PARAID: A *Gear-Shifting* Power-Aware RAID

Charles Weddle, Mathew Oldham, Jin Qian, An-I Andy Wang – Florida St. University

Peter Reiher – University of California, Los Angeles

Geoff Kuenning – Harvey Mudd College

Motivation

- Energy costs are rising
  - An increasing concern for servers
    - No longer limited to laptops
- Energy consumption of disk drives
  - 24% of the power usage in web servers
  - 27% of electricity cost for data centers
  - More energy → more heat → more cooling → lower computational density → more space → more costs
- Is it possible to reduce energy consumption without degrading performance while maintaining reliability?

Challenges

- Energy
  - Not enough opportunities to spin down RAIDs
- Performance
  - Essential for peak loads
- Reliability
  - Server-class drives are not designed for frequent power switching

Existing Work

- Most trade performance for energy savings directly
  - e.g. vary speed of disks
- Most are simulated results

Observations

- RAID is configured for peak performance
  - RAID keeps all drives spinning for light loads
- Unused storage capacity
  - Over-provision of storage capacity
  - Unused storage can be traded for energy savings
- Fluctuating load
  - Cyclic fluctuation of loads
  - Infrequent on-off power transitions can be effective

Performance vs. Energy Optimizations

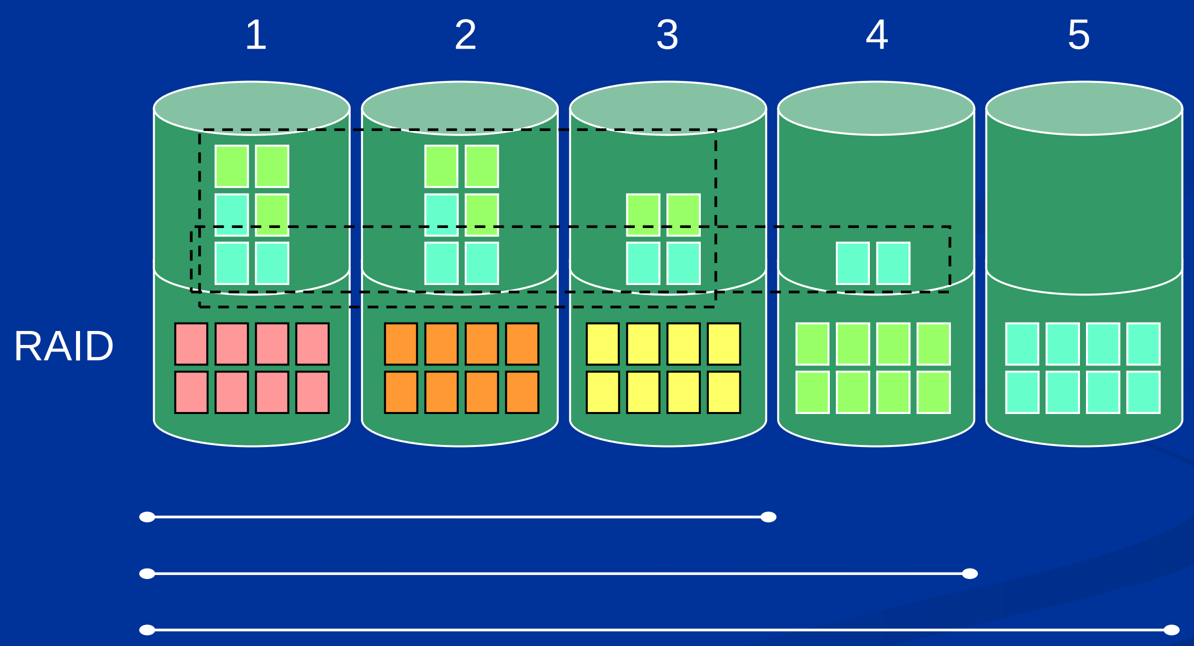
- Performance benefits
  - Realized under heavy loads
- Energy benefits
  - Realized instantaneously

Power-Aware RAID

- Skewed striping for energy savings
- Preserving peak performance
- Maintaining reliability
- Evaluation
- Conclusion
- Questions

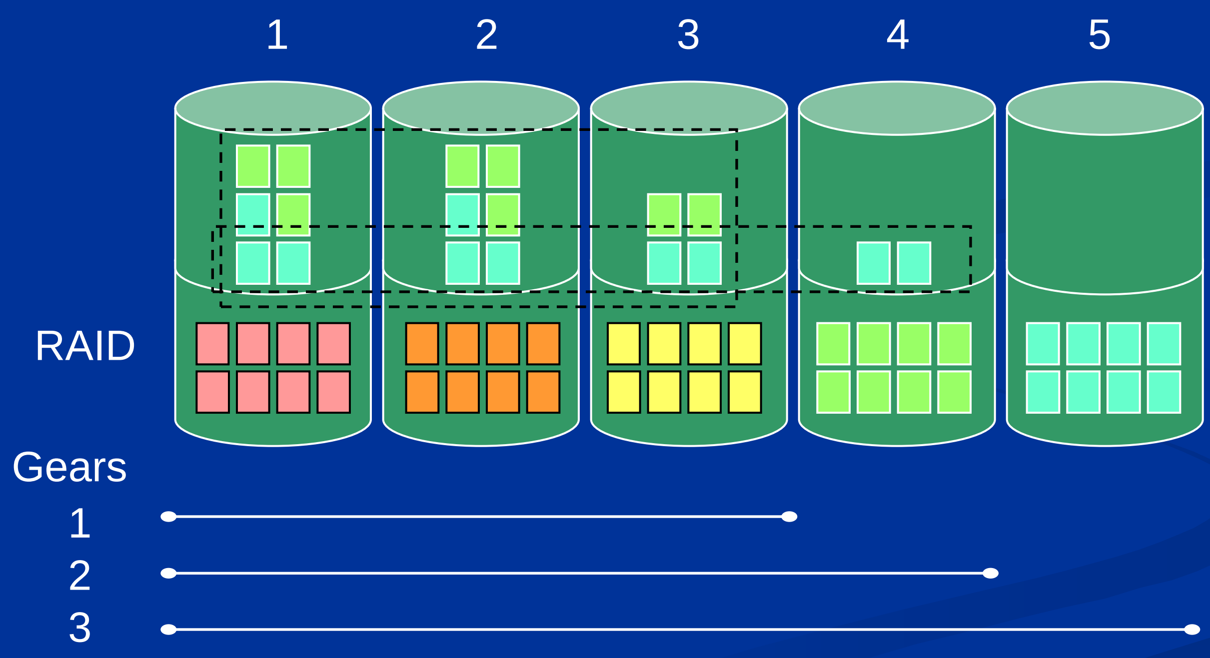
Skewed Striping for Energy Saving

- Use over-provisioned spare storage
  - Organized into hierarchical overlapping subsets



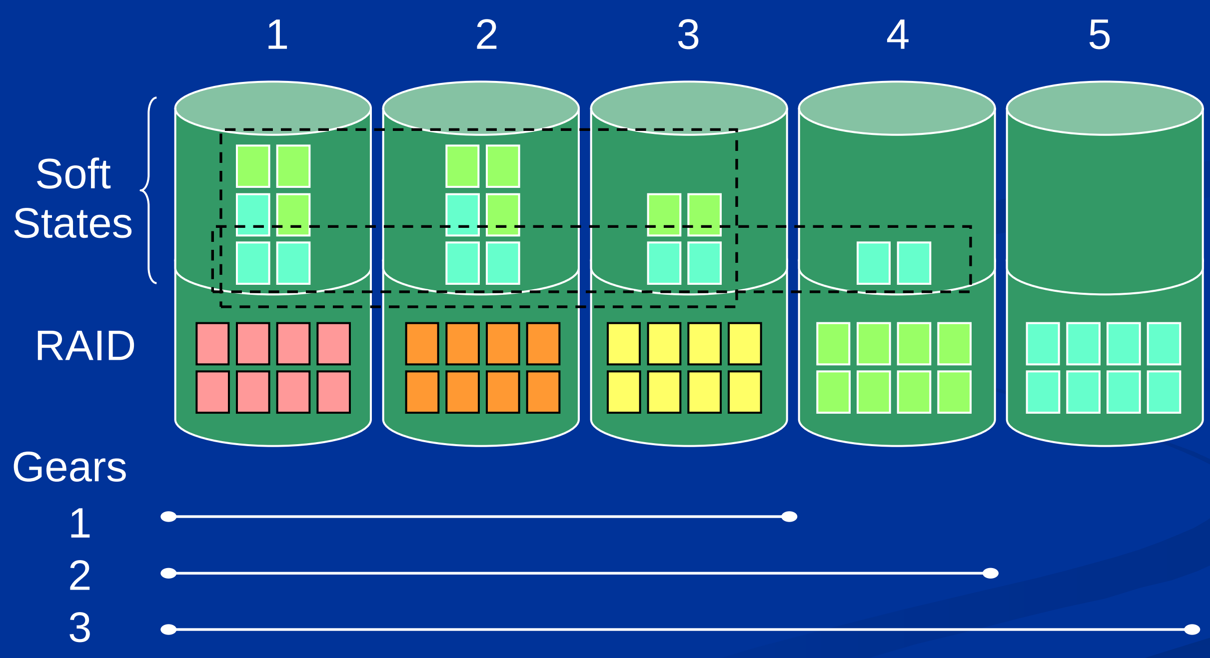
Skewed Striping for Energy Saving

- Each set analogous to gears in automobiles



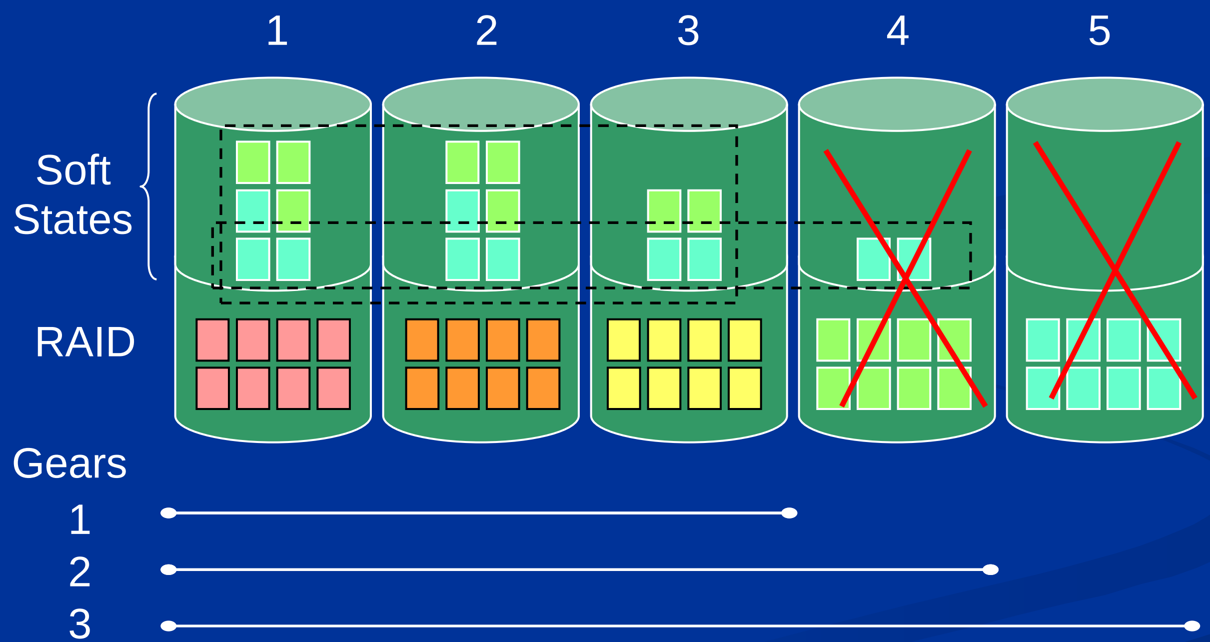
Skewed Striping for Energy Saving

- Soft states can be reclaimed for space
  - Persist across reboots



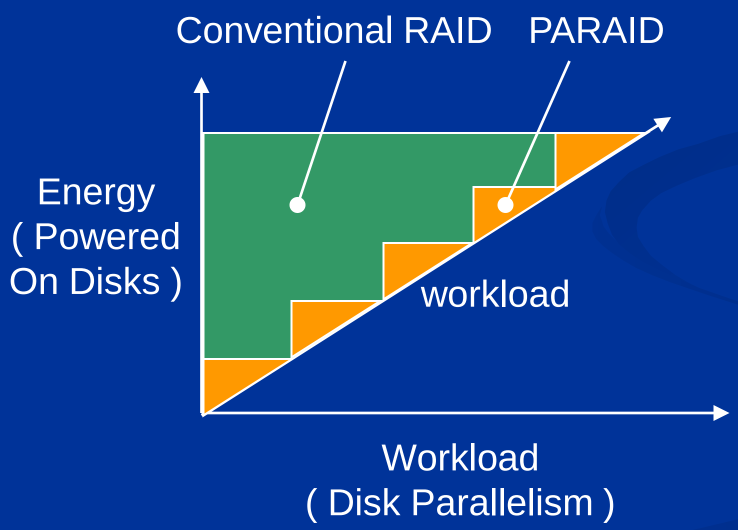
Skewed Striping for Energy Saving

- Operate in gear 1
- Disks 4 and 5 are powered off



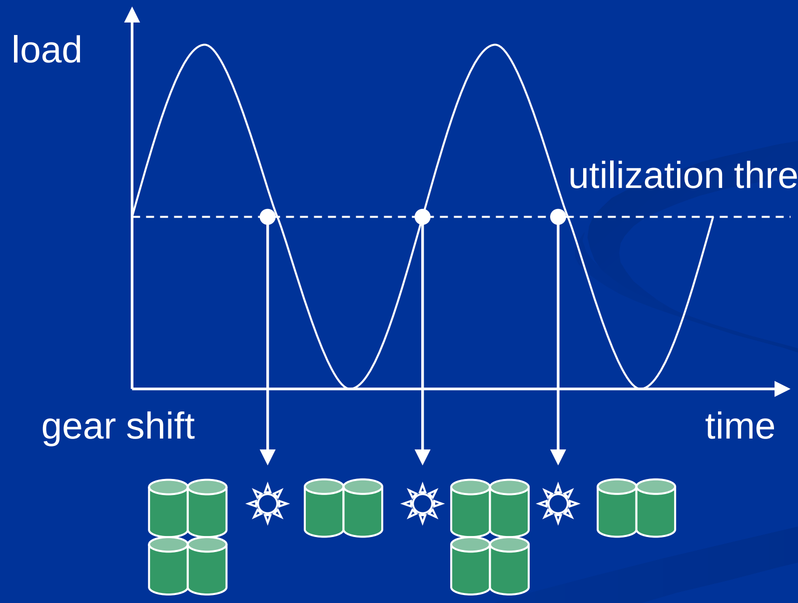
Skewed Striping for Energy Saving

- Approximate the workload
- Gear shift into most appropriate gear
  - Minimize the opportunity lost to save power



Skewed Striping for Energy Saving

- Adapt to cyclic fluctuating workload
- Gear shift when gear utilization threshold is met



Preserving Peak Performance

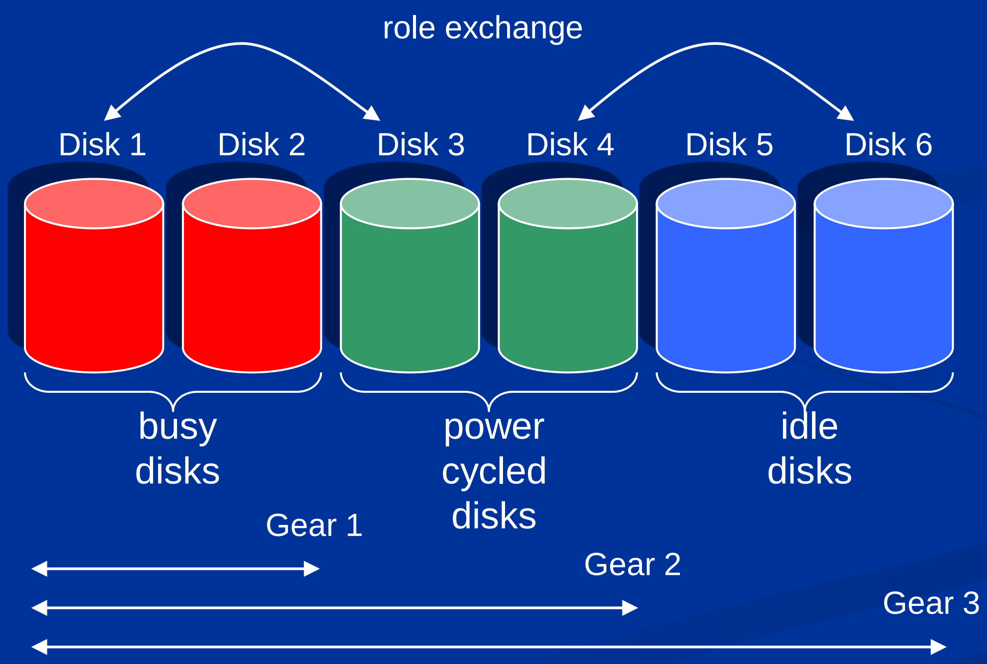
- Operate in the highest gear
  - When the system demands peak performance
  - Uses the same disk layout
- Maximize parallelism within each gear
  - Load is balanced
  - Uniform striping pattern
- Delay block replication until gear shifts
  - Capture block writes

Maintaining Reliability

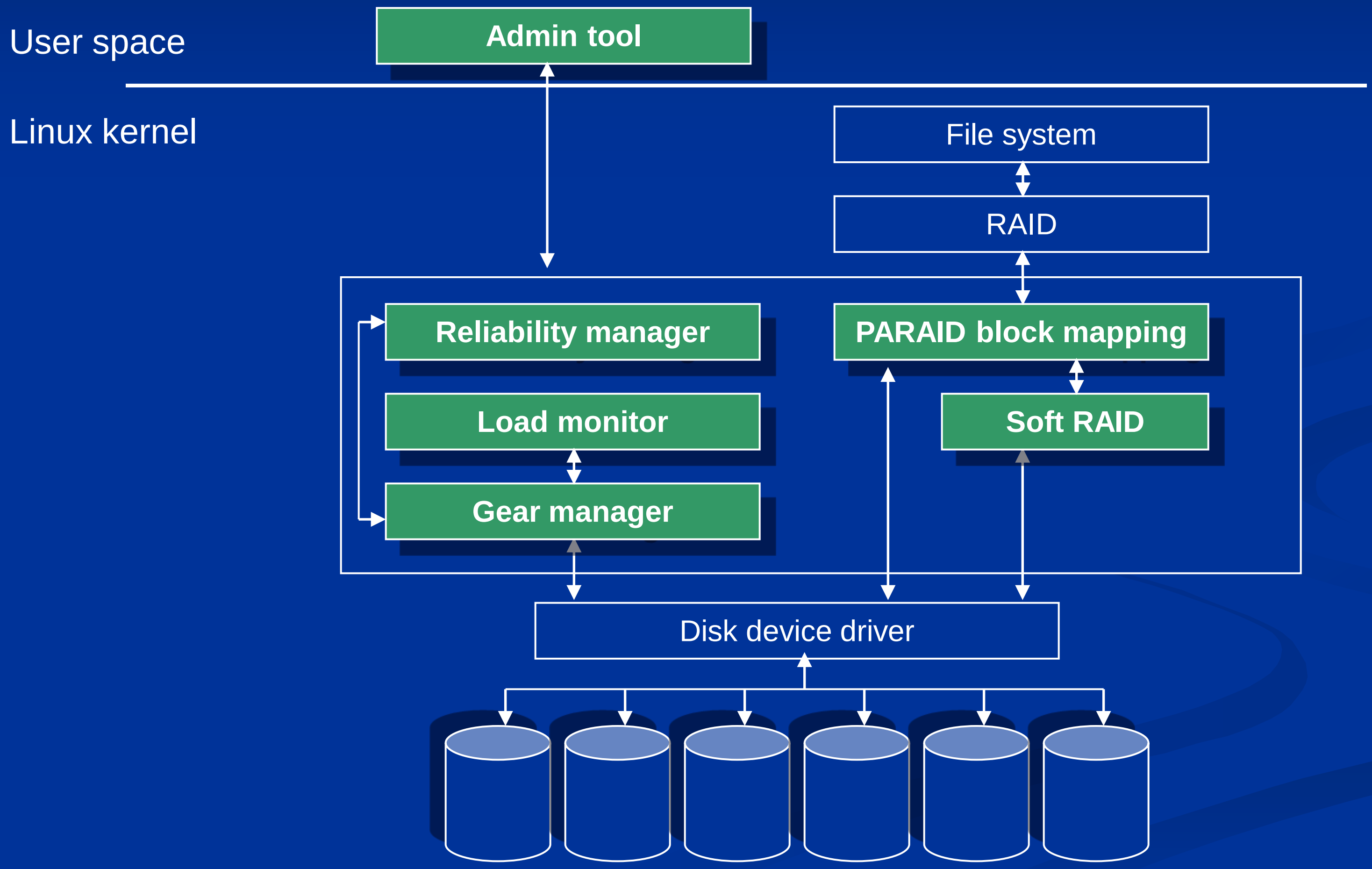
- Reuse existing RAID levels (RAID-5)
  - Also used in various gears
- Drives have a limited number of power cycles
  - Ration number of power cycles

Maintaining Reliability

- Busy disk stay powered on, idle disks stay powered off
- Outside disks are role exchanged with middle disks



Logical Component Design



Data Layout

- Resembles the data flow of RAID 1+0
- Parity for 5 disks does not work for 4 disks
  - For example, replicated block 12 on disk 3

	Disk 1	Disk 2	Disk 3	Disk 4	Disk 5
Gear 1 RAID-5	(1-4)	8	12	((1-4),8,12)	
	16	20	(16,20,_)	_	
Gear 2 RAID-5	1	2	3	4	(1-4)
	5	6	7	(5-8)	8
	9	10	(9-12)	11	12
	13	(13-16)	14	15	16
	(17-20)	17	18	19	20

Data Layout

- Cascading parity updates
  - For example, updating block 8 on disk 5

	Disk 1	Disk 2	Disk 3	Disk 4	Disk 5
Gear 1 RAID-5	(1-4)	8	12	((1-4),8,12)	
	16	20	(16,20,_)	_	
Gear 2 RAID-5	1	2	3	4	(1-4)
	5	6	7	(5-8)	8
	9	10	(9-12)	11	12
	13	(13-16)	14	15	16
	(17-20)	17	18	19	20

Update Propagation

- Up-shift propagation (e.g. shifting from 3 to 5 disks)
  - Full synchronization
  - On-demand synchronization
    - Need to respect block dependency
- Downshift propagation
  - Full synchronization

Asymmetric Gear-Shifting Policies

- Up-shift (aggressive)
  - Moving utilization average + moving standard deviation > utilization threshold
- Downshift (conservative)
  - Modified utilization moving average + moving standard deviation < utilization threshold
  - Moving average modified to account for fewer drives and extra parity updates

Implementation

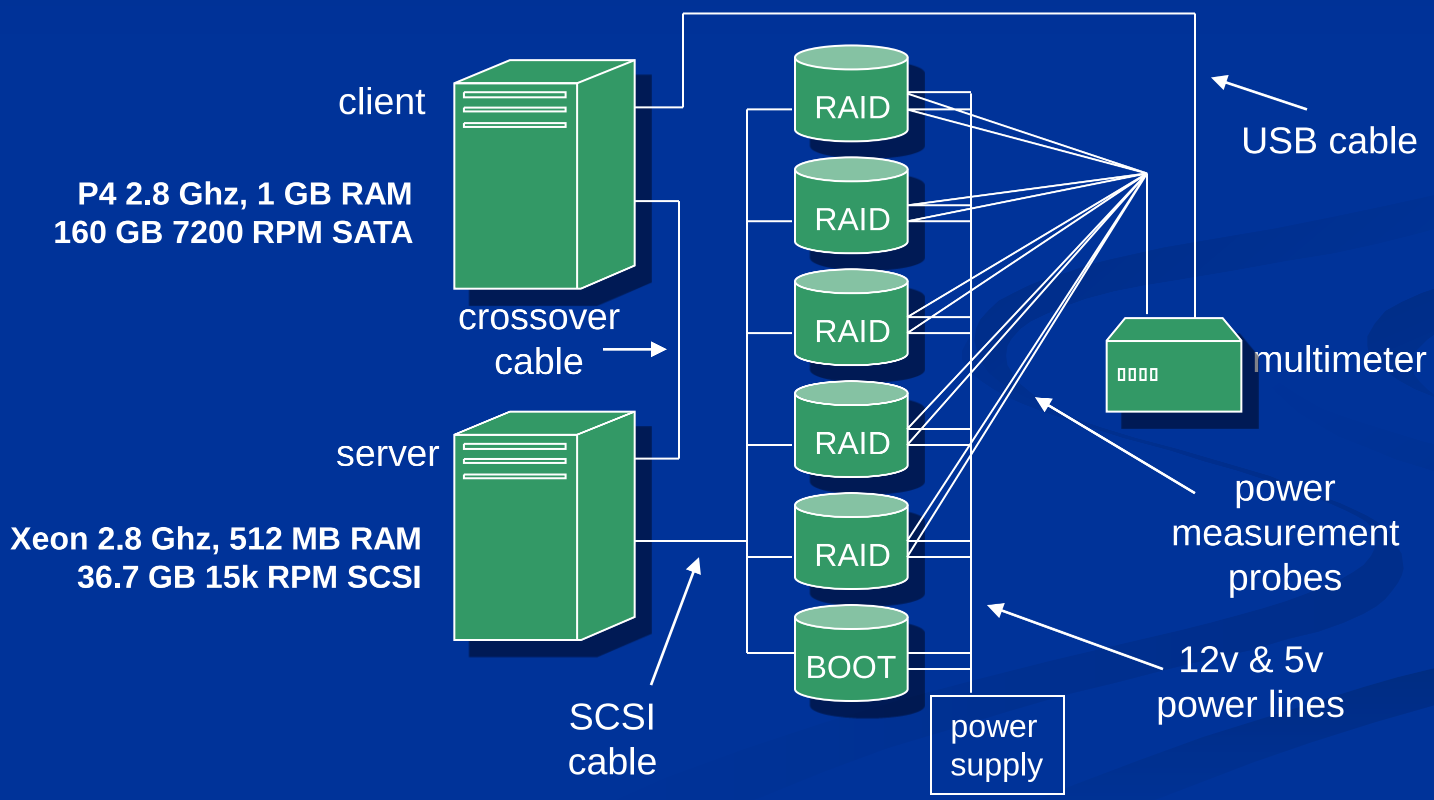
- Prototyped in Linux 2.6.5
  - Open source, software RAID
- Implemented block I/O handler, monitor, disk manager
- Implemented user admin tool to configure device
- Updated Raid Tools to recognize PARAID level

Evaluation

- Challenges
  - Prototyping PARAID
  - Commercial machines
  - Conceptual barriers
  - Benchmarks designed to measure peak performance
  - Trace replay
  - Time consuming

Evaluation

- Measurement framework



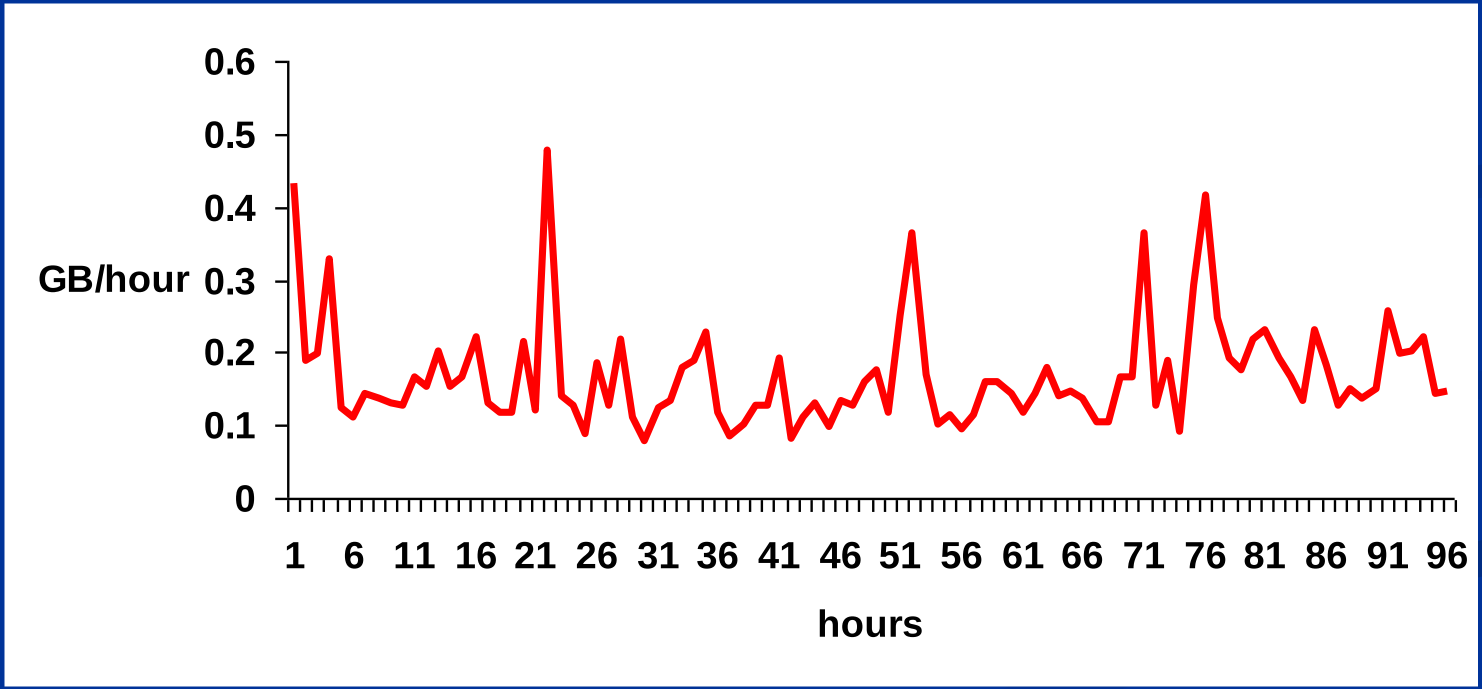
Evaluation

- Three different workloads using two different RAID settings
  - Web trace - RAID level 0 (2-disk gear 1, 5-disk gear 2)
    - Mostly read activity
  - Cello99 - RAID level 5 (3-disk gear 1, 5-disk gear 2)
    - I/O-intensive workload with writes
  - PostMark - RAID level 5
    - Measure peak performance and gear shifting overhead

- Speed up trace playback
  - To match hardware
  - Explore range of speed up factors and power savings

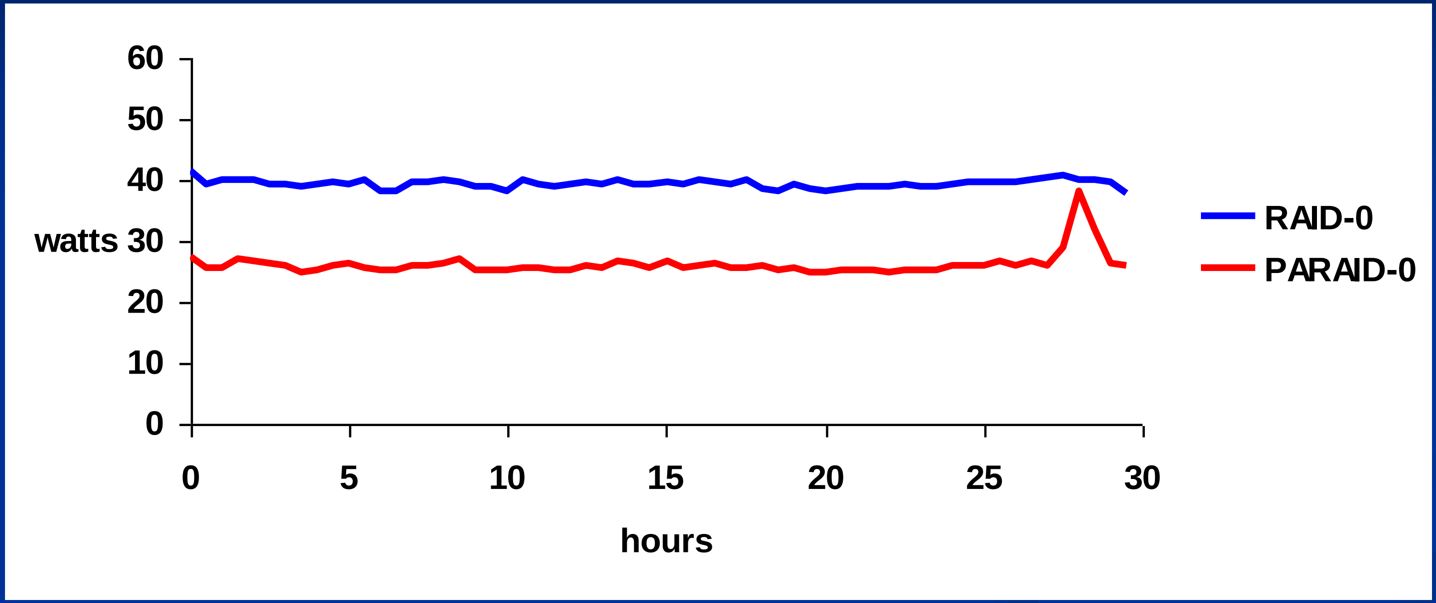
Web Trace

- UCLA CS Dept Web Servers (8/11/2006 – 8/14/2006)
- File system: ~32 GB (~500k files)
- Trace replay: ~95k requests with ~4 GB data (~260 MB unique)



Web Trace Power Savings

64x – 60 requests/sec



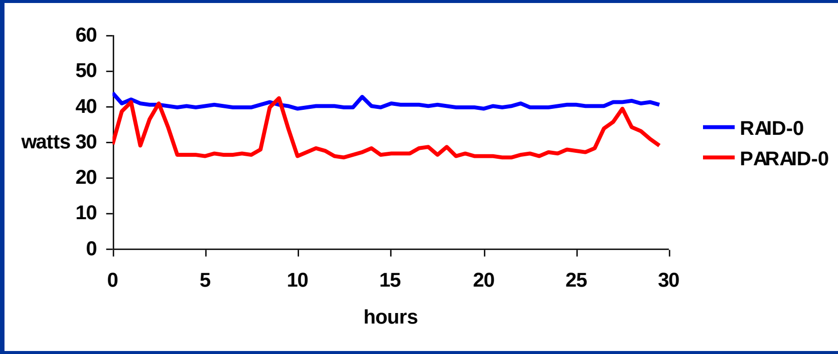
Energy Savings

64x - 34%

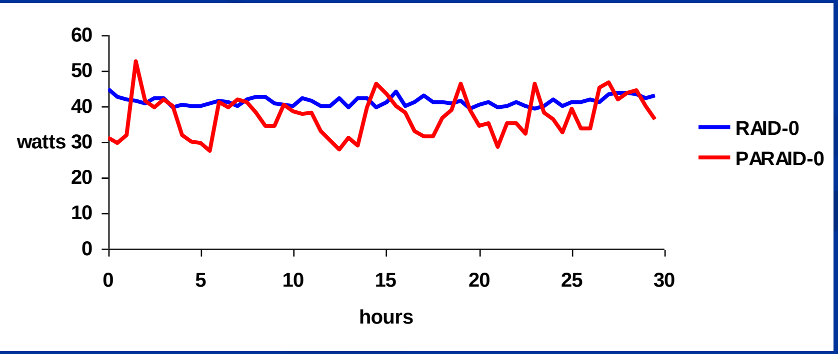
128x - 28%

256x - 10%

128x – 120 requests/sec

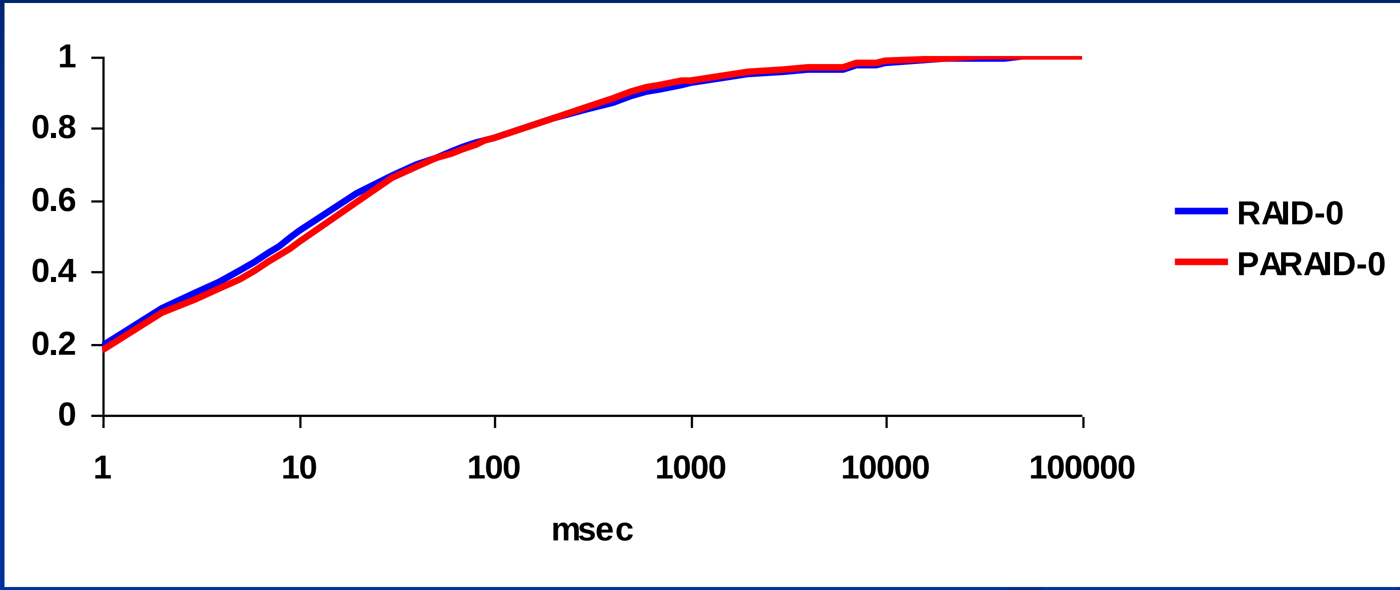


256x – 240 requests/sec



Web Trace Latency

256x

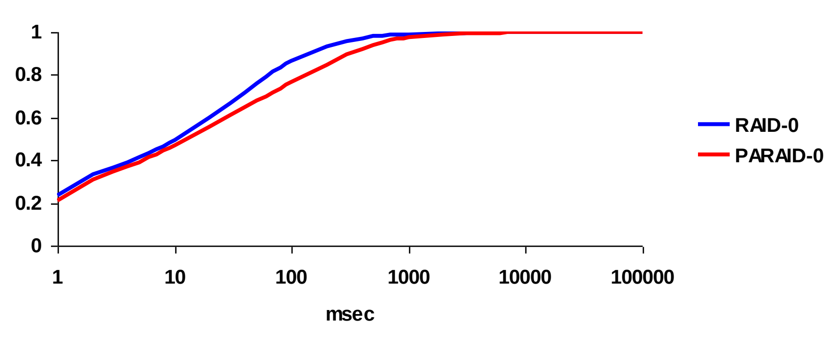


Overhead

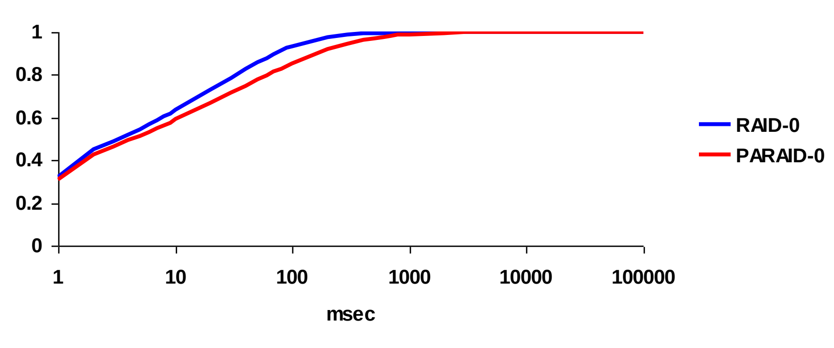
256x - within 2.7%

64x - 240%
80ms vs. 33ms

128x

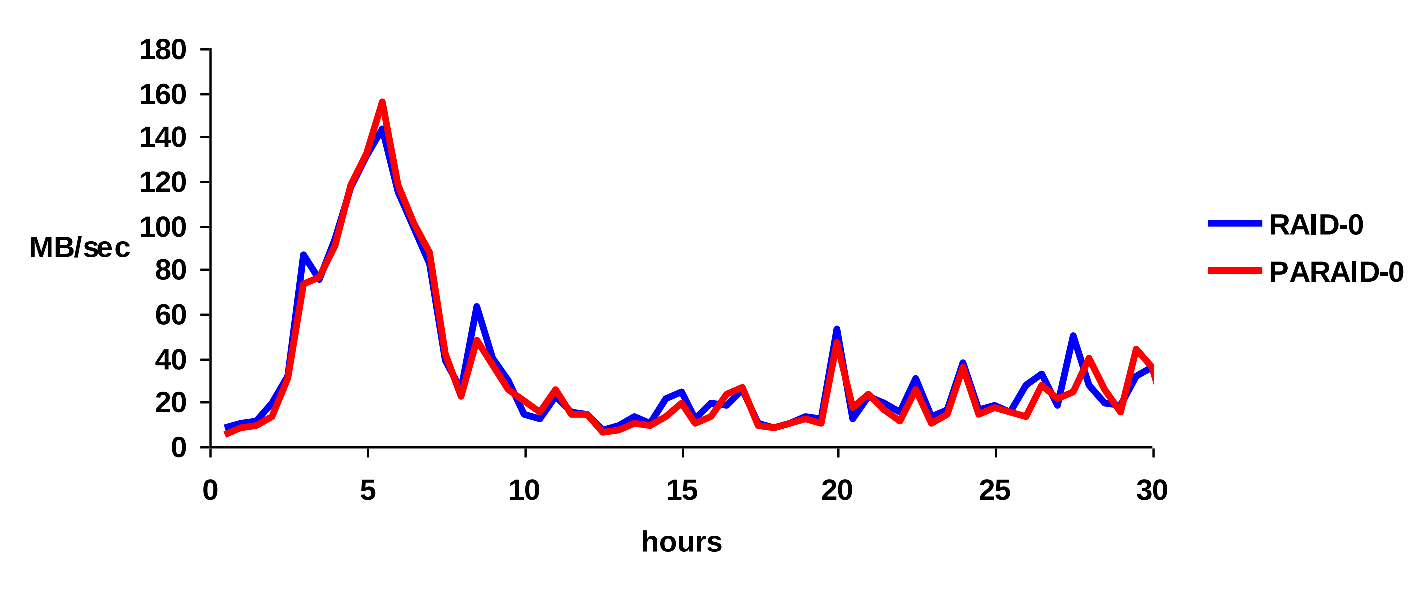


64x



Web Trace Bandwidth

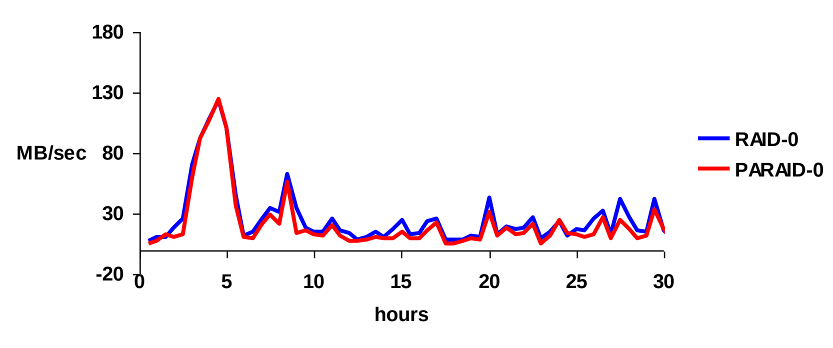
256x



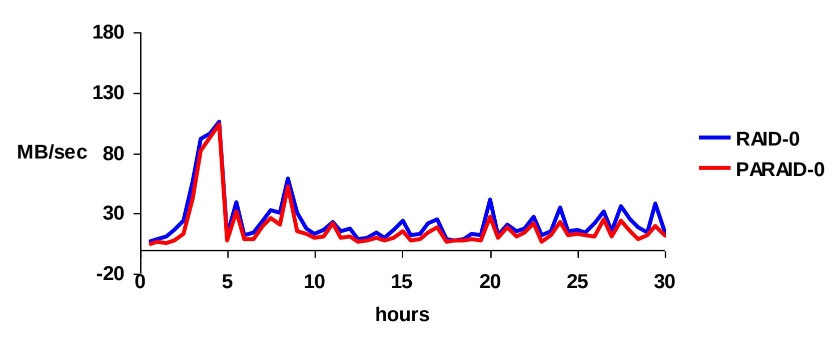
Overhead

256x - within 1.3% in high gear

128x



64x

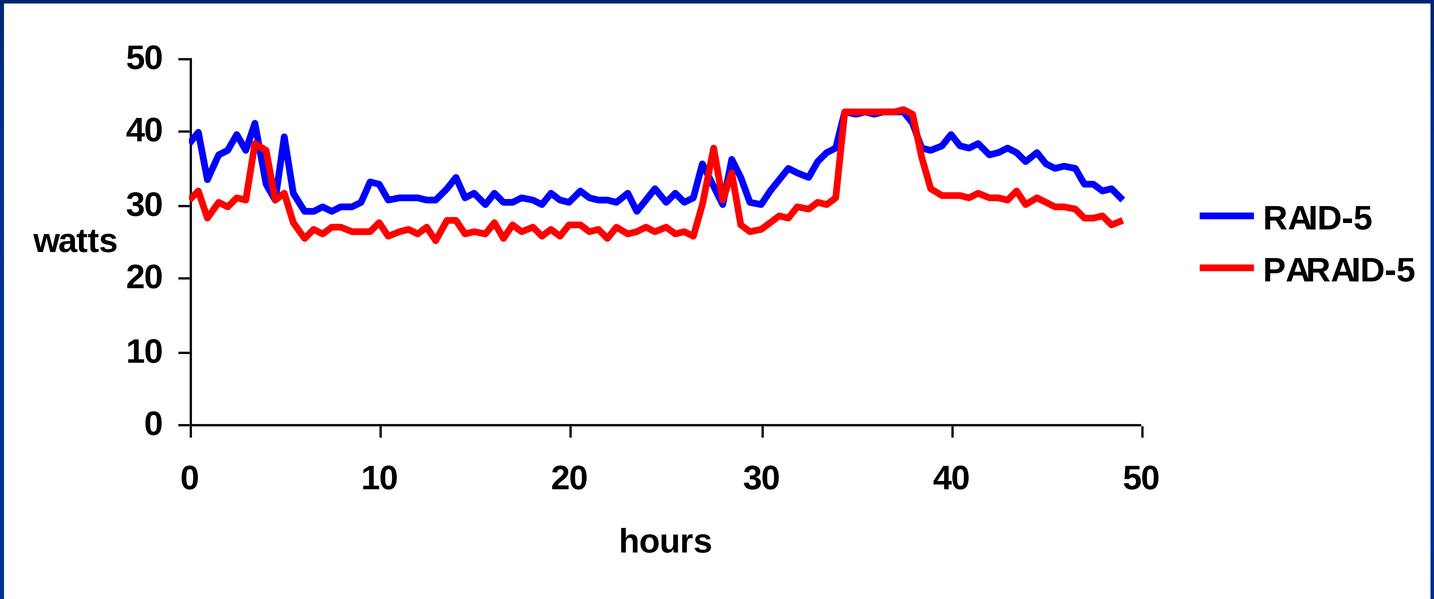


Cello99 Trace

- Cello99 Workload
  - HP Storage Research Labs
  - 50 hours beginning on 9/12/1999
  - 1.5 million requests (12 GB) to 440MB of unique blocks
  - I/O-intensive with 42% writes

Cello99 Power Savings

32x – 270 requests/sec



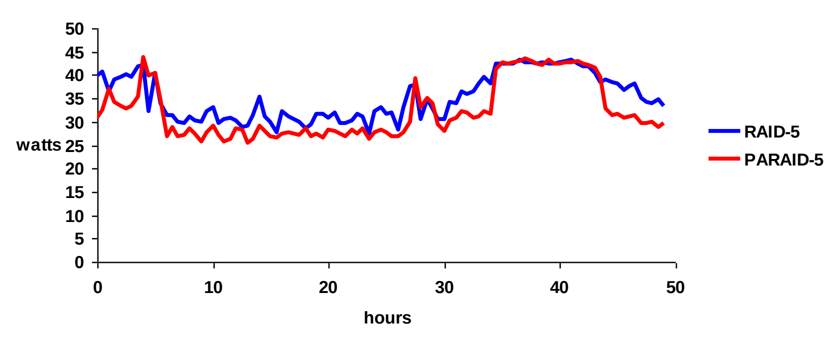
Energy Savings

32x - 13%

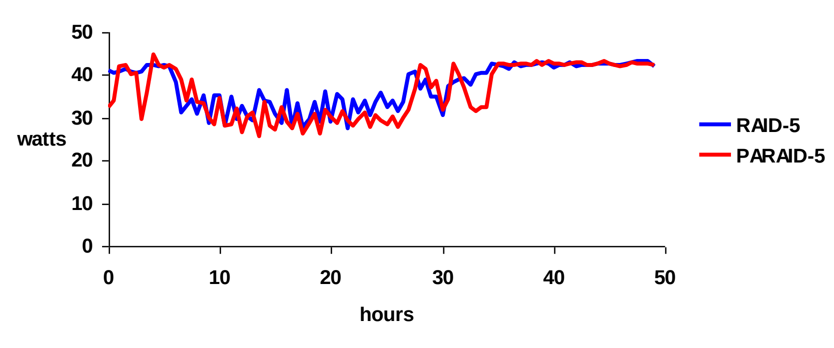
64x - 8.2%

128x - 3.5%

64x – 550 requests/sec

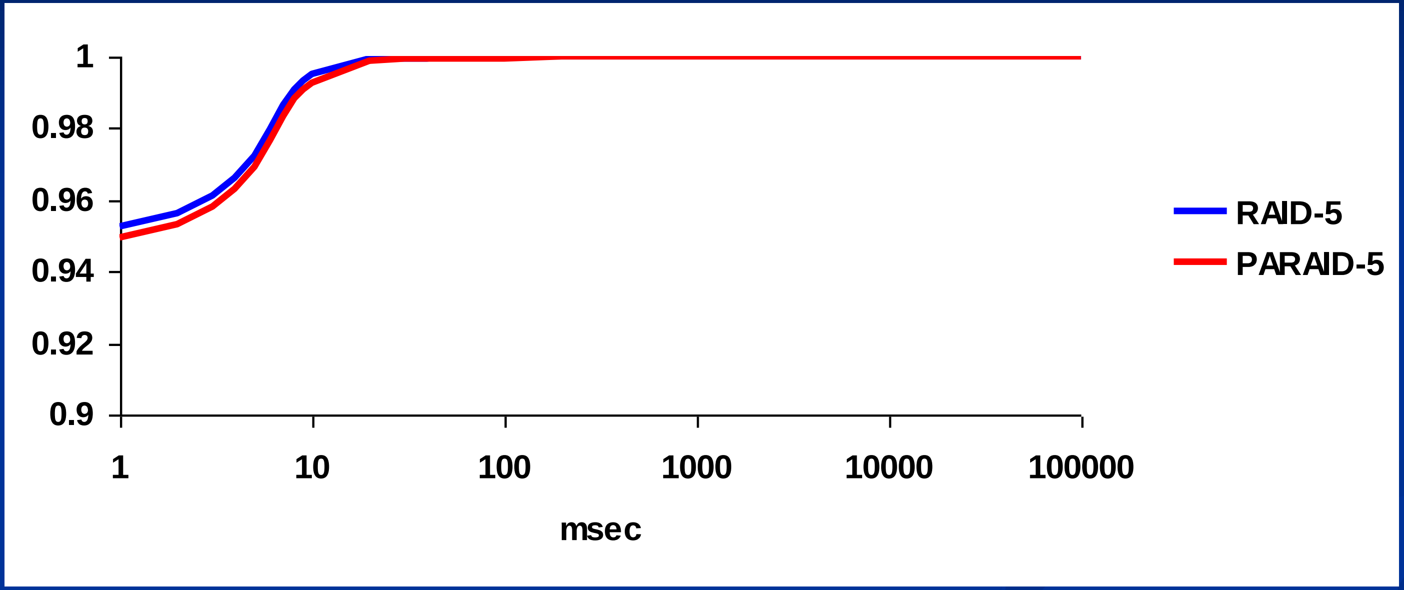


128x – 1000 requests/sec



Cello99 Completion Time

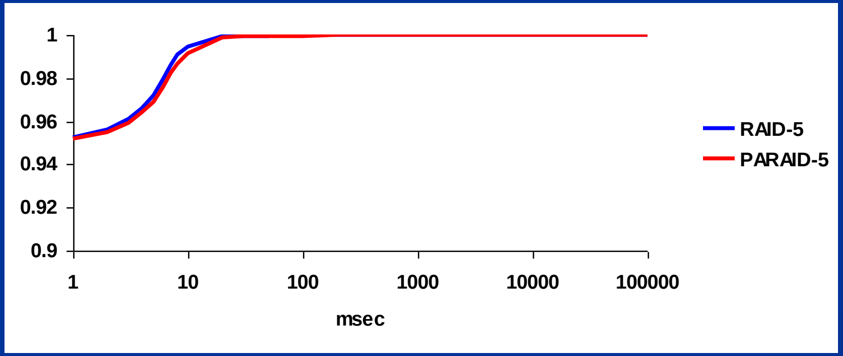
128x



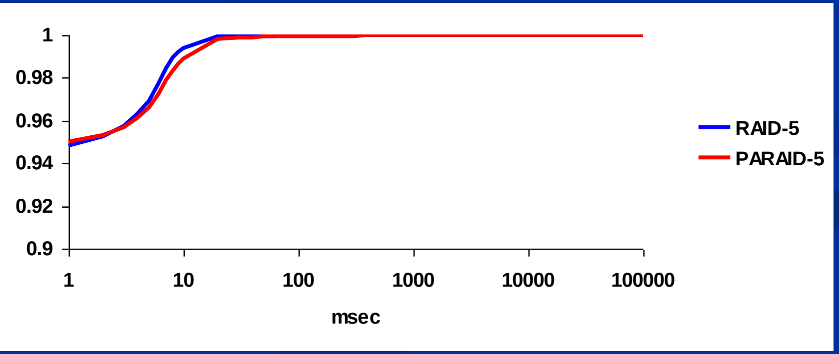
Overhead

32x - 1.8ms, 26% slower due to time spent in low gear

64x

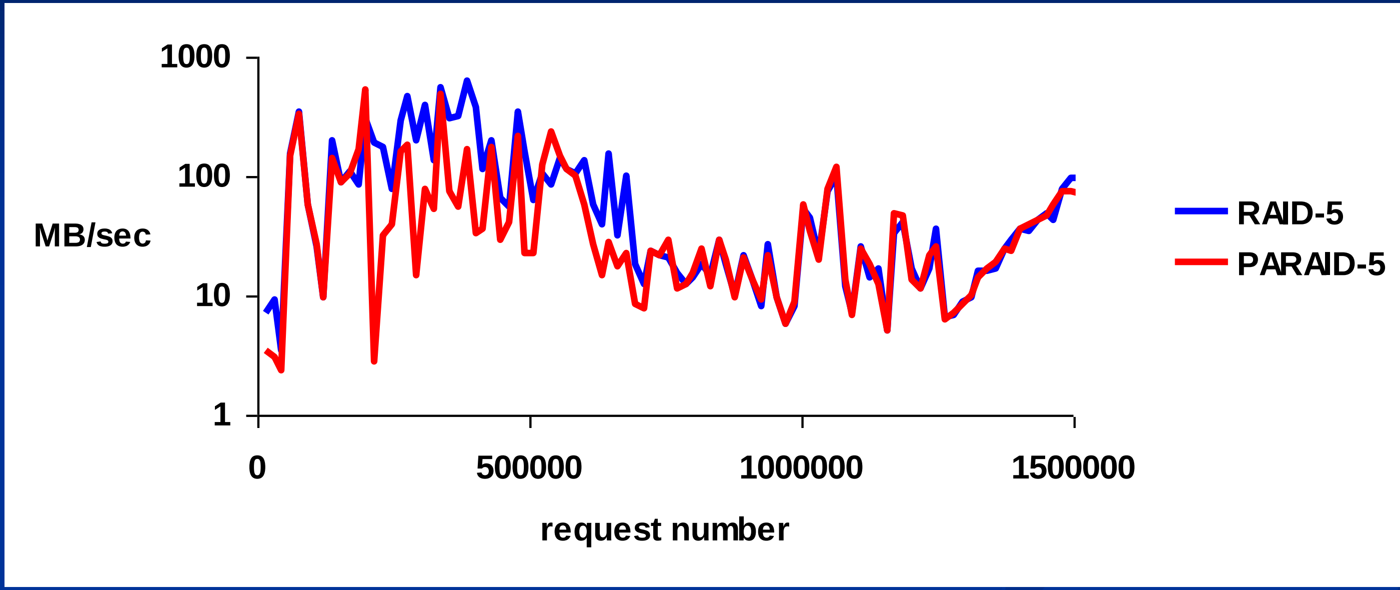


32x



Cello99 Bandwidth

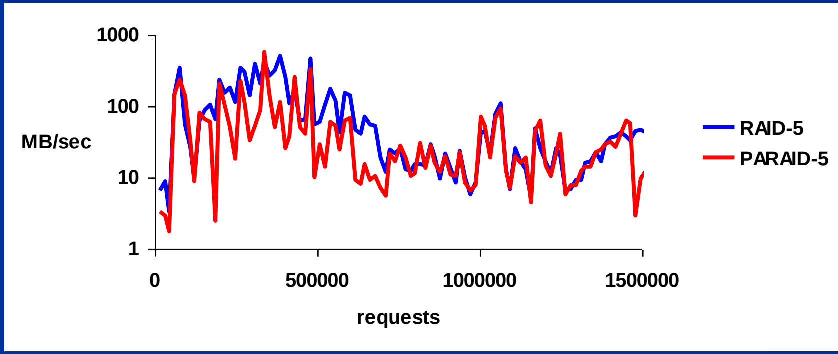
128x



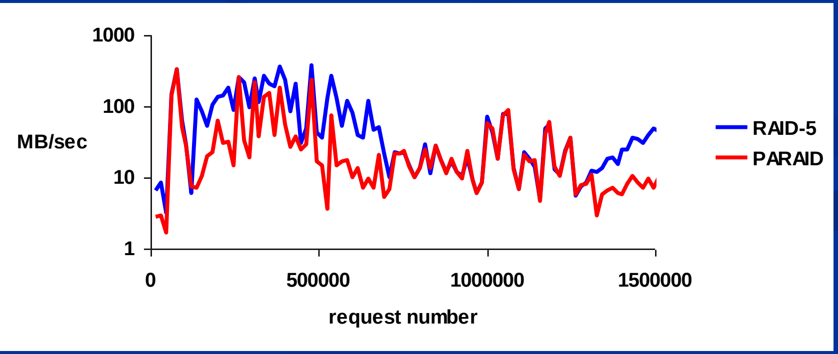
Overhead

< 1% degradation during peak hours

64x



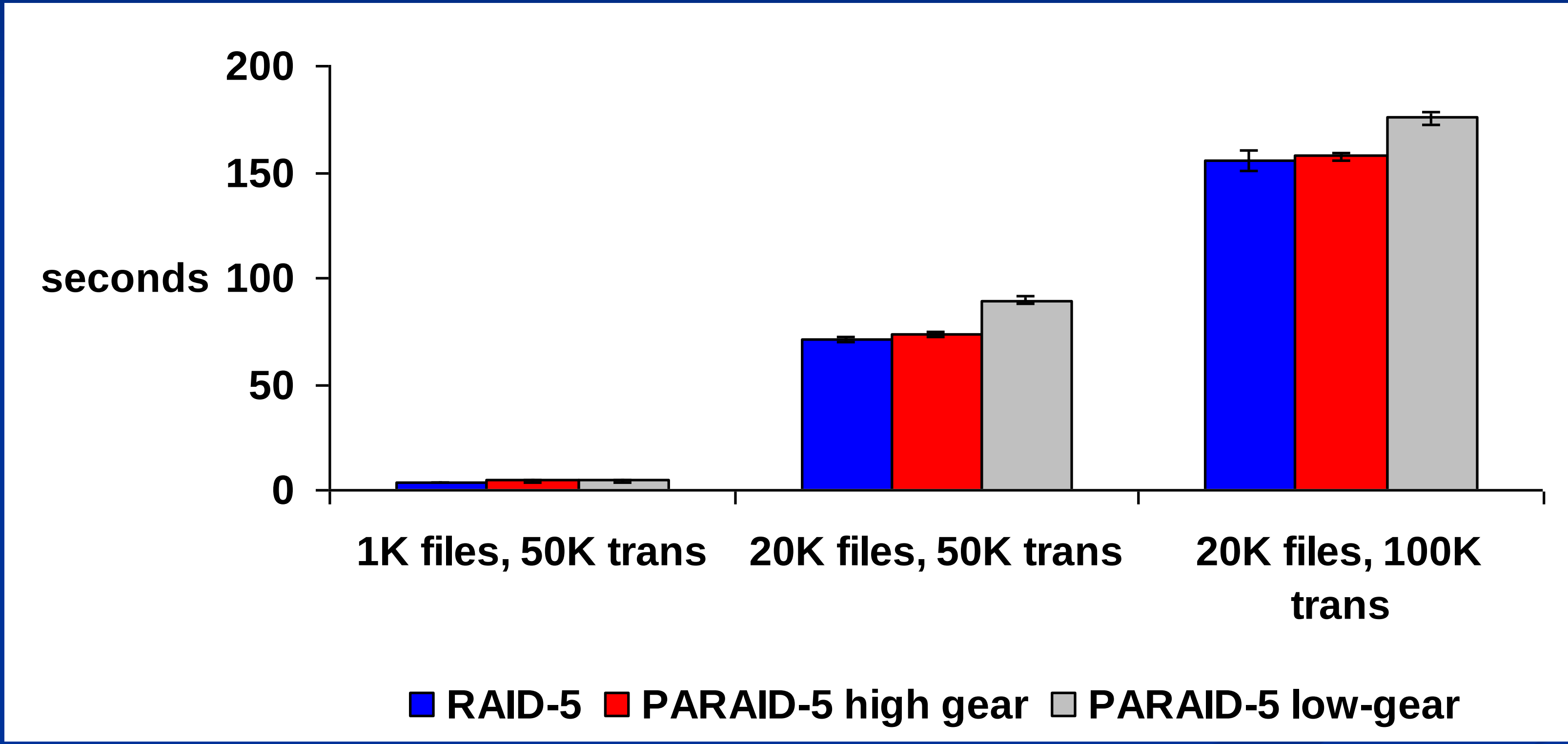
32x



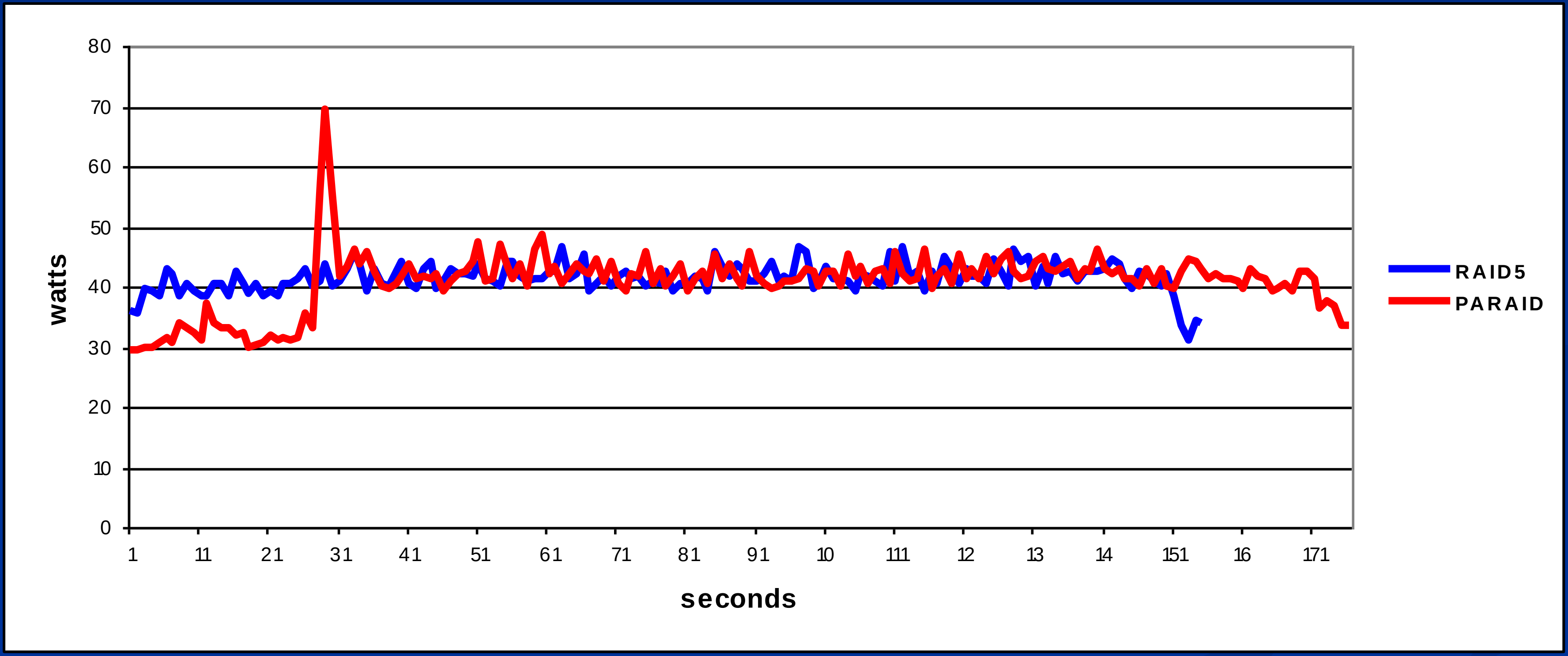
PostMark Benchmark

- Popular synthetic benchmark
- Generates ISP-style workloads
- Stresses peak read/write performance of storage device

Postmark Performance



Postmark Power Savings



Related Work

- Pergamum
- EERAID
- RIMAC
- Hibernator
- MAID
- PDC
- BlueFS

Future Work

- Try more workloads
- Optimize PARAID gear configuration
- Explore asynchronous update propagation
- Speed up recovery
- Live testing

Lessons Learned

- Third version of design, early design too complicated
- Data alignment problems
- Difficult to measure system under normal load
- Hard to predict workload transformations due to complex system optimizations
- Challenging to match trace environments

Conclusion

- PARAID reuses standard RAID-levels without special hardware while decreasing their energy use by 34%.
  - Optimized version can save even more energy
- Empirical evaluation important

Research Theme

- Data flow management
  - Storage
  - MANETs
- Current state
  - Reminiscent of plumbing industry 200 years ago
    - Limited interchangeable parts
    - Poorly understood interactions

Research Areas

- Power-Aware RAID
- Electric-field-based routing for MANETs
- Conquest disk-persistent-RAM hybrid file system
- Optimistic replication
- Real-time systems

Questions

PARAID: A *Gear-Shifting*

Power-Aware RAID

- Contact
  - Andy Wang – awang@cs.fsu.edu
- http://www.cs.fsu.edu/~awang/conquest-2

PARAID Recovery

- 2.7 times slower than conventional raid
  - For example, 2 gear PARAID device
    - First, the soft state must recover
    - Second, data must be propagated
    - Third, conventional raid must recover
- Recovery not as bad for read intensive workloads

PARAID Gear-Shifting

Web Trace Gear-Shifting Stats

	256x	128x	64x
Number of gear switches	15.2	8.0	2.0
% time spent in low gear	52%	88%	98%
% extra I/Os for update propagations	0.63%	0.37%	0.21%

Cello99 Gear-Shifting Stats

	128x	64x	32x
Number of gear switches	6.0	5.6	5.4
% time spent in low gear	47%	74%	88%
% extra I/Os for update propagations	8.0%	15%	21%