On the Energy (In)efficiency of Hadoop Clusters

Jacob Leverich, Christos Kozyrakis

Presented by: Rini Kaushik

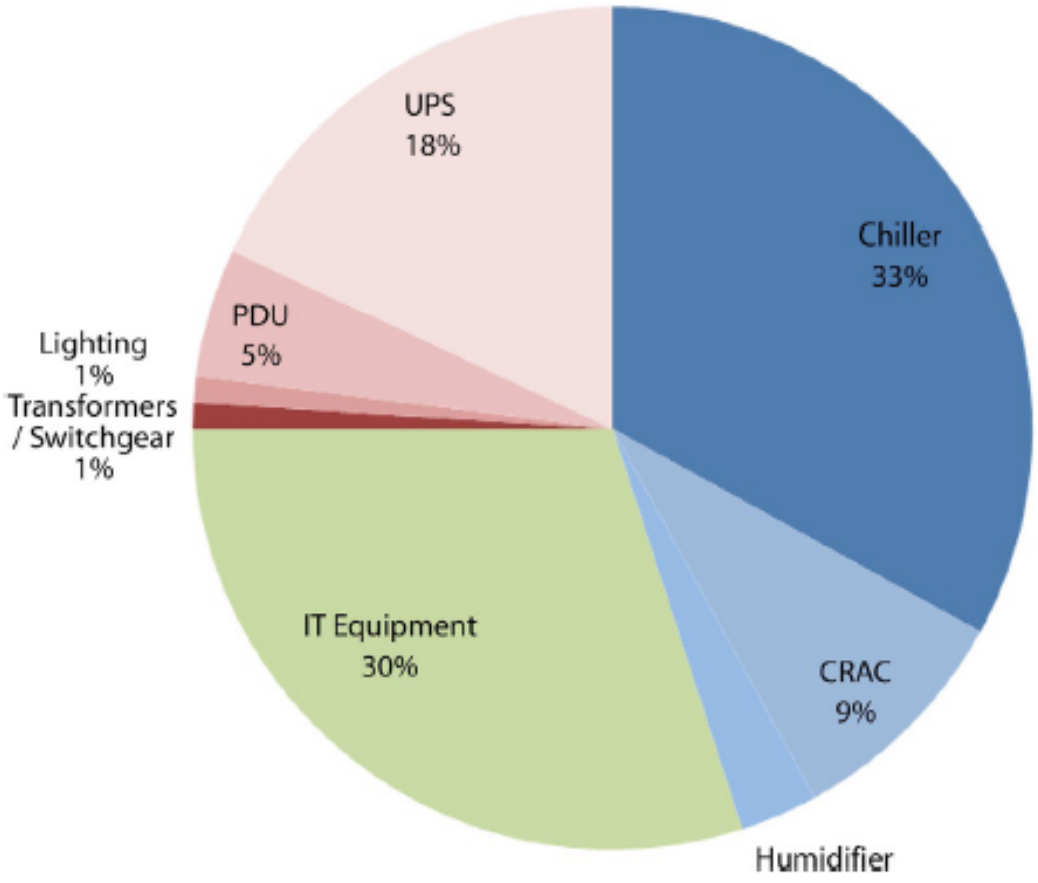
Why is energy-management important?

- US datacenters
  - Energy costs (*EPA*):
    - 2003 - $2Billion
    - 2011 - $10 Billion
    - ~1.6% of all energy consumed
  - (Un)Green
    - 12M tons of CO2 annually*
  - Servers worldwide
    - 2005 - 27.3 million (*InformationWeek*)

**Jeff Chase et. al., Managing Energy and Server Resources in Hosting centers*

** J. Jackson, energy needs in an internet economy: a closer look at the datacenters*

Where is the energy consumed?



PUE = 3.0
2011 = 1.4

Courtesy: Luiz André Barroso, Urs Hölzle

Towards energy-efficiency

Opportunity: Reality on CPU Utilization

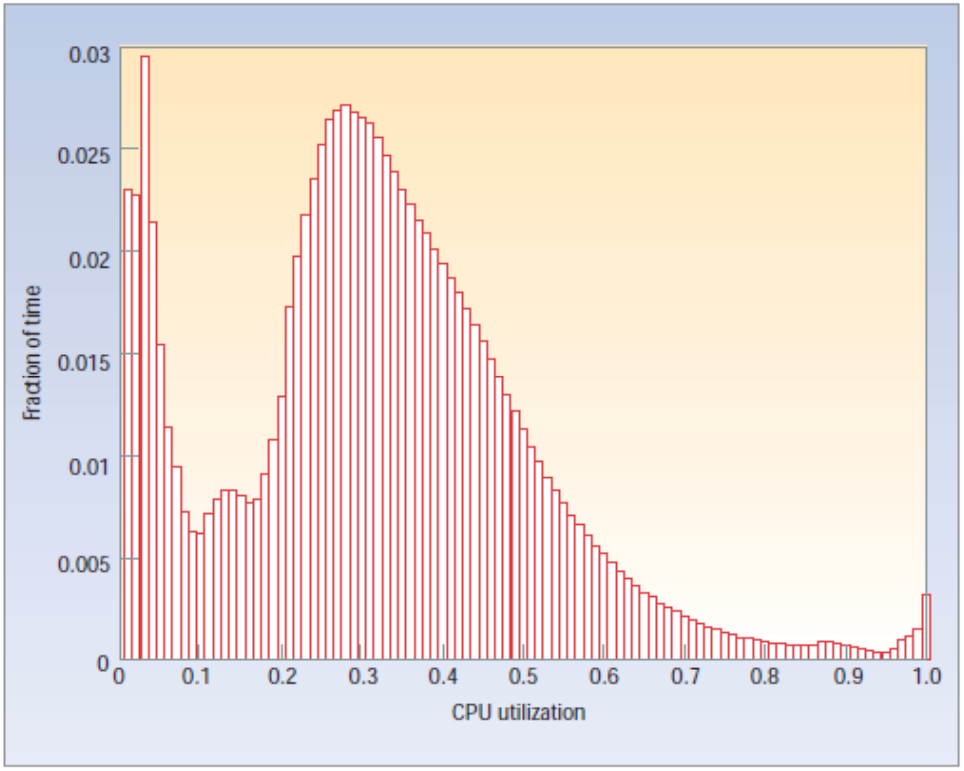
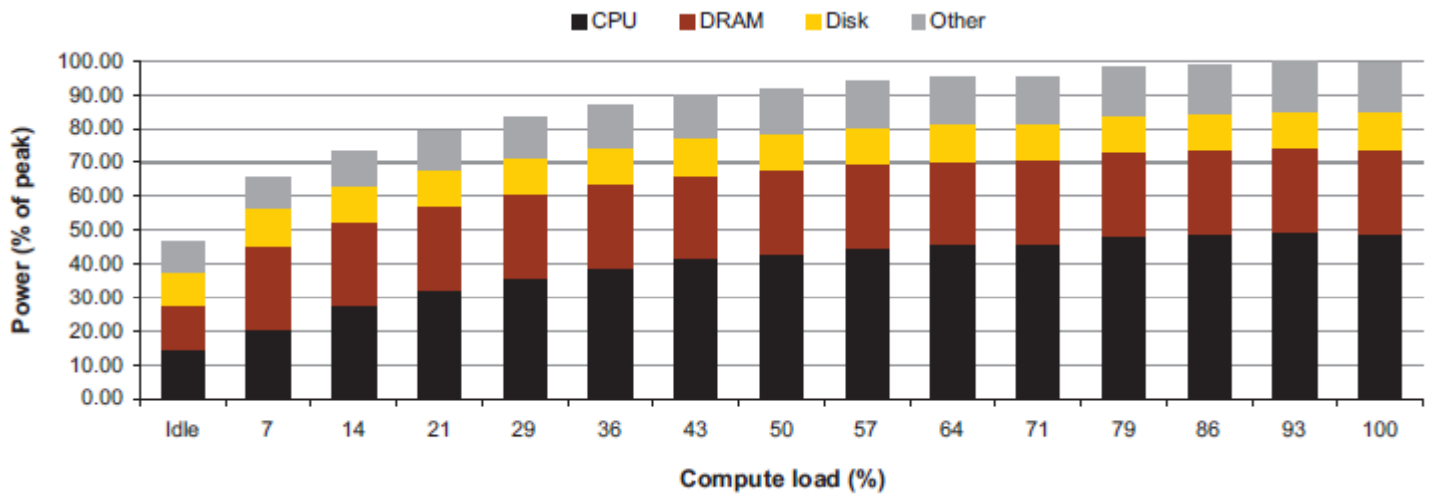


Figure 1. Average CPU utilization of more than 5,000 servers during a six-month period. Servers are rarely completely idle and seldom operate near their maximum utilization, instead operating most of the time at between 10 and 50 percent of their maximum utilization levels.

Ref: The Case for Energy-Proportional Computing, Luiz André Barroso and Urs Hölzle

Power variation in a typical server



Courtesy: Luiz André Barroso, Urs Hölzle

Power vs efficiency

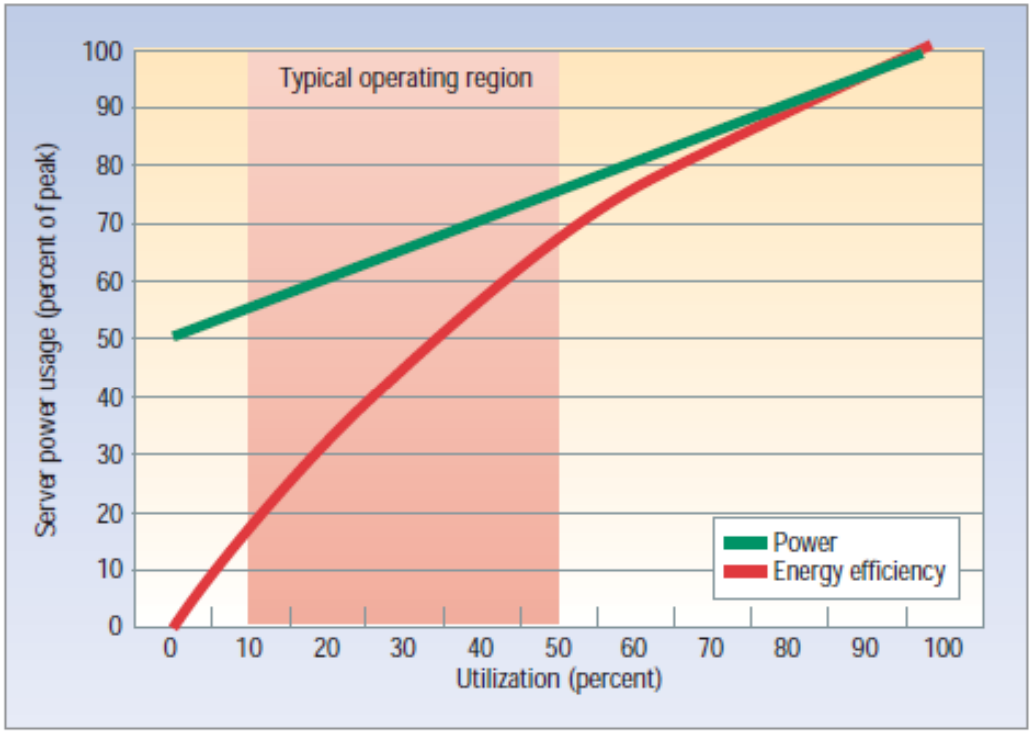


Figure 2. Server power usage and energy efficiency at varying utilization levels, from idle to peak performance. Even an energy-efficient server still consumes about half its full power when doing virtually no work.

Efficiency = utilization/power

Ref: The Case for Energy-Proportional Computing, Luiz André Barroso and Urs Hölzle

Easy energy-efficient option

- Scale-down
  - Match number of active nodes to workload needs
  - Turn-off remaining nodes to save power
    - Multiple papers use this approach:
      - *Managing Energy and Server Resources in Hosting Centers, SOSP'2001*
- *Easy when*
  - *Only requires computation consolidation*
  - *Servers stateless (i.e., serving data that resides on a shared NAS or SAN)*
  - *Simple replication model*
    - *Workloads can be migrated to fewer machines during periods of low activity*
- *Hard when*
  - *Servers with significant state*
  - *Data locality important*

Hadoop Primer

- Distributed data processing framework
- The MapReduce programming model has emerged as a scalable way to perform data-intensive computations on commodity cluster computers
  - Commodity datacenter
  - HDFS

Unique scale-down challenges of Hadoop clusters

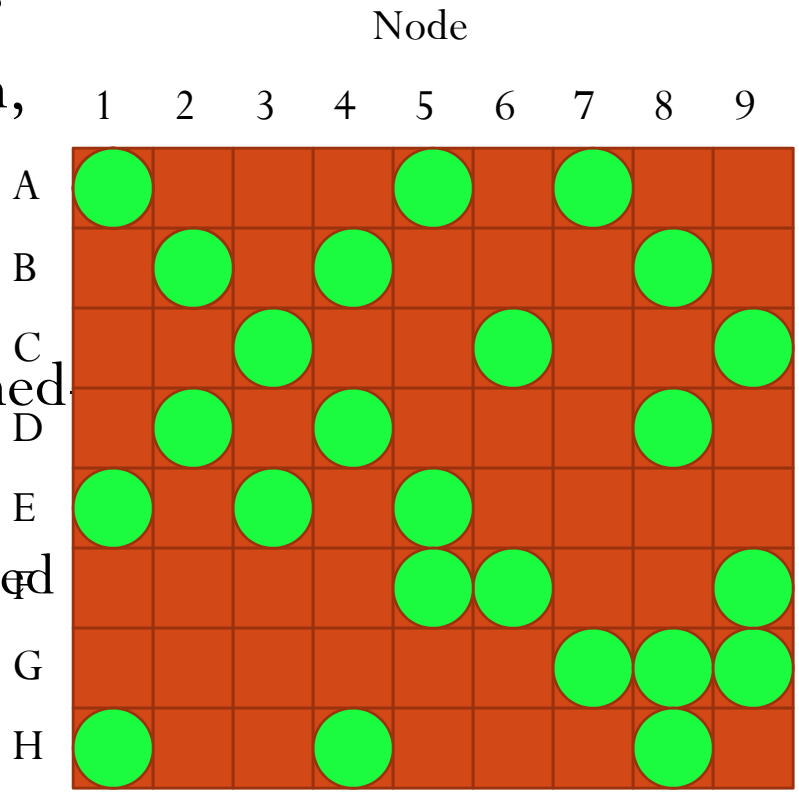
- Computation and data co-located on servers
  - Servers stateful
- Servers rarely completely idle
  - Design principles:
    - Load Balancing for better performance
      - Even in low activity, low load in multiple servers than high load in few servers
      - Data striped across nodes
        - High aggregate IO
  - Commodity servers usage raises reliability and availability concerns
    - N-way replication a norm
- Result - Hard to turn-off servers

Scale-down Opportunity: BlockReplication

- Invariants
  - No two replicas on same node
  - Replicas on atleast two racks
- If inactive node turned down, data still available on replica

Naïve approach

- Only n-1 servers can be turned-off
- At best, only one rack off
  - Otherwise, availability affected



Courtesy: Leverich, HotPower'09

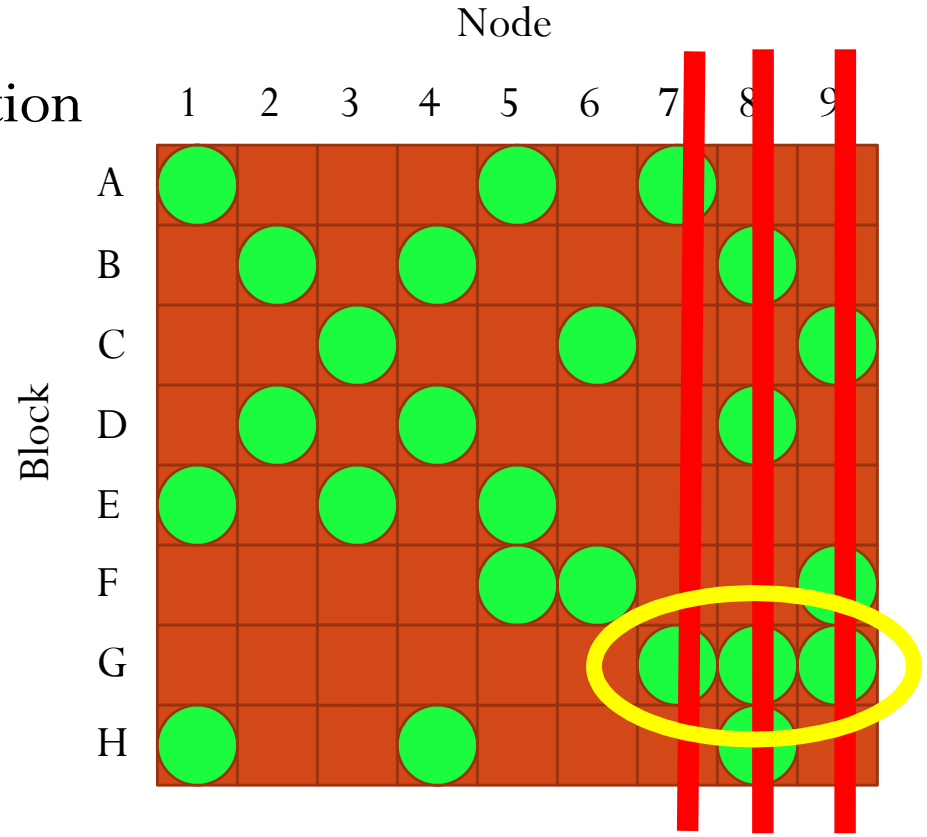
Raises Questions

Which node to disable?

- Data availability consideration

How to distinguish sleeping node from Down node?

- To prevent rereplication



Covering subset invariant

- Invariant:

  Every block must have one replica in the covering subset.

Covering subset considerations

- Too large
  - - less energy savings
  - - Rest of the system suffers bottlenecks
  - + Performance of the covering set good
- Too small
  - - Limited in storage capacity
  - - Performance bottleneck
  - + higher energy saving
- Paper assumes 10 – 30%

Missing considerations and issues

- Assume system admin will establish covering subset
  - Has no knowledge of the workload patterns
  - No adaptability
- Adhoc 10 – 30% allocation of set can have serious consequences on performance and not cognizant of the workload patterns
- Number of files not accounted for

Changes to Hadoop

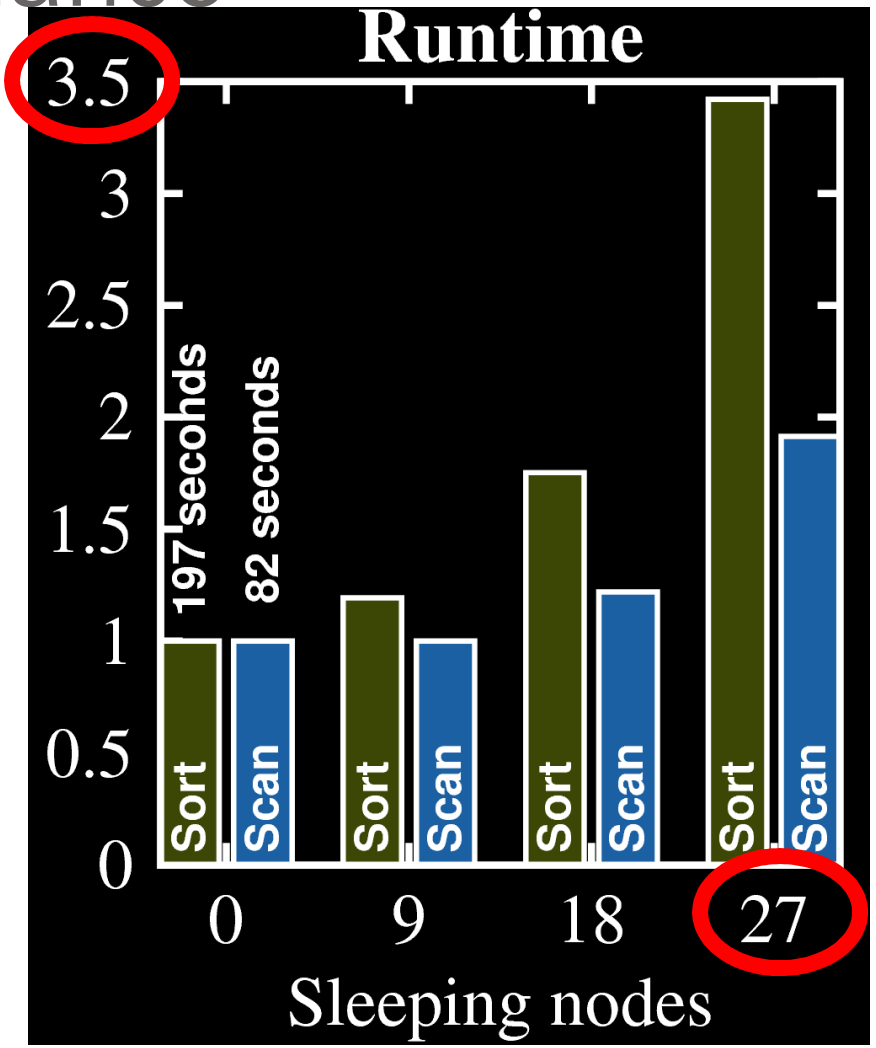
- ReplicationTargetChooser
  - One replica in local node
  - One replica in covering subset
  - One replica on a different rack
- No re-replication of the blocks on sleeping nodes
- Nodes disabled and enabled manually

Evaluation

- Disable *n* nodes, compare Hadoop job energy & perf.
  - Individual runs of webdata_sort/webdata_scan from GridMix
  - 30 minute job batches *(with some idle time!)*
- Cluster
  - 36 nodes, HP Proliant DL140 G3
  - 2 quad-core Xeon 5335s each, 32GB RAM, 500GB disk
  - 9-node covering subset (1/4 of the cluster)
- Energy model
  - Validated estimate based on CPU utilization
  - Disabled node = 0 Watts
  - Possible to evaluate hypothetical hardware

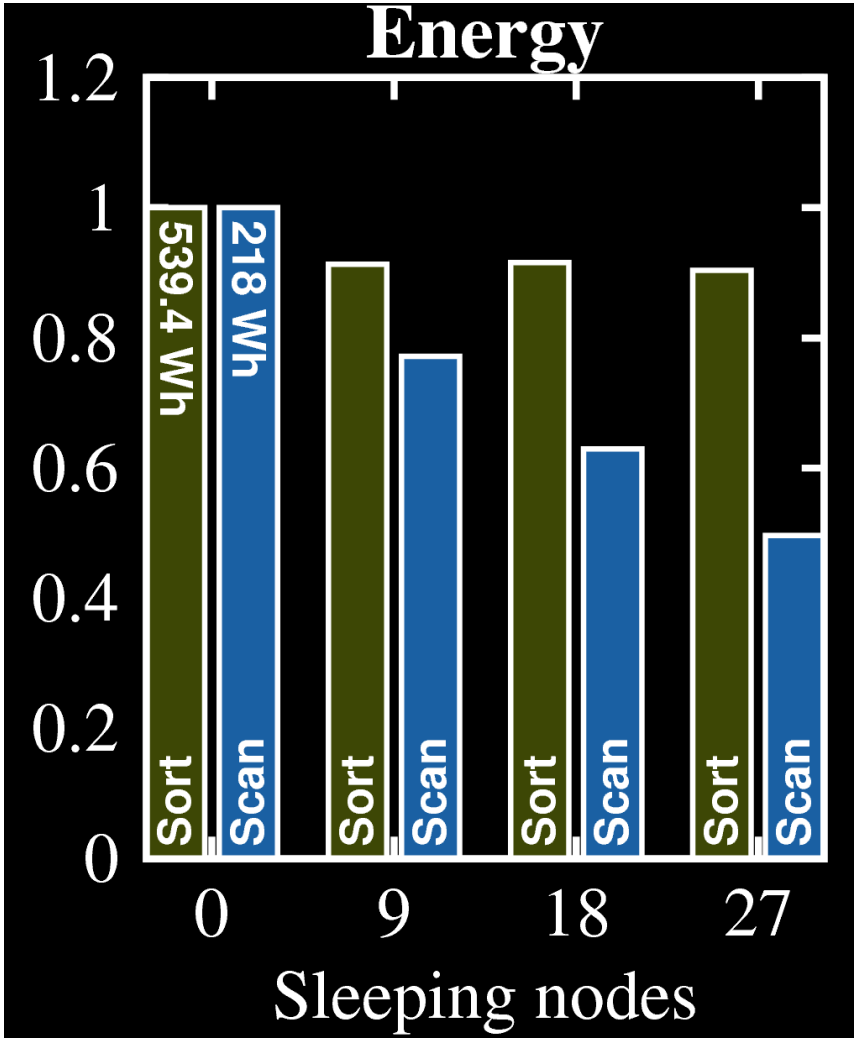
Results: Performance

- It slows down (obviously)
  - Peak performance benchmark
- Sort worse off than Scan



Results: Energy

- Less energy consumed for same amount of work
  - 9% to 51% saved



Evaluation

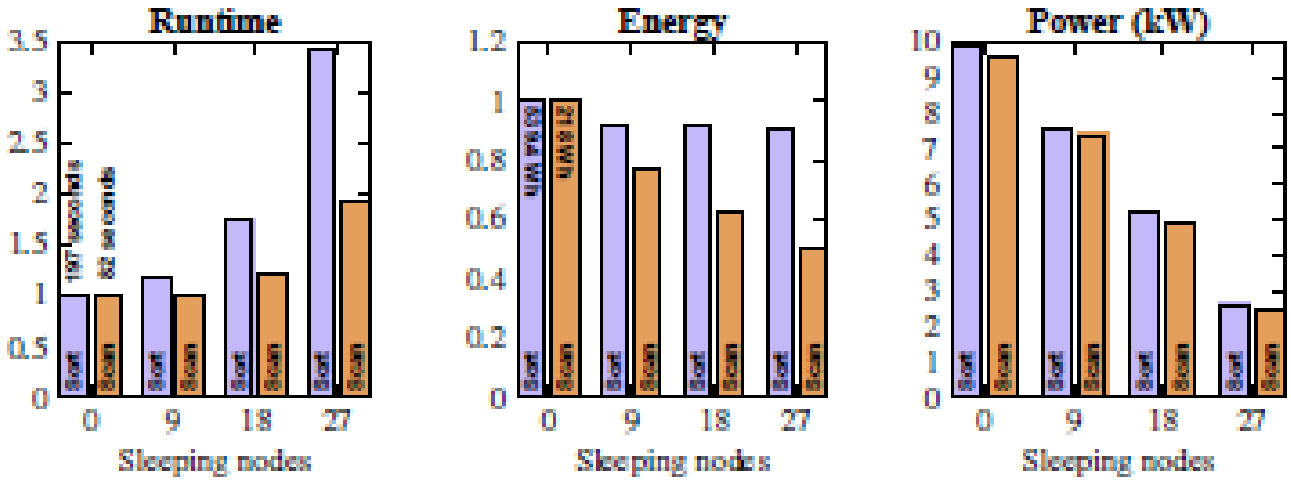


Figure 2: Runtime, Energy Consumption, and Average Power Consumption for the 32GB Sort and 32GB Scan workloads as nodes are disabled. Runtime and Energy are normalized to when all nodes are active.

Interesting observation – power goes down as the number of sleeping nodes is increased

However, energy-consumption may not.

Energy = power X time

Cost = Energy X cost/Kwh

Sleeping nodes ^ → performance v power

Sort – 9%, Scan – 51% energy saving

Performance impact Sort - 71%

Discussion

- Used a very small dataset in their experiments
- Made a statement that there is no impact on data availability which is incorrect
  - Fault injection experiments needed
- Assumed a power model where power used is dependent only on the cpu utilization. This may not be accurate. IO bound benchmarks will have a different characteristic.
- Replication is meant for performance also
  - Hot spots
- Tradeoff between availability, performance and energy-efficiency

Future work

- Impact on durability of sleeping nodes
- Revisiting reliability via replication assumption
  - Replication does have performance implications
- Dynamic scheduling
  - Responds to changes in utilization of the cluster
  - Collaboration between the hadoop's job scheduler and power controller
- Different workloads and their characteristics
  - Some may value QoS and throughput more than runtime