Intel Turbo Memory-Nonvolatile disk caches in the storage hierarchy of mainstream computer systems

Jeanna Matthews, Sanjeev Trika,
Debra Hensgen, Rick Coulson,
and Knut Grimsrud

Intel Corporation

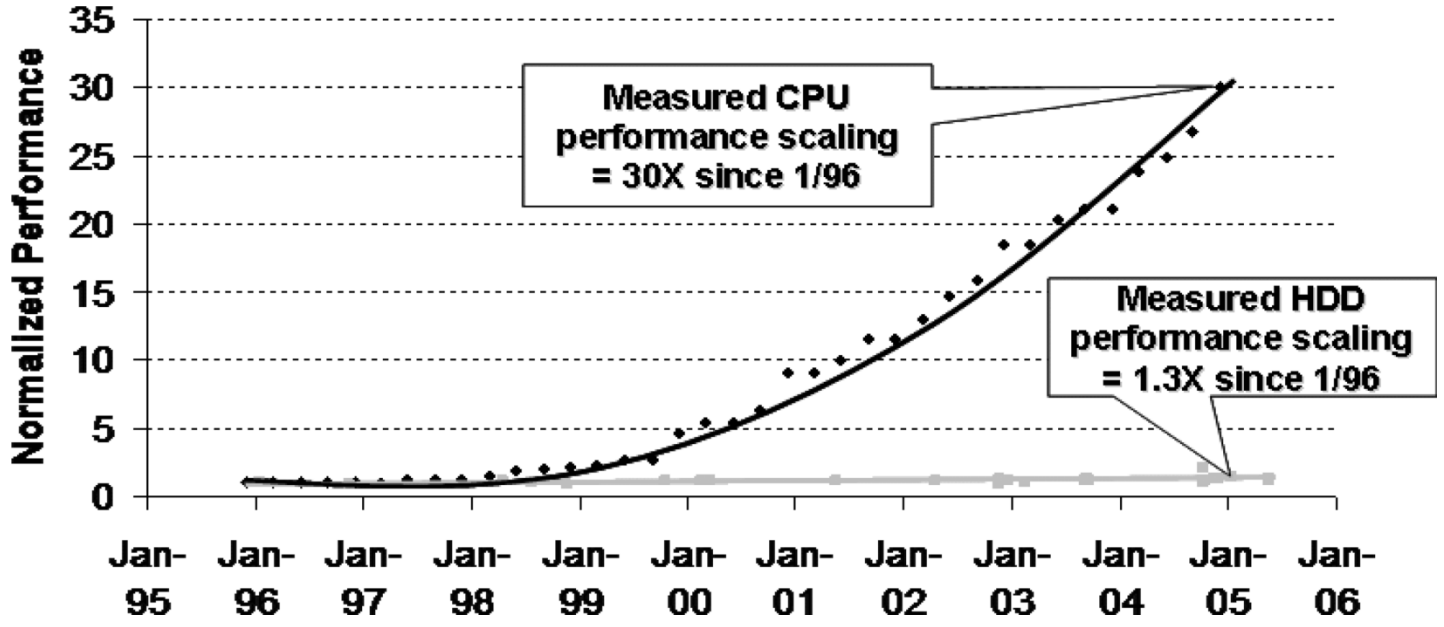
ACM Transactions on Storage, Vol. 4, No. 2,
Article 4, Publication date: May 2008

Outline

- Introduction
- Intel Turbo Memory Architecture
- Cache Management And Caching Policies
- Results
- Conclusions

Introduction (1/3)

- The disk- or I/O bottleneck is a well-recognized problem in the performance of computing systems
- CPU-disk performance gap



Introduction (2/3)

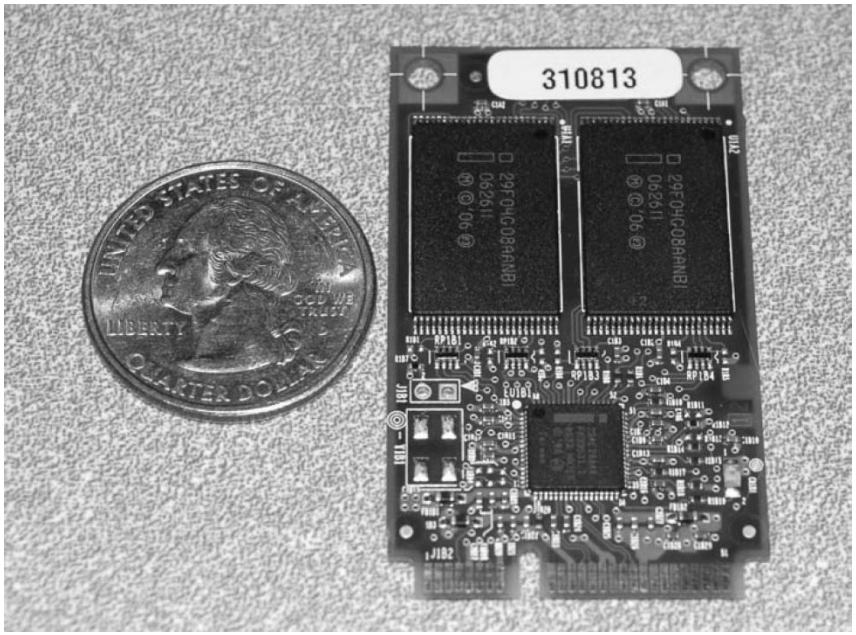
- NAND memory is organized into a set of blocks called Erase Blocks (EBs)
- Each EB contains a set of pages
- The entire EB must be erased before pages in the EB may be written again

Introduction (3/3)

- Intel Turbo Memory (ITM) technology supports both Ready-Boost and ReadyDrive
- Ready-Boost
  - Use a NAND-based SSD as a temporary extension of the system's volatile DRAM space
- ReadyDrive
  - Use devices that support the ATA NVCache feature set to cache data written to the hard-drive platters

Intel Turbo Memory Architecture (1/2)

- Intel Turbo Memory includes a PCI-Express device with on-board NAND, and an intelligent software driver for device control and management



Intel Turbo Memory Architecture (2/2)

- Hardware
  - NAND controller
    - Read operations come in three forms: metadata reads, data reads, and both
    - Write operation writes both data and metadata atomically
  - NAND memory
  - Option-ROM

- Driver software
  - OS interface manager
  - Caching client
  - SSD client
  - NAND management layer

Cache Management And Caching Policies(1/15)

- Cacheline Metadata
- Request Planning
- Caching Policies for Improving Performance
- Caching Policies for Power Savings

Cache Management And Caching Policies (2/15)- Cacheline Metadata

- To keep track of data that is cached, metadata is stored along with each cacheline in the NAND cache

- Nonvolatile metadata
- Volatile metadata

Cache Management And Caching Policies (3/15)- Cacheline Metadata

- Elements of the nonvolatile metadata structure per cacheline

```
typedef struct _NONVOLMETA_CACHE
{
      BOOLEAN      Valid;
      ULONG_48     LBATag;
      ULONG        DiskID;

      BITMASK      PerSectorValid;
      BITMASK      PerSectorDirty;
      BITMASK      PerSectorPinned;
      BITMASK      PerSectorBad;
      ULONG        SequenceNumber;
} NONVOLATILE_CACHE_METADATA;
```

Cache Management And Caching Policies (4/15)- Cacheline Metadata

- Elements of the volatile metadata structure per cacheline

```
typedef struct _VOLMETA_CACHE
{

      ULONG        lastAccess;
      ULONG        hitsAfterInsert;

      PVOID        LastRequest;
      ULONG        AccessCount;

      BOOLEAN      flushBoundary;

} VOLATILE_CACHE_METADATA;
```

Cache Management And Caching Policies (5/15)- Request Planning

- Planning
  - Determine how best to satisfy the request with a combination of accesses to disk and NAND

Cache Management And Caching Policies (6/15)- Dividing Requests into Potential Cachelines

- Potential cacheline
  - Divide each request issued by the OS into a series of disk ranges
- Use the first LBA in each disk range as a tag
- A cacheline offset value is determined such that the majority of requests issued by the OS will be aligned with the logical boundaries drawn on the disk

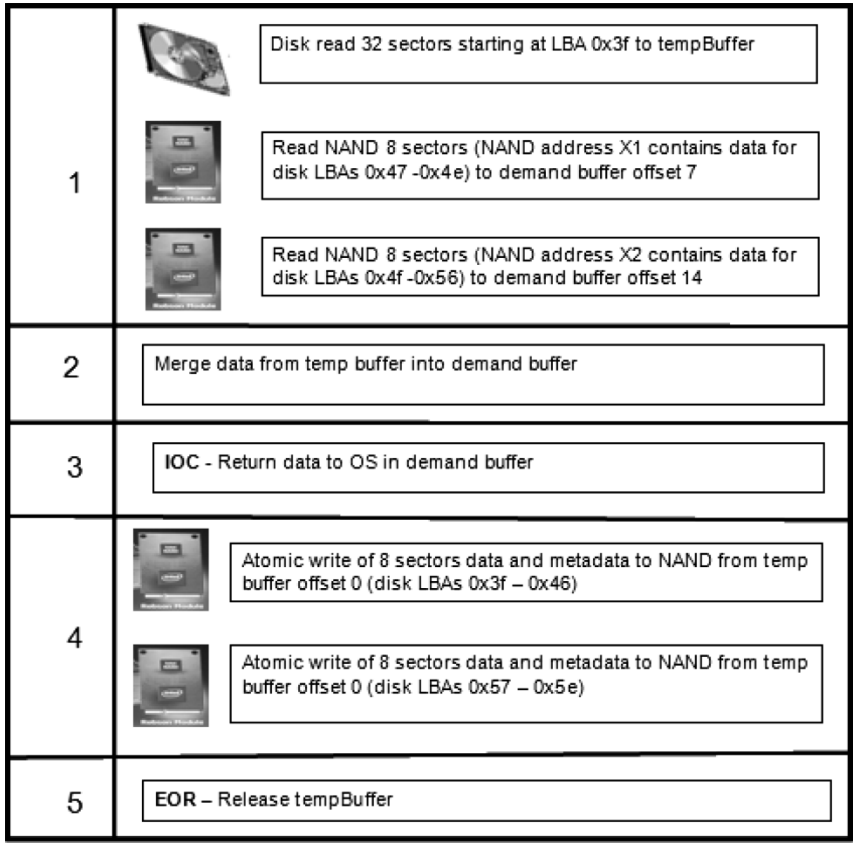
Cache Management And Caching Policies (7/15)- Per-Request Plan Structure

- The plan structure consists of a series of phases
- All operations in one phase must complete before any in the next phase may be issued
- For example
  - In the case of a read miss

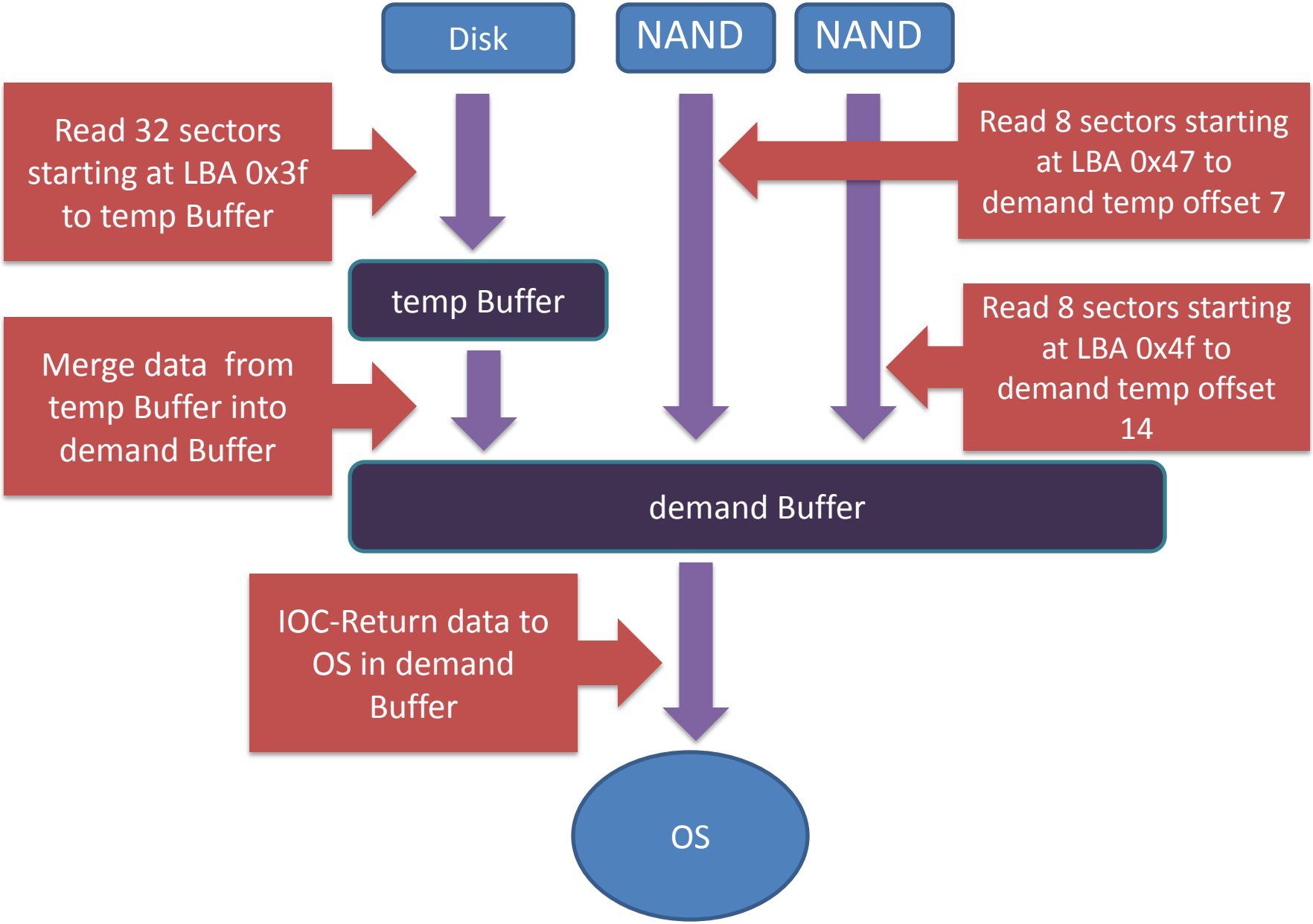
Cache Management And Caching Policies (8/15)- Per-Request Plan Structure

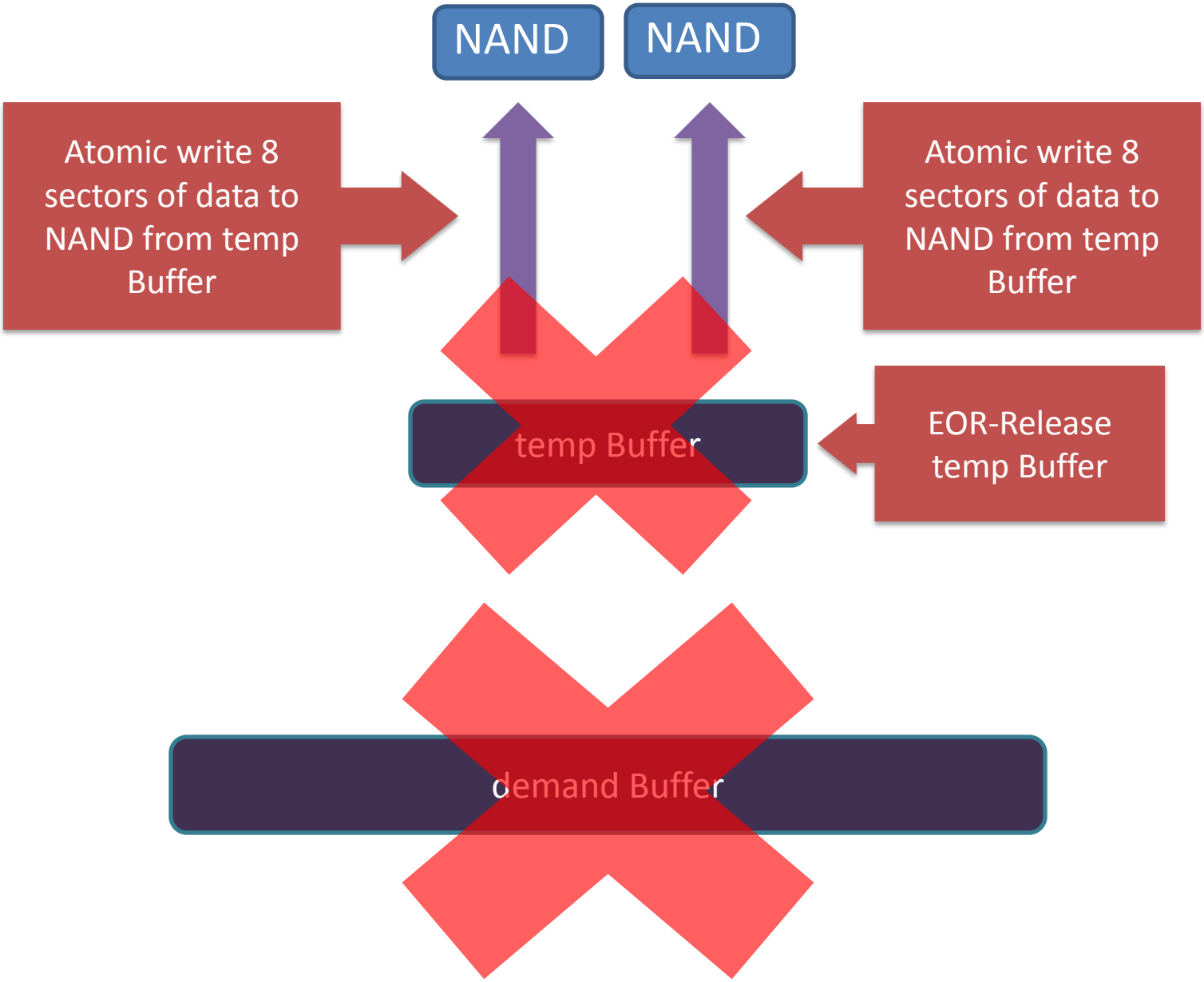
- Phase
  - "I/O cmplete," or IOC
    - The OS is signaled that the request is complete
  - pre-IOC
    - Any operations that take place before IOC phase
  - post-IOC
    - Any operations that take place after IOC phase
  - "end of request," or EOR
    - All resources allocated to this request are released

Cache Management And Caching Policies (9/15)- Sample per-request phase diagram



- A read request for 13KB (or 26 disk LBAS), starting at disk LBA 0x40
- Assuming a cacheline offset value of 1 and cacheline size of 4KB
- 4 logical disk ranges with tags 0x3f, 0x47, 0x4f, and 0x57
- Assume that the first and last disk ranges are not present in the cache, while the middle two disk ranges are present in the cache and dirty


Disk
NAND
NAND
Read 32 sectors starting at LBA 0x3f to temp Buffer
Read 8 sectors starting at LBA 0x47 to demand temp offset 7
temp Buffer
Read 8 sectors starting at LBA 0x4f to demand temp offset 14
Merge data from temp Buffer into demand Buffer
demand Buffer
IOC-Return data to OS in demand Buffer
OS


NAND
NAND
Atomic write 8 sectors of data to NAND from temp Buffer
Atomic write 8 sectors of data to NAND from temp Buffer
temp Buffer
EOR-Release temp Buffer
demand Buffer

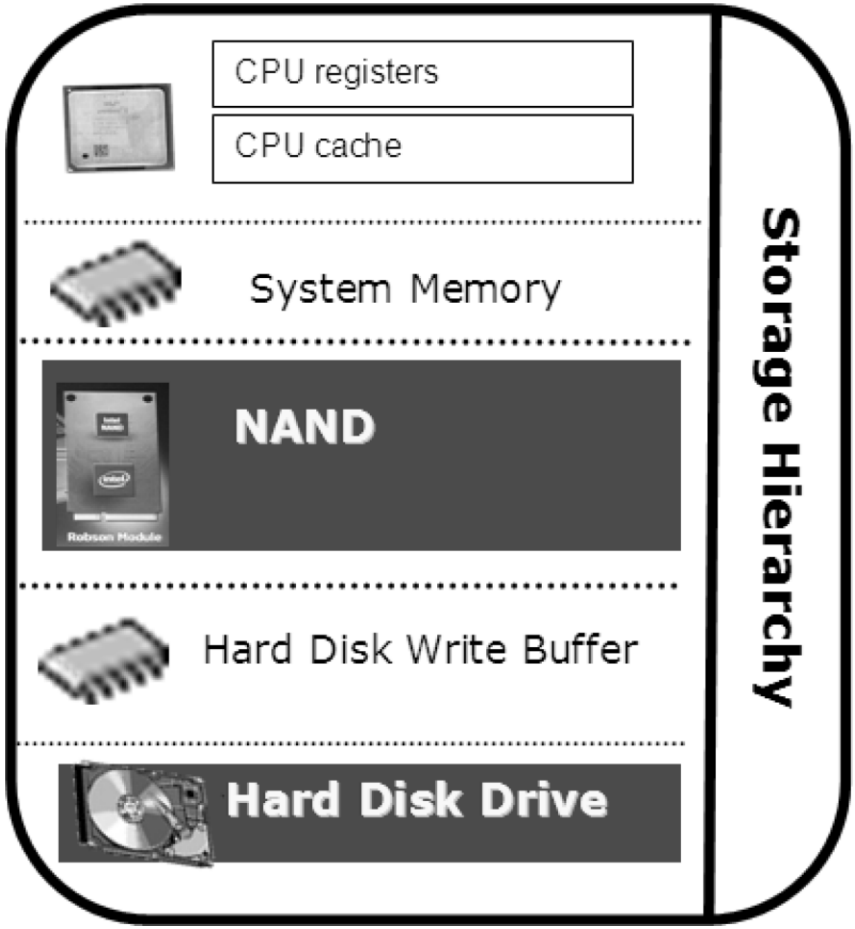
Cache Management And Caching Policies (10/15)- Error Handling

- When an error occurs, the speculative planning must be rolled back and the requests replanned
- To facilitate such roll-back, maintain two copies of metadata for each cacheline
  - Predictive metadata
  - Snapshot metadata

Cache Management And Caching Policies (11/15)- Error Handling

- If any operation planned in a particular phase of a particular request fails, then
  - Any subsequent phase of the same request are aborted
  - The predictive metadata is rolled back to the values in the snapshot metadata
  - Any request that was blocked by that request is aborted in its entirety
  - The request is placed back on the entry queue

Cache Management And Caching Policies (12/15)- Alternating Layers of Volatile and Nonvolatile Storage



Cache Management And Caching Policies(13/15)-Caching Policies for Improving Performance

- The caching policies used in ITM is that of limiting insertion into the cache based on the size of request

- Aggressively limiting writes into the cache avoids unnecessary NAND writes, which are more expensive than reads

Cache Management And Caching Policies(14/15)-Caching Policies for Improving Performance

- Allows both disk and NAND to be used in parallel
- Avoids filling the cache with streaming access patterns
- Another policy is to insert data from any requests to the page file

Cache Management And Caching Policies(15/15)- Caching Policies for Power Savings

- Hard-disk-drive power is one important component of the overall system power consumption

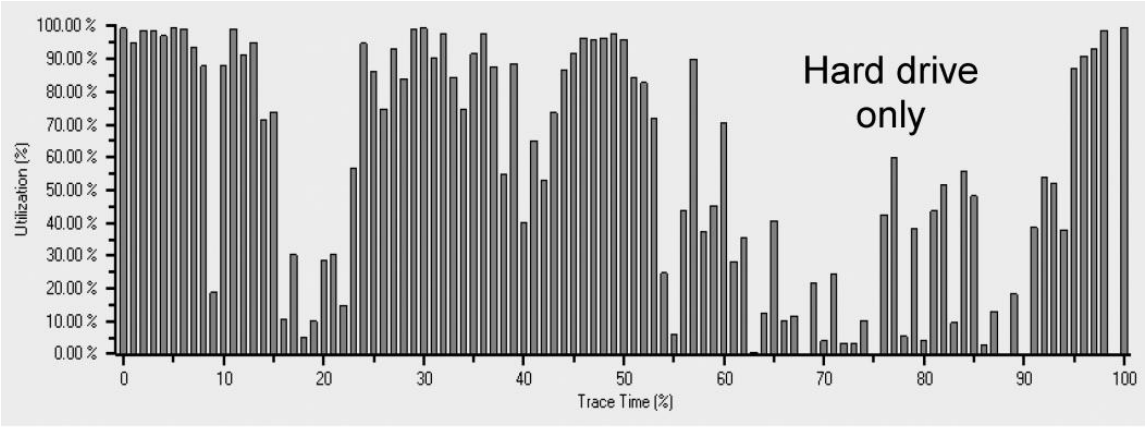
- Keeping the hard drive spun-down requires that the NAND cache absorb all traffic, both reads and writes

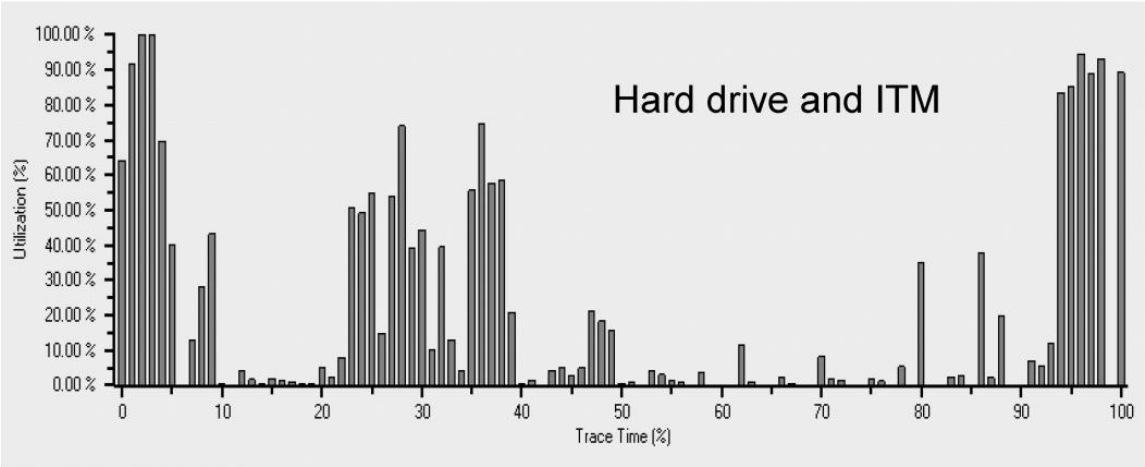
Results(1/3)

- Performance Improvements
- Power Savings

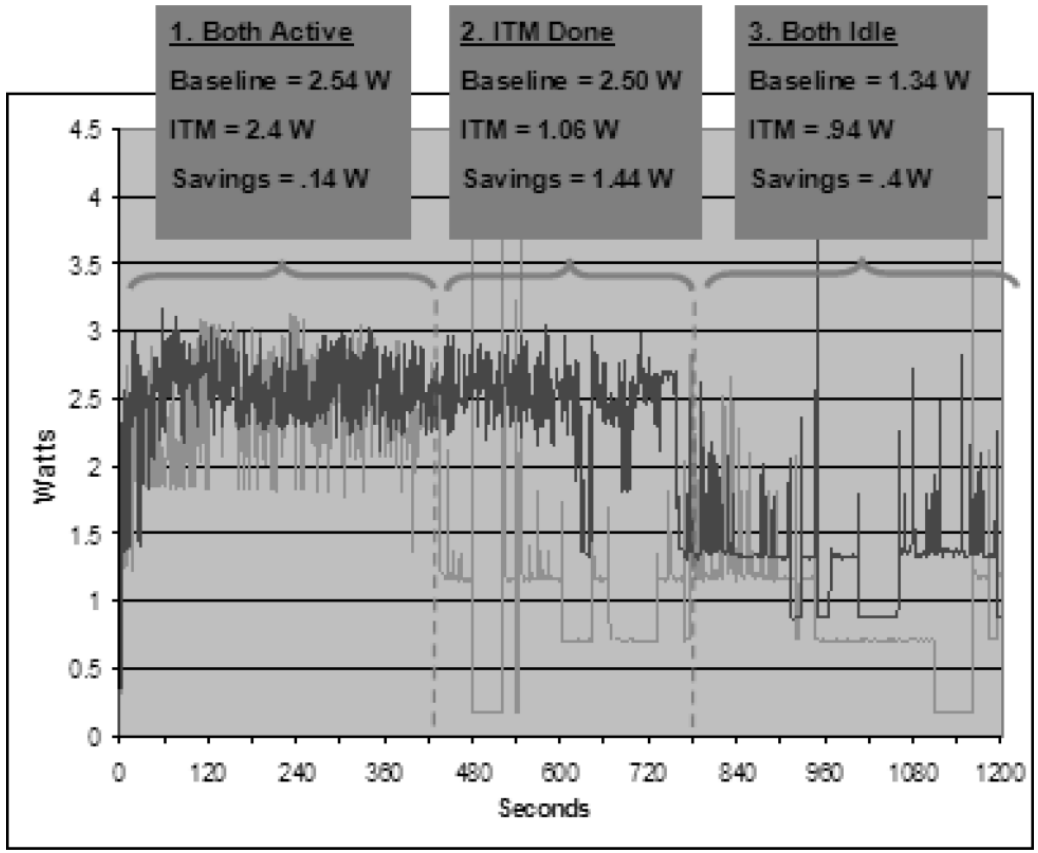
Results(2/3)- Performance Improvements

- Analyze trace data showing disk utilization of a Photoshop trace





Results(3/3)-Power Savings



- Disk power consumption
  - Darker line shows hard-drive-only configuration
  - Lighter line shows hard drive and ITM

Conclusions

- Intel Turbo Memory is a novel NAND-based disk-cache solution to alleviate the I/O bottleneck problem on PCs
- Not only does it deliver good performance and power savings, but it also paves the path for smaller, lighter, more reliable systems