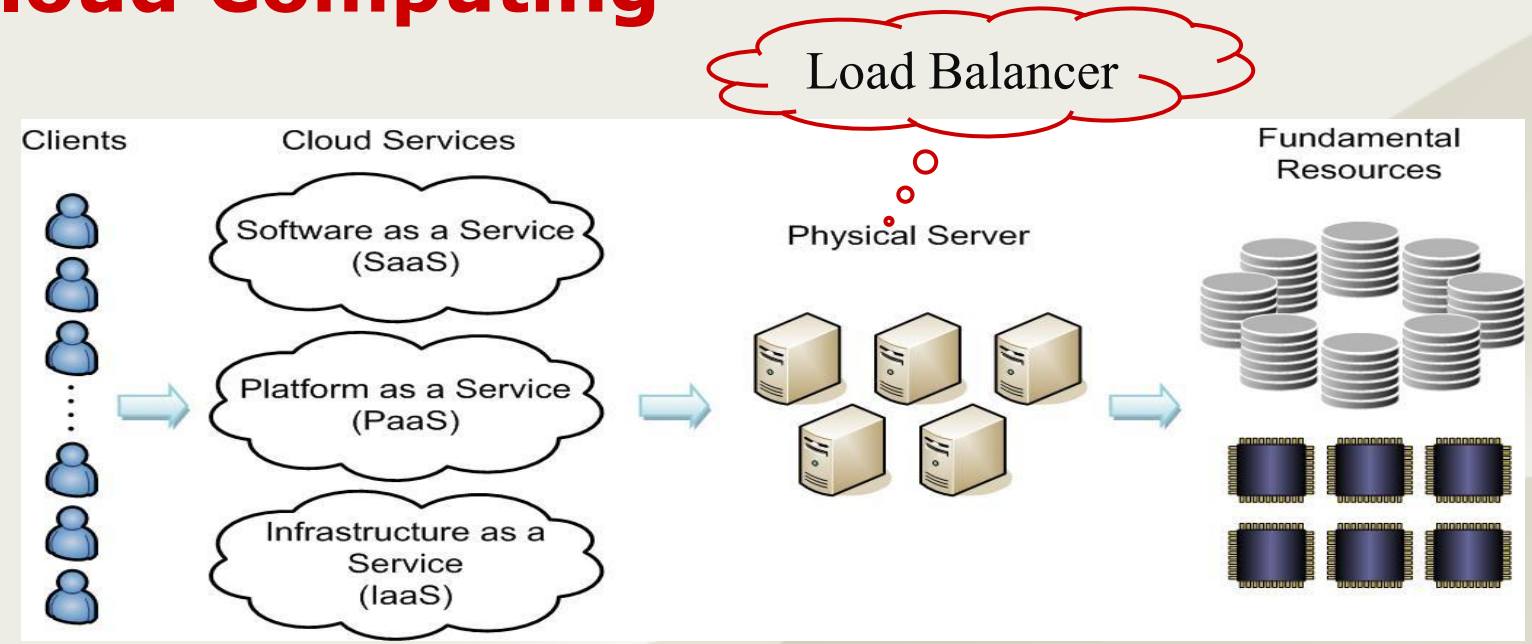
Northeastern

Adaptive Resource Allocation for Clouds under Bursty Workloads

J. Tai, J. Zhang, W. Meleis and N. Mi

Electrical and Computer Engineering Dept.
Northeastern University
{jtai, jzhang, meleis, ningfang}@ece.neu.edu

Cloud Computing

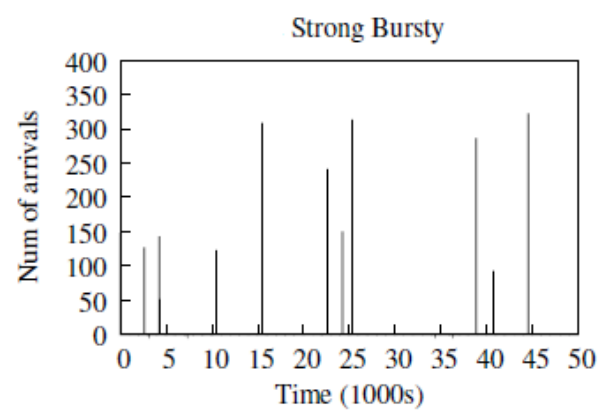


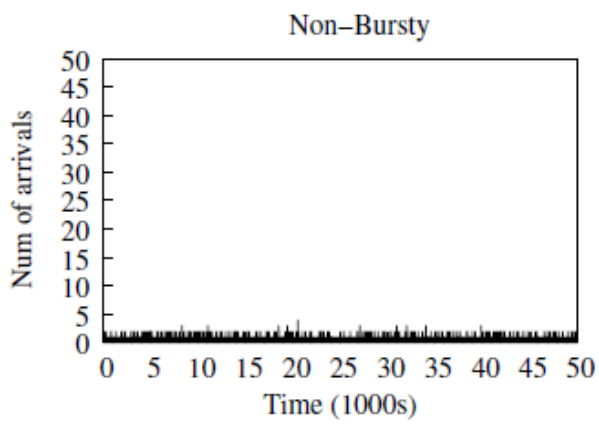
❖ **Two-level load balancing**

- *Level 1*: balance the load across a set of instances of the same application
- *Level 2*: balance the load of multiple applications among physical computers

Existing Problems (1)

- **Burstiness in computer systems**
  - Dramatically degrade the performance





- **Burstiness in Clouds**
  - Multi-remote-users
  - Not single-program-single-execution
  - application variety increases

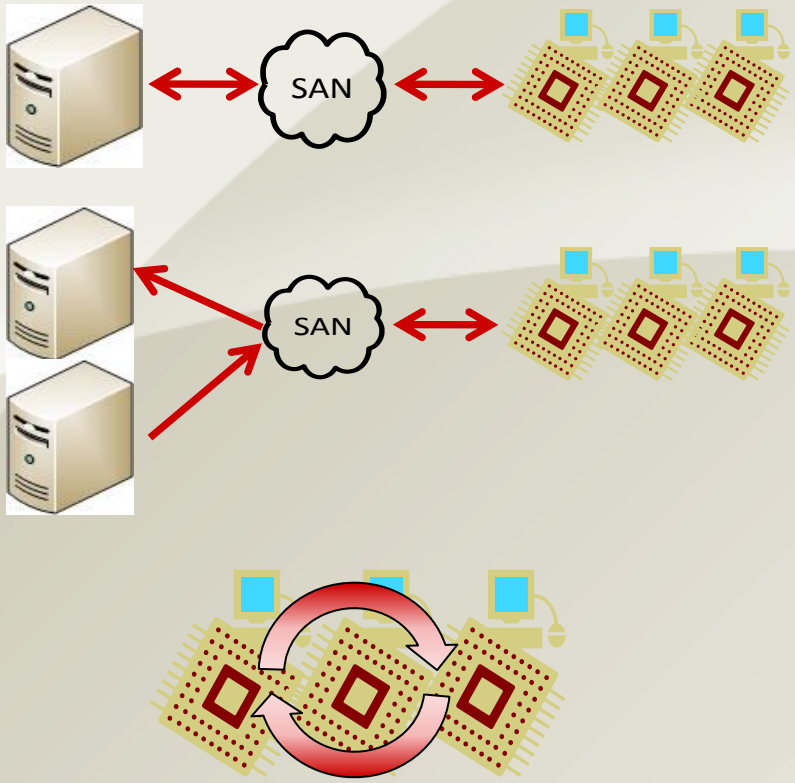
Existing Problems (2)

- **Information query delay**

Network delay

Inter-server communication delay

System load self-update delay

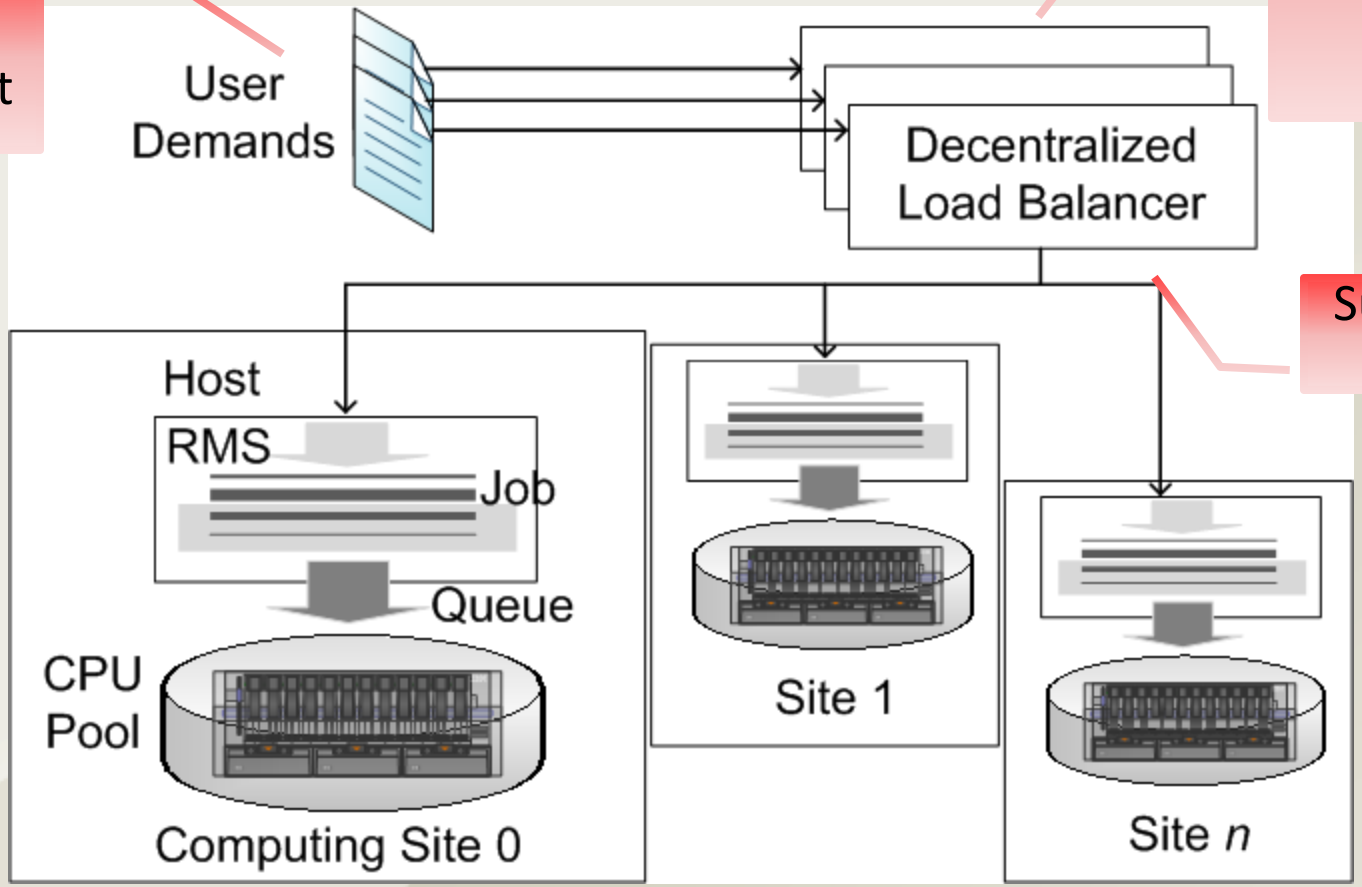


Simulation Environment

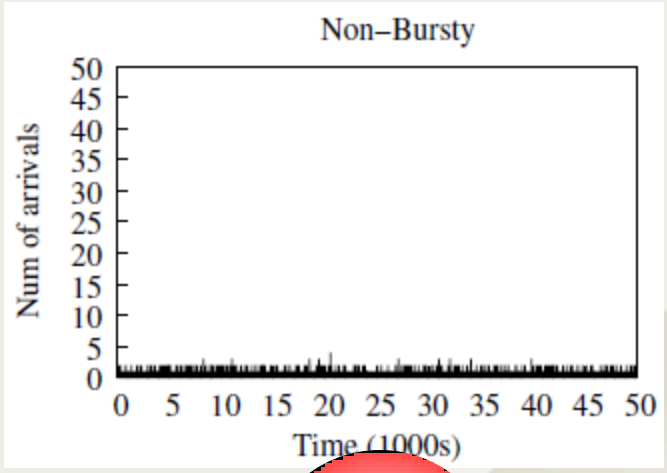
Burst/
Non-burst

Round Robin
Random
Qlen, Est. QT,
Act. QT
…

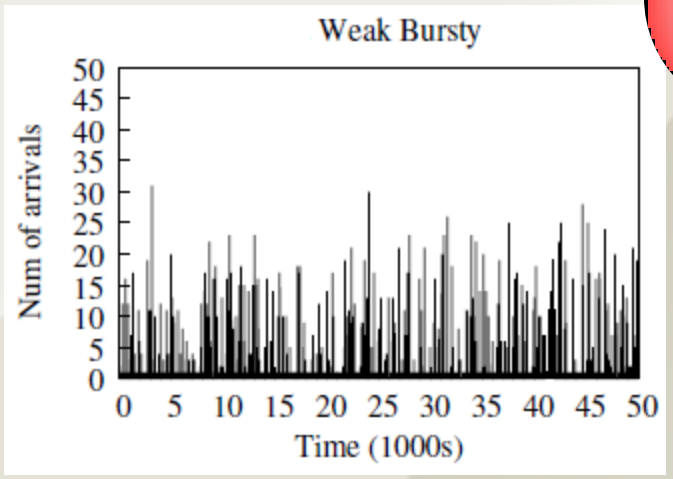
Submission
Delay

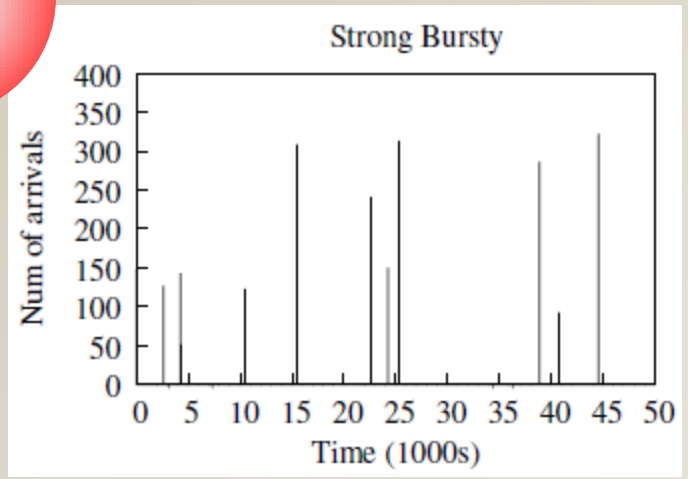


Arrival Traces



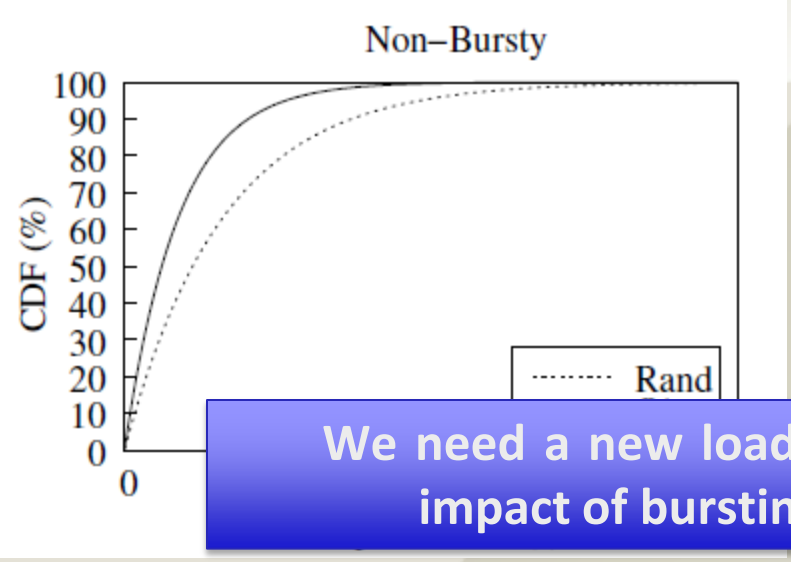
same mean

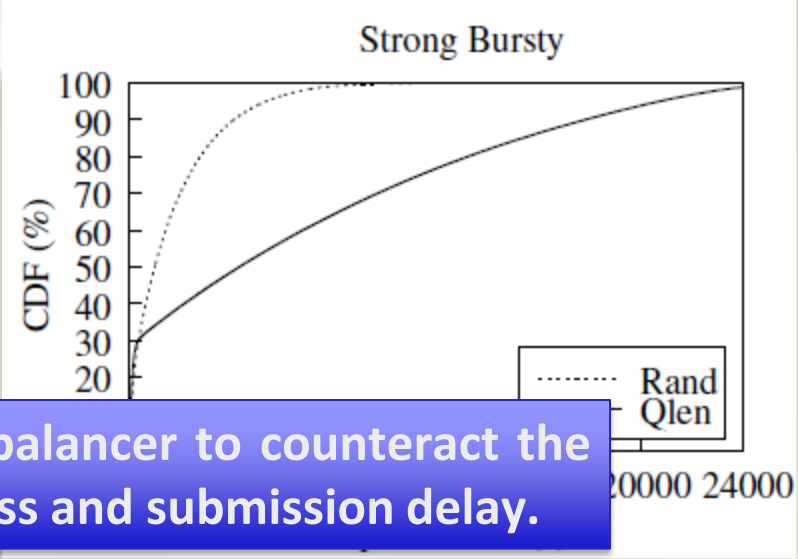




Burstiness & Delay Effect

Workload	*Rand*	*Qlen*	*Est. QT*	*Act. QT*
Non-bursty	80.5	7.6	7.6	7.4
Weak bursty	168.5	466.5	466.5	466.2
Strong bursty	1520.9	6541.5	6540.8	6541.0





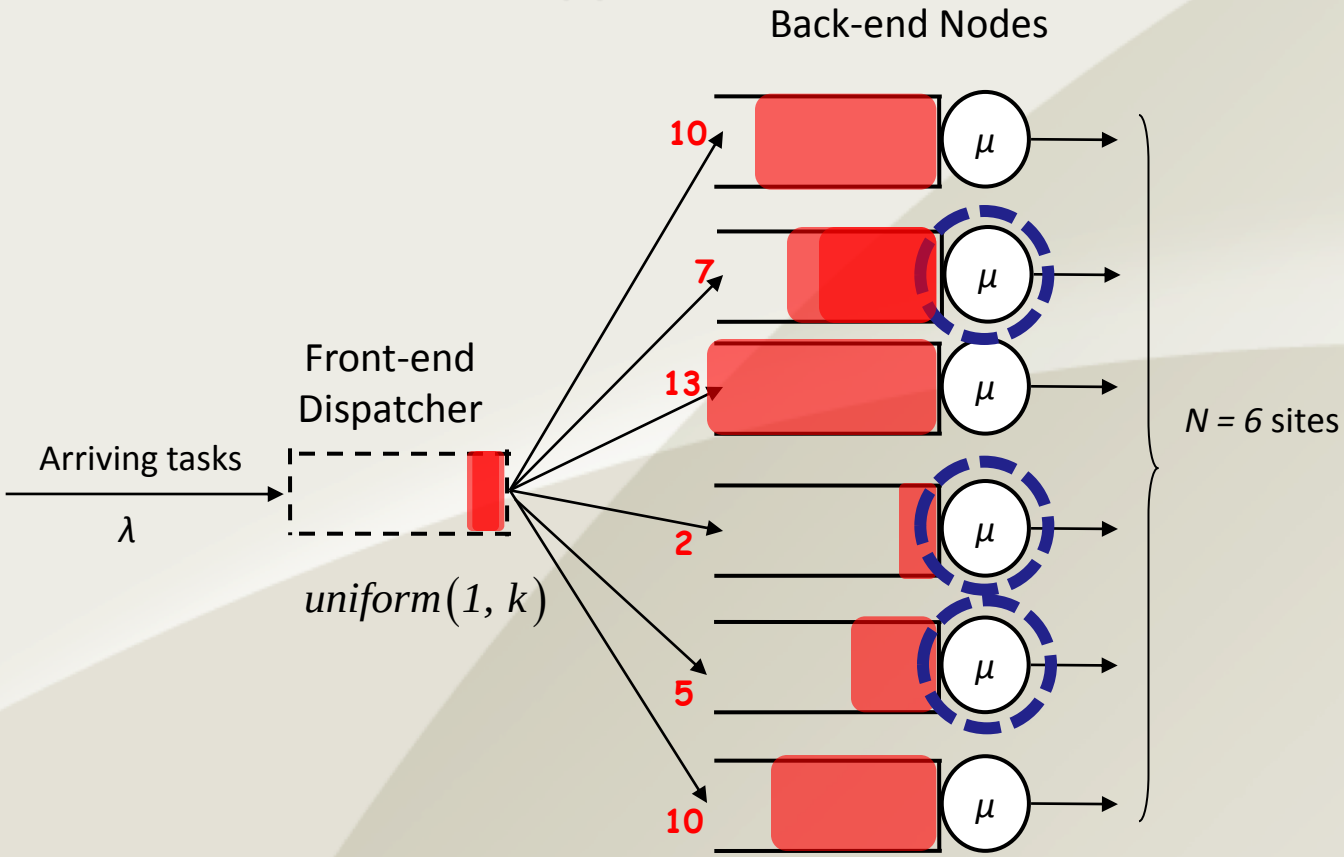
We need a new load balancer to counteract the impact of burstiness and submission delay.

Project Goal

- Adaptive resource allocation for clouds
  - Forecasts changes in user demands and system loads
  - Develop effective two-level load balancers
- Our expectations
  - Allow cloud users to experience higher quality of service
  - Allow cloud systems to make better use of their infrastructure

Our Initial Step: A new load balancer

- Balance the load within an application

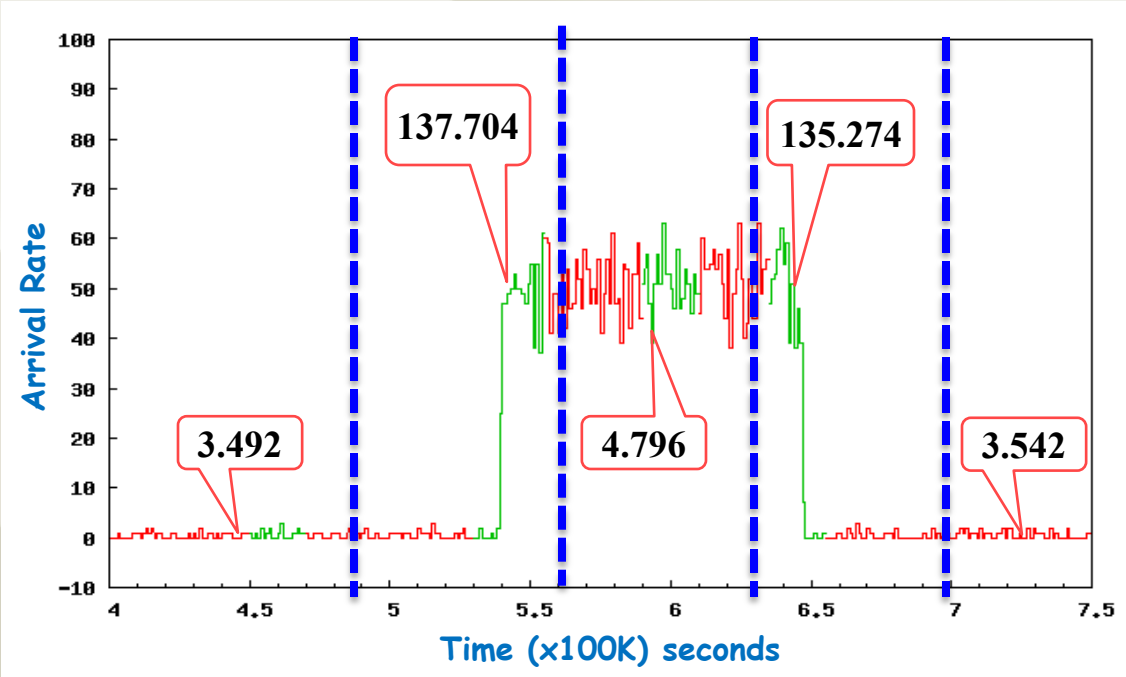


Our Initial Step: A new load balancer

- Balance the load within an application
- Online adjust *k* candidates

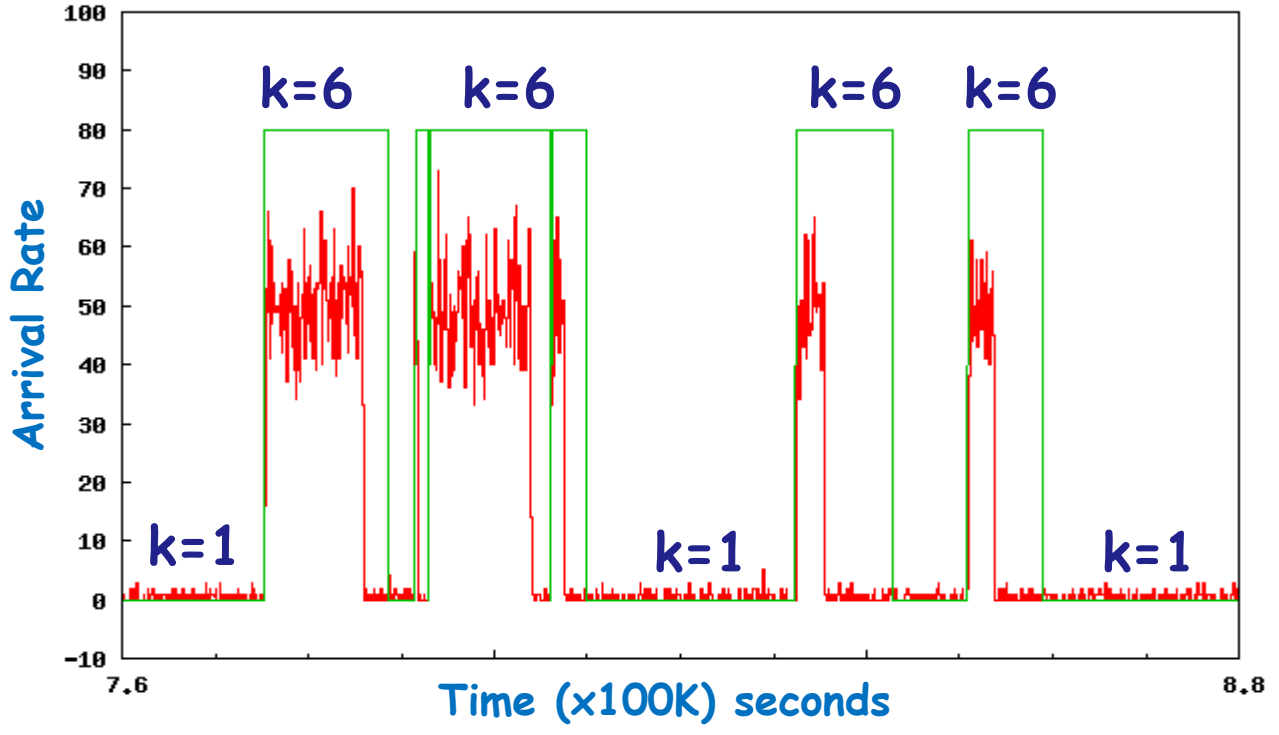
- Index of dispersion based prediction

$$I = SCV\left(1 + 2\sum_{k=1}^{\infty}\rho_k\right)$$

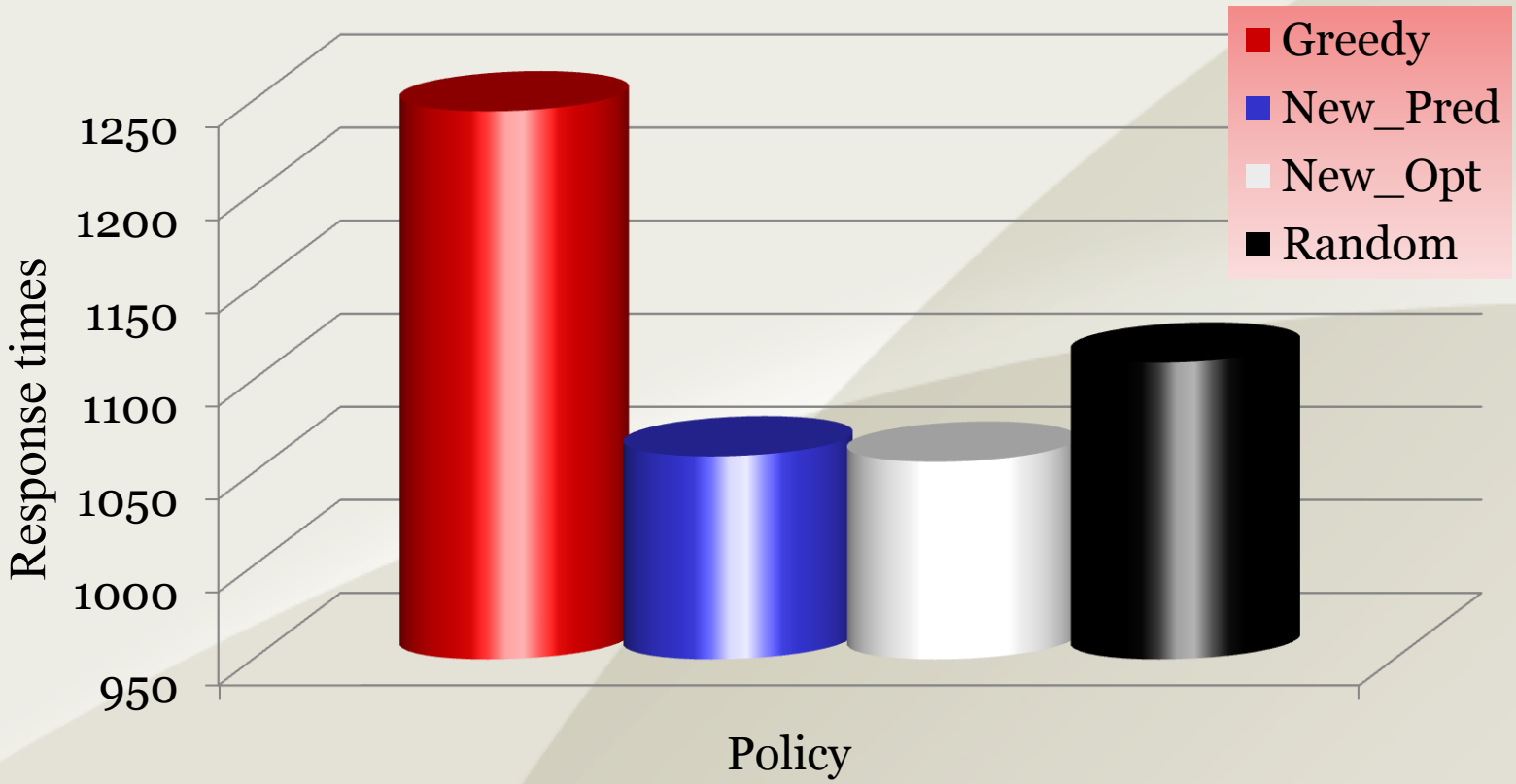


Our Initial Step: A new load balancer

- Balance the load within an application
- Online adjust *k* candidates
  - Idle phase: small *k*
  - Busy phase: large *k*



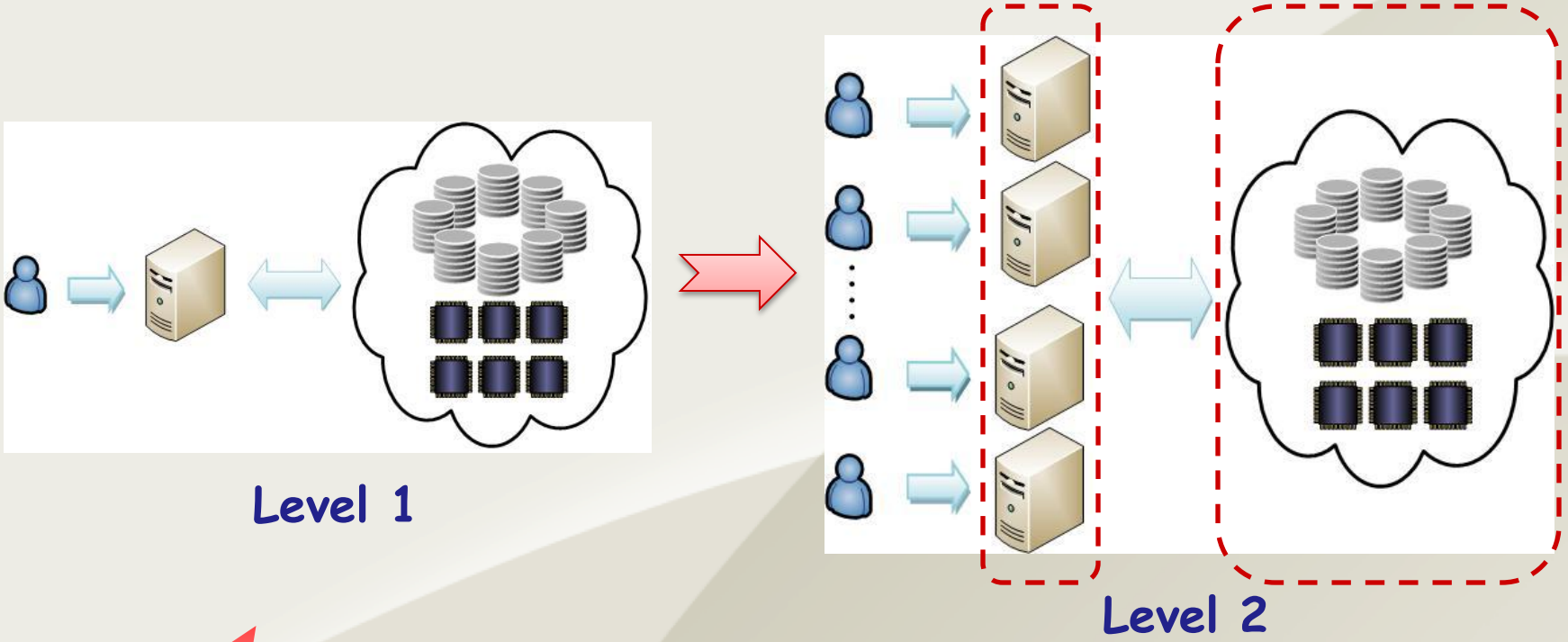
Some Preliminary Results



Next Steps …

Competition

Black Box



Level 1

Level 2

Challenge!

- Global views of user demands and system loads
- Coherency and dependency of arrival request
- Diversity of applications and client behaviors
- Failure and security issues
- Implementation in real systems (scalability)

Thanks

Q & A