



# Modeling the Performance and Energy of Storage Arrays

*Sankaran Sivathanu(Georgia Tech), Ling Liu(Georgia  
Tech) and Cristian Ungureanu(NEC Labs)*

**The First IEEE International Conference on  
Green Computing, 2010.**



**Georgia Institute  
of Technology®**

**NEC**

# Outline

- **Introduction**
- Motivation
- Modeling RAID
- PARAID
  - Overview
  - Modeling PARAID
- E-PARAID
  - Design
  - Modeling E-PARAID
  - Evaluation
- Model Validation
- Conclusion

# Notion of Storage system power

- Increased focus on power consumption in data centers
  - Power bills contribute to millions of dollars per year in large data centers
  - Storage equipments account for 30-50% of data center power consumption
  - Hard disks need most of that power for keeping itself spinning
- Power conservation
  - Overall goal : Minimize disk spinning time and/or minimize total number of spinning platters
  - Leverage spatial and temporal locality of data
- Effects
  - Often reduces performance, reliability and availability of data

# Power vs. Performance

- Using lesser number of platters
  - Probability of a disk failure leading to a data failure is more
  - Parallelism in data access is reduced
- Leveraging temporal locality to spin disks down
  - Spinning a disk up again often takes more than 10 seconds!
  - Requests to a shut disk suffer very high latency
- Power conservation techniques give raise to background jobs
  - Dynamically moving data around to optimize for least platters
  - Background jobs affect foreground requests

# Outline

- Introduction
- **Motivation**
- Modeling RAID
- PARaid
  - Overview
  - Modeling PARaid
- E-PARaid
  - Design
  - Modeling E-PARaid
  - Evaluation
- Model Validation
- Conclusion

# Storage performance models

- Existing models accurately predict performance of RAID based storage systems
  - Includes most of the storage components like memory cache, disk cache, controller delays, etc.,
  - Models recovery and rebuild costs of RAID arrays
- Still opaque to energy efficient storage systems
  - Doesn't account for many complex interactions brought by energy conservation algorithms
- Considers synthetic workloads with inaccurate assumptions

# Our Approach

- We model performance of energy-aware RAID systems
  - Effects of power optimization on performance is accurately captured
  - Computes bandwidth & latency of a storage array that is controlled by an energy algorithm
- Our model works off from raw block-level traces
  - No assumptions on workload patterns
  - A few key parameters from the trace are extracted and fed to our model

# Overview of our model

- Parameters considered
  - *Run-length* and *Run-count* (captures degree of sequentiality in workload)
  - Seek time as a function of seek distance
  - Total data transferred in every time window
  - Number of disks in the array that actively services at any time window
- Bandwidth = Total bytes transferred/Time Taken ( $et_{\tau}$ )
- For a single disk,

$$et_{\tau} = \sum_{len_{\tau}=1}^{max\_len_{\tau}} runcount[len]_{\tau} \left( \frac{len_{\tau}}{Seq} + sd_{\tau} + rd \right)$$

# Overview of our model

- Latency = Elapsed time/No. of Requests + Queuing Delay
- With Elapsed time, seek time, and rotational delay, power values for each micro-activity are substituted to get overall energy consumed
  - Seek power, rotational power, etc., are obtained from disk specifications

$$E = \sum_{\tau=1}^{max} [\tau \cdot rP + st_{\tau} \cdot (sP - oP) + et_{\tau} \cdot (oP - rP)]$$

- rP : Rotational Power, sP – Seek Power, oP – Operating Power

# Outline

- Introduction
- Motivation
- **Modeling RAID**
- PARaid
  - Overview
  - Modeling PARaid
- E-PARaid
  - Design
  - Modeling E-PARaid
  - Evaluation
- Model Validation
- Conclusion

# Modeling RAID

- Bandwidth computation for the RAID-0 Configuration
  - Every request is striped across  $N_{active}$  disks in the array
- RAID bandwidth is aggregate of all disks, capped by RAID controller bandwidth

$$Bandwidth_{\tau} = \min \left( \frac{N_{active} \times dsize_{\tau}}{et_{\tau}}, max\_ctrl\_bw \right)$$

# Outline

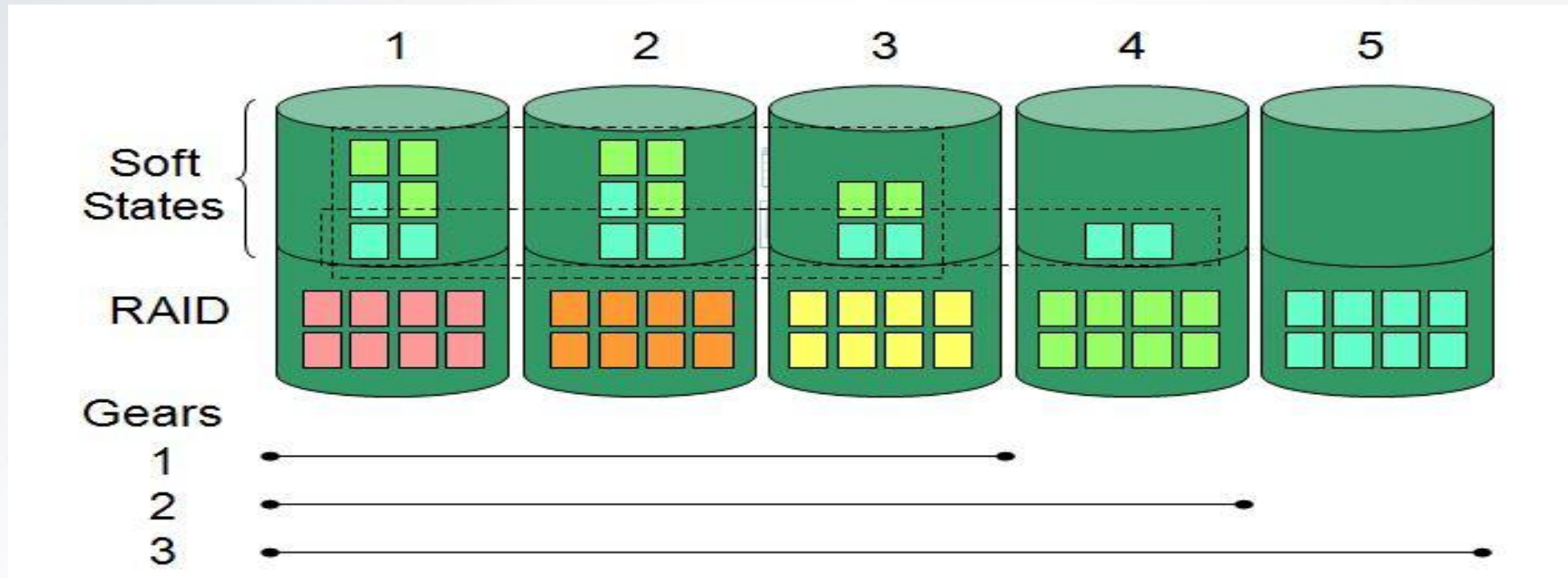
- Introduction
- Motivation
- Modeling RAID performance
- **PARAID**
  - **Overview**
  - **Modeling PARAID**
- E-PARAID
  - Design
  - Modeling E-PARAID
  - Evaluation
- Model Validation
- Conclusion

# Overview

- Leverages cyclic fluctuations in workload
- Dynamically switches between configurations based on load
  - Every configuration has varied number of active disks
- During peak load, serves with all disks
- During reduced load, data is compacted in lesser number of disks
- Degree of redundancy is maintained constant even with lesser number of disks

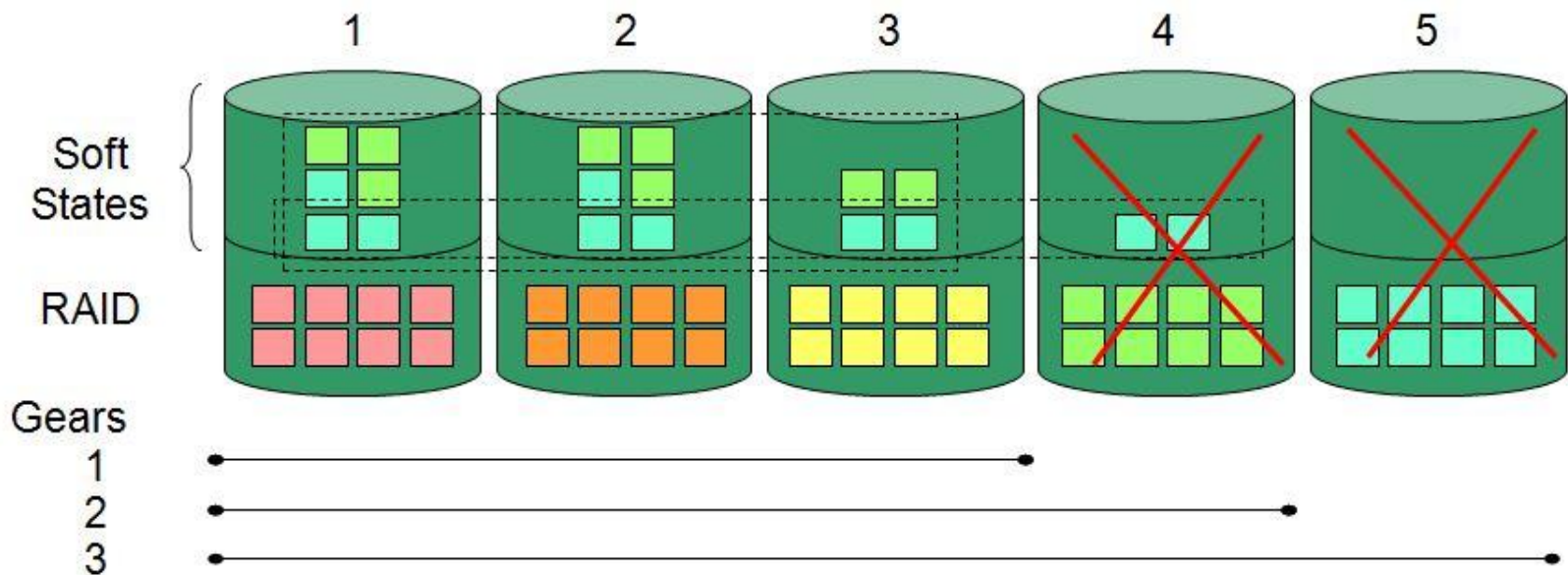
# Design

- Uses over-provisioned spare storage
  - Organized into hierarchical overlapping subsets
  - Each set analogous to gears in automobiles
  - Soft states can be reclaimed for space
  - Persistent across reboots



# Design

- Peak performance is preserved
- Data update propagation is done during gear-shifts
  - Newly-updated data alone are propagated



# Modeling PARaid

- Large run lengths are no more sequential
- Intermittent seeks are to be introduced
- Captured by capping run length  $N_{\text{total}} - m$ , where 'm' is number of shut disks
- Seek time parameter modified to account for extra seeks in low-gear mode
- Data migration modeled tracking new writes to affected disks

$$E_{\text{PARaid}} = \frac{N_{gs}}{2} \cdot n \cdot spE + \sum_{\tau=1}^{max} (N_{\text{active}(\tau)} \cdot E + \tau \cdot n \cdot iP)$$

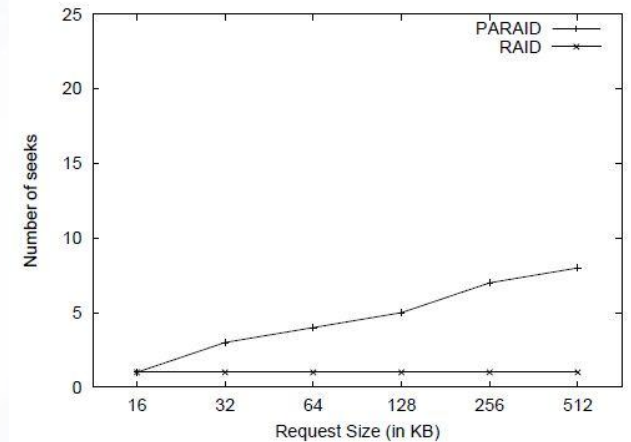
- spE – Spin-up Energy, iP – Idle Power

# Outline

- Introduction
- Motivation
- Modeling RAID performance
- PARaid
  - Overview
  - Modeling PARaid
- **E-PARaid**
  - **Motivation**
  - **Design**
  - **Modeling E-PARaid**
  - **Evaluation**
- Model Validation
- Conclusion

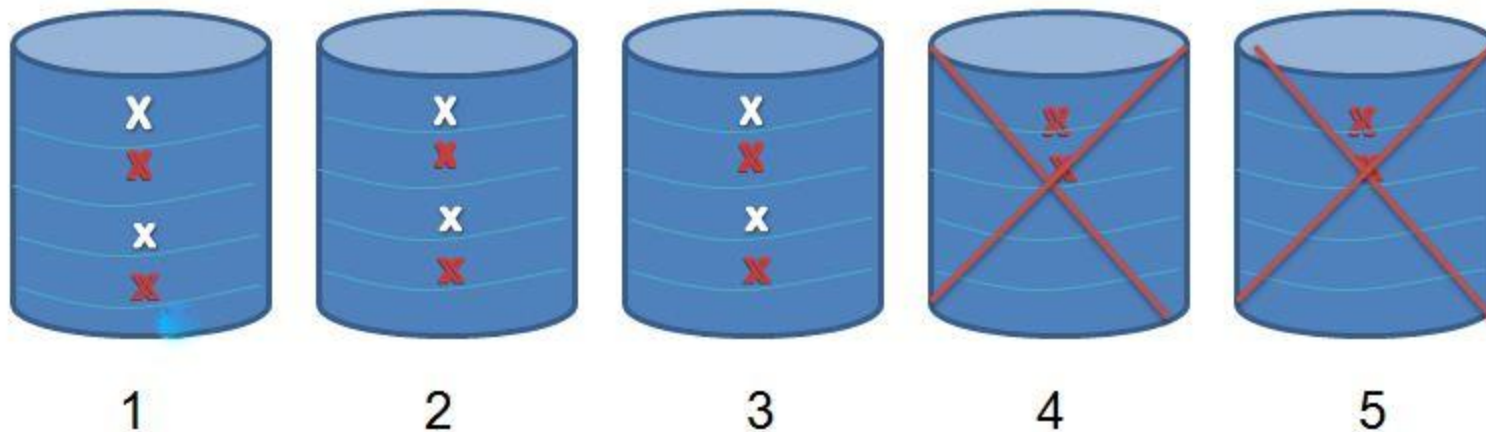
# Motivation

- ‘Soft states’ in PARaid are allocated at remote disk locations
  - Re-mapping logic for low-gear is straight-forward
  - Ease of reclamation of soft states
- Sequential requests are mapped to remote locations
  - Most requests turn random
  - Significant impact on performance during low-gear operation
- **Solution** : Interleaved soft states

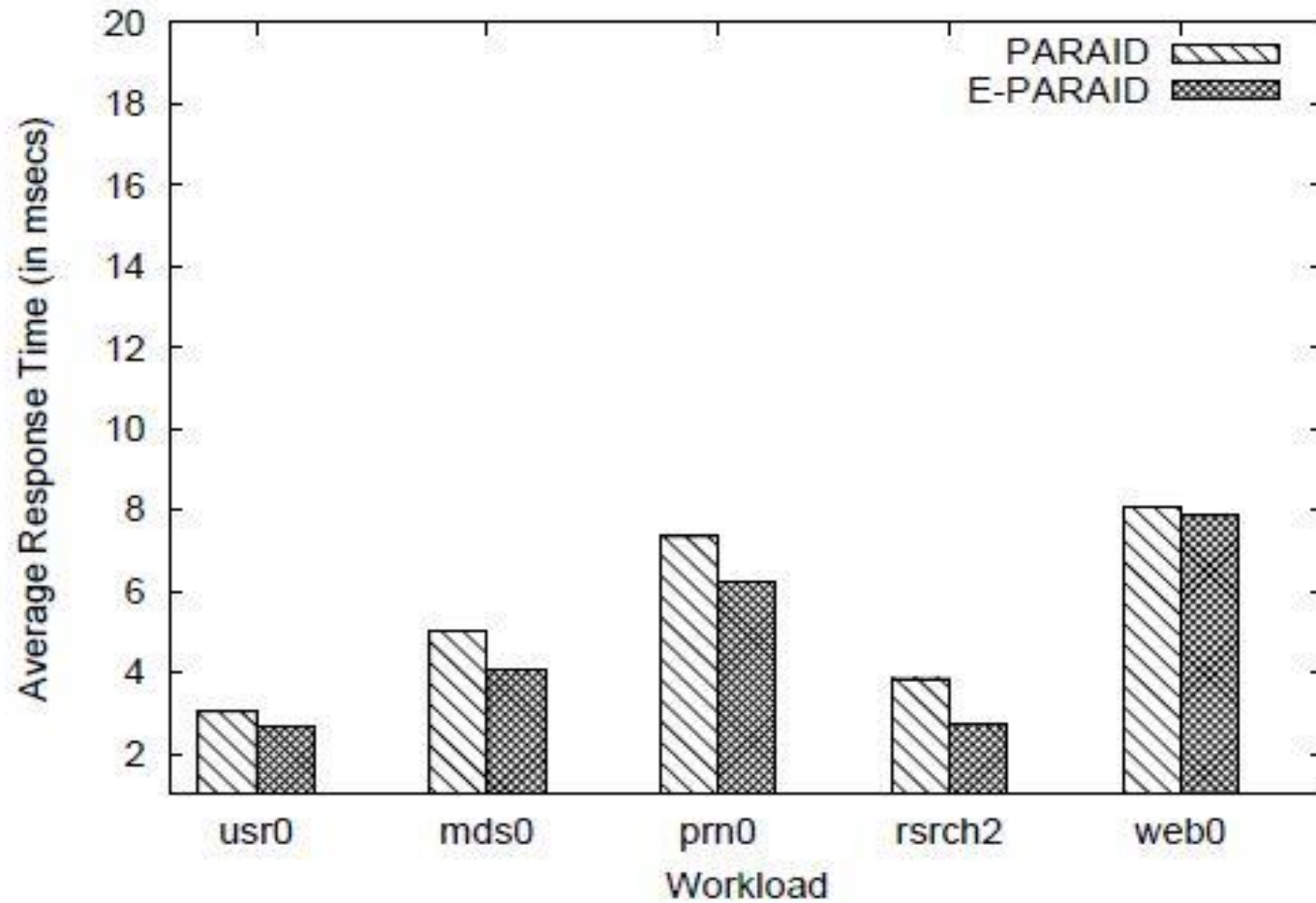


# Design

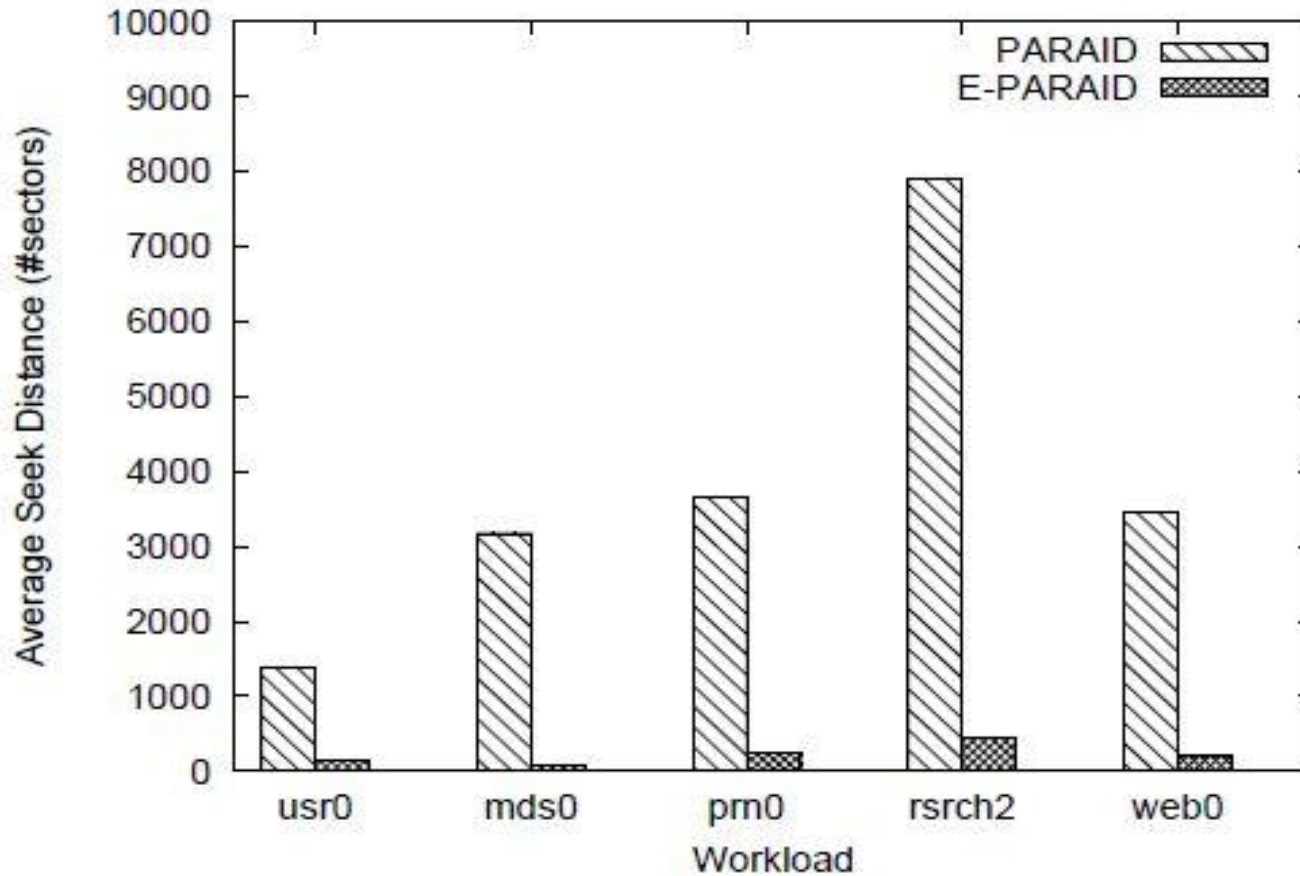
- ‘Buckets’ are smaller chunks of free spaces placed all over the disk
  - Eg: if 2 out of 5 disks are to be shutdown, the entire footprint of the 2 disks are spread over remaining 3 disks.
  - Buckets add-up to entire footprint of shut disks
- Sequential requests are still closer in low-gear operation (often in the same extents)
- Distance between two buckets should be decided strategically



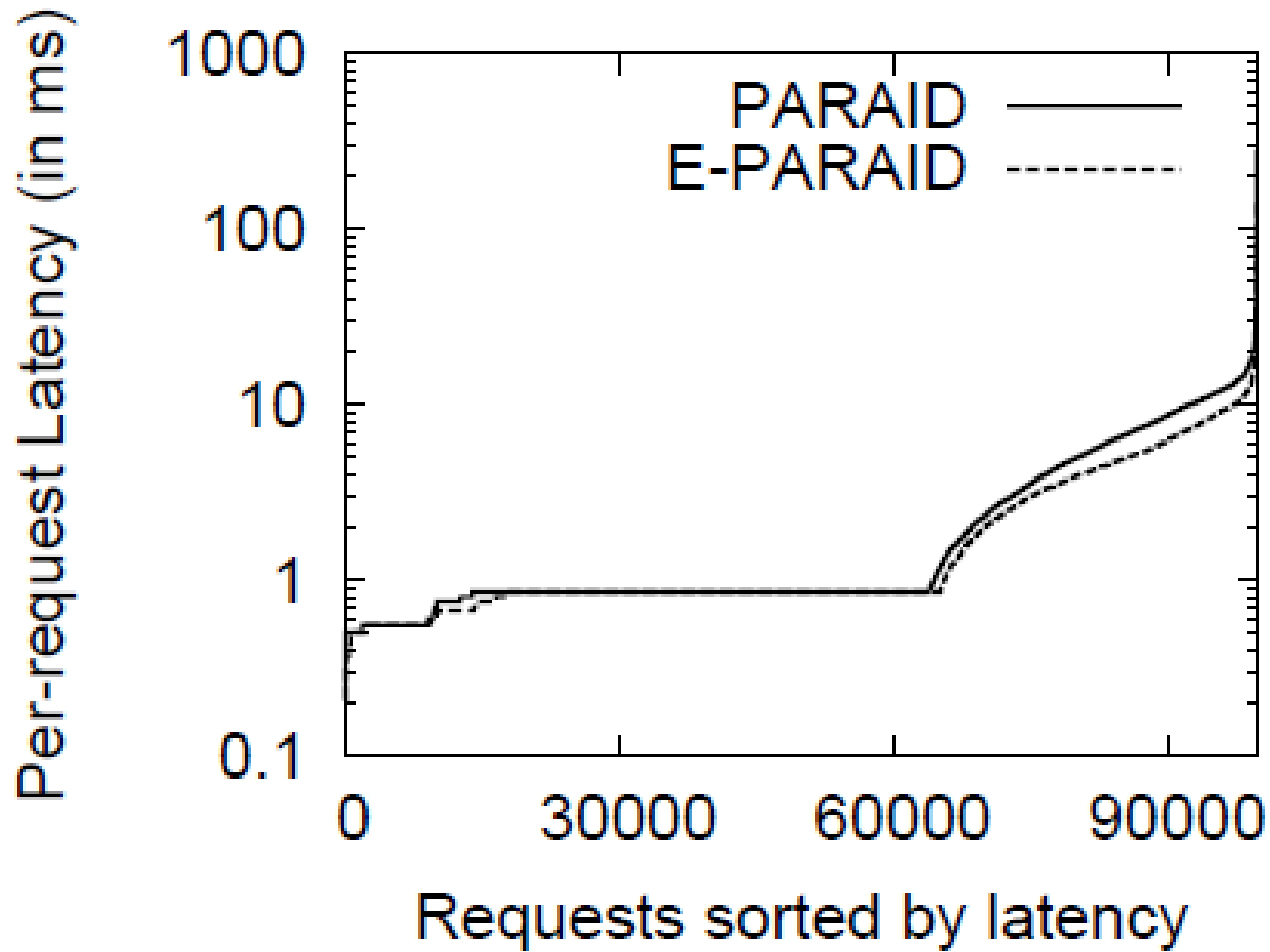
# Evaluation



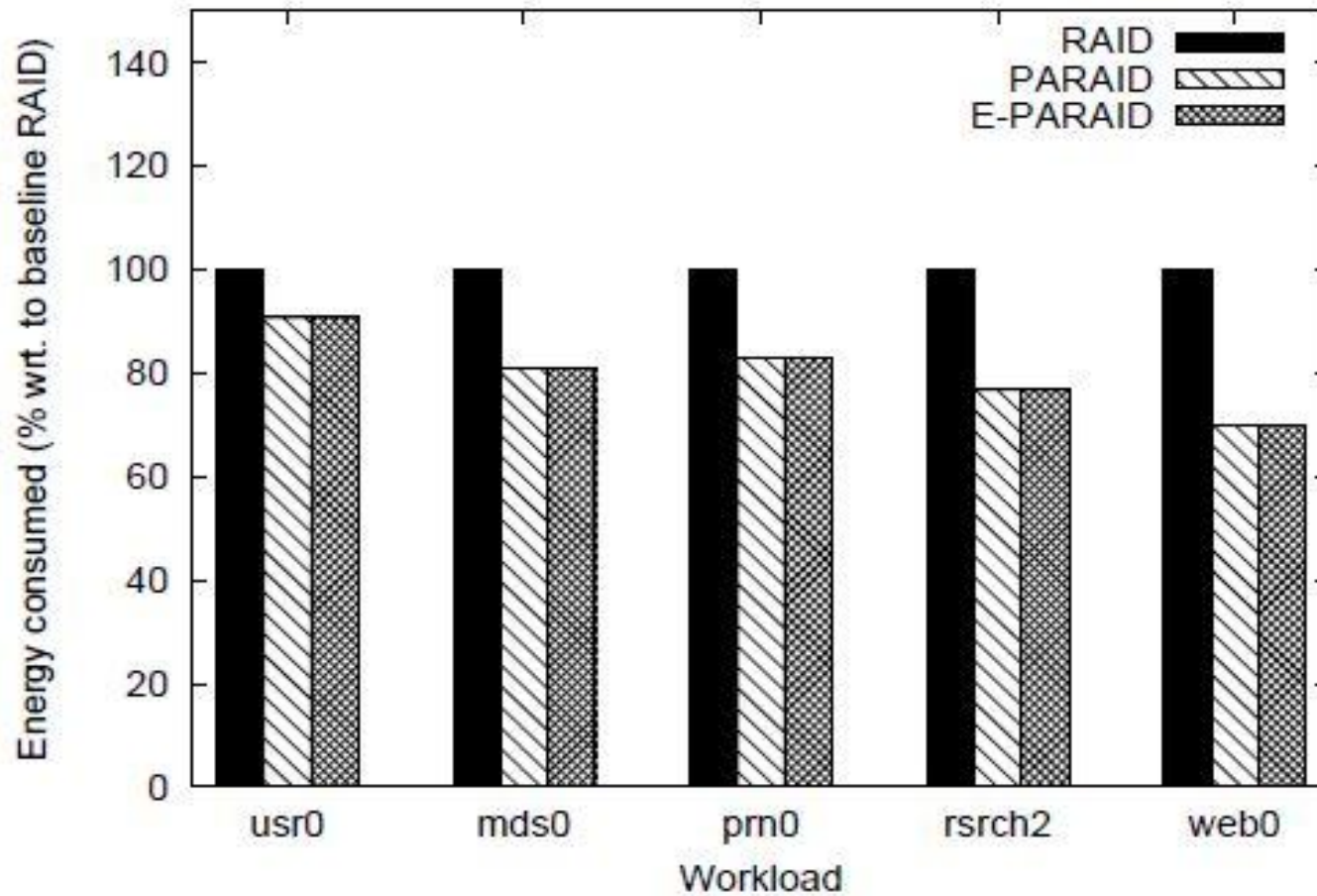
# Evaluation



# Evaluation



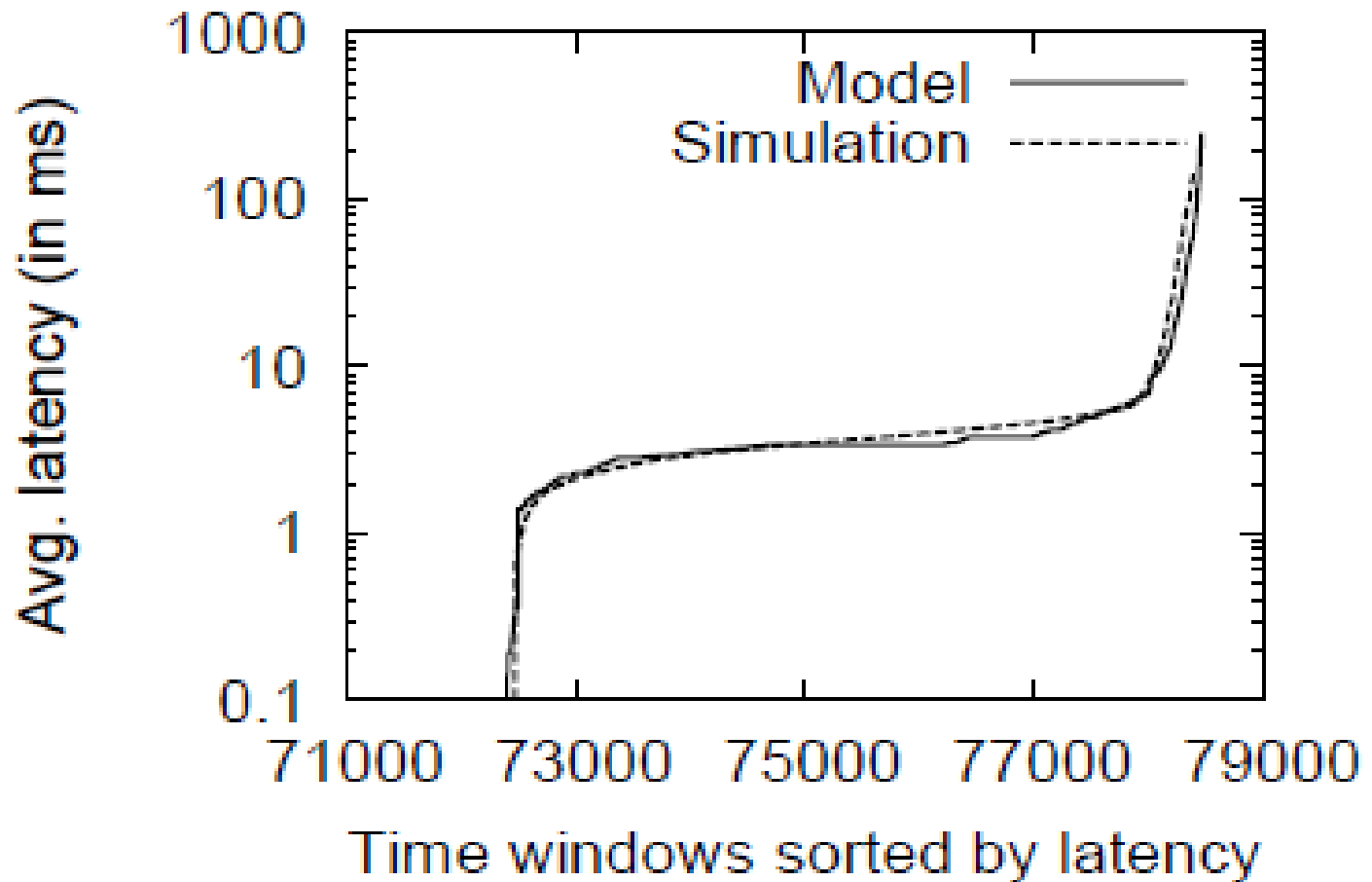
# Evaluation



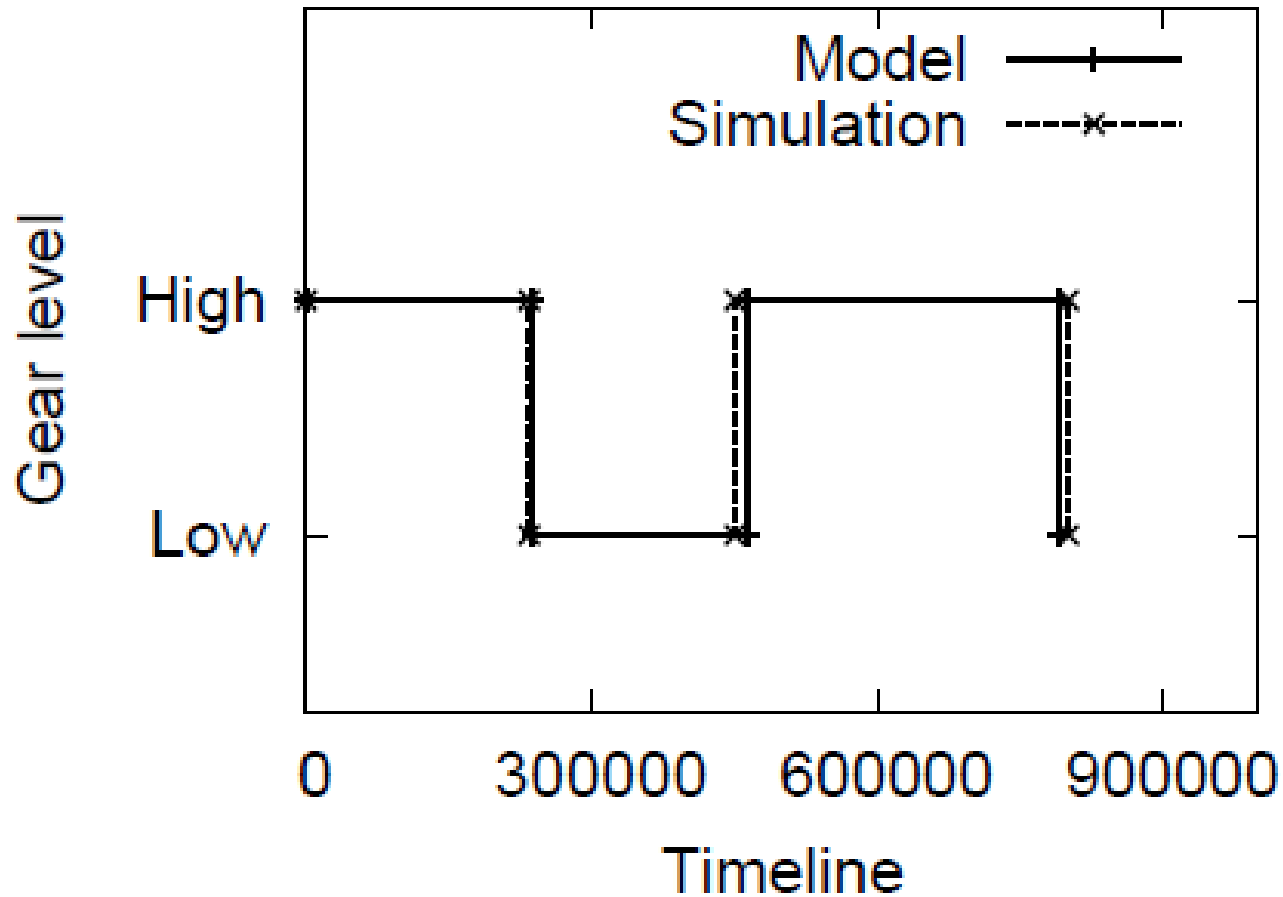
# Outline

- Introduction
- Motivation
- Modeling RAID performance
- PARAID
  - Overview
  - Modeling PARAID
- E-PARAID
  - Design
  - Modeling E-PARAID
  - Evaluation
- **Model Validation**
- Conclusion

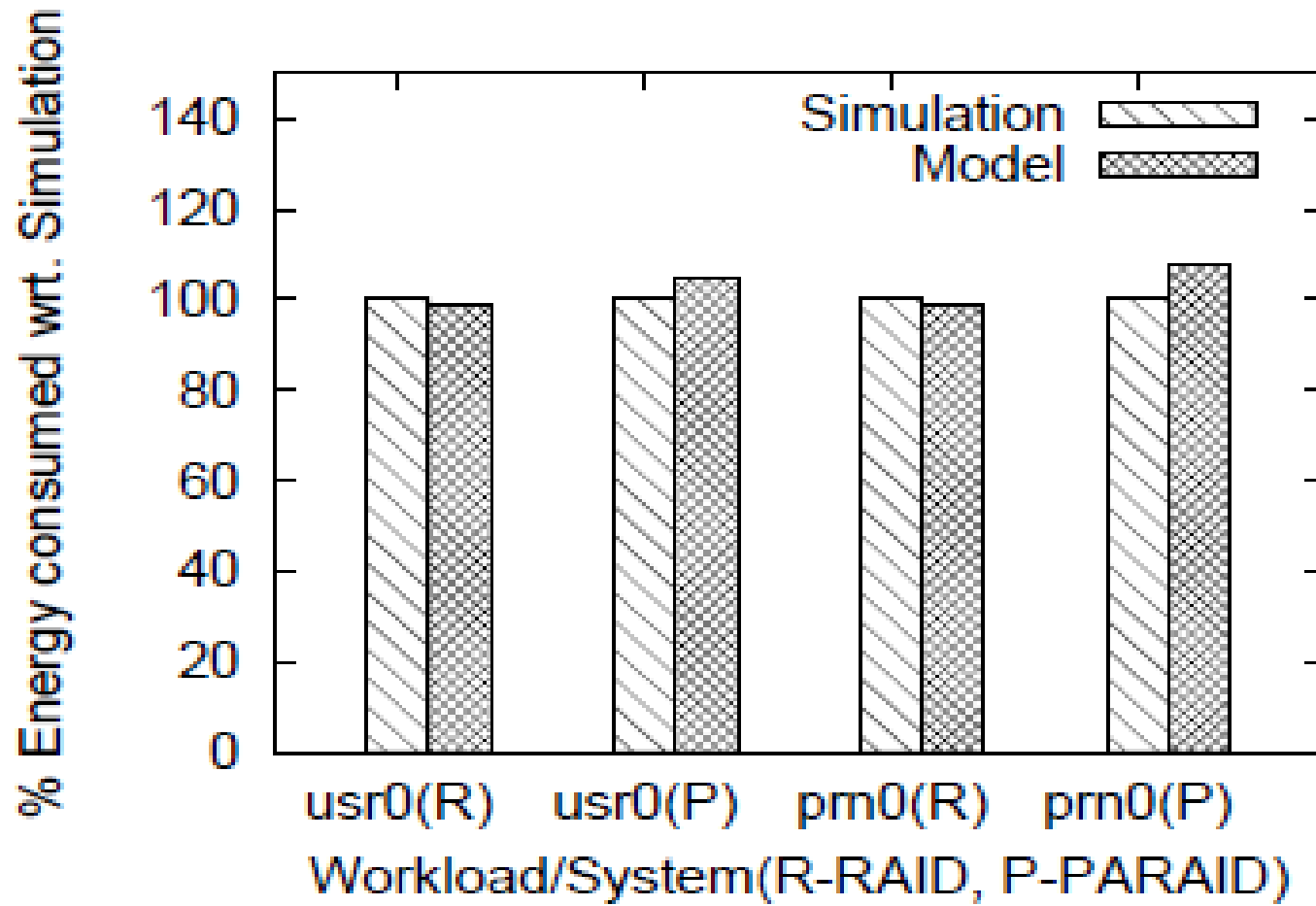
# Comparison with simulation results



# Comparison with simulation results



# Comparison with simulation results



# Outline

- Introduction
- Motivation
- Modeling RAID performance
- PARAID
  - Overview
  - Modeling PARAID
- E-PARAID
  - Design
  - Modeling E-PARAID
  - Evaluation
- Model Validation
- **Conclusion**

# Summary

- Proposed a novel and simplified method of modeling performance impact of power on storage systems
- Given a block-level request trace, our system computes near-accurate bandwidth/latency and energy consumption values
- Proposed an enhancement to PARaid technique that improves performance significantly
  - While conserving the same amount of energy as the original PARaid

# Conclusion & Future Work

- Model generic enough to represent wide range of energy algorithms
  - Those involving data migrations on a RAID based storage
- E-PARAID saves as much energy as PARAID does, with much better performance
- We plan to extend the model to represent other well-known energy algorithms
- Model can be enhanced to account for disk scheduling, controller-level caching effects, etc.,

**QUESTIONS ?**