

# A Survey of NVM Research at MSR-SVC

John D. Davis

Researcher

Microsoft Research Silicon Valley

In Collaboration with Nitin Agrawal, Mahesh Balakrishnan, Andrew Birrell, Michael Isard, Asim Kadav, Dahlia Malkhi, Mark Manasse, Rina Panigrahy, Vijayan Prabhakaran, Abhishek Rajimwale, Gokul Soundararajan, Ted Wobber, Lintao Zhang, and Lidong Zhou

# MSR Silicon Valley (most of us)



# Who We Are

- Four groups with varying research missions
- Focus on Distributed Computing (broadly interpreted)
- In aggregate, approximately 70 researchers
  - MSR worldwide is about 850 researchers
- Expertise spans theory and practice
  - No formalized subgroups
  - Cross-specialty collaboration
- High visibility in professional community
  - Papers, program committees, editorial boards, etc.
- Extensive academic collaboration
- Prestigious awards recognize world-class contributions
- More than 30 significant technology transfers to Microsoft businesses in the last 5 years

# Technical Focus Areas

- Algorithms and Theory
- Distributed Systems
- Security and Privacy
- Software Tools
- System Architecture
- Web Search and Data Mining

# Outline

- High-Performance Flash Disks (OSR '07)
- Transactional Flash (OSDI '08)
- SSD Performance (Usenix '08)
- SSD Block Managements (Usenix '09)
- Flash Research Platform (WISH '09)
- SSD Write Cache (FAST '10)
- Differential Raid (Eurosyst '10)
- Conclusions

# High-Performance Flash Disks

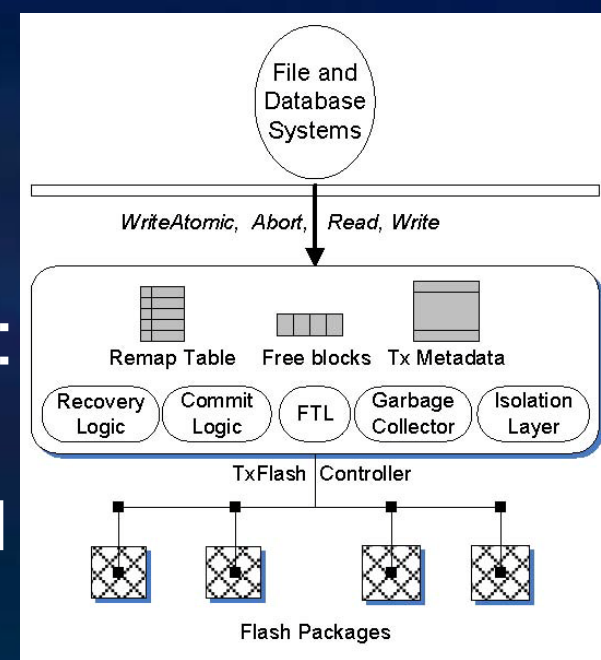
(OSR '07)

- Volatile and non-volatile data structures
- RAM for logical to physical page/block mapping: LBA Table, Free Blocks, Usage Block, Next sequence, Active Page, and Sequence List.
- Flash: Data and 1 page of summary information or metadata
- Design goals: high performance and reliability, but requires a lot of volatile memory

# Transactional Flash

(OSDI '08)

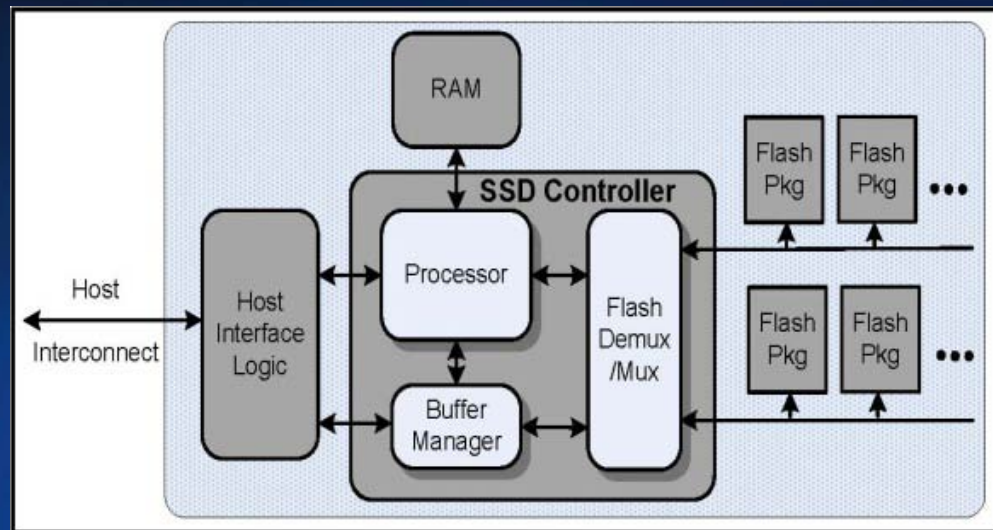
- Export new abstraction from SSD:
  - WriteAtomic( $p_1, \dots, p_N$ )
  - All pages  $p_1$  to  $p_N$  are either updated fully or not modified at all
- Simplifies file and database systems
- Flash is suitable for the new abstraction
  - Non-overwrite pages, fast random reads, and high parallelism
- Use a novel “cyclic commit” protocol
- Evaluate the design with a simulator and file system modifications



# SSD Performance

(Usenix '08)

- Study fundamental tradeoffs in SSDs
- Build SSD simulator for evaluating design choices
- Traces from existing systems: TPC-C, Exchange. File system benchmarks



# SSD Performance

- Key findings:
  - Performance is highly workload dependent
  - SSDs eliminate IOPS bound on current workloads
  - High-level system issues appear in firmware

Techniques	Positives	Negatives
Large allocation pool	Load balancing	Few intra-chip ops
Large page size	Small page table	Read-modify-writes
Overprovisioning	Less cleaning	Reduced capacity
Ganging	Sparser wiring	Reduced parallelism
Striping	Concurrency	Loss

# SSD Performance

- Numerous tradeoffs in designing SSDs
- Significant hardware and software interplay
- Performance & lifetime highly workload sensitive
- Problems at device firmware mimic problems higher in the storage stack
- Careful SSD design has the potential to change the storage landscape

# SSD Block Management

(Usenix '09)

- Several assumptions are no longer valid

Assumptions	Disks	SSDs
Sequential accesses much faster than random accesses	✓	✗
Data issued equals data written (no write amplification)	✓	✗
Storage device passive (little or no background activity)	✓	✗
Media does not wear down	✓	✗
Distant LBNs lead to longer access time	✓	✗

## Implications

- ▣ Need to tune storage stack for SSDs

# SSD Block Management

- Modifications to tune storage stack for SSDs
  - Cope up with violated assumptions
- Rich interface to convey more information to device
  - IO priorities
  - Free block information
- More functionality in device
  - Low level block management
- Solution
  - Object based storage (OSD)

# SSD Block Management

- Write amplification
  - Need “stripe size” from device
- Background activity (Priority aware cleaning)
  - Need “IO priority” information from OS
- Device wear-down (Informed cleaning)
  - Need “free block” information from FS
- **Need richer interface**

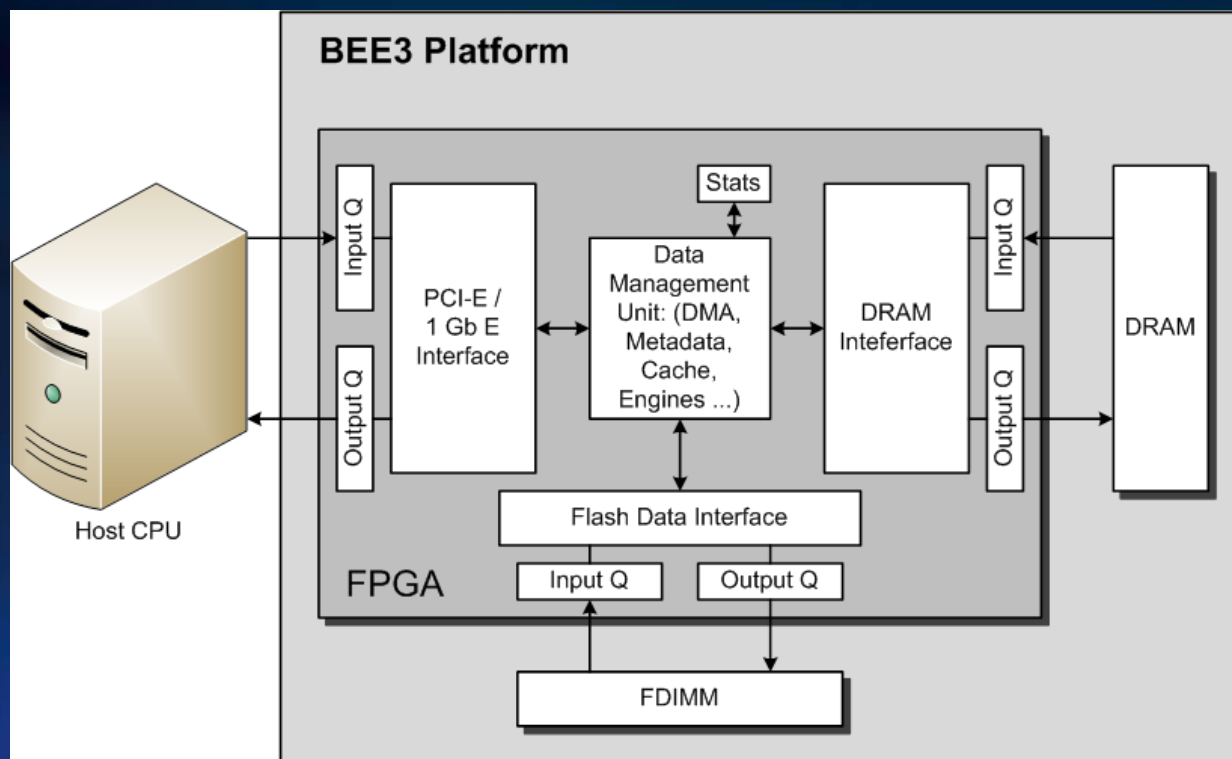
# SSD Block Management

- OSD manages space for objects
  - Informed cleaning
  - Stripe aligned accesses
  - Logical to physical mapping
- Object attributes in OSD
  - Wear-leveling using cold data information
  - Priority assigned to objects
- OSD handles low-level operations
  - Block management in SSD

# Flash Research Platform

(MISH '09)

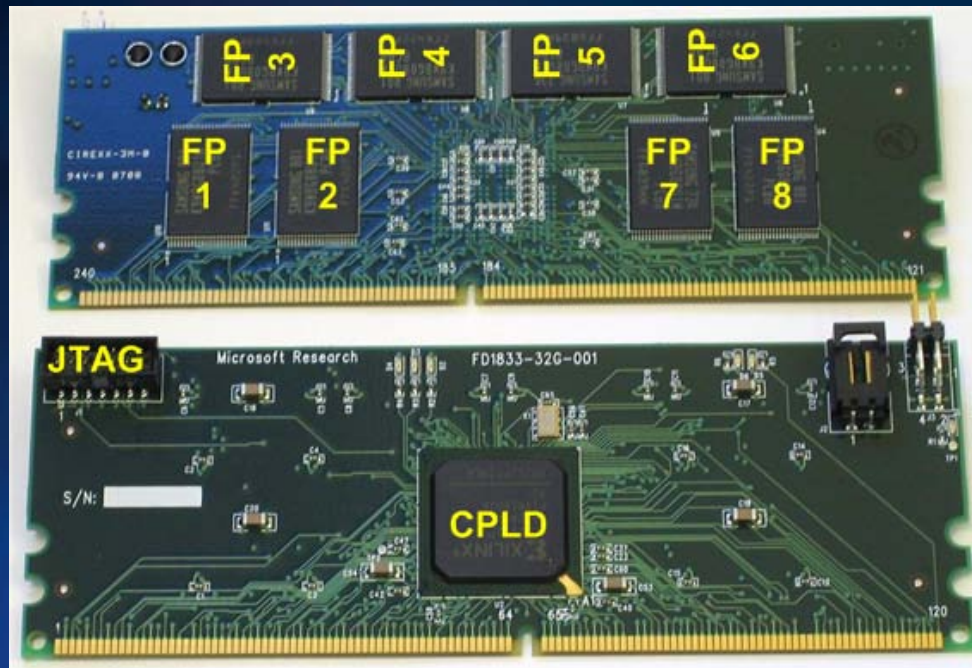
- Explore architectural units built from flash (e.g., SSD)
- Large capacity (up to 1/2 TB) for realistic experimentation
- Variety of I/O options (to PC) via BEE3
- Allows exploration via hardware, software, or both.



April 12, 2010 UCSD NVM Workshop

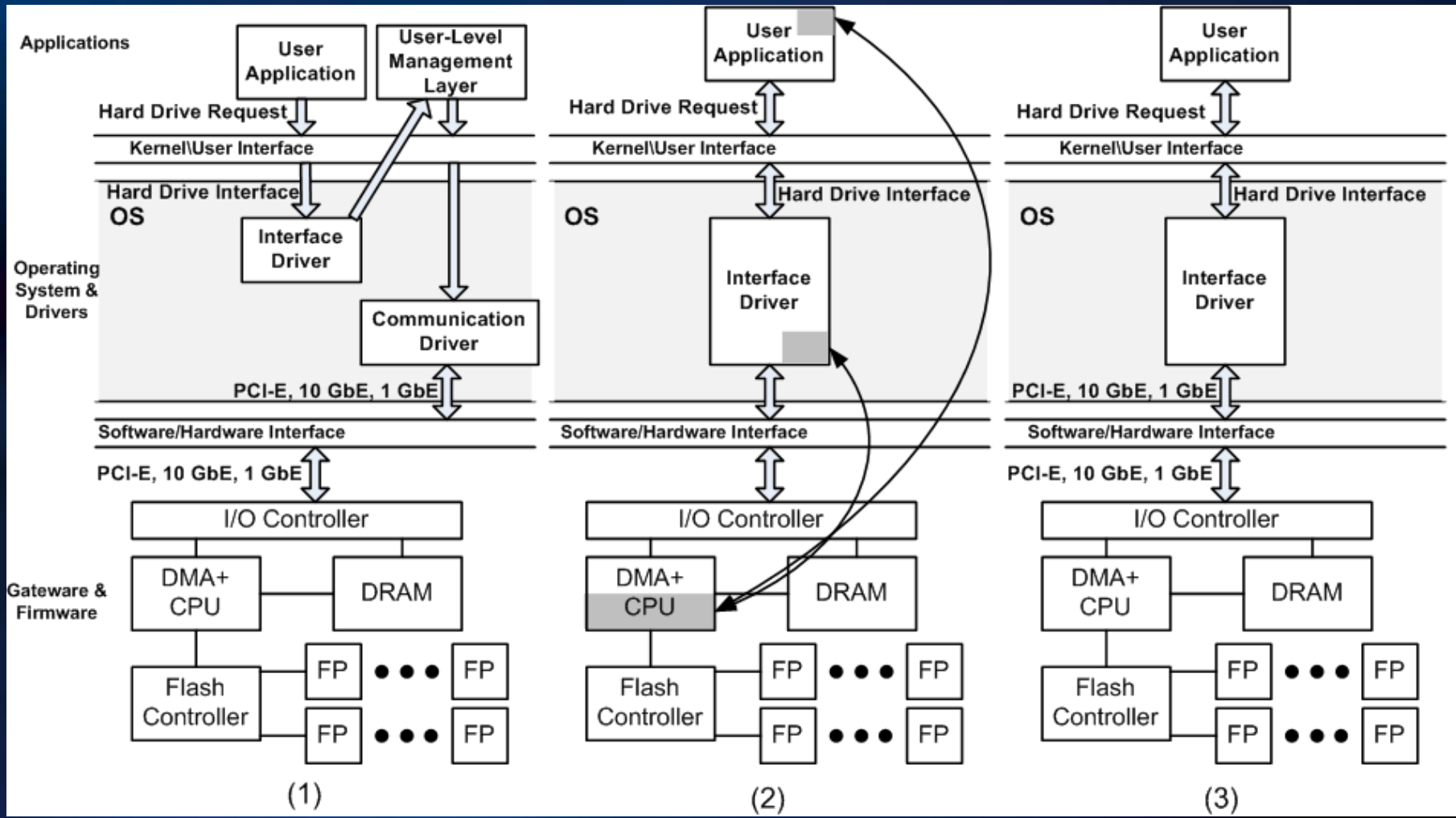
# FRP FDIMM

- 8 Samsung 4 GB SLC NAND Flash Devices
- 8 channels per FDIMM
- CPLD is a pass-through device



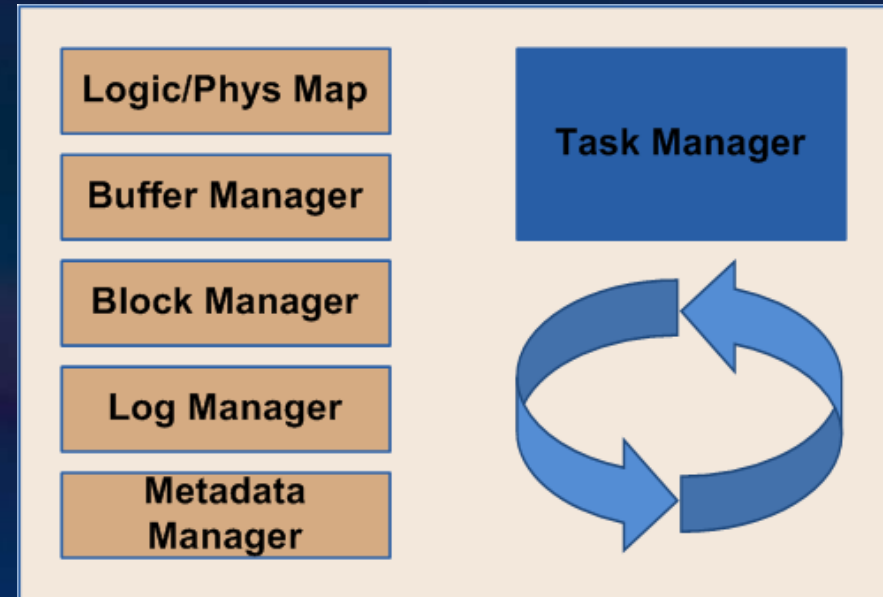
April 12, 2010 UCSD NVM Workshop

# FRP Software



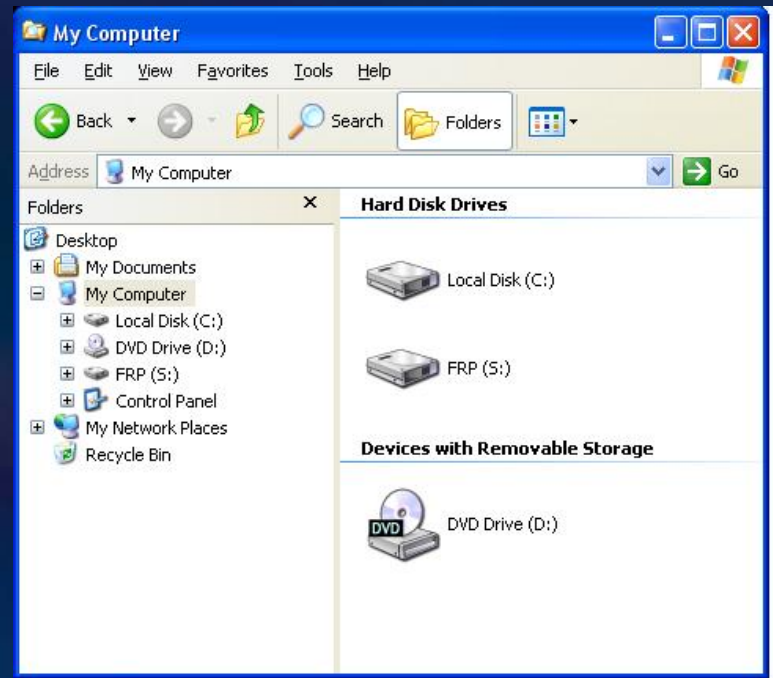
# FRP Management SW

- All FTL management is done in software
  - ECC: 8-bit BCH
  - Log-structure
  - Wear-leveling
  - Garbage collection
  - Page Mapping
  - Metadata management
- Cooperative multithreaded “OS”

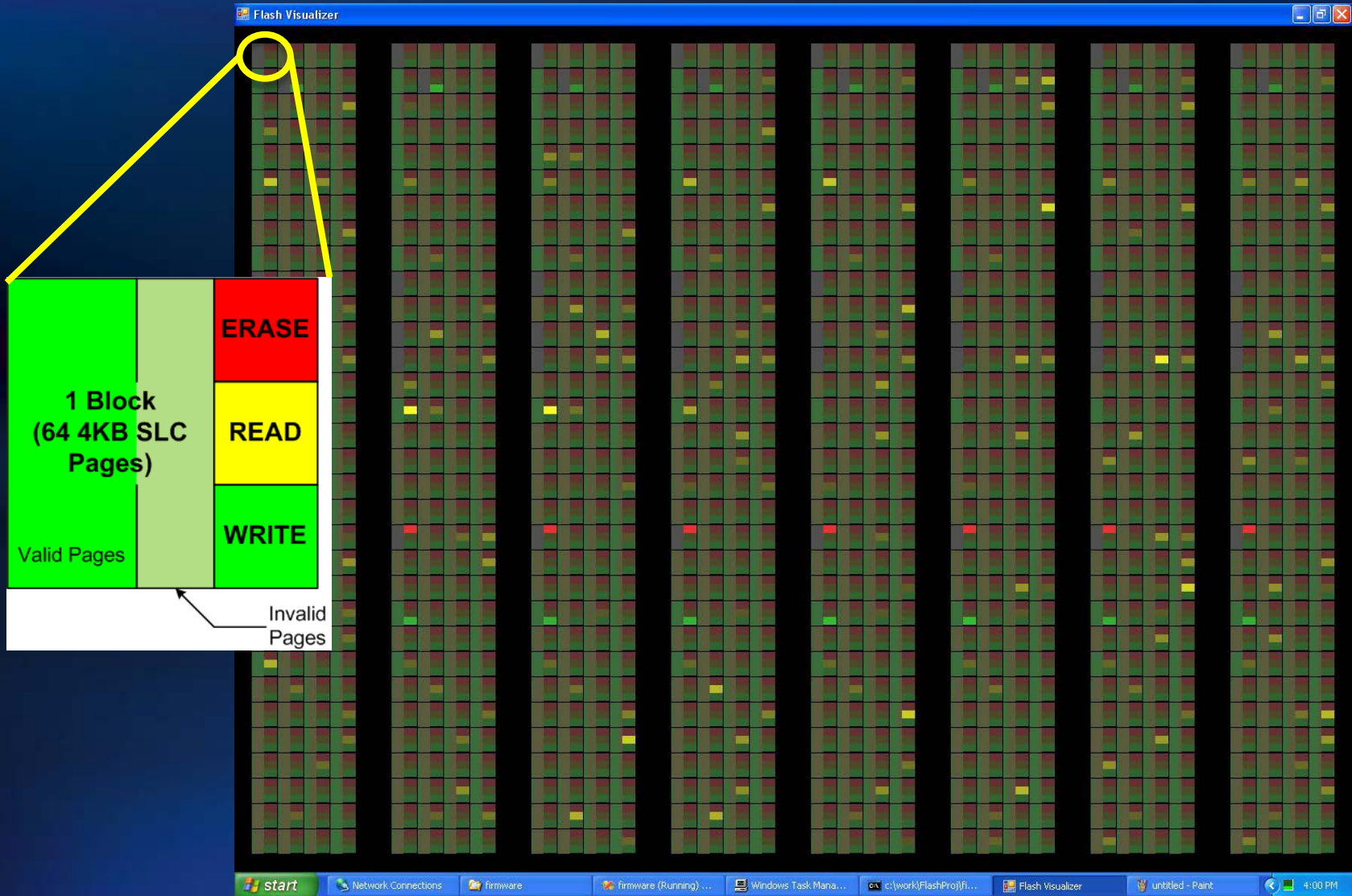


# Current Status

- Mount FRP as a Disk in XP
  - Format
  - Properties
  - Normal operations
    - Read/Write/Erase files
    - Mount/Unmount
- Stress testing system
- SLC and MLC FDIMMS
  - 8 - 128GB
- Up to 2TB



# Current Status



# FRP on BEE3



# SSDs with HDD Write Cache

(FAST'10)

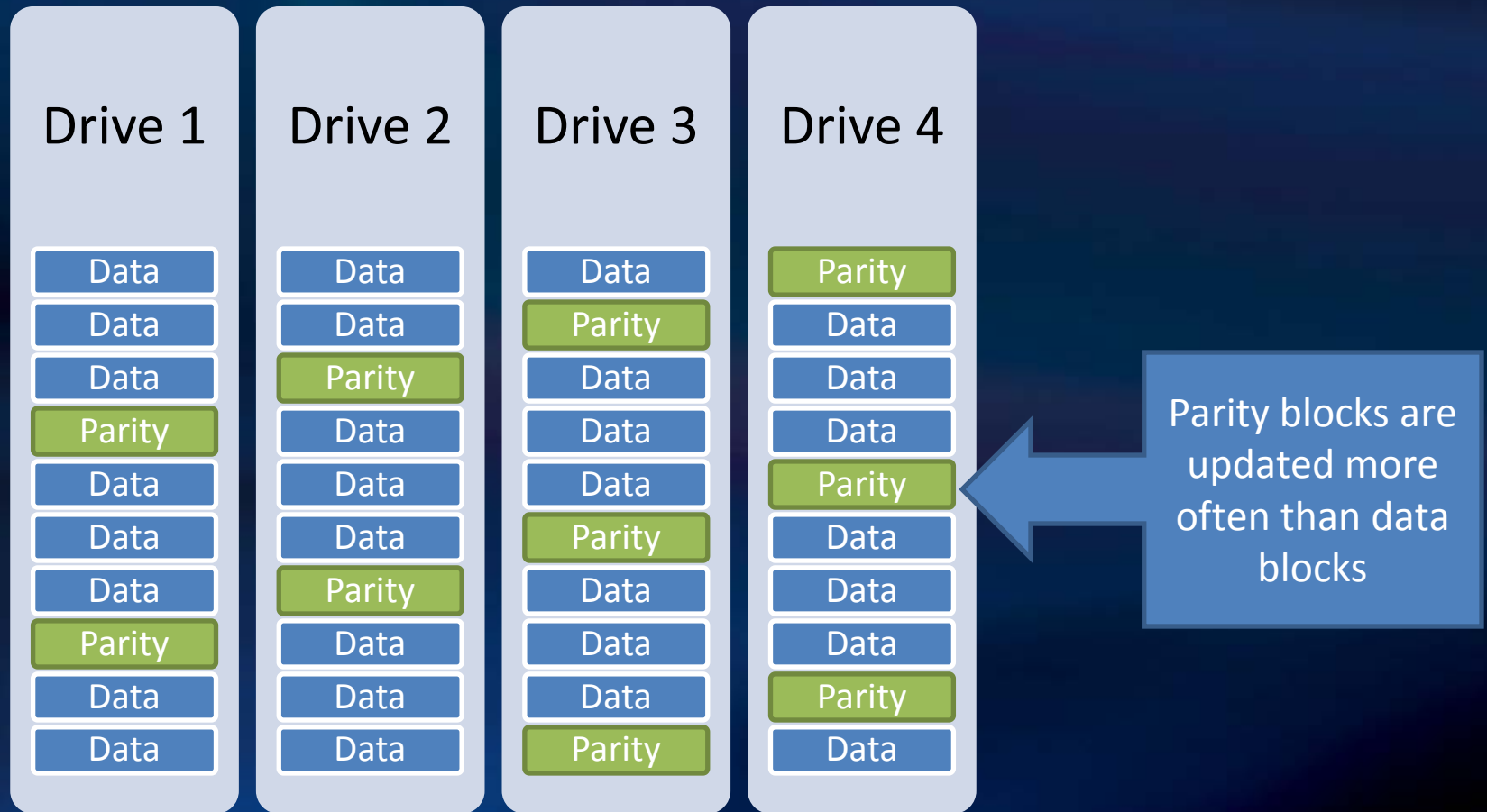
- **Built Griffin hybrid disk**
  - Uses hard drive as a write-cache
- **Reduces writes while maintaining performance**
  - Reduces writes by 52% (< 5% HDD reads)
  - Improves lifetime by factor of 2
  - Reduces average I/O latency by 56%

# Differential Raid for MLC SSDs

(Eurosys '10)

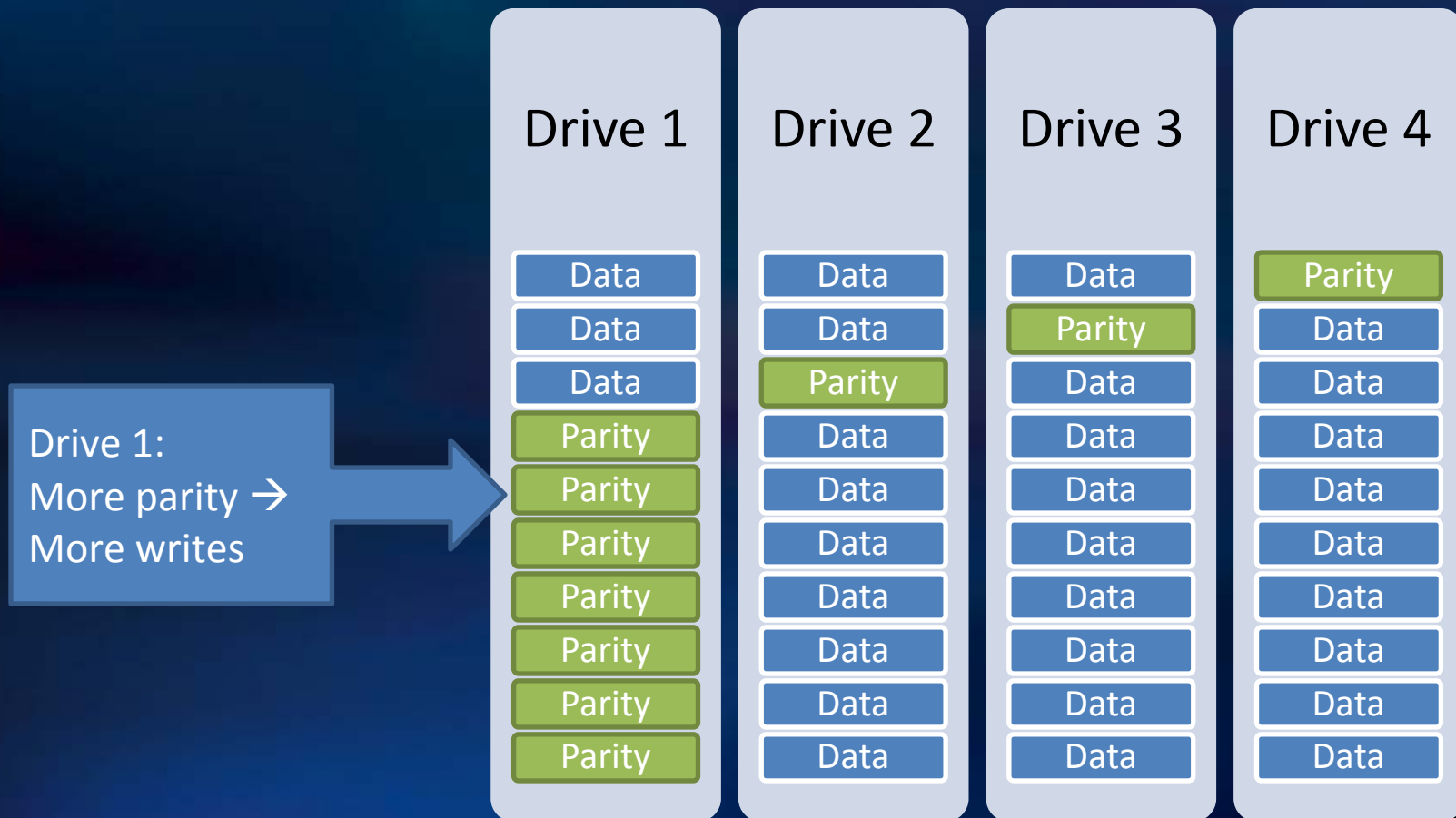
- RAID-5 is a bad idea for Hard Disks:
    - ✗ Performance: Slow random writes
    - ✗ Cost: Storage is cheap → Just use RAID-1/10
    - ✗ Reliability: High probability of data loss in large arrays
  - RAID-5 is a great idea for SSDs!
    - ✓ Performance: Fast random writes (5 SSDs = 14K/sec)
    - ✓ Cost: Storage is expensive → Can't use RAID-1/10
    - ? Reliability: **Correlated Failures!**
- (Not just RAID-5: RAID-1, RAID-4, RAID-10, RAID-6...)

# Differential Raid



In RAID-5, parity blocks are evenly distributed across drives

# Differential Raid



# Differential Raid

- Don't use standard RAID with SSDs!
- Differential RAID:
  - Key Idea: Age SSDs at different rates
  - More reliable, same space overhead as RAID-5
  - Push SSDs safely beyond rated erasure limit

# Summary

- Hardware prototypes and systems
- Rethink the software and hardware storage stack
- Richer SSD interfaces and more visibility
- Improved system properties: reliability, fault-tolerance, and TCO
- Several open questions...
  - Good for research!

# Microsoft®

*Your potential. Our passion.™*