Solid State File Cache

A New Infrastructure for Growing Your Business

Michael Casey

Solid Data Systems

mcasey@soliddata.com

Session 9326

HP WORLD 2002 Conference & Expo

Agenda

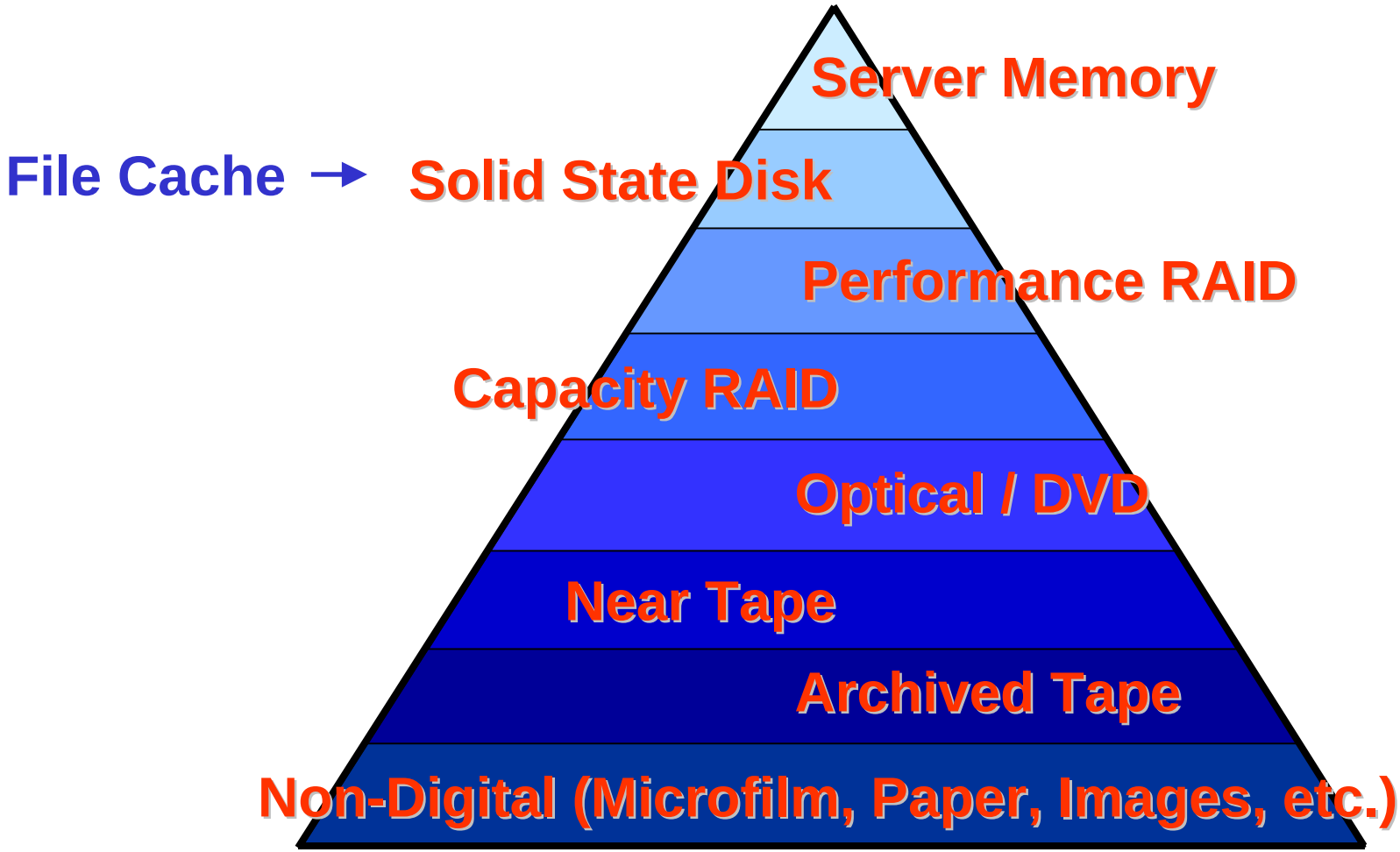
- Why File Cache?
- File Cache for Transaction Servers
- Benefits: Speed + Persistence in Real Applications
- Diagnostic tools – at CPU, I/O, & Application Levels
- New Applications = New Infrastructure
  - Telecom
  - Financial Services
- File cache for Storage Networks
- Conclusions

Why File Cache?

Waiting for I/O:

- Performance drops radically as users are added
- 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

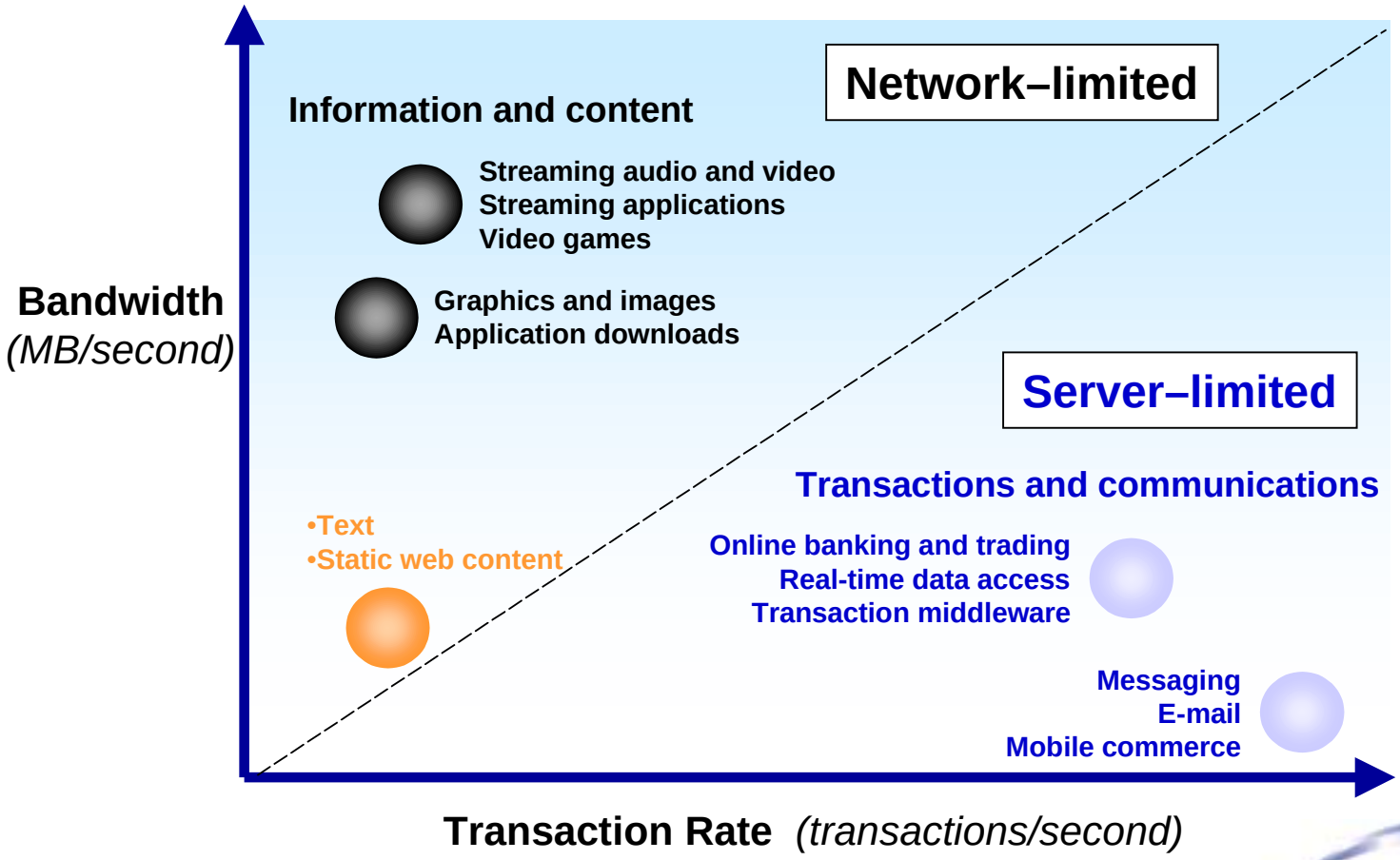
The Solution: Solid-State File Cache



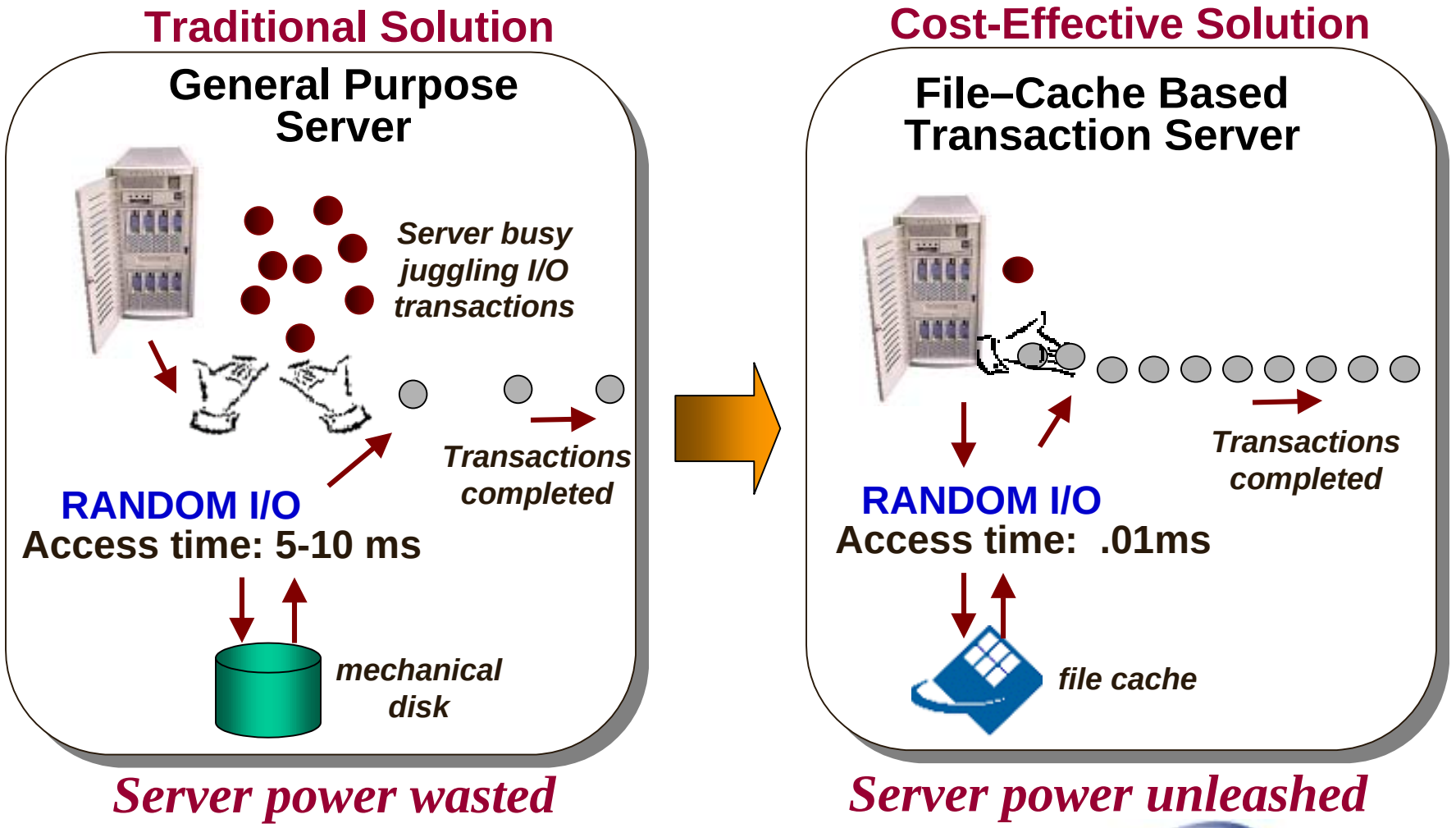
Source: Fred Moore, Horison Information Strategies; 1999

Bandwidth Bottleneck → Transaction Trauma

From Content to Commerce



File Cache for Transaction Servers



File Cache 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

Speed and Persistence

Real-World Application Benefits

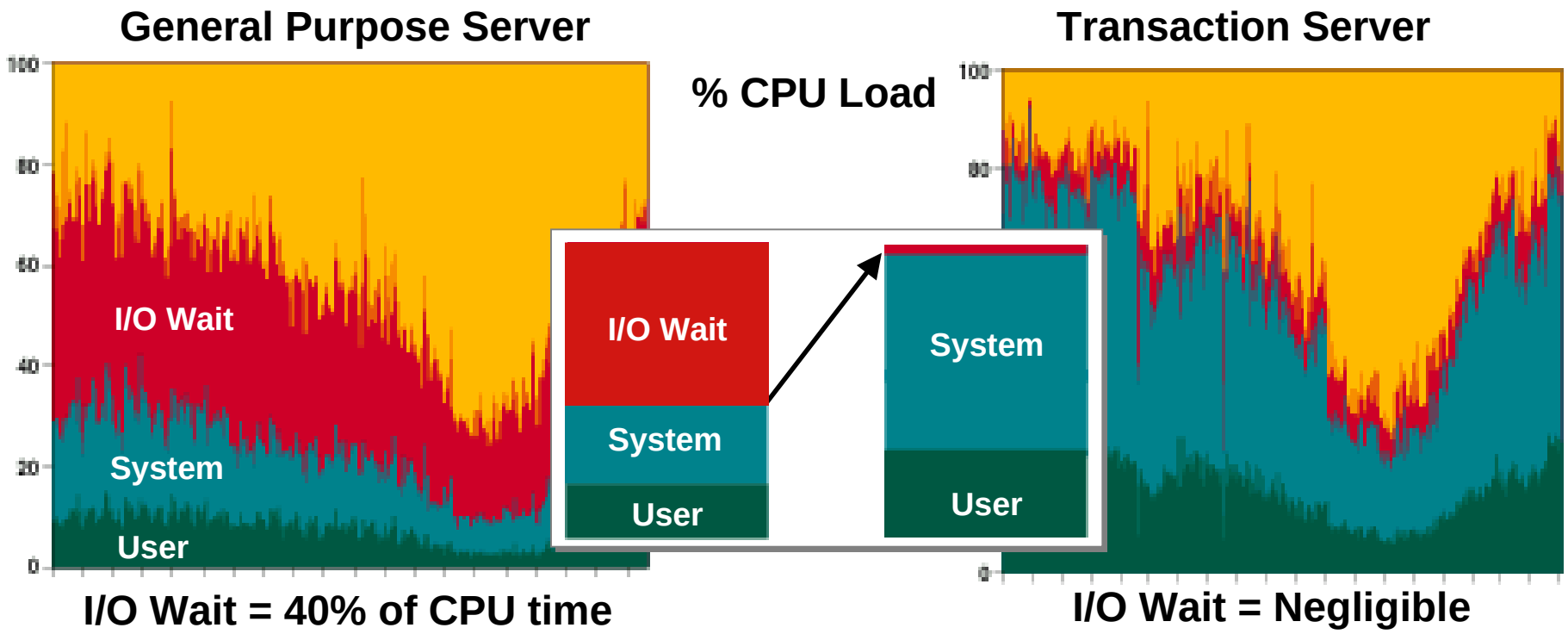
- Email server message capacity increased from 5/sec to over 40/sec
- Satellite-based stock data recording backlog reduced from 4 hours to 10 milliseconds
- Server cluster fail-over reduced from 15 minutes to 7 minutes
- Overall file system speed increased by 25%
- Billing application reduced from 5 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

Diagnostic Tools

- CPU utilization – is the server I/O bound?
  - Standard system tools – SAR, Perfmon, etc.
  - Third-party utilities – e.g., I/O Dynamics*
- I/O profile: is it skewed, with hot files?
  - Standard sys admin tools and displays
  - Third-party utilities – e.g., I/O Dynamics*
- Application-level benchmark tests
  - 100% reads, 100% inserts, 100% updates
  - Synthetic workload generator

* Available for free download

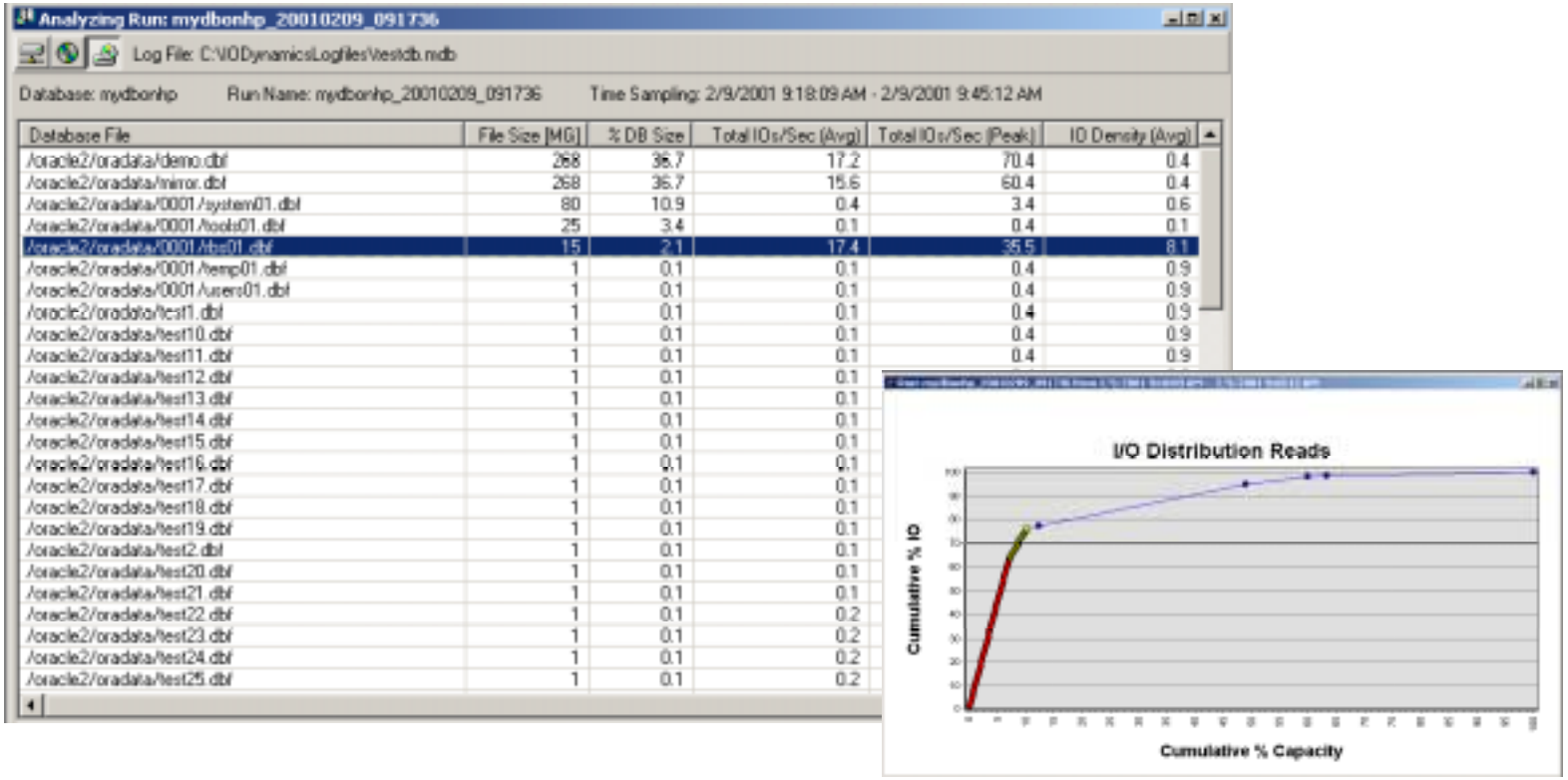
CPU Utilization – % Wait for I/O



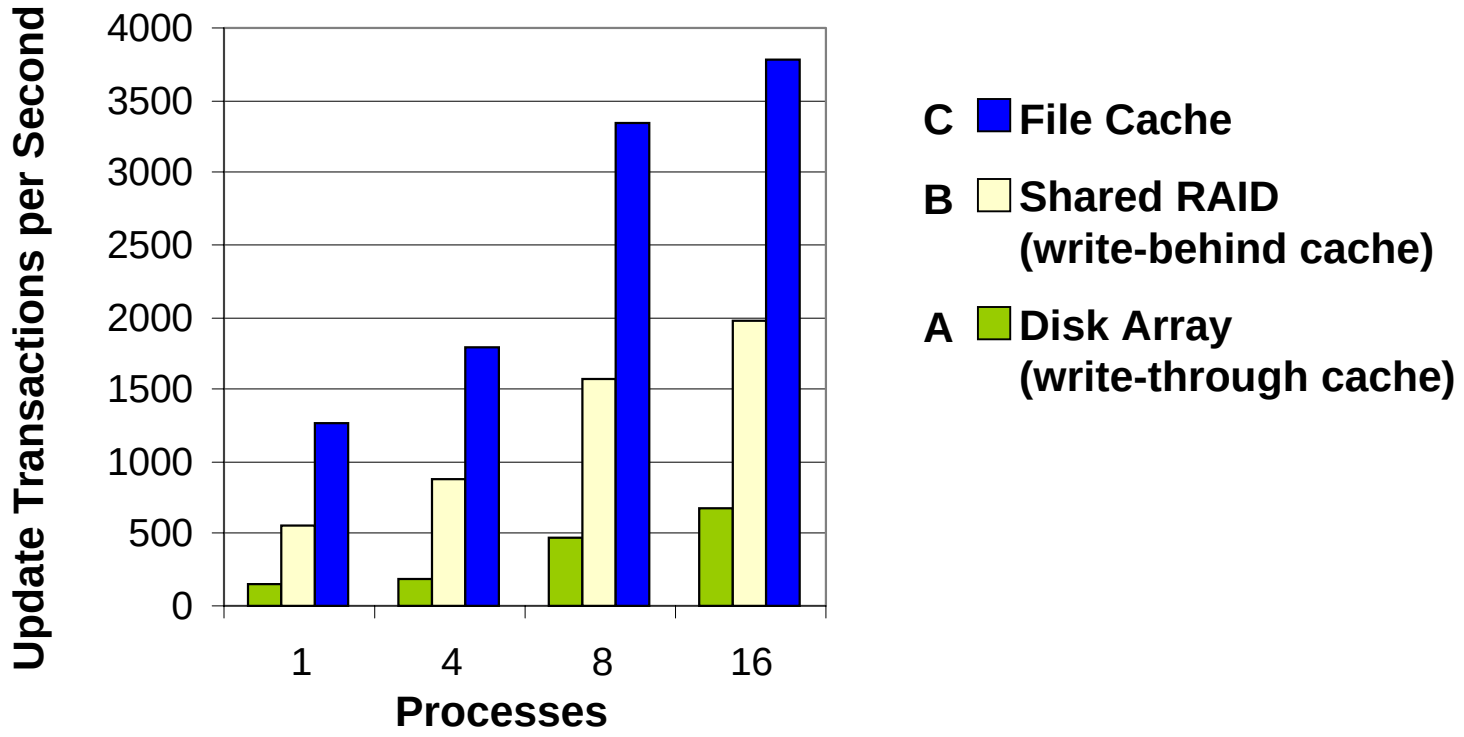
10,500 messages/hr → 44,600 messages/hr

Source: Solid Data measurement of an e-mail system workload over 24-hour period

I/O Activity vs. File Size



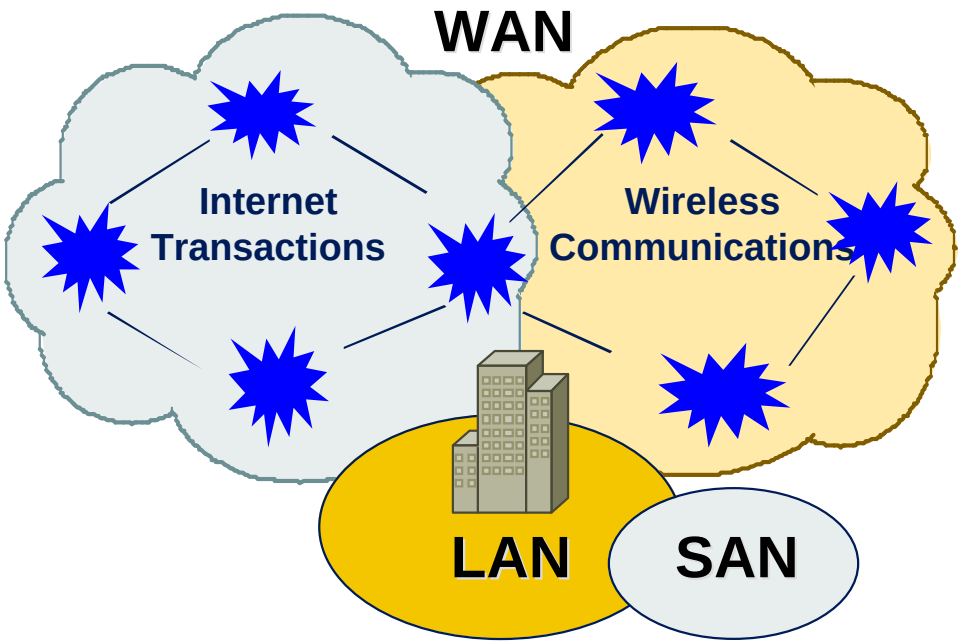
Real-Time Database Benchmark



Source: ***Speed and Persistence for Real-Time Transactions***
July 2002 white paper – www.soliddata.com/timesten

New Applications → New Infrastructure

High-Volume Transaction Demand



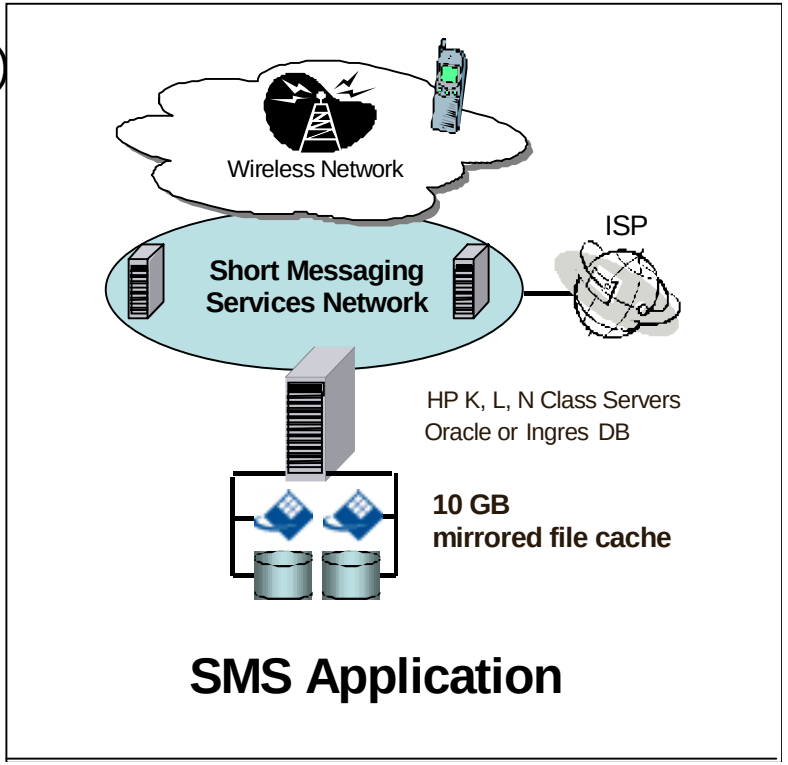
Rich functionality with real-time response

Telecommunications: Enabling New Services for Revenue Growth

- Wireless text messaging (SMS)
- Prepaid mobile billing
- Anti Spamming
- Call detail record (CDR) mediation
- Fraud detection
- Instant Messaging
- Ring-tone download
- Location-based services
- Mobile Internet (WAP)
- Multimedia Messaging Service (MMS)
- Micro-payments

Case Study: Wireless Text Messaging

- Problem
  - Superior product offering (functionality) was not meeting throughput and response time requirements
- File Cache-based Solution
  - 2 mirrored 10GB file caching systems
- Business Benefits
  - Higher throughput at lower cost per subscriber
  - Accelerated purge expands daily window
  - Richer functionality
  - Higher reliability
  - Improved scalability
  - Lower cost of management

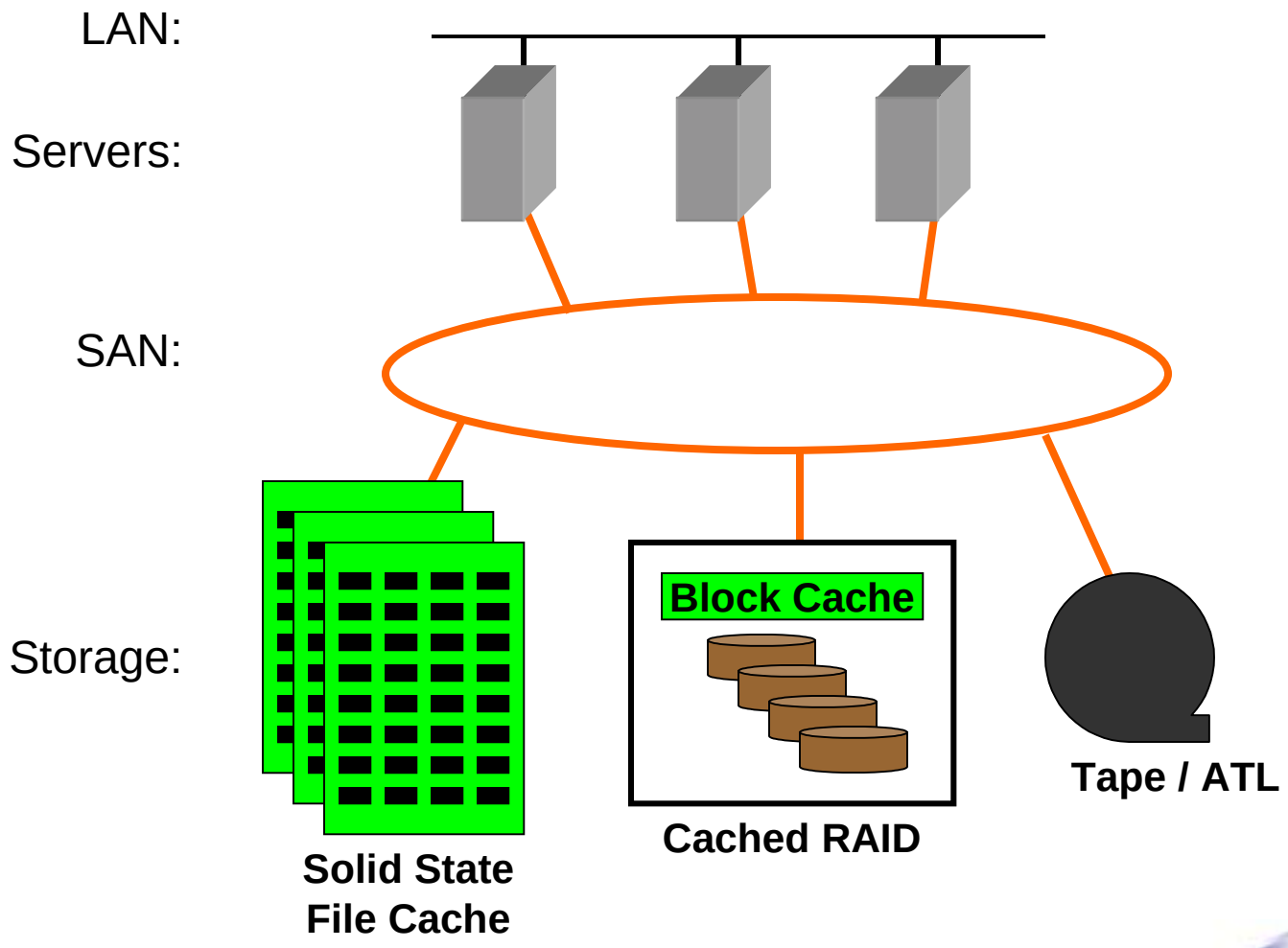


SMS Application

Financial Services: Real-Time Transactions

- Online securities trading
- Portfolio risk management
- Messaging middleware
- Real-time databases
- Credit card fraud detection
- T+1 Trade Settlement
- Straight Through Processing (STP)

File Cache for Storage Networks



Shared File Cache in the 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
- Virtualization lookup tables (address mapping)
- Databases
  - Rollback segments - Transaction logs
  - Temp spaces - Hot indexes
  - Hot tables
- SRM – *policy based management*

Conclusions

- 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
- Diagnostic tools and benchmarks can help diagnose application issues – and determine the fit for file cache
- Transaction and messaging applications – especially in telecom and financial services – require a new infrastructure
- SAN connections and management software can extend the benefits of solid-state file cache to more servers and applications