File Caching in a SAN Environment

Mark W. Hall

Solid Data Systems, Inc.
2945 Oakmead Village Court
Santa Clara, CA 95051

Voice: 408.845.5747
Fax: 408.727.5496
mhall@soliddata.com

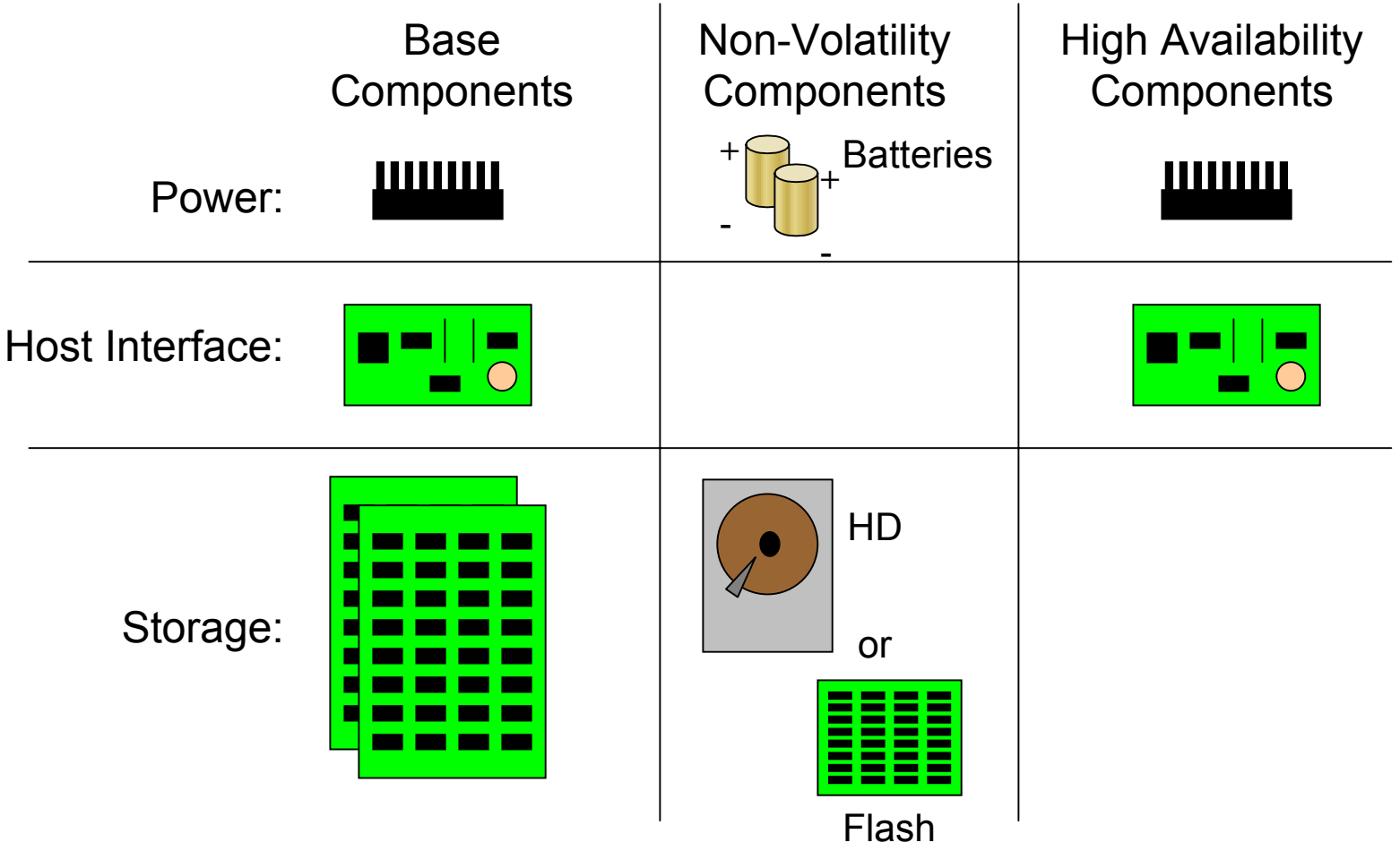
Overview

- Introduction to Solid-state File Cache
- Brief review of SAN
- Understanding Transactions & Latency
- Performance Effects of Solid-state File Cache
- Applications of Solid-state File Cache in SAN
- Features to look for in Solid-state File Cache
- Q&A

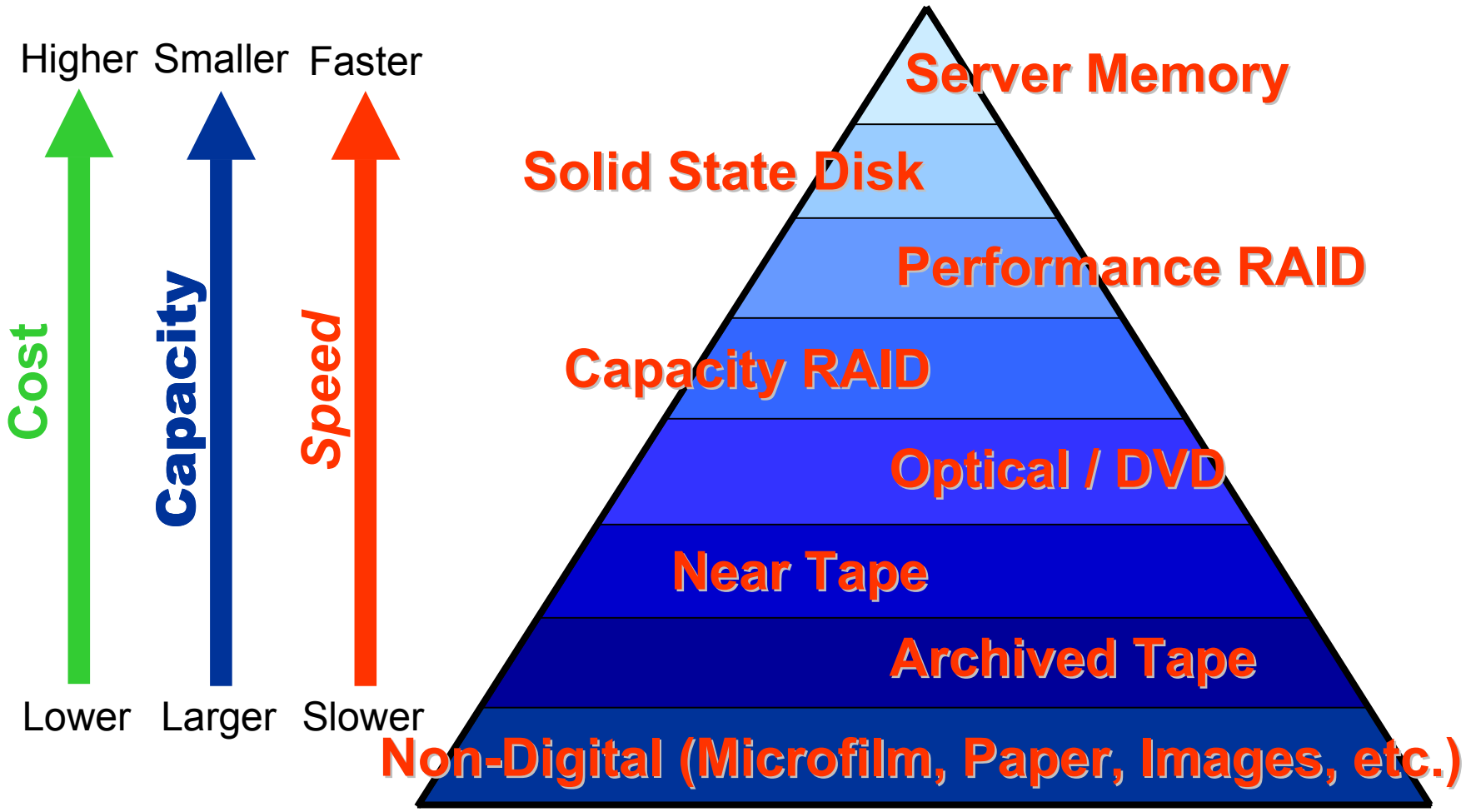
Benefits of Solid-state File Cache in I/O Constrained SAN

- Multiply performance and scalability of existing servers by 200%-800%+
- Reduce capital expenditures for additional servers by up to 80%+
- Better utilize highly qualified DBA and IT professionals
- Extend useful life of existing RAID

Solid-state File Cache Components



Storage Hierarchy



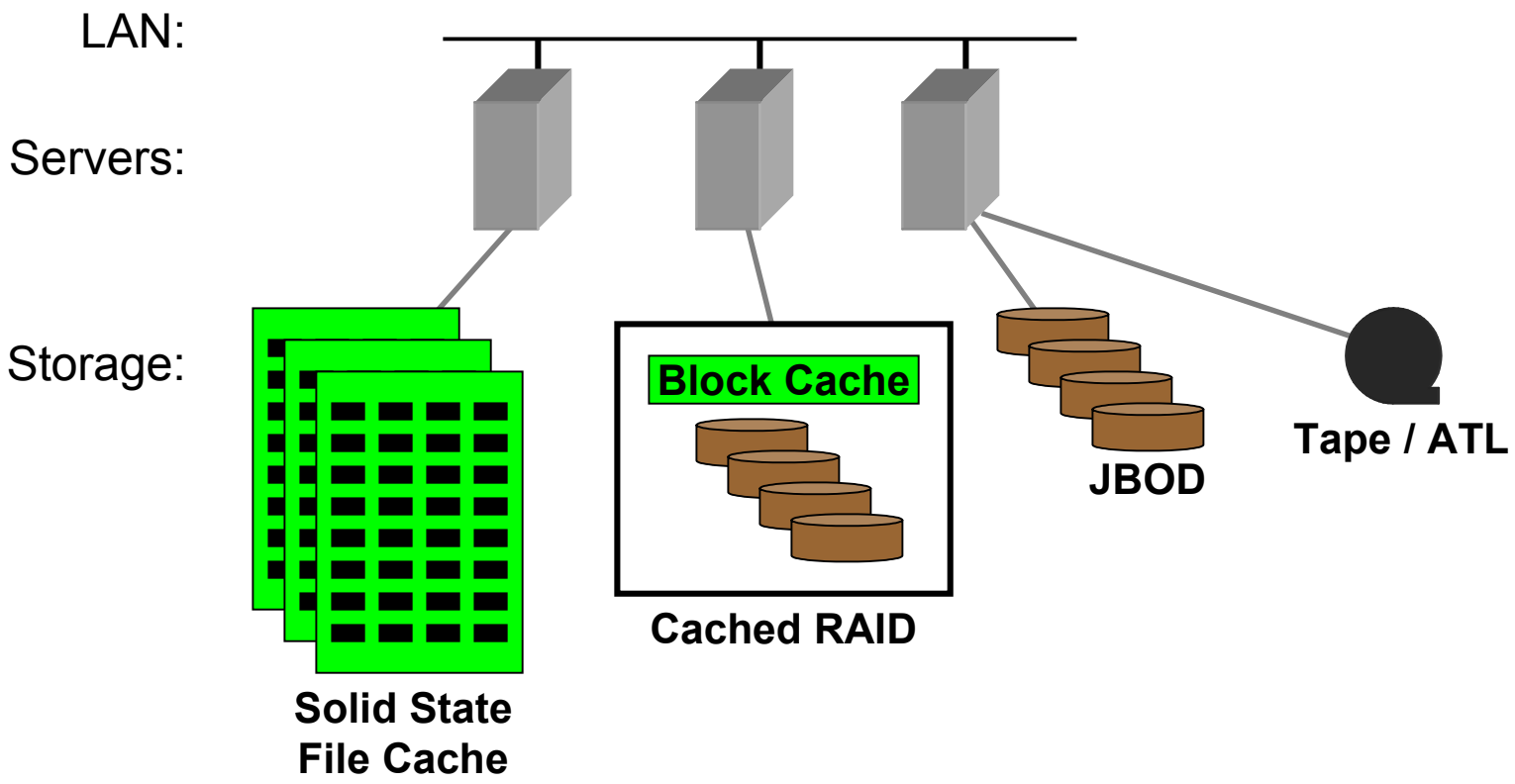
Source: Fred Moore, Horison Information Strategies; 1999

File Caching vs Cached RAID

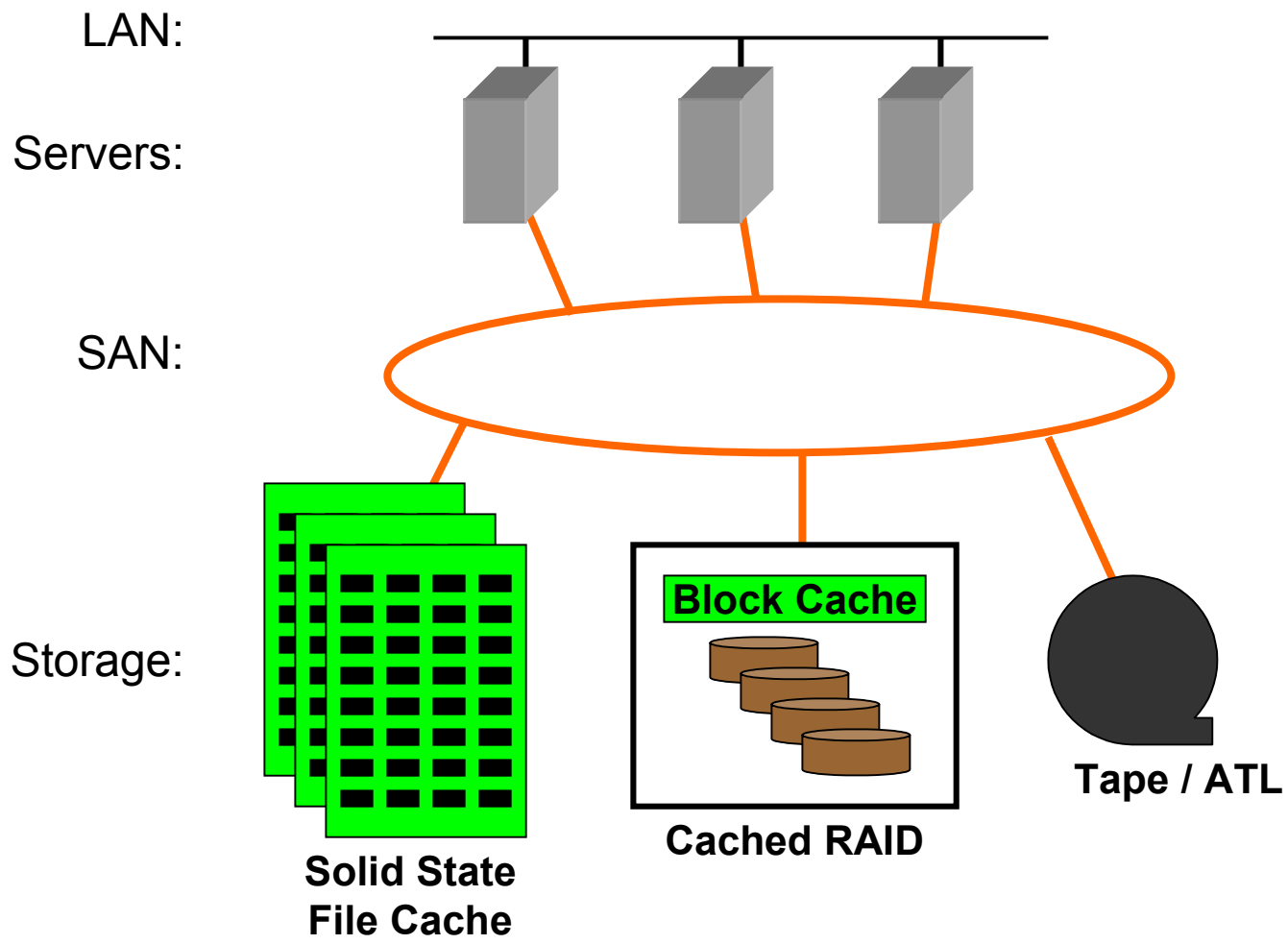
- Entire file in RAM
- Latency as low as 0.014 ms
- Consistent performance
- No tuning required
- Best for small block random I/O
- No moving parts - very high MTBF
- Independently scalable

- Selected blocks in RAM - other blocks on disk
- Latency of 7.000 ms or greater
- Performance depends on cache algorithms
- Extensive tuning sometimes required
- Best for large block streaming I/O
- Mechanical complexity reduces MTBF
- Limited cache to RAID ratio

Server Connected Storage



Networked Storage



Benefits of SAN

- **Today**
  - Heterogeneous access to storage device
  - Elimination of stranded capacity
  - LAN-free / server-free backup
  - Reduced TCO through centralization of management functions
  - Longer connection distances
- **Future**
  - Heterogeneous access to files
  - Virtualization of storage
  - Dynamic reallocation of capacity
  - Automatic “hot file” detection and movement to faster storage

Do you need a SAN?

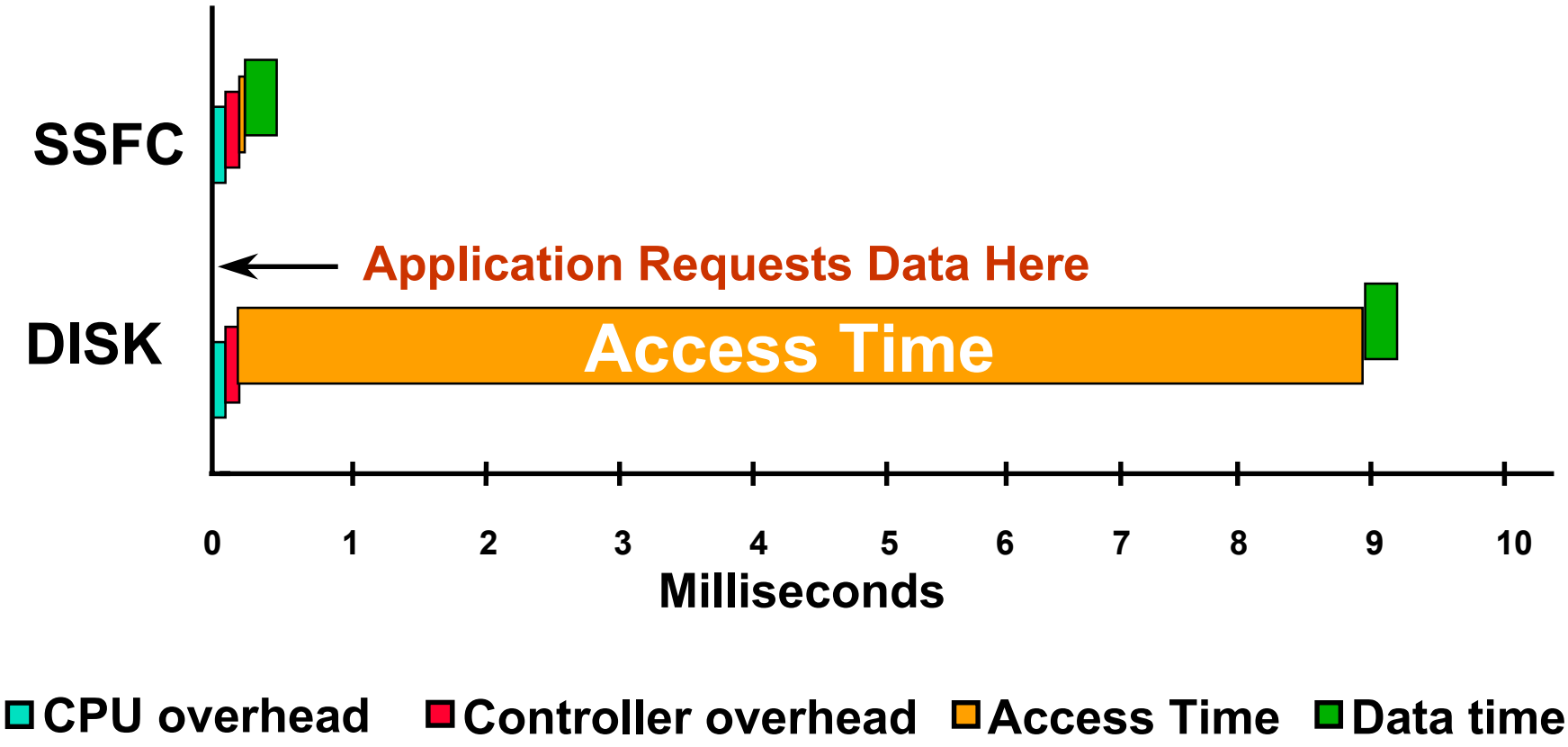
- 50%+ unused storage capacity in distributed servers?
- Backup operations impacting application performance?
- Backup window interfering with operational hours?
- Spending more on storage management than on the storage itself?
- Need longer connection distances?
- Need more data access bandwidth?

Transaction

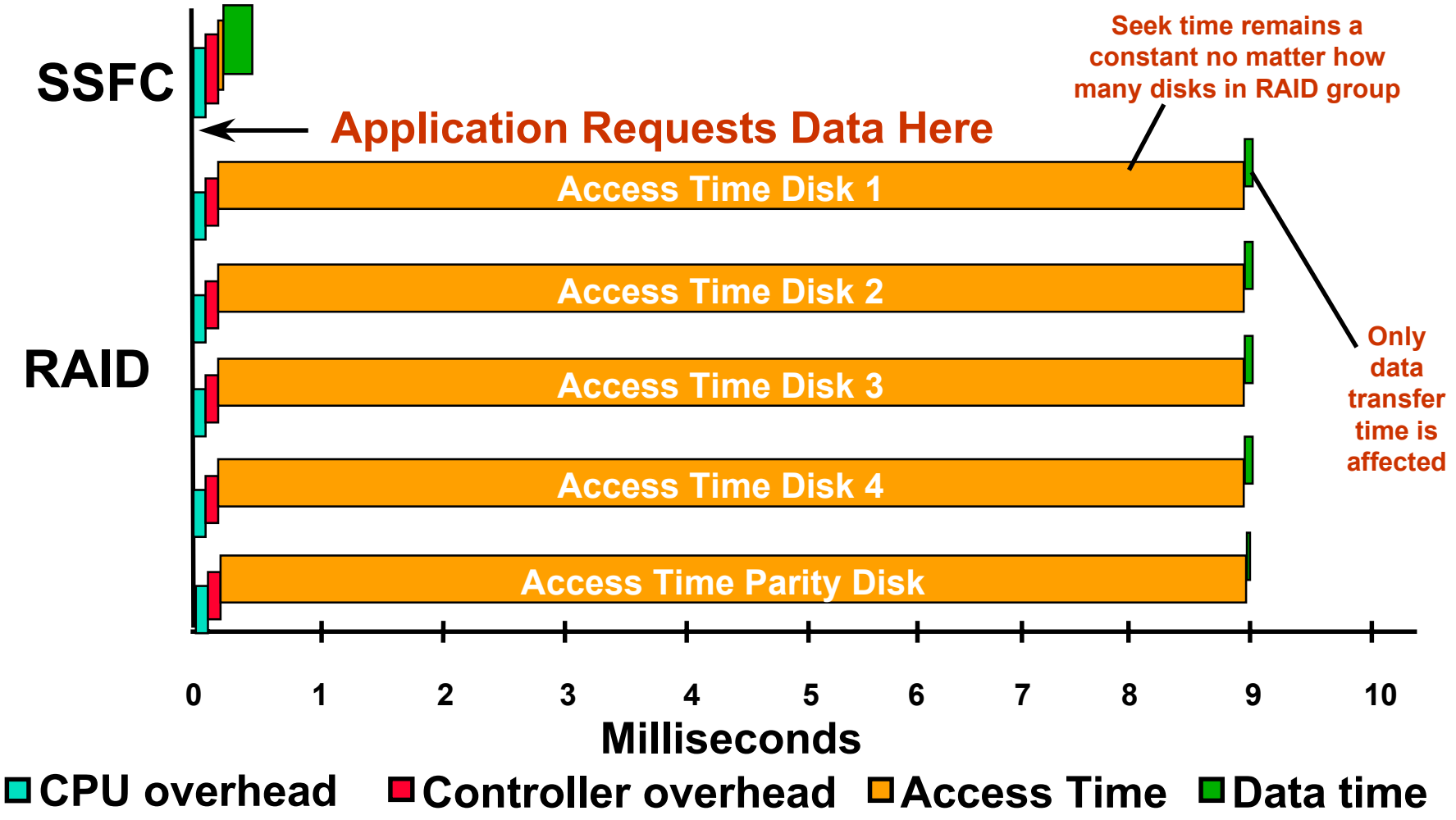
- Is an application specific unit of work
  - Dynamic Web page generation
  - E-mail message
  - Credit card electronic verification
  - Database insertion
  - Etc.

Each transaction may represent *dozens* to *hundreds* of disk I/Os

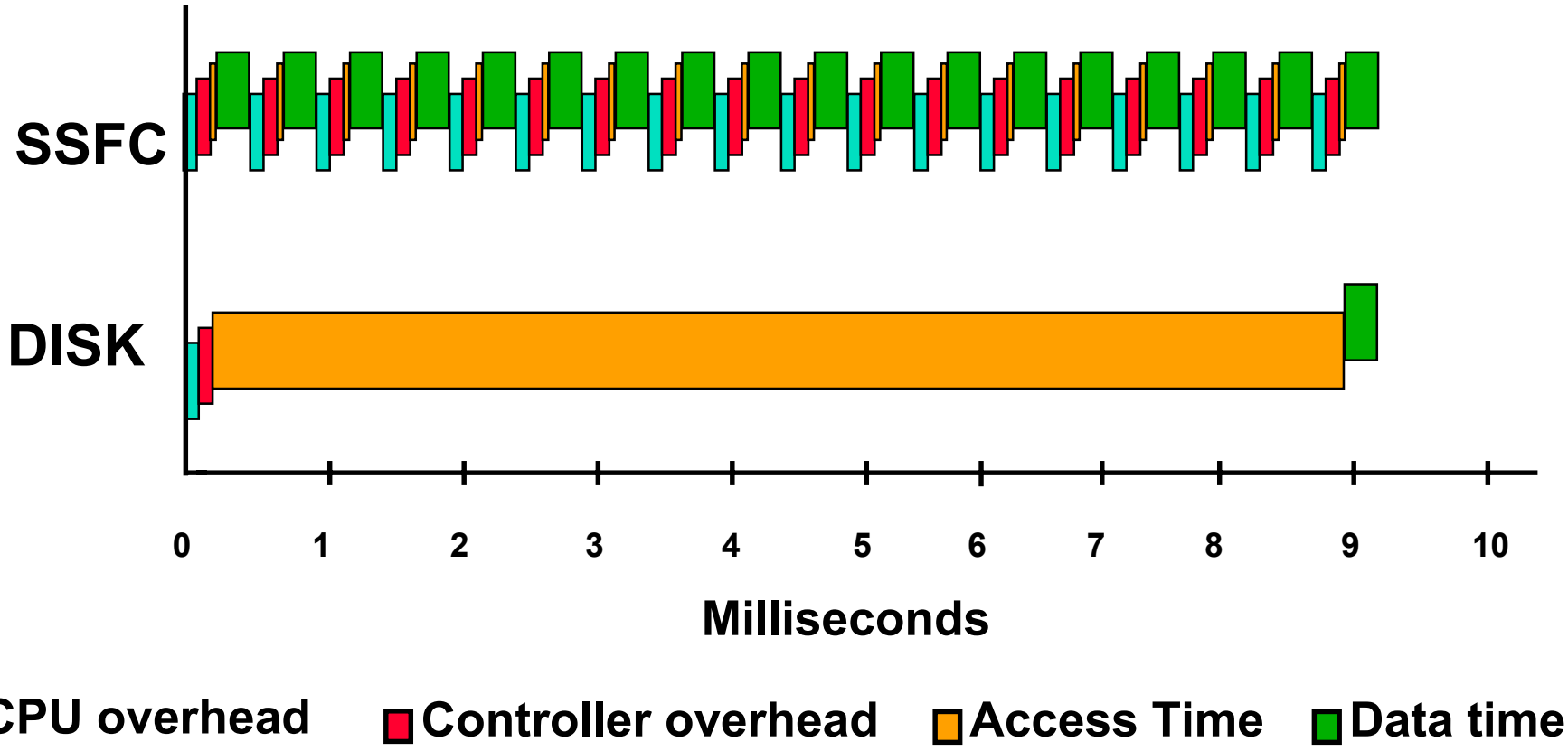
Latency During Data Access



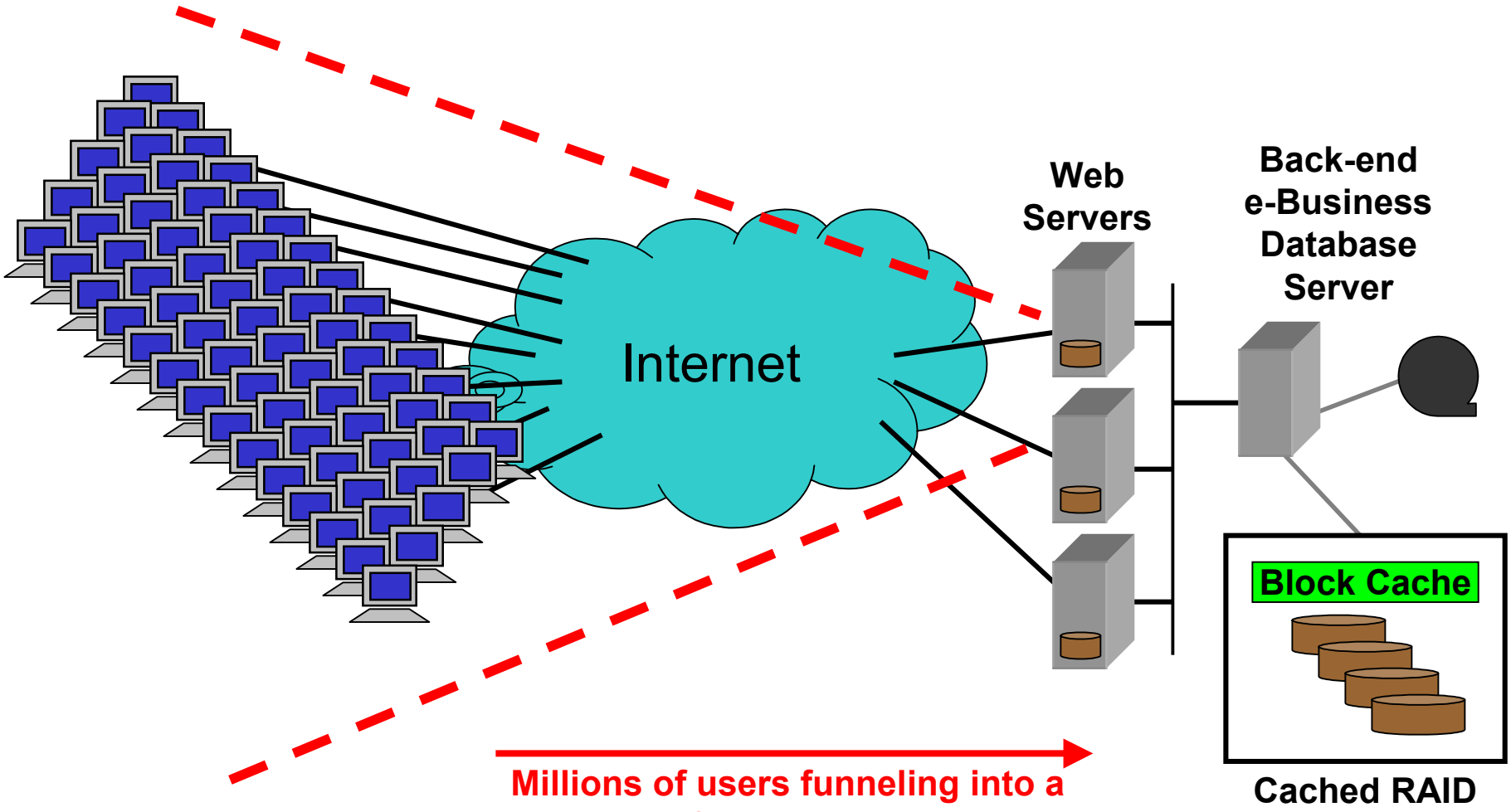
RAID Has no Effect on Latency



Solid-state File Cache Performance Advantage is Orders of Magnitude



Internet's Effect on Transactions

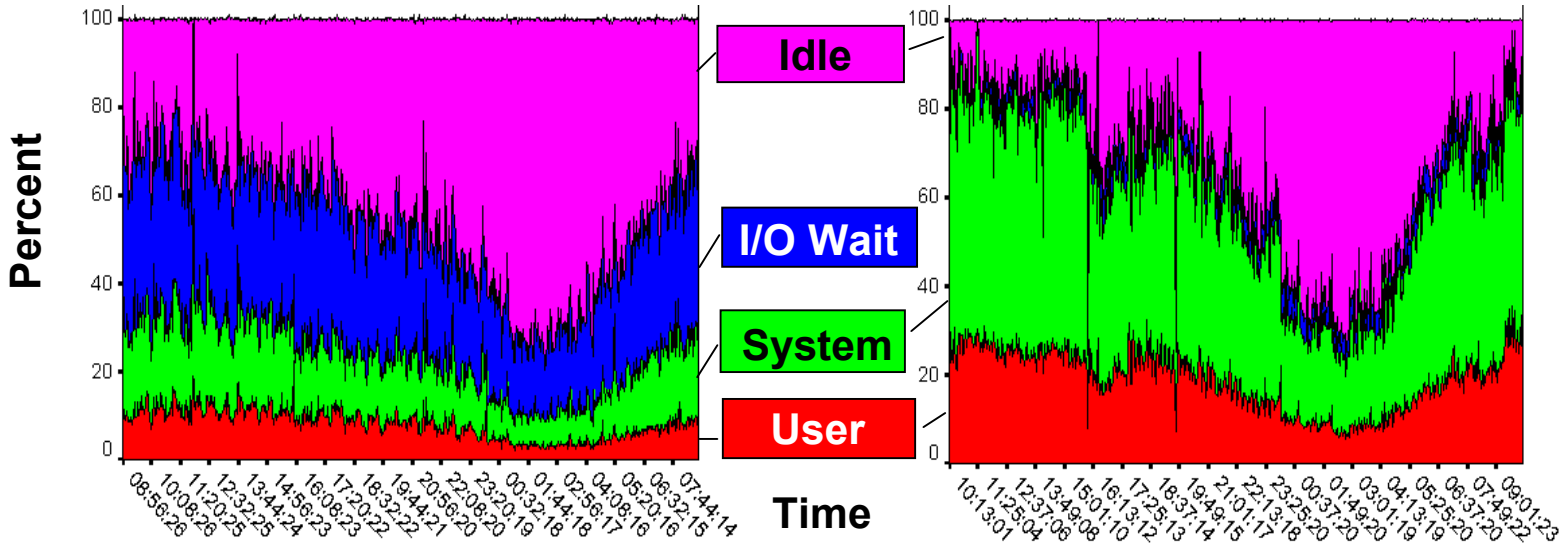


I/O Wait - The Performance Enemy

CPU utilization analysis -- E-mail server peak workload

a) Before file cache

b) After hot files moved



I/O Wait = 45% of CPU time at peak load

13 messages/second

I/O Wait = virtually eliminated

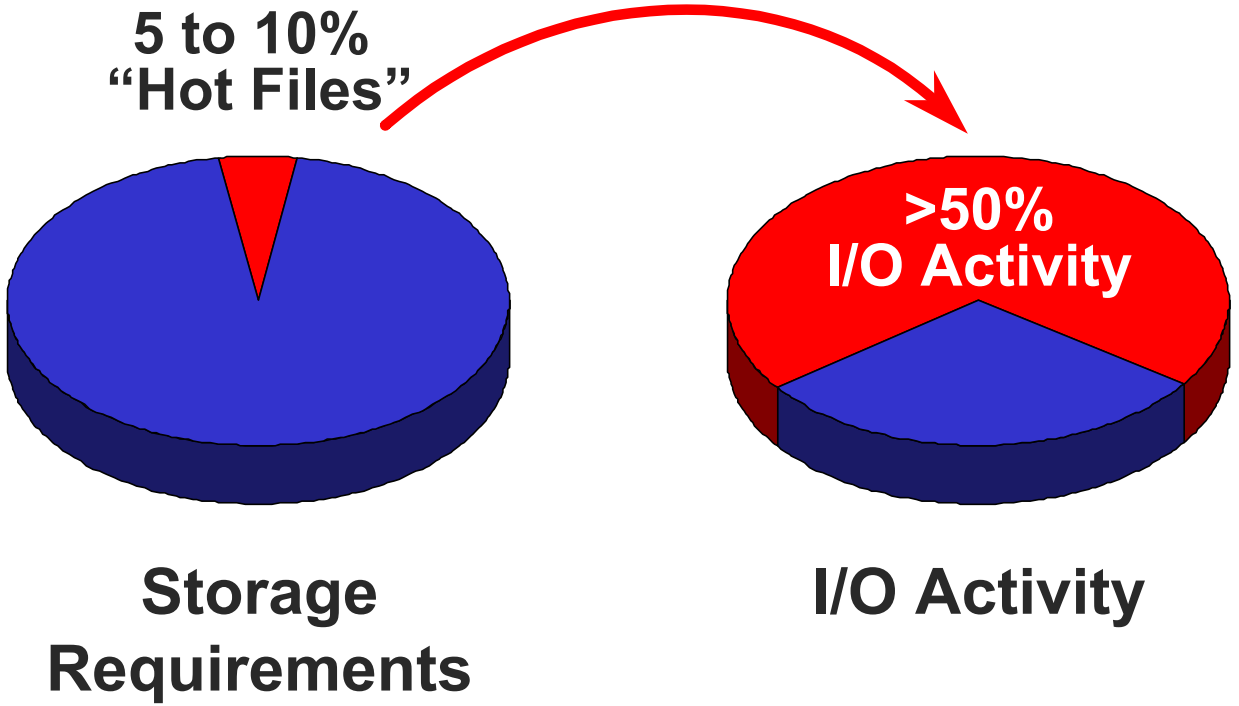
55 messages/second

>4x Improvement

Symptoms of Potential I/O Bottlenecks

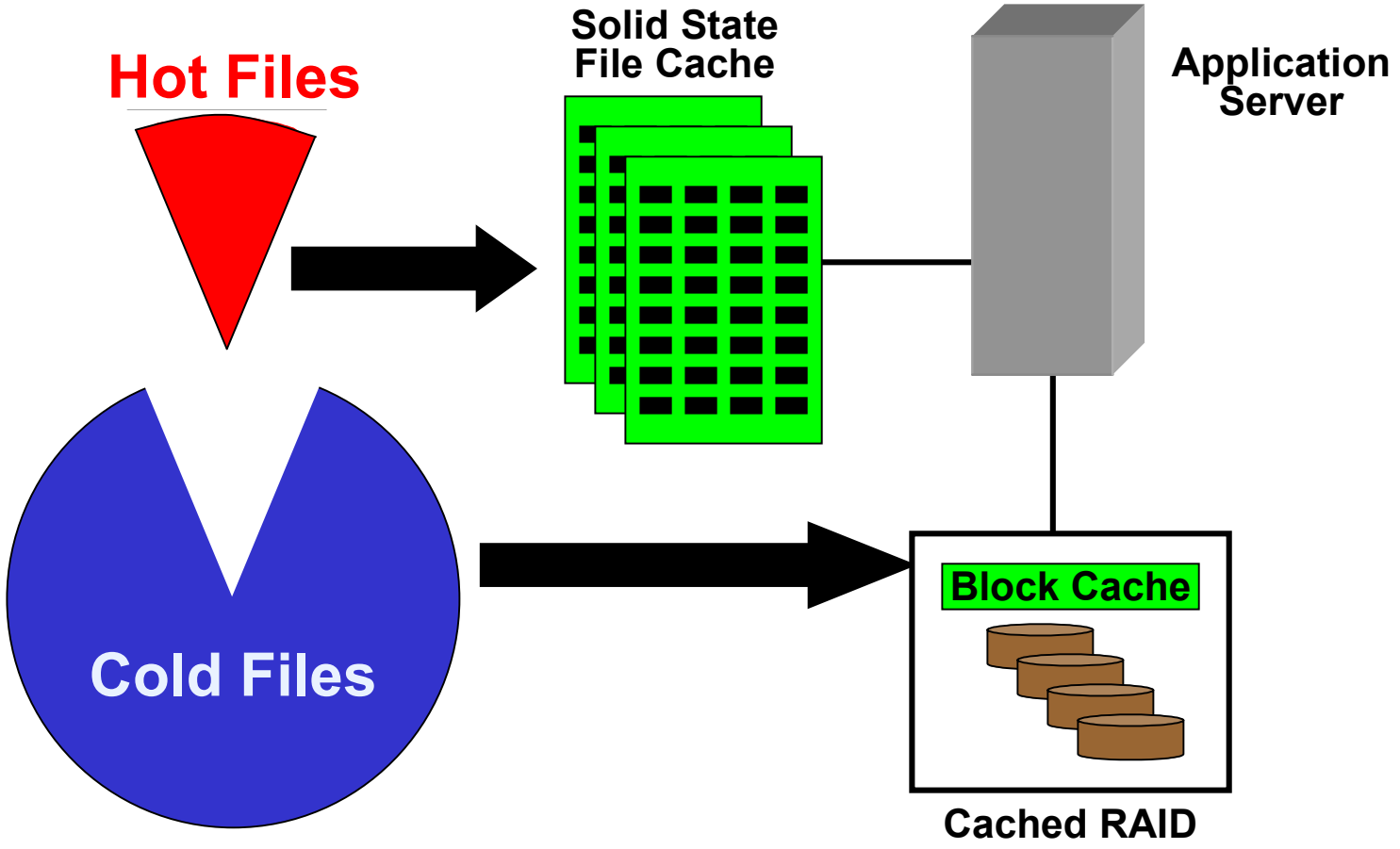
- Performance drops radically as users are added to a system
- System no longer able to “keep up”
- Batch jobs don’t complete within the available processing time window
- Month-end close takes days instead of hours
- Extensive tuning has not solved performance issues
- SAR (Unix) or PerfMon (NT) indicate >50% I/O Wait

Solid-state File Caching Criteria



When a small number of files represent a large percentage of total I/O activity, solid-state file cache is highly likely to multiply performance by 200%-800%+

File Cache - a New I/O Architecture



Separate file cache is independently scalable & manageable

Three Easy Steps to Performance Gains

1) Install Solid-state File Cache
2) Format & mount the SSFC device using standard OS tools
3) Move “hot” files

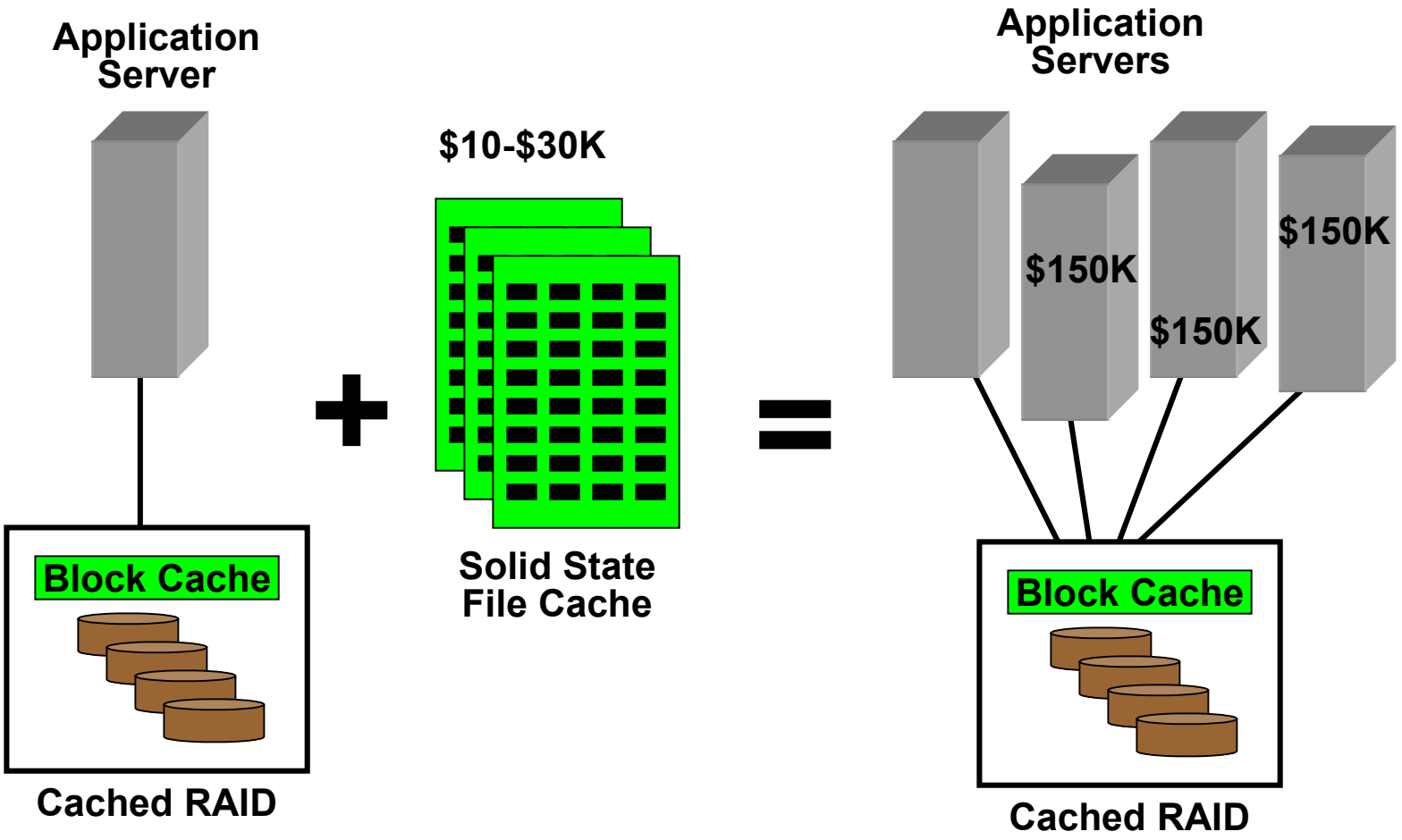
In most cases, the entire process takes only 20-40 minutes

Results are immediate!

Real-world Performance Gains with Solid-state File Caching

- Email server message capacity increased from 5/sec to over 40/sec
- Satellite-based stock data recording backlog reduced from 4 hours to 10 milliseconds
- Server cluster fail-over reduced from 15 minutes to 7 minutes
- Overall file system speed increased by 25%
- Billing application reduced from 5 days to 2 days
- Batch job reduced from 8 hours to 2 hours
- Batch job reduced from 72 hours to 8 hours
- System response time after data entry reduced from 15 seconds to 3 seconds

Value Proposition



Existing server plus Solid-state File Cache equals the performance of 4 servers.

Applications of Solid-state File Cache in a SAN

- E-Mail queues & server-to-server messaging queues
- Non-volatile shared memory for server clusters
- File system journaling
- Snap-shot device for backup / remote copy
- Device lookup directories
- OCR scanning queue
- Databases
  - Rollback segments - Transaction logs
  - Temp spaces - Hot indexes
  - Hot tables

Features to look for in Solid-state File Cache

- Appropriate form factor - *3.5", desktop, rack-mount*
- Low latency - *industry range is 14-50 microseconds*
- Non-volatile architecture
- Connectivity - *Fibre or SCSI*
- LUN mapping / masking
- Redundancies to match availability requirements
- Upgradable capacities to meet future growth
- Ease in serviceability
- Field proven reliability
- Out-of-band monitoring / SNMP support
- Available 4 hour onsite & 24x7 phone support

Summary

- The Internet has pushed many applications to the limits of scalability causing I/O bottlenecks
- Solid-state file cache is low latency storage complementary to cached RAID
- Solid-state file cache is a cost efficient way to multiply performance and scalability of existing servers
- Using solid-state file cache in a SAN multiplies performance across multiple servers, allows amortization of costs

Q&A

For a copy of this presentation with full speakers notes or answers to other questions please send e-mail request to me:

mhall@soliddata.com