PACMan: Performance Aware Virtual Machine Consolidation

Alan Roytman (UCLA)
Joint work with:
Aman Kansal (Microsoft Research), **Sriram Govindan** (Microsoft),
Jie Liu (Microsoft Research), **Suman Nath** (Microsoft Research)

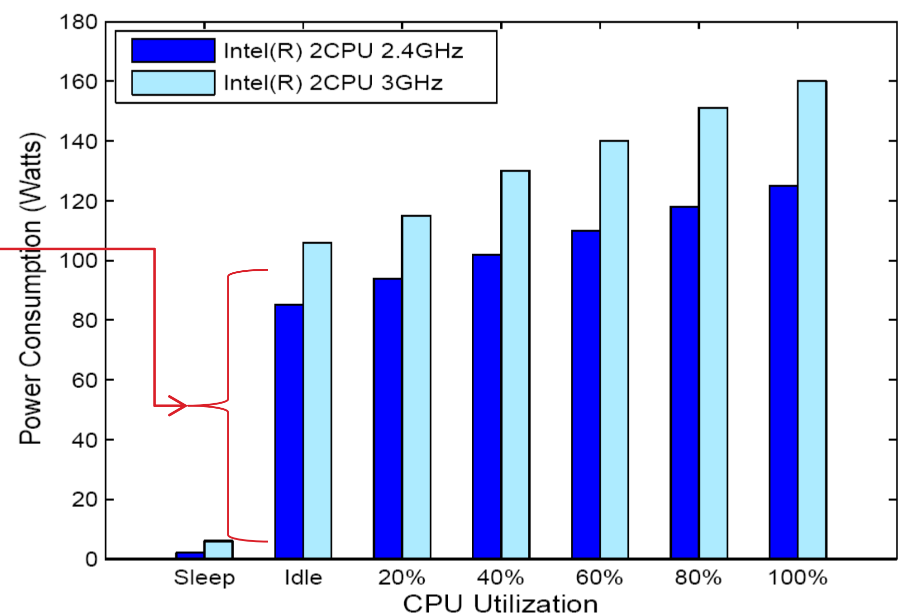
Motivation

- Server cost is the largest expense for data centers
- Data centers operate at very low utilization
  - Eg. Microsoft: over 34% servers at less than 5% utilization (daily average). US average 4%.
- VM Consolidation increases utilization, **decreases idling costs**

Idle power = 50 to 70%

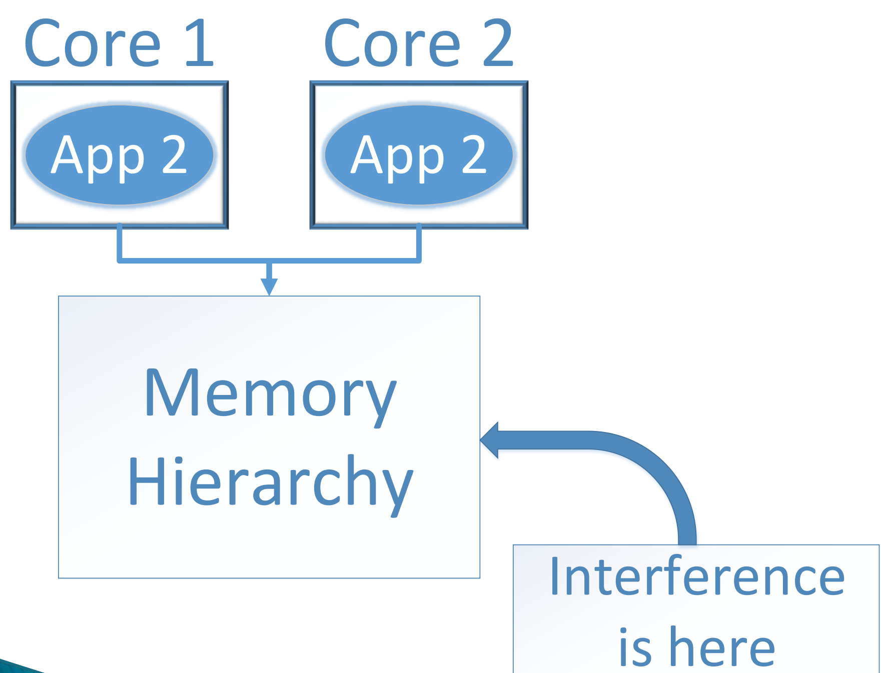
Adding more work to active server is more efficient

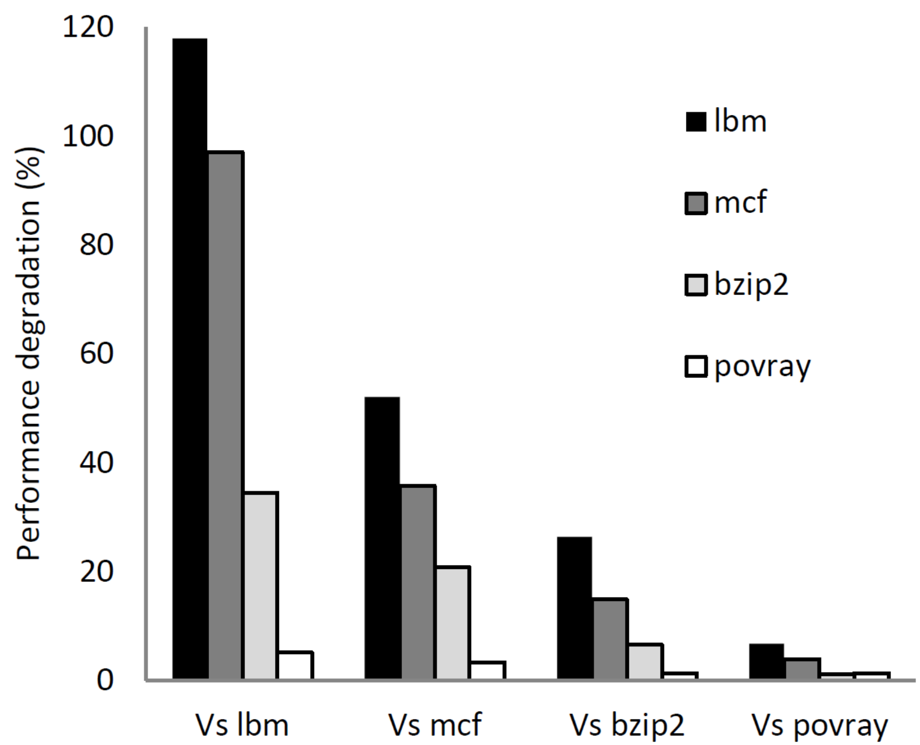
[Chen et al, NSDI' 08]



Motivation

- But VM consolidation **degrades performance** due to interference in the memory hierarchy
  - Interference occurs throughout memory hierarchy (e.g., multiple cores can share a cache)

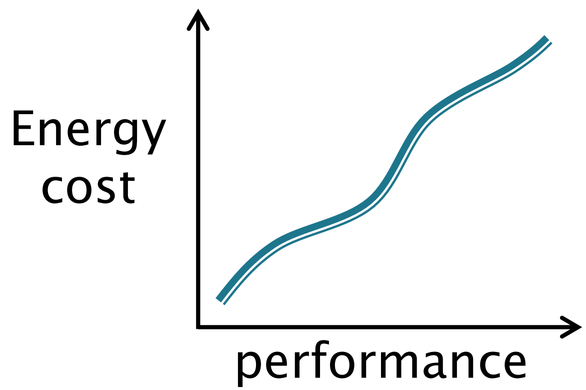




[Govindan–Liu–Kansal–Sivasubramaniam 2011]

Motivation

Goal: Consolidate intelligently to trade-off energy efficiency and performance

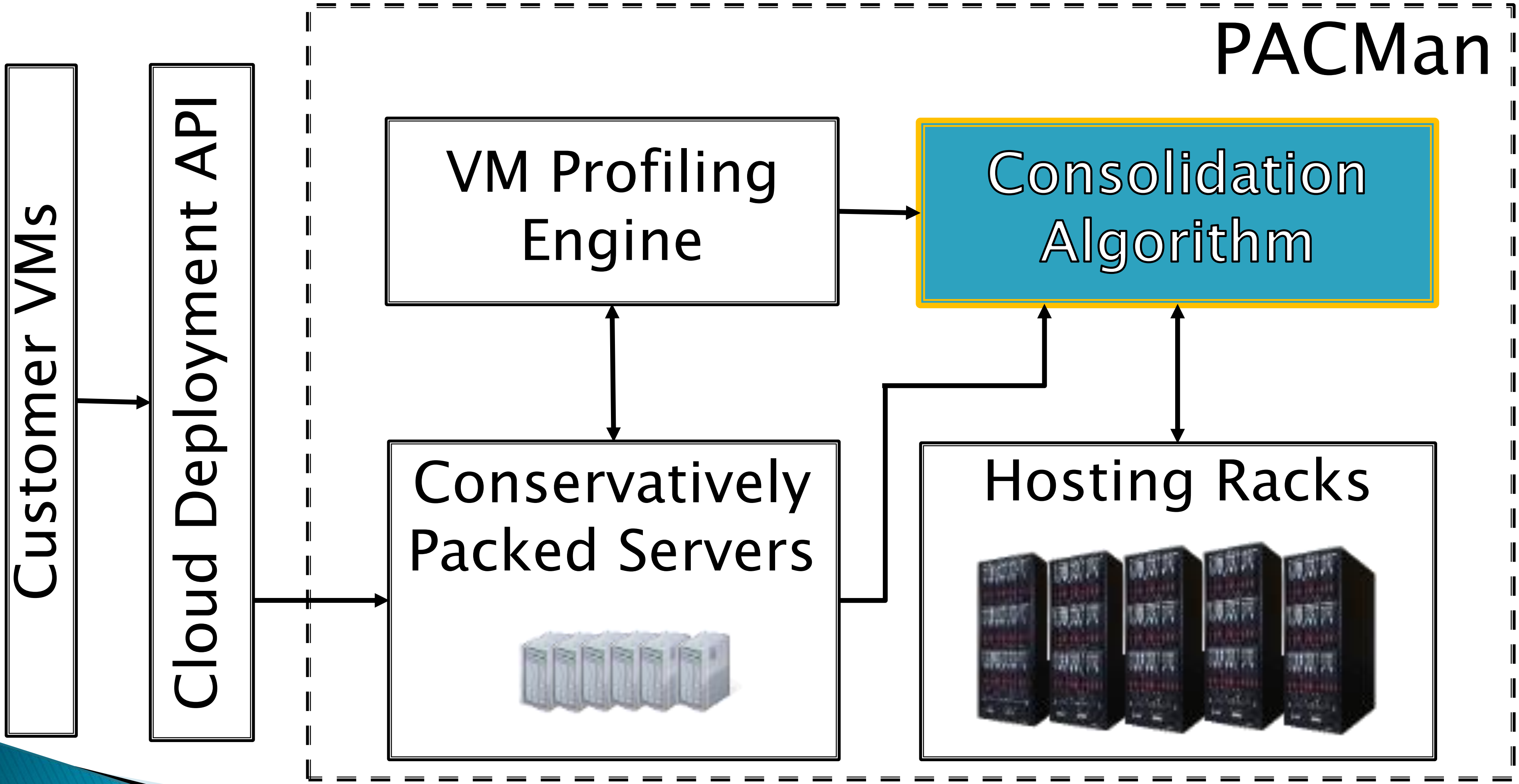


- How do we minimize resource cost while staying within a performance bound?
  - (e.g., minimize energy consumption or active machines)
- How do we maximize the worst case performance?
  - (e.g., Map-Reduce)

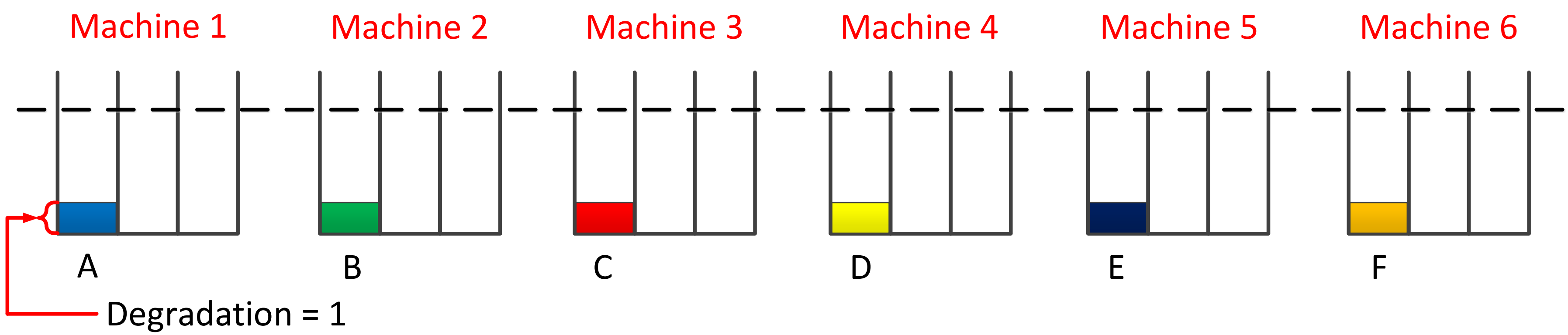
Talk Outline

- Introduction
- <u>P</u>erformance <u>A</u>ware <u>C</u>onsolidation <u>**Man**</u>ager
  - Performance-Mode: Minimize Energy Under Constraint
  - Energy-Mode: Minimize Maximum Degradation
- Experimental Results
- Conclusions and Future Work

Framework Focus

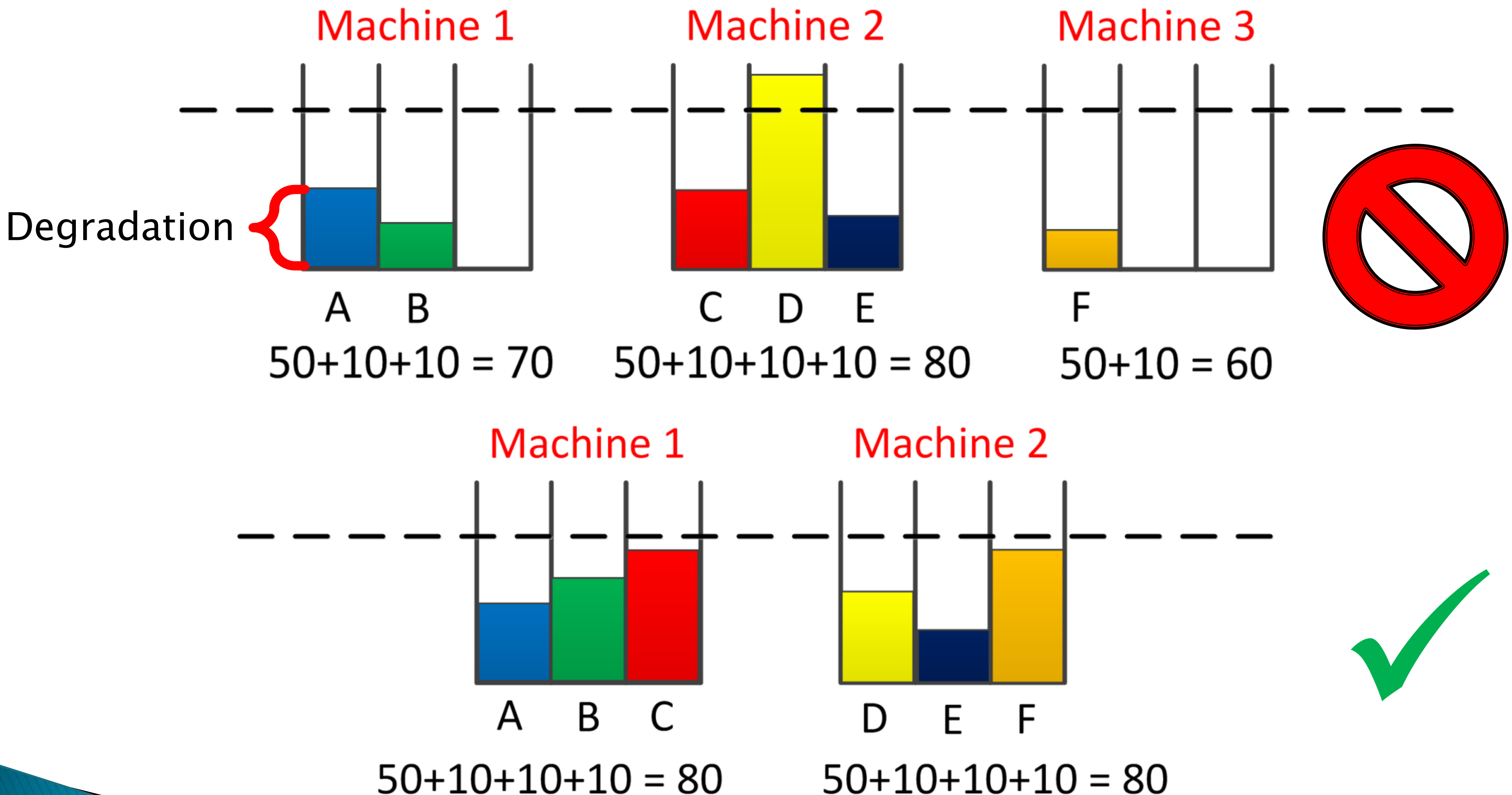


First Problem: Perf-Mode Example



Each machine incurs a cost of 50 for being active, plus 10 per VM assigned
Total cost of schedule = 6 * (50 + 10) = 360

First Problem: Perf-Mode Example



Perf-Mode Problem: Definition

Minimize Energy Under Performance Constraint

- We have n VMs, along with a degradation constraint $D \geq 1$, machines with k cores
- We are given feasible sets $|S| \leq k$ (all VMs experience degradation at most D)
- Each set S has a cost $w(S)$ (e.g., energy)
- Goal: $\min_{partitions} \sum_S w(S)$

Perf-Mode Problem: Outline

- We give a polynomial time optimal solution for the two-core case
- Bad news: for $k \geq 3$ cores, this problem is NP-Complete
- Good news: we design and analyze an approximation algorithm with approximation ratio $\alpha = H_k \approx \ln(k)$

We can solve it close to optimal!

Multi-Core Case

- This problem is approximable within a factor $\alpha = H_k = \sum_{i=1}^{k} \frac{1}{i} \approx \ln(k)$
- This means, for all inputs: $w(ALG) \leq H_k w(OPT)$
- Proof similar to the k–Set Cover Problem
- Need two assumptions:

Closure Under Subsets: S feasible implies any subset $T \subseteq S$ is feasible

Monotonicity: If $S \subseteq T$, then $w(S) \leq w(T)$

Approximation Algorithm

- First consider the case when all costs are 1 (minimizing cost = minimizing # machines)

Algorithm:

- Sort sets (ascending order) according to $\frac{1}{|S|}$
- Greedily pick disjoint sets going down the list

Algorithm Example

Suppose there are $n = 5$ VMs and $k = 3$ cores

S	{A,B}	{A,C}	{B,C}	{A,B,C}	{D,E}	{A}	{B}	{C}	{A,B,D}	{A,B,E}
$\frac{1}{\lvert S\rvert}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{2}$	1	1	1	X	X

…

Sorted order:

{A,B,C} {A,B} {A,C} {B,C} {D,E} {A} {B} {C}

✓ 🚫 🚫 🚫 ✓

Solution uses two machines

Analysis

- The proof generalizes to the case when the costs of sets can be arbitrary!
  - e.g., $w(S) = \ c_f + \sum_{j \in S} d_j^S, \quad w(S) = \max_{j \in S} d_j^S$

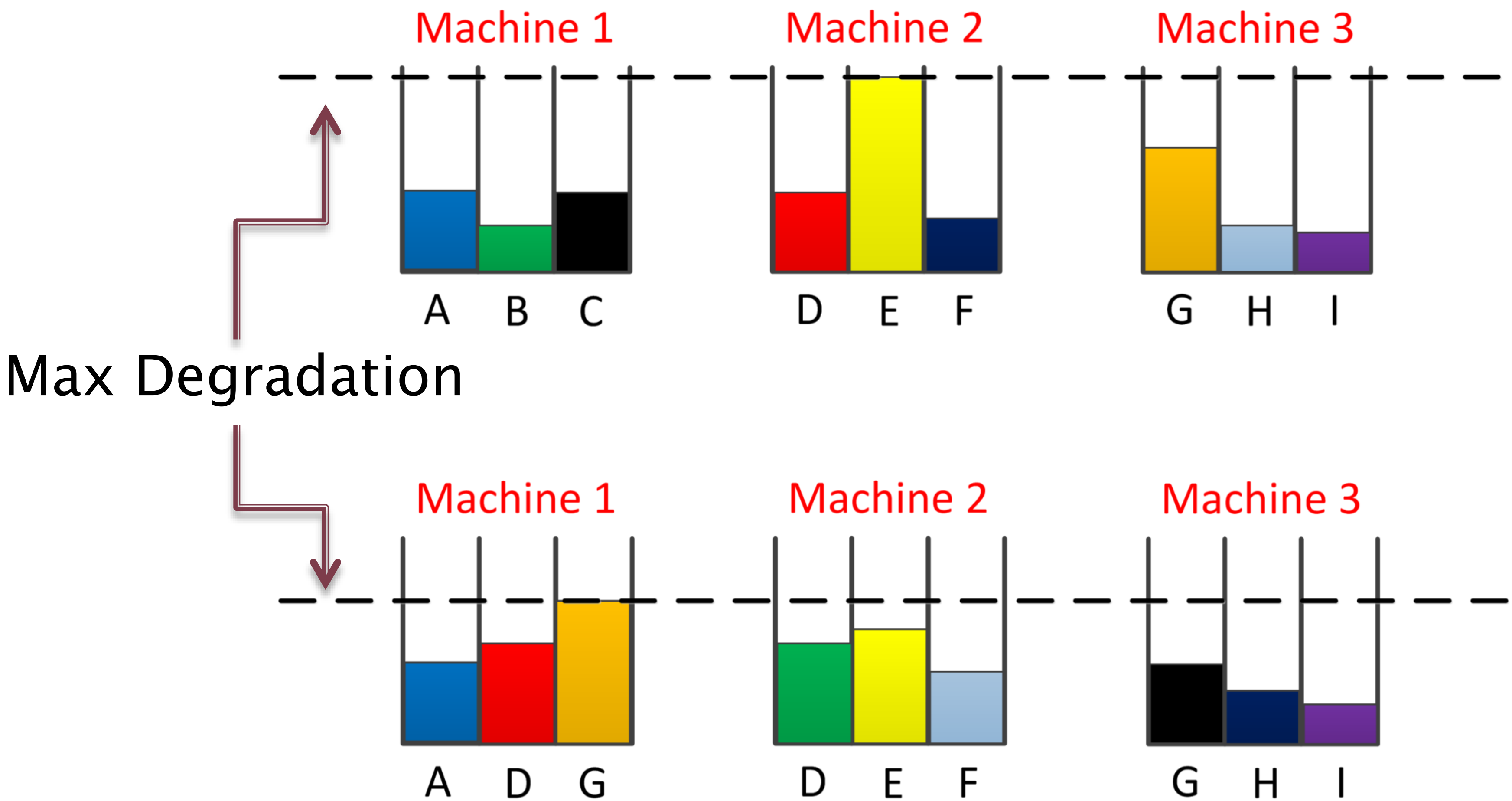
New Algorithm:

- Sort sets (ascending order) according to $\frac{w(S)}{|S|}$
- Greedily pick disjoint sets going down the list

Perf-Mode: Take-Away

- We can solve the two-core case optimally and efficiently
- For more cores, the problem is NP-Complete
- We give an asymptotically tight approximation algorithm with $\alpha \approx \ln(k)$
- The algorithm is greedy and easy to implement

Second Problem: Energy-Mode Example



Energy-Mode Problem: Definition

Minimizing Maximum Degradation

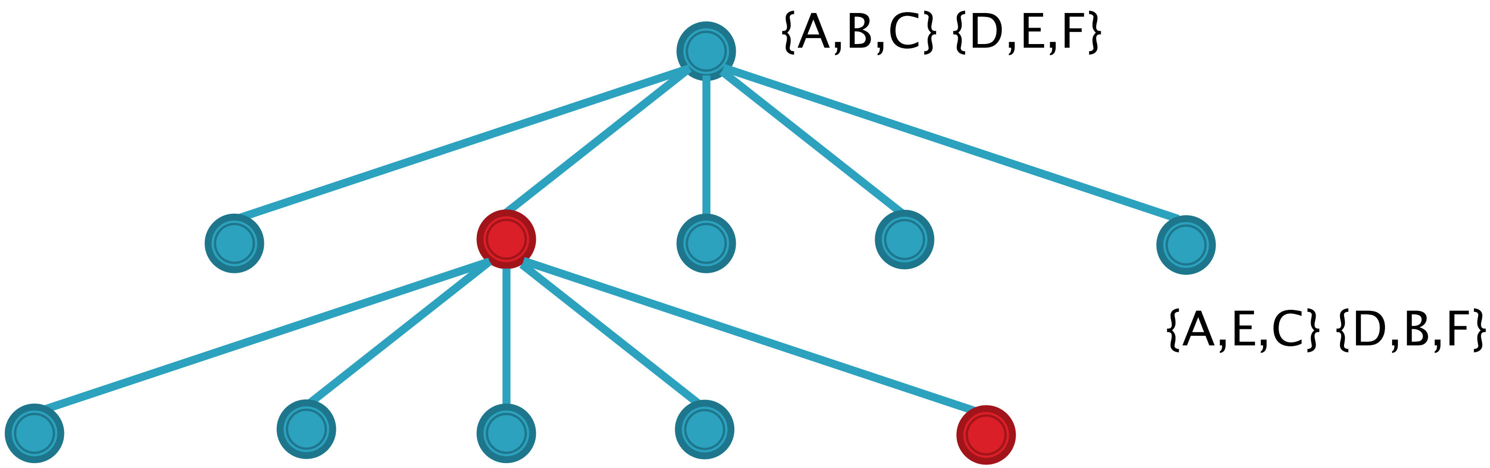
- Input is similar to before: n VMs, m machines, k cores
- For a set B of VMs, VM $j \in B$ experiences degradation $d_j^B \geq 1$
- New Objective Function:
- Goal: Minimize $\max_{1 \leq i \leq m} \max_{j \in S_i} d_j^{S_i}$ (S_i is the set of VMs on server i)

Energy-Mode: Outline

- For two cores, the problem is polynomial-time solvable
- We give an inapproximability result for this problem
- We give heuristics since the problem is provably difficult to approximate

Heuristic Algorithm

- We implement a greedy heuristic:
  - Start from an arbitrary initial schedule
  - For all ways of swapping VMs, go to the schedule with smallest sum of maximum degradations
  - We set number of swaps to be $G = (k-1) \cdot (m-1)$

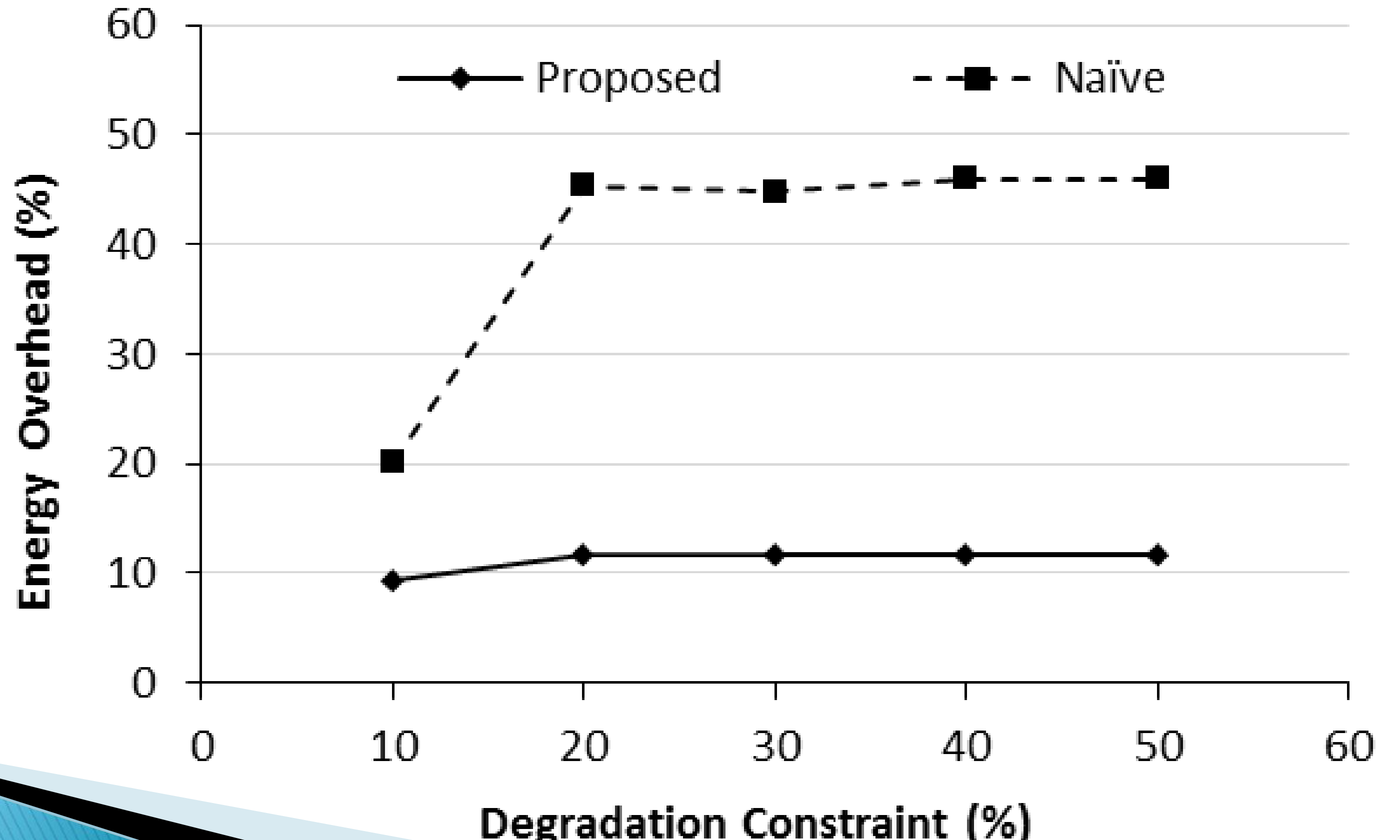


Experimental Setup

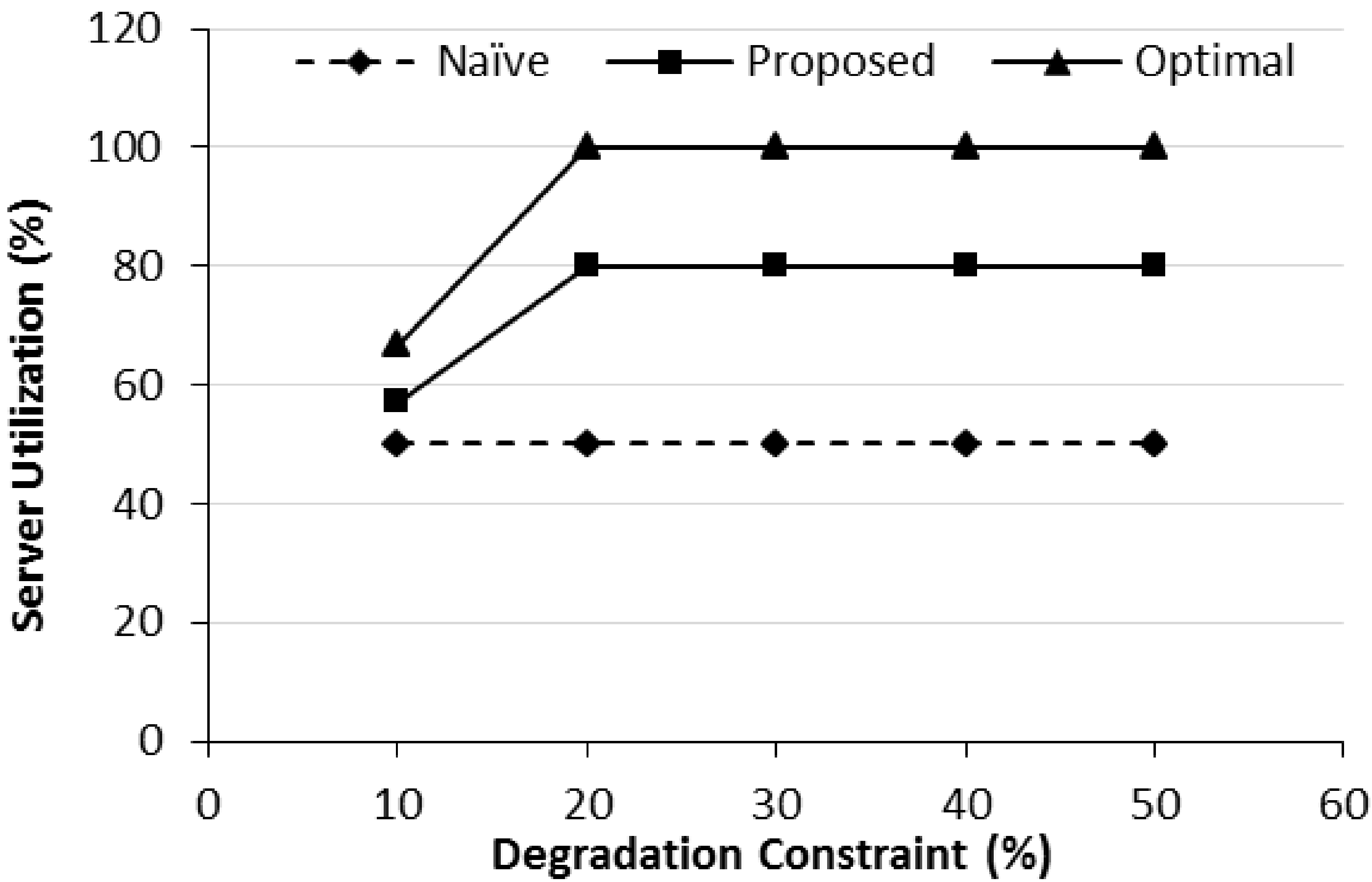
- Small inputs:
  - $n = 16$ VMs, on servers with $\mathrm{k} = 4$ cores
  - Can compute optimal solution for small instances
- Large inputs:
  - Up to $n = 1000$ VMs, on servers with $k = 4$ cores
  - Compare solutions against a lower bound
- Use real-world degradations with SPEC CPU 2006 applications (lbm, soplex, povray, sjeng)

Experiments: Perf-Mode (Small Inputs)

- We use costs $w(S) = c_f + \sum_{j \in S} d_j^S$, where $c_f = 4$
- Comparison against OPT
- Naïve leaves every other core empty, which is the current practice [Mars-Tang-Hundt-Skadron-Soffa 2011]

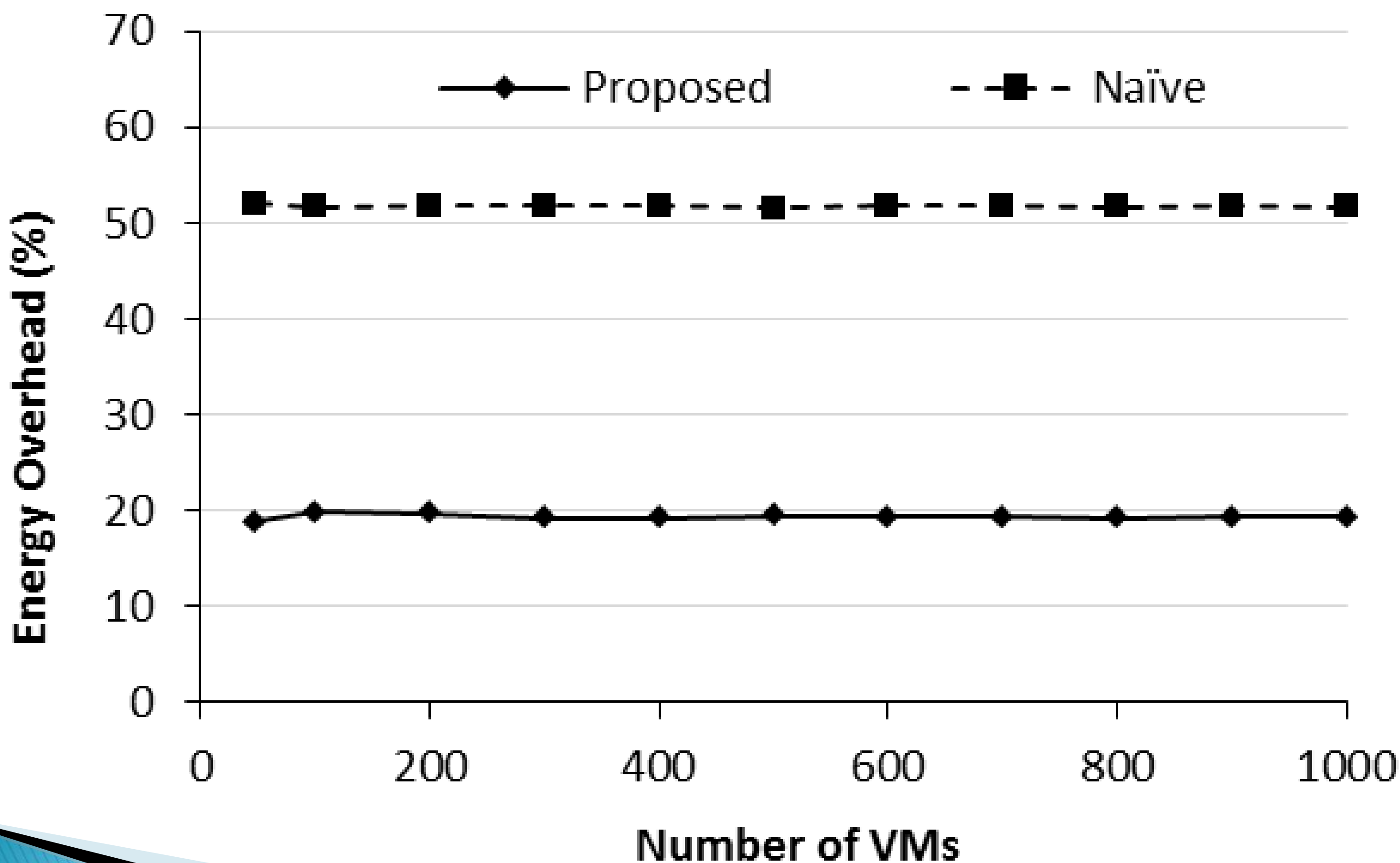


Experiments: Perf–Mode (Core Use)



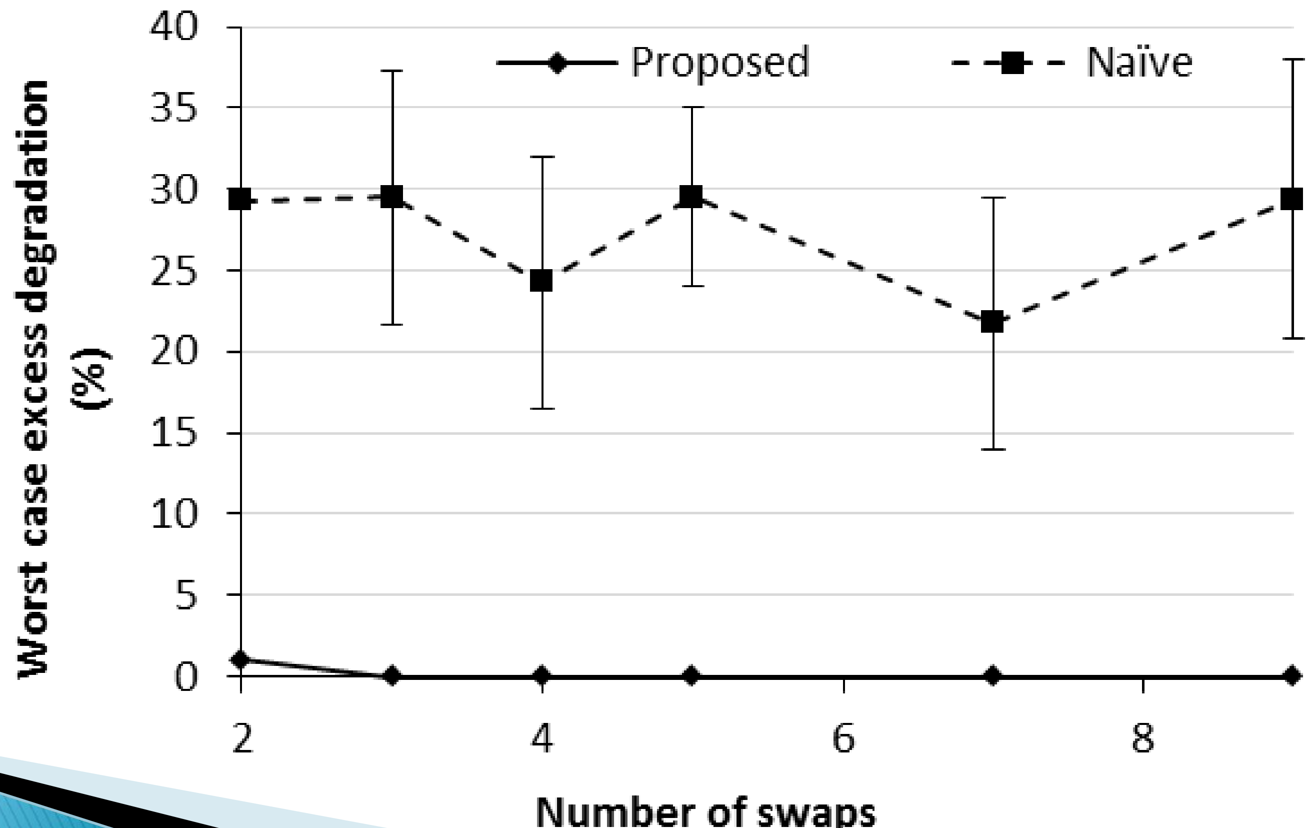
Experiments: Perf-Mode (Large Inputs)

- Comparison against lower bound



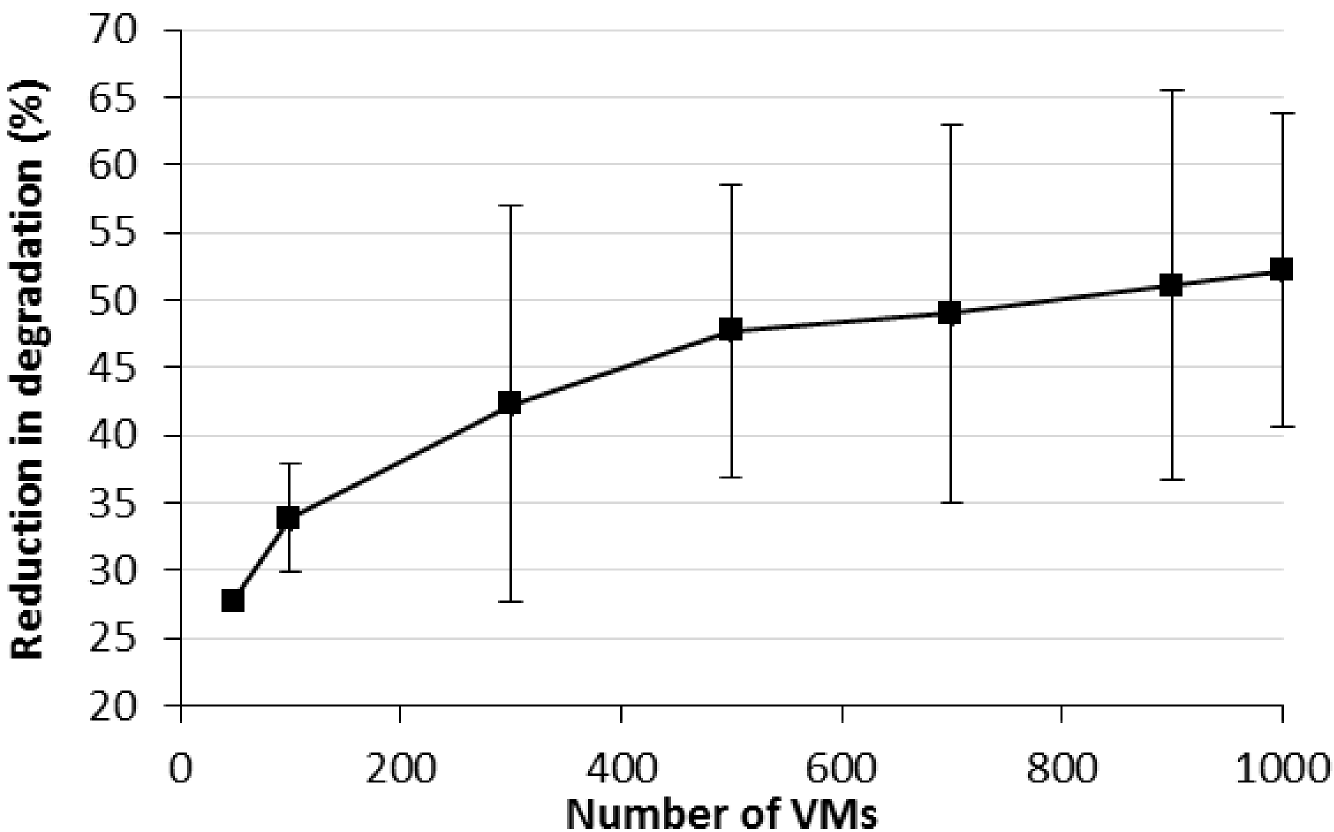
Experiments: Energy-Mode (Small Inputs)

- Comparison against OPT
- Up to $G = (k-1) \cdot (m-1) = 9$ swaps
- Naïve solution randomly places VMs, error bars show standard deviation for 10 runs



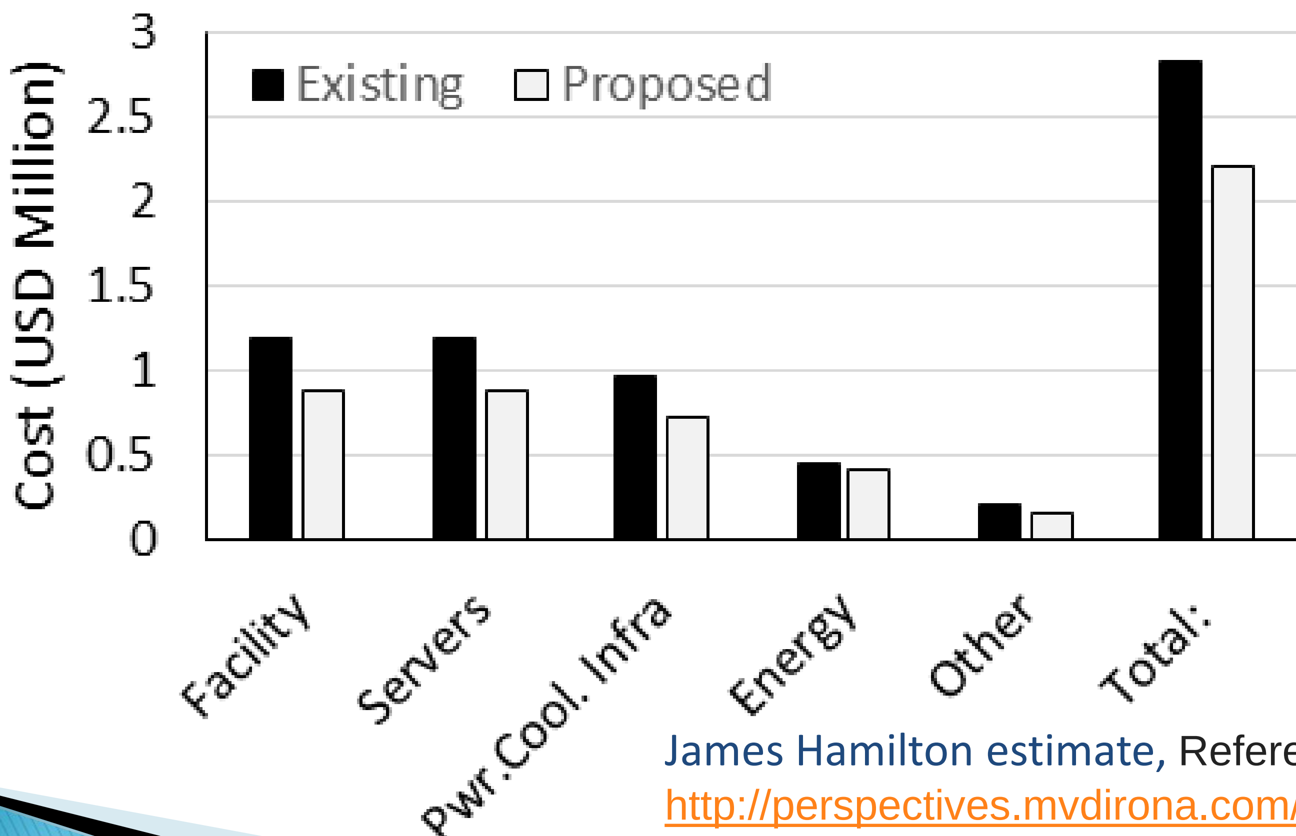
Experiments: Energy-Mode (Large Inputs)

- Reduction in degradation relative to naïve solution
- Up to 1000 VMs



Total Cost of Ownership (TCO)

- Amortized cost calculation for data centers
- 22% reduction in costs when comparing Performance-Mode algorithm to current practice
- For 10MW data centers, costs are reduced from $2.8M to $2.2M per month (costs are related to energy expenditure)



James Hamilton estimate, Reference: http://perspectives.mvdirona.com/content/binary/OverallDataCenterCostAmortization.xlsx

Related Work

- [Jiang–Shen–Chen–Tripathi 2008]
  - Consider minimizing sum of degradations
  - 2–core case is poly–time solvable
  - k–core is NP–Complete for $k \geq 3$ (give heuristics)
- [Tian–Jiang–Shen 2009]
  - Consider different length tasks, allow migrations
- [Jiang–Tian–Shen 2010]
  - Proactive co–scheduling, heuristic runtime scheduler

Conclusion

- Give a provably near-optimal algorithm such that resource waste is minimized
- Consider new objectives for the VM consolidation problem: Performance-Mode and Energy-Mode
- Important for energy minimization to consider cache interference
- Even small percentage improvement can have huge practical impact

Future Work

- Energy-Mode: consider variable number of swaps while incurring cost for each swap
- Consider online versions of all variants
- Perform more experiments on real data centers

Thank You!

Questions?