



Education

Scaling Data Center Application Infrastructure

Gary Orenstein, Gear6

SNIA Legal Notice

- The material contained in this tutorial is copyrighted by the SNIA.
- Member companies and individuals may use this material in presentations and literature under the following conditions:
 - ◆ Any slide or slides used must be reproduced without modification
 - ◆ The SNIA must be acknowledged as source of any material used in the body of any document containing material from these presentations.
- This presentation is a project of the SNIA Education Committee.
- Neither the Author nor the Presenter is an attorney and nothing in this presentation is intended to be nor should be construed as legal advice or opinion. If you need legal advice or legal opinion please contact an attorney.
- The information presented herein represents the Author's personal opinion and current understanding of the issues involved. The Author, the Presenter, and the SNIA do not assume any responsibility or liability for damages arising out of any reliance on or use of this information.
NO WARRANTIES, EXPRESS OR IMPLIED. USE AT YOUR OWN RISK.

➤ Scaling Data Center Application Infrastructure

- ◆ Data center managers must support ever-increasing application workloads for up to tens of thousands of users.
- ◆ The demands placed upon the underlying infrastructure require proper planning and architecture in order to scale efficiently.
- ◆ Application managers can choose to deploy application infrastructure internally using readily available technology solutions.
- ◆ Additionally, there are options to extend application infrastructure with cloud computing offerings from Amazon Web Service and Google AppEngine.
- ◆ Even if application managers do not make use of the cloud computing offerings directly, the respective architectures provide an excellent reference model for private infrastructure deployment.
- ◆ In all cases, application managers need to know what tools and resources are available to help scale infrastructure to support an ever increasing user base.

INTRODUCTION

- Systems and data center level view
- The File Explosion and Storage Impact
- Three Case Studies: Background
- Examining the I/O Bottleneck and Conventional Solutions
- Caching for Scale: Data Center Strategies
- Caching in Context: Case Study Review

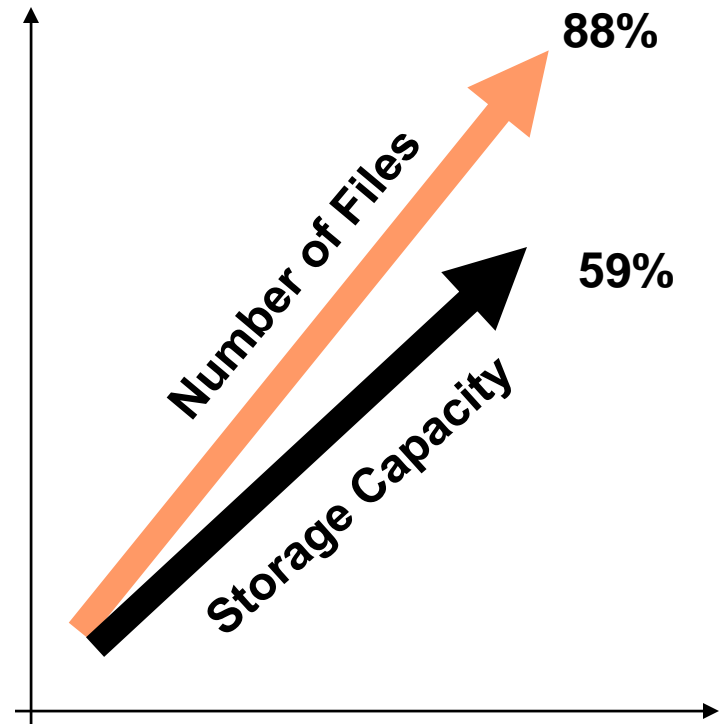
Huge File Counts Driving New Bottlenecks **SNIA**

- Old bottleneck
 - ◆ Limited capacity

- New bottlenecks
 - ◆ Huge file counts
 - ◆ Deep directory requests
 - ◆ Simultaneous users
 - ◆ Unpredictable access patterns

- All leading to...
 - ◆ Painful access times

Compound Growth, 2007-2011



Source: IDC 2008
<http://www.emc.com/collateral/analyst-reports/diverse-exploding-digital-universe.pdf>

File Explosion Issues Facing Individual Companies

Music downloads
in 5 years

5 billion

100 million

Uploaded
photos
per week

Files concurrently
accessed by 30,000 clients
in under 1 millisecond

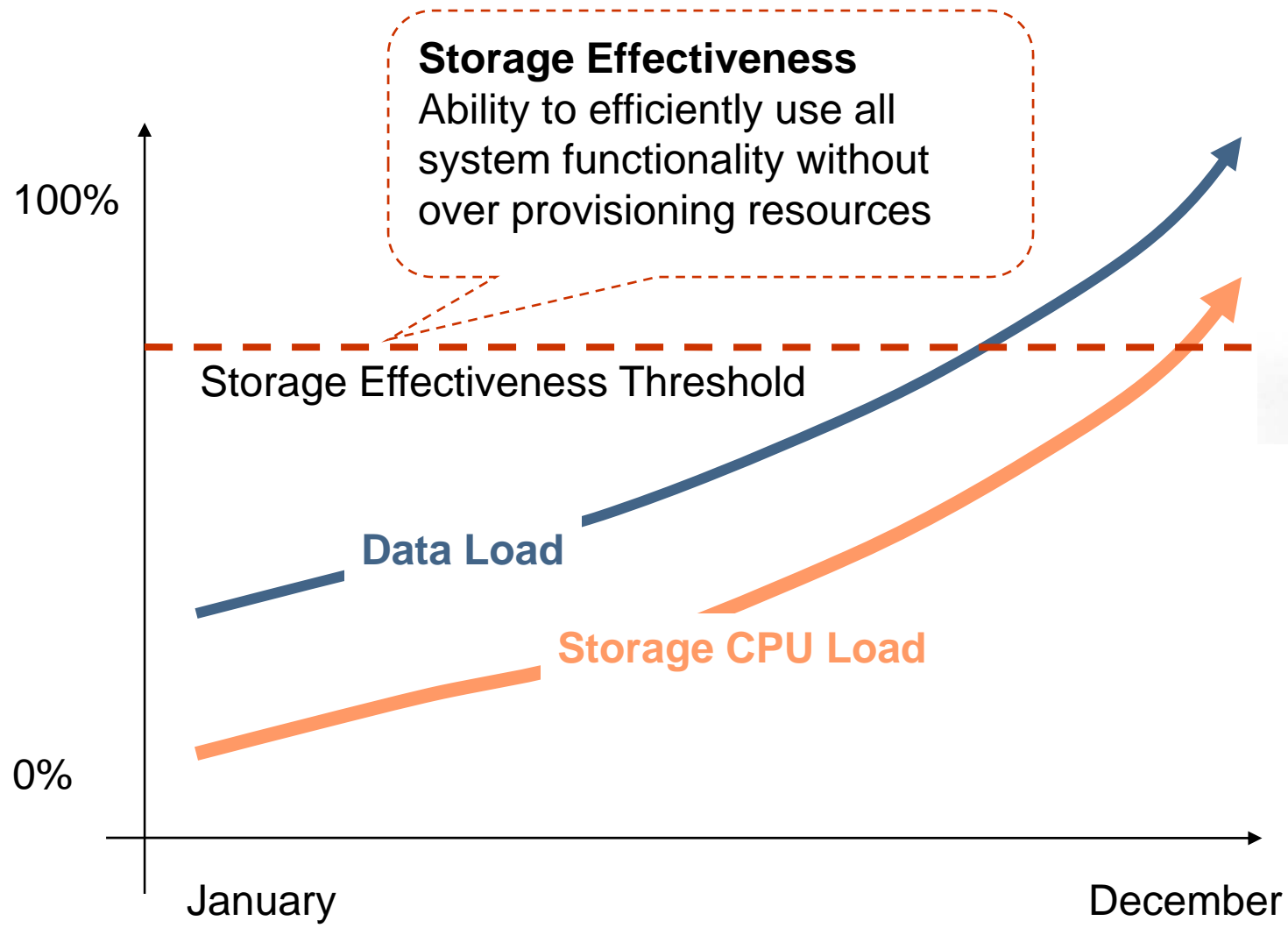
> 100K

1 billion

Searchable
videos by 2009

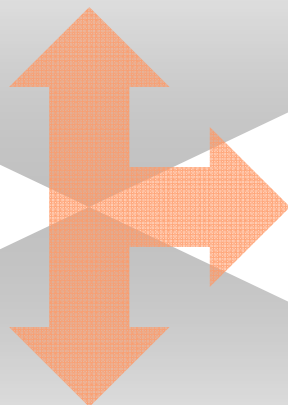
http://money.cnn.com/news/newsfeeds/articles/djf500/200809091346DOWJONESDJONLINE000554_FORTUNE5.htm
<http://www.flowgram.com/p/2qi3k8eicrfgkv/>
<http://www.searchenginejournal.com/truveo-forecasts-1-billion-searchable-online-videos-by-2009/6203/>

Data Load and Storage CPU Load

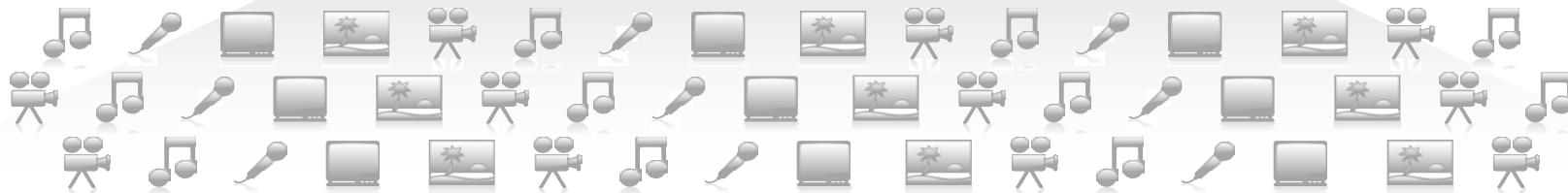


Warning Zone!

The Rise of Indexing Bottlenecks



**Common
Index
Overload!**



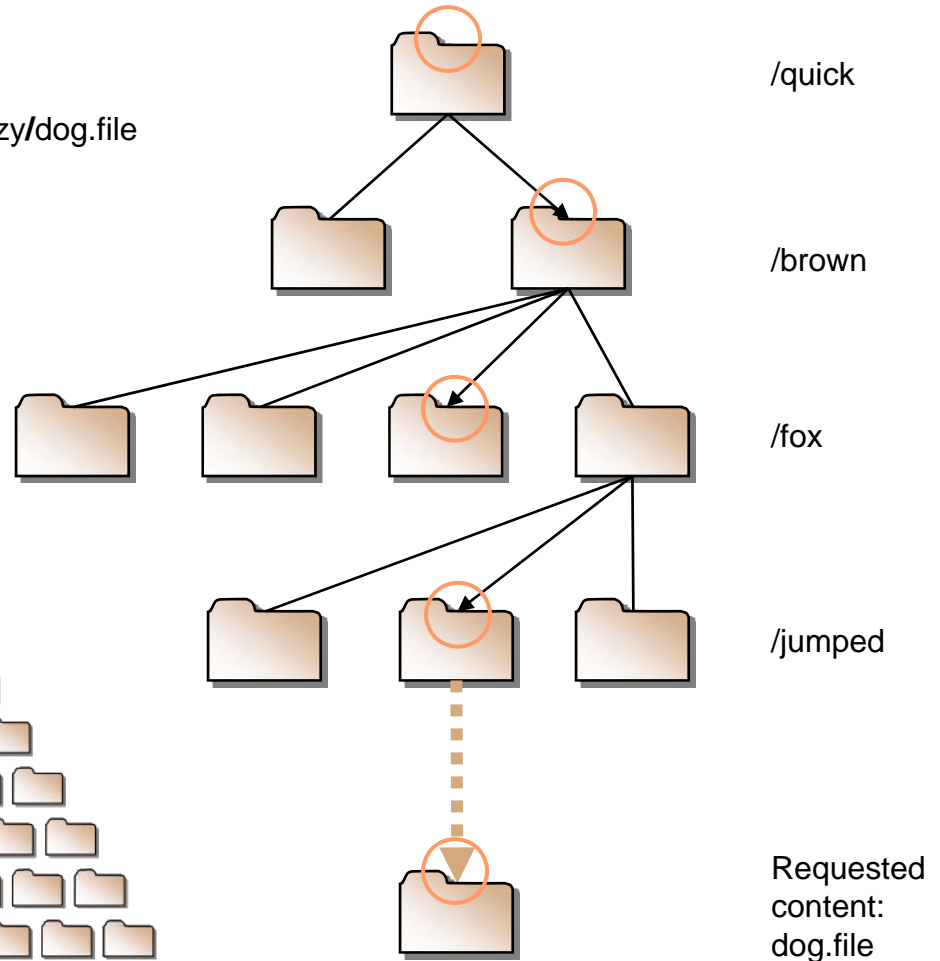
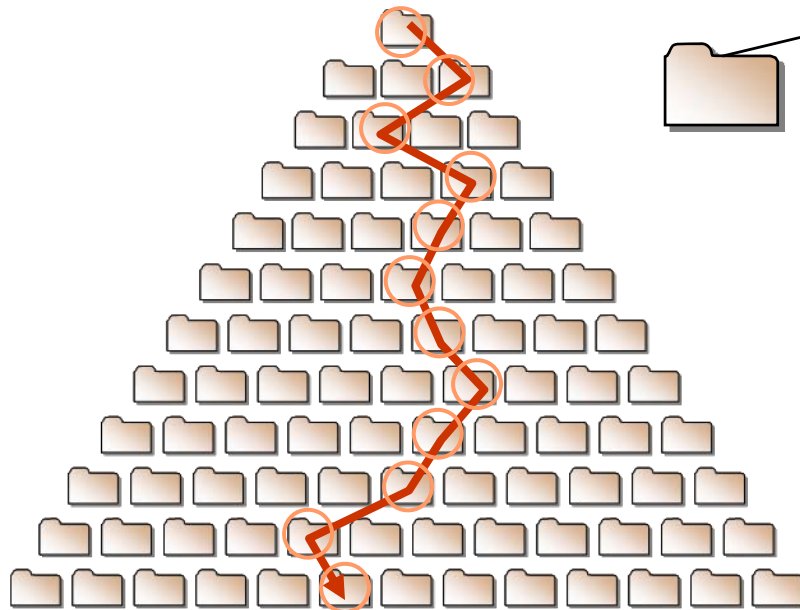
Walking the Directory Tree

Global namespaces can add to performance concerns

Sample NFS directory lookup

`/quick/brown/fox/jumped/over/the/lazy/dog.file`

○ Additional NFS operation



/quick

/brown

/fox

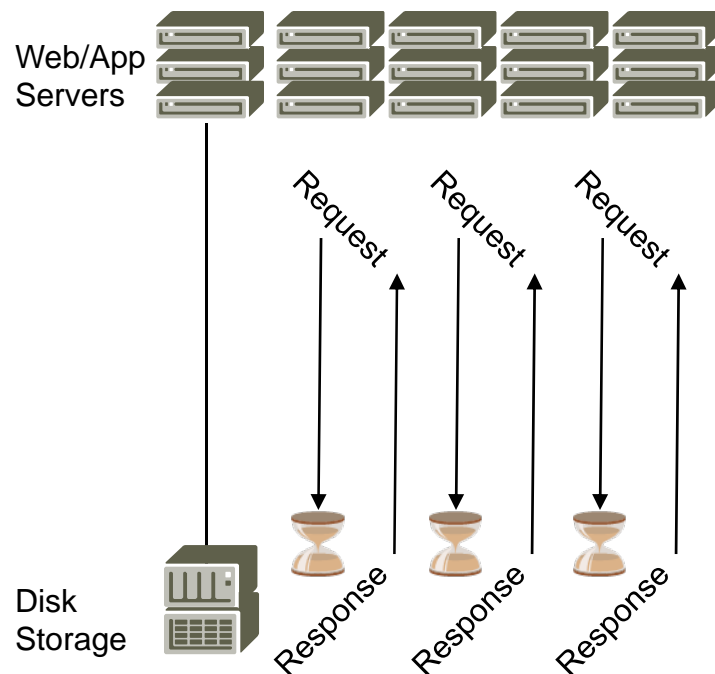
/jumped

Requested
content:
dog.file

The Impact of High File Counts

➤ Conventional Model

- ◆ Numerous metadata requests
- ◆ Lengthy response times
- ◆ Inability to scale the number of users



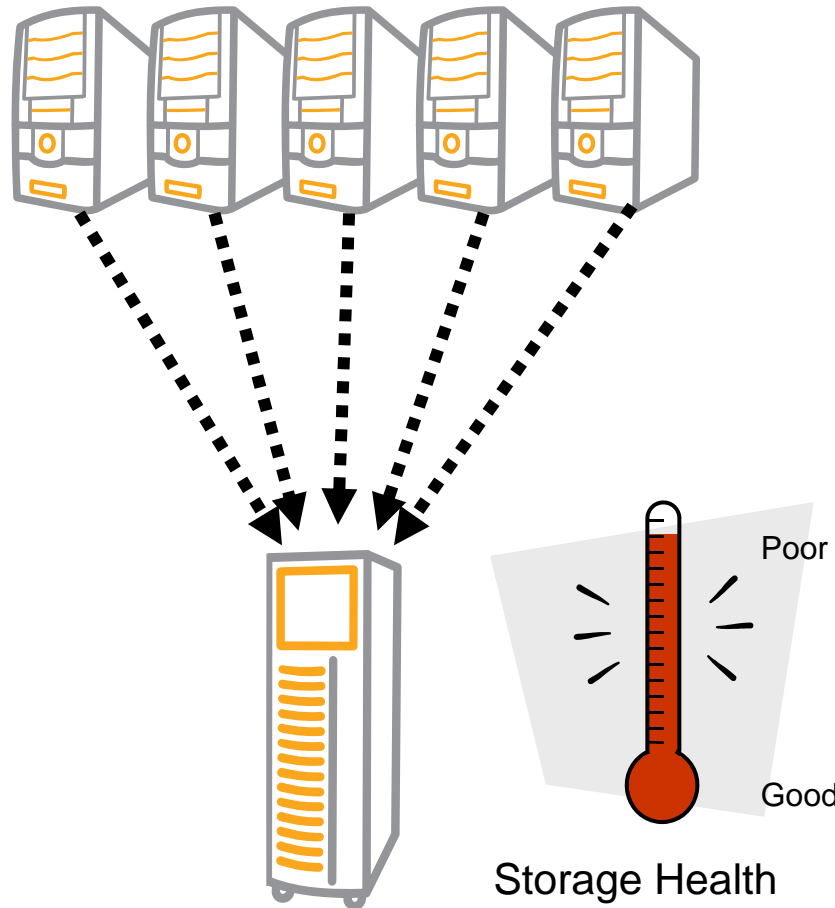
➤ Storage System Impact

- ◆ High CPU utilization
- ◆ Slow response times
- ◆ Inability to use all functionality
 - Snapshots
- ◆ Disk over provisioning
- ◆ System over provisioning

Three Case Study Scenarios

- Data warehousing
- Software development
- Web scale

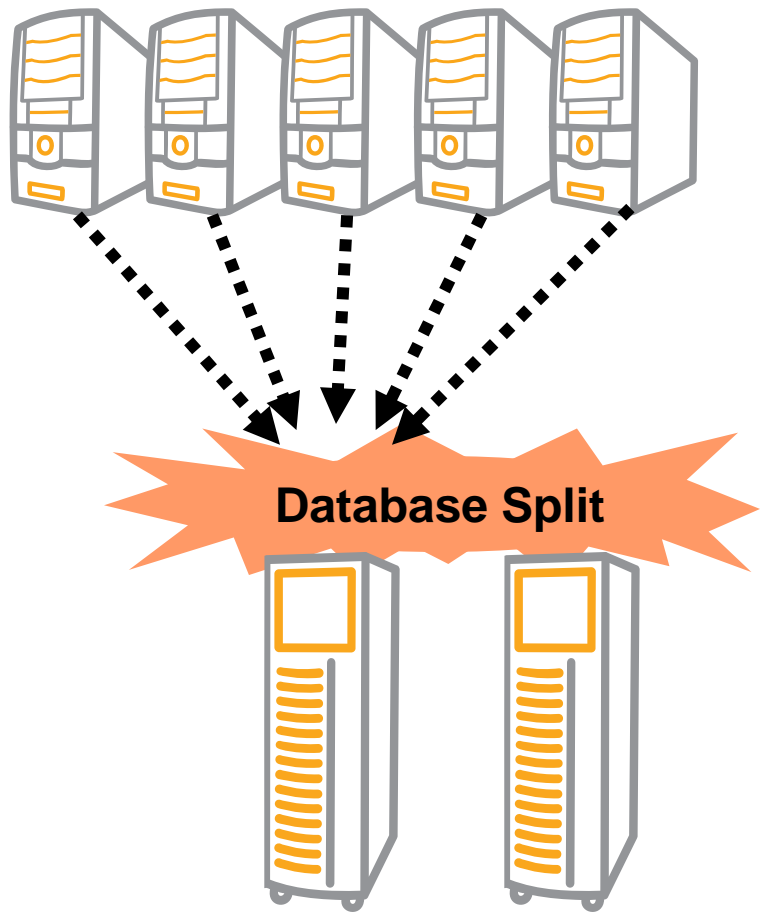
Enterprise Data Warehouse Configuration **SNIA**



➤ Current environment

- ◆ Many databases
 - > Large and small
 - > Highly active and less active
- ◆ Large number of concurrent users
 - > Access control and authentication mechanism in place
- ◆ Single storage repository streamlines management but is prone to bottlenecks

Enterprise Data Warehouse Configuration **SNIA**

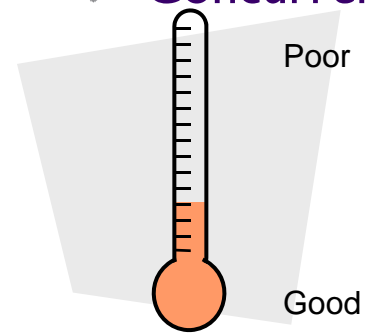


➤ Pros

- Reduce Single System Workload

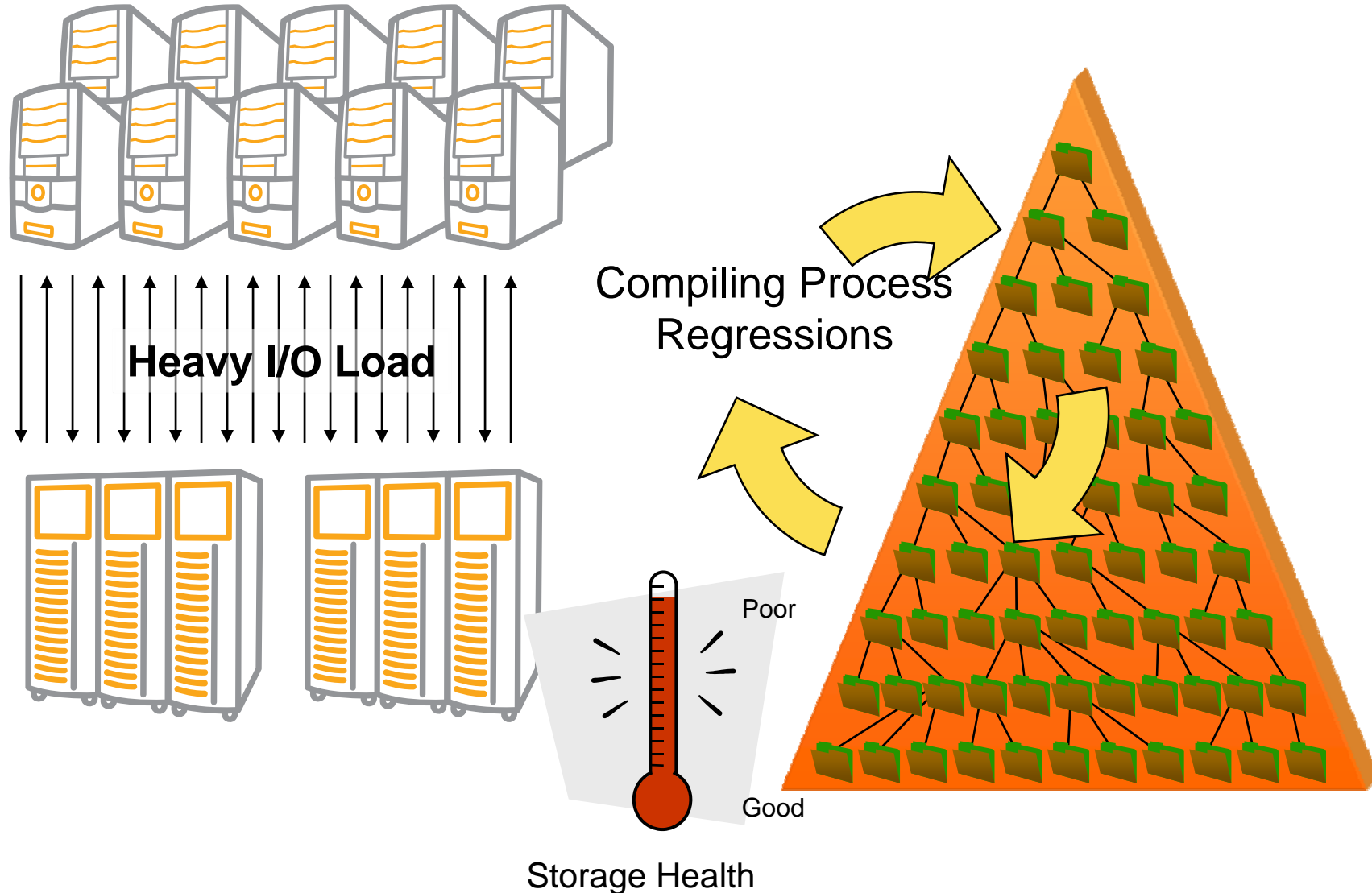
➤ Cons

- Pain to split database
- Excessive overhead / management
- Concurrency challenges

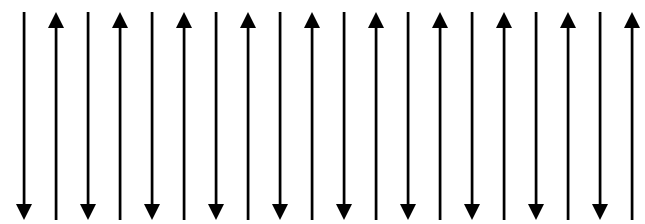


Storage Health

Software Development Bottlenecks

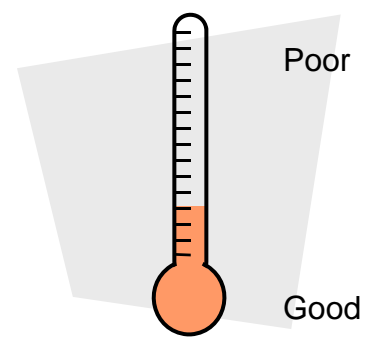


Software Development - Replicas



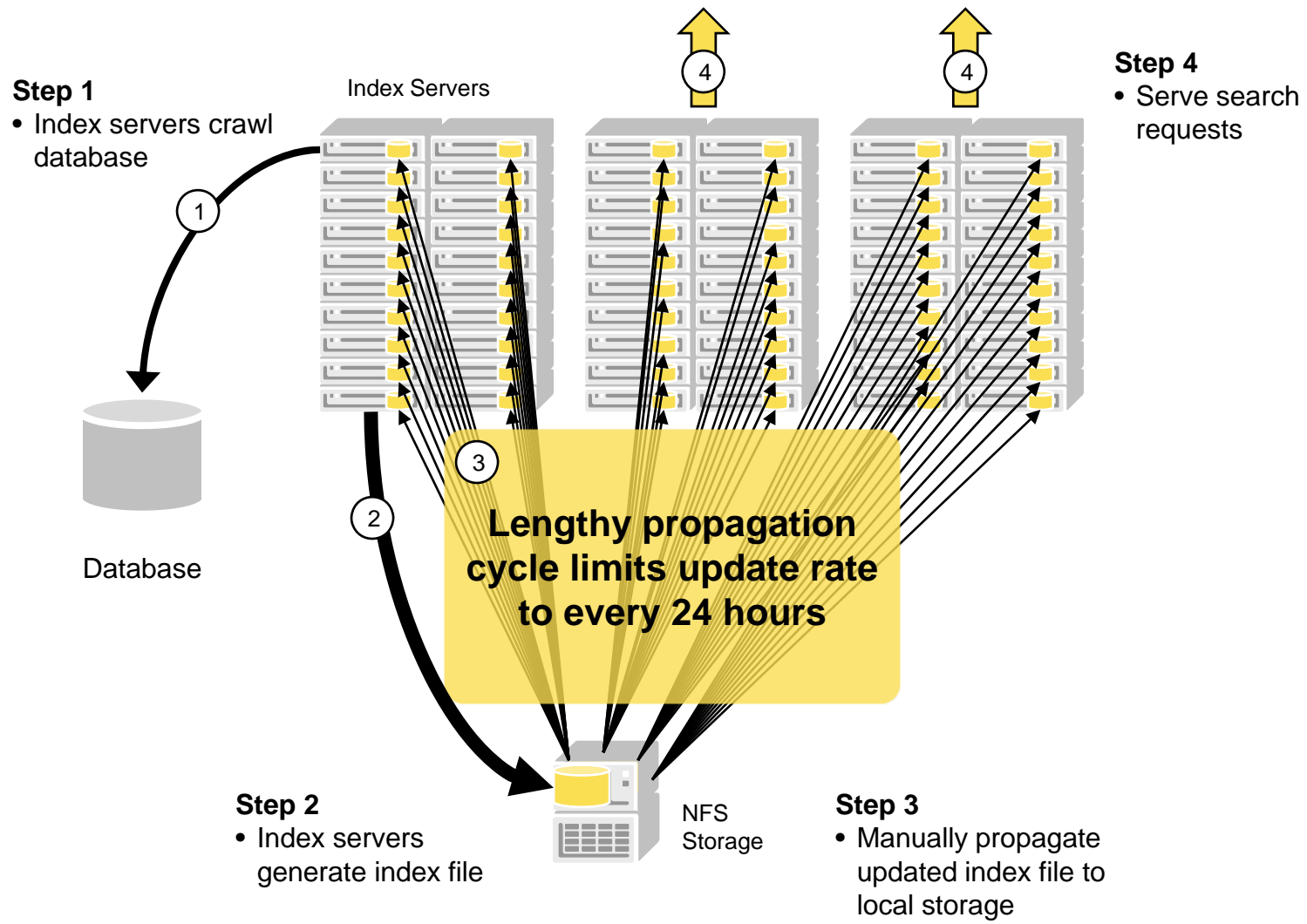
- Pros
 - Reduce storage CPU load
- Cons
 - Over-provisioned storage
 - Excess manual administration

Manually administered disk-based replicas



Storage Health

Web Scale Applications

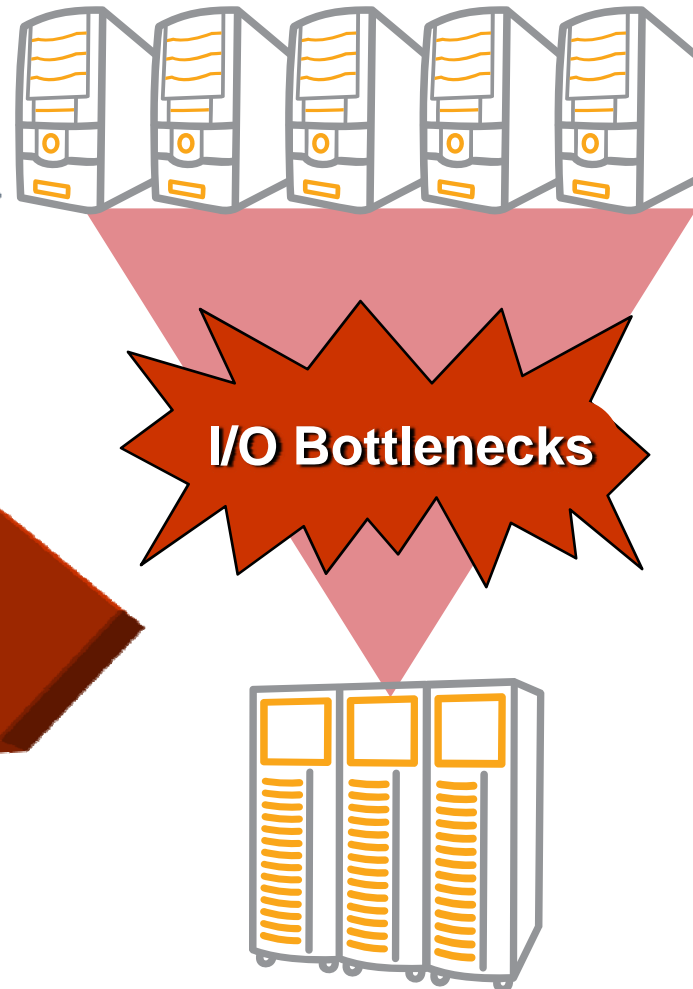


Current Trends Driving Increasing I/O Bottlenecks

Application traffic trends

- Shared I/O applications
- File-content explosion
- Web-scale applications
- Server virtualization

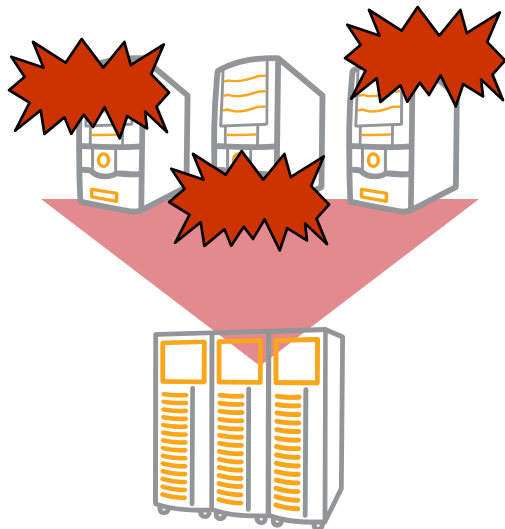
Current trends driving
painful storage problems



Current Ineffective Performance Approaches

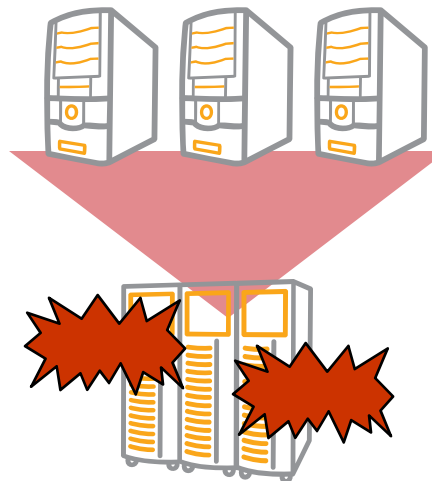
Client caching

- ◆ Limited capacity
- ◆ Inefficient
- ◆ Isolated



Subsystem caching

- ◆ Limited capacity
- ◆ Difficult to scale
- ◆ Resources anchored to each subsystem



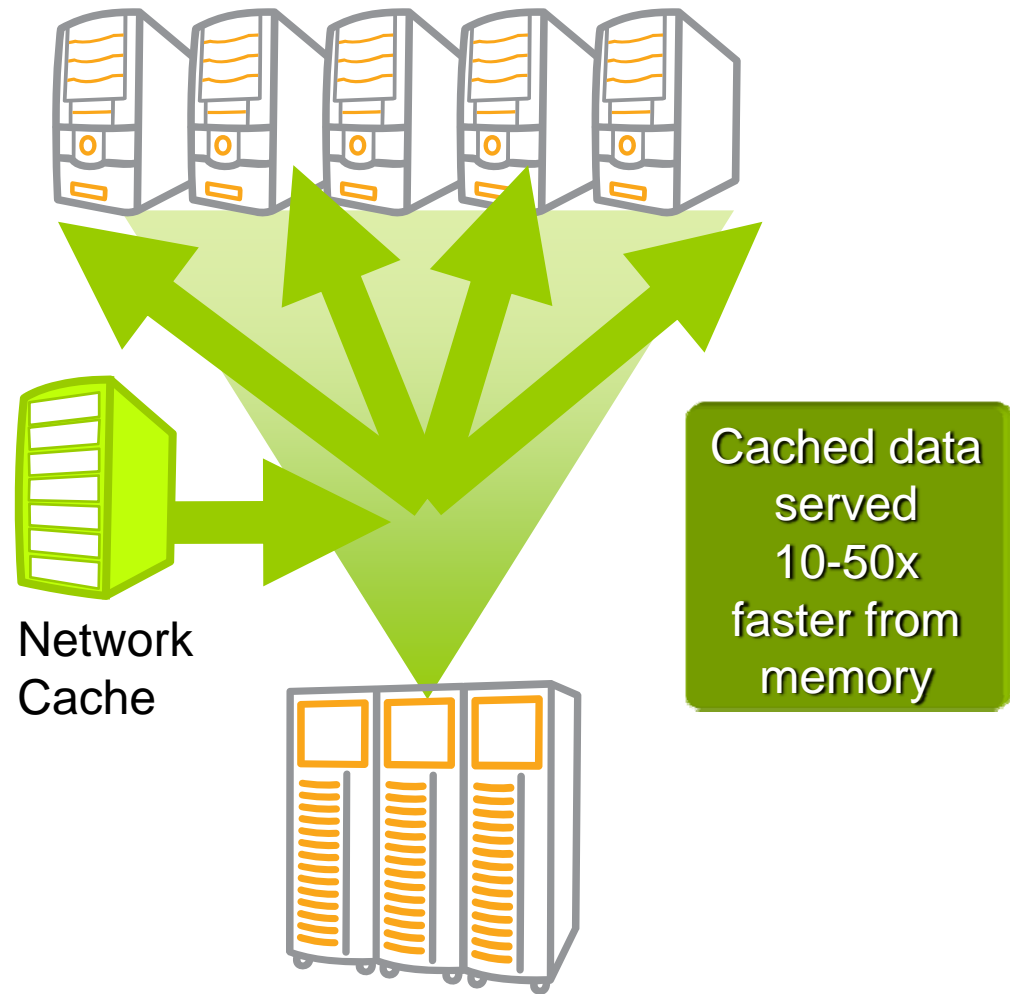
Over provisioning

- ◆ “Hot Spots”
- ◆ No latency reduction
- ◆ High CAPEX and OPEX



A Network-Centric Approach: Centralized Caching

- Increase performance
- Reduce total system costs
- Leverage existing infrastructure
- Scale easily



Solutions Needed At All Layers



Server



Networking



Storage

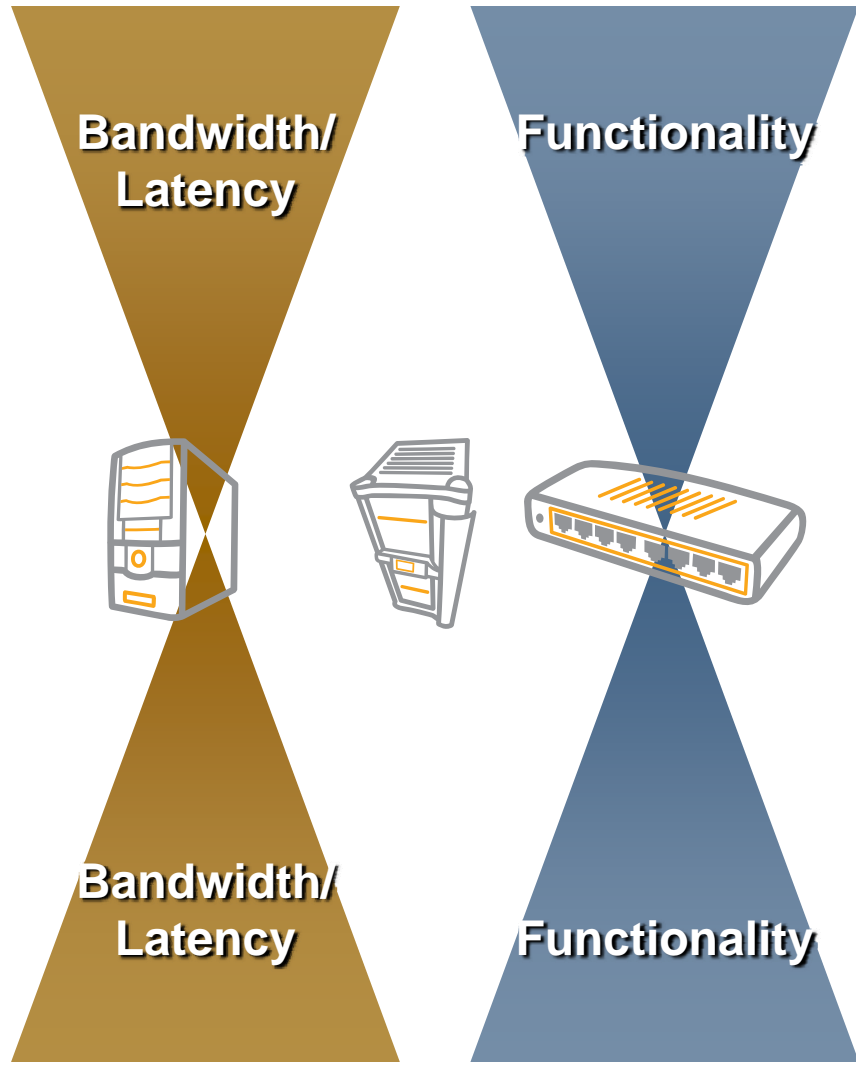
Server Layer – Application Scaling



Server

- Virtualization
- Parallelization
- Clustering

Networking Layer



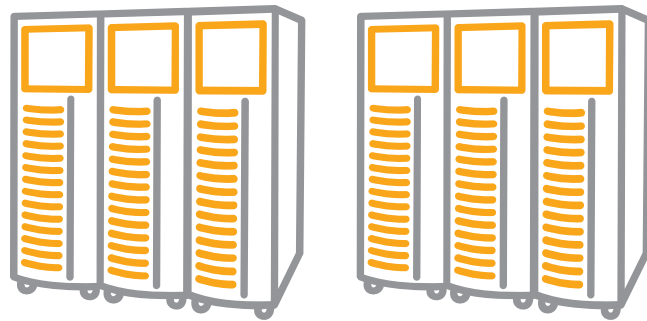
- Bandwidth
- Latency

Networking

- File Acceleration
- Load Balancing
- File Access Optimization

Storage Layer

- Scalable File Systems
- Parallel / Clustered
- Global Namespaces
- Persistence
- Protection



Storage

Why Are We Here?

➤ Typical Data Center

- ◆ Lots of Servers with Lots of Processors and Cores



- ◆ Lots of Disk Drives with Rotating Mechanical Media



- ◆ **SOMETHING HAS TO CHANGE!**

Data Center Memory Options

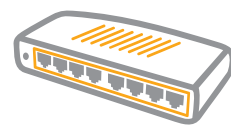
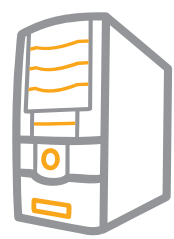


Servers

Appliances

Network Devices

Storage Systems

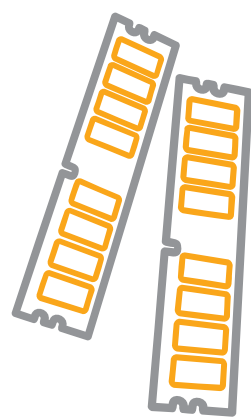
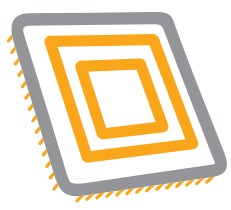
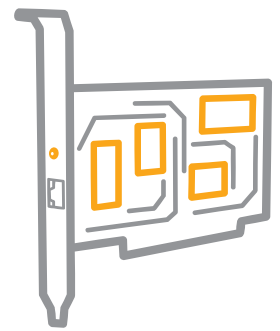


PCI Cards

Processors

Memory Modules

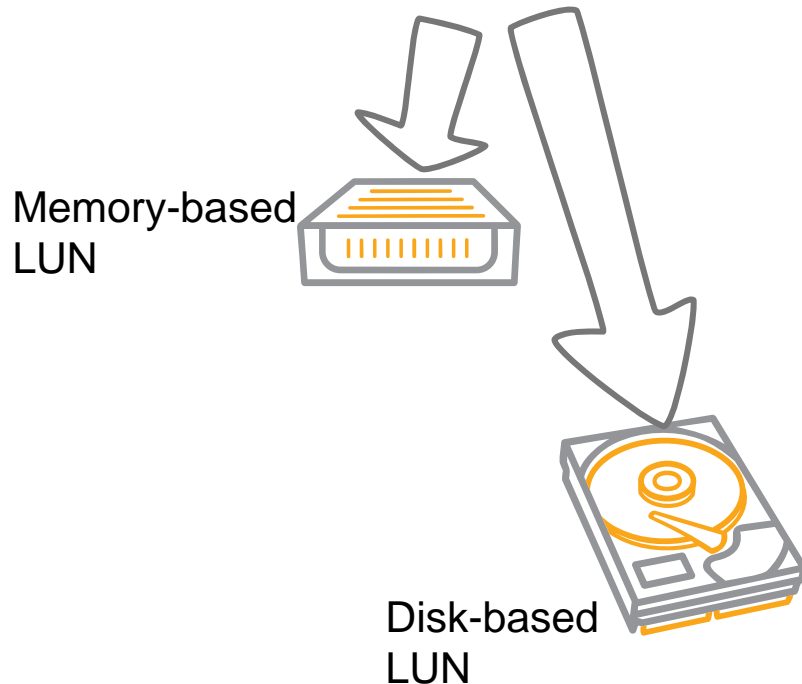
SSDs



Ways to Use Memory

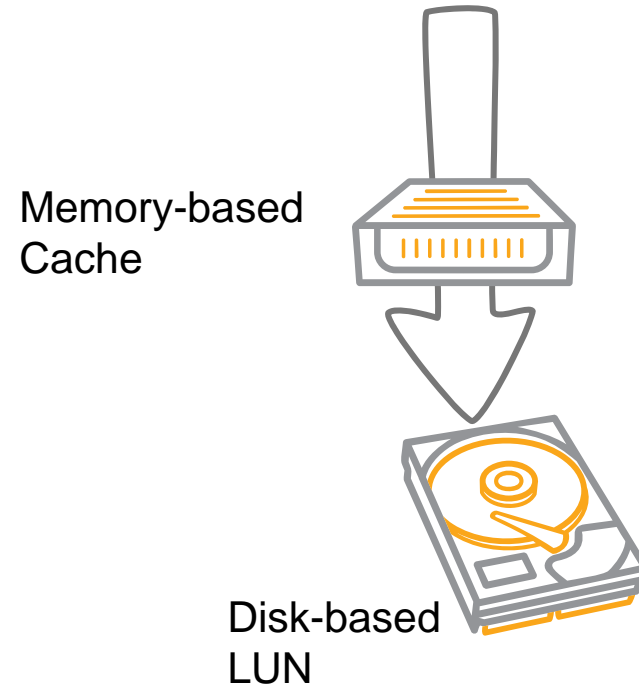
➤ Memory as Disk

- ◆ Individual host-visible LUN
- ◆ Actively managed storage
- ◆ Manual or software-assisted active data extraction



➤ Memory as Cache

- ◆ Transparent view
- ◆ Passively managed
- ◆ Automatic caching of active data set



Making Use of Near-Infinite Disk Capacity **SNIA**

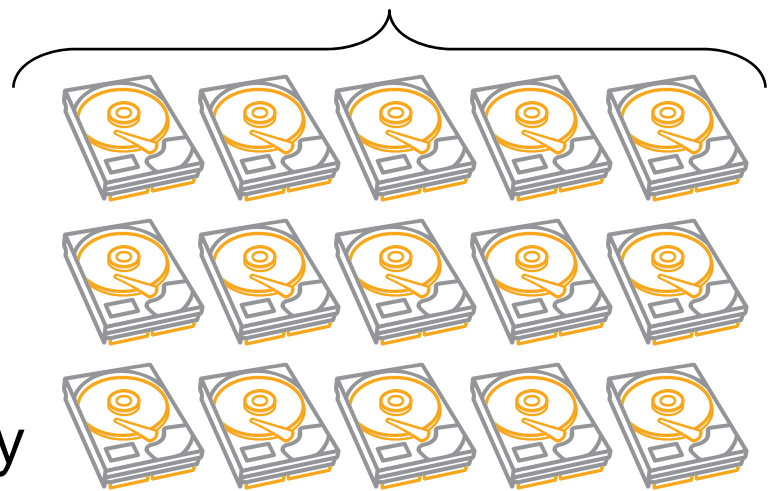
Memory-based LUN



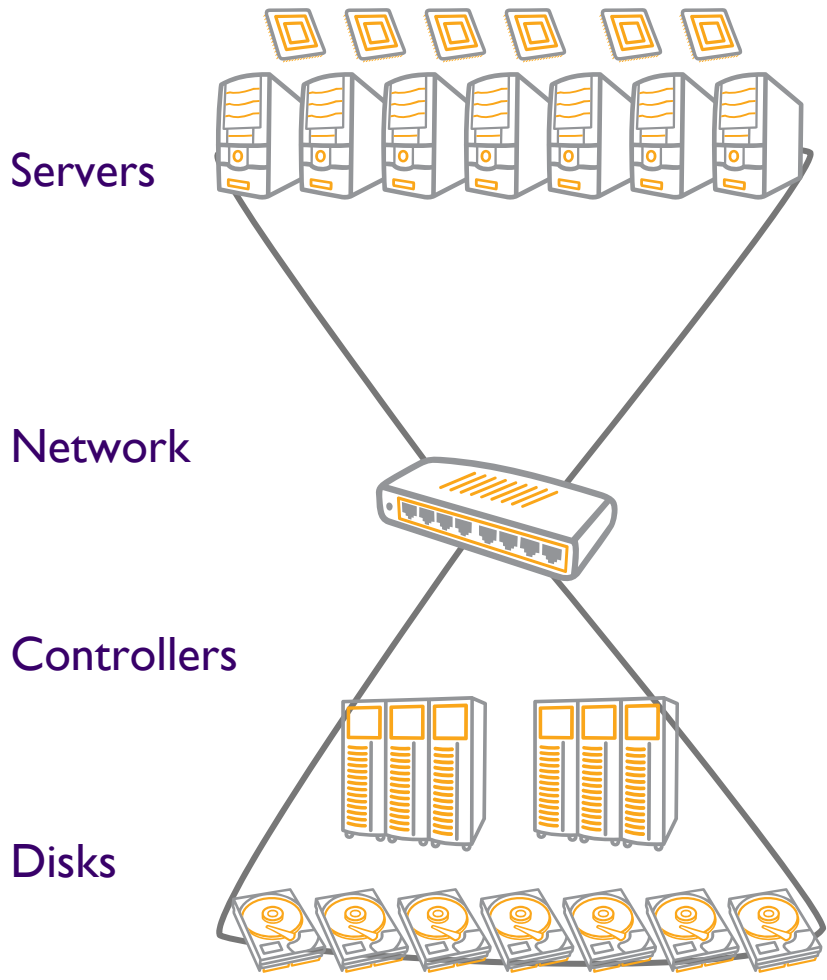
Memory-based Cache



Near-Infinite Disk-based Capacity



Where to Cache



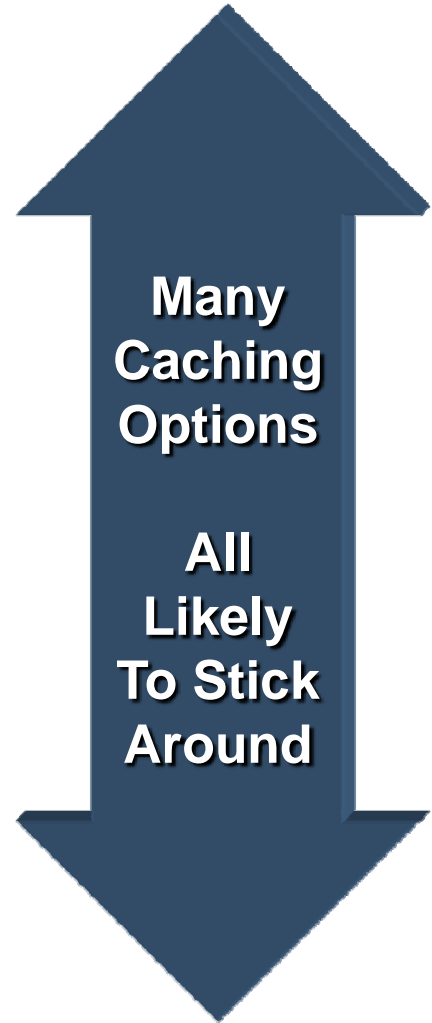
LI, L2, L3 Cache

Server Cache

Network Cache

Controller Cache

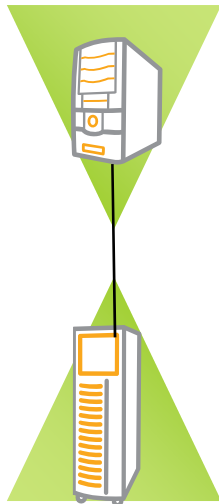
Disk Cache



Comparing Cache Locations

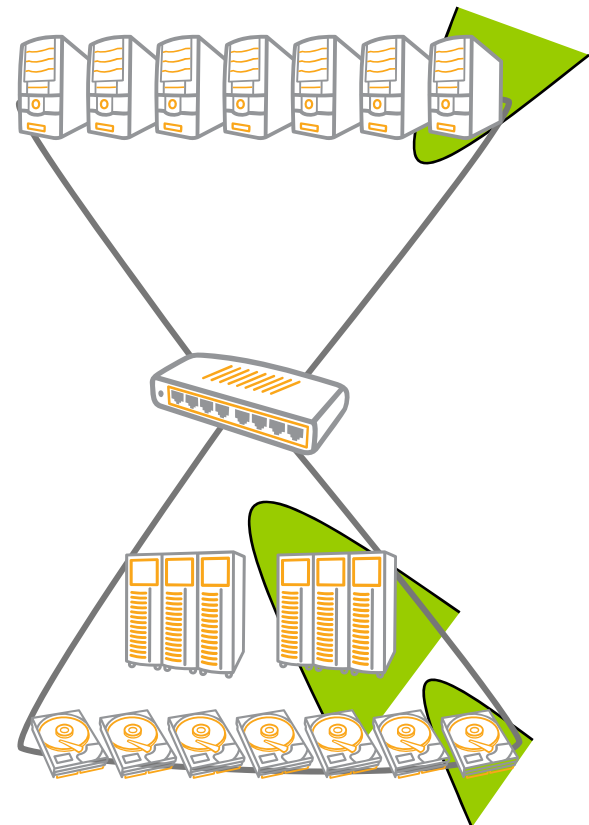
➤ Low Device-Count Configurations

- ◆ Server or storage caching provides comprehensive reach



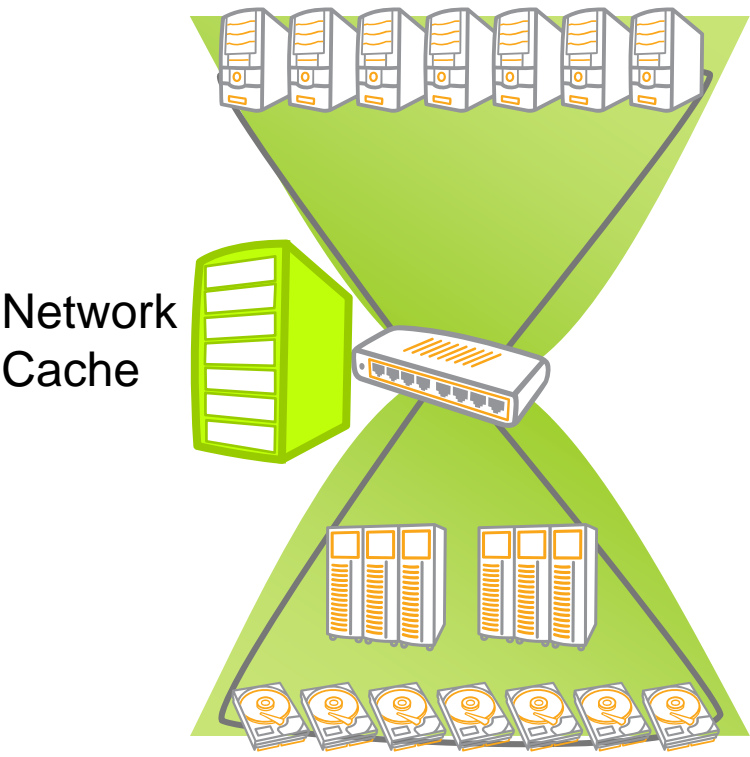
➤ Multiple Device Configurations

- ◆ Server or storage caching provides limited reach

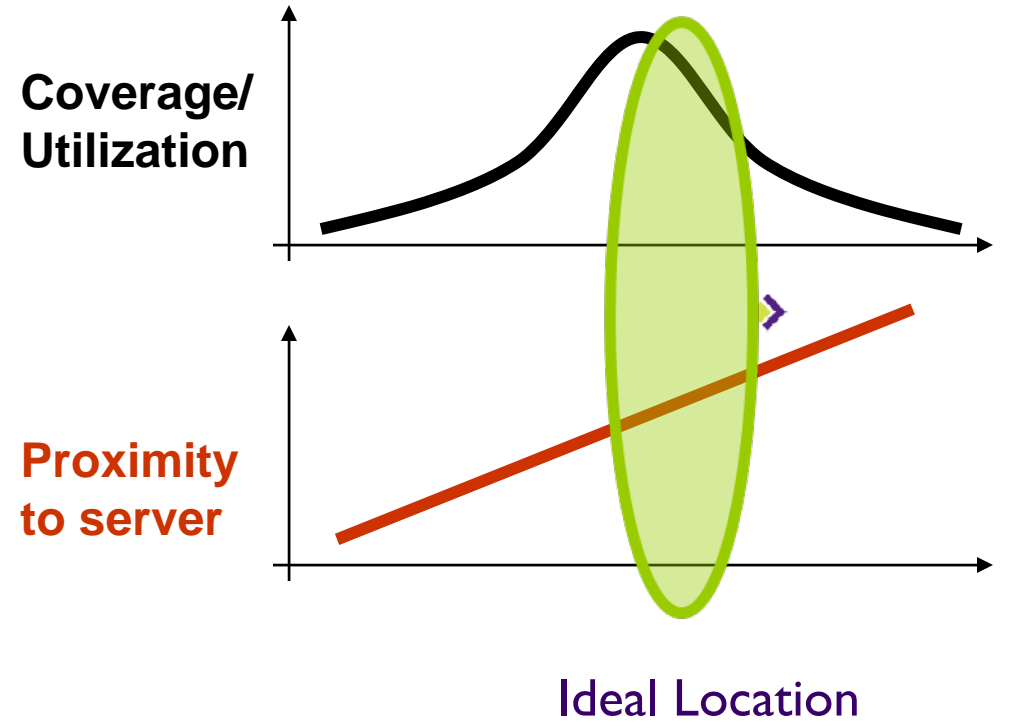


Advantages of Network Caching

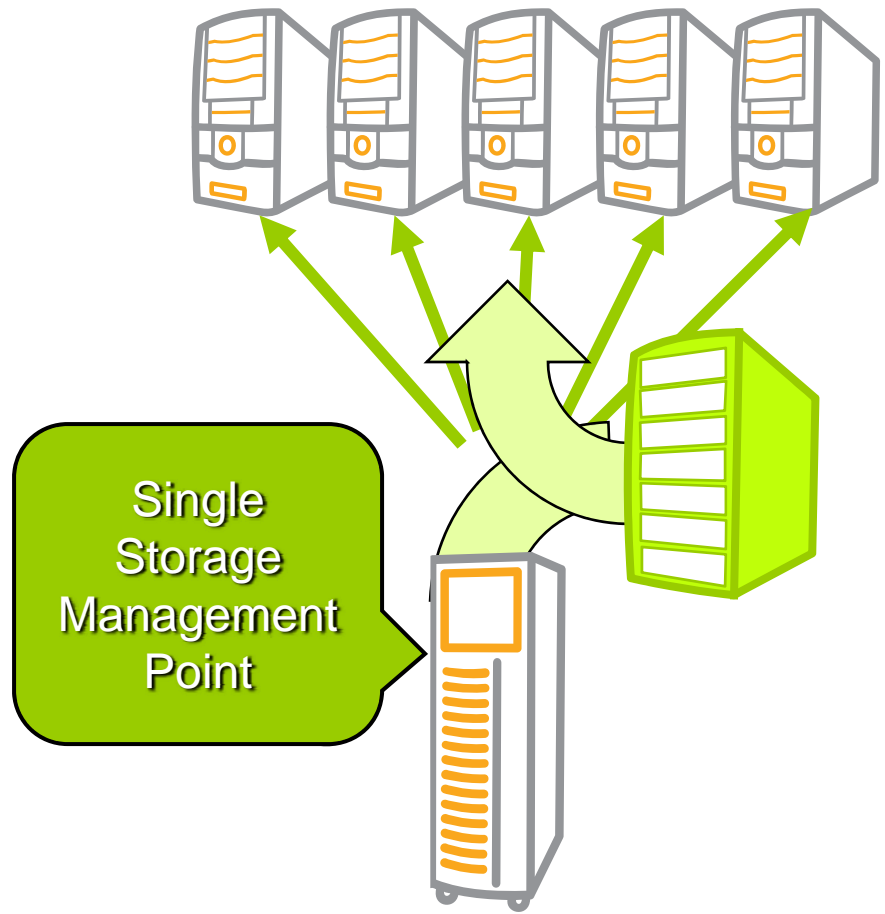
- Network Caching for Multi-Device Configurations
 - ◆ Maximum effectiveness and efficiency



- Optimizing Cache Locations

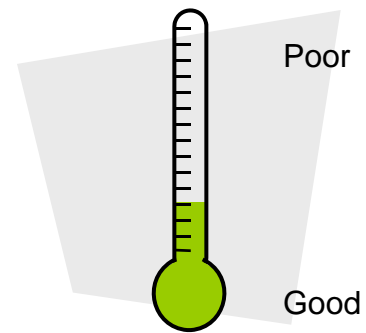


Enterprise Data Warehouse Solution SNIA



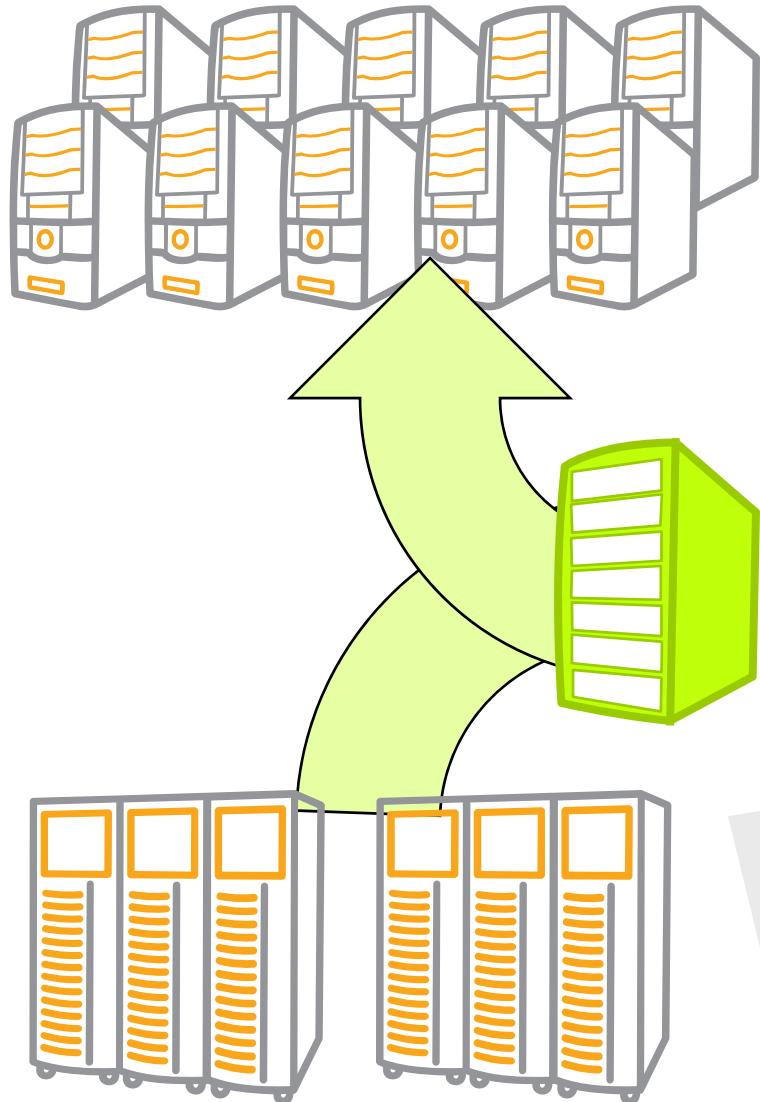
➤ Pros

- ◆ Single system, streamlined management
- ◆ Network caching for peak load handling



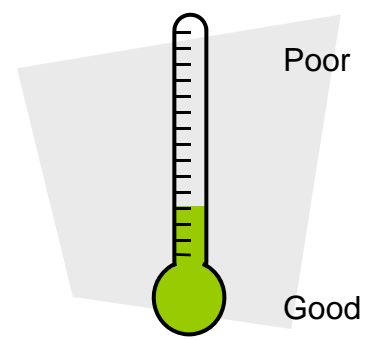
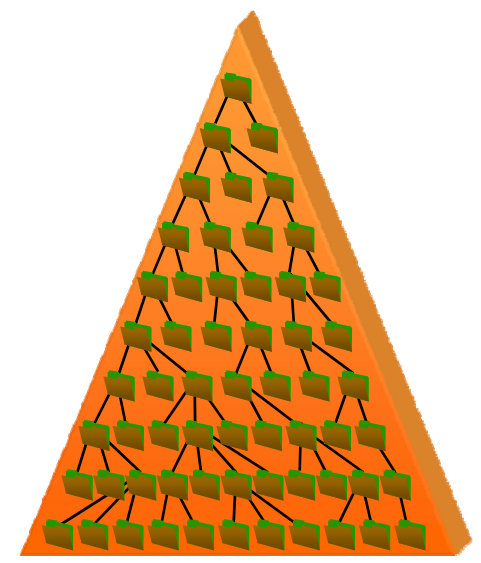
Storage Health

Software Development Solution



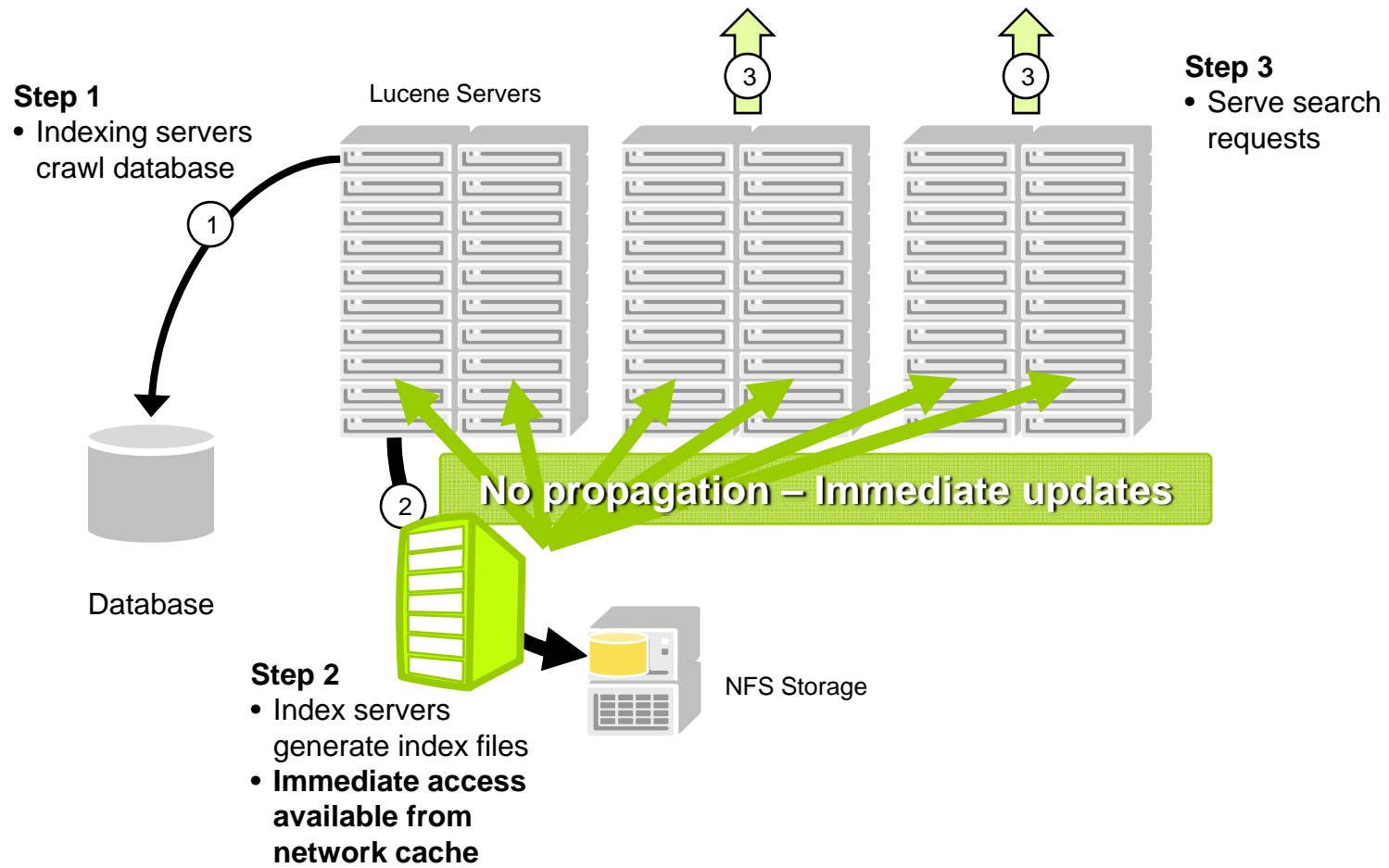
➤ Pros

- Memory-based, network caching for handling of small file and metadata requests



Storage Health

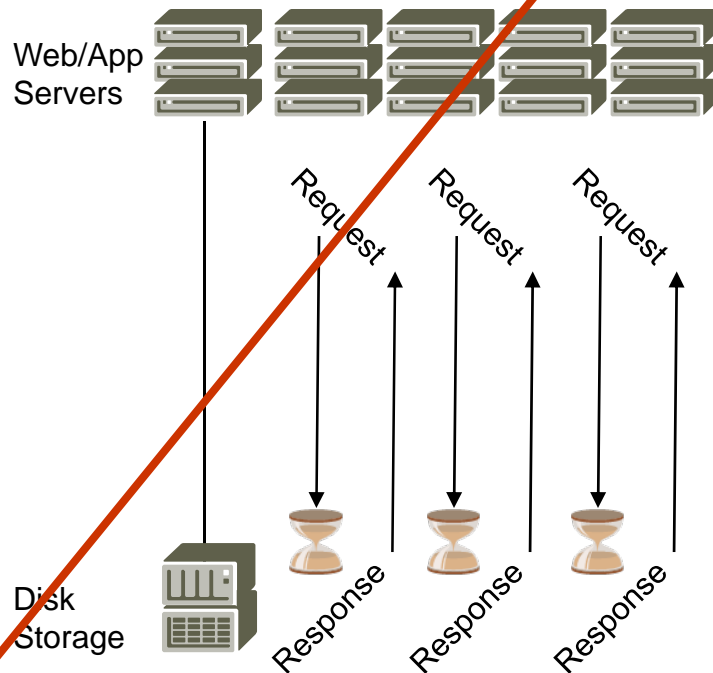
Web Scale Application Solution



Controlling High File Counts

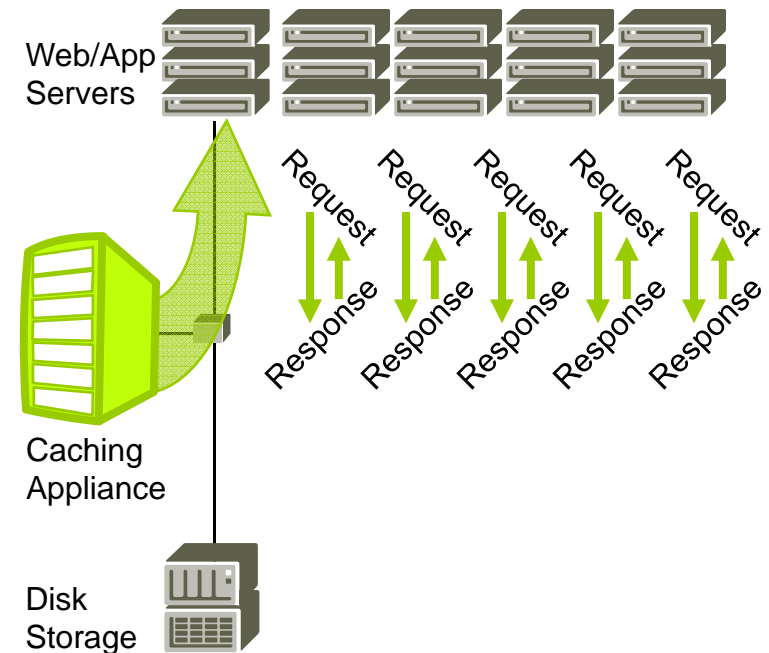
➤ Conventional Model

- ◆ Numerous metadata requests
- ◆ Lengthy response times
- ◆ Inability to scale the number of users



➤ Centralized Caching Model

- ◆ Cache frequent requests
- ◆ Immediate response times
- ◆ Accelerate existing infrastructure performance



- Please send any questions or comments on this presentation to *SNIA: Application Track*

**Many thanks to the following individuals
for their contributions to this tutorial.**

- SNIA Education Committee

**Josh Tseng, Track Lead
Rob Peglar**