Towards More Efficient Utilization of Computer Systems

Niklas Carlsson
Linköping University
Sweden

Martin Arlitt
HP labs
USA/Canada

LiU expanding reality

Motivation

- Delay-sensitive (interactive) workloads common
- Systems typically dimensioned to achieve good response times
  - Often utilization of 10-50% (owing to diurnal access patterns)
- Turning off resources (to save energy costs) not necessarily a good solution ...
  - E.g., consider “value generation” / TCO

The value of resources

“ … if you have additional work that is more valuable than the cost of electricity, then it makes sense to use the servers rather than turn them off …”

- James Hamilton (during ACM SIGMETRICS keynote 2009)

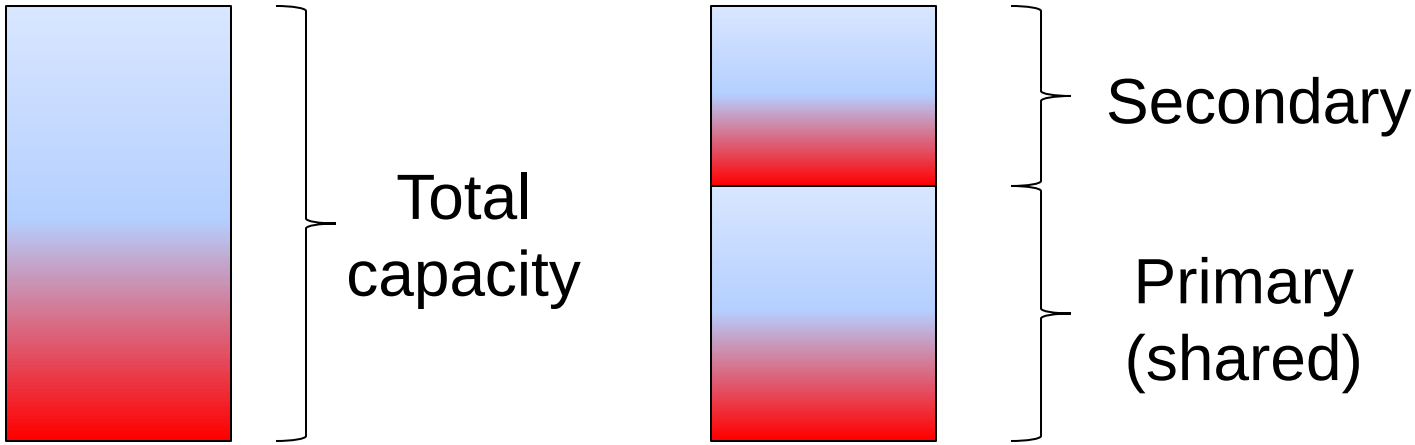
System Model

- Workloads
  - Delay-sensitive (prioritized)
  - Delay-tolerant (background)

- System objectives
  - Service guarantees (average or upper percentiles) for delay-sensitive workload
  - High system utilization (i.e., high throughput of delay-tolerant jobs)
  - Non-preemptive delay-tolerant jobs

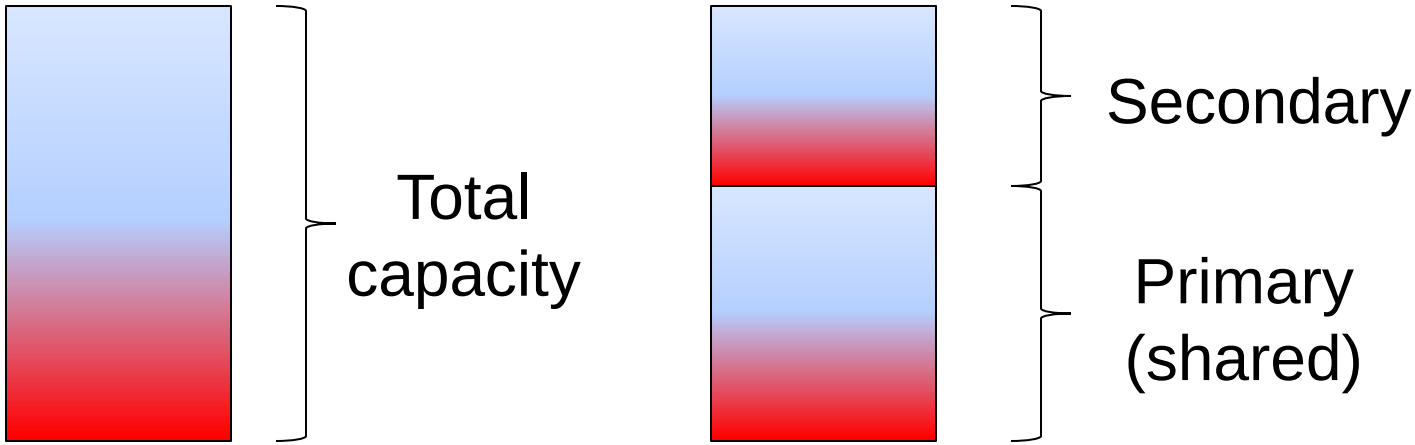
Server partitioning

- Primary partition (potentially shared)
  - Delay-sensitive workload(s)
  - Delay-tolerant workload(s)
- Secondary partition
  - Delay-tolerant workload(s)



Basic questions

- Goal: maximize utilization, given level of service (response time)
  - How to partition resources?
  - How to distribute delay-tolerant workload?
    - Insulated vs shared use of primary partition



Steady state analysis

- Consider primary partition
  - Shared resource
  - B (service rate)

- Vacation-period model
  - Delay-sensitive (“jobs”)
  - Delay-tolerant (“vacations”)
  - Idle periods (“infinitesimal vacation”)

Average response time

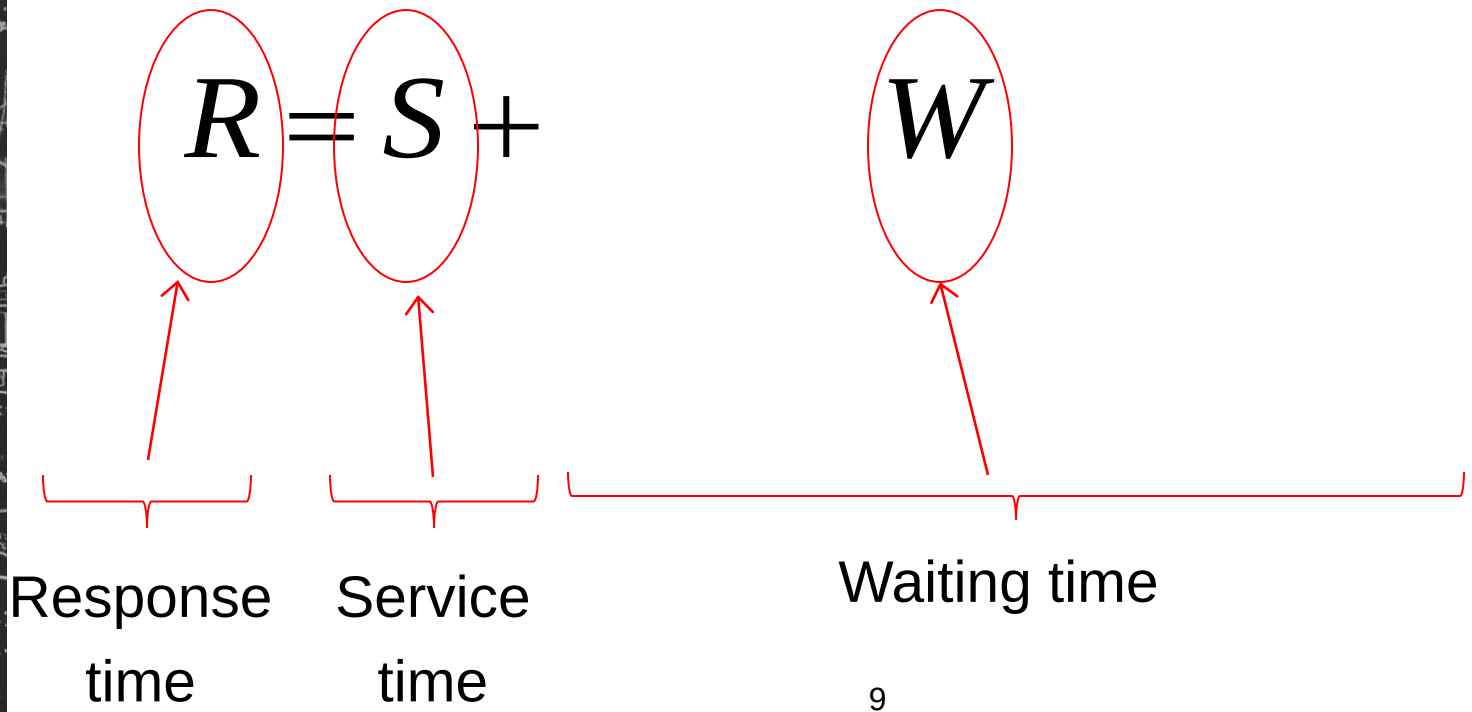
$$R = S + W$$

Response time

Service time

Waiting time

Average response time



Average response time

$$R = S + W$$

Response time

Service time

Waiting time

Average response time

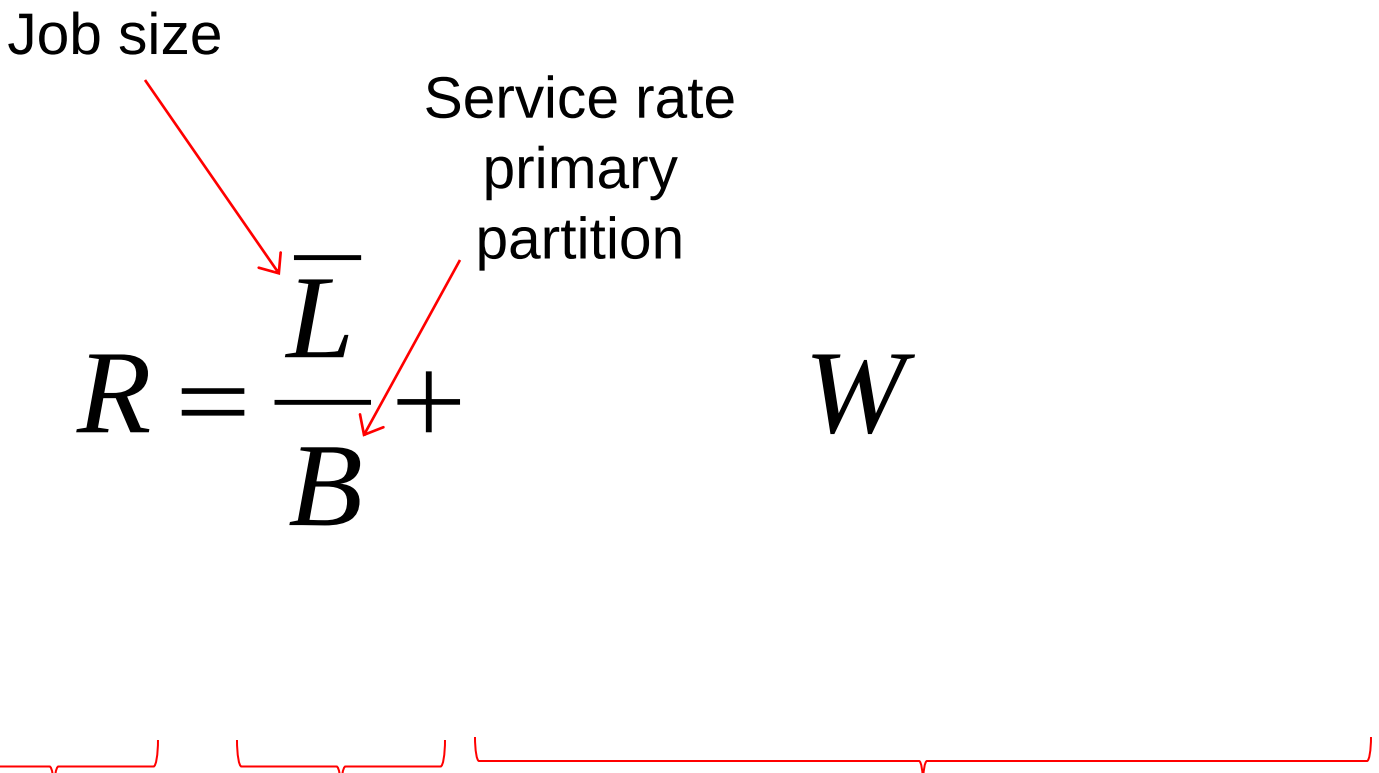
$$R = \frac{\bar{L}}{B} + W$$

Response time

Service time

Waiting time

Average response time



Average response time

$$R = \frac{\bar{L}}{B} + W$$

Response time

Service time

Waiting time

Average response time

$$R = \frac{\bar{L}}{B} + W$$

Response time

Service time

Waiting time

Average response time

$$R = \frac{\overline{L}}{B} + \frac{\overline{L}}{B}\frac{\lambda\left(\overline{L^2}/B^2\right)}{2(1-\rho)} + \frac{\overline{U^2}}{2\overline{U}}$$

Response time

Service time

Waiting time

Average response time

$$R = \frac{\overline{L}}{B} + \frac{\overline{L}}{B}\frac{\lambda\left(\overline{L^2}/B^2\right)}{2(1-\rho)} + \frac{\overline{U^2}}{2\overline{U}}$$

Delay-sensitive

Delay-tolerant

Response time

Service time

Waiting time

Average response time

$$R = \frac{\overline{L}}{B} + \frac{\overline{L}}{B}\frac{\lambda\left(\overline{L^2}/B^2\right)}{2(1-\rho)} + \frac{\overline{U^2}}{2\overline{U}}$$

Average response time

$$R = \frac{\bar{L}}{B} + \frac{\bar{L}}{B}\frac{\lambda\left(\overline{L^2} / B^2\right)}{2(1-\rho)} + \frac{\overline{U^2}}{2\bar{U}}$$

Effects of larger (shared) primary partition?

Effects of larger job-size variation?

Average response time

$$R = \frac{\overline{L}}{B} + \frac{\overline{L}}{B}\frac{\lambda\left(\overline{L^2}/B^2\right)}{2(1-\rho)} + \frac{\overline{U^2}}{2\overline{U}}$$

Effects of larger (shared) primary partition

$B \uparrow \rightarrow \rho \downarrow$ $(\rho = \lambda L/B)$

$B \uparrow \rightarrow R \downarrow$

Good …

Average response time

$$R = \frac{\overline{L}}{B} + \frac{\overline{L}}{B}\frac{\lambda\left(\overline{L^2}/B^2\right)}{2(1-\rho)} + \frac{\overline{U^2}}{2\overline{U}}$$

Effects of larger (shared) primary partition

$B \uparrow \rightarrow \rho \downarrow$

$B \uparrow \rightarrow R \downarrow$

Effects of larger job-size variation

$U^2/U \uparrow \rightarrow R \uparrow$

Bad …

Average response time

$$R = \frac{\overline{L}}{B} + \frac{\overline{L}}{B}\frac{\lambda\left(\overline{L^2}/B^2\right)}{2(1-\rho)} + \frac{\overline{U^2}}{2\overline{U}}$$

Effects of larger (shared) primary partition

$B \uparrow \rightarrow \rho \downarrow$

$B \uparrow \rightarrow R \downarrow$

Effects of larger job-size variation

$U^2/U \uparrow \rightarrow R \uparrow$

Average response time

$$R = \frac{\overline{L}}{B} + \frac{\overline{L}}{B}\frac{\lambda\left(\overline{L^2}/B^2\right)}{2(1-\rho)} + \frac{\overline{U^2}}{2\overline{U}}$$

Effects of larger (shared) primary partition

$B\uparrow \rightarrow \rho\downarrow$

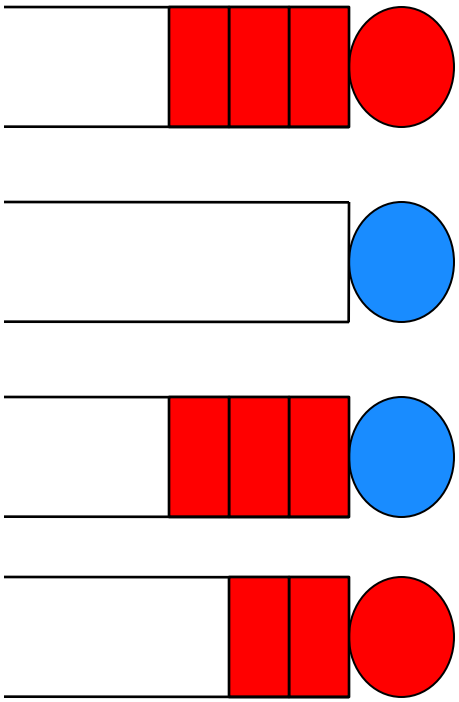
$B\uparrow \rightarrow R\downarrow$ Bigger shared resource positive …

Effects of larger job-size variation

$U^2/U\uparrow \rightarrow R\uparrow$ … unless too high job-size variability

Percentile analysis

- Queue behavior



Delay-sensitive served

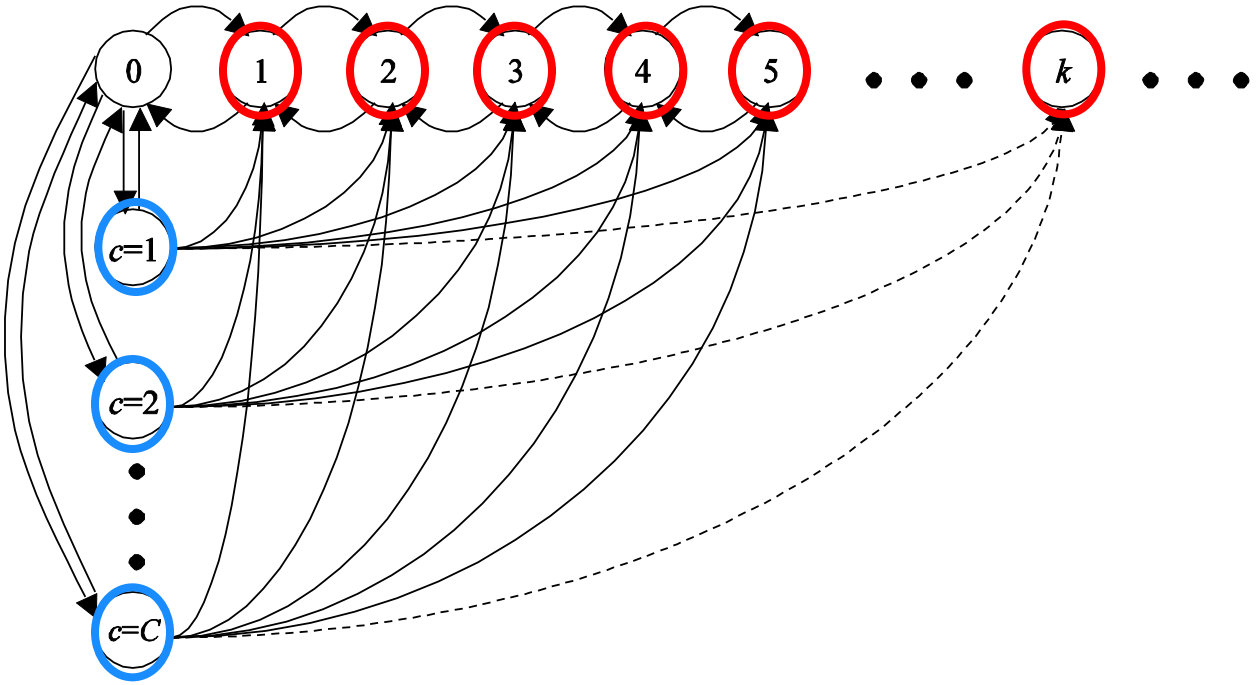
Delay-tolerant (only when free)

Can still build queue …

But as soon as done …

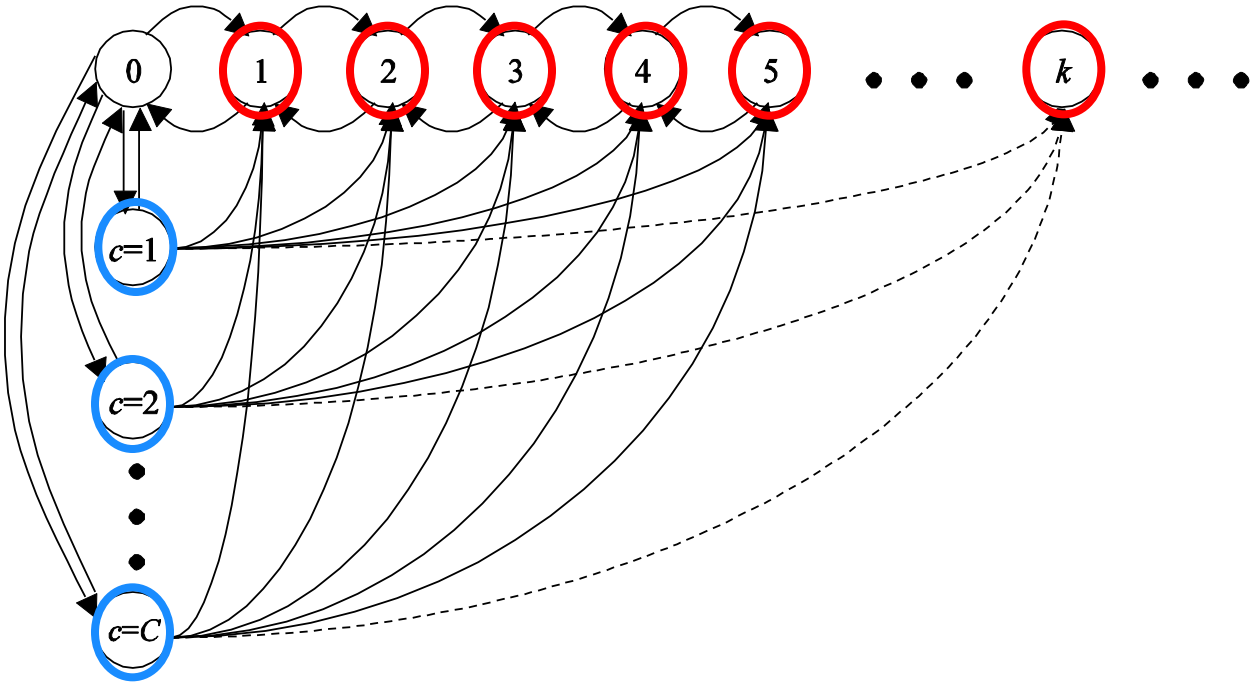
Percentile analysis

- State transitions



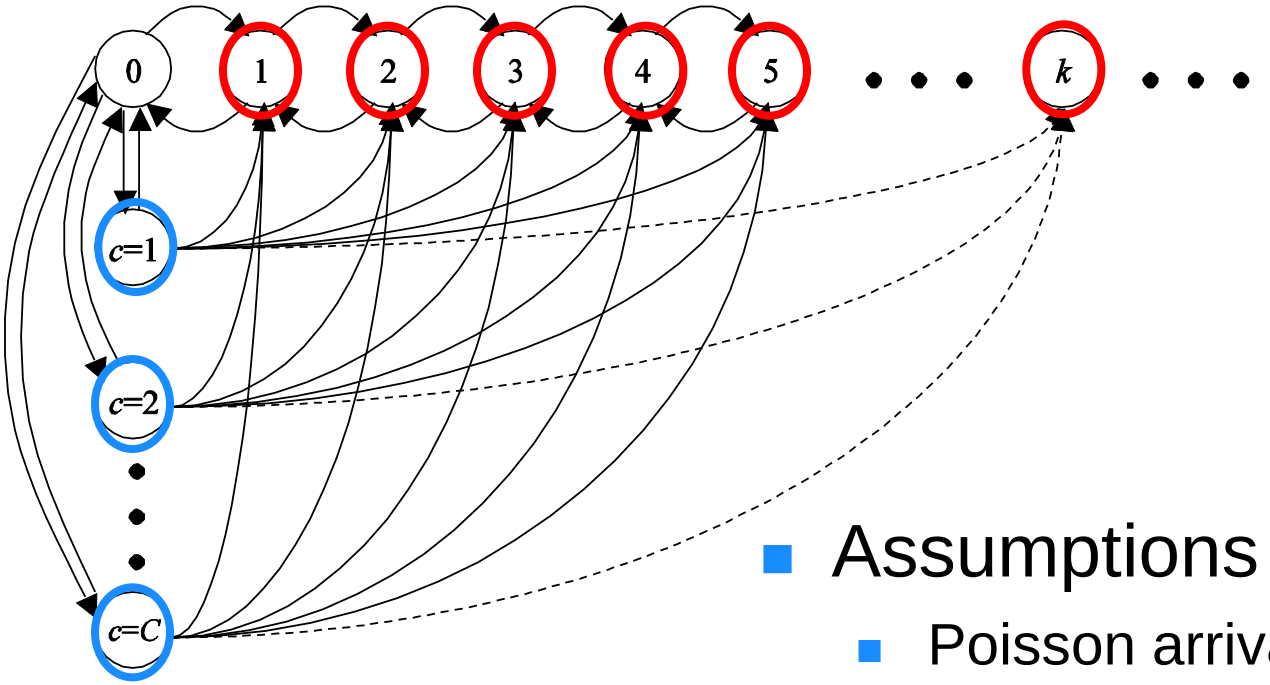
Percentile analysis

- State probabilities



Percentile analysis

- State probabilities



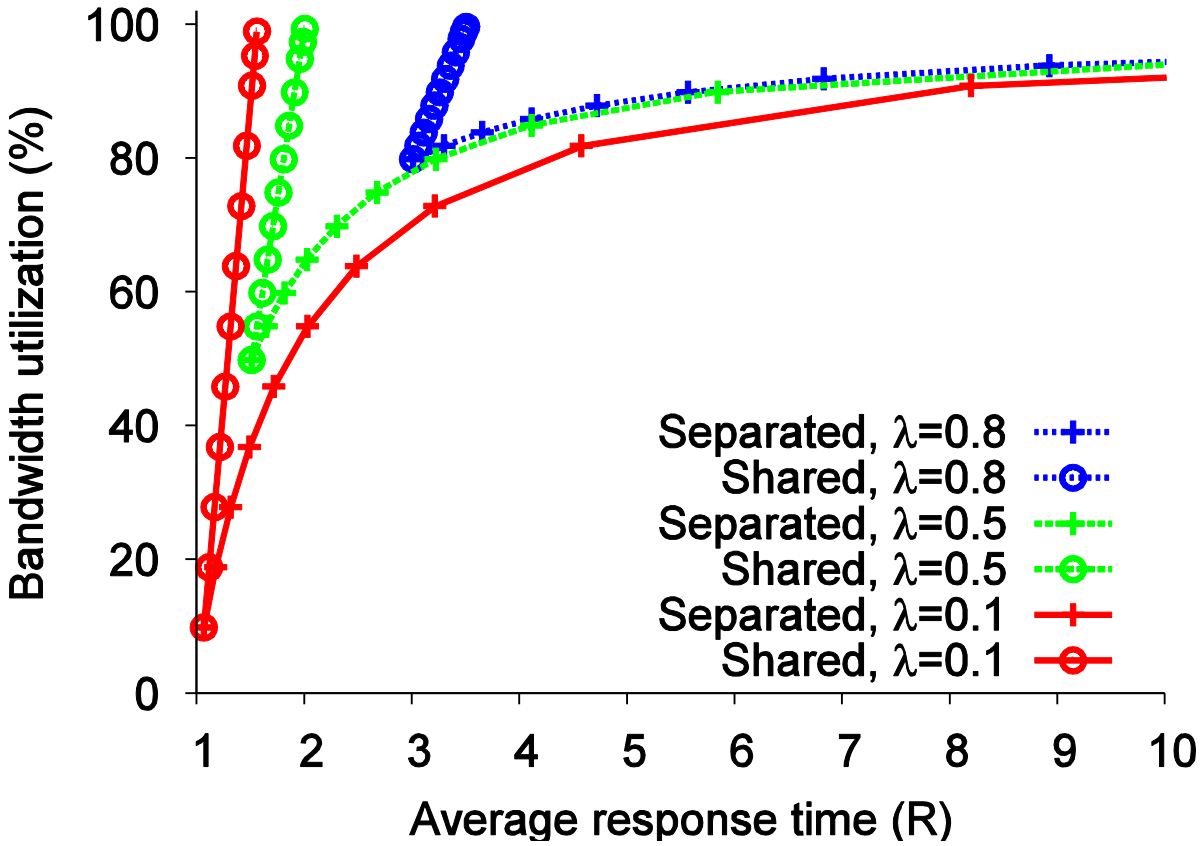
- Assumptions
  - Poisson arrivals
  - Exponential service
- Solve for p_k and q_c

Percentile analysis

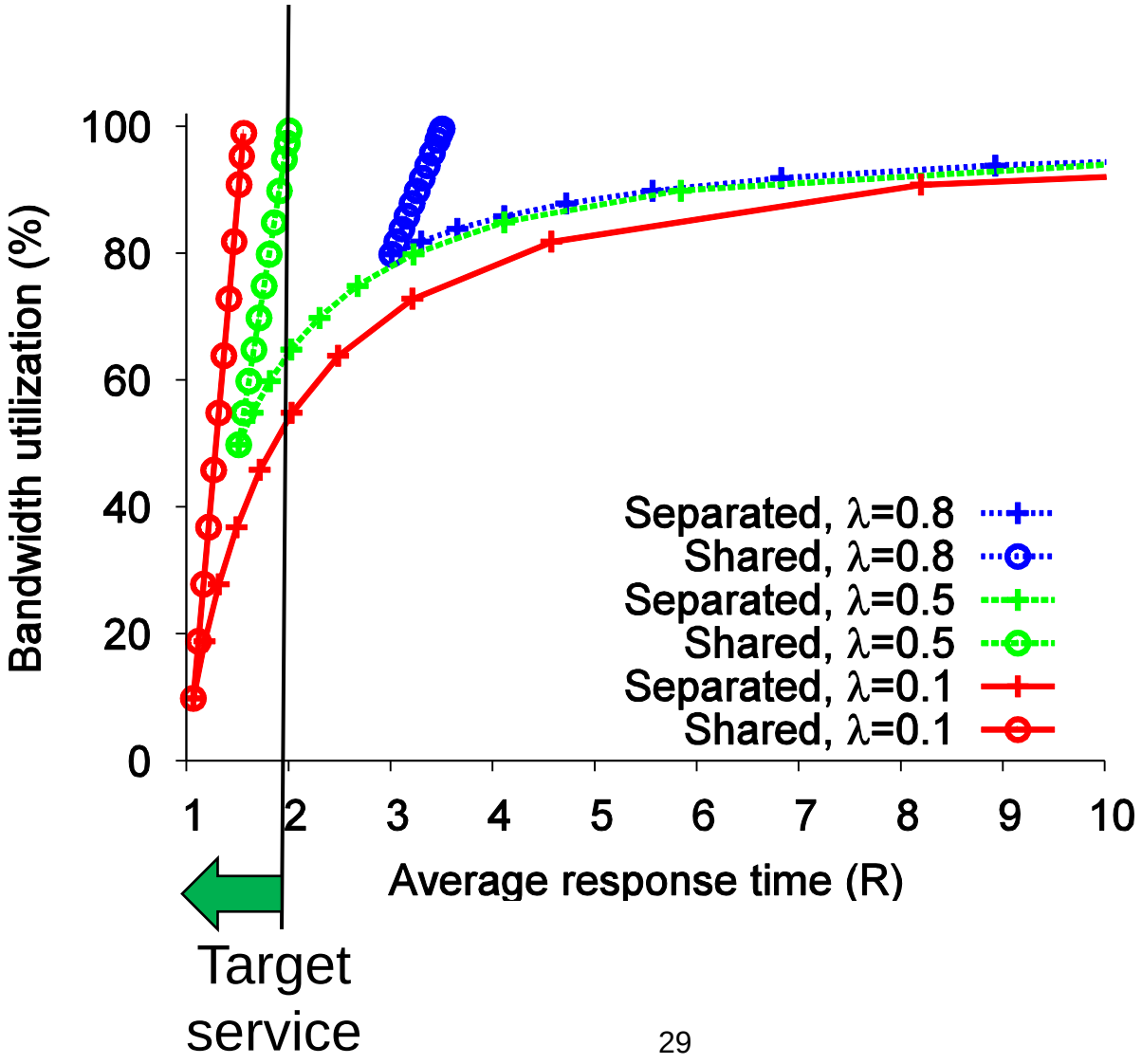
- Waiting time distribution
- PASTA
  - Poisson arrivals see time averages
- Sum of distributions

$$f(w) = \sum_k p_k f_k(w) + \sum_c p_c g_c(w)$$

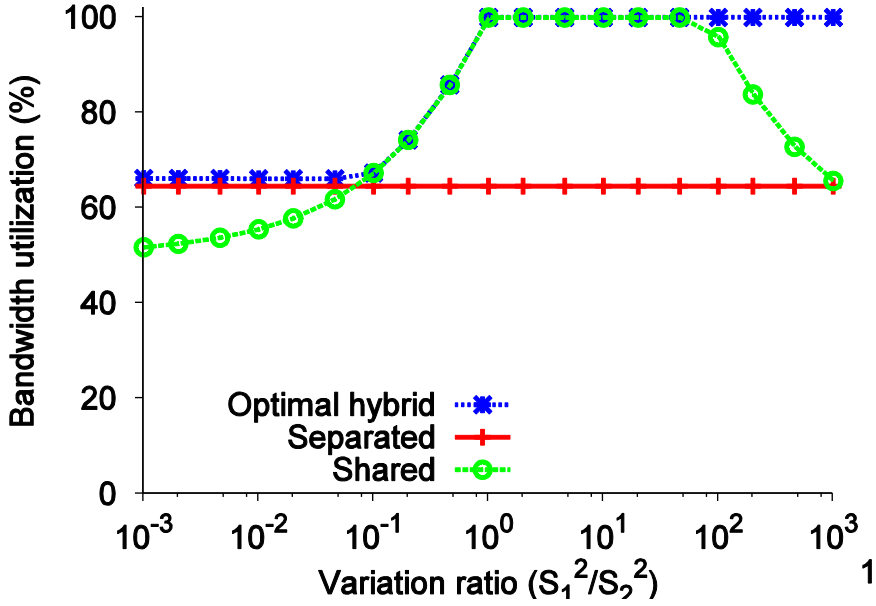
Example Results

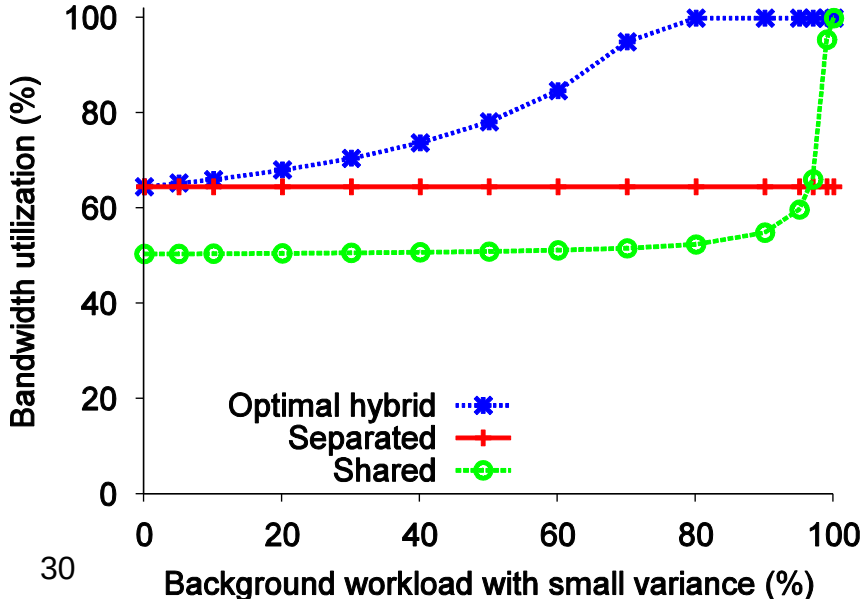


Example Results

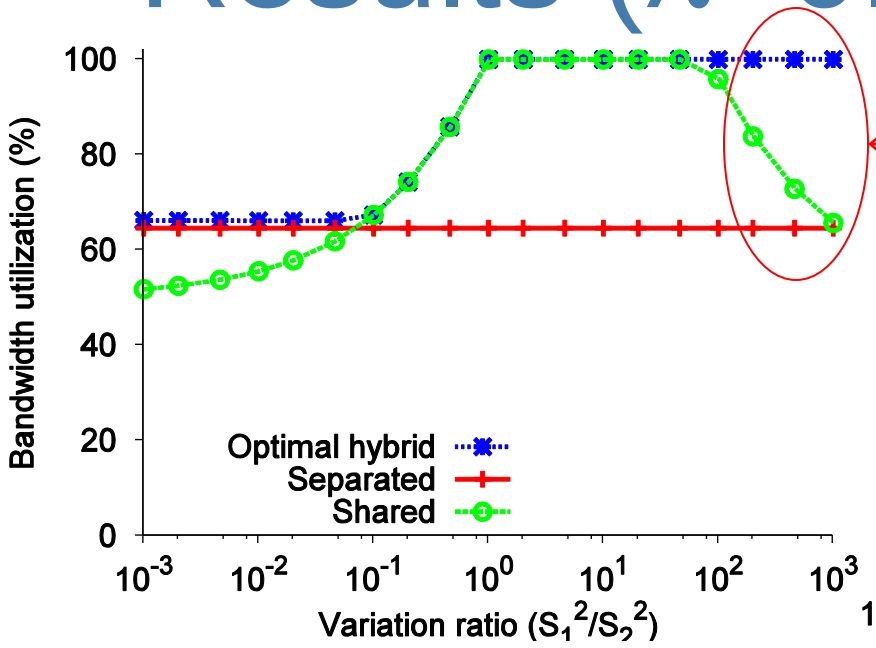


Results (λ=0.5; $R \leq 2$)

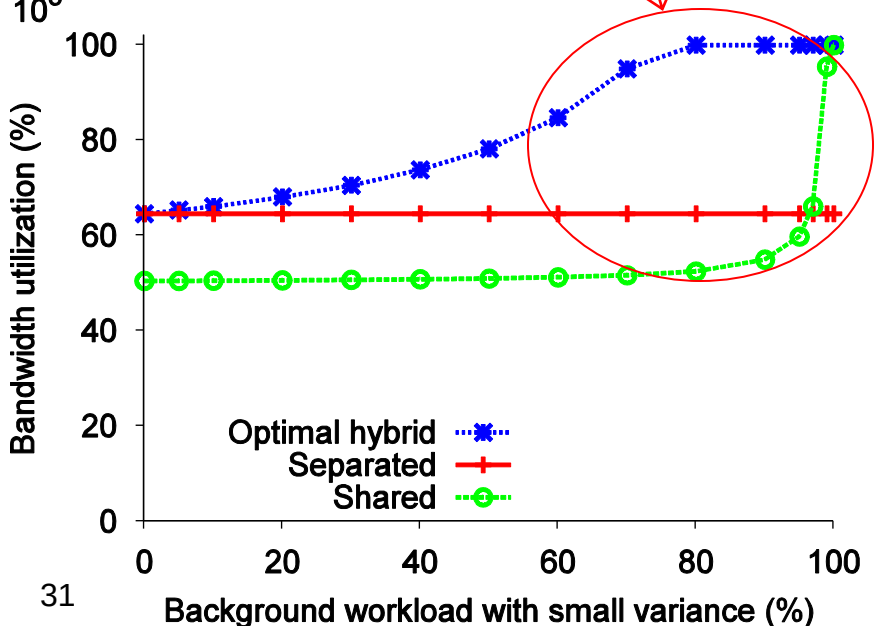




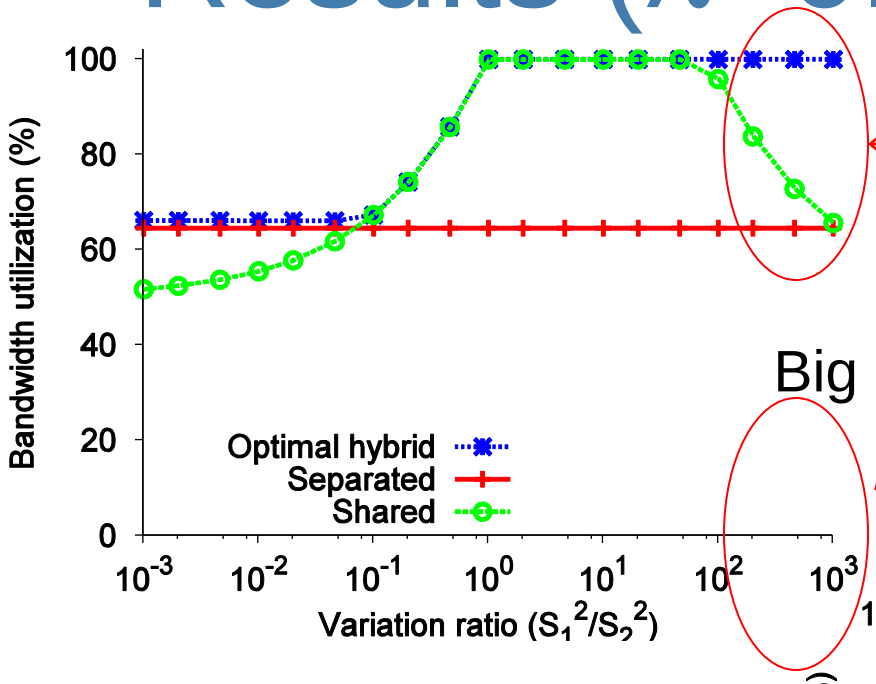
Results (λ=0.5; $R \leq 2$)



30% improvement

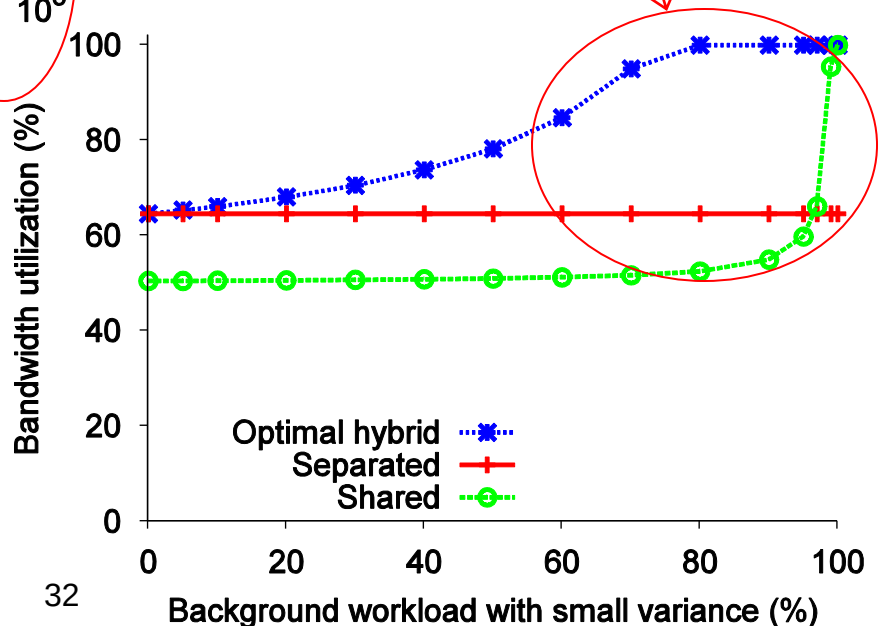


Results (λ=0.5; $R \leq 2$)

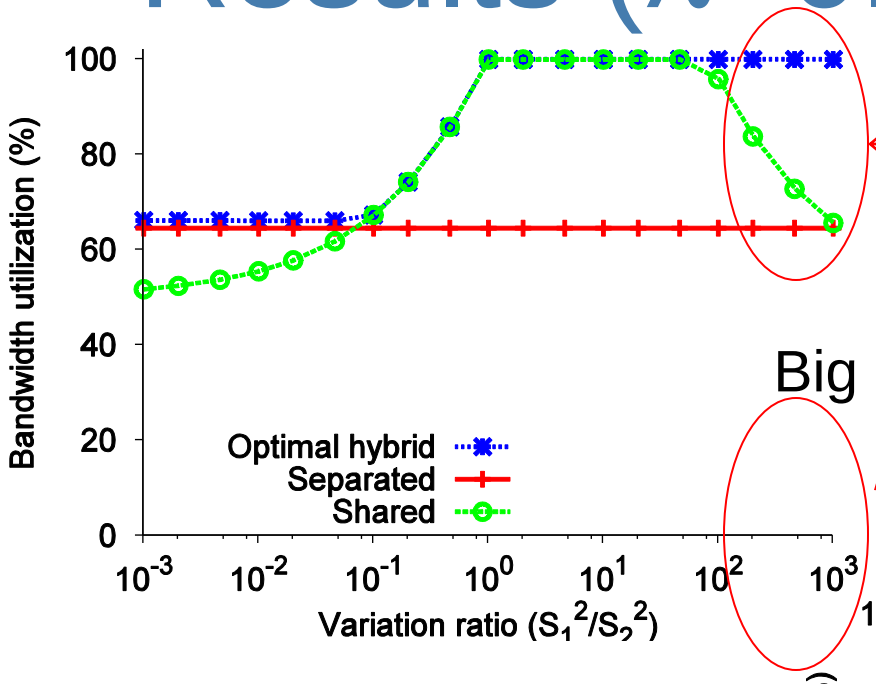


30% improvement

Big U^2/U difference

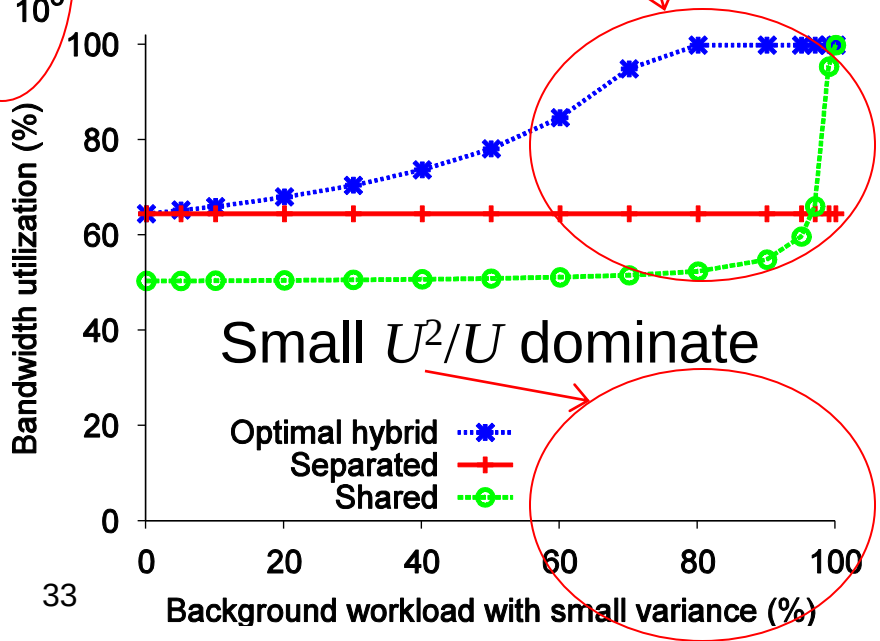


Results (λ=0.5; $R \leq 2$)



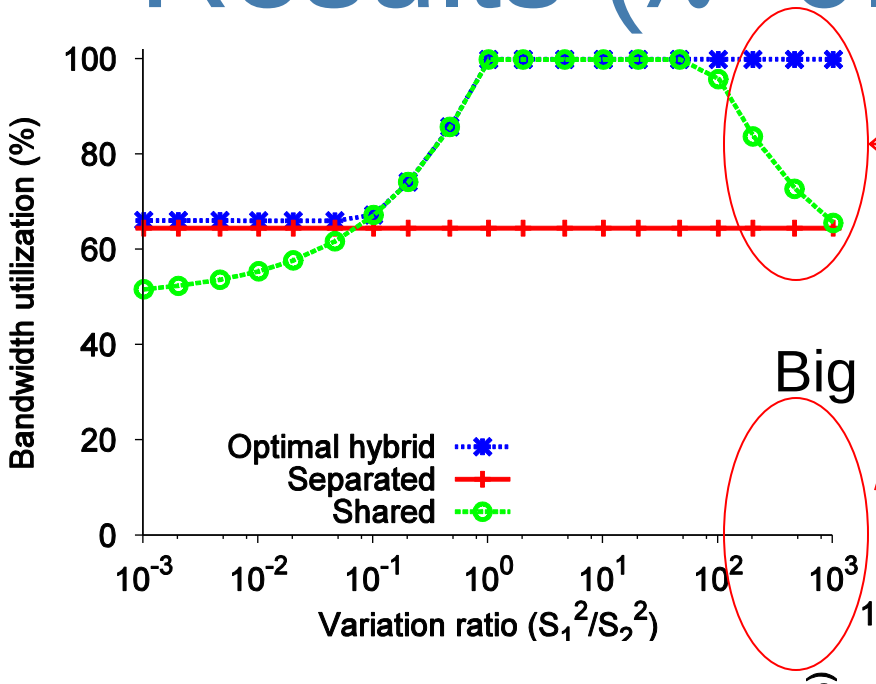
30% improvement

Big U^2/U difference



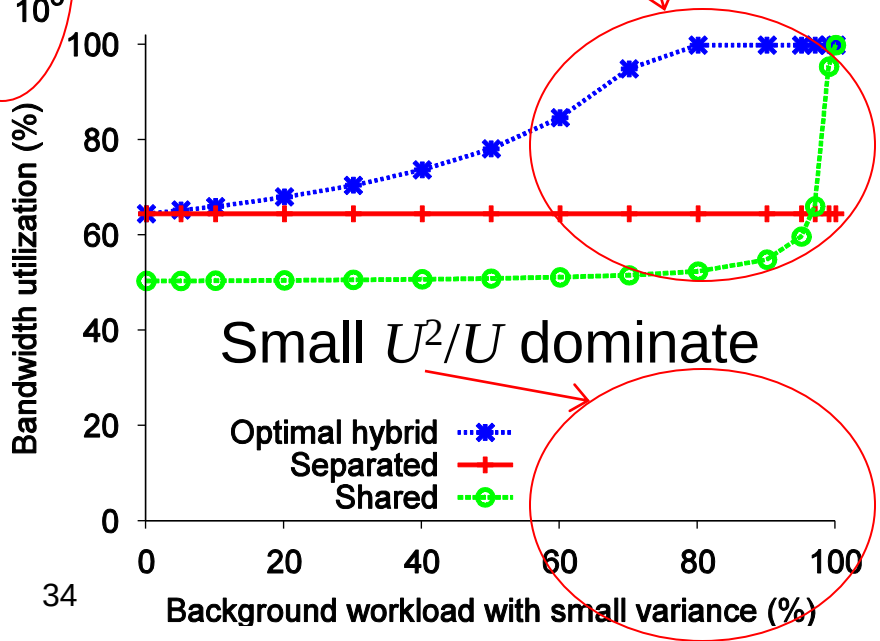
Small U^2/U dominate

Results (λ=0.5; $R \leq 2$)



30% improvement

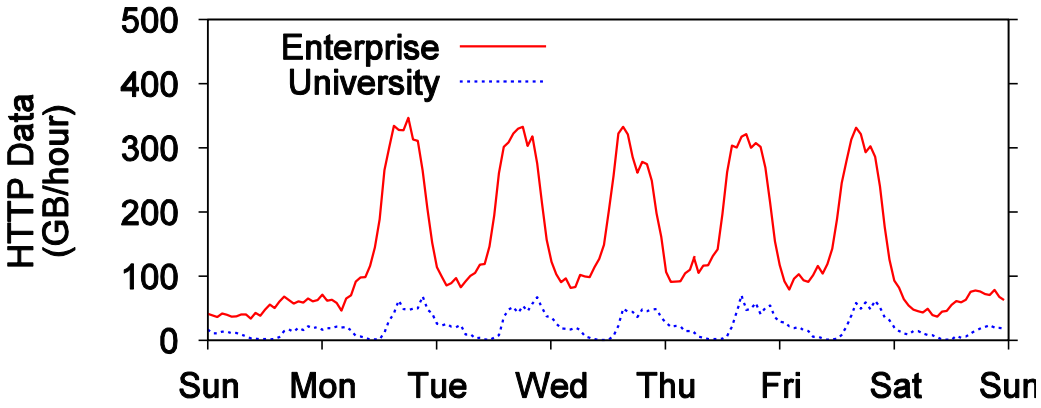
Big U^2/U difference

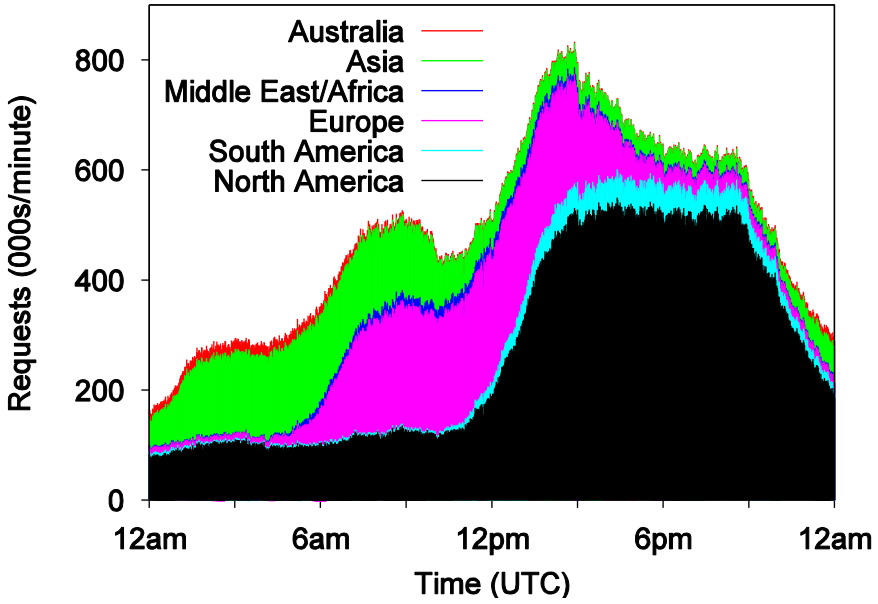


Small U^2/U dominate

- Small *job-size variability* primary (shared)
- Large *job-size variability* secondary (separated)

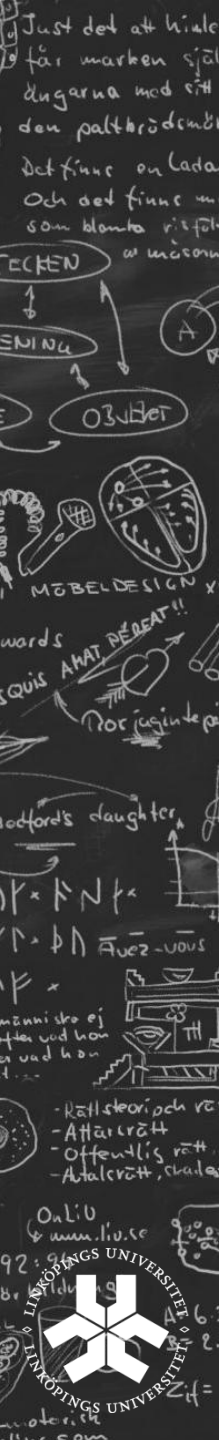
Diurnal traffic patterns





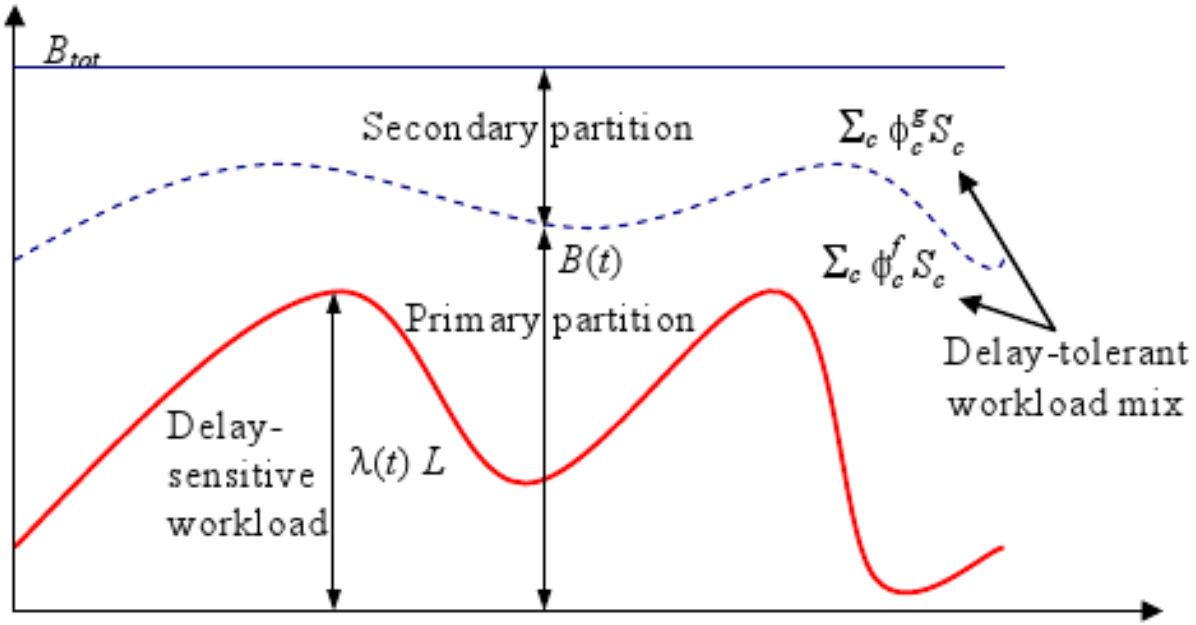
Workload management

- Maximize server resource usage
  - Prioritized delay-sensitive workload(s)
  - Background delay-tolerant workload(s)
- Workload management
  - Split vs. shared resources



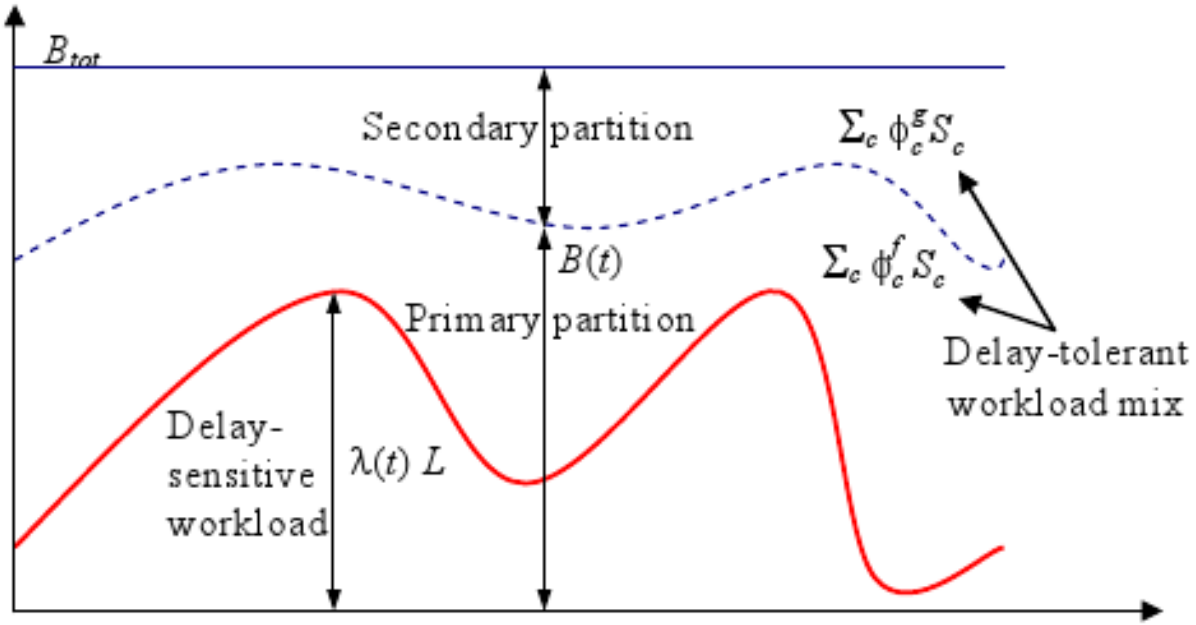
Workload management

- Maximize server resource usage
  - Prioritized delay-sensitive workload(s)
  - Background delay-tolerant workload(s)
- Workload management
  - Split vs. shared resources



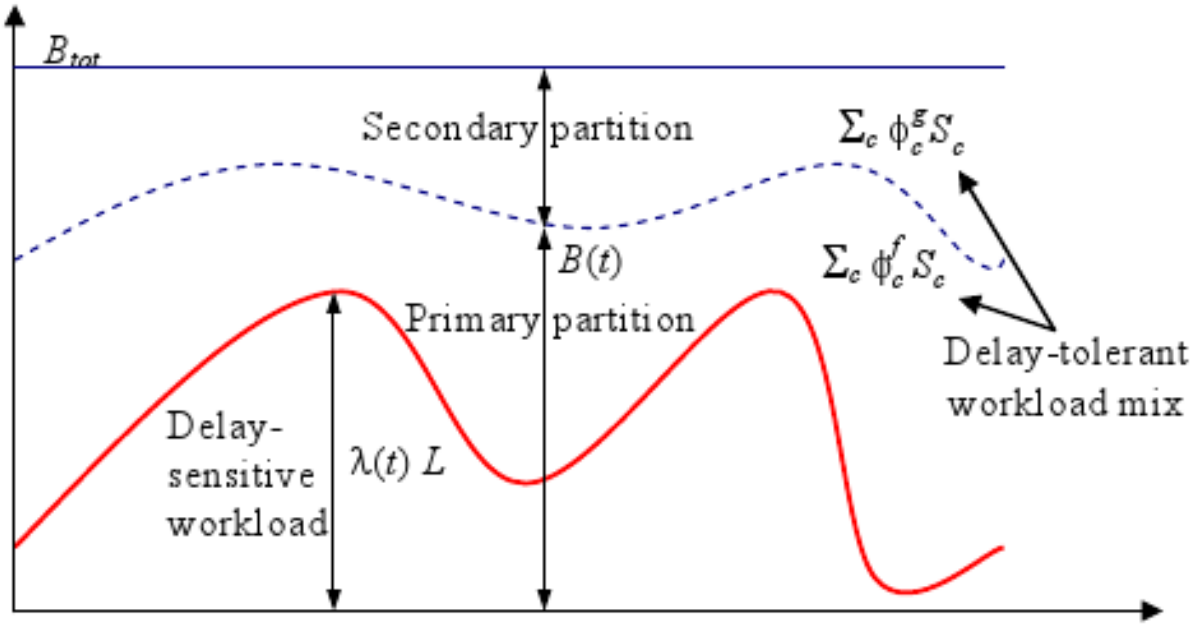
Basic policy classes

- Two dimensions



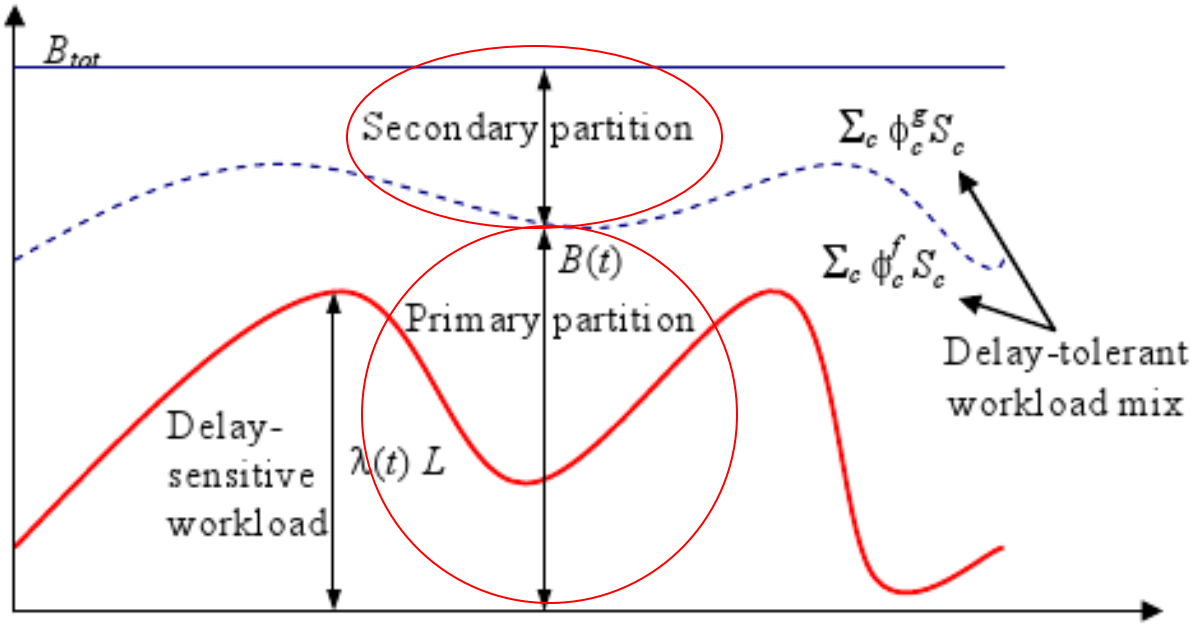
Basic policy classes

- Two dimensions
  - Adaptive vs static bandwidth partitioning



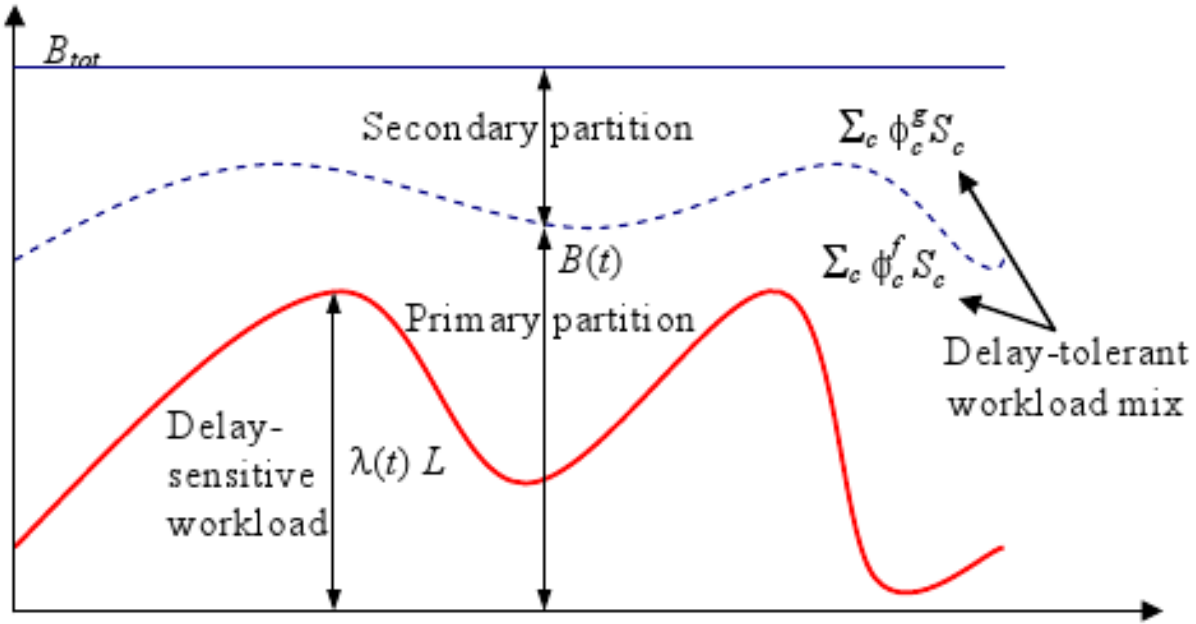
Basic policy classes

- Two dimensions
  - Adaptive vs static bandwidth partitioning



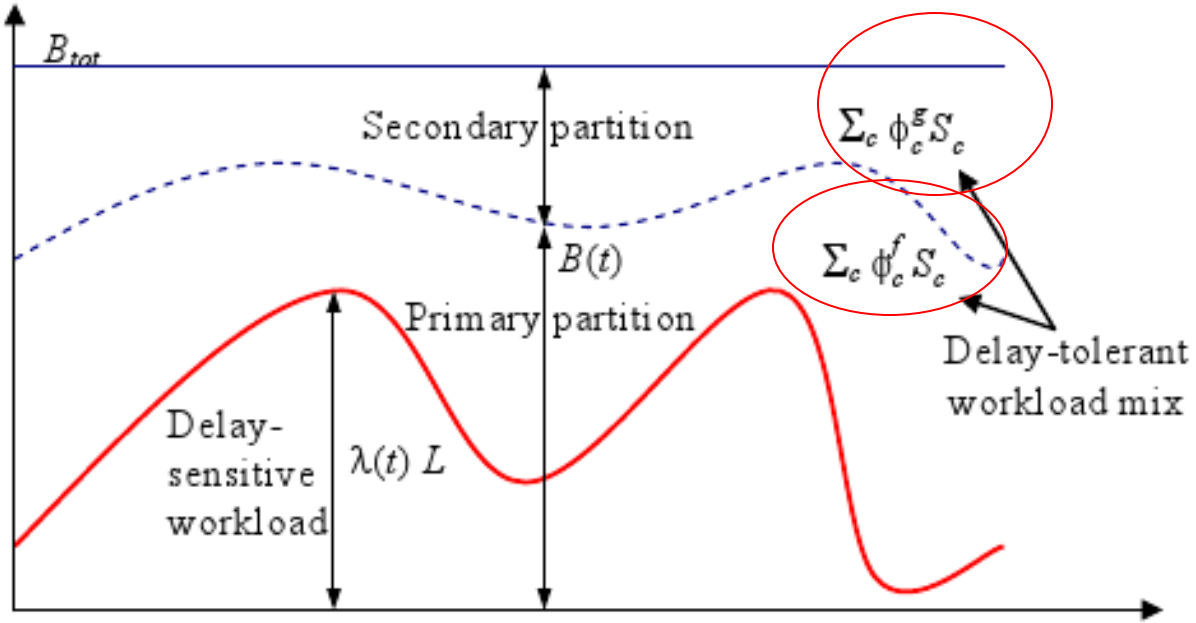
Basic policy classes

- Two dimensions
  - Adaptive vs static bandwidth partitioning
  - Adaptive vs static mix of delay-tolerant workloads



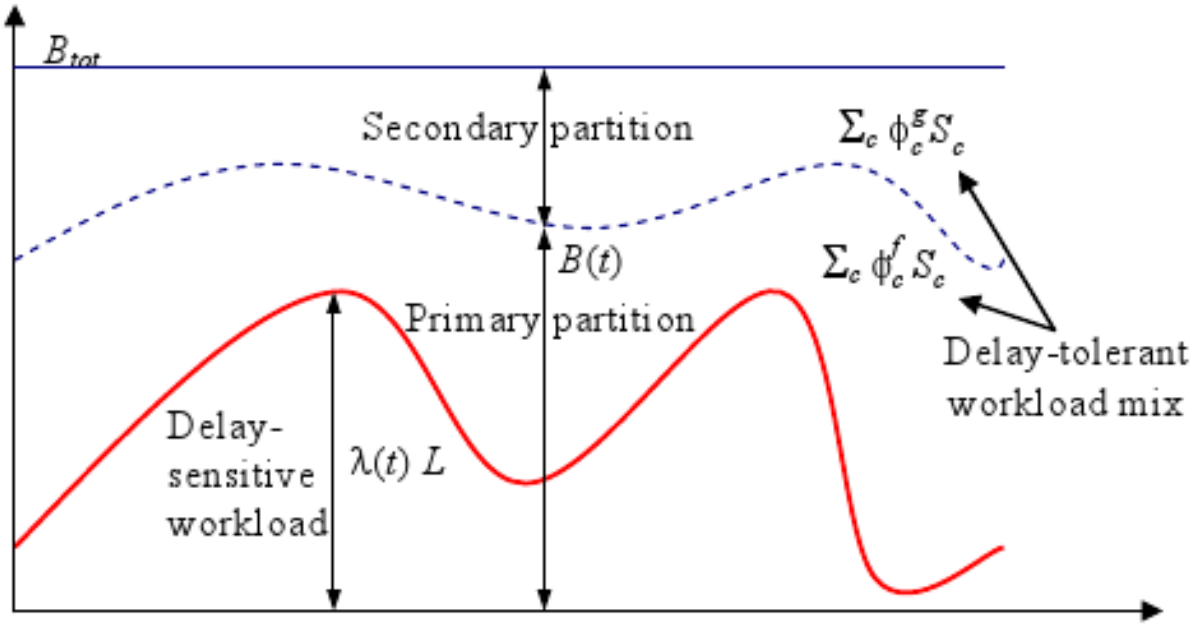
Basic policy classes

- Two dimensions
  - Adaptive vs static bandwidth partitioning
  - Adaptive vs static mix of delay-tolerant workloads



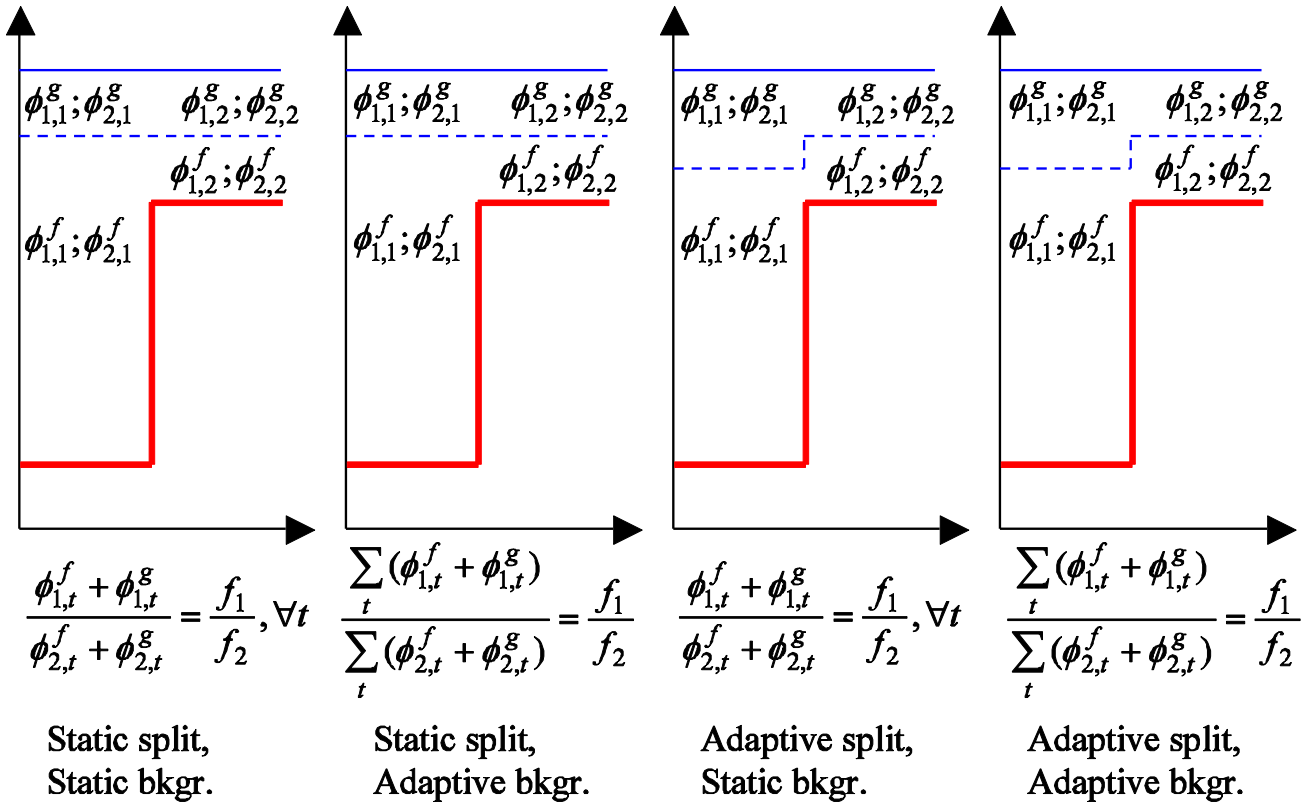
Basic policy classes

- Two dimensions
  - Adaptive vs static bandwidth partitioning
  - Adaptive vs static mix of delay-tolerant workloads



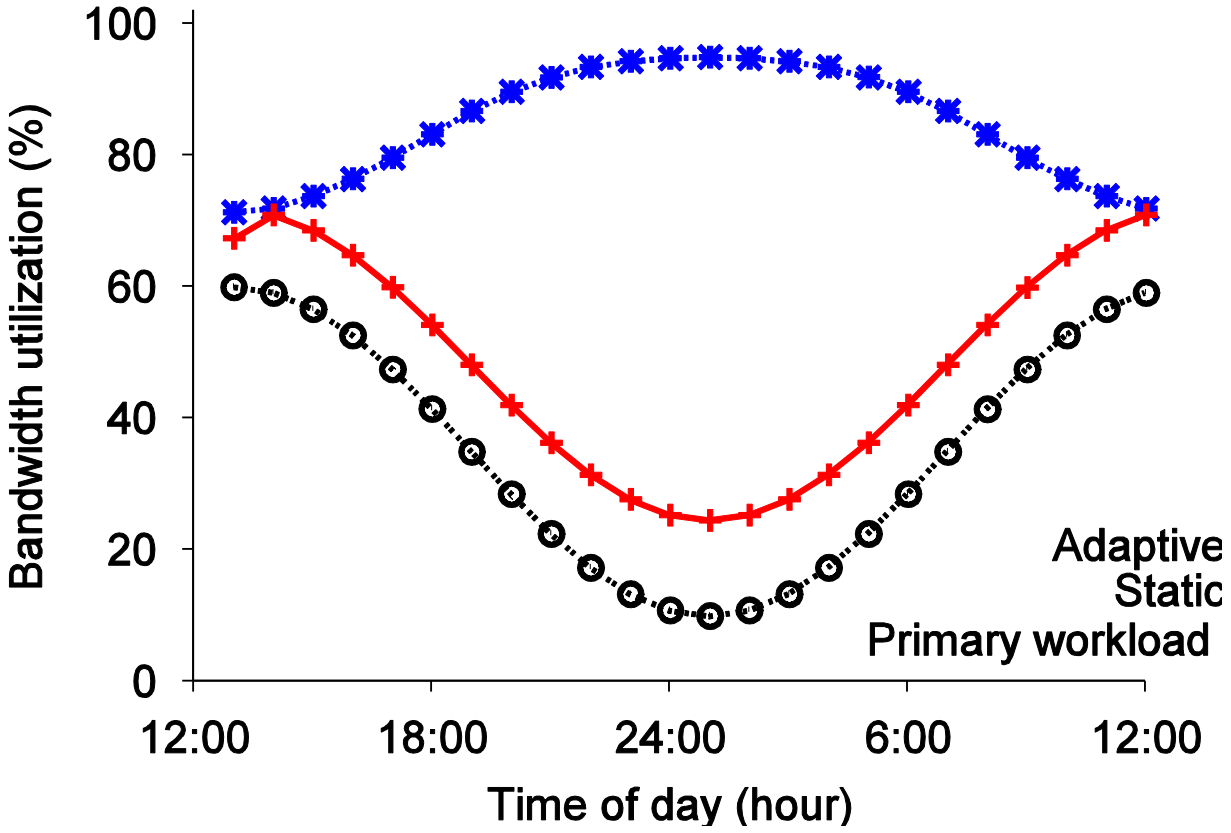
Basic policy classes

- Two dimensions
  - Adaptive vs static bandwidth partitioning
  - Adaptive vs static mix of delay-tolerant workloads

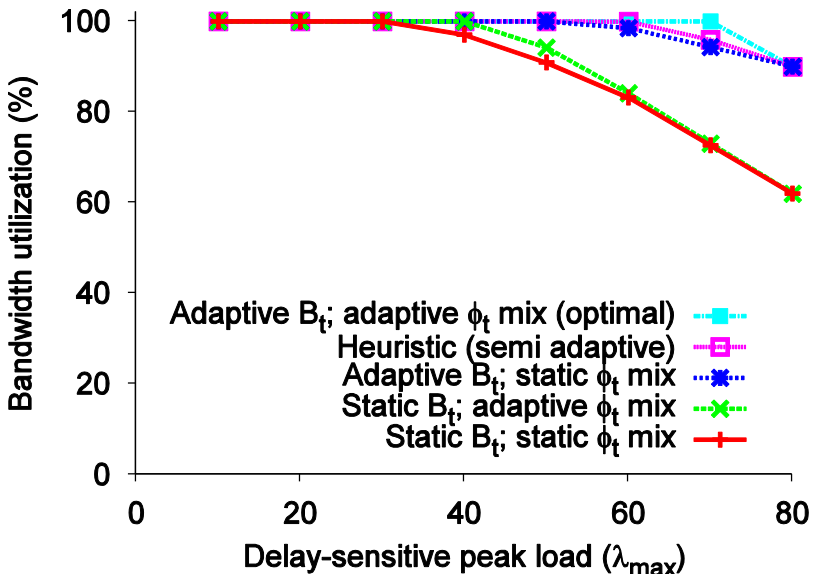


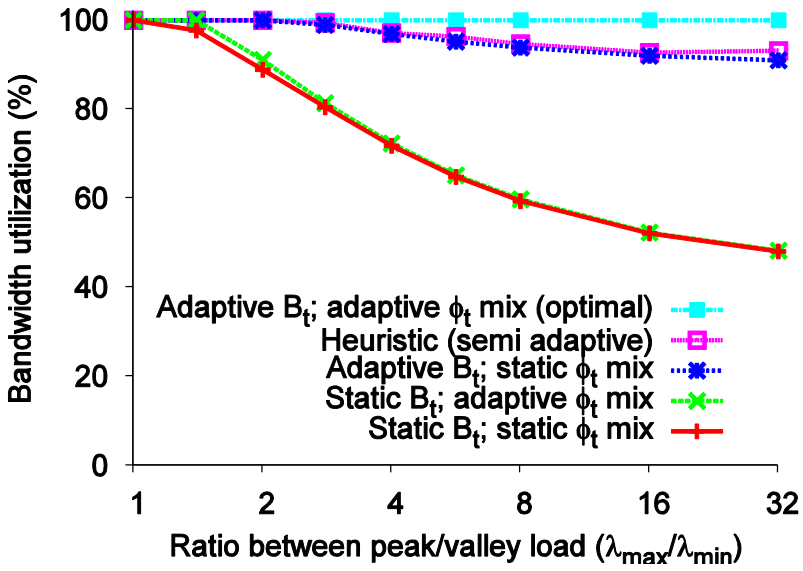
Bandwidth partitioning

- Adaptive vs static bandwidth partitioning

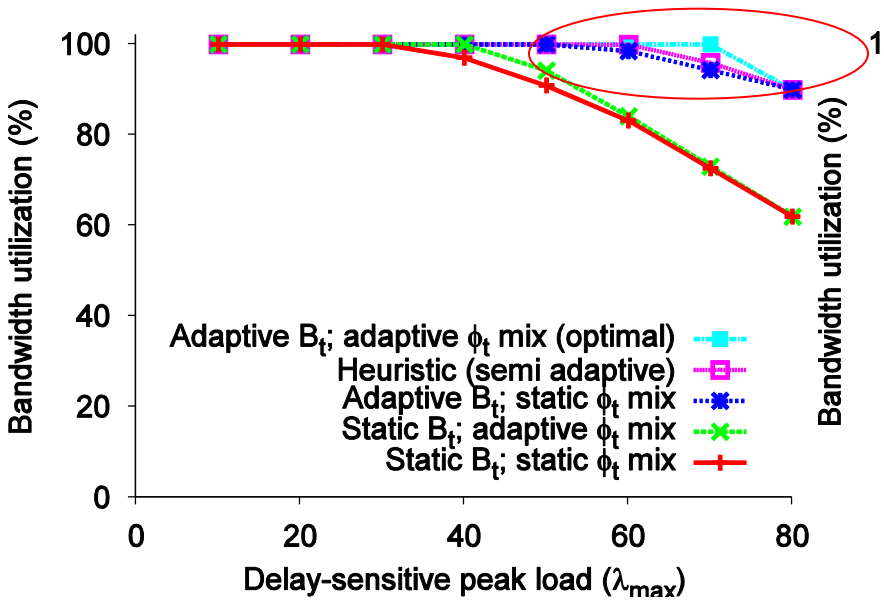


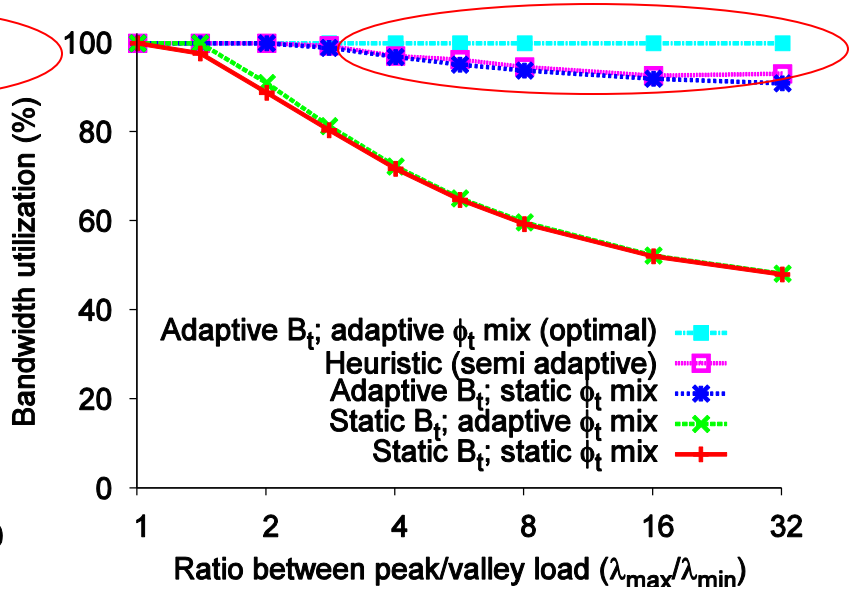
Policy comparison



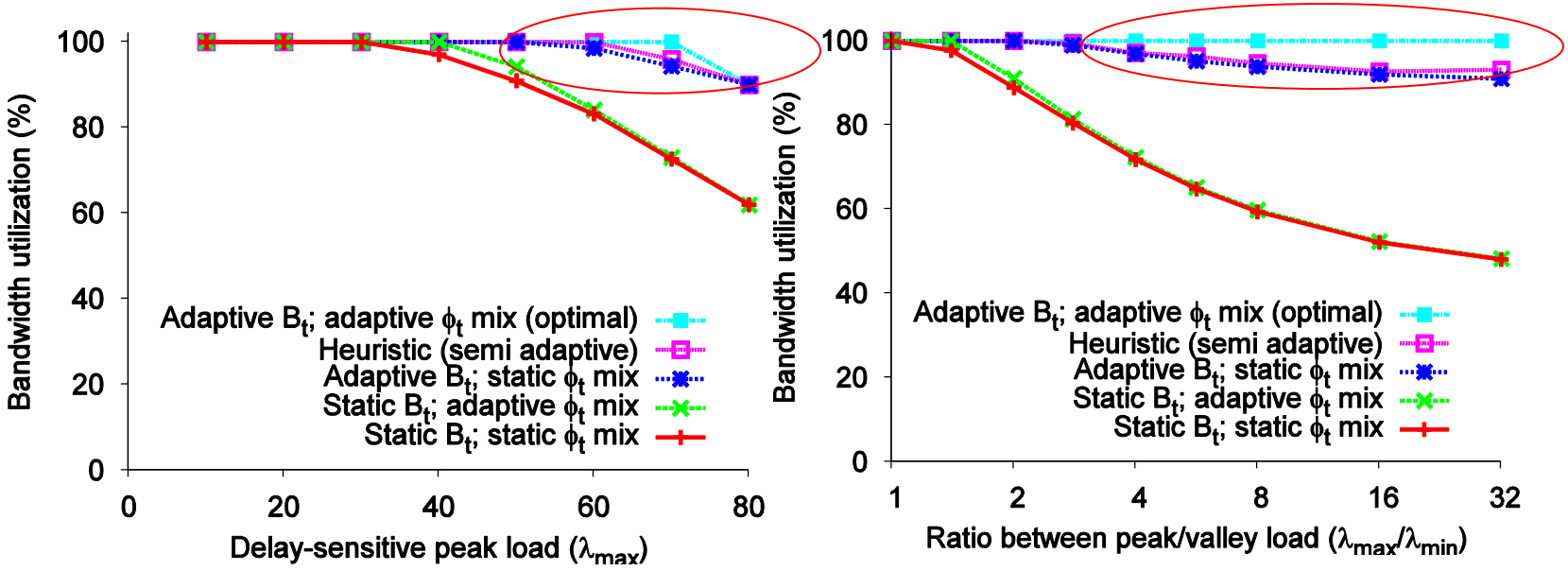


Policy comparison





Policy comparison



- Most of the benefits achieved with adaptive bandwidth partitioning
  - Less gained by adapting mix of delay-tolerant workloads

Conclusions

- Case for better resource utilization …
  - Value creation per TCO (or other “cost”)
- Utilizations improvements
  - Small job-size variability (U^2/U) → primary (shared)
  - Large job-size variability (U^2/U) → secondary (separated)
- Great value in careful workload scheduling and server-resource management
  - Most benefits with adaptive bandwidth partitioning
  - Less gained by adapting mix of delay-tolerant workloads

Thank you!

Niklas Carlsson

Email: niklas.carlsson@liu.se

Martin Arlitt

Email: martin.arlitt@hp.com