RAS: Continuously Optimized Region-Wide Datacenter Resource Allocation

主讲人: 伊丹翔

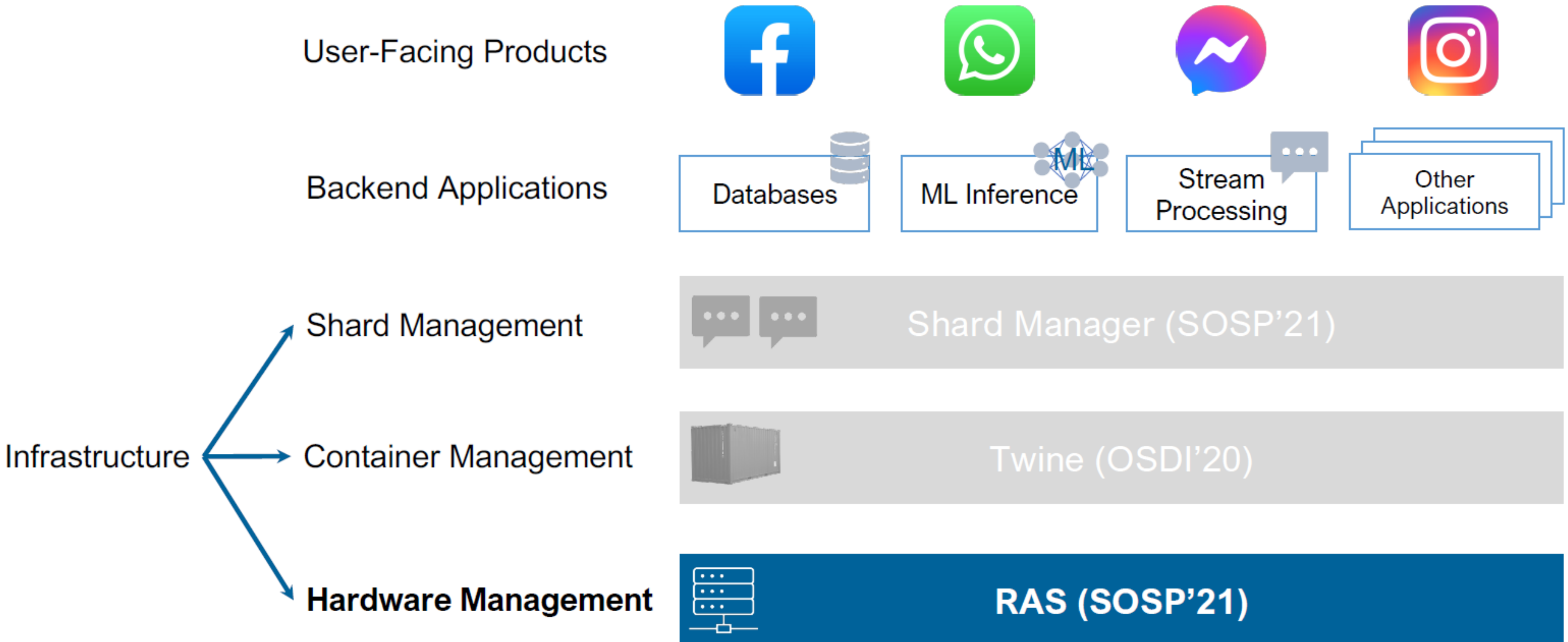
2021 年 11 月 23 日

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

目录 Contents

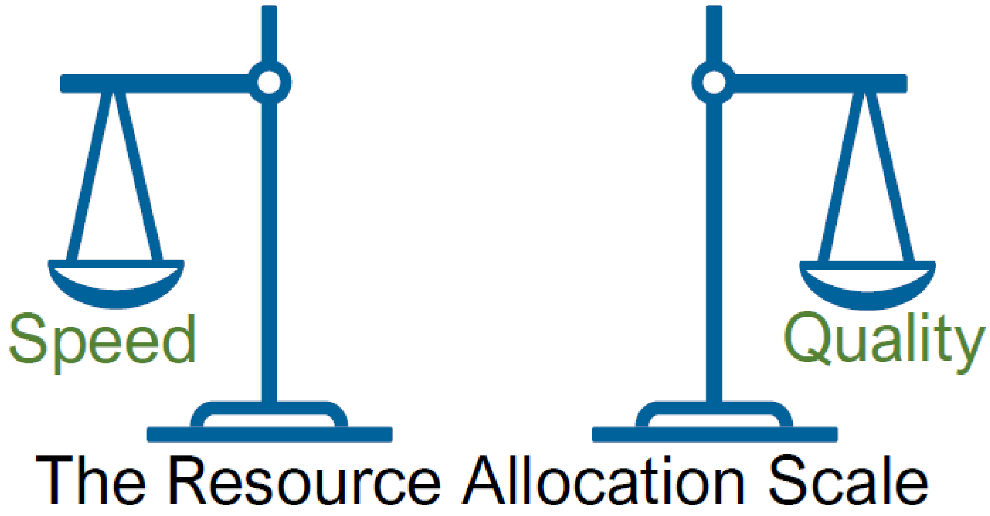
上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

RAS 是什么

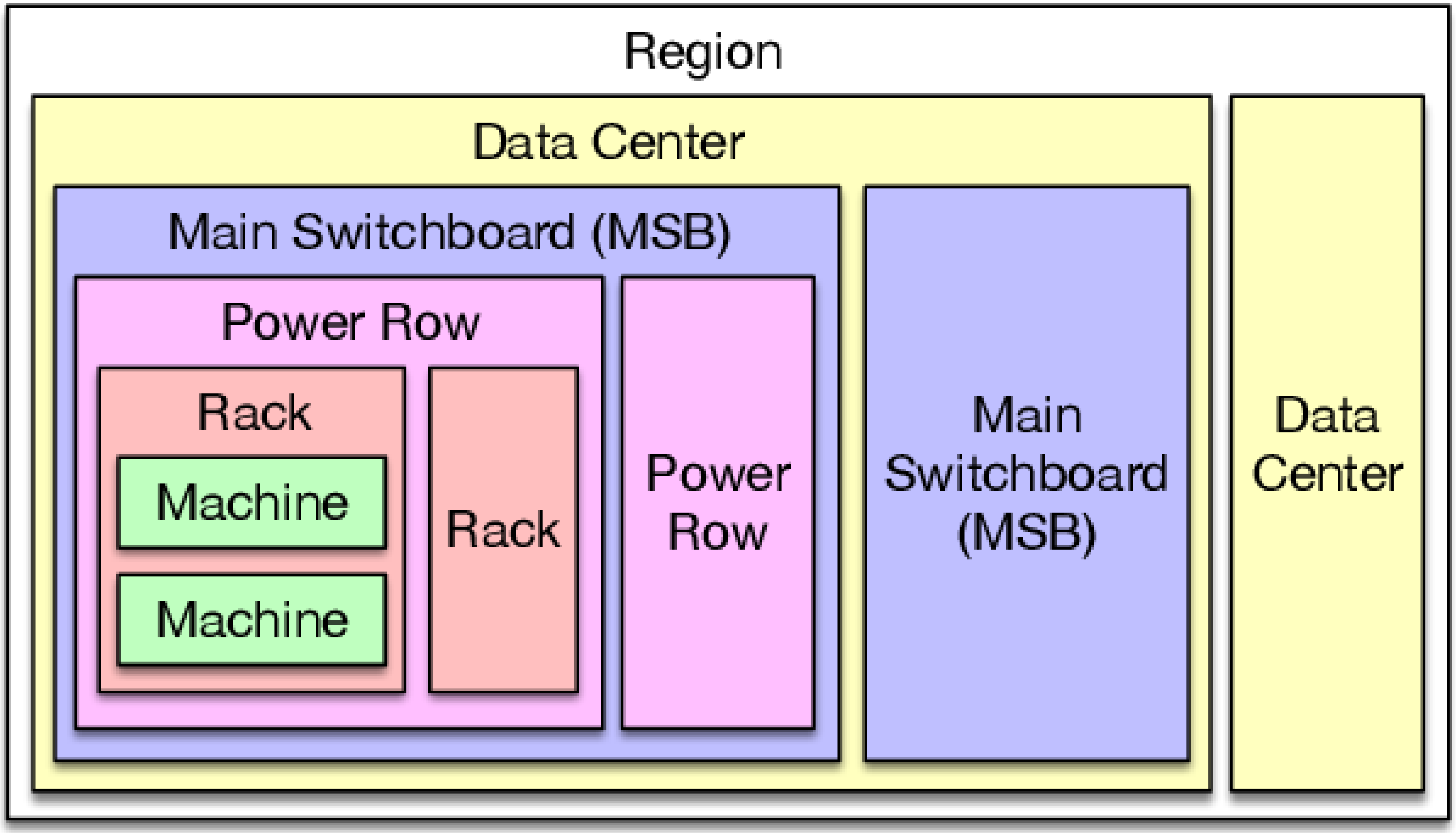

User-Facing Products
Backend Applications
Databases
ML Inference
Stream Processing
Other Applications
Infrastructure
Shard Management
Container Management
Hardware Management
Shard Manager (SOSP’21)
Twine (OSDI’20)
RAS (SOSP’21)

RAS —— 资源分配系统

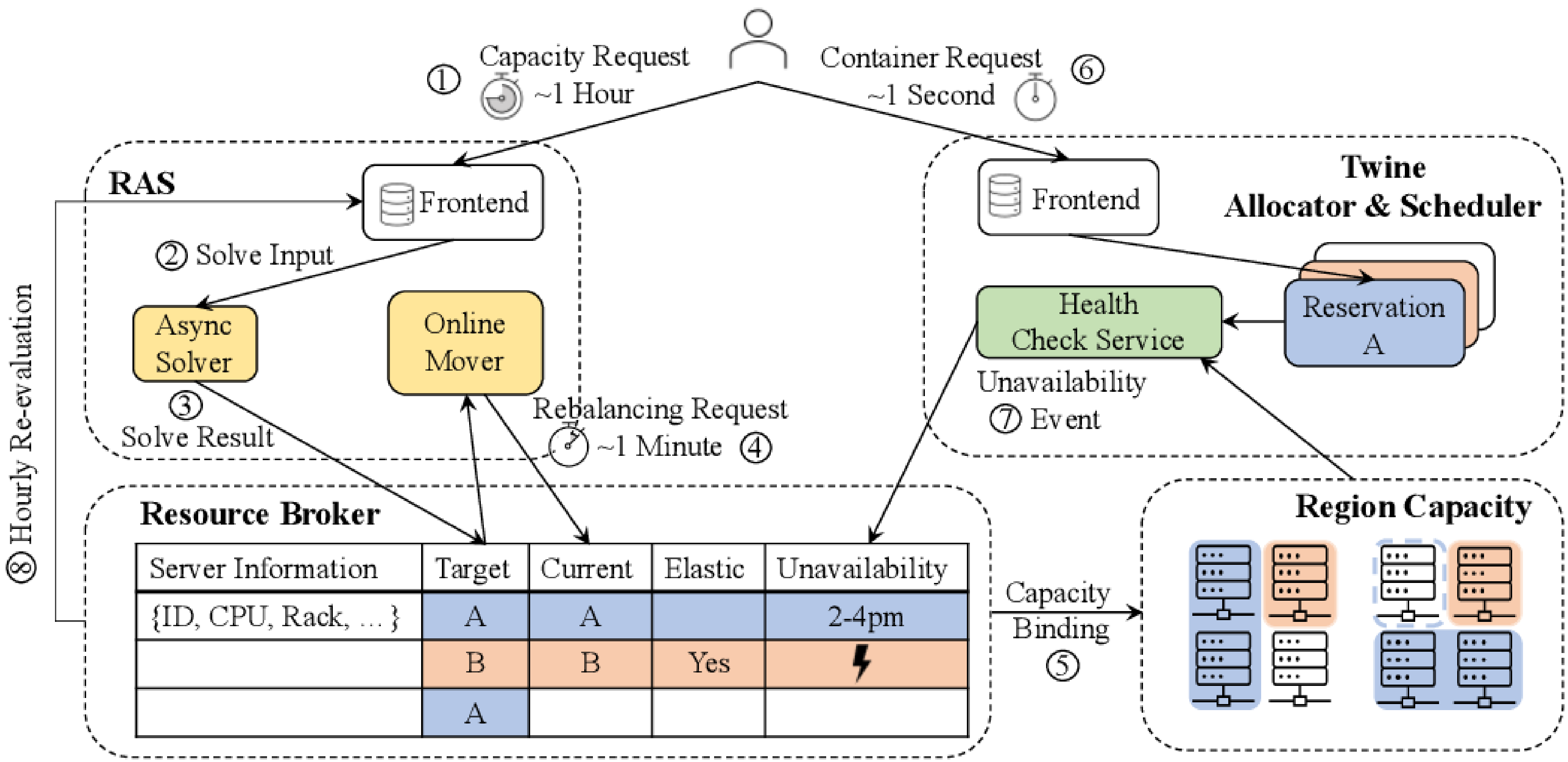
基于 Twine 的新的服务器分配组件



Facebook 的 datacenter 拓扑图



RAS 工作原理


① Capacity Request
~1 Hour
Container Request ⑥
~1 Second
RAS
Frontend
② Solve Input
Async Solver
Online Mover
③ Solve Result
Rebalancing Request
~1 Minute ④
Twine
Allocator & Scheduler
Frontend
Health Check Service
Reservation A
Unavailability
⑦ Event
⑧ Hourly Re-evaluation
Resource Broker
Server Information
Target
Current
Elastic
Unavailability
{ID, CPU, Rack, … }
A
A
2-4pm
B
B
Yes
A
Capacity Binding
⑤
Region Capacity

Failure Buffers

Random Failure：

Failure Buffers

Random Failure:

Correlated Failure:

Failure Buffers

Random Failure: Shared buffer

Correlated Failure:

Failure Buffers

Random Failure:　　Shared buffer —> Online Mover

Correlated Failure:

Failure Buffers

Random Failure:　　Shared buffer —> Online Mover

Correlated Failure: Embedded buffer

Failure Buffers

Random Failure： Shared buffer —> Online Mover

Correlated Failure： Embedded buffer —> Twine Allocator

Async Solver

Two-phase solving:

1. Solve without any rack-related goals
2. Solve with all goals in phase 1 plus rack goals

Async Solver

Constraints:

1. Capacity
2. Server availability
3. Network
4. Correlated failure

Objectives:

1. move unused servers
2. spreads reservations across MSBs
3. reduce hotspots that may overload rack switch uplinks

Async Solver

Minimize:

$$\sum_{s \in S, r \in R} M_s * \max(0, X_{s,r} - x_{s,r}) \qquad (1)$$

$$+\beta * \sum_{r \in R, G \in \Psi^K} \max\left(0, \sum_{s \in G} (V_{s,r} * x_{s,r}) - \alpha^K * C_r\right) \qquad (2)$$

$$+\beta * \sum_{r \in R, G \in \Psi^F} \max\left(0, \sum_{s \in G} (V_{s,r} * x_{s,r}) - \alpha^F * C_r\right) \qquad (3)$$

$$+\tau * \sum_{r \in R} \max_{G \in \Psi^F} \left(\sum_{s \in G} V_{s,r} * x_{s,r}\right) \qquad (4)$$

Subject to:

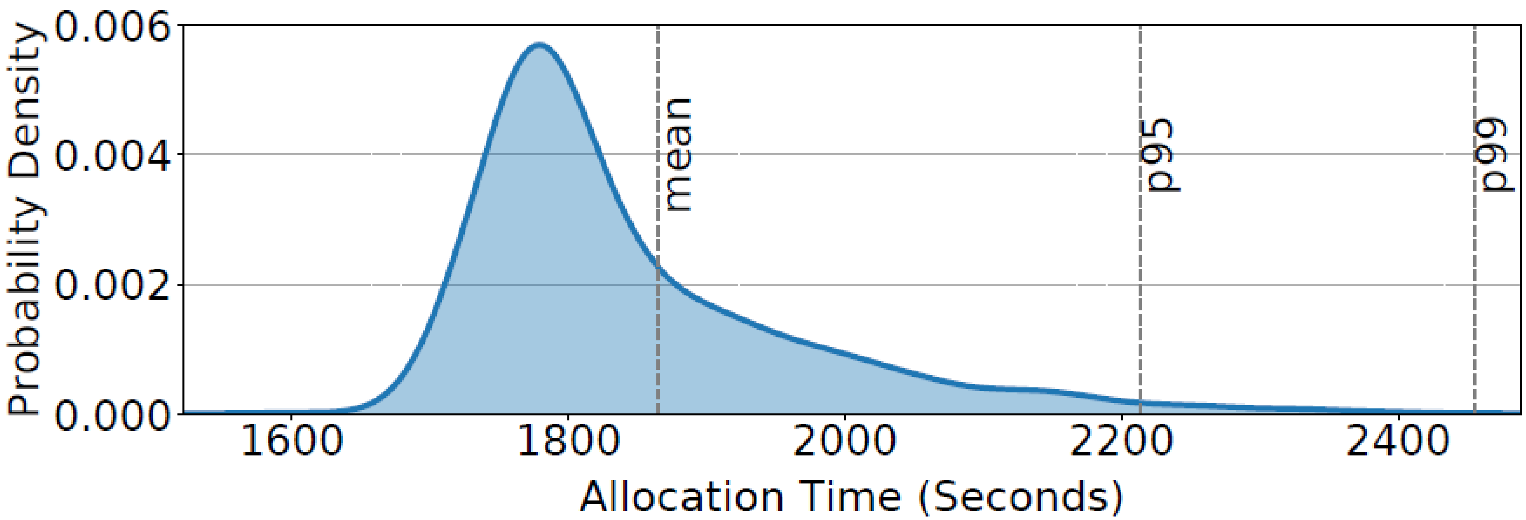
$$\sum_{r \in R} x_{s,r} \le 1, \qquad \forall s \in S \qquad (5)$$

$$\sum_{s \in S} (V_{s,r} * x_{s,r}) - \max_{G \in \Psi^F} \left(\sum_{s \in G} V_{s,r} * x_{s,r}\right) \ge C_r, \qquad \forall r \in R \qquad (6)$$

$$\left| \frac{\sum_{s \in G} (V_{s,r} * x_{s,r})}{C_r} - A_{r,G} \right| \le \theta, \qquad \forall r \in R, G \in \Psi^D \qquad (7)$$

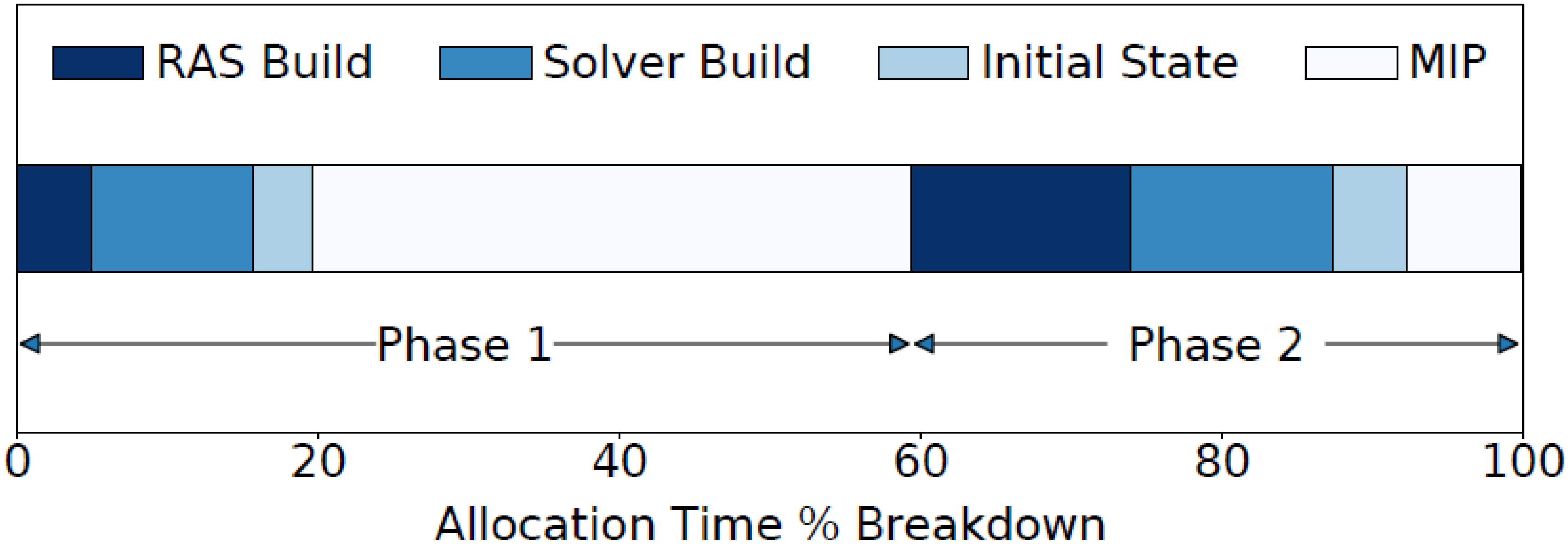
Notation	Description
S	Set of all servers
R	Set of all reservations
$x_{s,r}$	Assignment variable which is 1 if server s is assigned to reservation r and 0 otherwise
$X_{s,r}$	Constant initial assignment value
M_s	Movement cost of server s
τ	Cost of each correlated-failure-buffer server
β	Cost of each server outside spread goals
$\alpha^{K,F}$	Proportional limit of reservation for spread in K (rack) or F (MSB fault domain)
$V_{s,r}$	RRU value of server s for reservation r
C_r	Capacity desired for reservation r
$\Psi^{K,F,D}$	Partition of servers based on K (rack), D (datacenter), or F (MSB fault domain)
$A_{r,G}$	Affinity of reservation r to a partition group G

RAS Performance



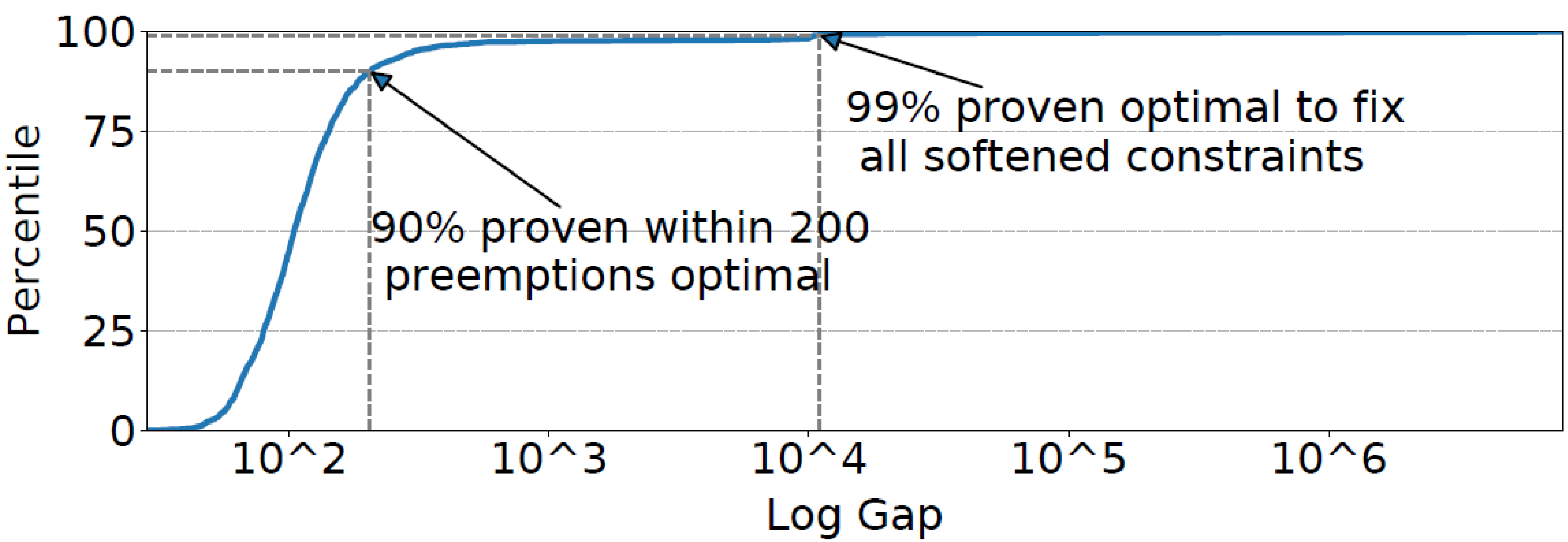
RAS regional allocation time distribution

RAS Performance



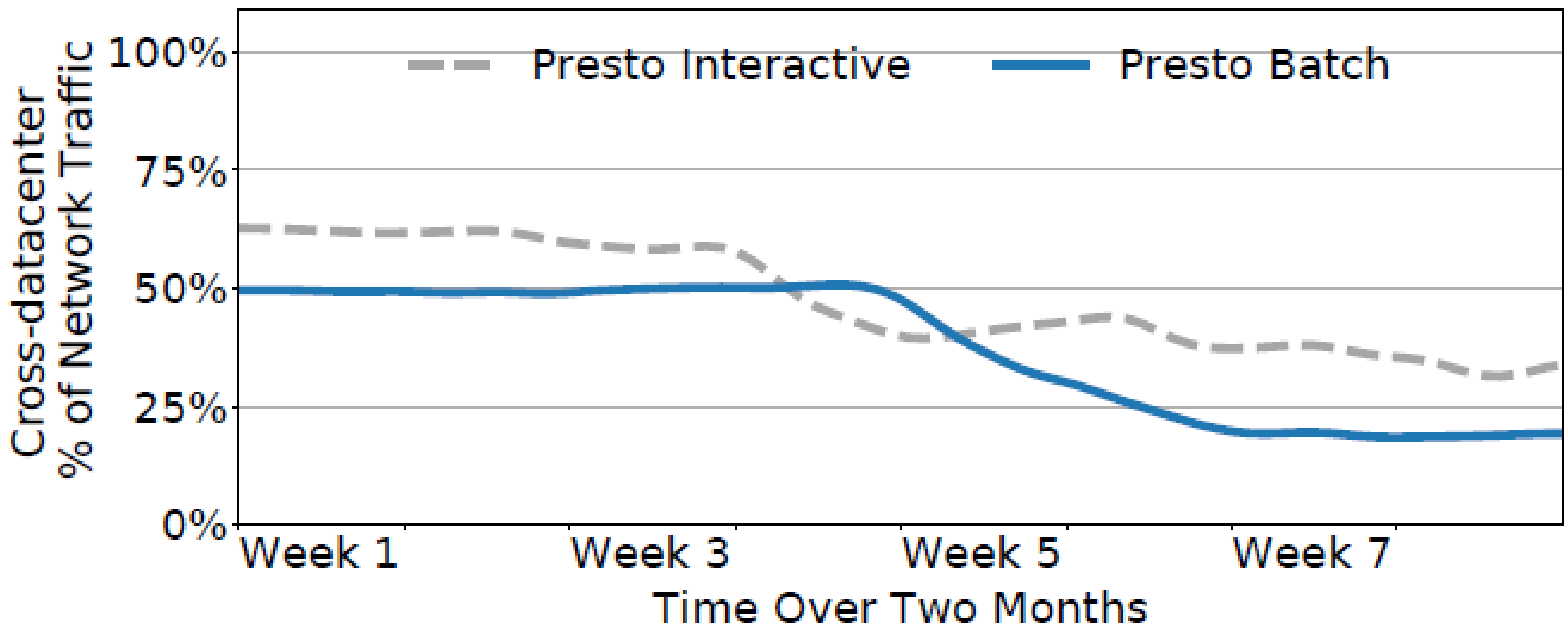
RAS allocation time breakdown

RAS Performance



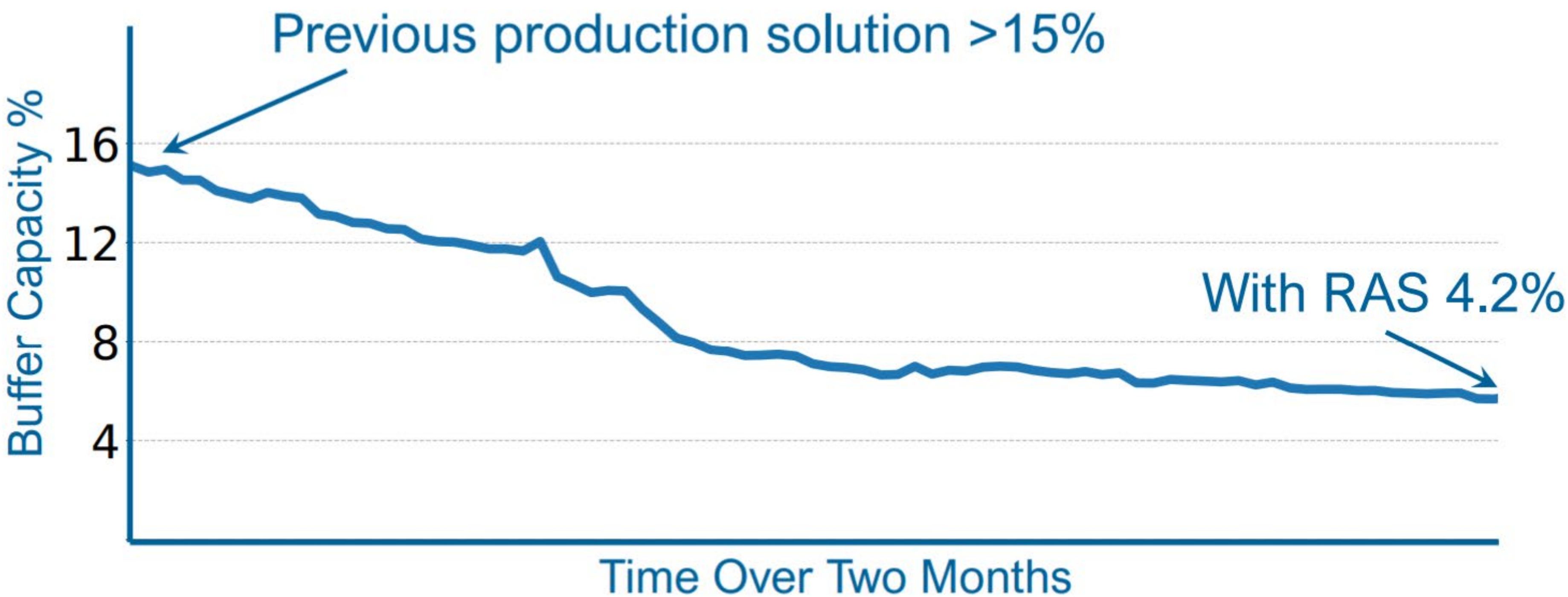
Allocation Quality: Phase 1 MIP quality gap

RAS Evaluation



RAS helps reduce cross-datacenter network traffic over a period of two months.

RAS Evaluation



RAS helps reduce correlated-failure buffers over a period of two months

Discussion

相信RAS的一些关键想法可以被其他系统考虑:

1. 给user介绍动态reservation而不是静态集群
2. 把服务器分配和容器放置解耦
3. 把服务器分配到reservation看成一个优化问题
4. …

Challenges

1. Capacity-request delays
2. Extra service preemption
3. …

谢谢！