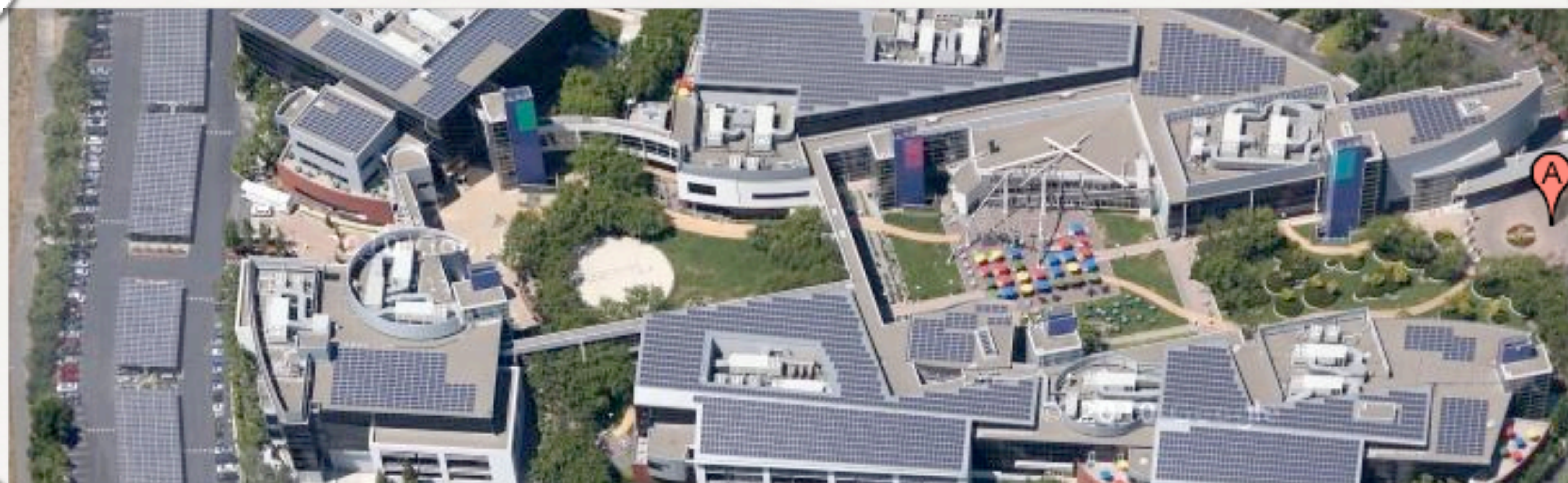
Energy Management of MapReduce Clusters

Jan Pohland
2518099



[maps.google.com]

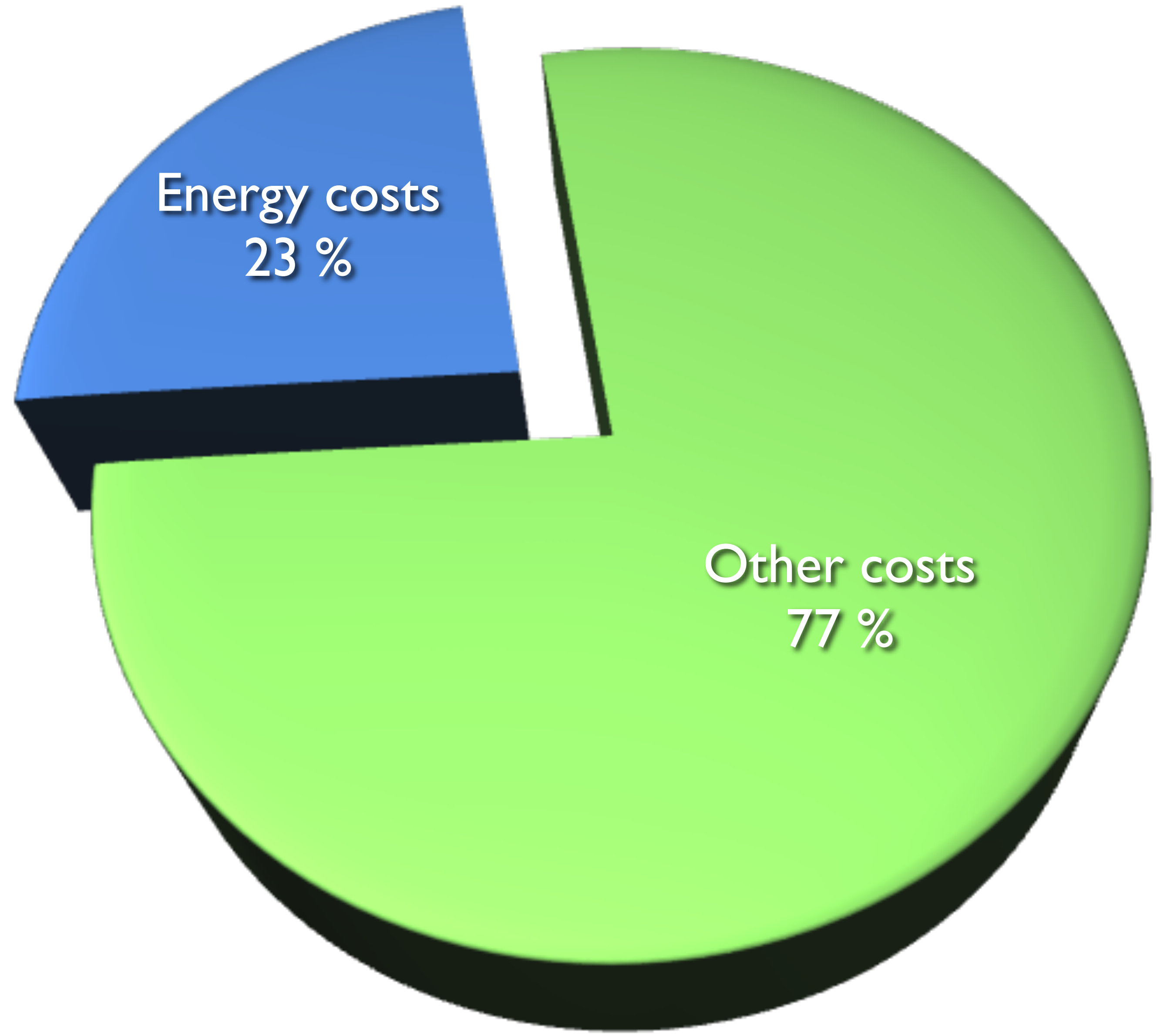
installed solar panels
on headquarters
1.6 MW
(1,000 homes)

invested $38.8 million
North Dakota wind farms
169.5 MW
(55,000 homes)

Introduction

Monthly costs of data center*:

Direct energy costs: 23 %



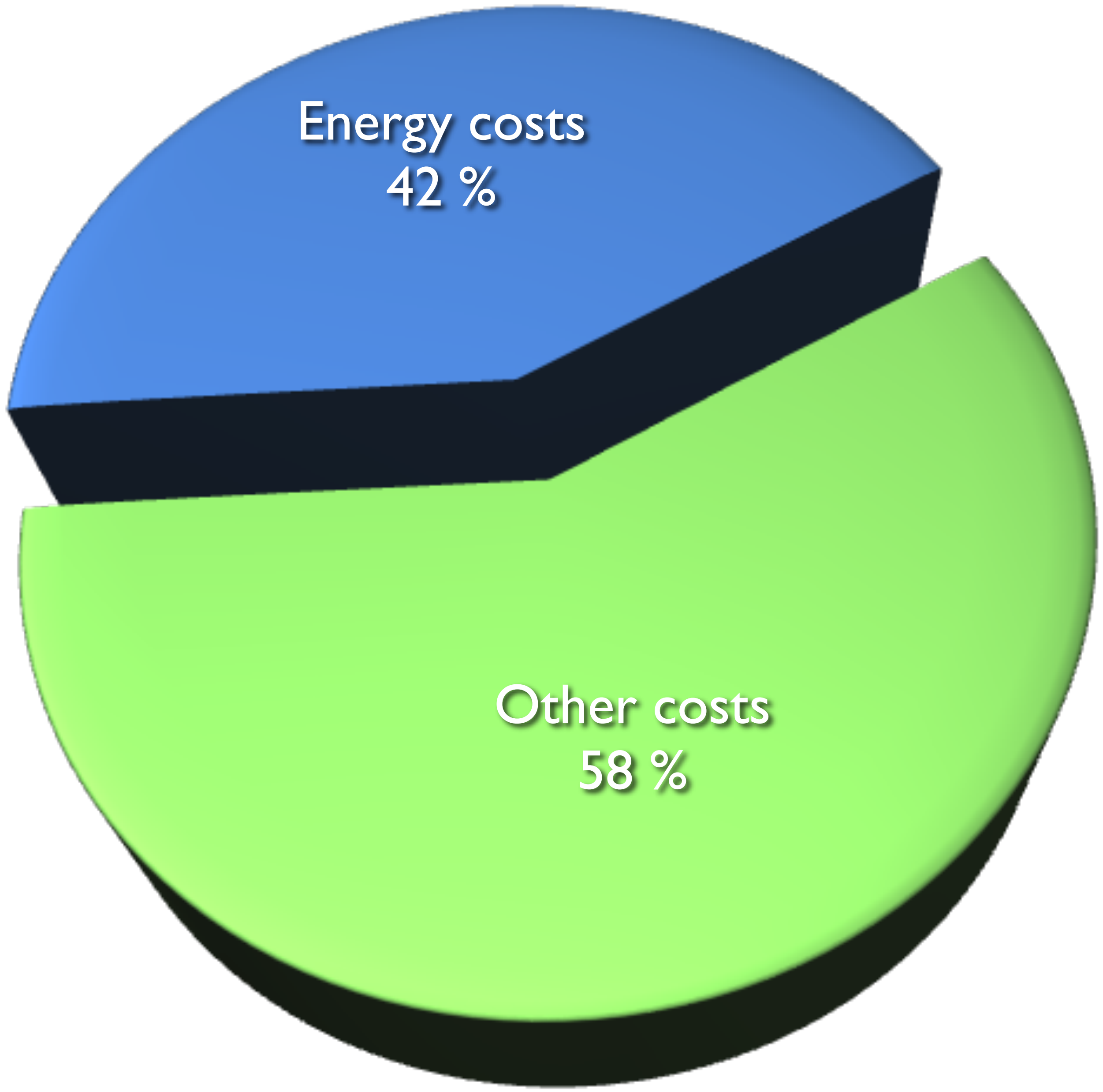
*: amortized

Introduction

Monthly costs of data center*:

All energy costs: 42 %

(incl. cooling infrastructure etc)



*: amortized

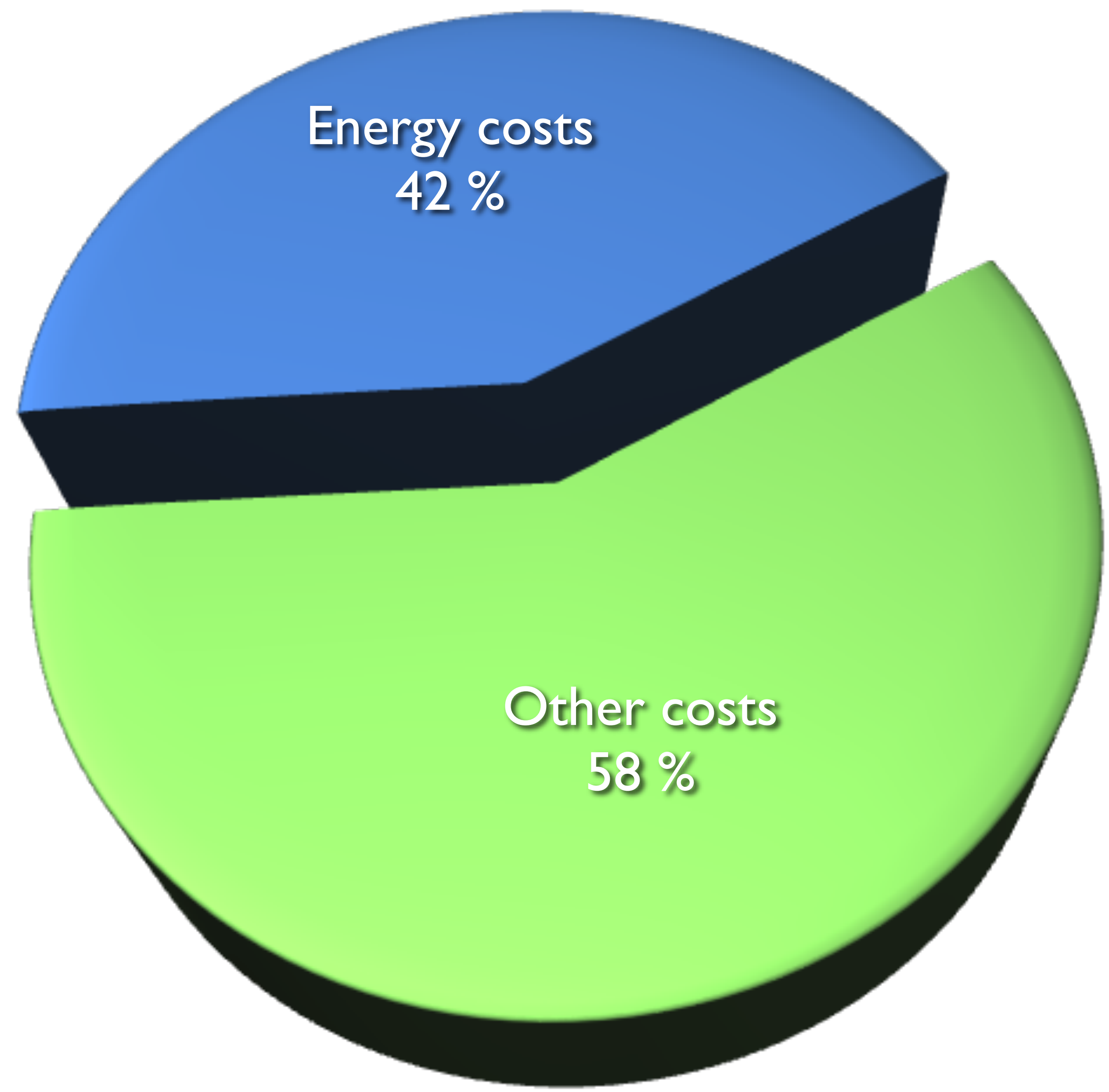
Introduction

Monthly costs of data center*:

All energy costs: 42 %

(incl. cooling infrastructure etc)

In 2011, servers will make up 3% of the total energy consumption in the U.S.



*: amortized

Introduction

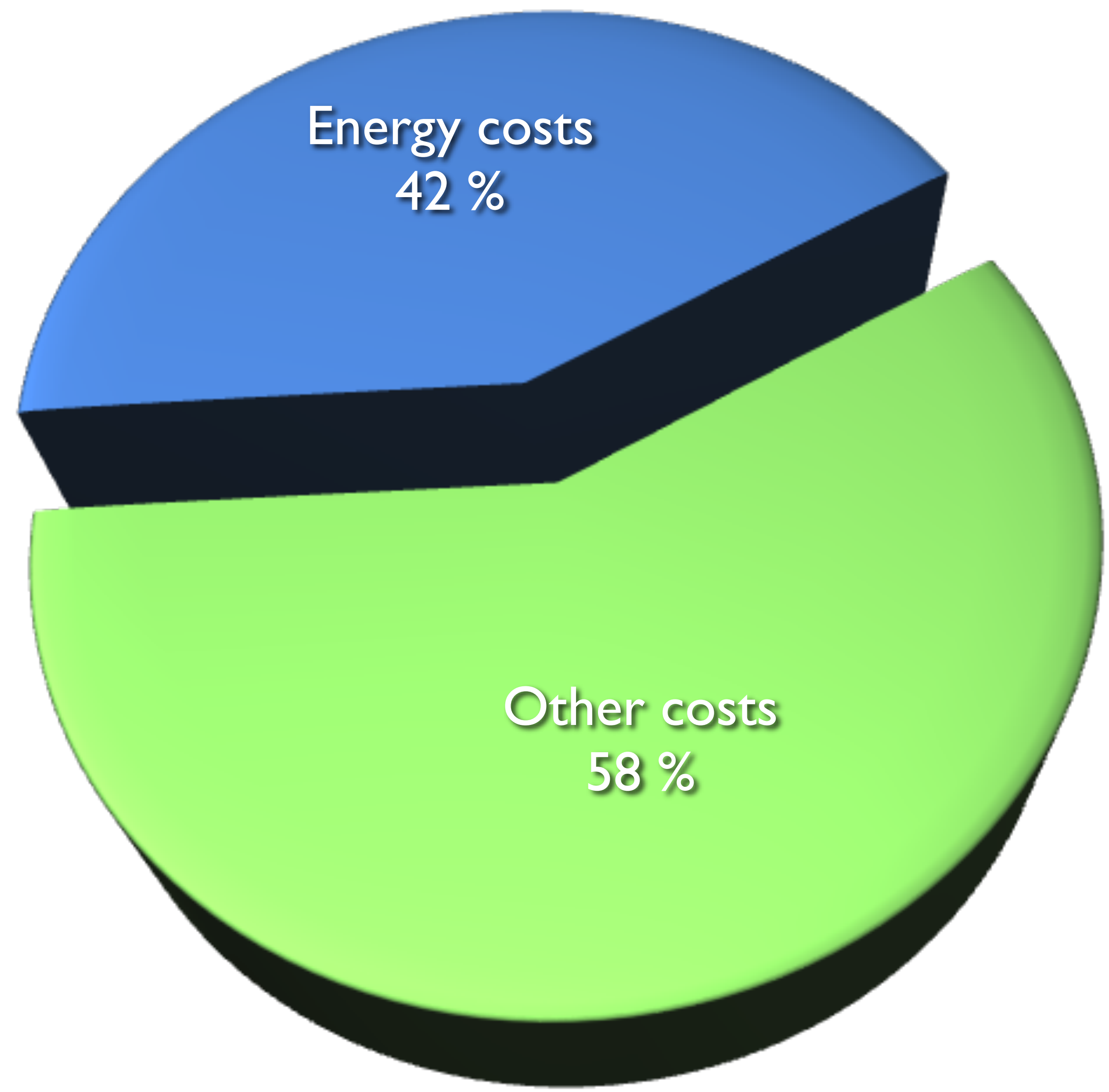
Monthly costs of data center*:

All energy costs: 42 %

(incl. cooling infrastructure etc)

In 2011, servers will make up 3% of the total energy consumption in the U.S.

Typically server node utilization: 20-30%



*: amortized

Outline

- Introduction
- Energy Management Framework
- Strategies:
    - Covering Set (CS)
    - All in Strategy (AIS)
- Evaluation
- Drawbacks of CS
- Related Work
- Conclusion

Energy Management Framework

- If system utilization drops → turn off nodes (and vice versa)
- Model to measure energy consumption:

$$E(\omega, \upsilon, \eta) = (P_{tr}T_{tr}) + (P_w^n + P_w^{\bar{n}})T_w + (P_{idle}^m + P_{idle}^{\bar{m}})T_{idle}$$

Energy Management Framework

- If system utilization drops → turn off nodes (and vice versa)
- Model to measure energy consumption:

$$E(\omega, \upsilon, \eta) = (P_{tr}T_{tr}) + (P_w^n + P_w^{\bar{n}})T_w + (P_{idle}^m + P_{idle}^{\bar{m}})T_{idle}$$

Energy = Power * Time

- η → hardware characteristics
- υ → time window
- ω → workload characteristics

Energy Management Framework

- If system utilization drops → turn off nodes (and vice versa)
- Model to measure energy consumption:

$$E(\omega, \upsilon, \eta) = (P_{tr}T_{tr}) + (P_w^n + P_w^{\bar{n}})T_w + (P_{idle}^m + P_{idle}^{\bar{m}})T_{idle}$$

↓

Energy for powering up/down nodes (transition)

Energy Management Framework

- If system utilization drops → turn off nodes (and vice versa)
- Model to measure energy consumption:

$$E(\omega, \upsilon, \eta) = (P_{tr}T_{tr}) + (P_w^n + P_w^{\bar{n}})T_w + (P_{idle}^m + P_{idle}^{\bar{m}})T_{idle}$$

Energy for running the workload
(power of online and offline nodes)

Energy Management Framework

- If system utilization drops → turn off nodes (and vice versa)
- Model to measure energy consumption:

$$E(\omega, \upsilon, \eta) = (P_{tr}T_{tr}) + (P_w^n + P_w^{\bar{n}})T_w + (P_{idle}^m + P_{idle}^{\bar{m}})T_{idle}$$

if time is left: Energy in idle mode
(power of online and offline nodes)

Energy Management Framework

- If system utilization drops → turn off nodes (and vice versa)
- Model to measure energy consumption:

$$E(\omega, \upsilon, \eta) = (P_{tr}T_{tr}) + (P_w^n + P_w^{\bar{n}})T_w + (P_{idle}^m + P_{idle}^{\bar{m}})T_{idle}$$

Outline

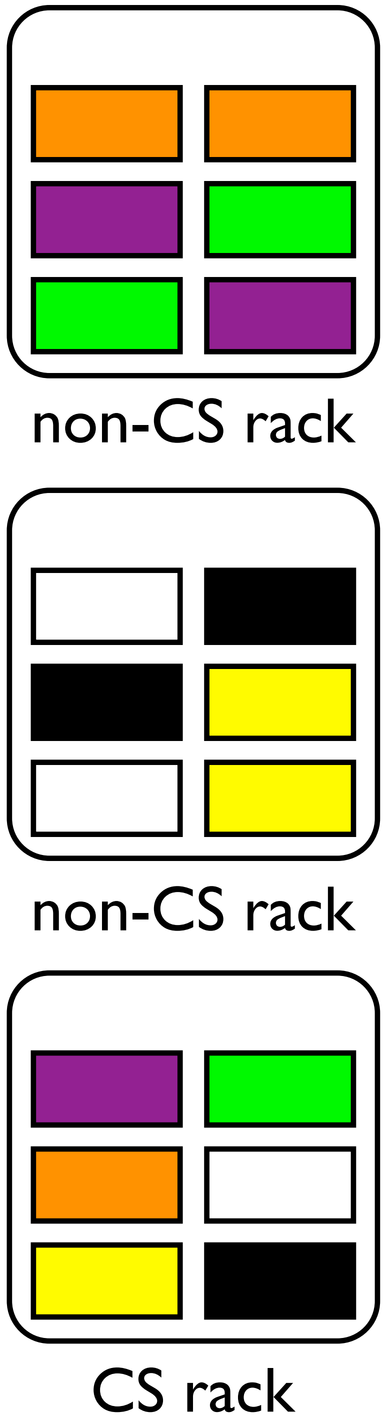
- Introduction
- Energy Management Framework
- Strategies:
  - Covering Set (CS)
  - All in Strategy (AIS)
- Evaluation
- Drawbacks of CS
- Related Work
- Conclusion

Covering Set (CS)

- Recently proposed for cluster energy management
- Power down some nodes (reduce idle energy)
- All data must be available:
    - data replication
    - one node must be active ($\Rightarrow$ CS node)

Covering Set (CS)

- HDFS default: triple replication
- Assume 3 racks:
  - one replica on the same rack
  - one replica on another rack
- designate one rack as Covering Set
- CS rack hold one copy of every data block



Power Down Strategies

- Random Power Down
- Load Balanced Power Down
- Round-Robin Random Power Down

Power Down Strategies

<u>Random Power Down</u>

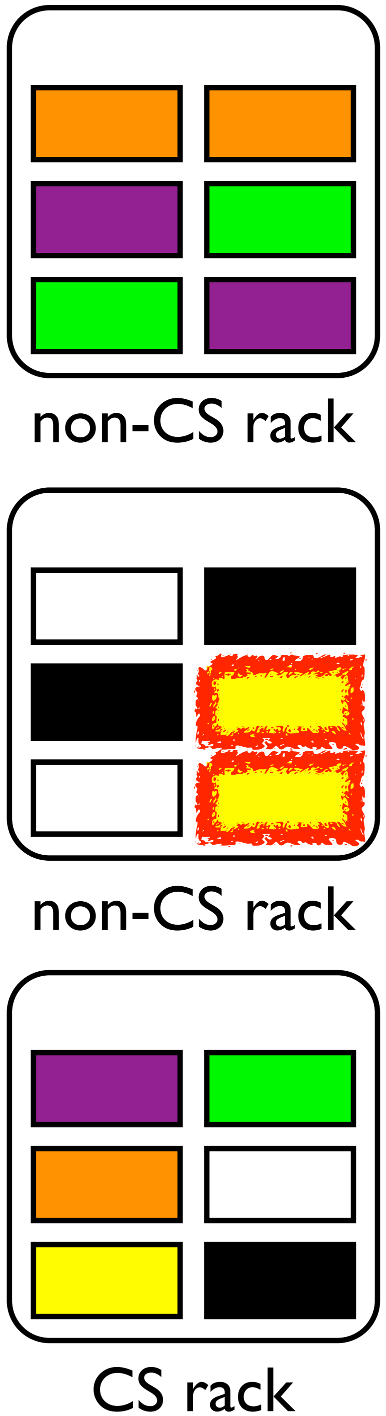
select a node at random and power down

⇒ second node with data coud be selected

⇒ CS-node is the only one with that data

⇒ data must be catched from CS node

⇒ network traffic (bottleneck)

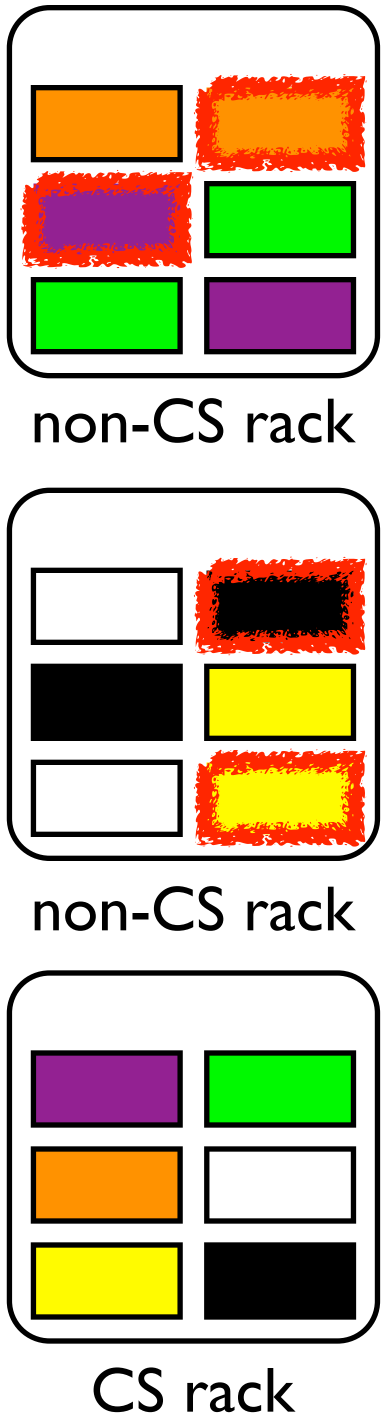


Power Down Strategies

<u>Load Balanced Power Down</u>

1. iterate over all nodes
2. compute all expected node-loads
3. save maximum expected node-load
4. shut down the smallest

⇒ expensive, but load-balanced



Power Down Strategies

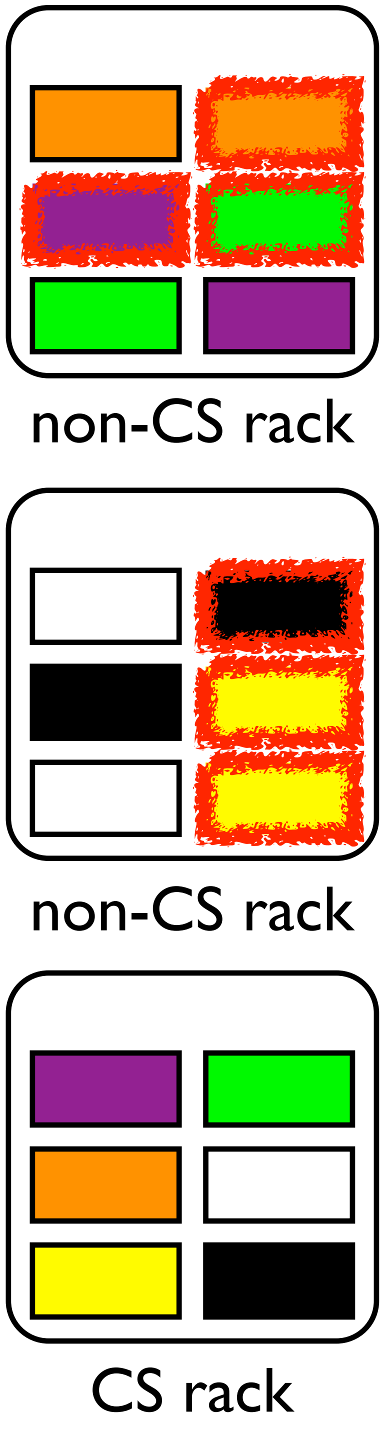
<u>Round-Robin Random Power Down</u>

select a node from the first rack

next selection → next rack

⇒ active nodes per rack is balanced

⇒ smaller probability of having no repication



All In Strategy (AIS)

- use all nodes to compute the workload
- power down all nodes afterwards
- no need to change distributed filesystem
- low utilization period:
  - batch jobs
  - periodically wake up and run the batch

Outline

- Introduction
- Energy Management Framework
- Strategies:
  - Covering Set (CS)
  - All in Strategy (AIS)
- Evaluation
- Drawbacks of CS
- Related Work
- Conclusion

Setup / Background

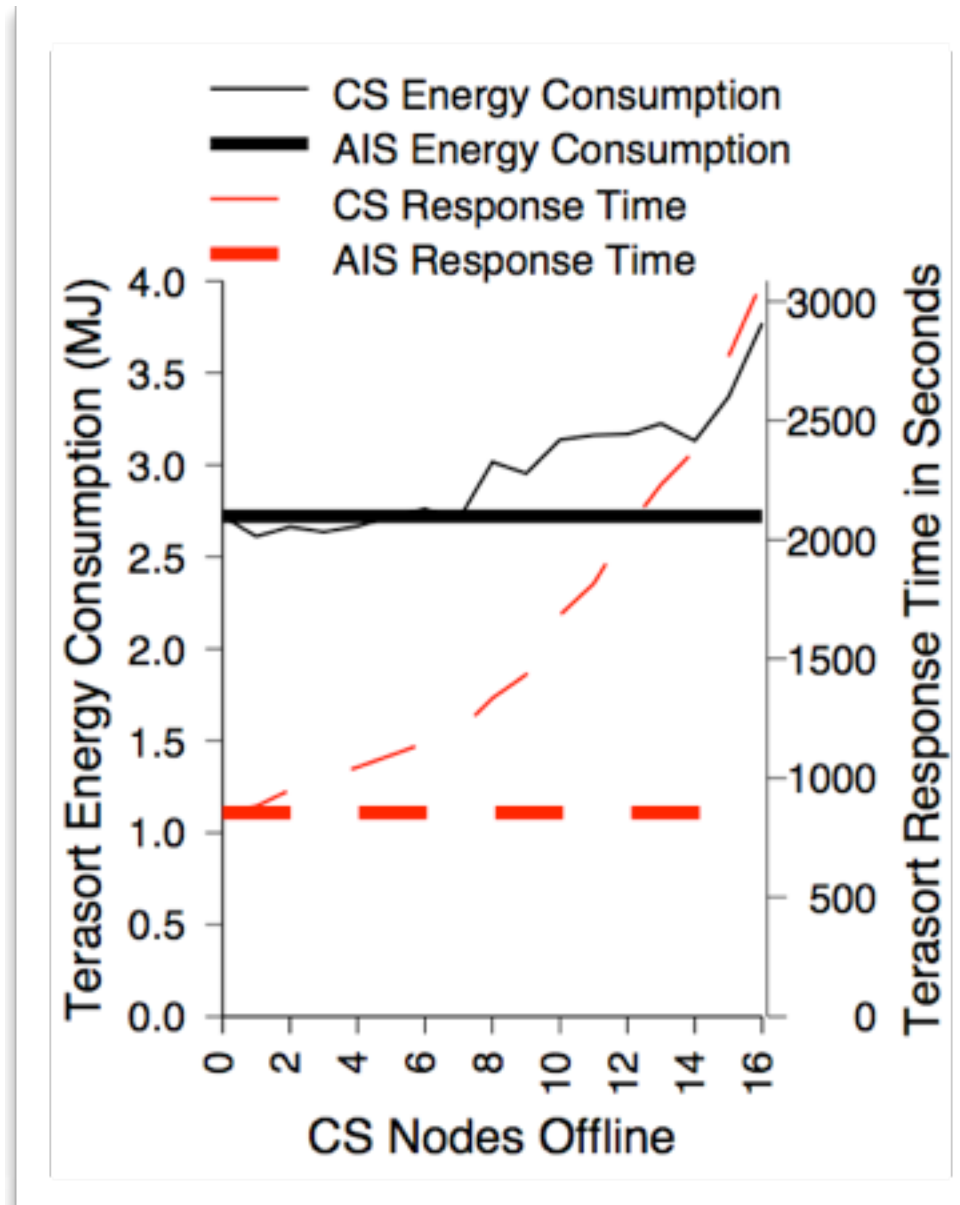
- 24 nodes (3 racks of 8 nodes)
  - 2.4 GHz Intel Core2Duo
  - 4GB RAM
  - 2x250 GB SATA-I
- Idle energy consumption:
  - Powered off (Hibernate): 10 W
  - Powered on (Stopgrant): 114 W

Workload-only Evaluation

- no idle time/energy
- system in desired state ⇒ no transition T/E
  - CS: desired number of nodes down
  - AIS: all nodes powered up

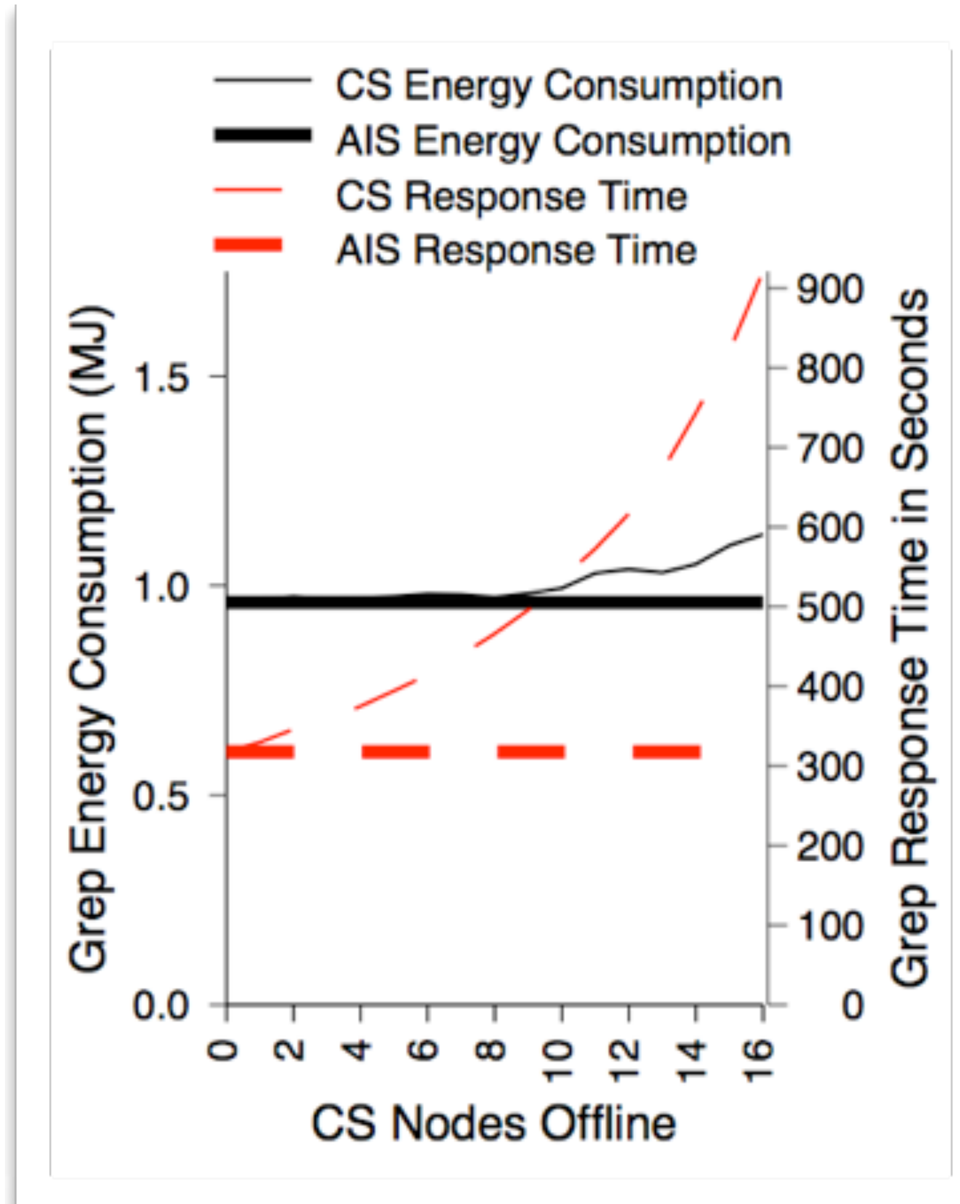
Workload-only Evaluation

- Terasort
- AIS = (CS 0 offline nodes)
- non-linear job⇒non-linear response time degradation
- all non-CS nodes offline: 39% more energy



Workload-only Evaluation

- Distributed Grep
- AIS = (CS 0 offline nodes)
- non-linear job⇒non-linear response time degradation
- all non-CS nodes offline: 17% more energy

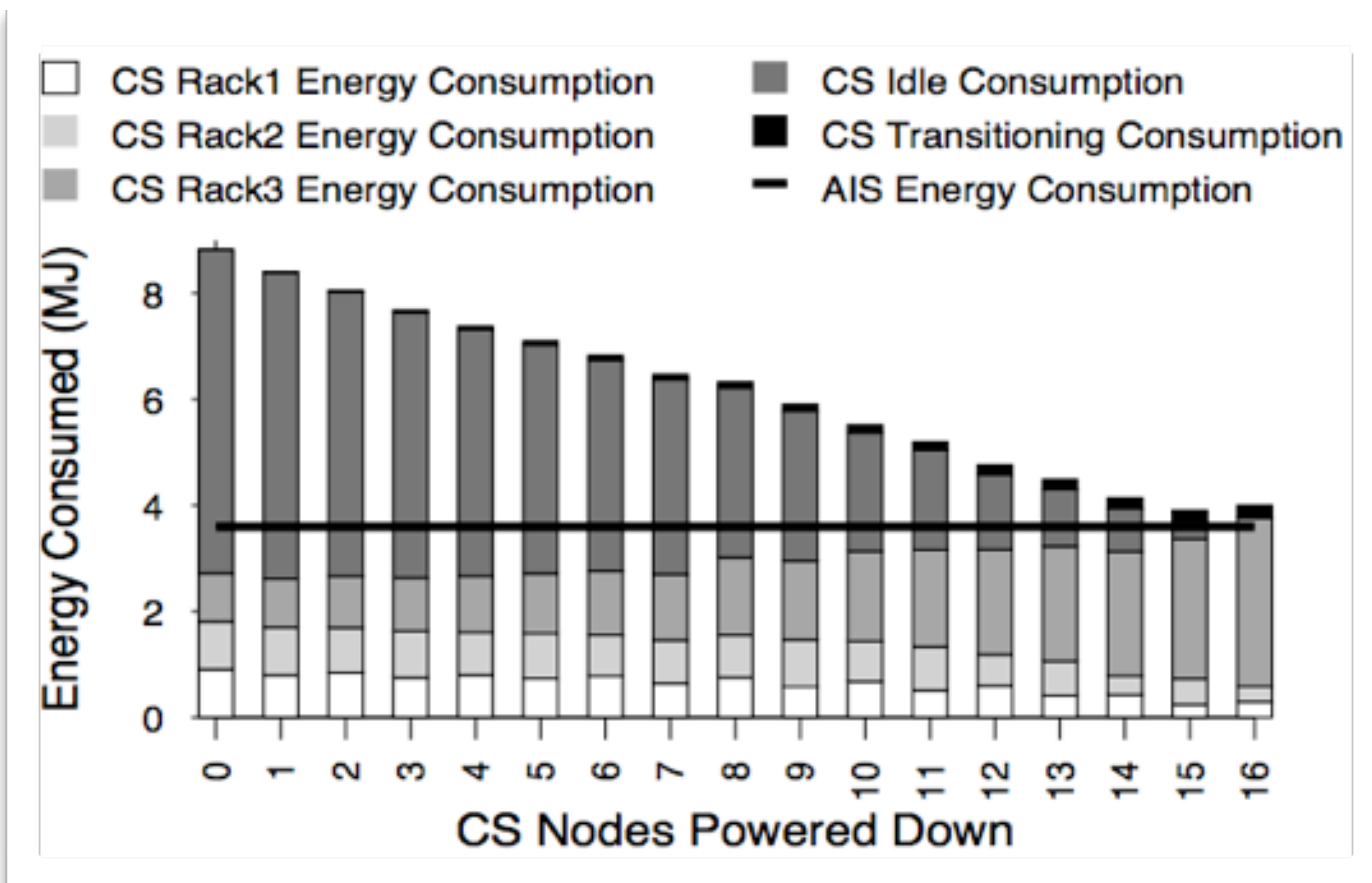


Workload with Idle Periods

Latency-sensitive Workloads

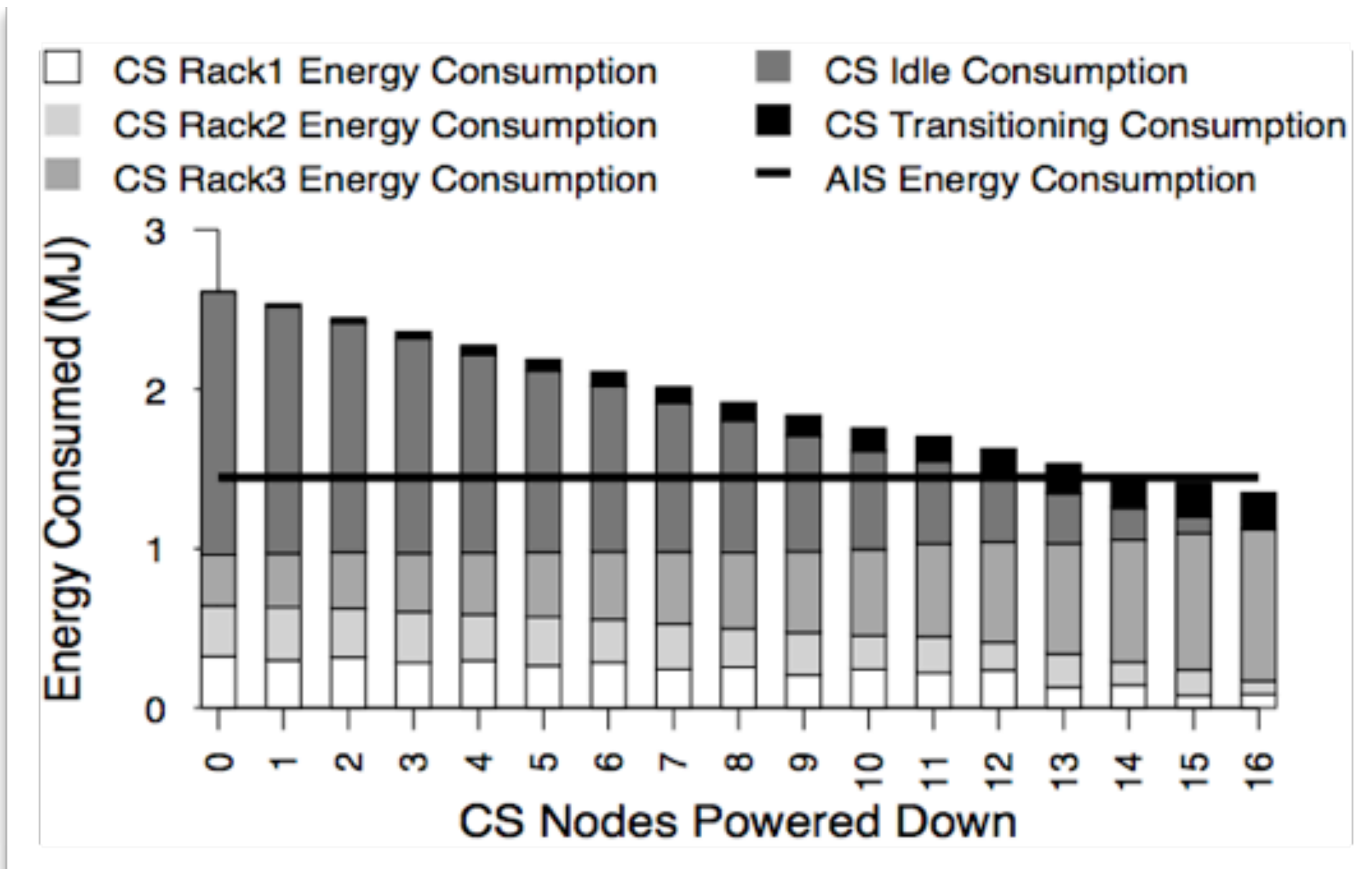
- Idle time/energy if time is left in window
- Initial and end state:
  - AIS: all nodes are powered down
  - CS: all nodes are powered up

Latency-sensitive Workloads



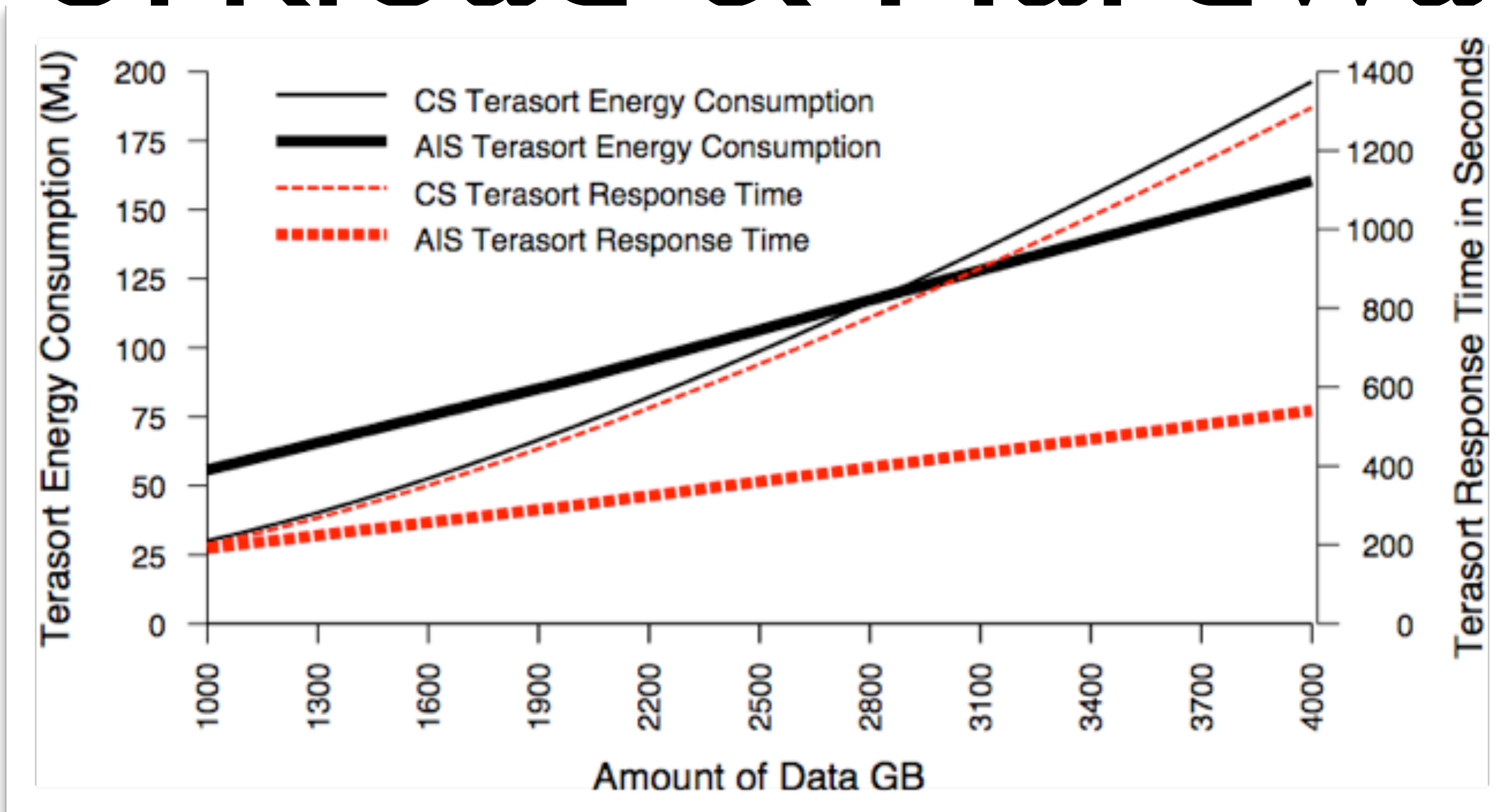
- Terasort
- time window: 3197s
  - power down: 11s
  - run (8 nodes): 3086s
  - power up: 100s

Latency-sensitive Workloads



- Distributed Grep
- time window: 1032s
  - power down: 11s
  - run (8 nodes): 921s
  - power up: 100s

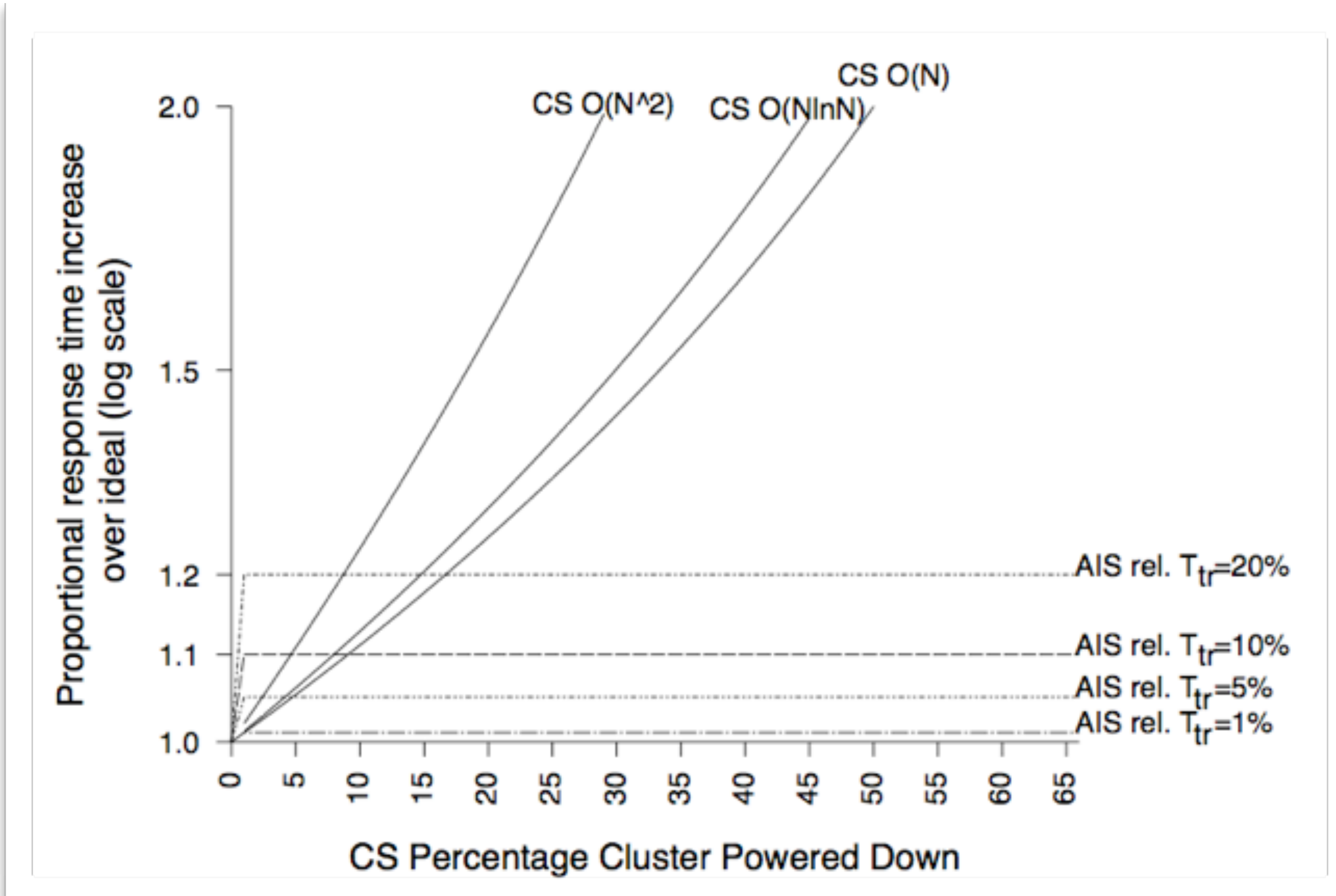
Effects of Workload & Hardware



Setup:
2000 nodes
CS: 50 %

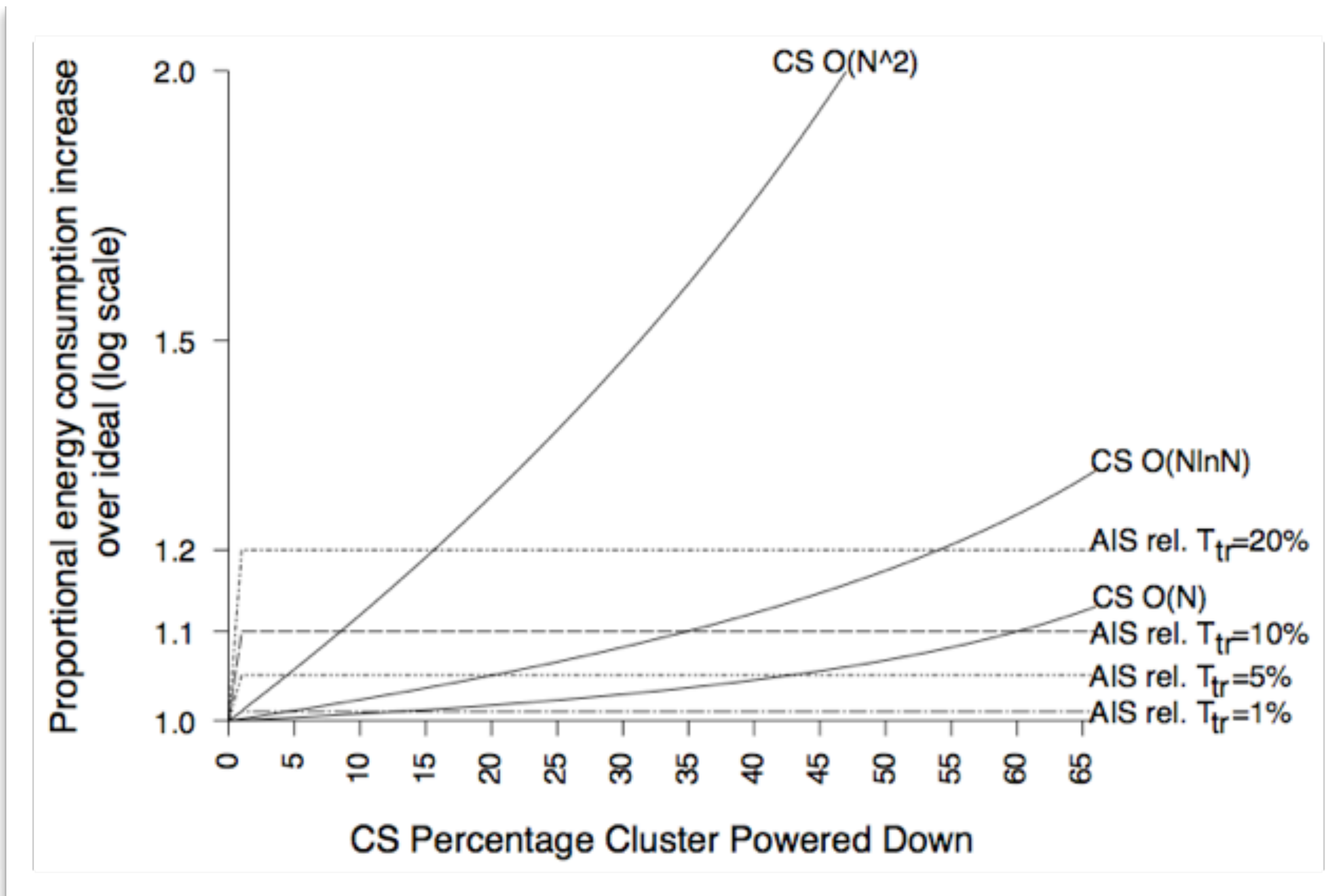
- CS: no transition cost, no idle cost
- AIS : no idle cost, full transition costs (111s)
- workload increase 2,8 TB (1,4 GB/node) ⇒ AIS is better

Effects of Workload & Hardware



- AIS has a better response time across almost all workloads

Effects of Workload & Hardware



- AIS need less energy for complex or hughe workloads

Effects of Workload & Hardware

Relative T_{tr}	$O(N)$	$O(NlnN)$	$O(N^2)$
1%	AIS	AIS	AIS
5%	CS/AIS	AIS	AIS
10%	CS	CS/AIS	AIS
20%	CS	CS	AIS

Outline

- Introduction
- Energy Management Framework
- Strategies:
  - Covering Set (CS)
  - All in Strategy (AIS)
- Evaluation
- Drawbacks of CS
- Related Work
- Conclusion

Drawbacks of CS

- need significant more storage:

  100 nodes (34 CS-nodes, 66 non-CS nodes)

  5 TB data, DFS with triple replication ⇒ 15 TB

  ⇒ 15 TB output ⇒ 30 TB

  30 TB/100 nodes= 300 GB/node

Drawbacks of CS

- assuming all non-cs nodes offline:

  5 TB input-data (10 TB on offline non-cs nodes)

  ⇒ 15 TB output ⇒ 20 TB

  20 TB / 34 CS-nodes ⇒ 600 GB/CS-node

Drawbacks of CS

- Update: all nodes with affected data must be active
- turning off nodes $\Rightarrow$ response time degradation
- distributed file system modification: complicated

Related Work

- speed-up transition time
- more efficient hardware (SSD/Flash memory, large arrays of cheap low-power processors (Atom))
- RAID-based system that can turn off disks
- optimized OS kernels that save energy in idle

Conclusion

- a lot of energy consumed by datacenters
- much of the energy unused
- 2 strategies to reduce this consumtion

References

- *W. Lang, J.M. Patel*: Energy Management for MapReduce Clusters, InVLDB `10
- http://www.google.com/corporate/green/clean-energy.html

Thank you!

Questions?