MANAGING ELECTRIC BILL AND PERFORMANCE IN SERVER FARMS

Michele Mazzucco[1] *(and Dmytro Dyachuk*[2]*)*
[1] *University of Tartu, Estonia*
[2]*University of Saskatchewan, Canada*

Introduction (1)

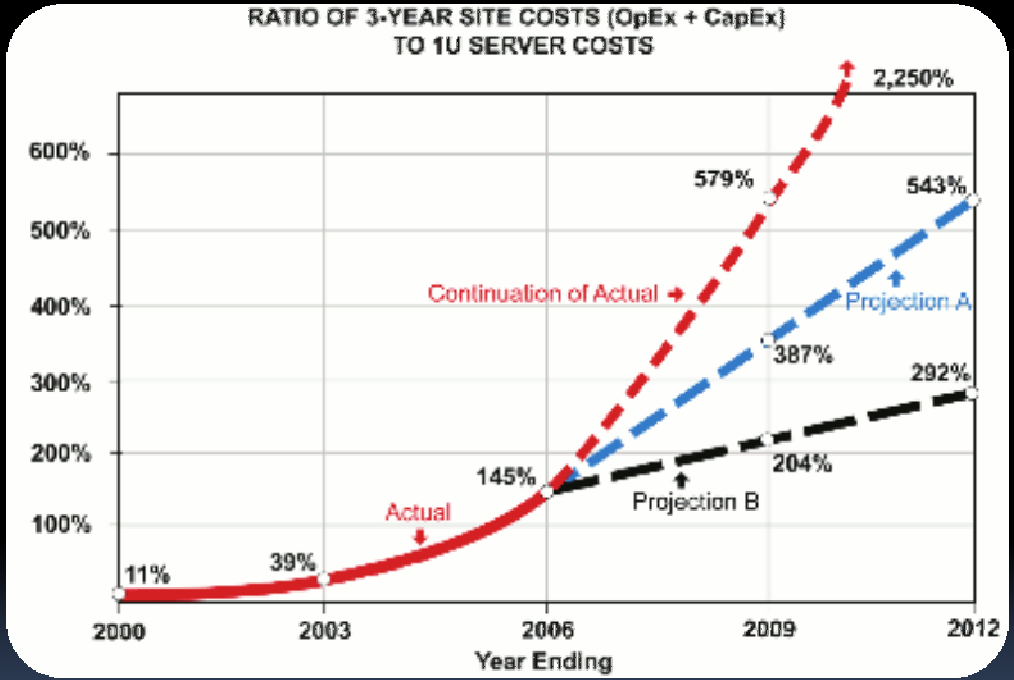
- Clients have performance expectations
- Load changes over the time
- **QoS is one of the critical factors determining the success (or failure) of service providers**
  - **E.g., Google reports that an extra 0.5 sec. in search page generation would imply a 20% traffic drop**
  - **E.g., trimming the page size of Google Maps by 30% resulted in 30% traffic increase**

Introduction (2)

- Electricity bill constitutes a significant fraction of the datacenter cost
  - A typical household consumes 1.14 kW
  - A typical data center consumes 15 MW = 877 households
- Different techniques exist for reducing energy consumption, e.g., voltage scaling
  - Idle servers consume up to 65% of their peak consumption
- Hence, we need to improve data center's utilization, i.e., by tearing down unused servers
  - How many servers to run?
    - Over-provisioning should be avoided
    - Service providers **must** meet performance/availability requirements

Introduction (3)

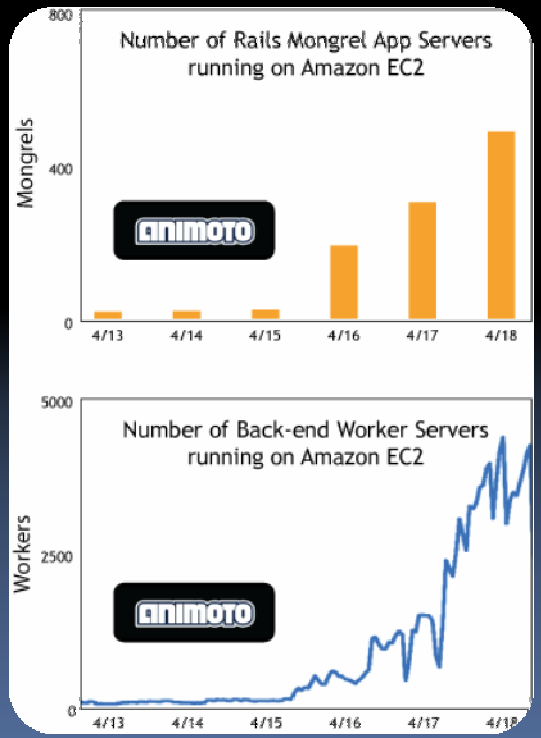
- Energy prices keep rising!

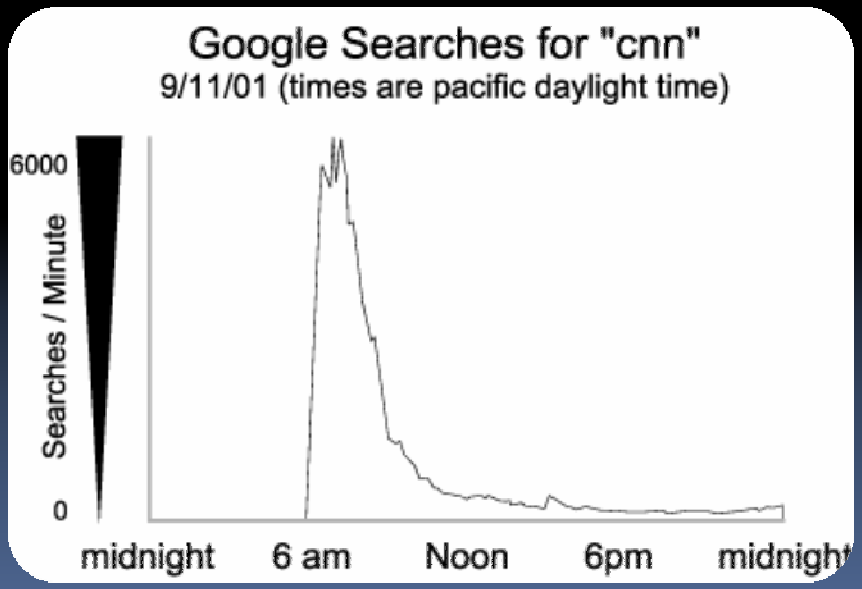


- **ENERGY EFFICIENCY = HIGHER PROFIT !**
  - Over provisioning should be avoided

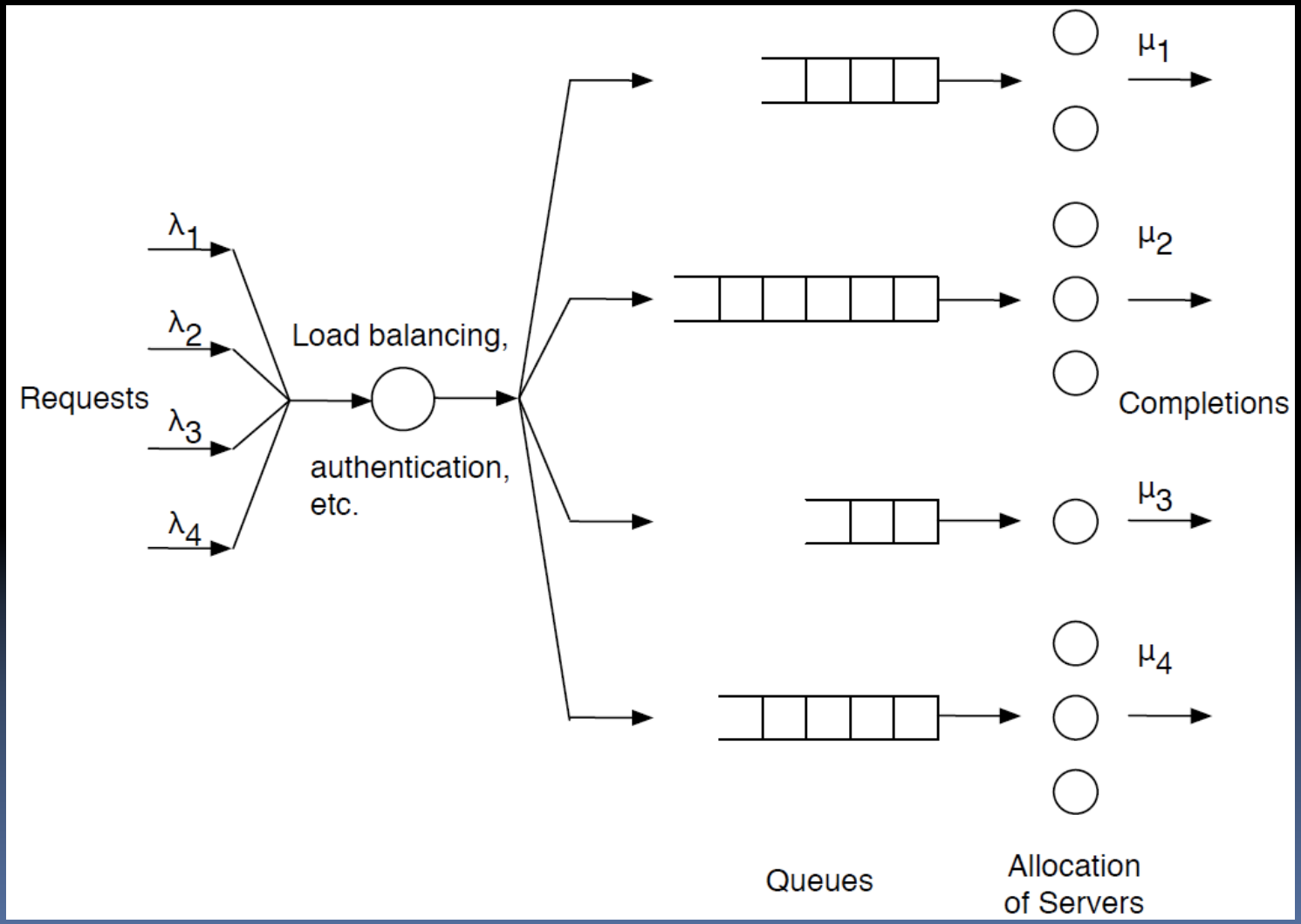
Problem definition

- Suppose we have a collection of servers
  - How to allocate them in the most efficient way?
  - User demand is very unpredictable
    - Each successful "transaction" brings c$: we don't want to miss business opportunities



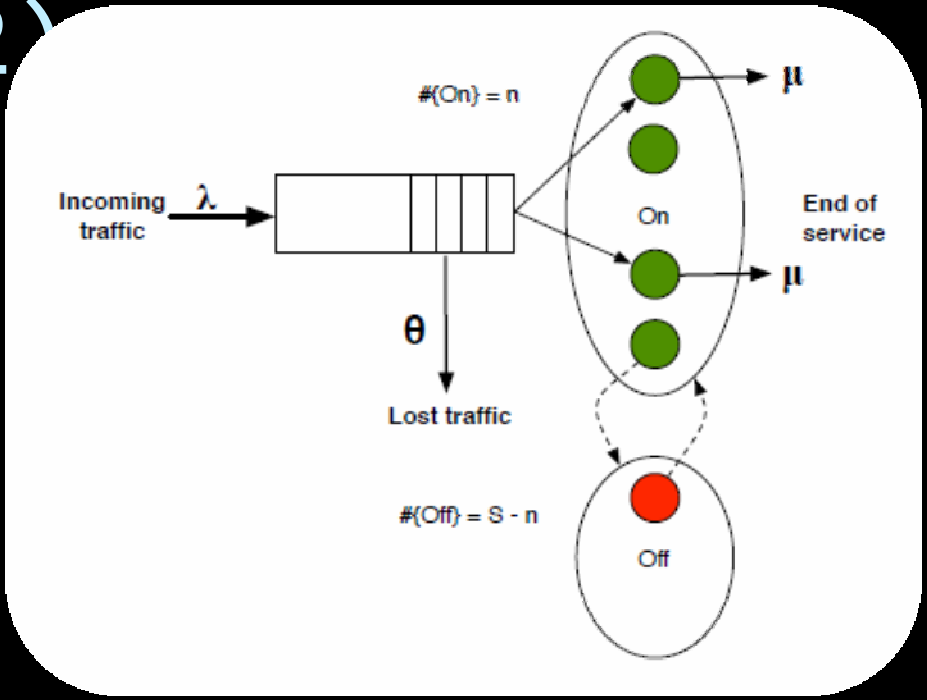


System model (1)



System model (2)

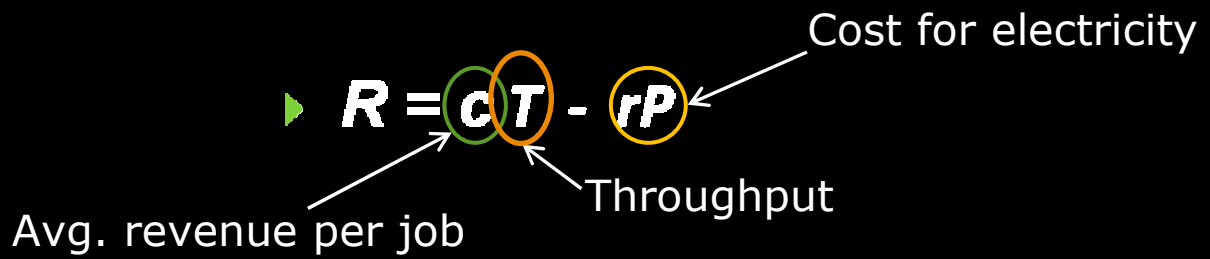
- *S* servers
  - *n* are running
  - *(S - n)* are switched off



- Every processed request generates a revenue of c$
- Electricity costs r$ per kWh
- If all servers are busy, further jobs are queued
  - But clients have limited patience!
  - If the waiting time exceeds a certain threshold (i.e., 1/θ on average), jobs are aborted
- **How to choose the "best" *n*?**

Proposed scheme (1)

- The performance of computing systems can be measured using different metrics
- The goal of the service provider is to maximize the long-term average earned revenue per unit time



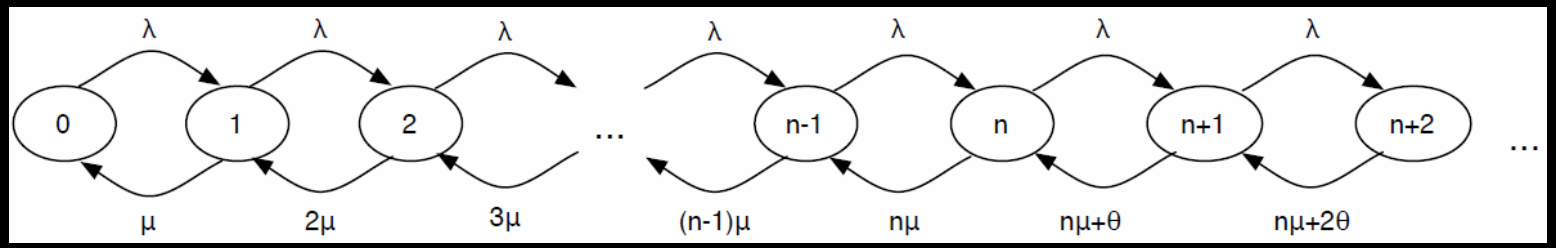
- For a given *n* and load, two variables are unknown and need to be estimated
  1. System throughput, T
  2. Power consumption, P
- For the following we will use the notation
  - ***R = r(n)***

Proposed scheme (2)

- One might try to model such a system using the Erlang-C model
  - 75% of people won't visit again a web site that took more than 4 sec. to load
  - We do allow HTTP timeouts as well as impatient customers
- … but Erlang-C does not acknowledge jobs abandonment

Throughput estimation (1)

- The system is treated as an M/M/n+M queue
  - The number of jobs inside the system is a state dependent Birth-and-Death process



- Let j denote the number of jobs inside the system

$$p_j = \begin{cases} \dfrac{n!}{j!\rho^{n-j}} p_n & \text{if } j \le n \\ \dfrac{(\lambda/\theta)^{j-n}}{j-n} p_n & \text{if } j > n \\ \prod_{k=1}^{} \left(\dfrac{n\mu}{\theta} + k\right) & \end{cases}$$

- **The system behaves like an M/M/∞ queue**
- **Jobs leave the system with rate $\mu_j = j\mu$**
- **The departure rate is nμ**
- **The abandonment rate depends on j**

Abandonment rate

- Hence, the system throughput is

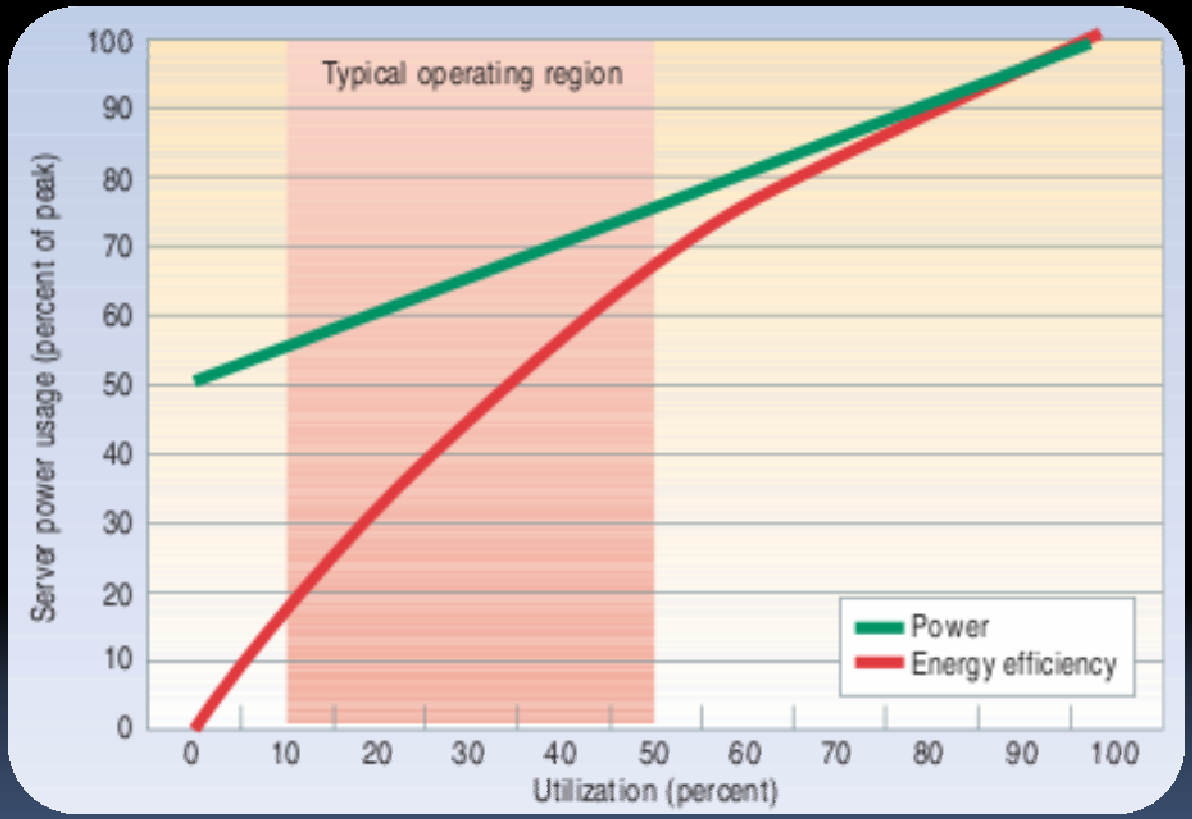
$$T = min(n\mu, \lambda[1 - P(Ab)])$$

Steady state probability of abandonment

$$P(Ab) = P(W > 0)P(Ab|W > 0)$$

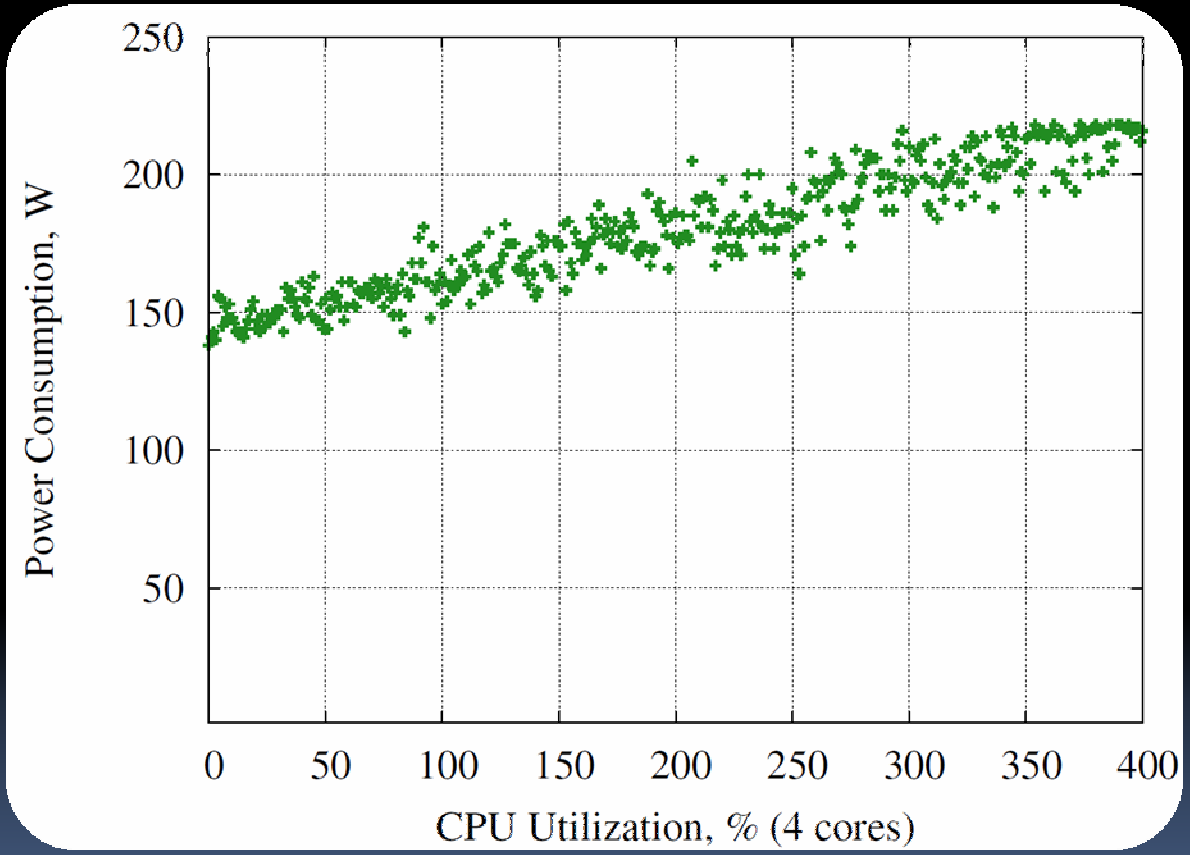
Power usage estimation

- Idle servers consume up to 65% of their peak consumption



Power usage estimation

- Idle servers consume up to 65% of their peak consumption



Avg. no. of busy servers

Consumption when busy

$$P = n e_1 + T/\mu (e_2 - e_1)$$

Consumption when idle

Optimal Allocation policy

- At each allocation epoch
  - Compute the new allocation, n'
  - Use the traffic estimates to evaluate the expected change in revenue, $\Delta r(n',n)$
    - $\Delta r(n',n) = r(n') - r(n) - Q$
      - Where Q is the cost paid for powering on (off) the $|(n'-n)|$ servers

$$Q = \frac{|\Delta n|}{t}\left(\sum_{i=1}^{l} d_i + kre_{max}\right)$$

  - Carry out the new allocation only if $\Delta r(n',n) > 0$
- The policy is implemented as a binary search algorithm
  - It can find the "best" n in O(log S) iterations

Heuristic policies

1. **Adaptive heuristic**
   - Large server farms working in the QED regime achieve both service quality and server efficiency
   - Allocate servers in proportion to the estimated load

$$n = \lceil \rho + \beta\sqrt{\rho} \rceil$$

Takes into account stochastic variability

   - Other schemes, i.e., use also the variance of the arrival rate over the time (see [3])

2. **Predictive heuristic**
   - Load changes over the time, it follows patterns, etc.
   - Allocate servers using double exponential smoothing

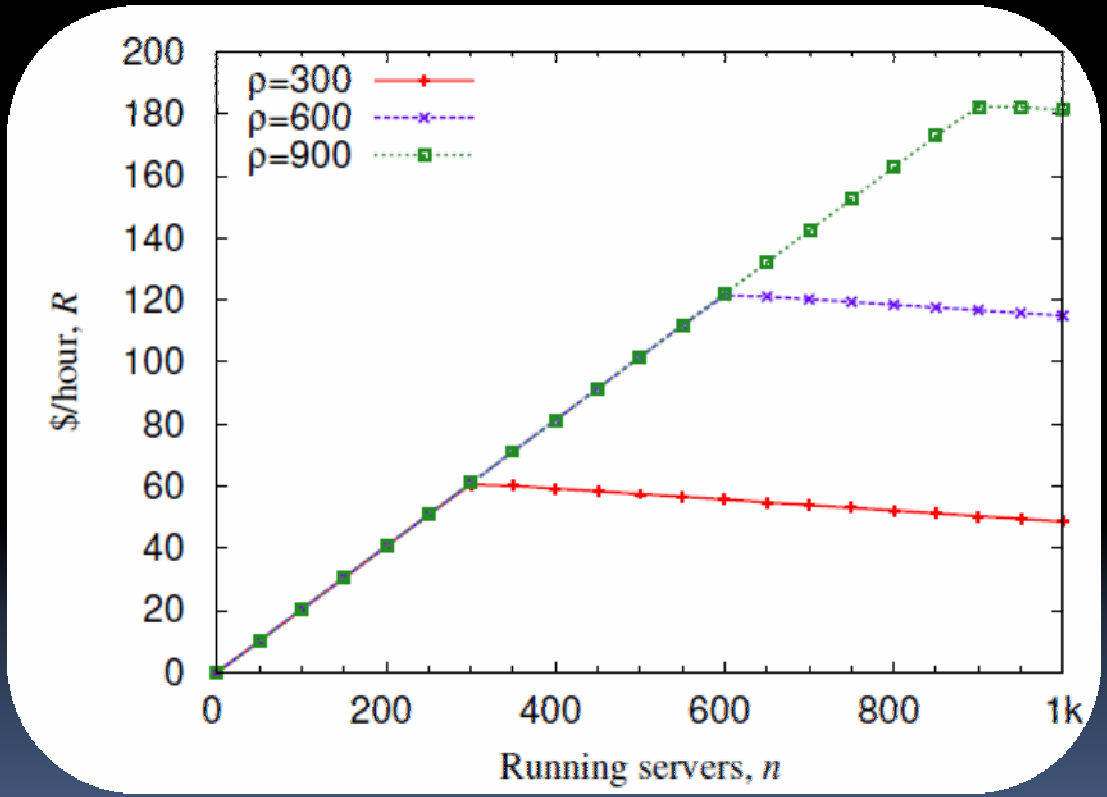
Adjusts the smoothed value →

Updates the trend →

$$\begin{cases} S_k = \alpha\lambda + (1-\alpha)(S_{k-1} + b_{k-1}) \\ b_k = \gamma(S_k - S_{k-1}) + (1-\gamma)b_{k-1} \end{cases}$$

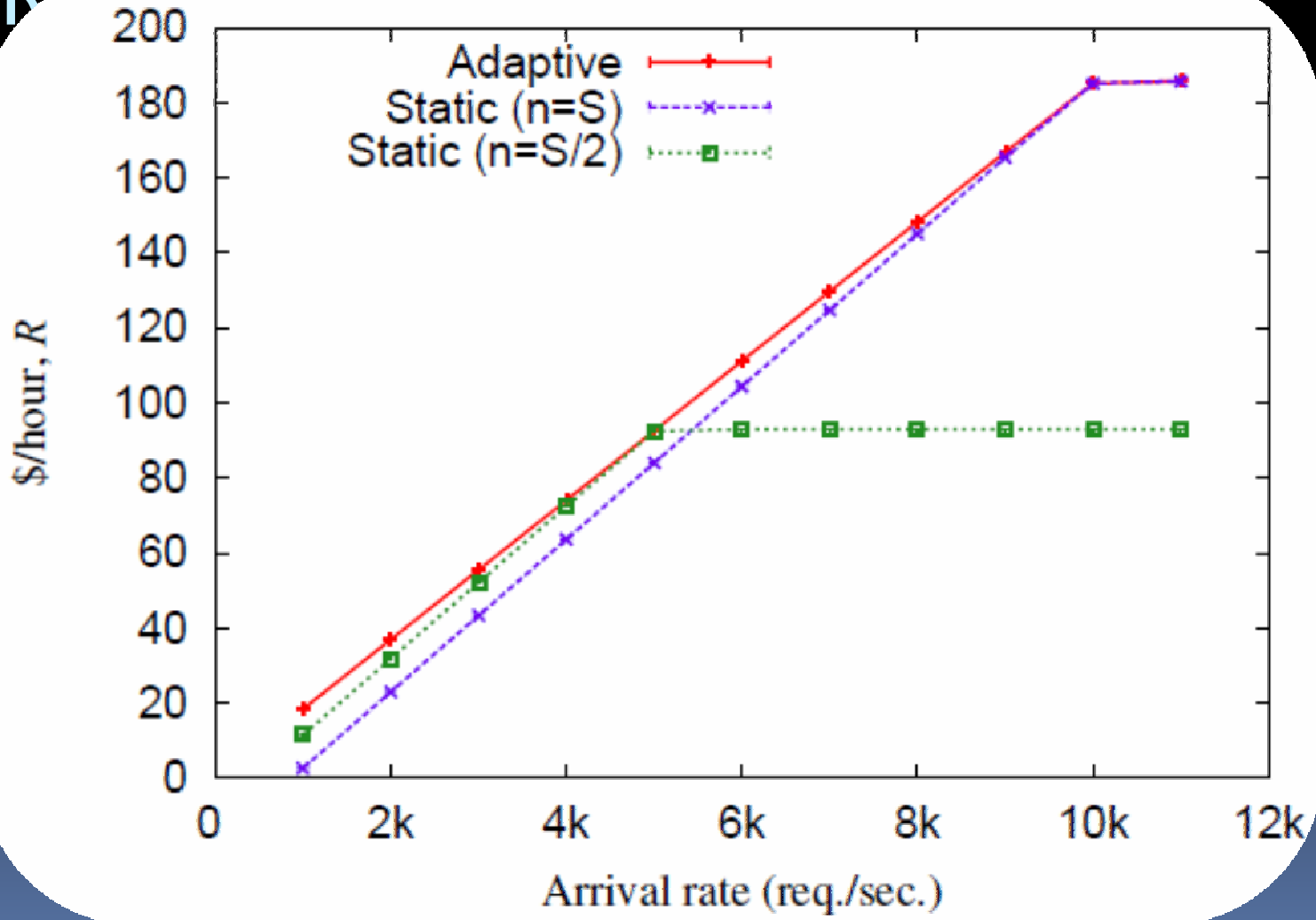
Performance evaluation

- Evaluates the effect of resource allocation on achieved revenue



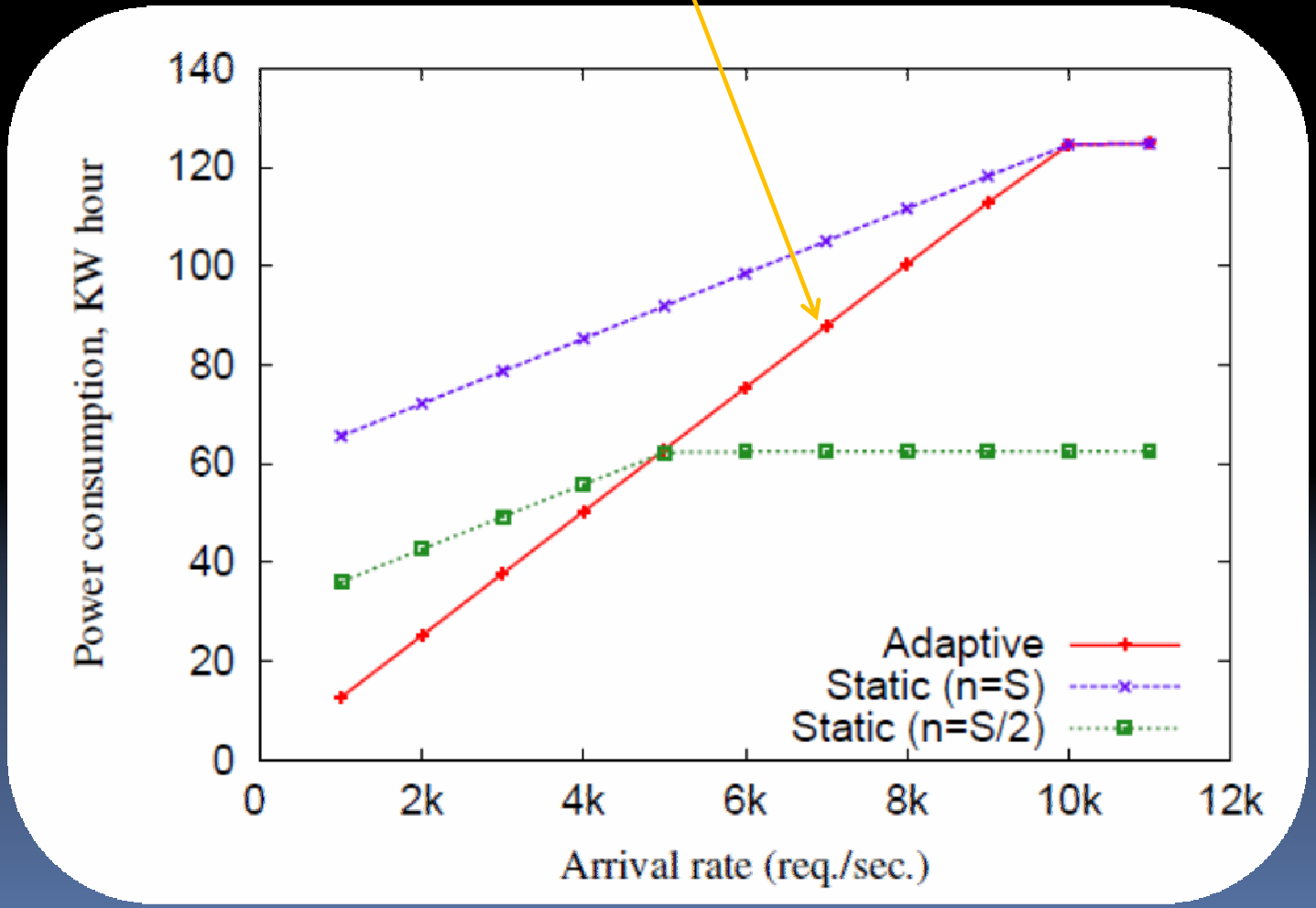
- S=1,000, r=0.1$/kWh, c=6.2*10^{-6} $
- 1/μ=0.1 sec, jobs are 70% CPU bound

Adaptive vs. Static, R

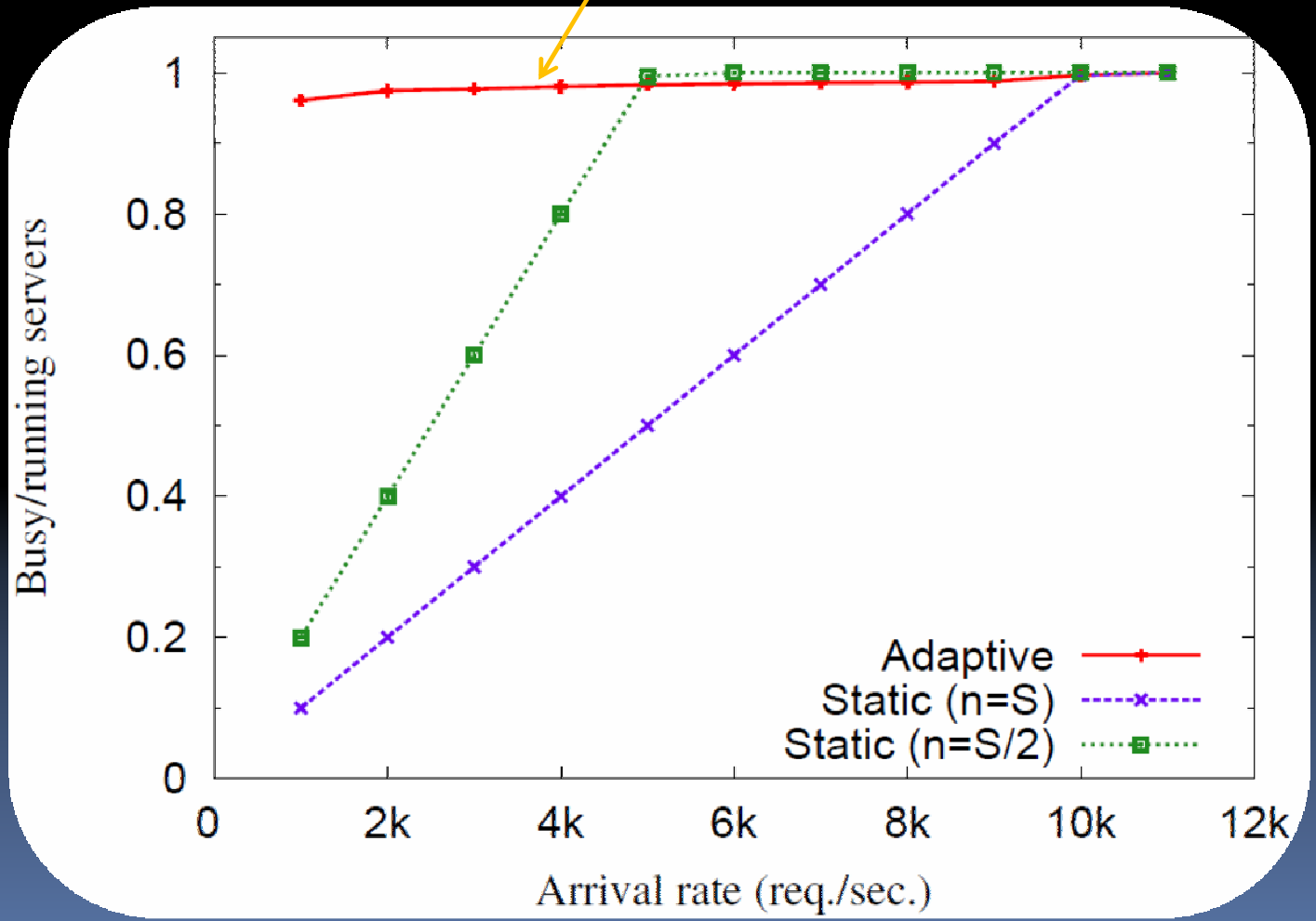


Adaptive vs. Static, Energy consumption

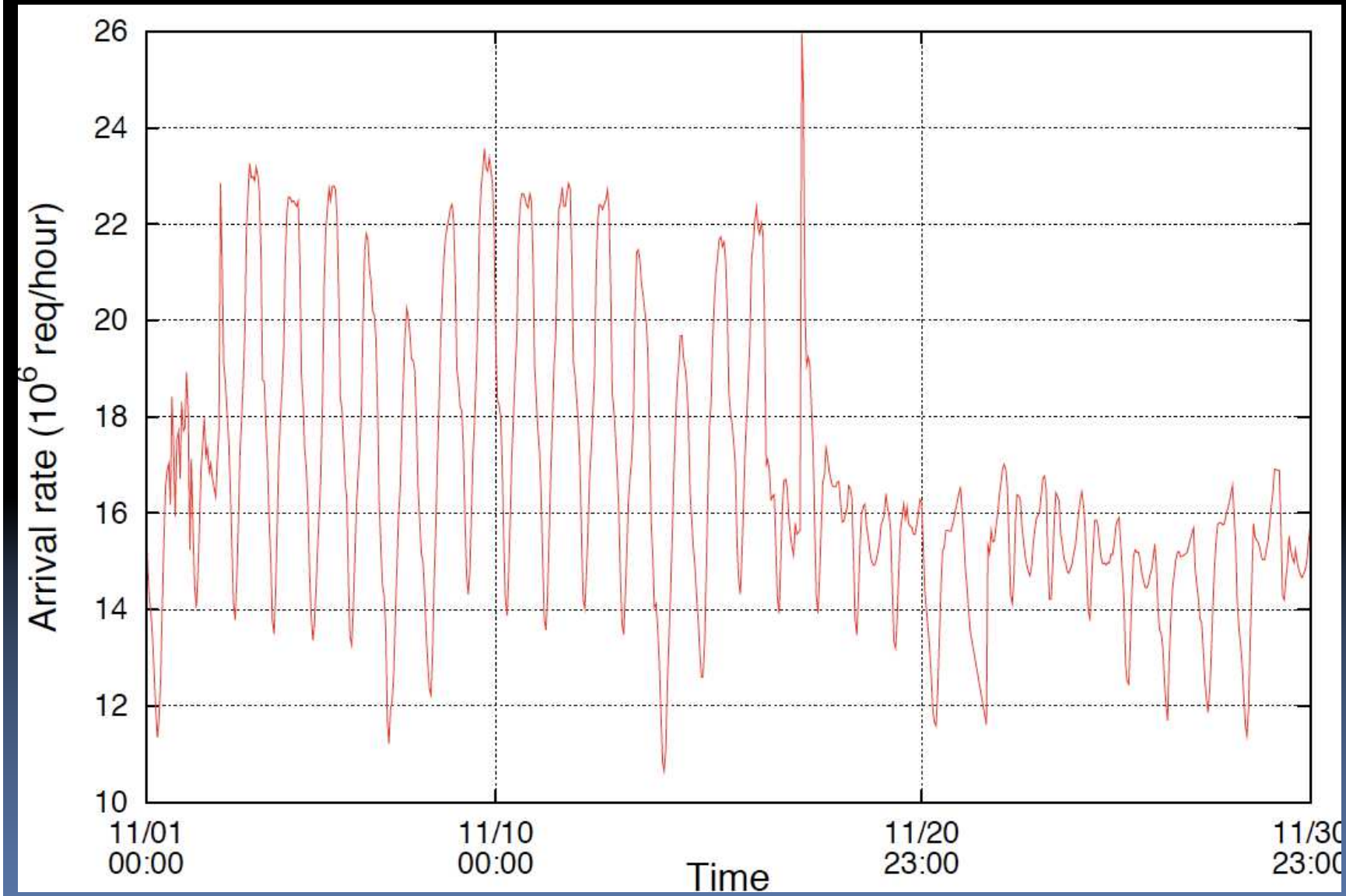
The adaptive policy runs servers only when necessary



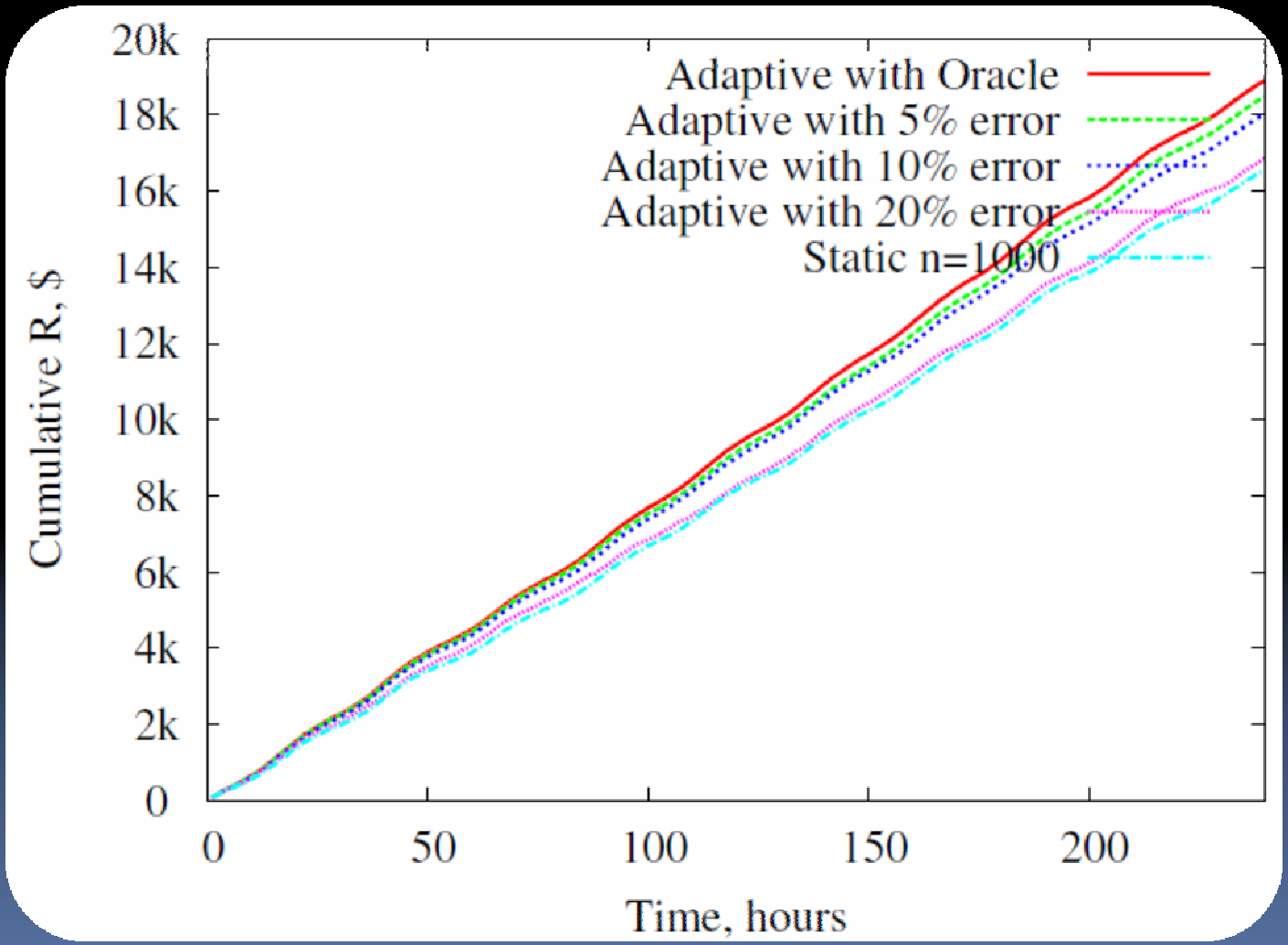
Adaptive vs. Static, Busy/Running servers



Wikipedia traces (November 2009)



Experiment with Wikipedia traces (sensitivity analysis)

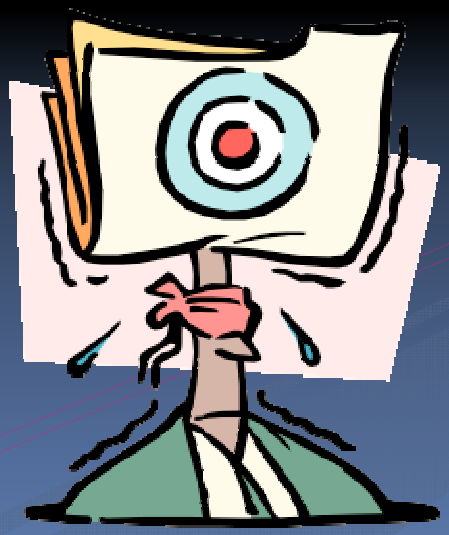


Conclusions

- Deciding how many servers to run has a significant effect on the revenue
  - The best decision depends on several factors
  - If the traffic is Markovian, our algorithm can find the best trade off between consumed power and performance
    - Violating the Markovian assumptions affects only the avg. queue length
- The policy we propose is robust against errors in parameters estimation

References

1. M. Mazzucco, D. Dyachuk, and R. Deters – "Maximizing Cloud Providers Revenues via Energy Aware Allocation Policies", in 3rd IEEE Cloud, Miami (USA), July 2010
2. M. Mazzucco, D. Dyachuk, and M. Dikaiakos - "Profit Aware Server Allocation for Green Internet Services", in 18th IEEE/ACM MASCOTS, Miami Beach (USA), August 2010
3. D. Dyachuk, and M. Mazzucco – "On Allocation Policies for Power and Performance", in 11th ACM/IEEE Grid (Energy Efficient Grids, Clouds and Clusters Workshop), Brussels (Belgium), October 2010



•Michele Mazzucco michele.mazzucco@ut.ee

•Dmytro Dyachuk dmytro.dyachuk@usask.ca