CPSL
Cyber-Physical Systems Laboratory

Predicting Latency Distributions of Aperiodic Time-Sensitive Services

Haoran Li, Chenyang Lu, Christopher Gill
Cyber-Physical Systems Laboratory
Department of Computer Science & Engineering
https://www.cse.wustl.edu/~lu/

Washington University in St. Louis

Time-Sensitive Services on Cloud

- Time-Sensitive Services
  - Interactive Services
    - Google Doc
  - Real-time Database Services
    - Redis on Amazon ElastiCache

- Response Time Matters
  - Meeting Service Level Objective (SLO)
    - Tail Latency: (e.g. 90th percentile latency less than 10ms)

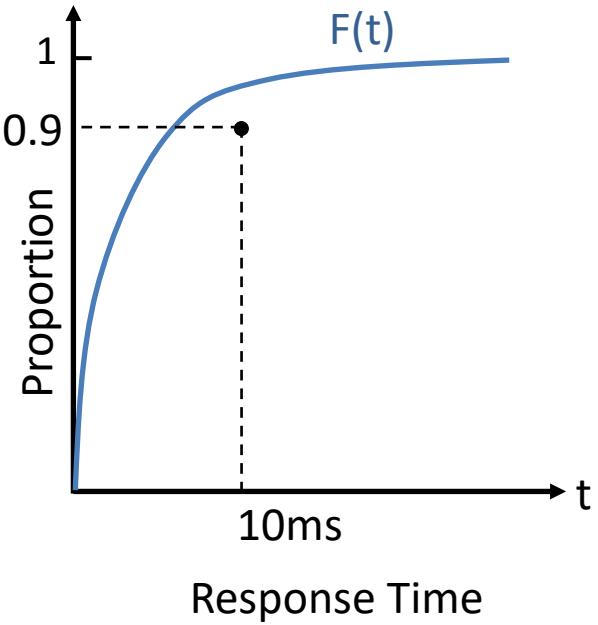
Motivation: How do we configure the Cloud to meet latency SLO?

Source: https://www.veracentra.com/modeling-and-analytics/
https://aws.amazon.com/elasticache/

Response Time Analysis

- Response Time Distribution
  - Response Time: Random Variable *R*
  - Cumulative Distribution Function (CDF)

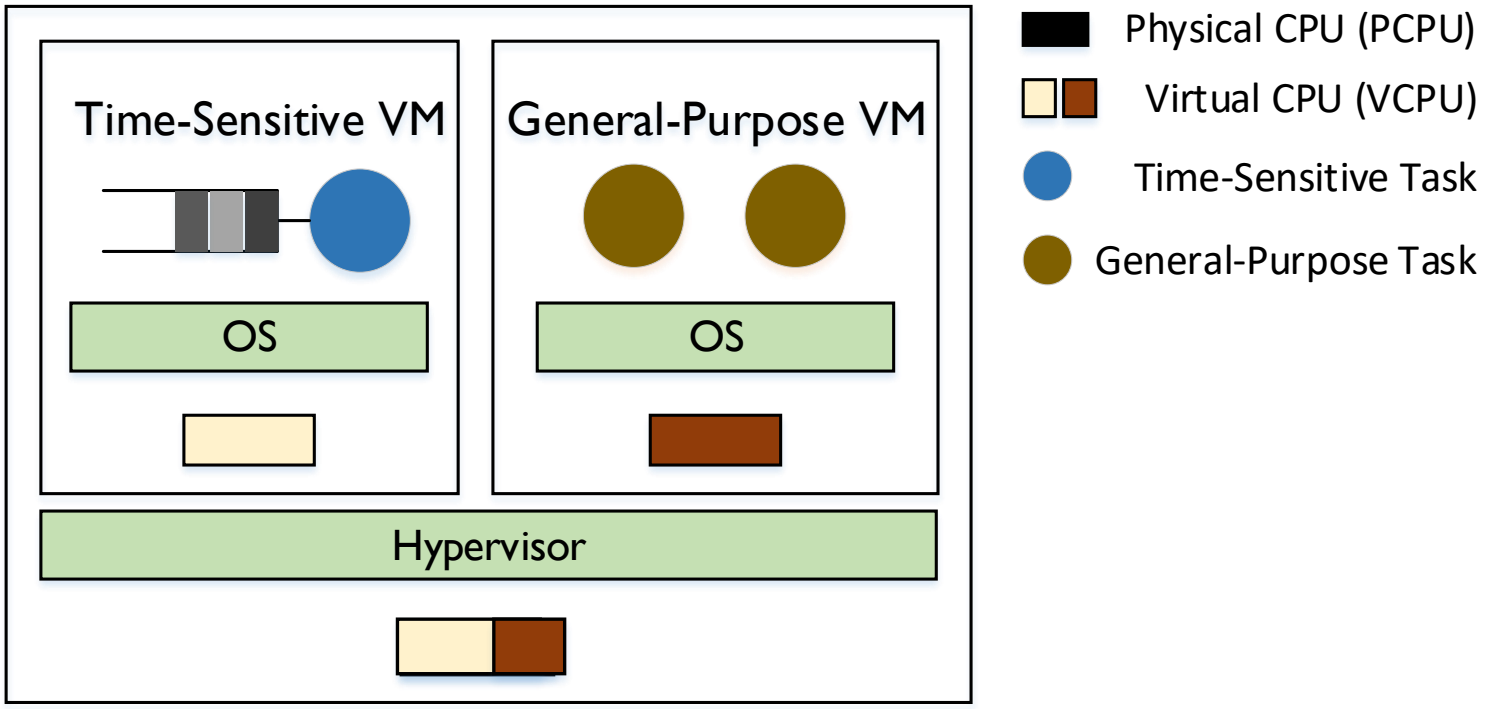
$$\mathrm{F(t)} = \Pr\{R < t\}$$



- Tail Latency SLO
  - "90th Latency should be less than 10ms"
  - A point (10ms, 0.9) on Response Time CDF

Problem: How to configure the scheduler to meet the tail latency SLO of a time-critical service sharing a CPU with other services?

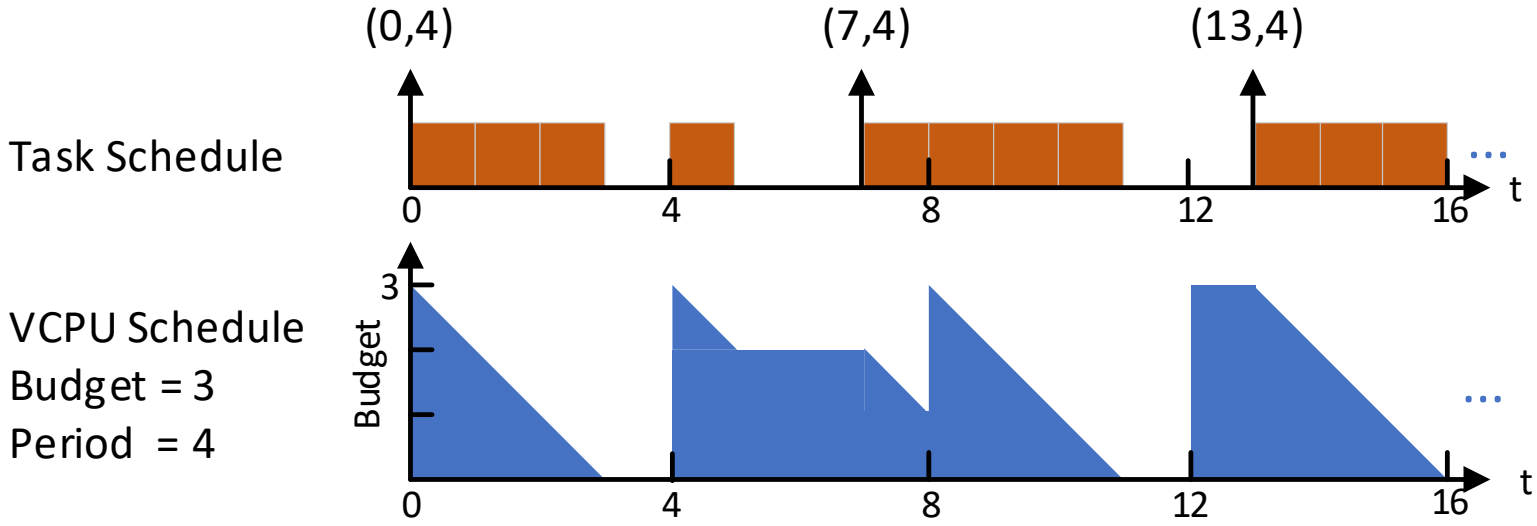
Time Sensitive Task within a Virtual Machine



- Hypervisor schedules VCPUs on PCPUs.
  - PCPU Shared by VCPUs
- The VCPU of time-sensitive service has **the highest priority.**
  - Resource Isolation: Avoid Time-sensitive service to monopolize the CPU

We need a mechanism to limit the CPU cycles consumed by the time-sensitive service.

VCPU scheduled as a Deferrable Server (DS)



- Deferrable Server is widely adopted for CPU resource isolation:
  - Flexible and Simple
  - RT-Xen[1], vMPCP[2] on KVM
- A Deferrable Server has two parameters ***<budget, period>***
  - The server consumes budget when executing jobs
  - When the budget exhausted, the server stops executing jobs
  - Budget replenishes at the start of each period

How do we configure (B, P) to meet tail latency SLO?

[1] Xi, Sisu, et al. "Real-time multi-core virtual machine scheduling in xen." *2014 International Conference on Embedded Software (EMSOFT)*. IEEE, 2014.
[2] Kim, Hyoseung, Shige Wang, and Ragunathan Rajkumar. "vMPCP: A synchronization framework for multi-core virtual machines." *2014 IEEE Real-Time Systems Symposium*. IEEE, 2014

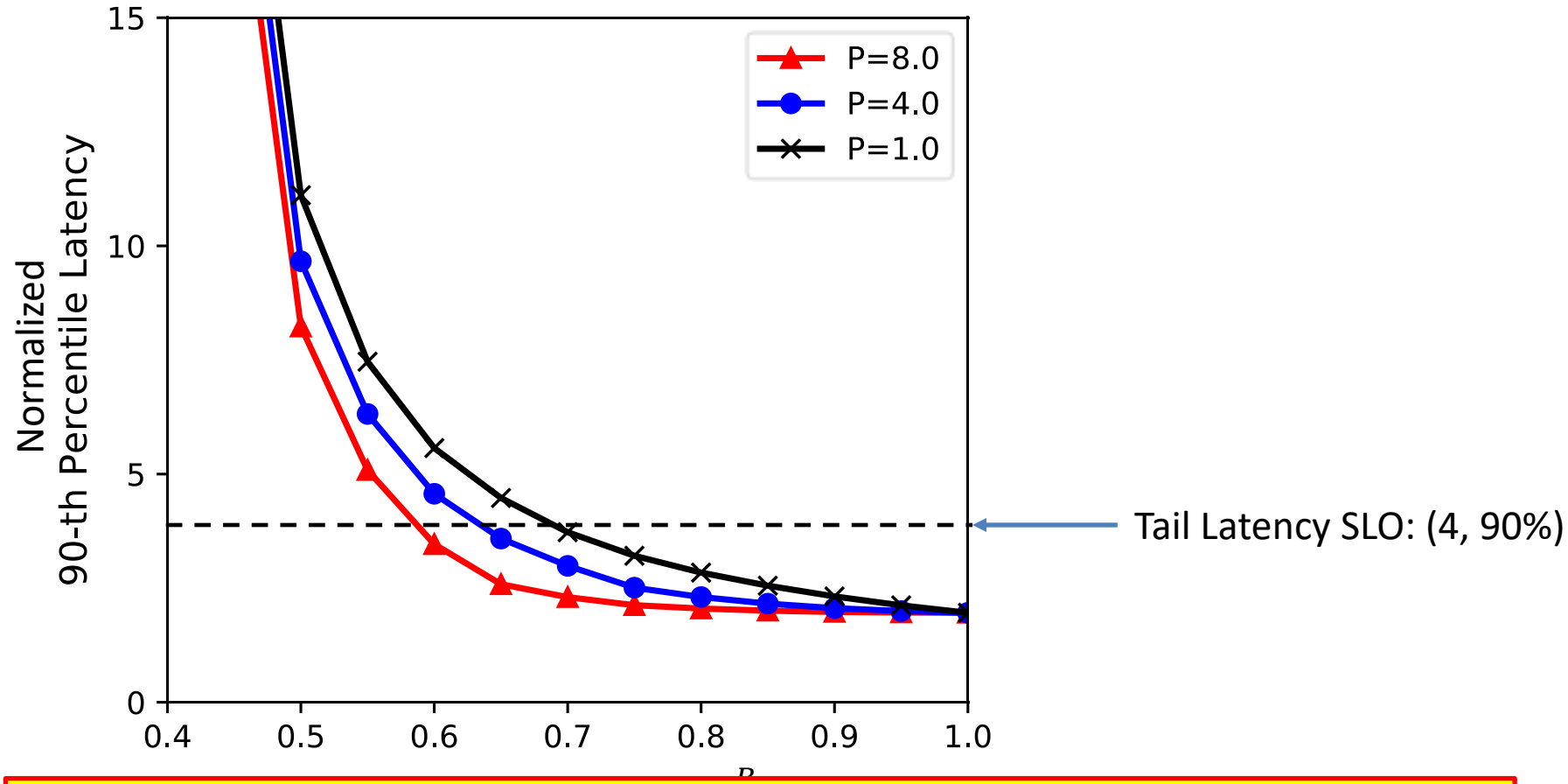
Problem Statement

- Given a time-sensitive service
  - Incoming job arrives as a Poisson Process (λ)
  - Each job needs a constant service time (d)
  - The execution of the service is governed by a deferrable server (B, P)

- Goal: predict the stationary response time distribution of the time-sensitive service with parameters (λ, d, B, P)

Meet Tail Latency Target

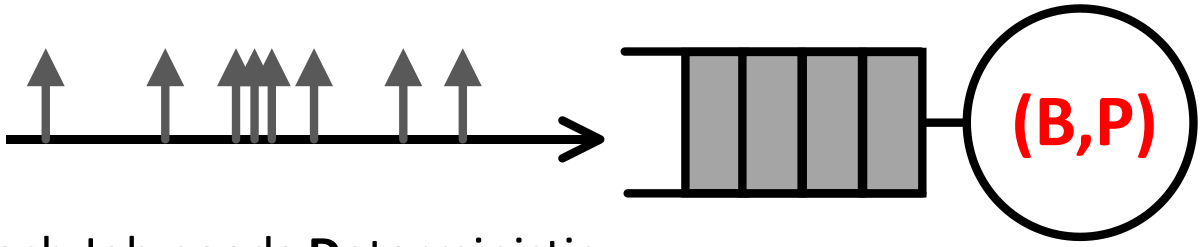
Response time distribution → scheduling parameters for tail latency



Simplify the configuration and management of time-sensitive services

A New Queueing Model: M/D(DS)/1

M/D(DS)/1



Poisson(λ)
– **M**emoryless

Each Job needs **D**eterministic Service time (d).

One Deferrable Server (B, P)

Goal: predict the stationary response time distribution of an M/D(DS)/1 queueing system with parameters (λ, d, B, P)

Open Challenges

- From the perspective of Real-Time Scheduling
  - ~~Periodic~~ Task Model

    Aperiodic: Poisson Process
  - Focused on ~~bounding the worst-case response time~~

    response time distribution
- From the perspective of queueing theory
  - ~~Always-On or Periodic On-Off~~ Model

    Deferrable Server Model

No existing solutions for our system.

The Numerical Solution Overview:

Discrete time approximation: Poisson Process ➔ Bernoulli Process

In this talk, we only highlight a key observation in this step.

Compute Virtual Waiting Time Distribution **at the start of each Period**.

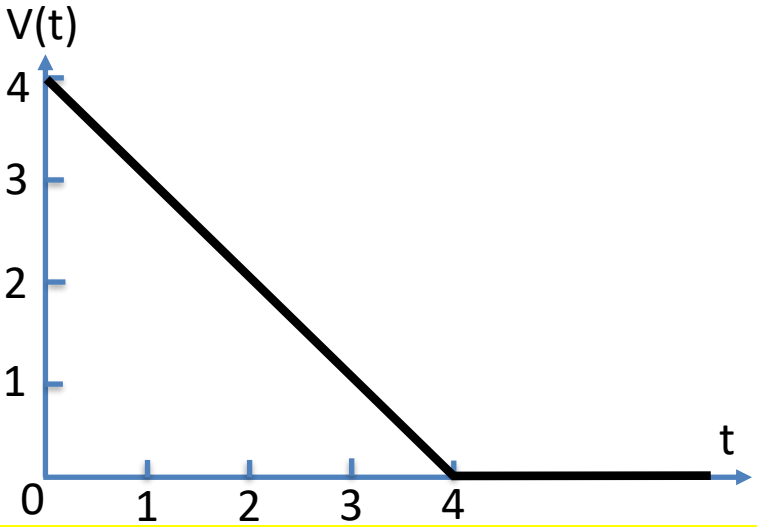
Recursively compute the virtual waiting time distribution at any time of each Period.

Compute Stationary Response Time Distribution of an M/D(DS)/1 Queue

Virtual Waiting Time

- Suppose at time 0, there are 4 people queueing at an ATM
  - Each person takes 1 time unit to deposit cash.

Q: **If** (**virtually**) you arrive at the ATM at time t,
how much time V(t) you should wait before you can start depositing?



Virtual Waiting Time: The pending jobs in a system at time t.

Virtual Waiting Time at the Start of a Period

- In a Deferrable Server (B, P), we need to consider a Joint Markov Chain (V, G)[n]
  - Two Random Variables:
    - V: Virtual Waiting Time
    - G: Remaining Budget

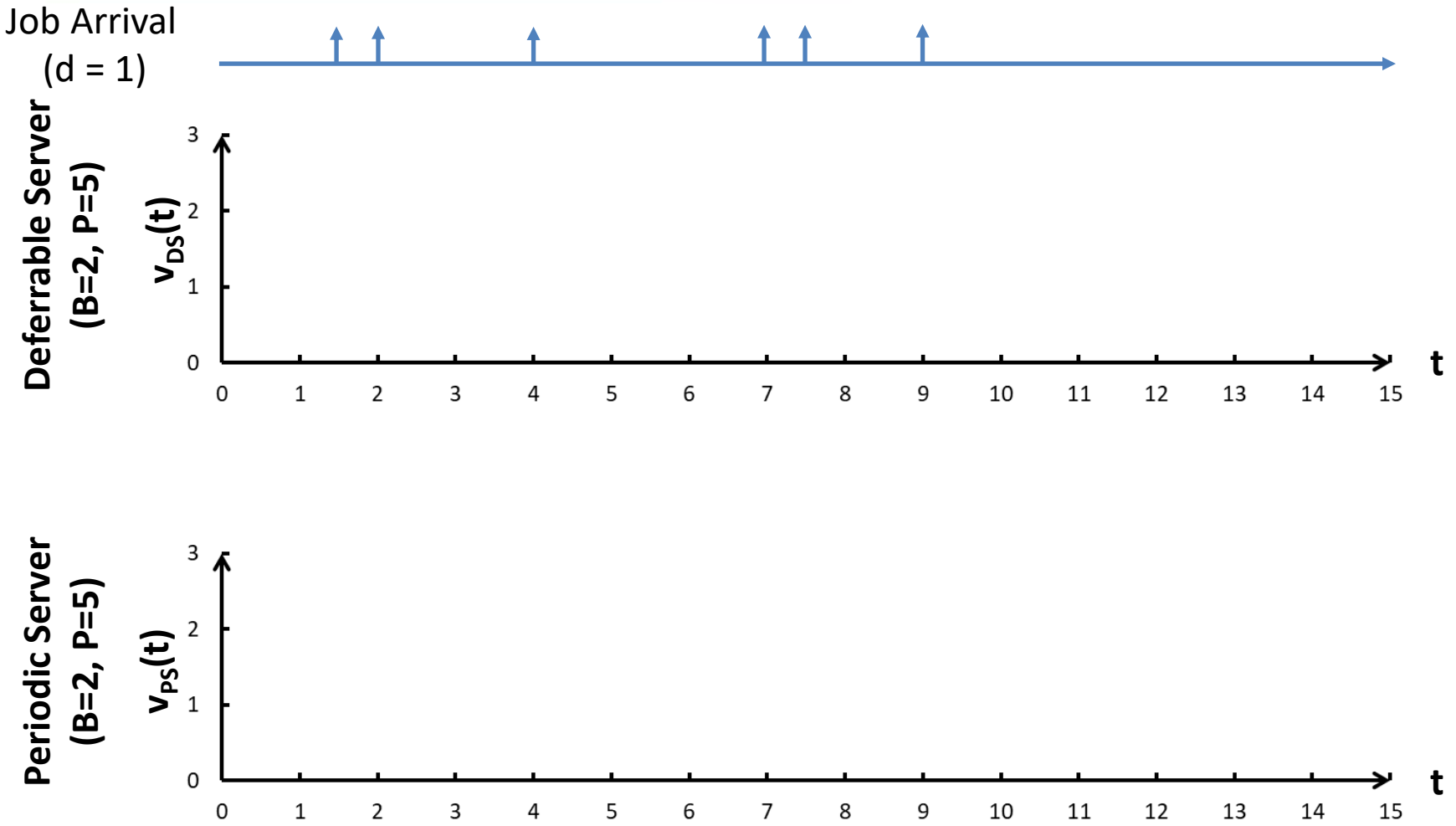
More states, harder to solve

- In a Periodic On-Off Server (B, P), we only need to consider a Markov Chain V'[n]
  - Periodic On-Off Server (B, P):
    - Off for (P-B) time unit; then on for B time unit

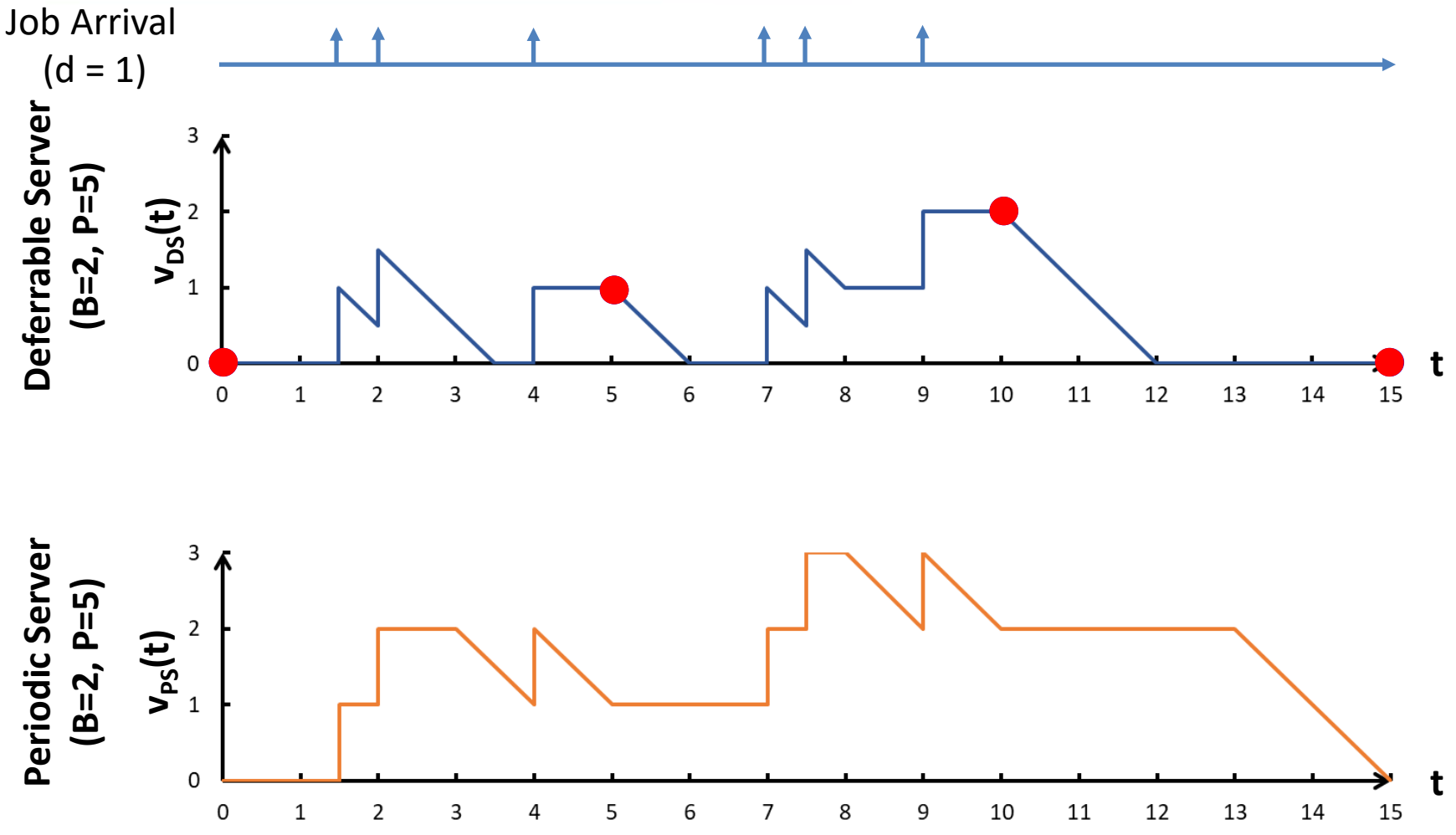
Less states, easier to solve

We observed, at the start of a period, the virtual waiting time of a DS is always equals to the one of a PS.

Virtual Waiting Time at the start of the Period



Virtual Waiting Time at the start of the Period



Theorem: In any arrival sequence, $v_{DS}(nP) = v_{PS}(nP)$

The Numerical Solution Overview:

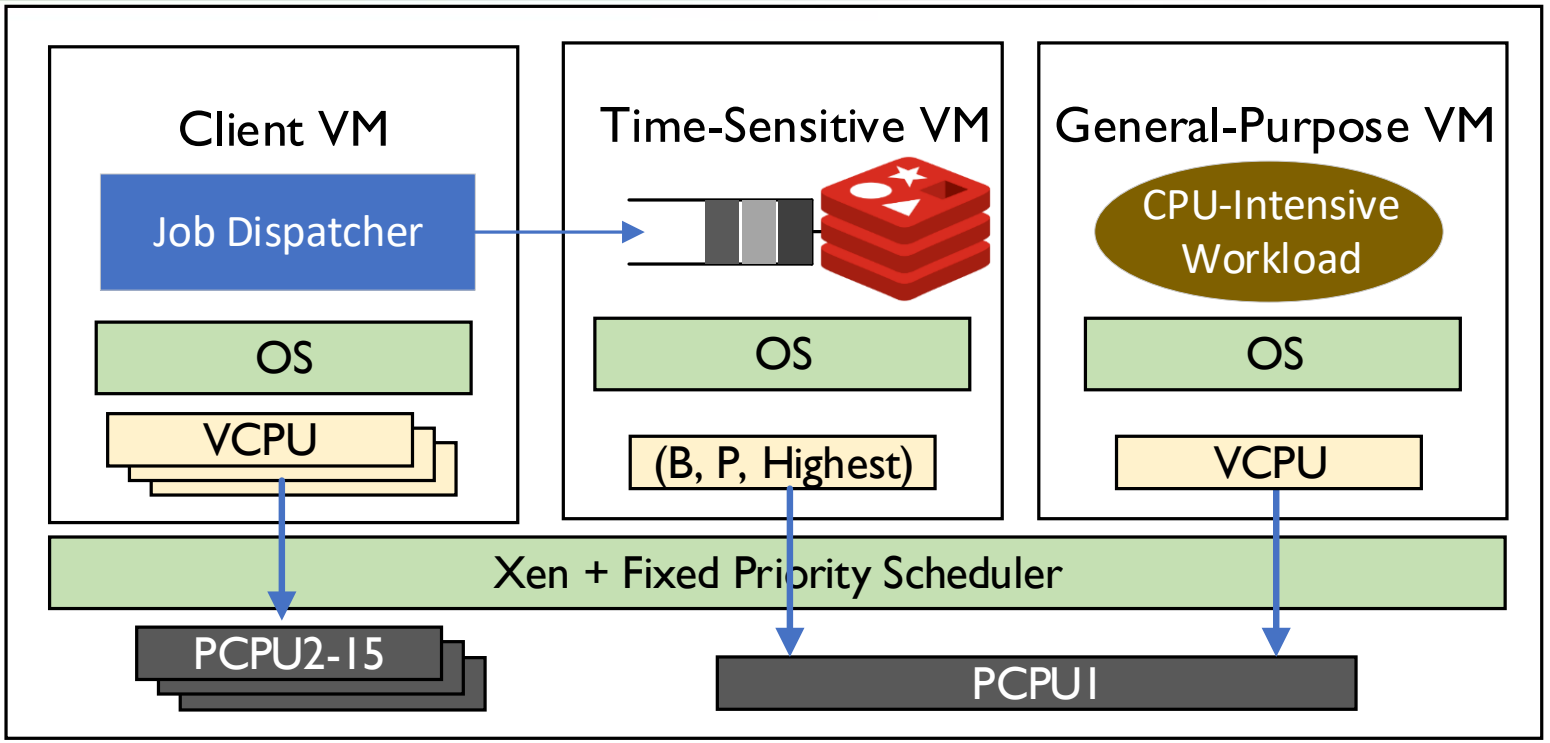
Discrete time approximation: Poisson Process ➔ Bernoulli Process

Compute Virtual Waiting Time Distribution **at the start of each Period**.

Recursively compute the virtual waiting time distribution at any time of each Period.

Compute Stationary Response Time Distribution of an M/D(DS)/1 Queue

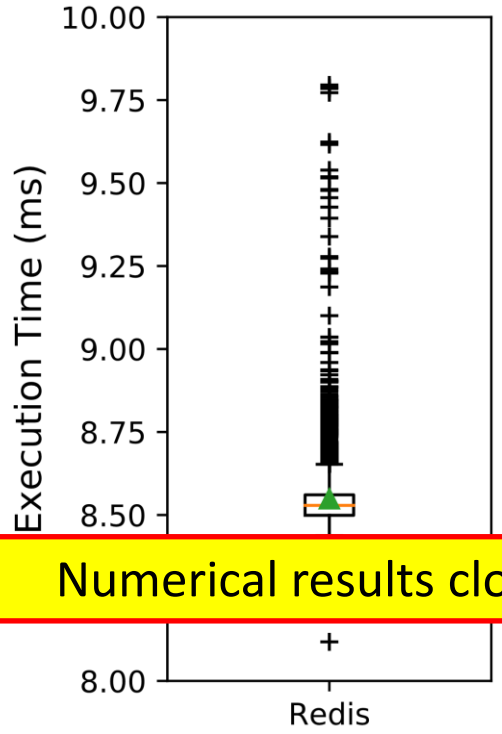
Experiment on Redis



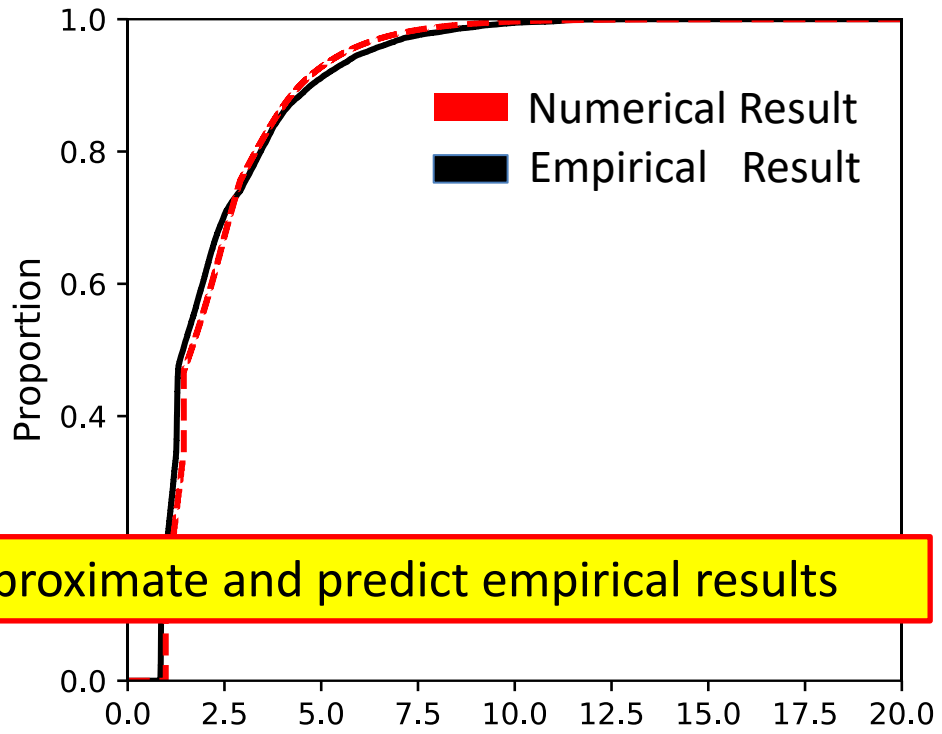
- Host: E5-2583v4 16-Core CPU @ 2.10GHz
- Xen 4.10.0 with Fixed Priority Scheduler Patch
- Linux 4.4.19 for all VMs
- Time Sensitive Service: **Redis**
- Compare Response Time CDFs of Numerical / Empirical Result

Redis Workload

- λ = 10 event/ms, d = 8.55 ms, P = 10 ms, B = 6 ms



Redis Query Execution Time



Numerical results closely approximate and predict empirical results

Conclusion

- A new queueing model, M/D(DS)/1, for aperiodic time-sensitive services running on a shared a virtualized platform.
- A numerical method for computing the stationary response time distribution of an M/D(DS)/1 queue.
- Experiments demonstrating the accuracy of the response time distribution on a real-time virtualization platform.

Meet tail-latency targets of time-critical services

- Future Work
  - Deterministic Service Time ➔ General Service Time
    - M/G(DS)/1

CPSL

Cyber-Physical Systems Laboratory

Thanks!

Washington University in St. Louis

From Virtual Waiting time to response time

- Goal: Response Time Distribution

$$Pr\{R=t\} = \sum_{l=0}^{\infty} \sum_{m=0}^{B} \sum_{n=0}^{P-1} (Pr\{R=t|V=l,U=m,T=n\} Pr\{V=l,U=m|T=n\} Pr\{T=n\})$$

Virtual Waiting Time (Joint Distribution)

PASTA: 1/P

- Either 0 or 1, can be determined via deferrable policy.

Solve The Markov Chain

Algorithm:

Vector $V_0 \leftarrow (1,0,0,0\ldots)$

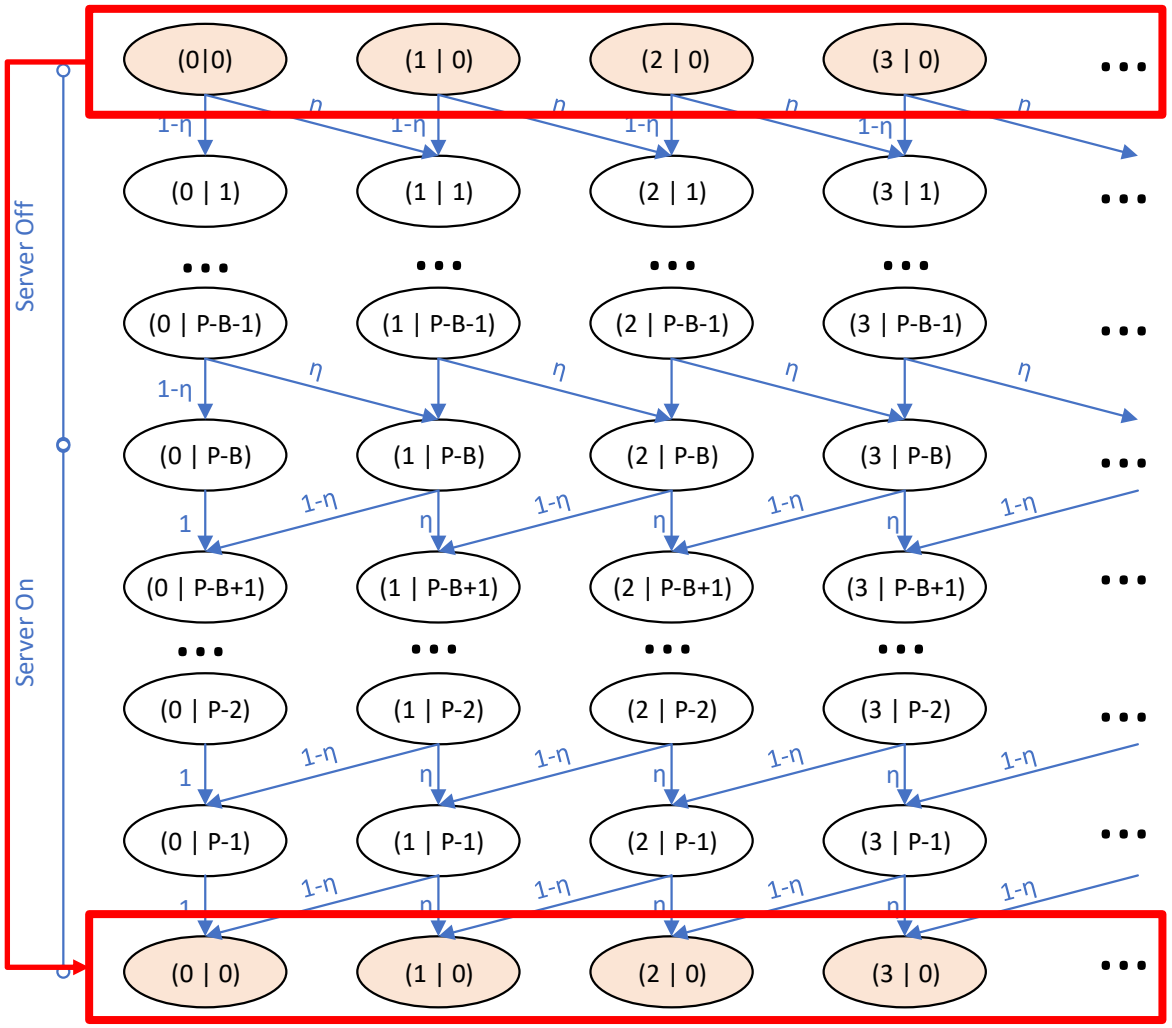
for k = 0 to 99:

Use V_k to get V_{k+1}

$V_{k+1} \leftarrow V_{k+1} / \text{sum}(V_{k+1})$

return V_{100}

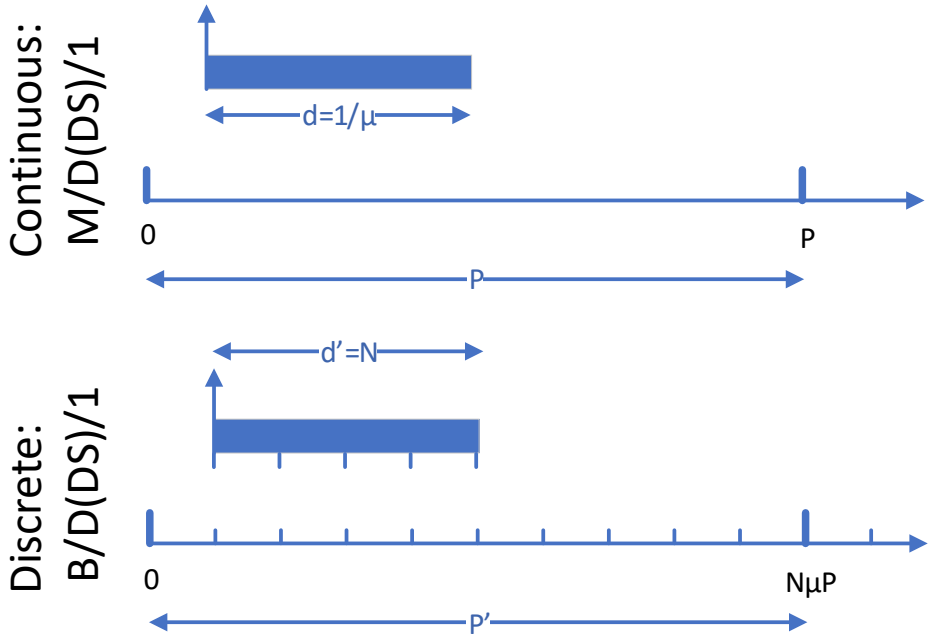
State Space: Offset to the start of the period:
Pr{V=x | T = n}



From Continuous to Discrete

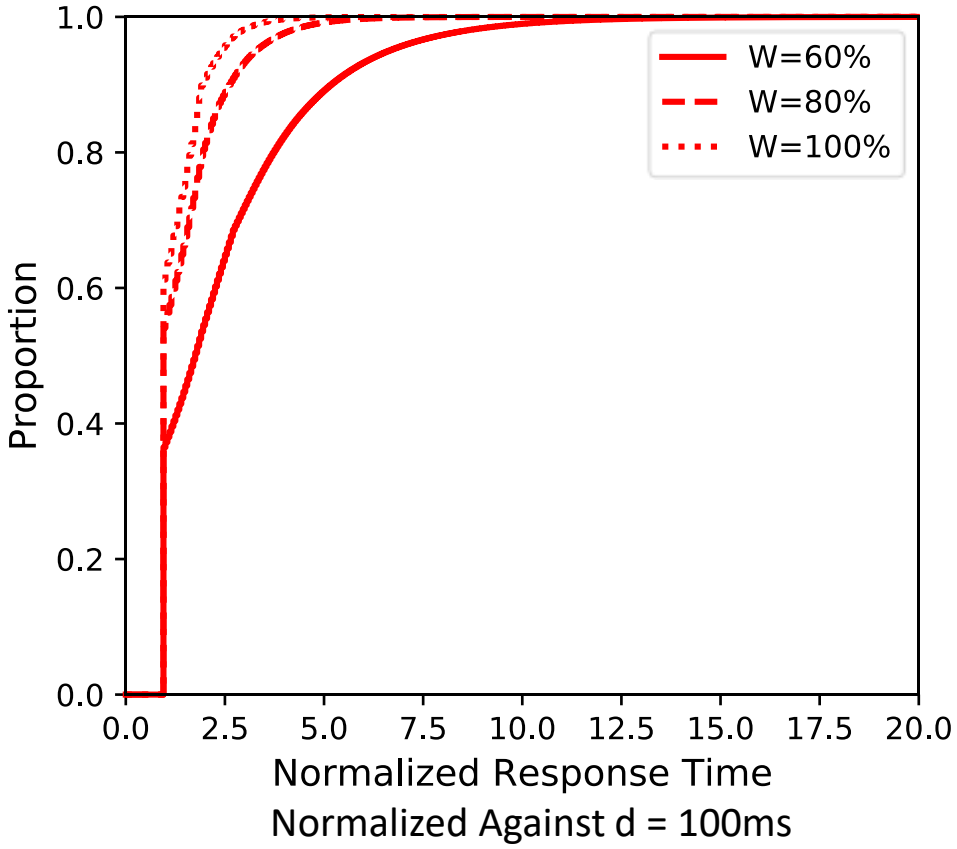
- Poisson Limit Theorem
  - Using Bernoulli Process to approximate Poisson Process
  - Parameter Mapping From M/D(DS)/1 to B/D(DS)/1

	M/D(DS)/1	B/D(DS)/1
Period	P	P'=NP/d
Budget	B	B'=NB/d
Duration	d	d'=N
Rate	λ	η=λd/N



Synthetic Workload

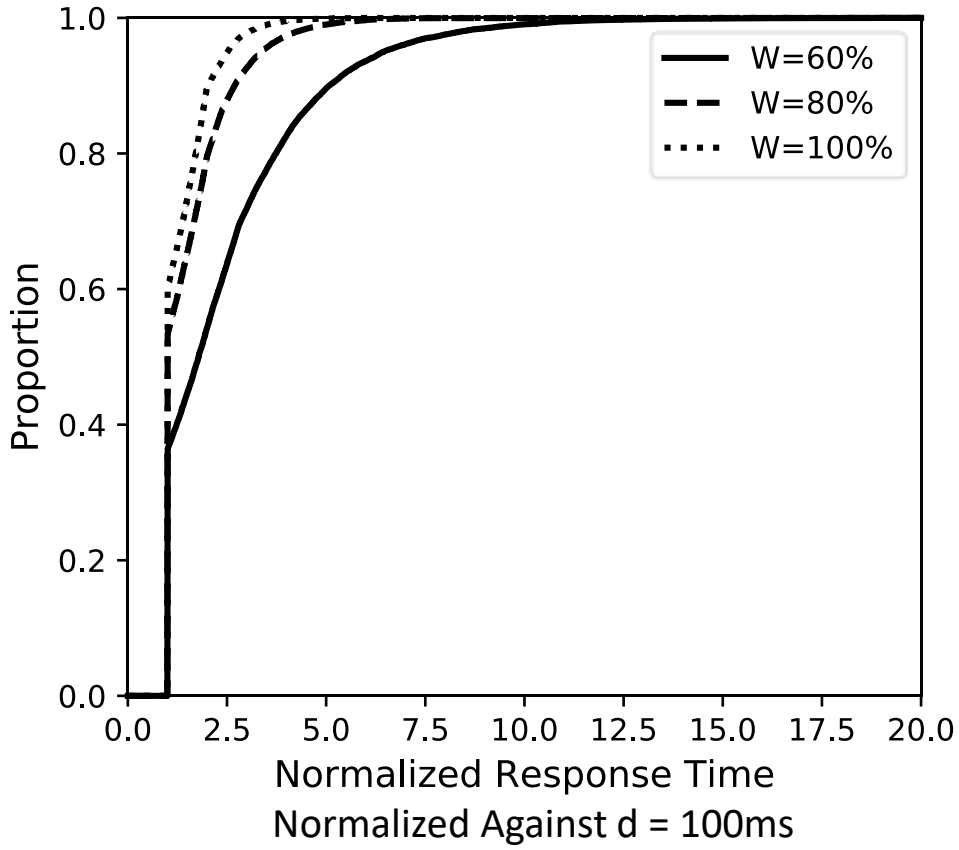
- λ = 0.004event/ms, d = 100ms, P = 200ms
- B : {120, 160, 200}ms ➔ $W = \frac{B}{P}$: {60%, 80%, 100%}



Numerical Result

Synthetic Workload

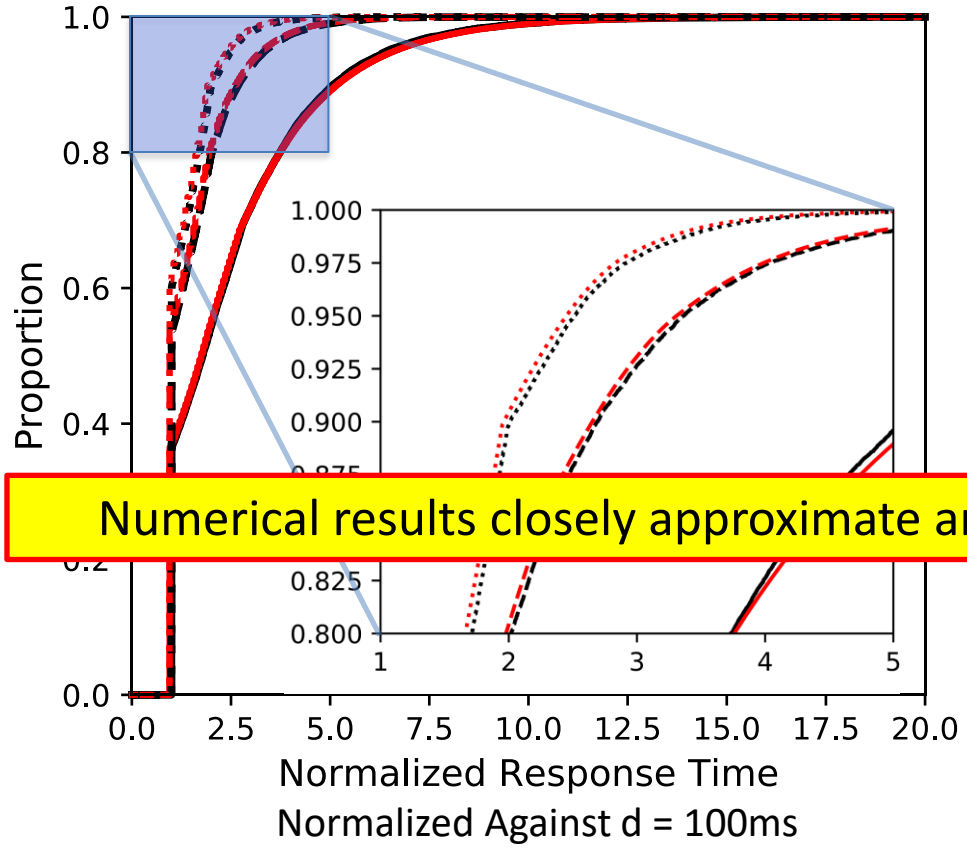
- λ = 0.004event/ms, d = 100ms, P = 200ms
- B : {120, 160, 200}ms ➔ $W = \frac{B}{P}$: {60%, 80%, 100%}



Empirical Result

Synthetic Workload

- λ = 0.004event/ms, d = 100ms, P = 200ms
- B : {120, 160, 200}ms ➔ $W = \frac{B}{P}$: {60%, 80%, 100%}



Numerical results closely approximate and predict empirical results