monitored systems
  - Number of monitored systems per host
  - Wikibench

Applications stressing end-to-end performance and scalability

`/cloud` | `cloud-inotify` | **`/cloud-history`**

File-level deduplication

Distributed Streaming Virtual Machine Introspection (DS-VMI)

`/cloud-history`

Strong consistency

Legacy FS Interface

File-level deduplicated snapshots of sets of VM file system subtrees

Method	**Skip Blocks**	**Skip Files**	**Skip Indexing**	**Resource Isolation**	**Not Misconfig.**
Local FS	✔	✔			
Distributed FS	✔	✔	✔		
In-guest Agent	✔	✔	✔		
Block-level				✔	✔
`/cloud-history`	✔	✔	✔	✔	✔

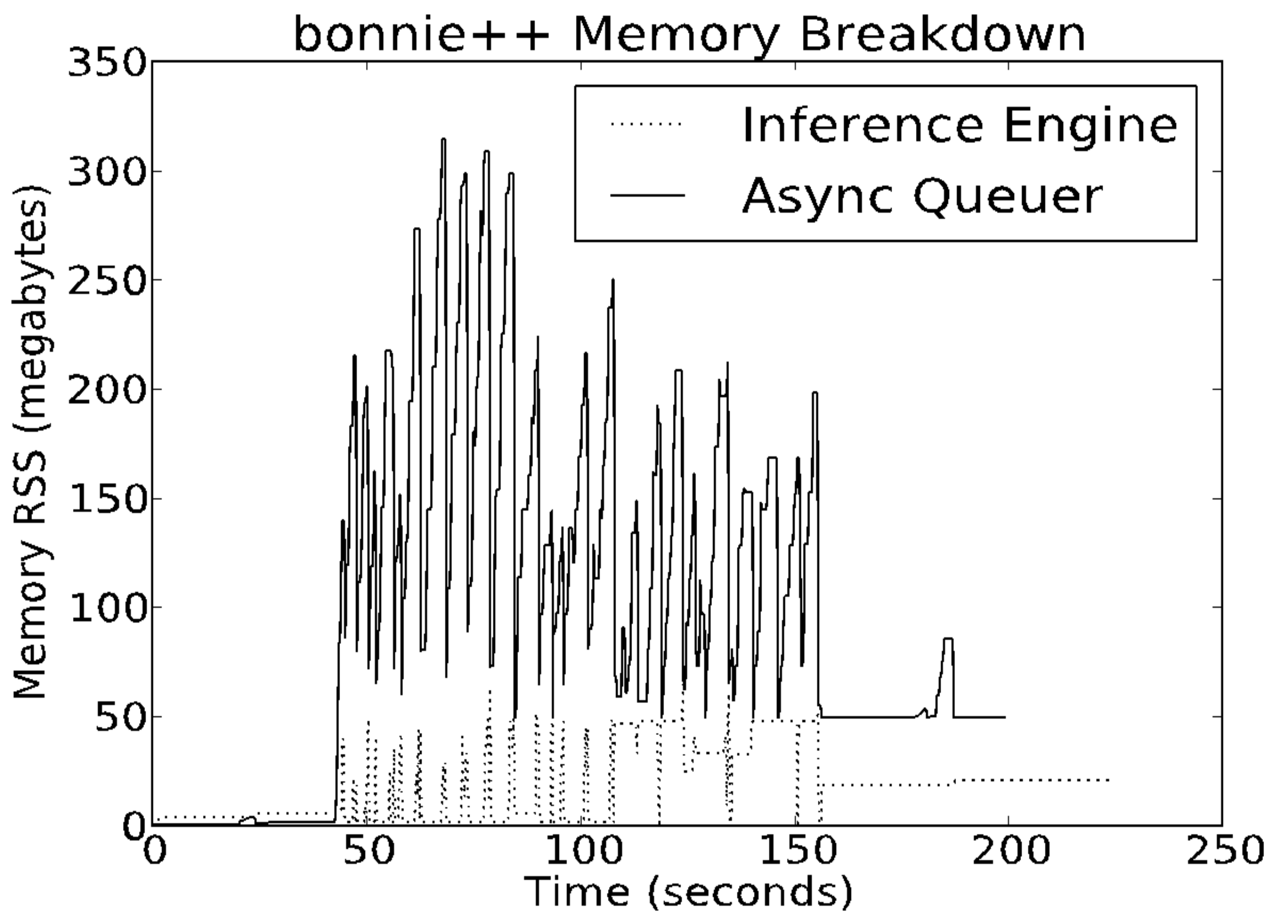
Timeline

January – March:	File-level deduplication
April – June:	`/cloud-history`
July – August:	Applications and measurements
September – October:	Writing
November:	Finish dissertation
December:	Defense

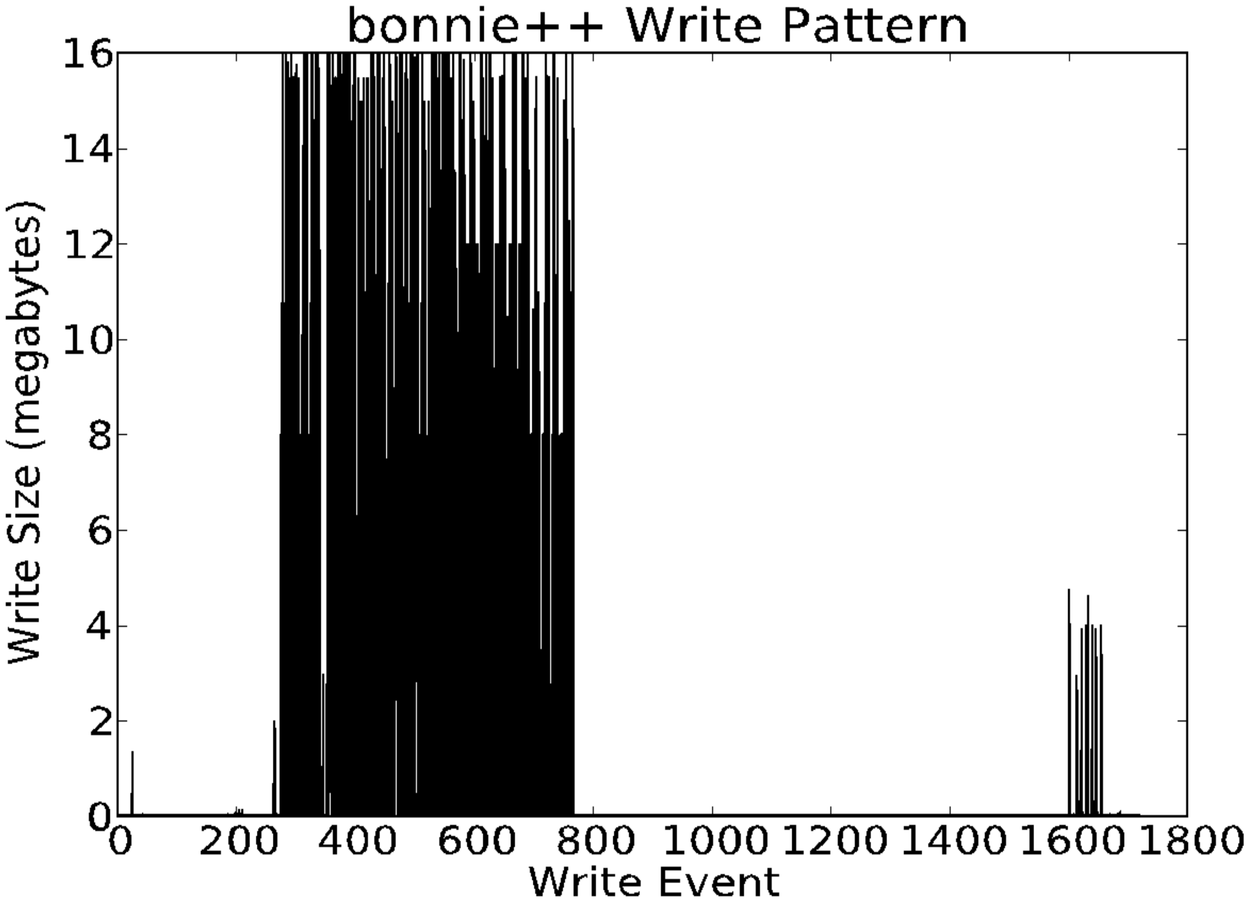
Host Memory Costs

Experiment	Async Q. (MB)	Inf. Eng. (MB)	w/ Redis (MB)
bonnie++	240.48	48.69	1043.48
Andrew	87.97	9.08	629.64
PostMark	214.14	26.89	738.81
SW Install	81.28	25.73	707.96

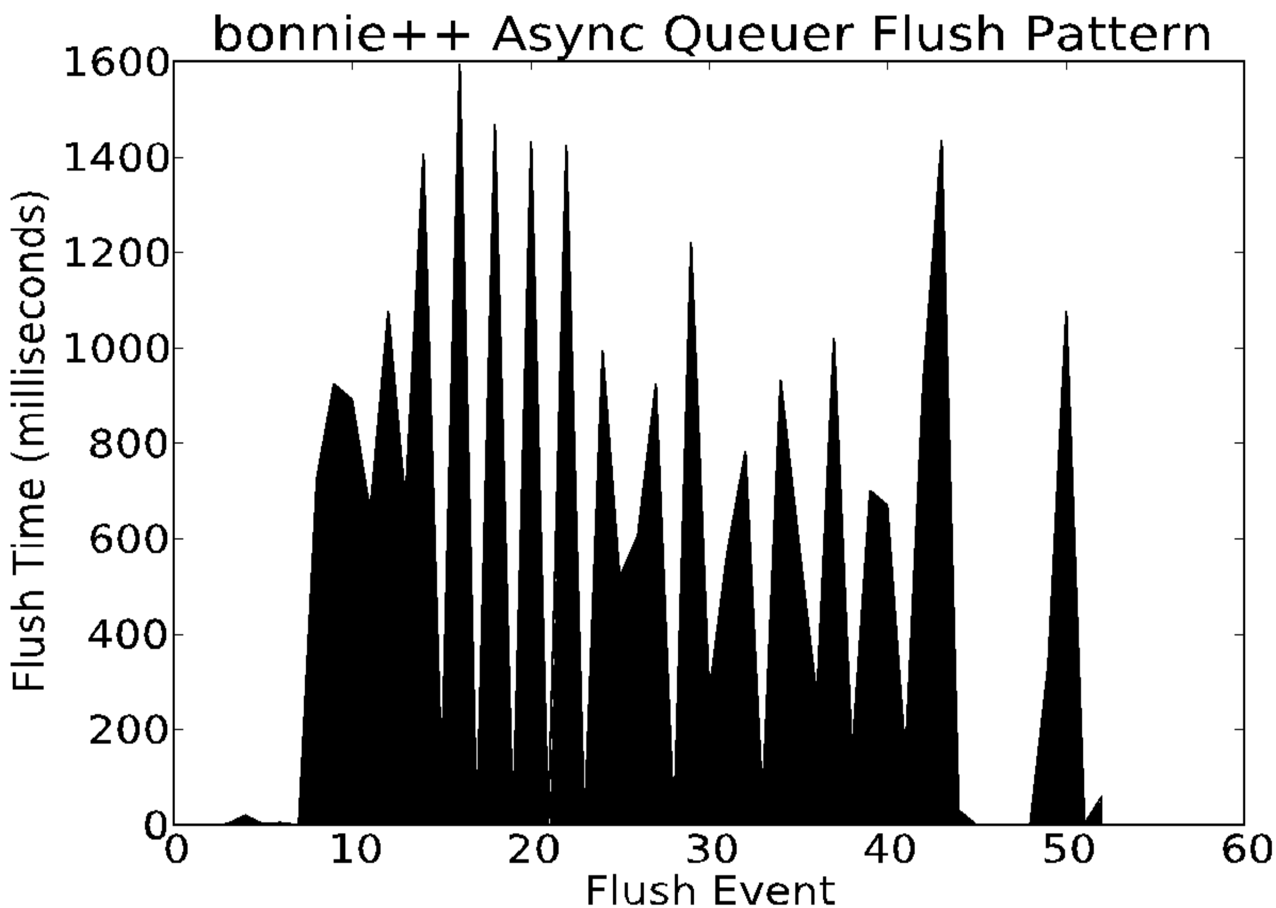
bonnie++ memory



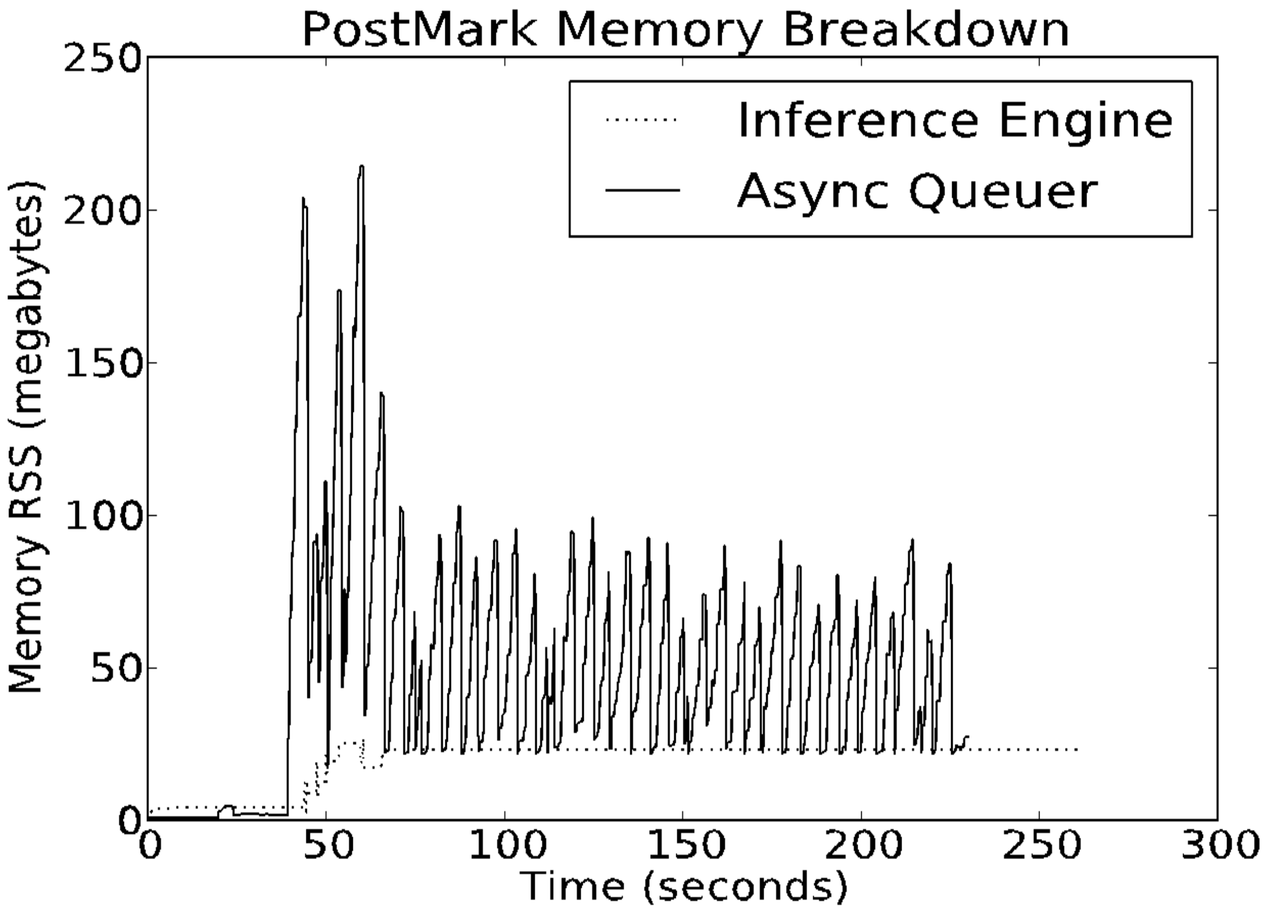
bonnie++ write pattern



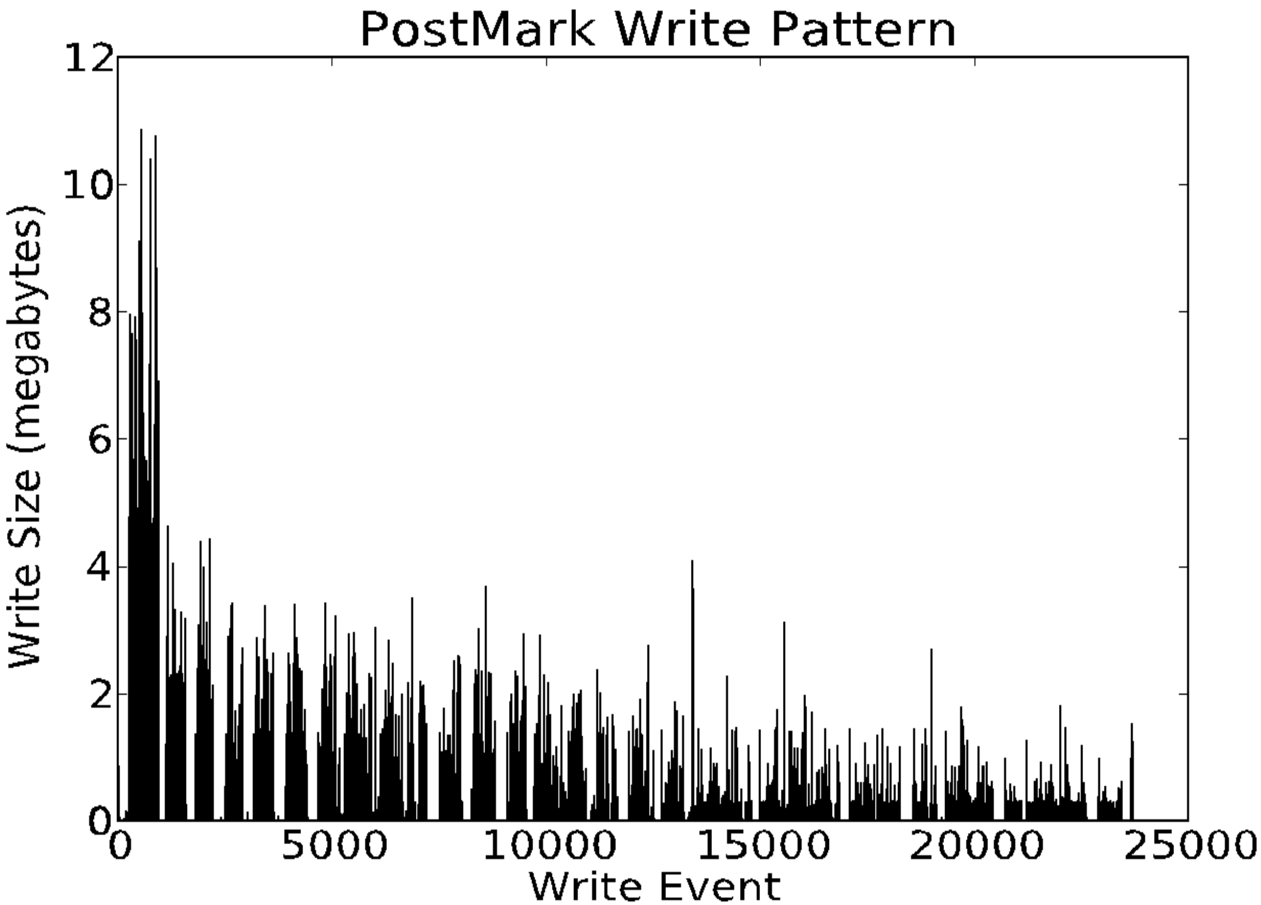
bonnie++ flush pattern



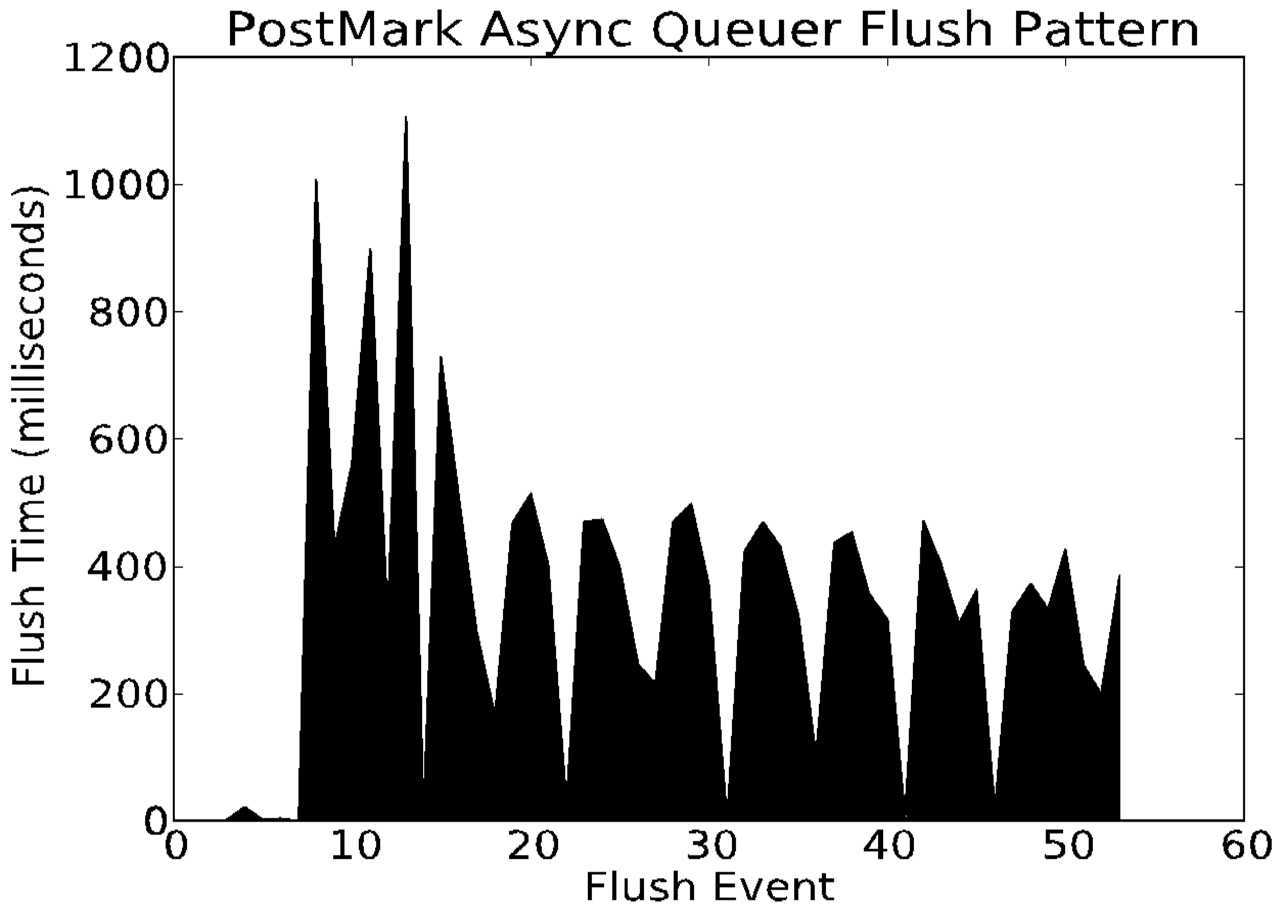
PostMark memory



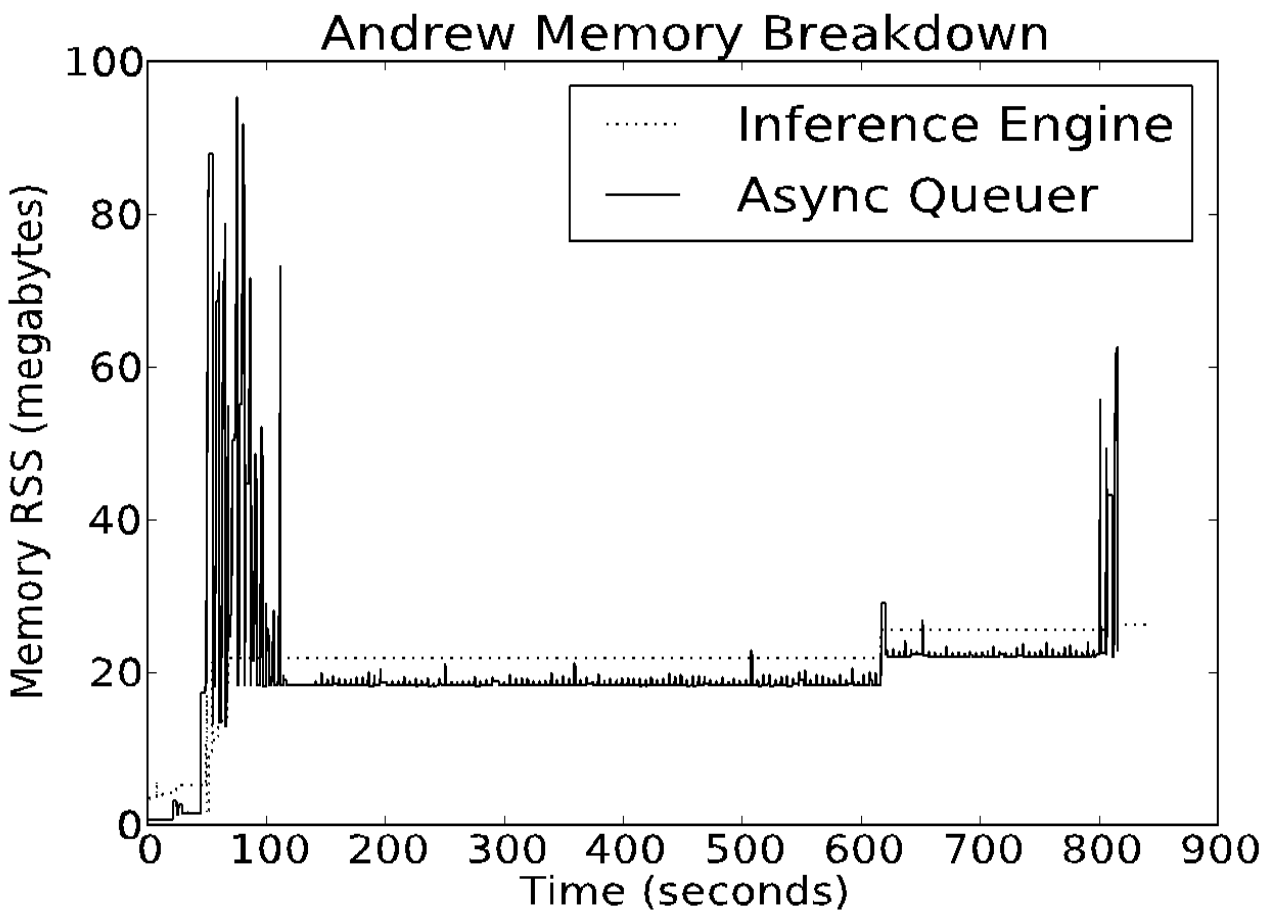
PostMark write pattern



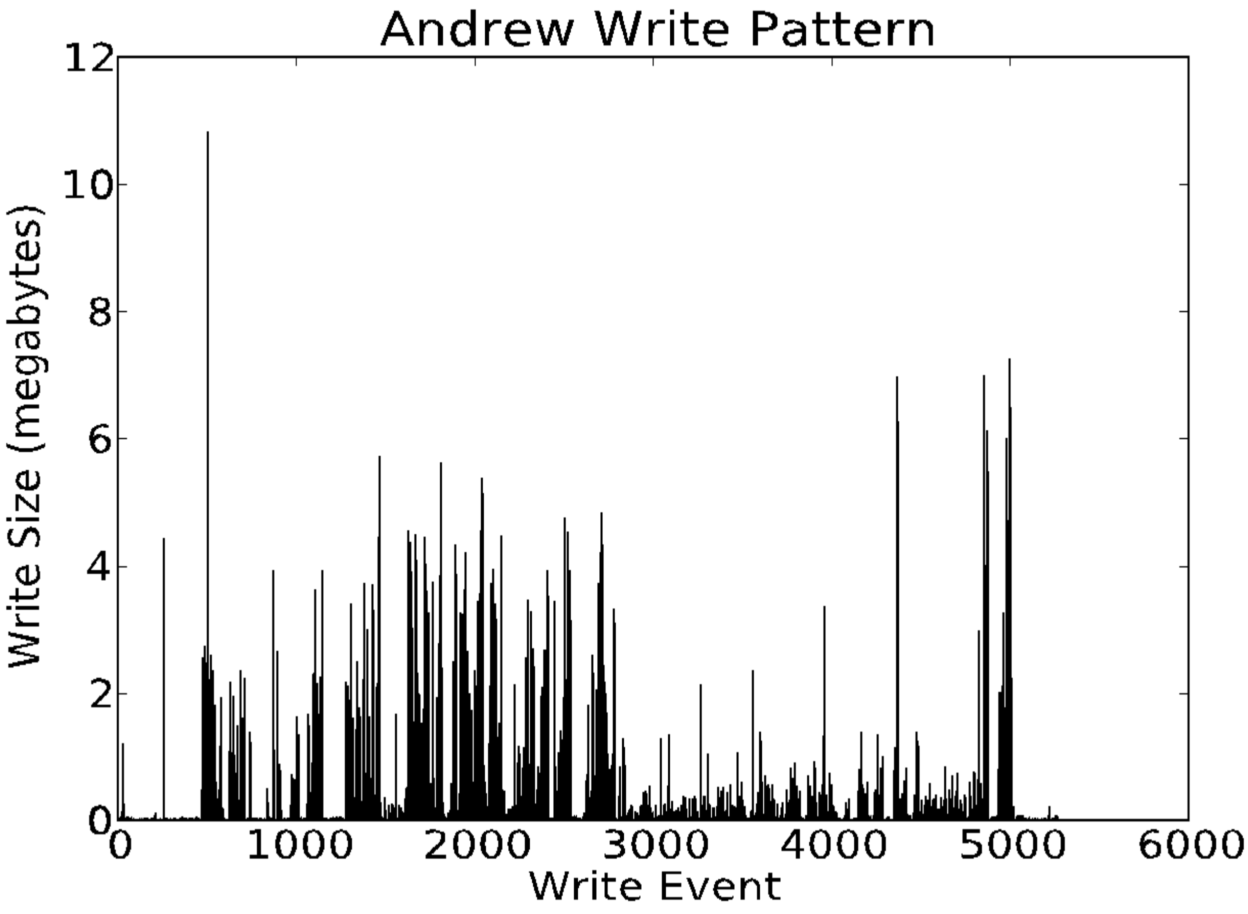
PostMark flush pattern



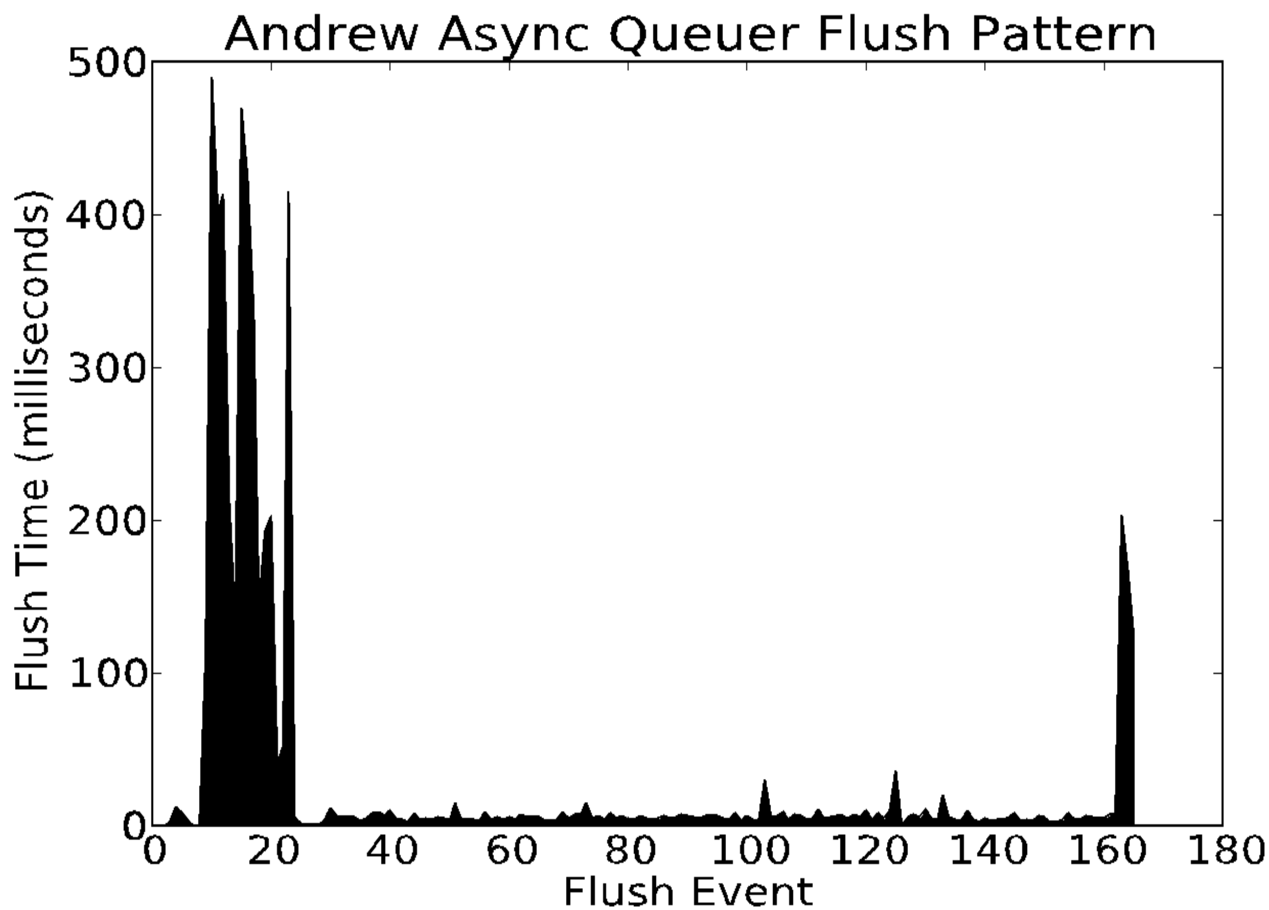
Andrew memory



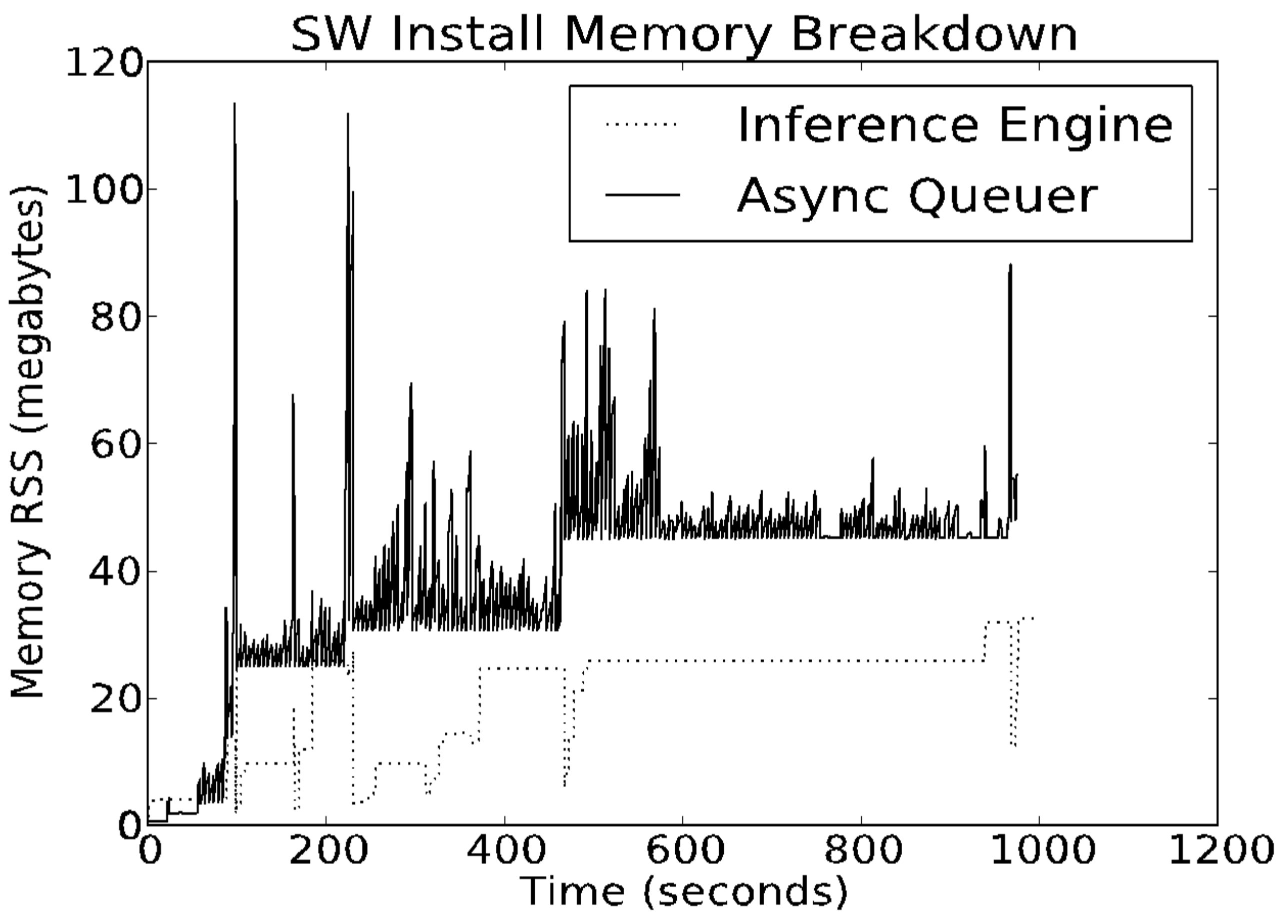
Andrew write pattern



Andrew flush pattern



sw_install memory



sw_install write pattern

sw_install flush pattern