

# Zero-Delay Load Balancing Algorithms in Large-Scale Data Centers

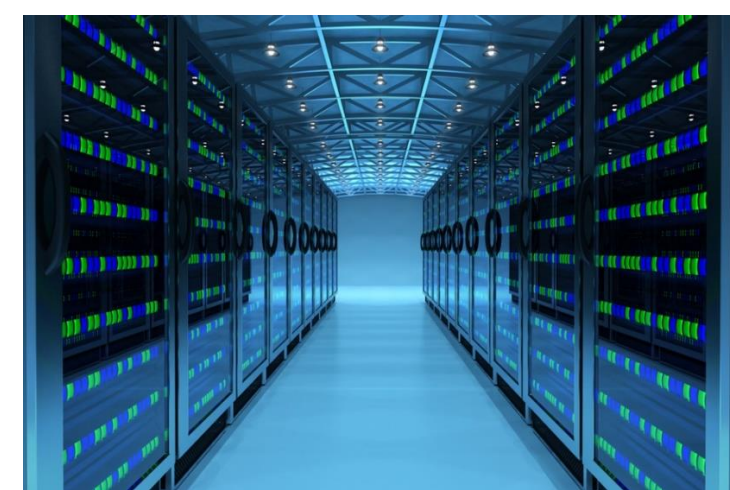
Xin Liu and Lei Ying  
Arizona State University

Award title "Demand Response & Workload Management for Data Centers with Increased Renewable Penetration"

## Motivation

Large server systems are thriving:

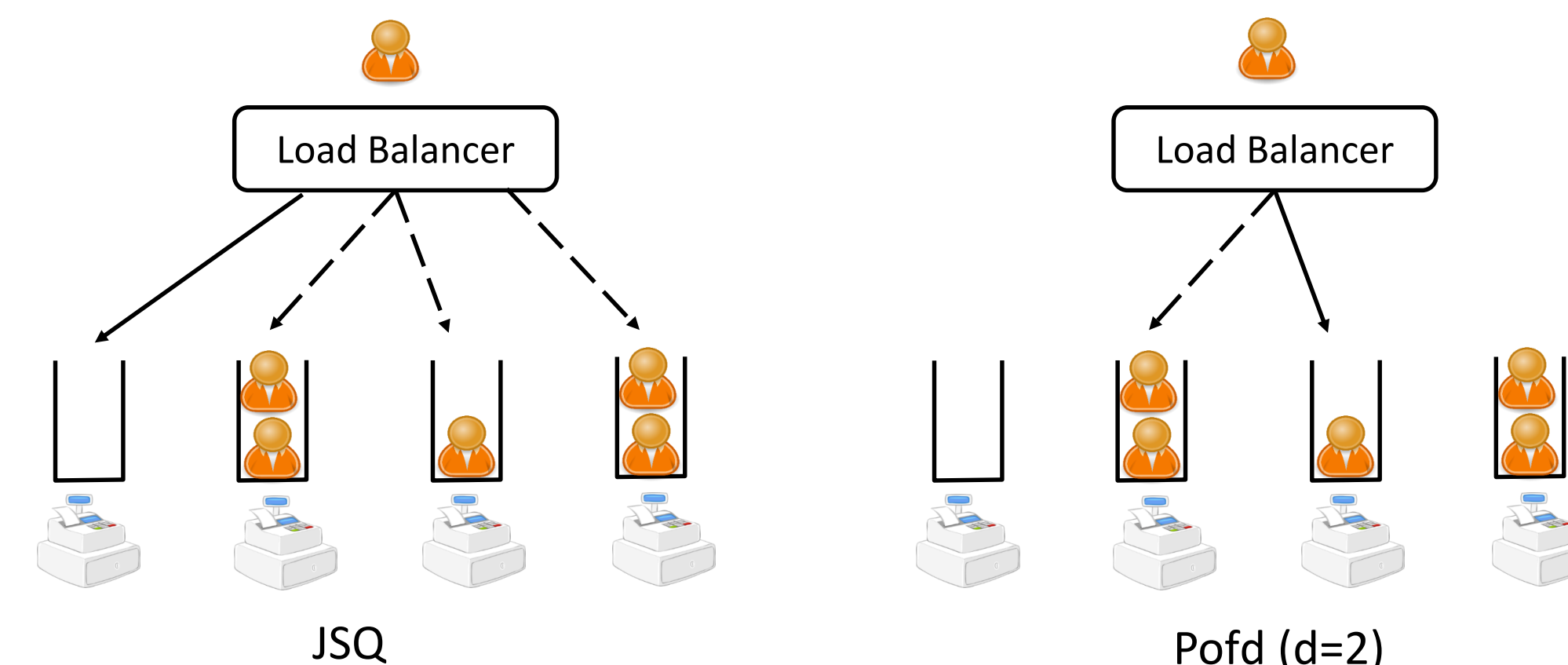
- Jobs (e.g. search requests, data mining requests ) are processed by many servers (e.g. large-scale data centers).
- Short response time and asymptotic zero latency are important for data centers.



Load balancing in server systems:

- Role - schedule incoming requests to servers
- Goal - low delay (**zero delay**)

## Load balancing algorithms



JSQ	Process-Level & Steady-State	Pofd with $d = \frac{\log N}{1-\lambda}$	Process-Level & Steady-State?
Servers with one job	$N - \Theta(\sqrt{N})$	Servers with one job	$N - \Theta(\sqrt{N})$
Servers with two or more jobs	$\Theta(\sqrt{N})$	Servers with two or more jobs	$\Theta(\sqrt{N})$

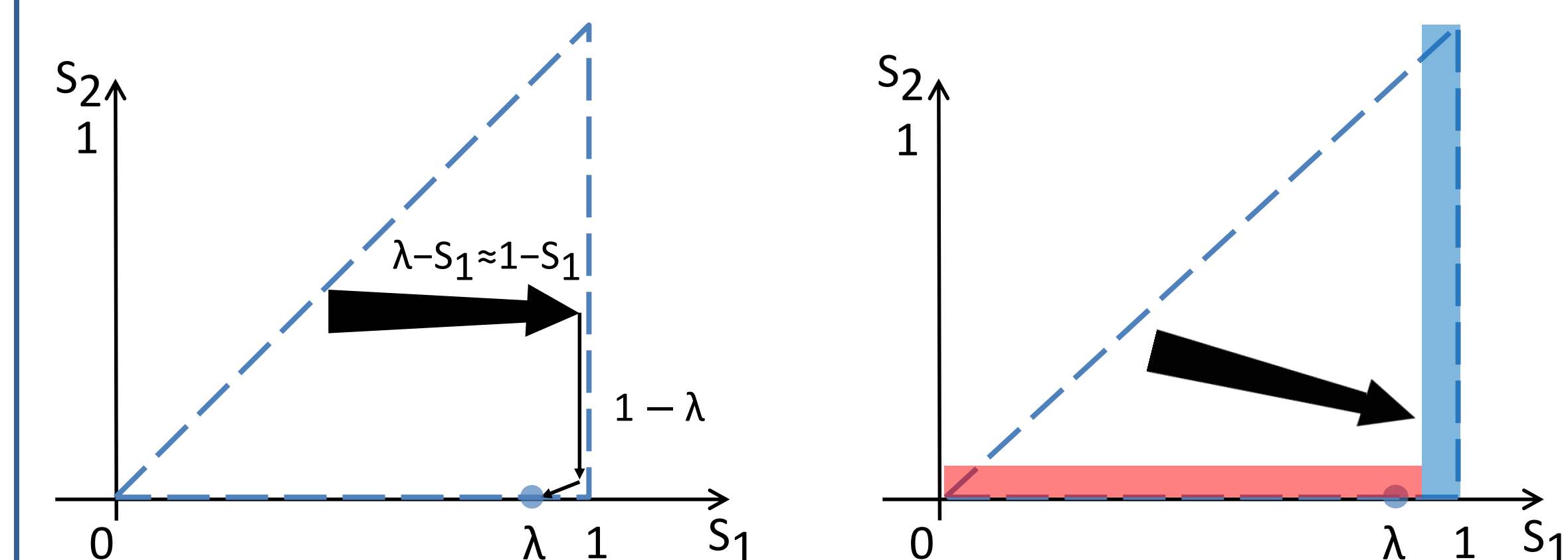
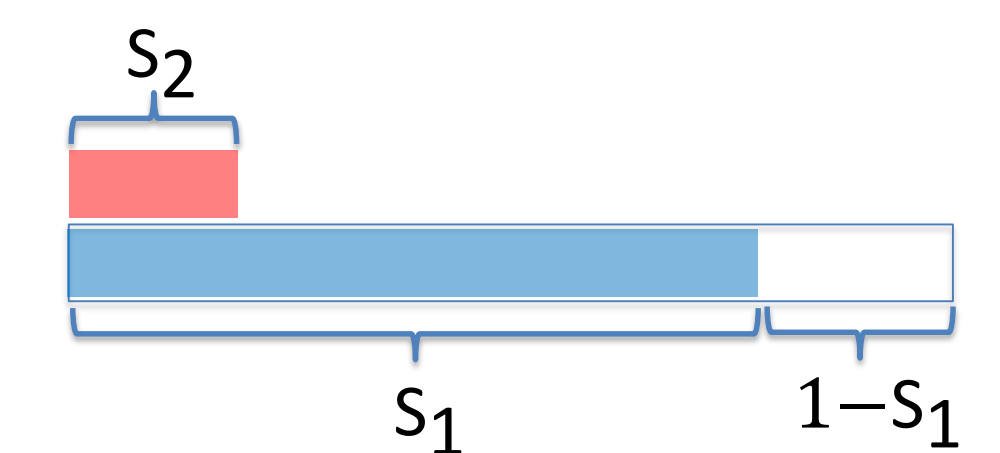
Related results on JSQ and Pofd

## Analysis framework: SSC + Stein's method

State Space collapse ( $b = 2$ )

State representation:

- $S_i$  is the fraction of servers with at least  $i$  jobs



Dynamic behavior and state space collapse

$(S_1, S_2)$  collapses to blue and red region:

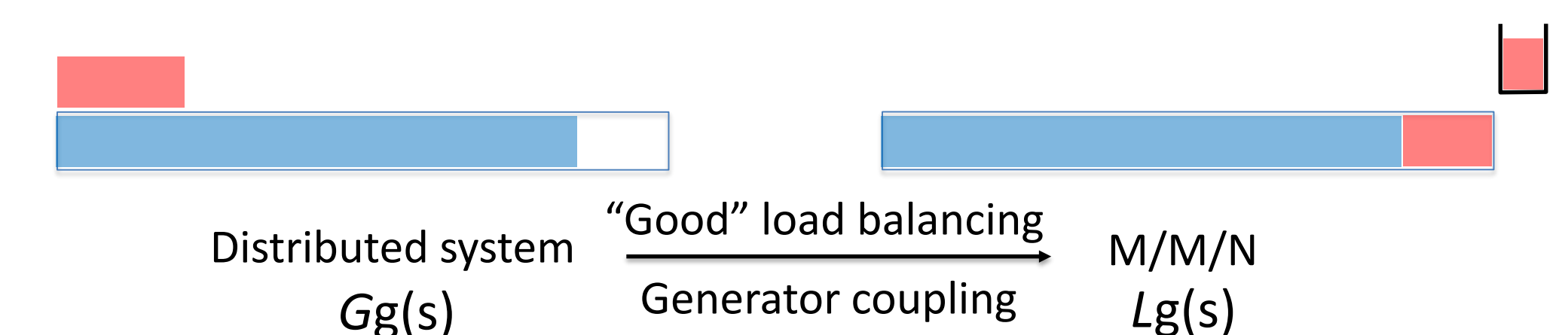
$$S_1 \geq \lambda + \frac{2 \log N}{\sqrt{N}} \text{ or } S_2 \leq \frac{2 \log N}{\sqrt{N}}$$

## Stein's method

A truncated distance function:

$$h(S) = \max \left\{ S_1 + S_2 - \lambda - \frac{2 \log N}{\sqrt{N}}, 0 \right\}$$

Coupling with M/M/N system:

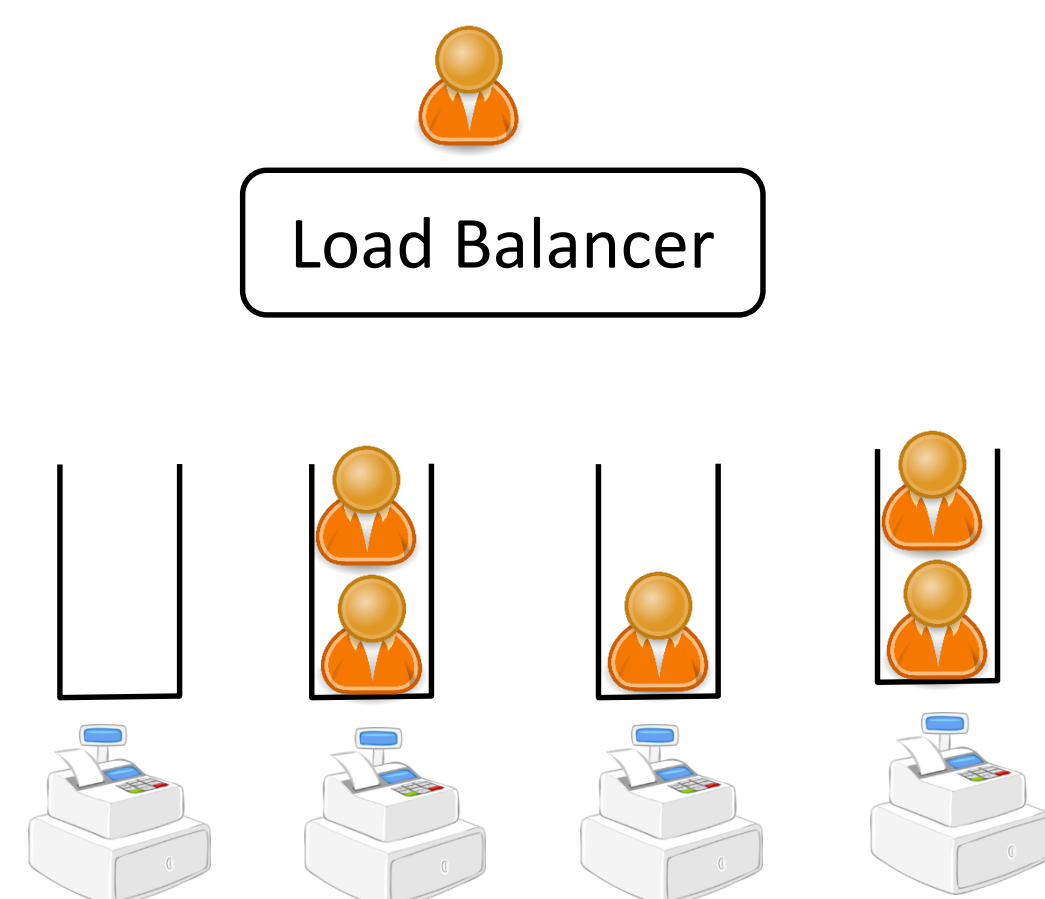


Steady-state approximation:

$$E[h(S)] = E[Lg(S) - Gg(S) | S \in \Omega] \Pr(S \in \Omega) + E[Lg(S) - Gg(S) | S \notin \Omega] \Pr(S \notin \Omega) \text{ SSC}$$

## Model

Distributed queue system:



Load balancing in server system

Key assumptions:

- ✓  $N$  homogenous servers
- ✓ FIFO queues
- ✓ Exponential service time ( $\mu = 1$ )
- ✓ Poisson arrival:  $\lambda N$
- ✓ Sub Halfin-Whitt regime:  
 $\lambda = 1 - N^{-\alpha}, \alpha < 0.5$
- ✓ Finite buffer  $b = o(\log N)$

## Research problems:

- what are good load balancing to achieve **zero delay** at steady-state in sub Halfin-Whitt regime?

## Main contribution:

- a sufficient condition to achieve **zero delay**
- simple analysis framework

## Main Results

JSQ, JIQ, I1F and Pod with  $d = \frac{\log N}{1-\lambda}$  achieve **zero delay** as  $N \rightarrow \infty$ .

	Steady-State		Steady-State
Servers with one job	$N - N^{1-\alpha}$	Waiting time	$\leq \frac{3 \log N}{\sqrt{N}}$
Servers with two or more jobs	$k\sqrt{N} \log N$	Waiting prob	$\leq \frac{4 \log N}{\sqrt{N}}$
Load balancing	JSQ, I1F Pod with $d = \frac{\log N}{1-\lambda}$	Load balancing	JSQ, I1F Pod with $d = \frac{\log N}{1-\lambda}$
			JIQ

Expected queue length, waiting time and prob at steady-state

Sufficient condition to achieve **zero delay**:

- A job routed to an idle server with a high probability ( $1 - N^{-0.5}$ ) given a fraction  $N^{-\alpha}$  of idle servers.

	Prob. to an idle server
JSQ, JIQ, I1F	1
Pod with $d = \frac{\log N}{1-\lambda}$	$\geq 1 - N^{-0.5}$

JSQ etc. satisfy the sufficient condition