
# Energy Proportionality and Performance in Data Parallel Computing Clusters

July 21, 2011

**Jinoh Kim,** Jerry Chou, and Doron Rotem
Scientific Data Management Group
High Performance Computing Research Department
Lawrence Berkeley National Laboratory

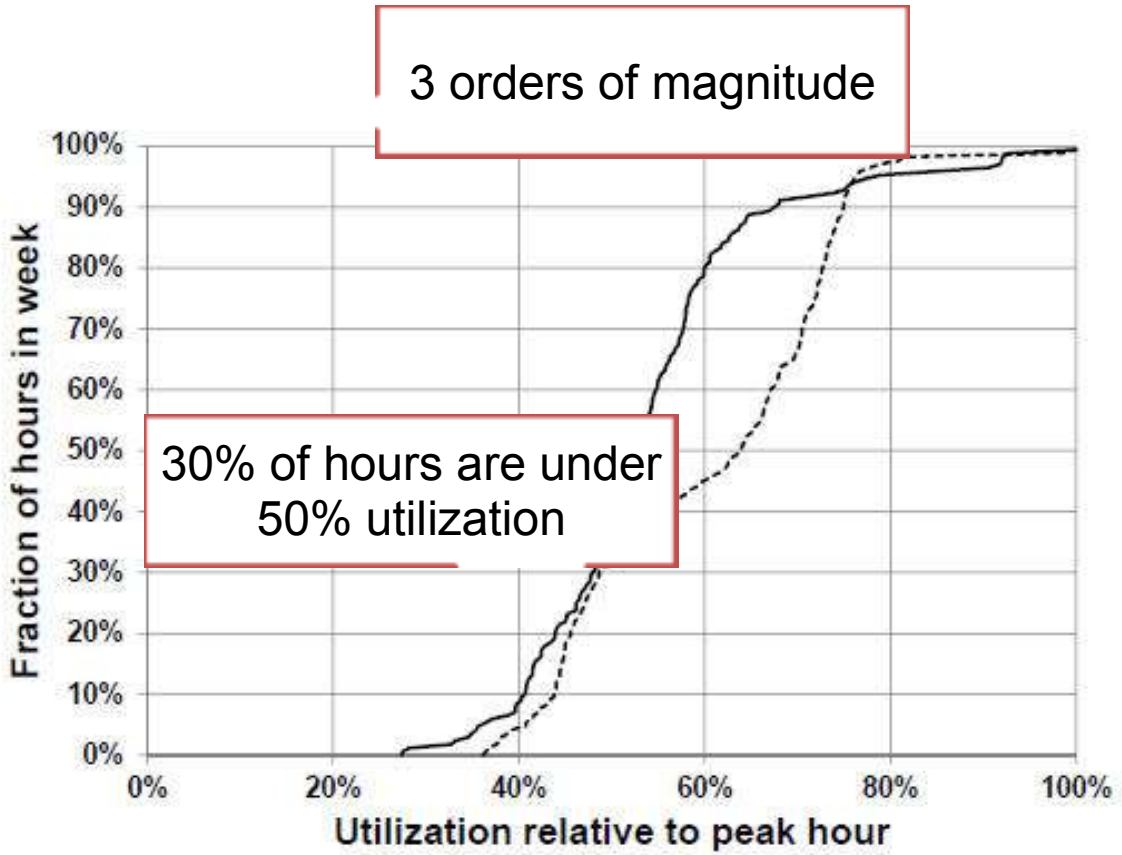
# Energy matters in datacenters!


- Datacenters consume over 2% of total power used in USA
- 70% of datacenters expect to be at their limit within the next three years

**Power-related costs occupy 47% of the total operational costs!**

(Source [Hamilton'09])

# Load is highly variable!

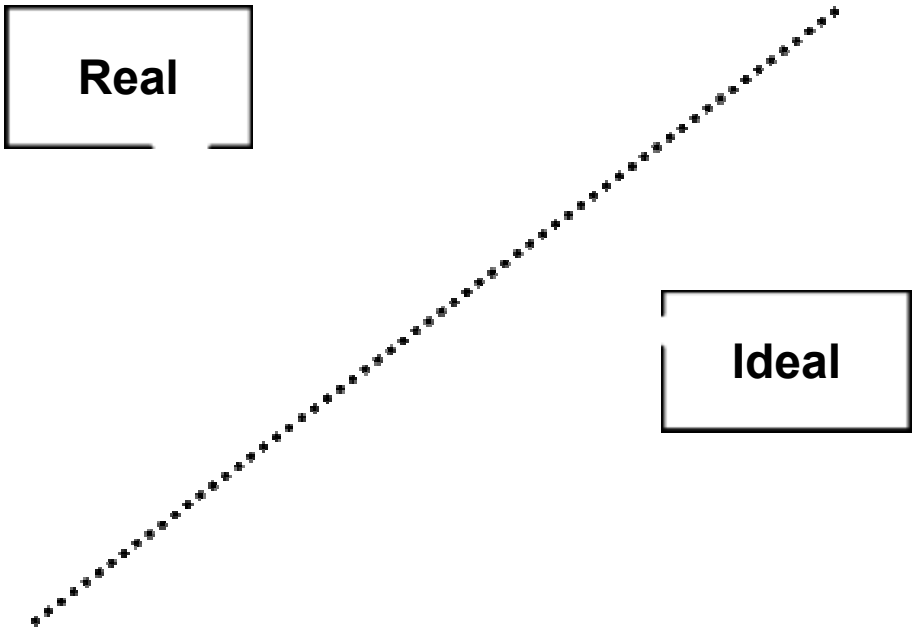



**Many opportunities for saving energy!**

Source [Thereska'10]

# Idle power is expensive!

- **Energy Proportionality**: consumes energy in proportion to load intensity




**Keeping servers in idle causes significant waste of energy!** [7]

# Outline

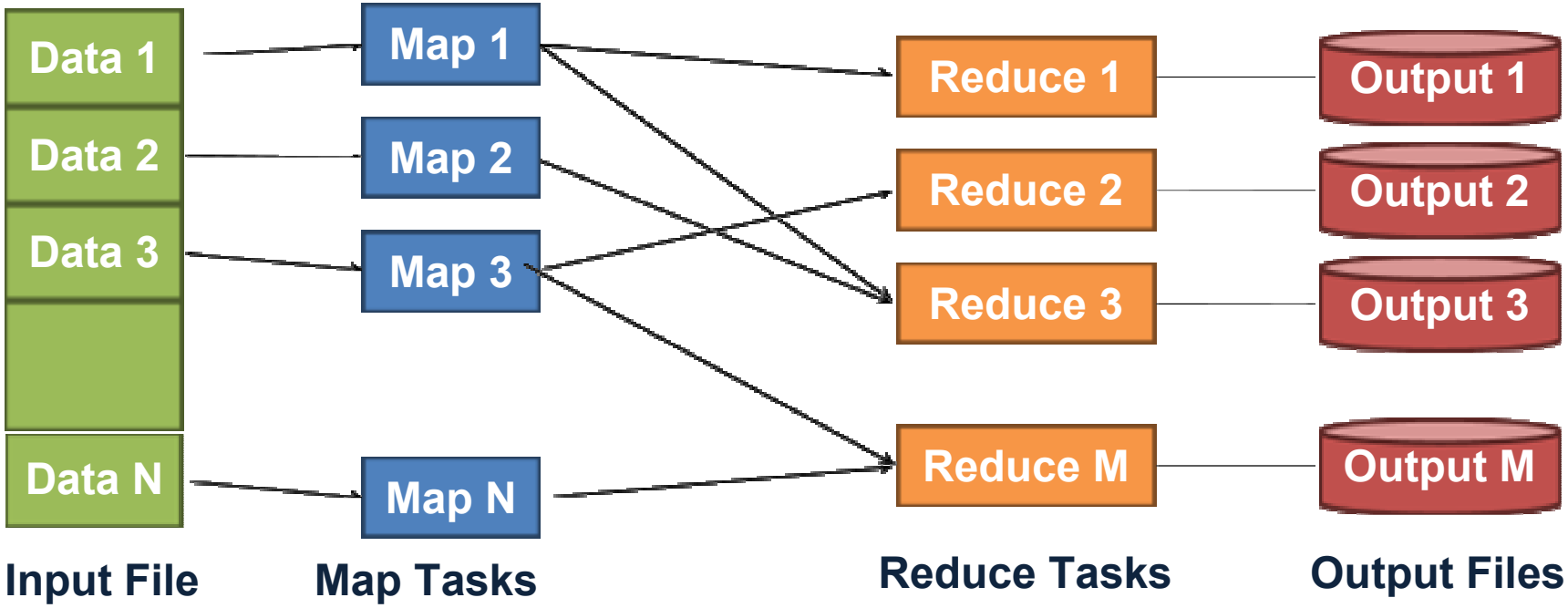
- Background
- Data Parallel Computing Clusters
  - MapReduce Clusters
  - Related Work on MR Energy Management
- Proposed Approach
  - Dynamic CS Discovery
  - Power-aware CS Discovery
  - Multi-level CS for Energy Proportionality
- Evaluation
- Conclusion

# Data Parallel Execution Clusters


- MapReduce (MR) clusters
  - *de facto* data parallel execution platform with map and reduce functions
  - Use data replication for parallelism and fault tolerance
  - Hadoop, Dryad, etc.
- Example: NY Times
  - Convert 4TB of raw images TIFF data into PDFs
  - $240 for Amazon MapReduce cloud service

# MapReduce Execution

- Applications: sort, grep, word count
- E.g., 1 TB sort



# MR Energy Management

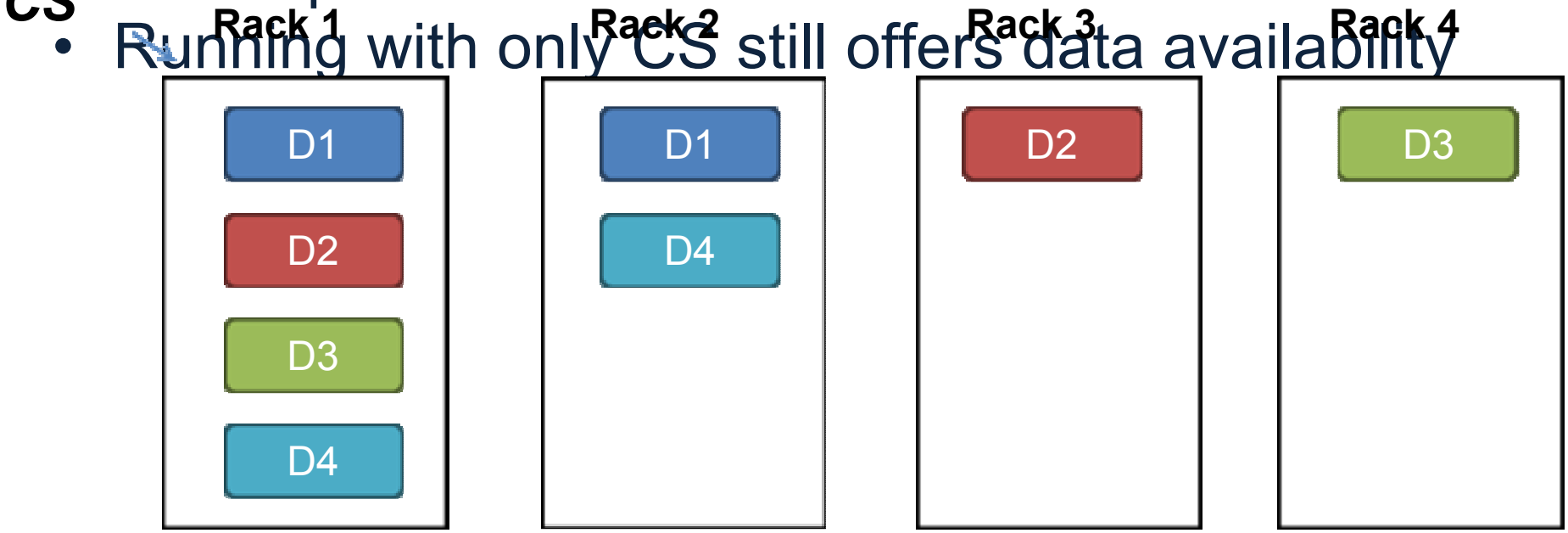

- MR Energy Management
  - Send (a subset of) nodes to low-power mode under light loads
  - Idle power >> standby power
    - E.g., Xeon node: idle = 260W, standby = 18W

# Challenges


- Data availability
  - Dehibernating node is expensive
- Node heterogeneity
  - Clusters can consist of multiple generations of H/W
  - Nodes may have different power requirements
- Energy proportionality
  - Use energy in proportion to given loads

# Existing Work

- Covering Subset (CS)
- One replica should be held in the set of CS nodes
- Running with only CS still offers data availability



# Existing Work (Cont'd)


- All-in Strategy (AIS)
  - Run jobs as quickly as possible, and send the entire cluster to low-power mode
  - Batch-style processing of MR clusters
- Rabbit
  - Propose a new skewed replication
  - Provides energy-proportional power management

# Existing Work (cont'd)

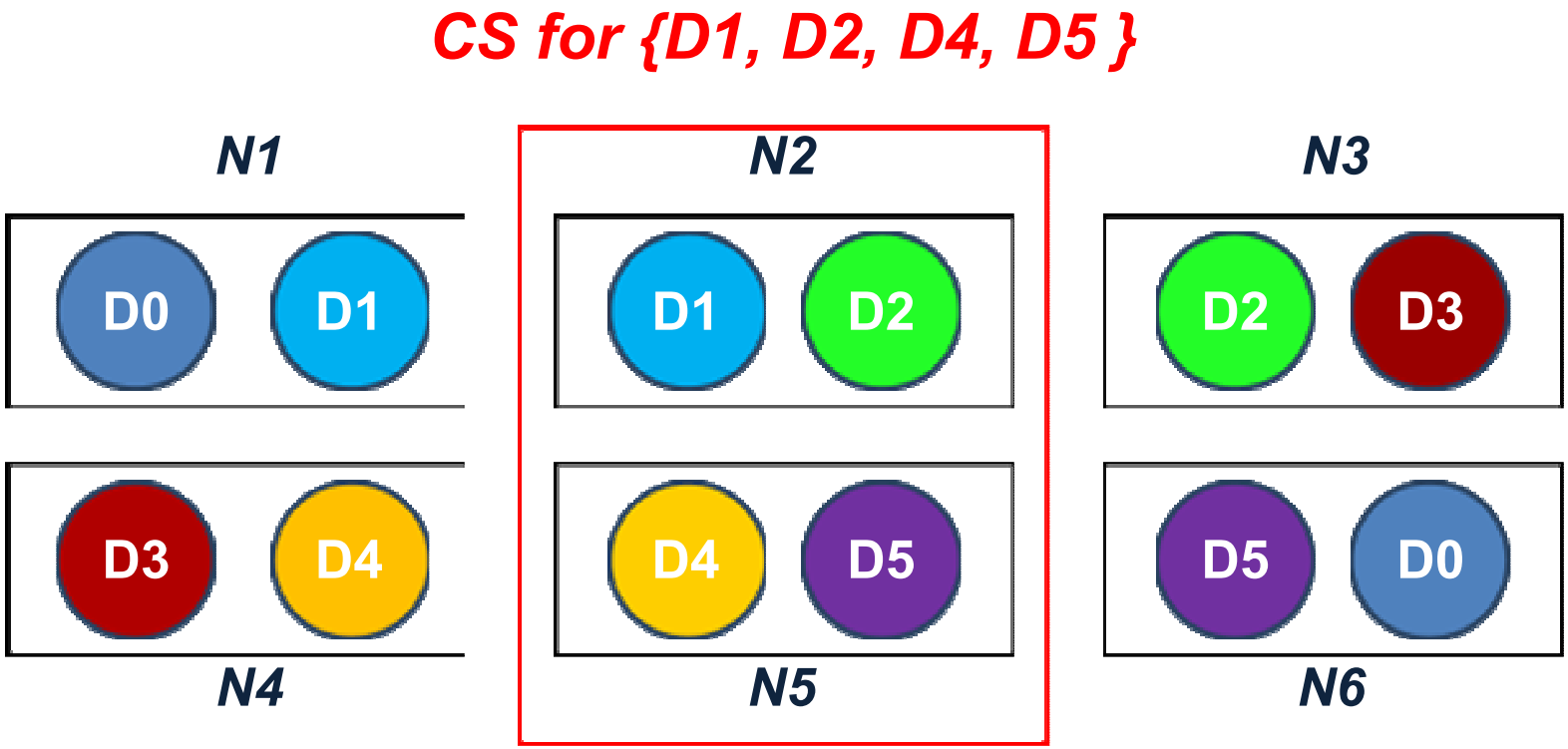
- (Potential) Drawbacks
  - No consideration of heterogeneity: All
  - No energy-proportional management: CS and AIS
  - Imbalanced data placement: CS and Rabbit

# Outline

- Background
- Data Parallel Computing Clusters
  - MapReduce Clusters
  - Related Work on MR Energy Management
- Proposed Approach
  - Dynamic CS Discovery
  - Power-aware CS Discovery
  - Multi-level CS for Energy Proportionality
- Evaluation
- Conclusion

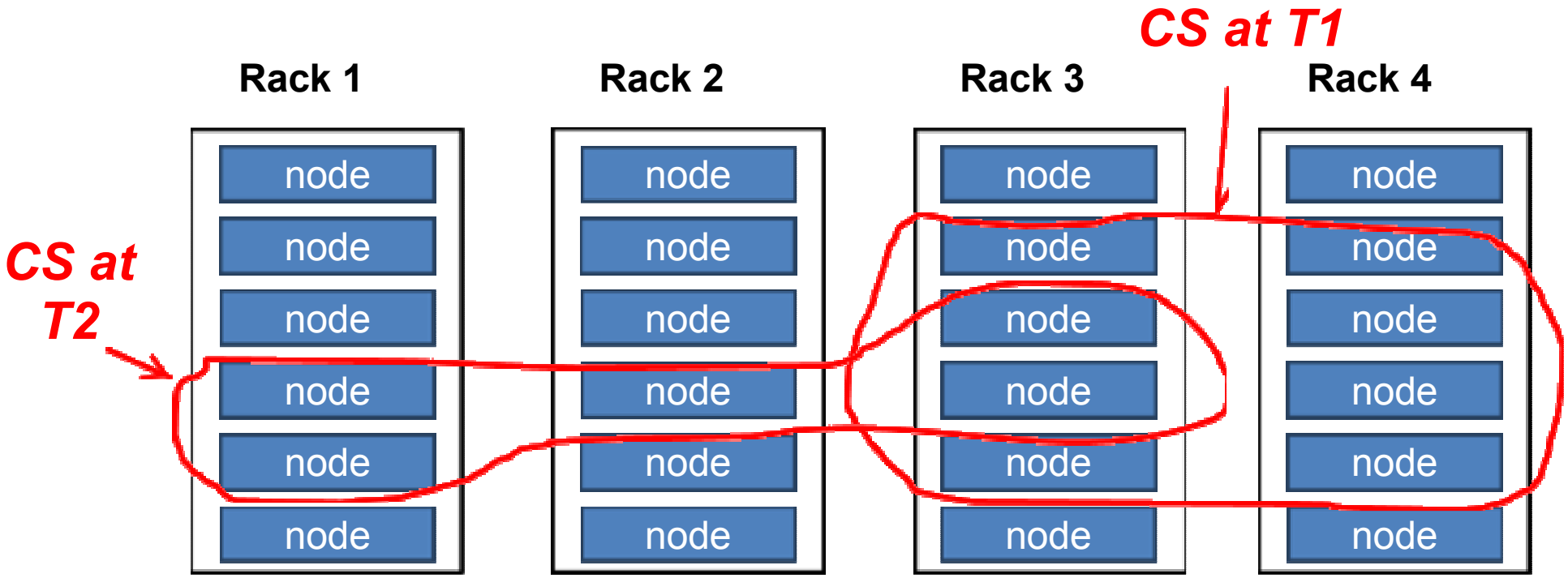
# Our Approach

- Activate a minimal set of nodes (CS) ensuring *data availability* under light loads
  - The other nodes (non-CS) are *hibernated* for power saving
  - No modification of uniform data replication



# Dynamic CS Discovery

- Constructs CS dynamically based on data req.
  - Data req. at T1 = { D1, D2, .., Dn }
  - Data req. at T2 = { Dx, Dy, .., Dz }



# Analysis of Minimal CS Size


**Theorem**: *The minimal m such that we can expect at least one CS from any given uniform data layout satisfies*:

- *n*: number of nodes in the cluster
- *m*: number of nodes randomly chosen
- *r*: replication factor
- *b*: number of data blocks

# Power-aware CS Discovery

- Nodes can be *heterogeneous* (w/ different power levels)
- Minimal CS size ≠ minimal power requirement
- Power requirement in CS approach:
- ***Select nodes with minimal gaps between active power and standby power for CS***

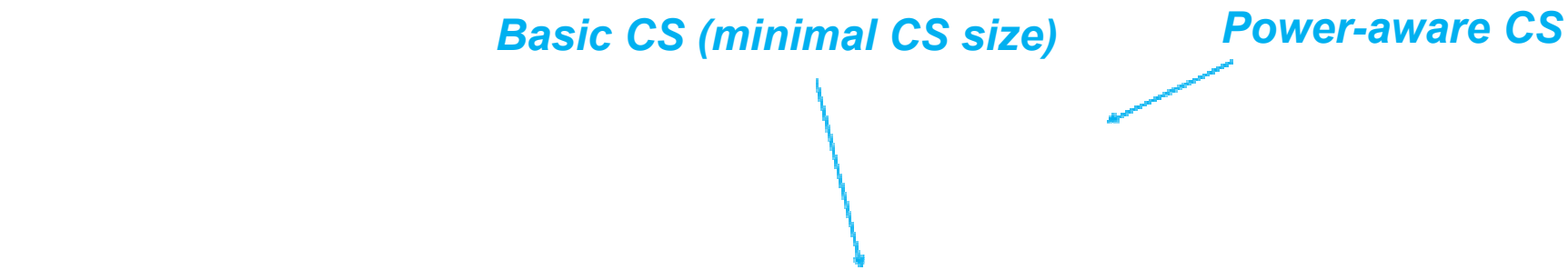
# Power-aware CS Algorithm

- Reduced to “weighted set cover” problem
- Algorithm:

***Weight is derived from the equation in previous slide and the number of replicas the node has for CS***

Run-time complexity = $O(|B||S|\ min(|B|,|S|)$

# % Low Power Nodes in CS



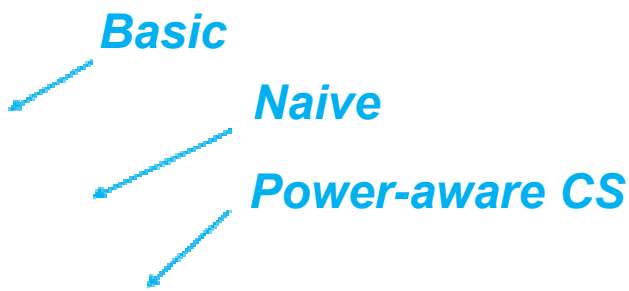

% LP Nodes in CS

Fraction of LP Nodes

- Two classes of nodes: HP (High Power) and LP (Low Power)
- Power-aware algorithm selects more LP nodes for CS (for power optimization)

# Power Requirements

- Four classes of nodes
  - with power consumption: LP < ML < MH < HP
- Naïve: selects a node with the lowest power first



# CS Reorganization

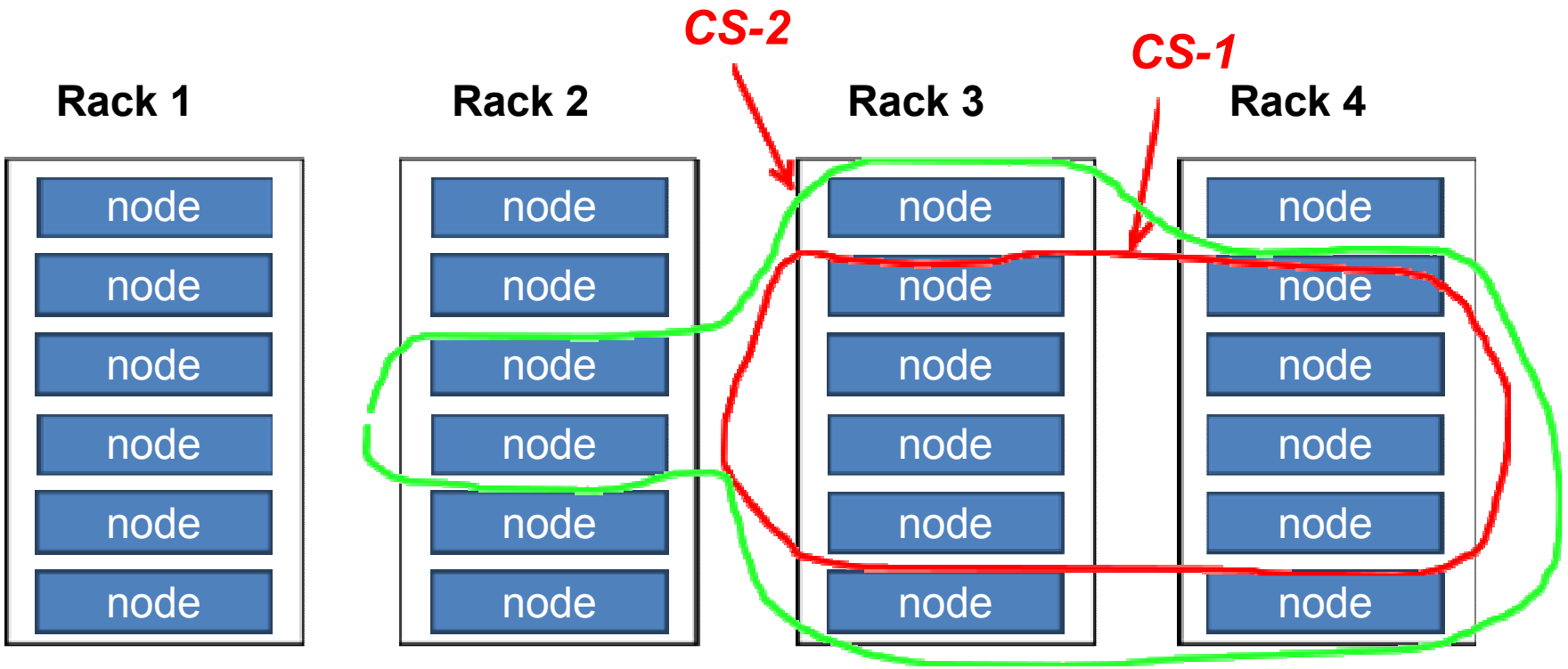

- MapReduce clusters are failure-prone
- Incrementally add nodes for unavailable data (due to failure)
- Periodically construct new CS for a new data set

**Number of active nodes**

**Number of nodes in CS**

# Multi-level CS

- Significant correlation between performance and a degree of data availability in MR clusters
- Find CS-*k* that provides *k* replicas
  - Data req. for CS-1 = { D1, D2, .., Dm }
  - Data req. for CS-2 = { D1, D2, .., Dm } X 2



# Outline

- Background
- Data Parallel Computing Clusters
  - MapReduce Clusters
  - Related Work on MR Energy Management
- Proposed Approach
  - Dynamic CS Discovery
  - Power-aware CS Discovery
  - Multi-level CS for Energy Proportionality
- Evaluation
- Conclusion

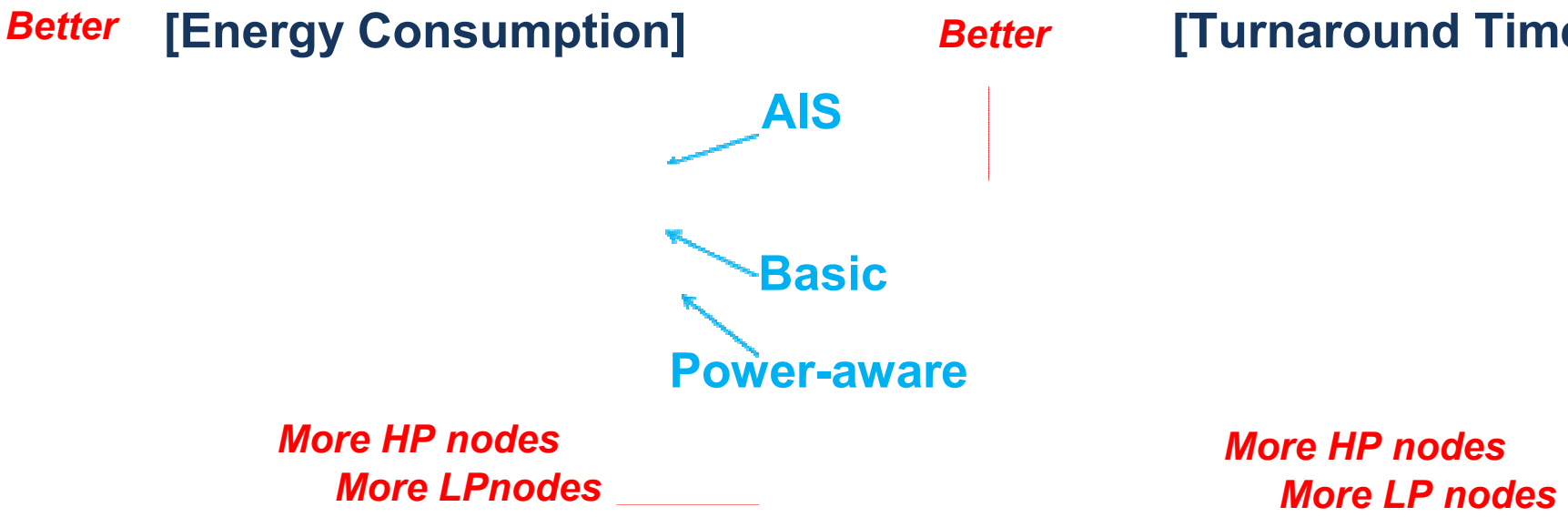
# Experimental Setting


- Develop a simulator based on OMNeT++
- Cluster size = 1024
  - Heterogeneous cluster: LP:HP = 50%:50%
  - Have different power and capacity
  - Power measures from SPECpower for Xeon and Atom
- Accessed data blocks = 1TB
- Interested in light load for power saving
- Considered networking overhead with distribution models

# Compared Techniques

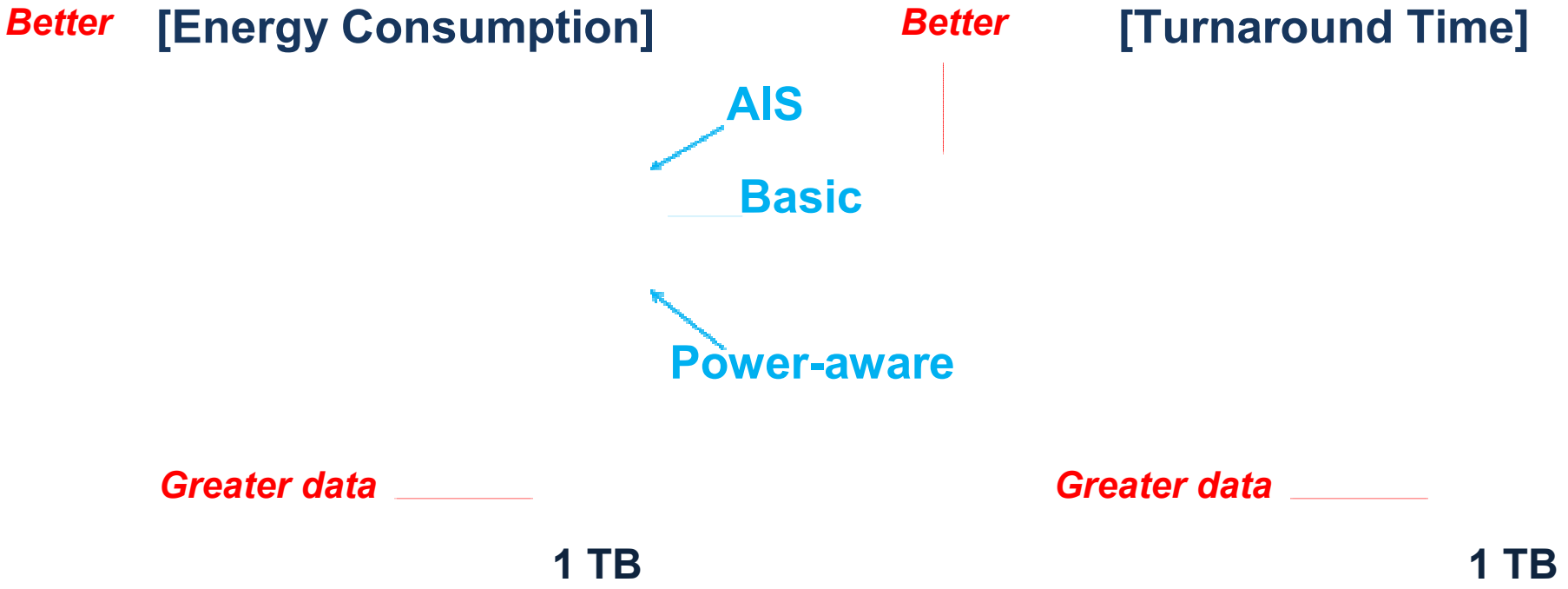

- NPS: No power saving
- AIS: All-in strategy
  - Run jobs as quickly as possible, then the entire cluster is hibernated
- Basic: Non power-aware CS
  - Dynamic CS discovery for minimal CS size
- PA: Power-aware CS
  - Dynamic CS discovery with power weights

# Impact of Heterogeneity




- Power-aware CS exploits heterogeneity!
  - > 25% energy saving with little performance loss
  - 55% energy saving where the cluster consists of 25% high-power and 75% low-power nodes

# Impact of Data Size

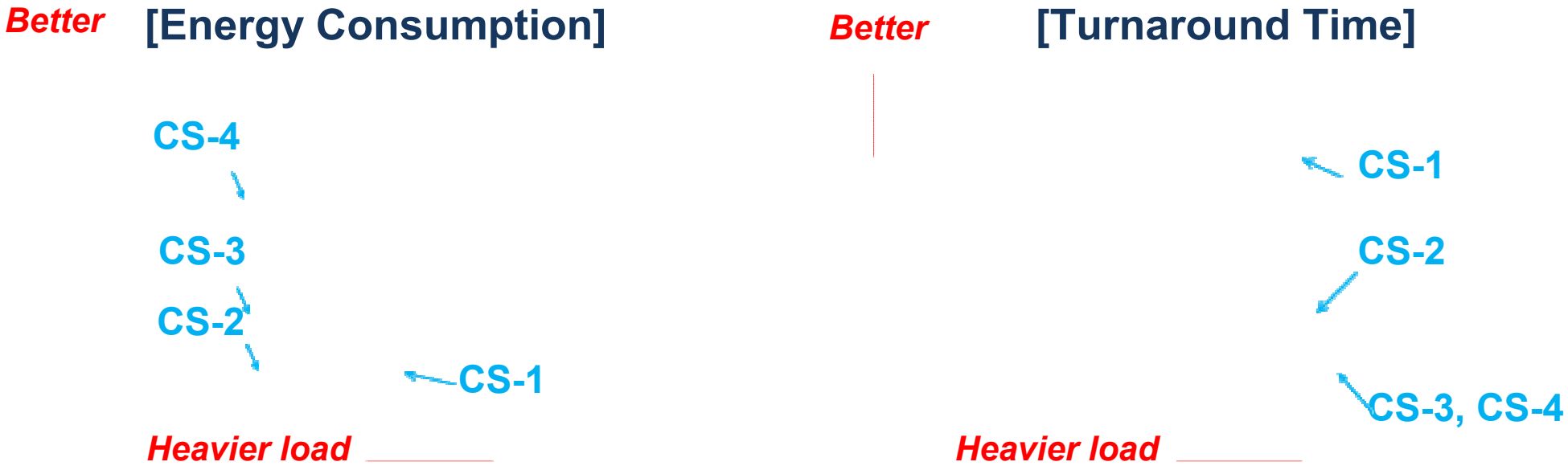



- CS yields greater power saving with smaller data requirements
- Power-aware CS saves 30-70% energy compared to NPS
- No impact on performance

# Data Transfer Overhead


- No previous study on data transfer distribution model
- *Norm(d)*: normal distribution with □ = *d*
- *Exp(d)*: exponential distribution with □ = *d*
- Data overhead is a function of given computation time and distribution with *d*

# Evaluation: Multi-level CS




- Replication factor = 5 and # data blocks = 1024
- Different energy saving and performance in different CS levels

# Conclusion


- Increasing needs of MapReduce clusters to handle large-scale data sets
- Existing energy management techniques
  - No energy proportionality
  - No consideration of heterogeneity
- Proposed power-aware CS discovery
  - Exploits heterogeneity
  - Resilient to node failure by reorganization
  - Provide energy proportionality with multi-level CS

# Future Work


- How to dynamically determine CS switching?
  - Which CS level is appropriate for the next epoch?
- How to optimize both performance and energy?
  - Currently our optimization is energy-centric
  - Need to think about performance requirements (e.g., service level agreements)
- Study on data transfer model in MR clusters


# *THANK YOU!*

**Jinoh Kim**
**E-mail: jinohkim@lbl.gov**
**https://hpcrd.lbl.gov/~jinohkim**