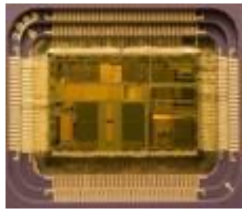
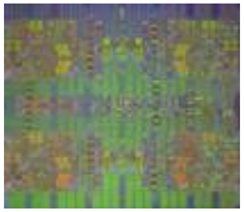
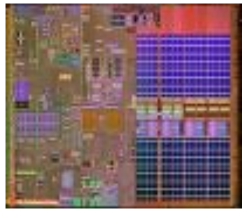
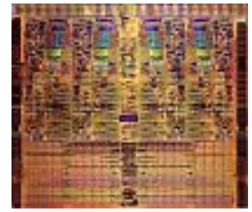


THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC




WASP: Workload Adaptive Energy-Latency Optimization in Server Farms using Server Low-Power States

Fan Yao, Jingxin Wu, Suresh Subramaniam, Guru Venkataramani

Department of Electrical and Computer Engineering

The George Washington University

Washington, DC, USA

CLOUD 2017, Honolulu, Hawaii, USA

THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

The Era of Data Explosion

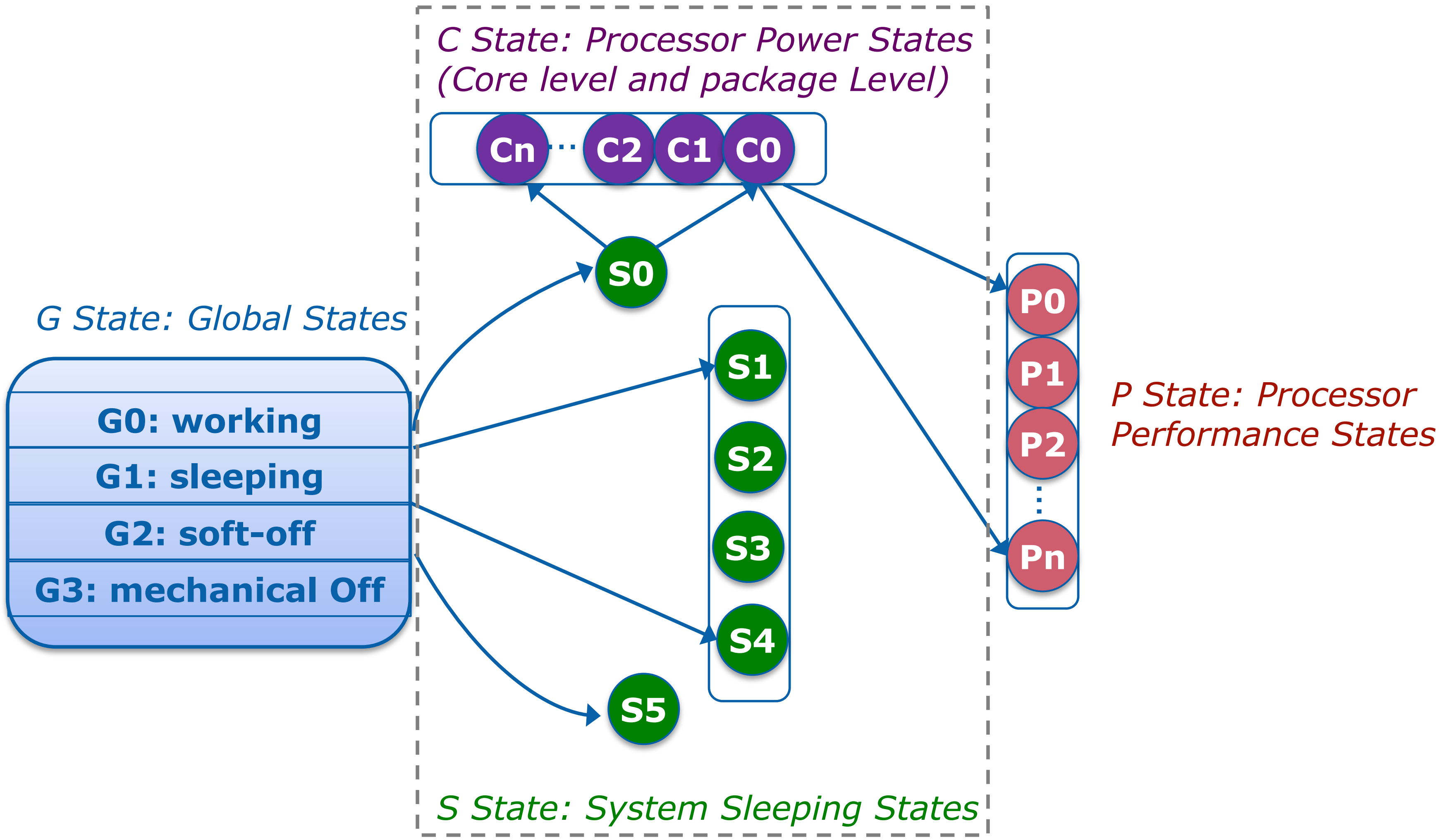
- **Tremendous advances in cloud computing**
  - Size of computing infrastructure grows rapidly
- **Modern data centers are increasingly power-hungry**
  - Power/cooling cost of data center increased over 400% last decade
  - IT equipment dominates the power consumption
- ***Limiting server farm energy envelope is critical***

Server Farm Energy Inefficiency

- **Servers are typically provisioned to match peak load**
  - Server farms are often under-utilized (30% utilization is common)
  - Wasteful energy is spent in keeping extra servers active

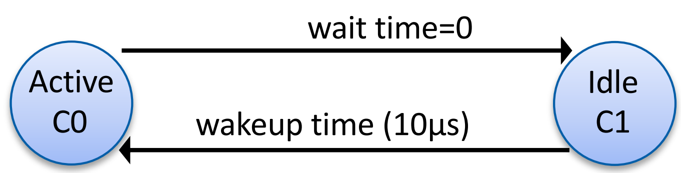
- **Today's server lacks energy-proportionality**
  - When active, server at 30% utilization consumes 60% of peak power
  - When idle, server consumes 20% ~ 55% of peak power

Server Low-power States

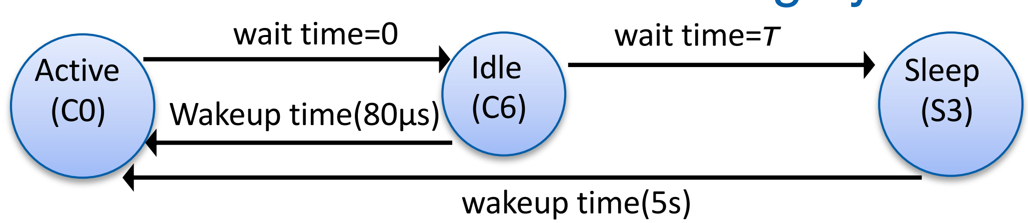


Exploration of Low Power States

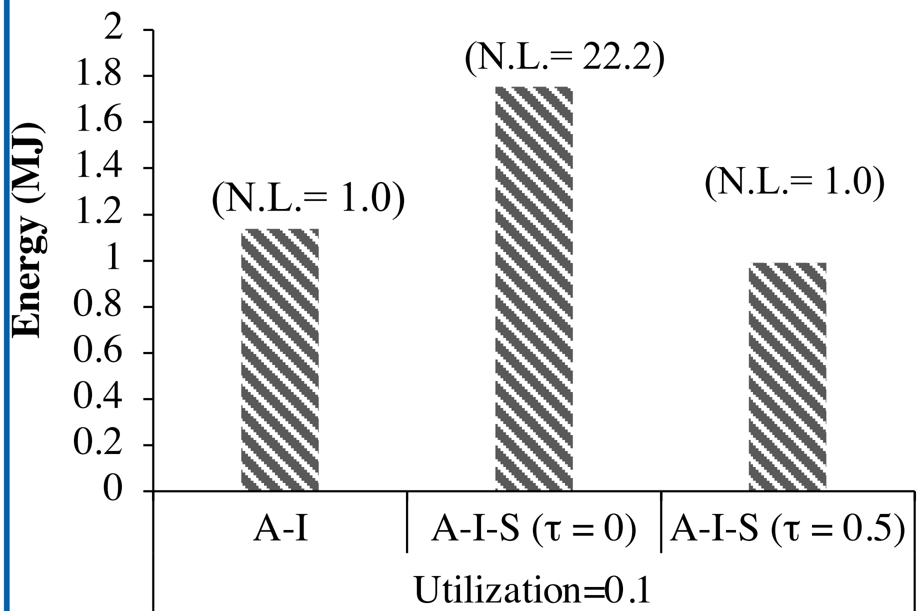
- **Two illustrative policies**
  - The *Active-Idle* configuration
    - Server alternates between active (*C0*) and idle state (*C1*)

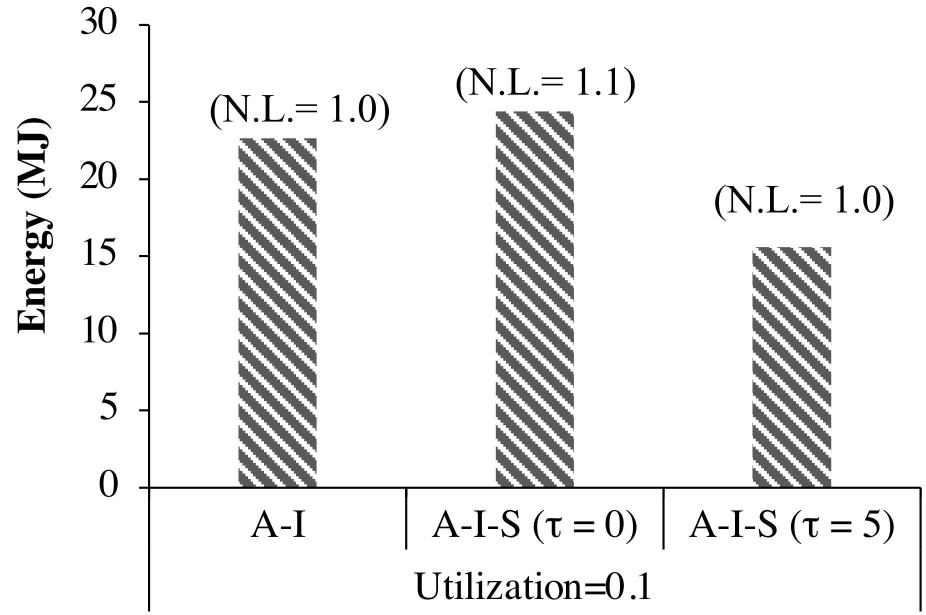


  - The *Delay-Doze (τ=c)* configuration
    - Transitions among active (*C0*), shallow sleep (*C6*) and deep sleep (system sleep *S3*)
    - Processor enters C6 when idle
    - Wait for τ seconds in C6 before entering system sleep.



Motivational Example





N.L. represents latency normalized to average service time.300K jobs are run for each workload

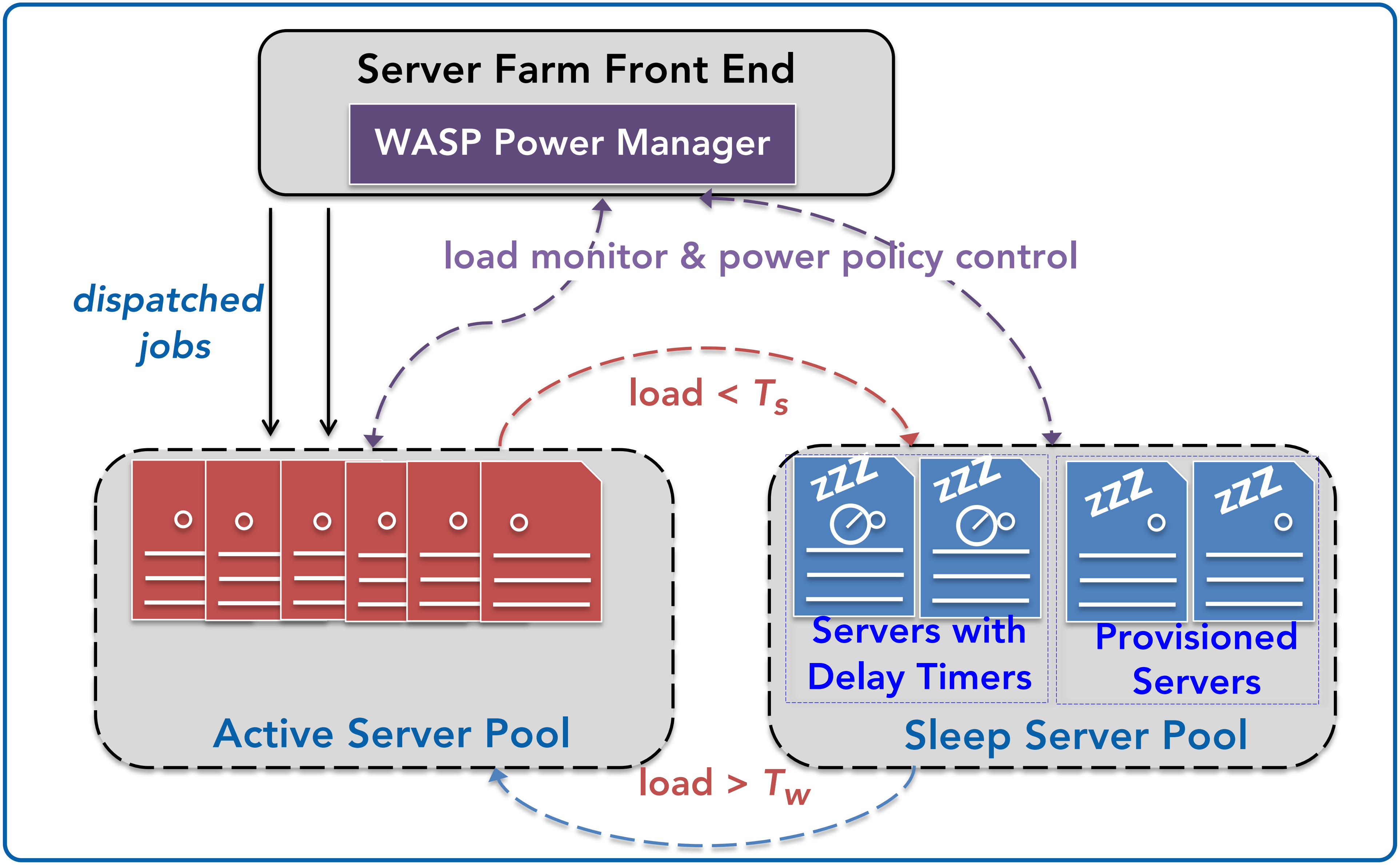
Energy consumption: **small workload**

Energy consumption: **large workload**

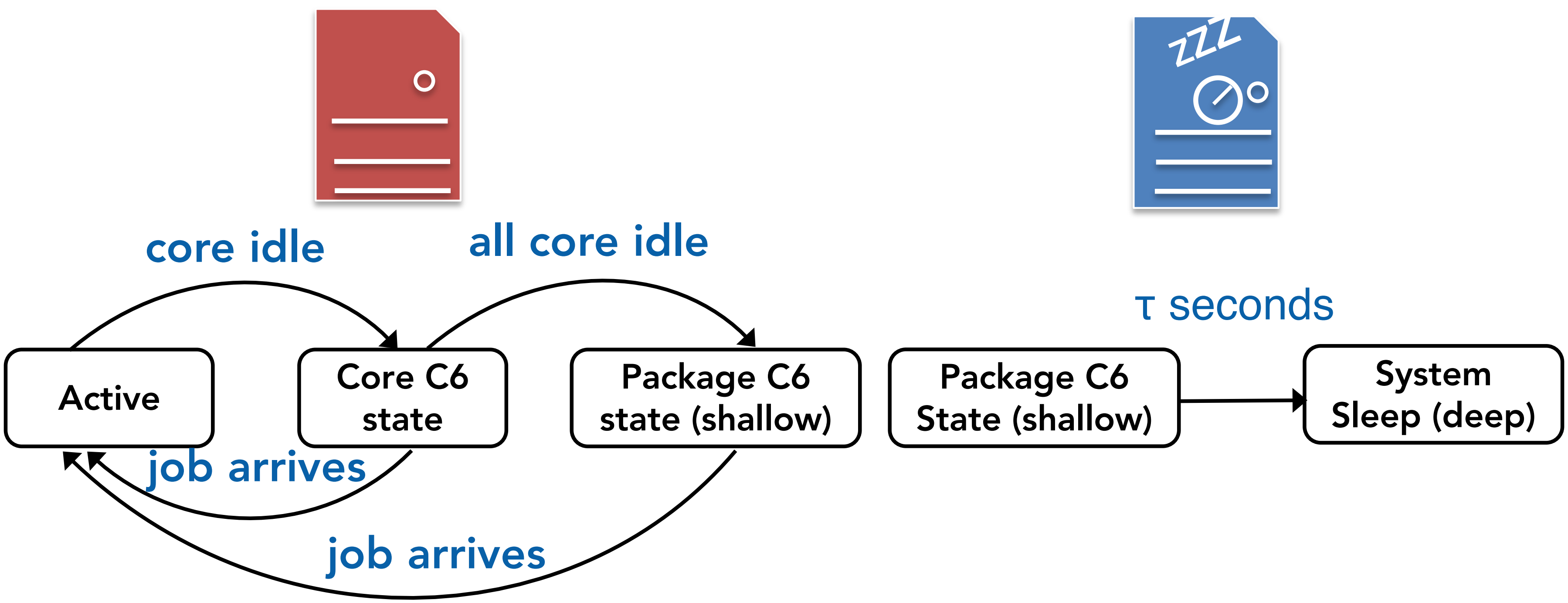
Workload Adaptive Energy-Latency Optimization

- **Optimize energy saving**
  - Leverage system/processor low power states
- **Maintain QoS constraints**
  - Satisfy tail latency requirements (e.g.,90th percentile response time)
- **Adaptive to distinct workload**
  - Adjust energy saving strategies according to various workloads

WASP Framework: Global Level



WASP Framework: Local Level



WASP Algorithm

- **Job scheduler dispatches jobs to schedulable servers**
  - Servers in active state but with free cores
  - Server in idle state with delay timer not expired
  - Job scheduler has a priority to select shallow sleep servers
  - When no such sever available, select deep sleep servers
- **Parameters to set:**
  - T_s: threshold (pending jobs per core) to put a server to sleep
  - T_w: threshold (pending jobs per core) to wake up a server
  - τ: waiting time threshold to enter system sleep state

Simulation Setup

- **Developed an event-driven simulator**
  - Model job queuing in multi-core, multi-server system
  - Models server processor and platform power
  - Reports job response time and energy consumption statistics
- **The simulated server farm configuration**
  - 50 servers
  - Each core is able to serve one job at a time
- **Simulation settings**
  - Small workload (average service time 1~10 ms, e.g., web services)
  - Large (average service time 100~200 ms, e.g., DNS services)
  - First 10,000 jobs are ignored for simulation warm-up

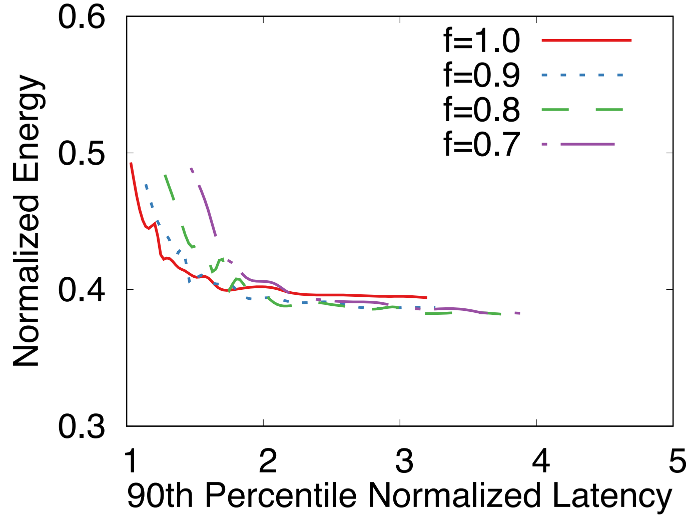
Server Power Model

Component	Core sleep C1*	Core sleep C6 †	Pkg. sleep C6	System sleep
CPU	$33.0+3.1\times(n_a-1)$	$23.0+3.8\times(n_a-1)$	8.3	8.3
RAM	10.8	10.8	4.9	1.4
Platform	45.5	45.5	23.6	4.8
Total Power	$89.3+3.1\times(n_a-1)$	$79.3+3.8\times(n_a-1)$	36.8	14.5

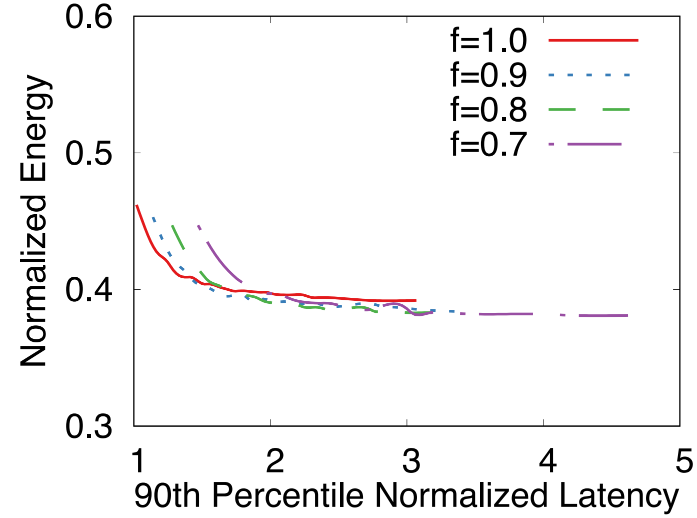
n_a : the number of cores in active state
Core sleep C1 : processor is active, idle cores are in c1 state
Core sleep C6 : processor is active, idle cores are in c6 state
Pkg. sleep C6 : entire processor in C6 state

CPU power is based on linear regression model using power profiles for the Intel Xeon E5-2680 processor

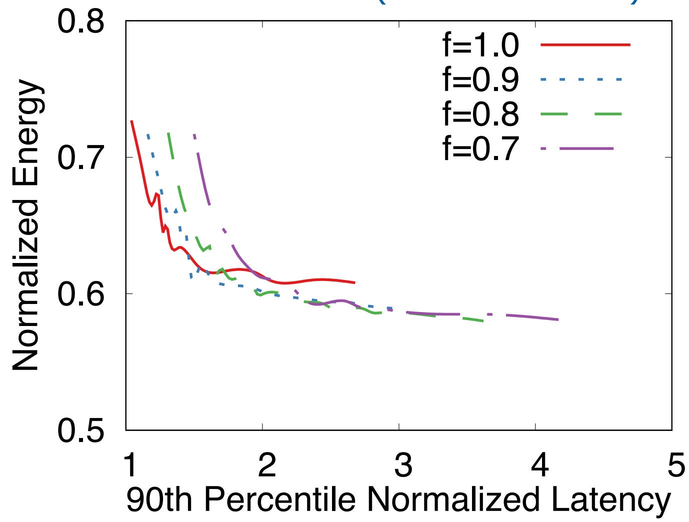
Pareto-optimal Space Exploration



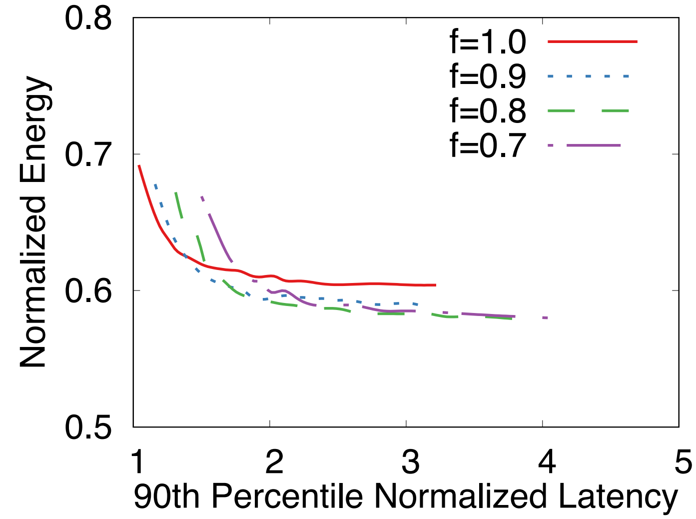
Web Service (Utilization 10%)



DNS Service(Utilization 10%)



Web Service (Utilization 30%)



DNS Service(Utilization 30%)

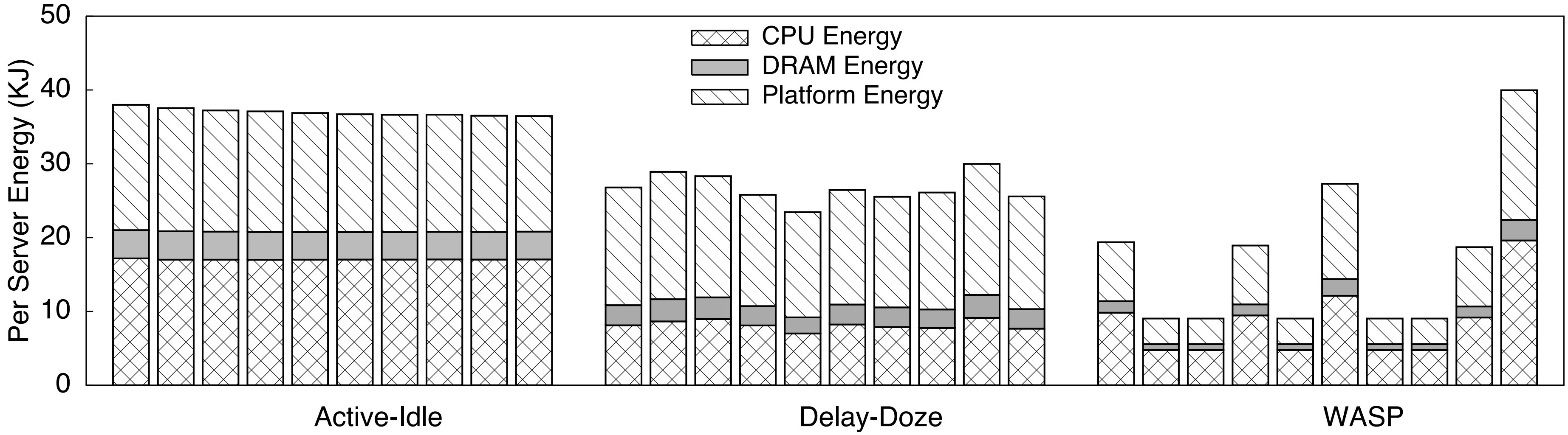
Exploration Observations

- τ is independent of utilization levels, but is job-size dependent.
- T_w is independent of utilization levels.
- T_s values are independent of job execution latencies and utilization levels.

System Evaluation

- **A cluster of 10 servers**
  - Dell M1000e cluster
- **Each server is equipped with 12 cores**
  - Dual-socket Intel X5650 processor
  - 12GB DRAM
  - 256GB Disk
- **Deployed with apache web service**
  - Wikipedia and NLANR workload
- **QoS Goal: 90th percentile latency as 2x service time**

Energy Savings for Wikipedia Workload

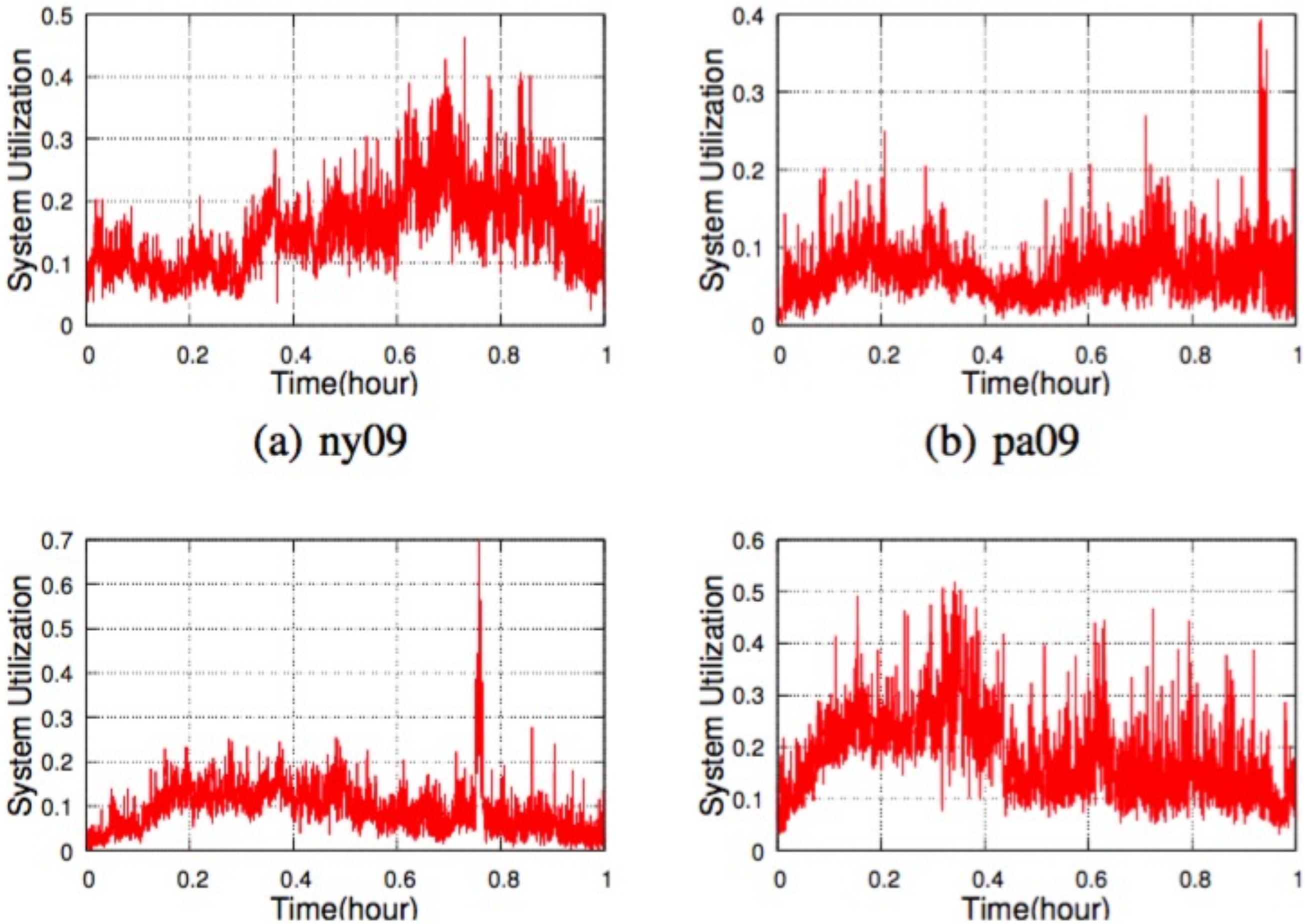


**For each configuration, the 10 bars represent the energy consumption for the 10 servers*

WASP achieves 57% energy saving active-idle
WASP achieves 39% energy saving over delay-doze

Bursty NLANR workloads

- **Workload Patterns**



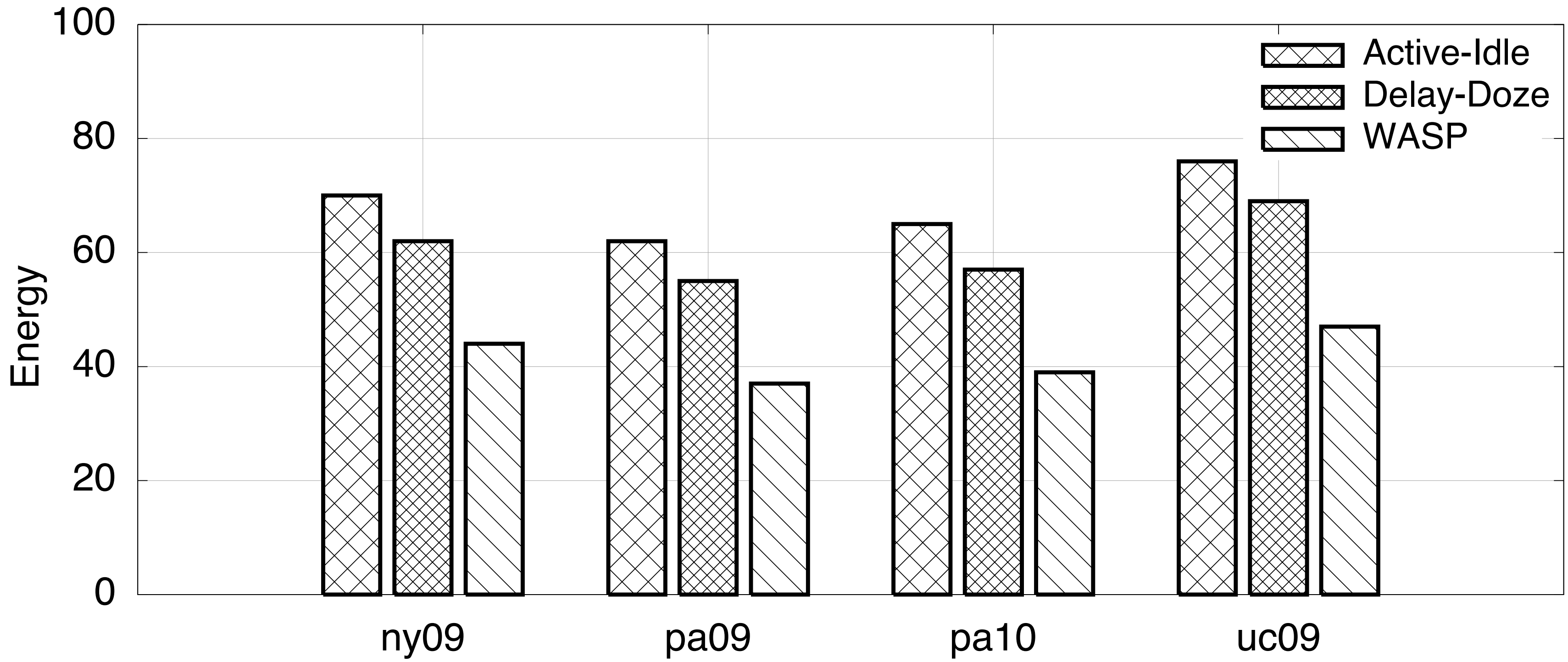
(a) ny09

(b) pa09

(c) pa10

(d) uc09

Energy Savings on Cluster



Energy is normalized to the peak energy the system consumes when all server are running at peak power

Delay-Doze only achieves 9%~12% energy saving
WASP achieve 34%~40% energy saving compared to Active-Idle

Conclusions

- We proposed techniques that makes smart use of processor/system low-power

- We performed an exploration of Pareto-optimal Energy-Latency tradeoffs

- We implement a prototype on real system and showed upto 57% energy saving with QoS guarantees.

- **Acknowledgements**