A Survey of Power-Saving Techniques for Storage Systems

An-I Andy Wang

Florida State University

May 3-4

Why Care about the Energy Consumption of Storage?

- Relevant for mobile devices
  - 8% for laptops
- Energy consumption of disk drives
  - 40% of electricity cost for data centers more energy → more heat → more cooling → lower computational density → more space → higher costs
- Cost aside, fixed power infrastructure
  - Need to power more with less

[Lampe-Onnerud 2008;Gallinaro 2009; Schulz 2010]

Compared to other components

- CPU
  - Xeon X5670
    - 16W per core when active
    - Near zero idle power
- Disks
  - Hitachi Deskstar 7K1000
    - 12W active
    - 8W idle

How about flash?

- Samsung SSD SM825
  - 1.8/3.4W active (read/write)
  - 1.3W idle
  - 10X $/GB
  - Green  but maybe too green 

- Energy-efficient techniques need to meet diverse constraints
  - Total cost of ownership (TCO), performance, capacity, reliability, etc.

TCO Example

- Facebook: 100 petabytes (10^{15})
- Assumption $1/year for 1W/hour
- Use Hitachi Deskstar 7K1000 1TB disks
  - $7M for 90K disks, $960K/year for electricity
- Use Hitachi Z5K500 500GB laptop disks
  - $11M for 190K disks, $261K/year for electricity
- Flash? Don't even think about it.

[Ziegler 2012]

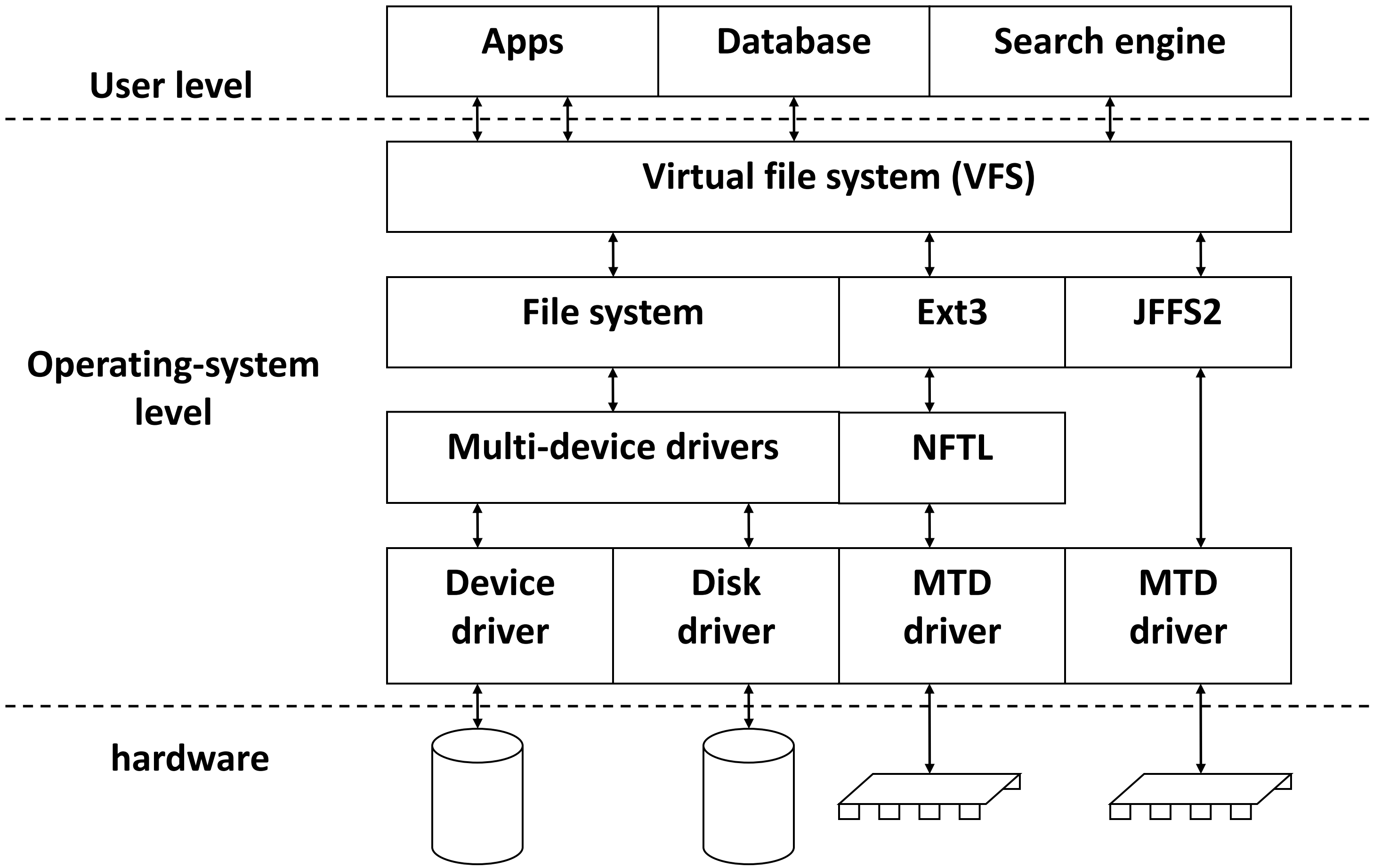
Worse…

- Exponential growth in storage demand
  - Data centers
  - Cloud computing
- Limited growth in storage density
  - For both disk and flash devices
- Implications
  - Storage can be both a performance and an energy bottlenecks…

Roadmap

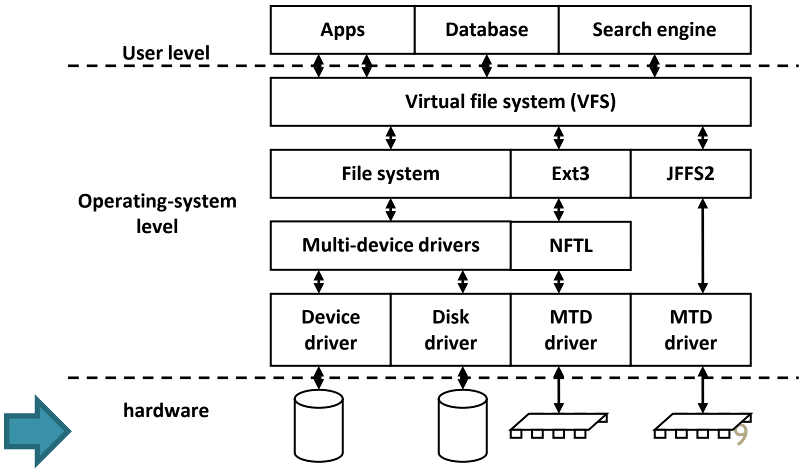
- Software storage stack overview
- Power-saving techniques for different storage layers
  - Hardware
  - Device/multi-device driver
  - File system
  - Cache
  - Application
- By no means exhaustive…

Software Storage Stack



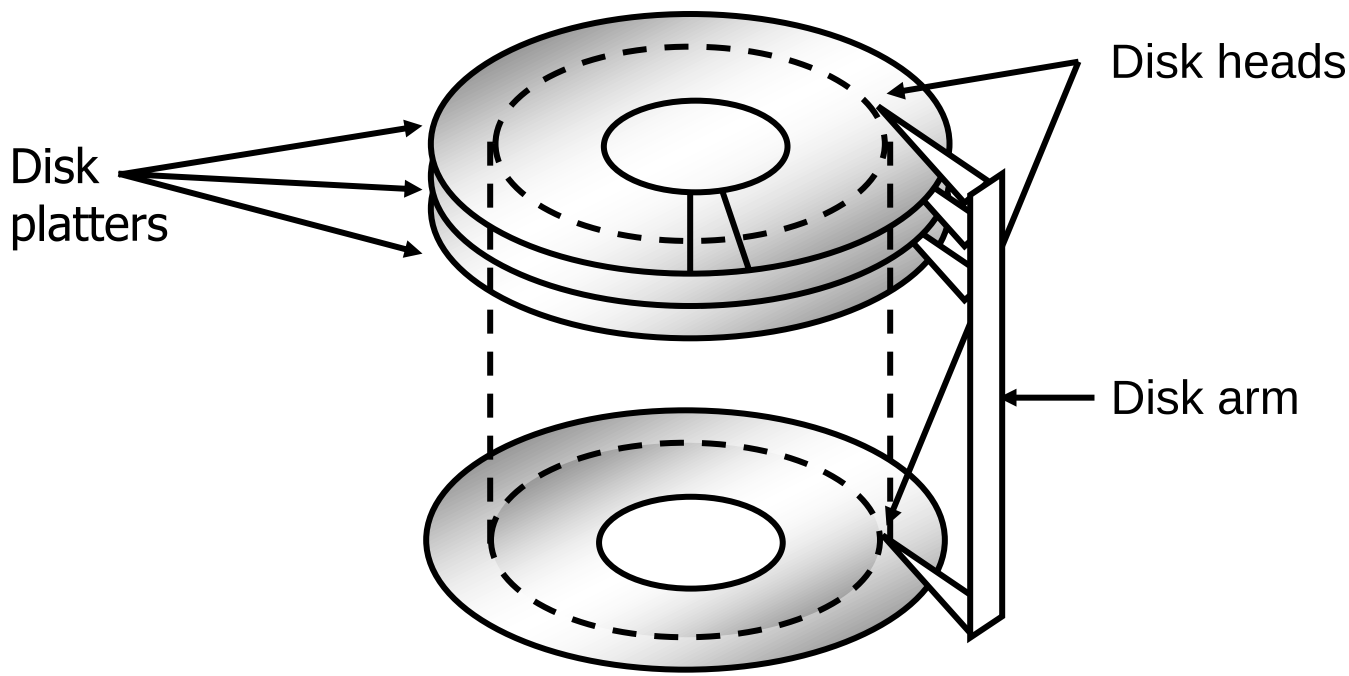
Hardware Level

- Common storage media
  - Disk drives
  - Flash devices
- Energy-saving techniques
  - Higher-capacity disks
  - Smaller rotating platter
  - Slower/variable RPM
  - Hybrid drives



Hard Disk

- 50-year-old storage technology
- ***Disk access time***
  - Seek time + rotational delay + transfer time



Energy Modes

- Read/write modes
- Active mode (head is not parked)
- Idle mode (head is parked, disk spinning)
- Standby mode (disk is spun down)
- Sleep mode (minimum power)

Hitachi Deskstar 7K1000 1TB

- Average access time: 13ms
  - Seek time: 9ms
  - 7200 RPM: 4ms for ½ rotation
  - Transfer time for 4KB: 0.1ms
    - Transfer rate of 37.5 MB/s
- Power
  - 30W startup
  - 12W active, 8W idle, 3.7W low RPM idle

Hitachi Deskstar 7K1000 1TB (continued)

- Reliability
  - 50K power cycles (27 cycles/day for 5 years)
  - Error rate: 1 in 100TB bytes transferred
    - 350GB/day for 5 years
    - Limits the growth in disk capacity
- Price: $80

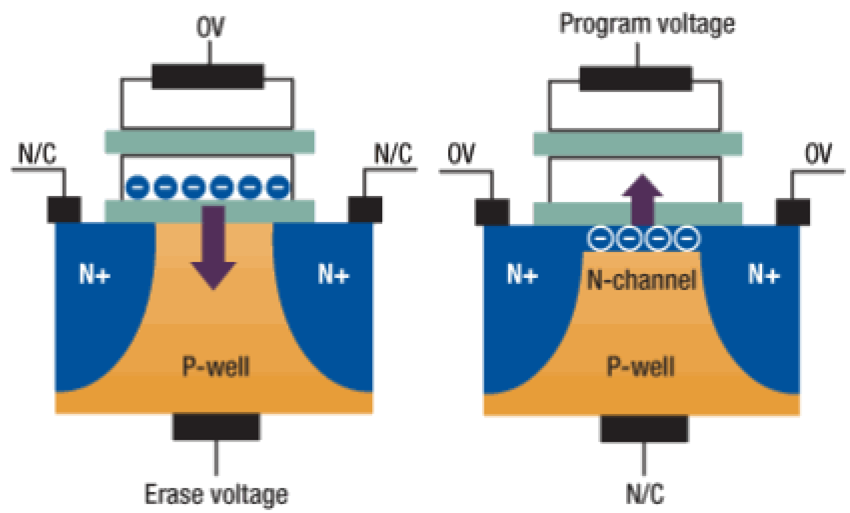
Hitachi Z5K500 500GB ($61)

- Average access time: 18ms
  - Seek time: 13ms
  - 5400 RPM: 5ms for ½ rotation
  - Transfer time for 4KB: 0.03ms
    - Transfer rate of 125.5 MB/s
- Power
  - 4.5W startup
  - 1.6W active, 1.5W idle, 0.1W sleep
- Reliability: 600K power cycles (13/hr)

Flash Storage Devices

- A form of solid-state memory
  - Similar to ROM
  - Holds data without power supply
- Reads are fast
- Can be written once, more slowly
- Can be erased, but very slowly
- Limited number of erase cycles before degradation (10,000 – 100,000)

Physical Characteristics



NOR Flash

- Used in cellular phones and PDAs
- Byte-addressable
  - Can write and erase individual bytes
  - Can execute programs

NAND Flash

- Used in digital cameras and thumb drives
- Page-addressable
  - 1 ***flash page*** ~= 1 disk block (1-4KB)
  - Cannot run programs
- Erased in ***flash blocks***
  - Consists of 4 - 64 flash pages

Writing In Flash Memory

- If writing to empty flash page (~disk block), just write
- If writing to previously written location, erase it, then write
- While erasing a flash block
  - May access other pages via other IO channels
  - Number of channels limited by power (e.g., 16 channels max)

Implications of Slow Erases

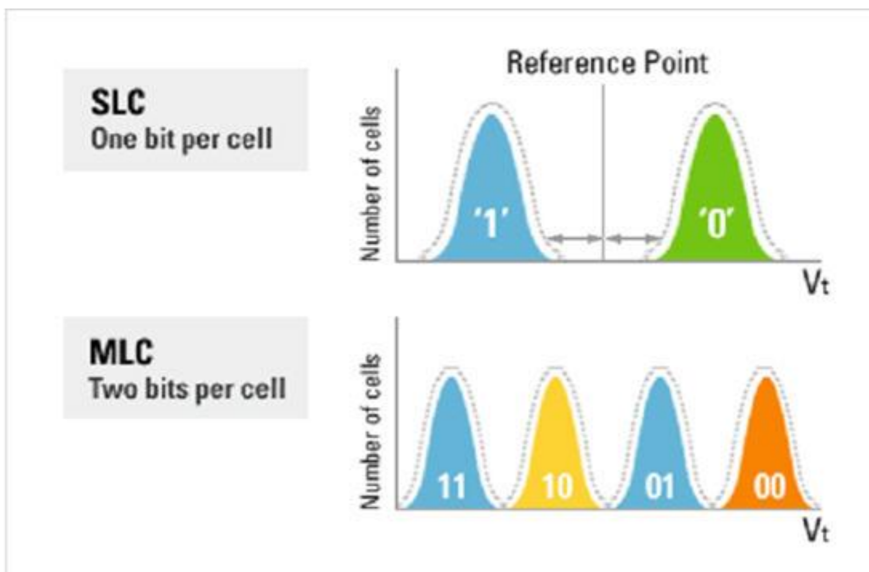
- Use of ***flash translation layer (FTL)***
  - Write new version elsewhere
  - Erase the old version later

Implications of Limited Erase Cycles

- ***Wear-leveling mechanism***
  - Spread erases uniformly across storage locations

Multi-level cells

- Use multiple voltage levels to represent bits



Implications of MLC

- Higher density lowers price/GB
- Number of voltage levels increases exponentially for linear increase in density
  - Maxed out quickly
- Reliability and performance decrease as the number of voltage levels increases
  - Need a guard band between two voltage levels
  - Takes longer to program
    - Incremental stepped pulse programming

[Grupp et al. 2012]

Samsung SM825 400GB

- Access time (4KB)
  - Read: 0.02ms
  - Write: 0.09ms
  - Erase: Not mentioned
  - Transfer rate: 220 MB/s
- Power
  - 1.8/3.4W active (read/write)
  - 1.3W idle

Samsung SM825 (continued)

- Reliability: 17,500 erase cycles
  - Can write 7PB before failure
    - 4 TB/day, 44MB/s for 5 years
    - Perhaps wear-leveling is no longer relevant
      - Assume 2% content change/day + 10x amplification factor for writes = 80 GB/day
  - Error rate: 1 in 13PB
- Price: not released yet
  - At least $320 based on its prior 256GB model

Overall Comparisons

- Average disks
  - + Cheap capacity
  - + Good bandwidth
  - - Poor power consumption
  - - Poor average access times
  - - Limited number of power cycles
  - - Density limited by error rate
- Flash devices
  - + Good performance
  - + Low power
  - - More expensive
  - - Limited number of erase cycles
  - - Density limited by number of voltage levels

HW Power-saving Techniques

- Higher-capacity disks
- Smaller disk platters
- Disks with slower RPMs
- Variable-RPM disks
- Disk-flash hybrid drives

[Battles et al. 2007]

Higher-capacity Disks

- Consolidate content with fewer disks

+ Significant power savings

- Significant decrease in parallelism

Smaller Platters, Slower RPM

- IBM Microdrive 1GB ($130)
  - Average access time: 20ms
    - Seek time: 12ms
    - 3600 RPM: 8ms for ½ rotation
    - Transfer time for 4KB: 0.3ms
      - Transfer rate of 13 MB/s
  - Power: 0.8W active, 0.06W idle
  - Reliability: 300K power cycles

Smaller Platters, Slower RPM

- IBM Microdrive 1GB ($130)

  + Low power

  + Small physical dimension (for mobile devices)

  - Poor performance

  - Low capacity

  - High Price

Variable RPM Disks

- Western Digital Caviar Green 3TB
  - Average access time: N/A
    - Peak transfer rate: 123 MB/s
  - Power: 6W active, 5.5W idle, 0.8W sleep
  - Reliability: 600K power cycles
  - Cost: $222

Variable RPM Disks

- Western Digital Caviar Green 3TB

  + Low power

  + Capacity beyond mobile computing

  - Potentially high latency

  - Reduced reliability?

    - Switching RPM may consume power cycle count

  - Price somewhat higher than disks with the same capacity

Hybrid Drives

- Seagate Momentus XT 750GB ($110)
  - 8GB flash
  - Average access time: 17ms
    - Seek time: 13ms
    - 7200 RPM: 4ms for ½ rotation
    - Transfer time for 4KB: Negligible
  - Power: 3.3W active, 1.1W idle
  - Reliability: 600K power cycles

Hybrid Drives

- Seagate Momentus XT 750GB ($110)

  + Good performance with good locality

    - Especially if flash stores frequently accessed read-only data

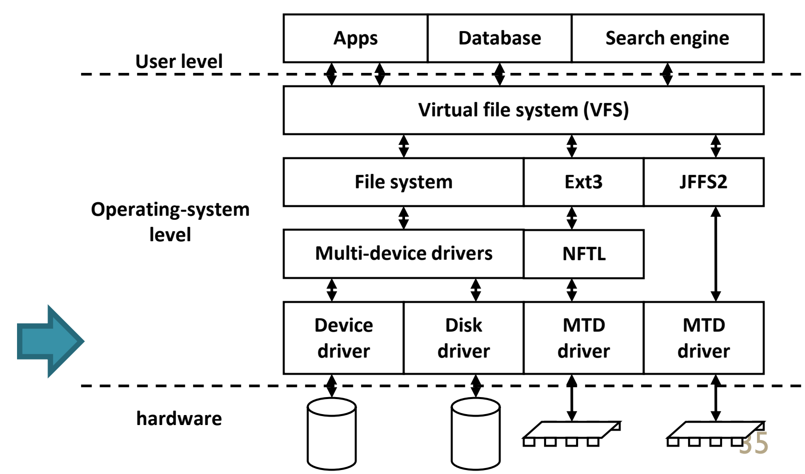
  - Reduced reliability?

    - Flash used as write buffer may not have enough erase cycles

  - Some price markups

Device-driver Level Techniques

- General descriptions
- Energy-saving techniques
  - Spin down disks
  - Use flash to cache disk content

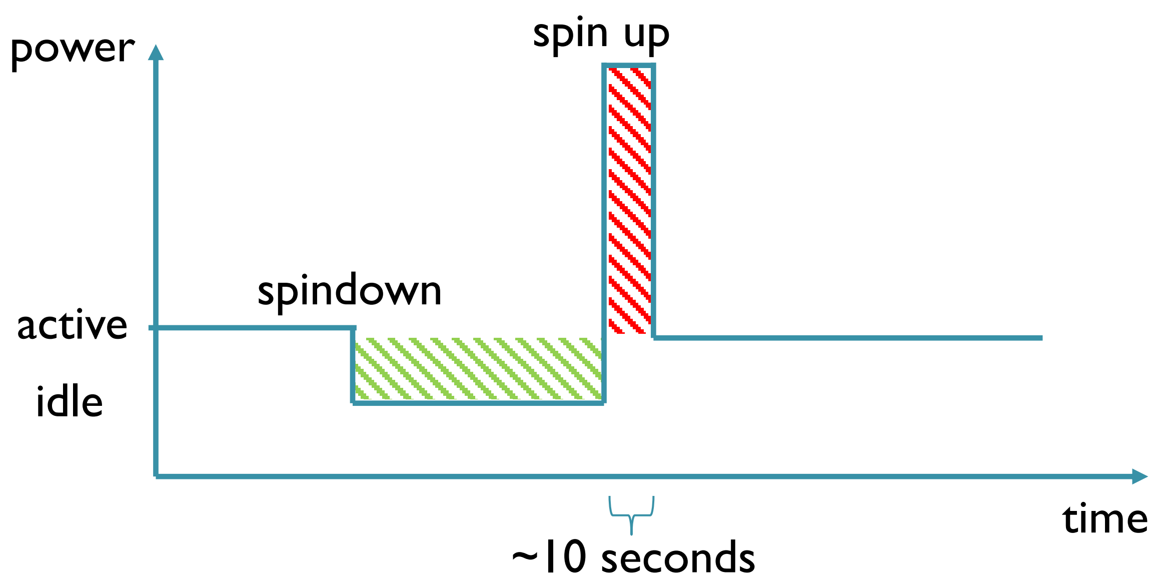


Device Drivers

- Carry out medium- and vendor-specific operations and optimizations
- Examples
  - Disk
    - Reorder requests according to seek distances
  - Flash
    - Remap writes to avoid erases via FTL
    - Carry out wear leveling

Spin down Disks When Idle

- Save power when
  - Power saved > power needed to spin up



Spin down Disks When Idle

- Prediction techniques
  - Whenever the disk is idle for more than x seconds (typically 1-10 seconds)
  - Probabilistic cost-benefit analysis
  - Correlate sequences of program counters to the length of subsequent idle periods

[Douglis et al. 1994; Li et al. 1994; Krishnan et al. 1999; Gniady et al. 2006]

Spin down Disks When Idle

+ No special hardware
- Potentially high latency at times
- Need to consider the total number of power cycles

Use Flash for Caching

- FlashCache
  - Sits between DRAM and disk
  - Reduces disk traffic for both reads and writes
    - Disk switched to low power modes if possible
  - A read miss brings in a block from disk
  - A write request first modifies the block in DRAM
    - When DRAM is full, flushed the block to flash
    - When flash is full, flushed to disk

[Marsh et al. 1994]

Use Flash for Caching

- FlashCache
  - Flash has three LRU lists
    - Free list (already erased)
    - Clean list (can be erased)
    - Dirty list (needs to be flushed)
  - Identified the applicability of the LFS management on flash

+ Reduces energy usage by up to 40%

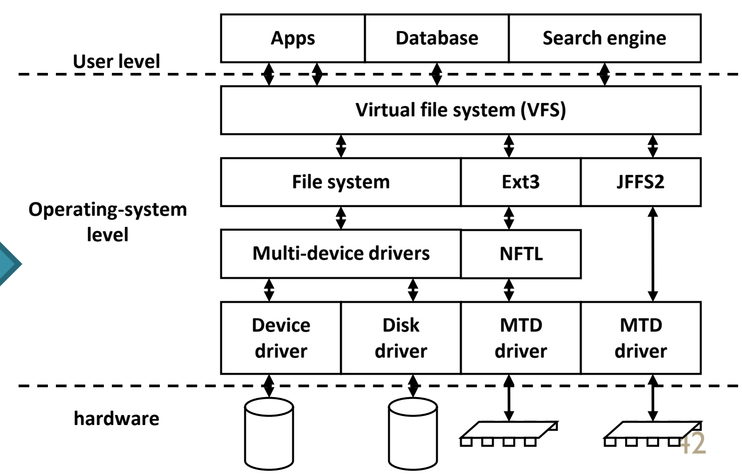
+ Reduces overall response time up to 70%

- Increases read response time up to 50%

[Marsh et al. 1994]

Multi-device-level Techniques

- General descriptions
  - Common software RAID classifications
- Energy-saving techniques
  - Spin down individual disks
  - Use cache disks
  - Use variable RPM disks
  - Regenerate content
  - Replicate data
  - Use transposed mirroring



Multi-device Layer

- Allows the use of multiple storage devices
  - But increases the chance of a single device failure
- ***RAID: Redundant Array of Independent Disks***
  - Standard way of organizing and classifying the reliability of multi-disk systems

RAID Level 0

- No redundancy
- Uses block-level ***striping*** across disks
  - 1st block stored on disk 1, 2nd block stored on disk 2, and so on
- Failure causes data loss

RAID Level 1 (Mirrored Disks)

- Each disk has second disk that mirrors its contents
  - Writes go to both disks
  - No data striping

+ Reliability is doubled

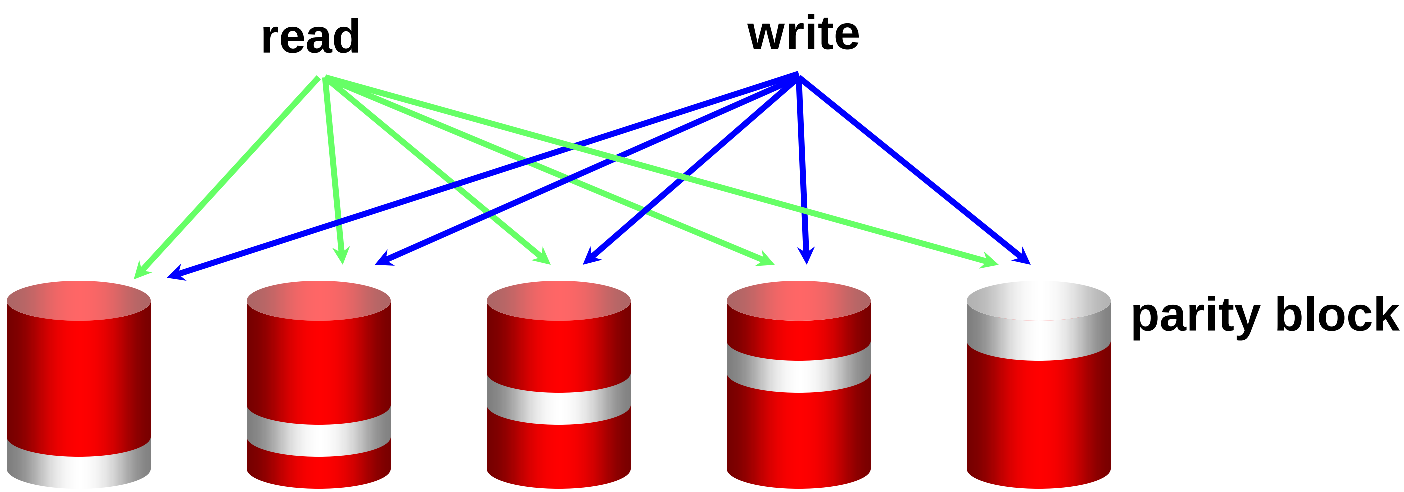
+ Read access faster

- Write access slower

- Double the hardware cost

RAID Level 5 (continued)

- Data striped in blocks across disks
- Each stripe contains a parity block
- Parity blocks from different stripes spread across disks to avoid bottlenecks



RAID Level 5

+ Parallelism

+ Can survive a single-disk failure

- Small writes involve 4 IOs
  - Read data and parity blocks
  - Write data and parity blocks
    - New parity block

      = old parity block ⊕ old data block ⊕ new data block

Other RAID Configurations

- RAID 6
  - Can survive two disk failures
- RAID 10 (RAID 1+0)
  - Data striped across mirrored pairs
- RAID 01 (RAID 0+1)
  - Mirroring two RAID 0 arrays
- RAID 15, RAID 51

Spin down Individual Drives

- Similar to the single-disk approach

\+ No special hardware

\- Not effective for data striping

[Gurumurthi et al. 2003]

Dedicated Disks for Caching

- MAID (Massive Array of Idle Disks)
  - Designed for archival purposes
  - Dedicate a few disks to cache frequently referenced data
  - Updates are deferred

+ Significant energy savings

- Not designed to maximize parallelism

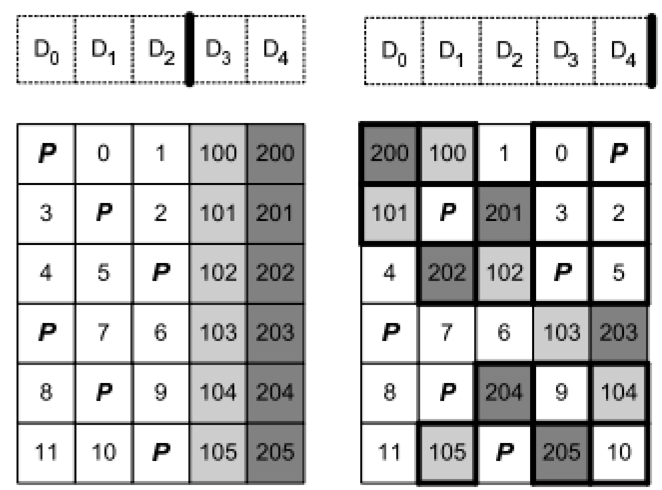
[Colarelli and Grunwald 2002]

Use Variable RPM Disks

- Hibernator
  - Periodically changes the RPM setting of disks
    - Based on targeted response time
  - Reorganizes blocks within individual stripes
    - Blocks are assigned to disks based on their temperatures (hot/cold)

+ Good energy savings

- Data migration costs



[Zhu et al. 2005]

Flash + Content Regeneration

- EERAID and RIMAC
  - Assumes hardware RAID
  - Adds flash to RAID controller to buffer updates
    - Mostly to retain reliability characteristics
    - Flushes changes to different drives periodically
      - Lengthen idle periods

[Li et al. 2004; Yao and Wang 2006]

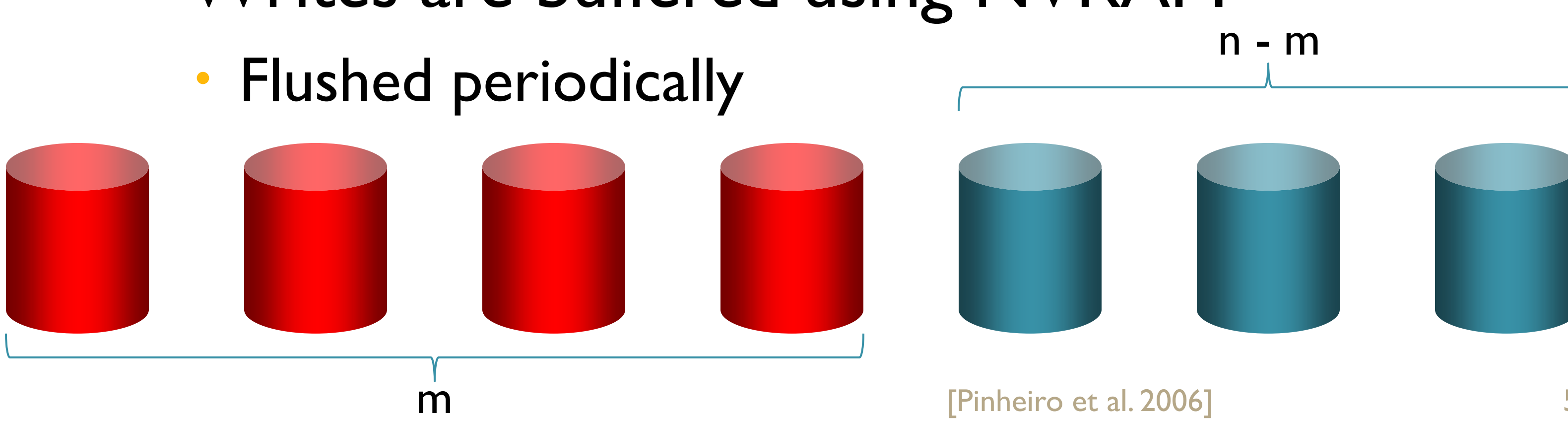
Flash + Content Regeneration (continued)

- EERAID and RIMAC
  - Use blocks from powered disks to regenerate the block from sleeping disk
    - Suppose B1, B2, B3 are in a stripe
    - Read B1 = read B2, read B3, XOR(B2, B3)
  - Up to 30% energy savings
    - Benefits diminishes as the number of disks increases
  - Can improve performance for ≤ 5 disks
  - Can degrade performance for > 5 disks

[Li et al. 2004; Yao and Wang 2006]

Flash + Content Regeneration (continued)

- Use (n, m) erasure encoding
  - Any m out of n blocks can reconstruct the original content
  - Separate m original disks from n – m disks with redundant information
  - Spin down n – m disks during light loads
  - Writes are buffered using NVRAM
    - Flushed periodically



[Pinheiro et al. 2006]

Flash + Content Regeneration (continued)

- Use (n, m) erasure encoding
  - \+ Can save significant amount of power
  - \+ Can use all disks under peak load
  - \- Erasure encoding can be computationally expensive

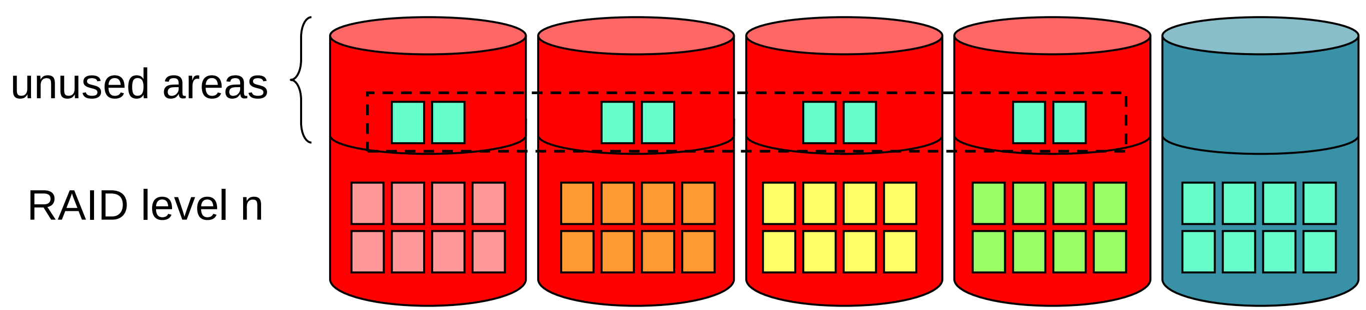
[Pinheiro et al. 2006]

Replicate Data

- PARAID (Power-aware RAID)
- Replicate data from disks to be spun down to spare areas of active disks

+ Preserves peak performance, reliability

- Requires spare storage areas



[Weddle et al. 2007]

Transposed Mirroring

- Diskgroup
  - Based on RAID 0 + 1 (mirrored RAID 0)
    - With a twist

1st RAID 0				2nd RAID 0			
Disk 0	Disk 1	Disk 2	Disk 3	Disk 0	Disk 1	Disk 2	Disk 3
A1	B1	C1	D1	A1	A2	A3	A4
A2	B2	C2	D2	B1	B2	B3	B4
A3	B3	C3	D3	C1	C2	C3	C4
A4	B4	C4	D4	D1	D2	D3	D4

[Lu et al. 2007]

Transposed Mirroring

- Diskgroup
  - Use flash to buffer writes

+ Can power on disks in the 2nd RAID 0 incrementally to meet performance needs

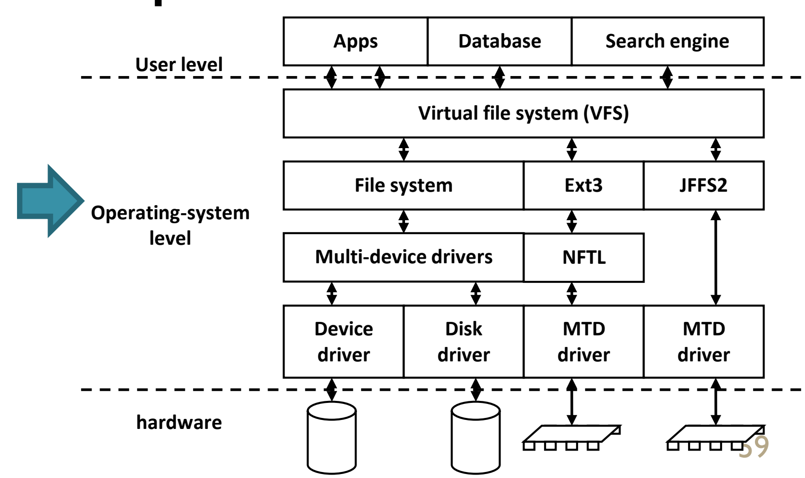
- Reduced reliability

1st RAID 0				2nd RAID 0			
Disk 0	Disk 1	Disk 2	Disk 3	Disk 0	Disk 1	Disk 2	Disk 3
A1	B1	C1	D1	A1	A2	A3	A4
A2	B2	C2	D2	B1	B2	B3	B4
A3	B3	C3	D3	C1	C2	C3	C4
A4	B4	C4	D4	D1	D2	D3	D4

[Lu et al. 2007]

File-system-level Techniques

- General descriptions
- Energy-saving techniques
  - Hot and cold tiers
  - Replicate data
  - Access remote RAM
  - Use local storage as backup



File Systems

- Provide names and attributes to blocks
- Map files to blocks
- Note: a file system does not know the physical medium of storage
  - E.g., disks vs. flash
  - A RAID appears as a logical disk

File Systems

- Implications
  - Energy-saving techniques below the file system should be usable for all file systems
  - File-system-level techniques should be agnostic to physical storage media
- In reality, many energy-saving techniques are storage-medium-specific
  - Cannot be applied for both disk and flash
  - Example: data migration is ill-suited for flash

Hot and Cold Tiers

- Nomad FS
- Uses PDC (Popular Data Concentration)
  - Exploits Zipf's distribution on file popularity
  - Use multi-level feedback queues to sort files by popularity
    - N LRU queues with exponentially growing time slices
    - Demote a file if not referenced within time slice
    - Promote a file if referenced > 2^N times

[Pinheiro and Blanchini 2004]

Hot and Cold Tiers

- PDC
  - Periodically migrate files to a subset of disks
    - Sorted by the popularity
    - Each disk capped by either file size or load (file size/interarrival time)

+ Can save more energy than MAID

- Not exploiting parallelism

- Data migration overhead

[Pinheiro and Blanchini 2004]

Replicate Data

- FS2
  - Identifies hot regions of a disk
  - Replicates file blocks from other regions to the hot region to reduce seek delays
  - Requires changes to both file-system and device-driver layers
    - File system knows the free status of blocks and file membership of blocks

[Huang et al. 2005]

Replicate Data

- FS2
  - For reads, access the replica closest to the current disk head
  - For writes, invalidate replicas
  - Additional data structures to track replicas
    - Periodically flushed

+ Saves energy and improves performance

- Need free space and worry about crashes

[Huang et al. 2005]

Access Remote RAM

- BlueFS
  - Designed for mobile devices
  - Decides whether to fetch data from remote RAM or local disk depending on the power states and relative power consumption
  - Updates persistently cached on mobile client
    - Propagated to server in aggregation via optimistic consistency semantics

[Nightingale et al. 2005]

Access Remote RAM

- BlueFS
  - One problem
    - If files are accessed one at a time over a network, and local disk is powered down, little incentive exists to power up the local disk
  - Solution
    - Provides ghost hints to local disk about the opportunity cost
  - Can reduce latency by 3x and energy by 2x

[Nightingale et al. 2005]

Use Local Storage as Backup

- GreenFS
  - Mostly designed for mobile devices
    - Can be applied to desktops and servers as well
  - When disconnected from a network
    - Use flash to buffer writes
  - When connected
    - Use flash for caching

[Joukov and Sipek 2008]

Use Local Storage as Backup

- GreenFS
  - Local disk is a backup of data stored remotely
    - Provides continuous data protection
    - Spun down most of the time
      - Access the remote data whenever possible
    - Spun up and synchronized at least once per day
      - When the device is shut down
      - When flash is near full
      - Used when network connectivity is poor
    - The number of disk power cycles are tracked and capped

[Joukov and Sipek 2008]

Use Local Storage as Backup

- GreenFS
  - Performance depends on the workloads

  + Can save some power (up to 14%)

  + Noise reduction

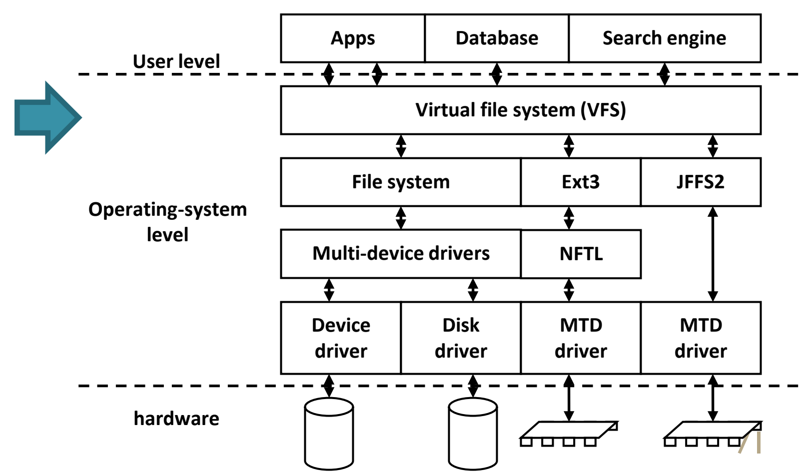
  + Better tolerance to shocks when disks are off most of the time

  - Does not work well for large files
    - Power needed for network transfers > disk
    - Assumed to be rare

[Joukov and Sipek 2008]

VFS-level Techniques

- General descriptions
- Energy-efficient techniques
  - Encourage IO bursts
  - Cache cold disks



VFS

- Enables an operation system to run different file systems simultaneously
- Handles common functions such as caching, prefetching, and write buffering

Encourage IO Bursts

- Prefetch aggressively based on monitored IO profiles of process groups
  - But not so much to cause eviction misses
- Writes are flushed once per minute
  - As opposed to 30 seconds
  - Allow applications to specify whether it is okay to postpone updates to file close
    - Access time stamps for media files
  - Preallocate memory based on write rate

[Papathanasiou and Scott 2004]

Encourage IO Bursts

- Negligible performance overhead

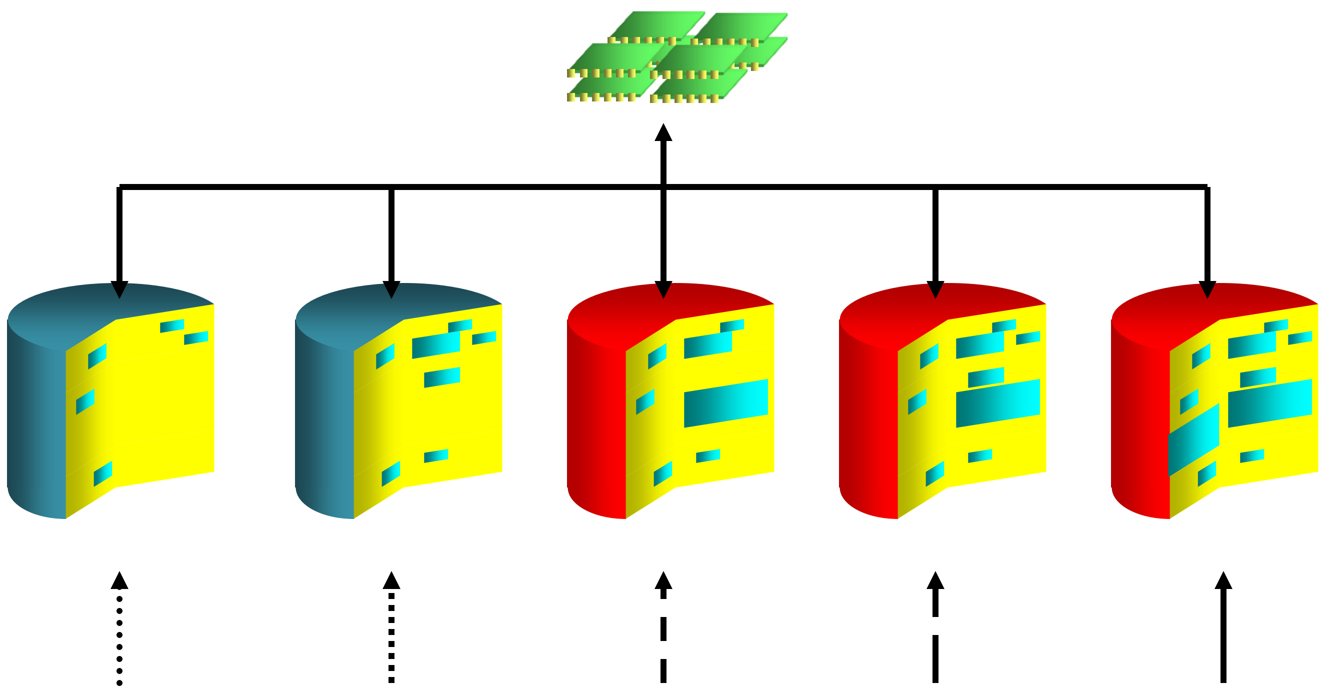
+ Energy savings up to 80%

- Somewhat reduced reliability

[Papathanasiou and Scott 2004]

Cache Cold Disks

- Cache more blocks from cold disks
  - Lengthen the idle period of cold disks
  - Slightly increase the load of hot disks



[Zhu et al. 2004]

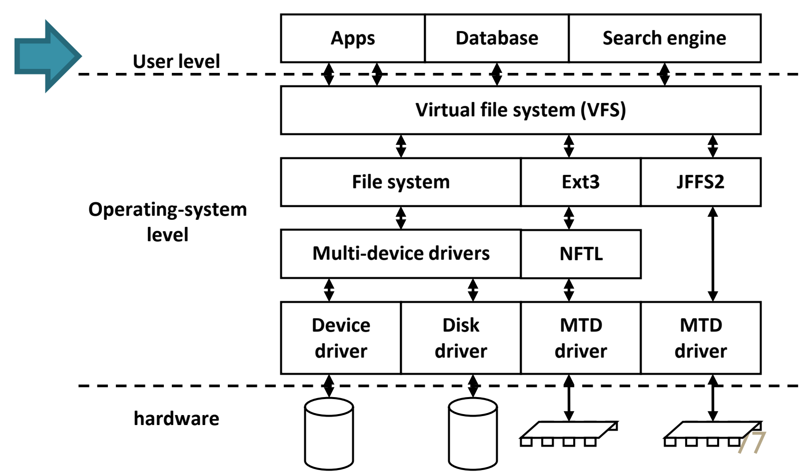
Cache Cold Disks (continued)

- Cache more blocks from cold disks
  - Characteristics of hot disks
    - Small percentage of cold misses
      - Identified by a Bloom Filter
    - High probability of long interarrival times
      - Tracked by epoch-based histograms
  - Improves energy savings by16%
  - Assumes the use of multi-speed disks
  - Assumes the use of NVRAM or a log disk to buffer writes

[Zhu et al. 2004]

Application-level Techniques

- Many ways to apply system-level techniques (e.g., buffering)
- Flash
  - Energy-efficient encoding



Energy-efficient Encoding

- Basic observation
  - Energy consumption for 10101010… = 15.6μJ
    - For 11111111 = 0.038μJ
  - Thus, avoid 10 and 01 bit patterns
- Approach: user-level encoder

Memory type	Operation	Time(μs)	Energy(μJ)
Intel MLC NOR 28F256L18	Program 00	110.00	2.37
	Program 01	644.23	14.77
	Program 10	684.57	15.60
	Program 11	24.93	0.038

[Joo et al. 2007]

Energy-efficient Encoding

- Tradeoffs
  - 35% energy savings with 50% size overhead

  + No changes to storage stack

  + Good energy savings

  - File size overhead can be significant

  - Longer latency due to larger files

  - One-time encoding cost

[Joo et al. 2007]

Large-scale Techniques

- General descriptions
  - Some can operate on RAID granularity
    - No need to handle low-level reliability issues
  - Involve distributed coordination
- Energy-efficient technique
  - Spun-up token
  - Graph coverage
  - Write offloading
  - Power proportionality

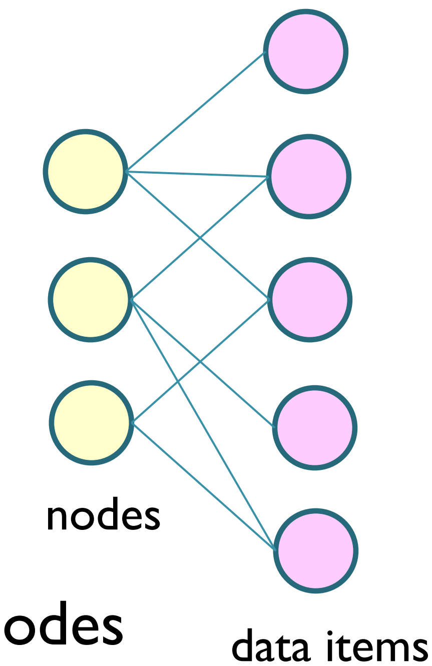
Spun-up Token

- Pergamum
  - Distributed system designed for archival
  - Used commodity low-power components
    - Each node contains a flash device, a low-power CPU, and a low-RPM disk
  - Flash stores metadata; disk data
  - Used erasure code for reliability
  - Used spun-up token to limit the number of power disks

[Storer et al. 2008]

Graph Coverage

- Problem formulation
  - A bipartite graph
    - Nodes and data items
    - An edge indicates an item's host
  - Power-saving goal
    - Minimize the number of powered nodes
    - While keeping all data items available
- For read-mostly workloads
- Writes are buffered via flash



[Harnik et al. 2009]

Graph Coverage

- SRCMap
  - For each node, replicate the working set to other nodes
  - Increases the probability of having few nodes covering the working sets for many nodes

[Verma et al. 2010]

Write Offloading

- Observations
  - Cache can handle most of read requests
  - Disks are active mostly due to writes
- Solution: write offloading
  - A volume manager redirects writes from sleeping disks to active disks
    - Invalidates obsolete content
  - Propagates updates when disks are spun up
    - E.g., read miss, no more space for offloading

[Narayanan et al. 2008]

Write Offloading

- Use versioning to ensure consistency

+ Retain the reliability semantics of the underlying RAID level

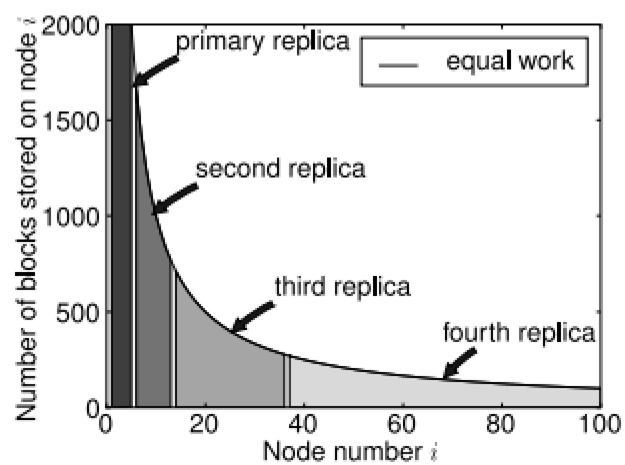
+ Can save power up to 60%
  - 36% just to spin down idle disks

+ Can achieve better average response time

- Extra latency for reads, but it's expected

Power Proportionality

- Rabbit
  - Matches power consumption with workload demands
  - Uses equal work data layout
  - Uses write offloading to handle updates



[Amur et al. 2010]

Summary

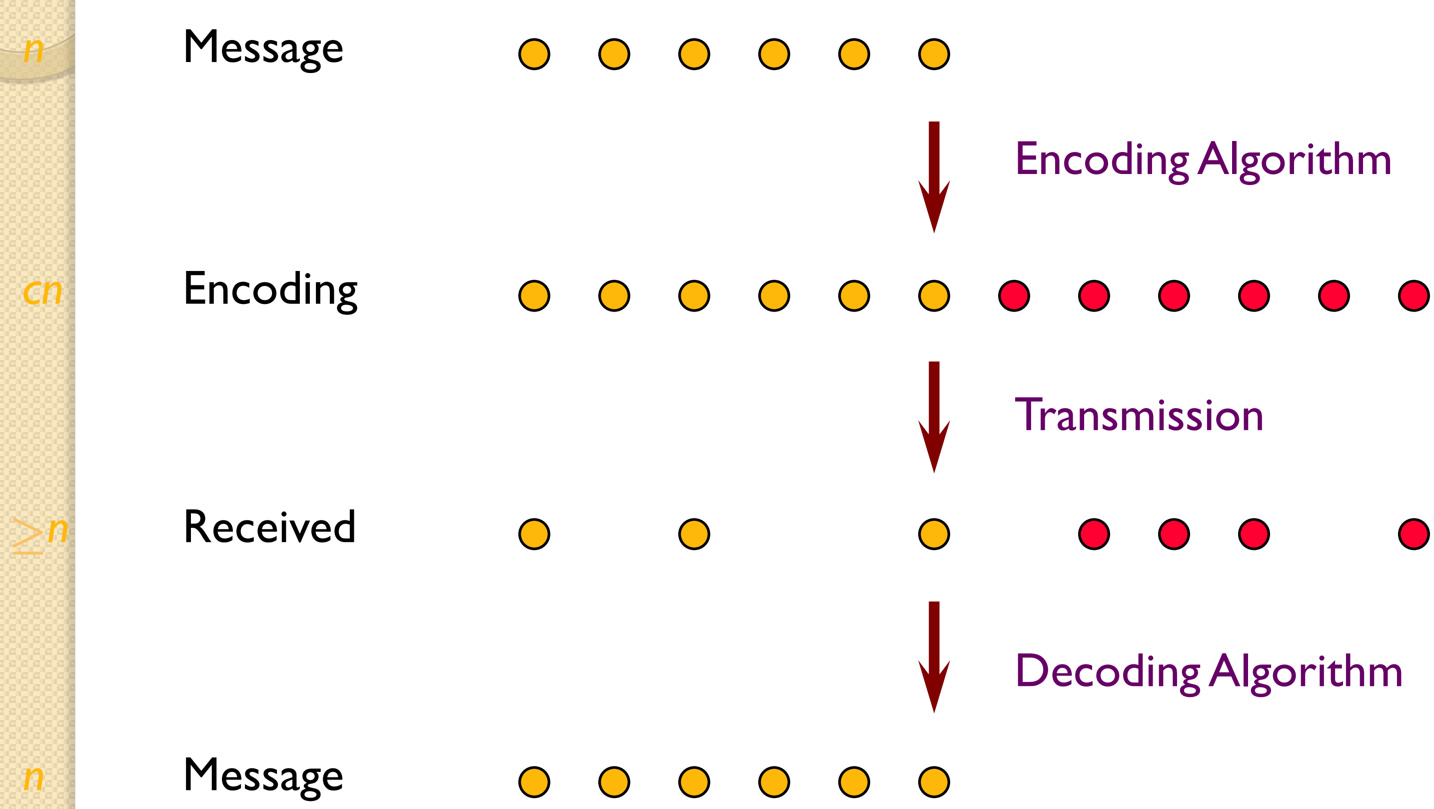
- Energy-efficient techniques
  - Specialized hardware, caching, power down devices, data regeneration, replication, special layouts, remote access, power provisioning, etc.
- Common tradeoffs
  - Performance, capacity, reliability, specialized hardware, data migration, price, etc.
- No free lunch in general…

Questions?

- Thank you!

Backup Slides

Erasure Codes



[Mitzenmacher]

Erasure Codes

- Examples
  - RAID level 5 parity code
  - Reed-Solomon code
  - Tornado code

back

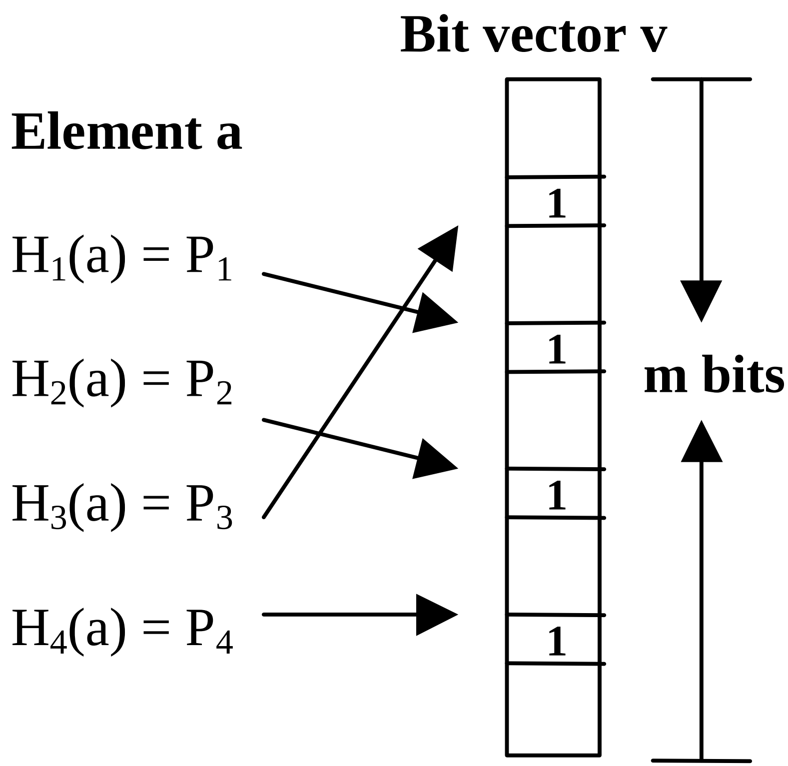
[Mitzenmacher]

Bloom Filters

- Compact data structures for a probabilistic representation of a set
- Appropriate to answer membership queries

[Koloniari and Pitoura]

Bloom Filters (cont'd)



Query for *b*: check the bits at positions $H_1(b)$, $H_2(b)$, ..., $H_4(b)$.

back

Suppose the goal is 1M IOPS…

- Samsung SM850: 11K IOPS
  - 91 units x $320?/unit = $30K + stuff (rack, space, network interconnect)
- Hitachi Deskstar 7K1000 : 77 IOPS
  - 13K units x $80/unit = $1M + 140x stuff
- If your goal is performance, forget about disks and energy costs