Sparrow: Distributed, Low Latency Scheduling

Shivaram Venkataraman

CS 294: Big Data Systems Research

Sparrow: Cluster Scheduling for Interactive Workloads

Sparrow

Low Overhead Scheduling for Interactive Jobs

Sparrow

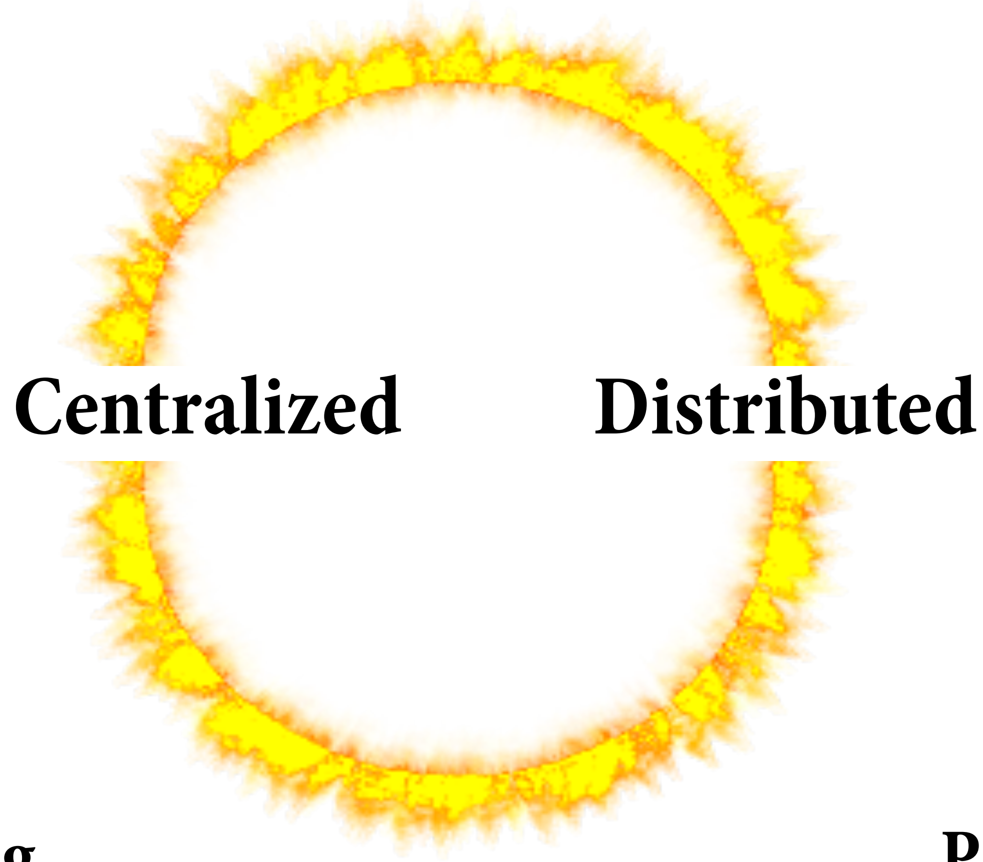
Distributed Low-Latency Scheduling

Kay Ousterhout

Ion Stoica

Kay Ousterhout, Patrick Wendell, Matei Zaharia, Io

Some History



Centralized

Distributed

Time sharing

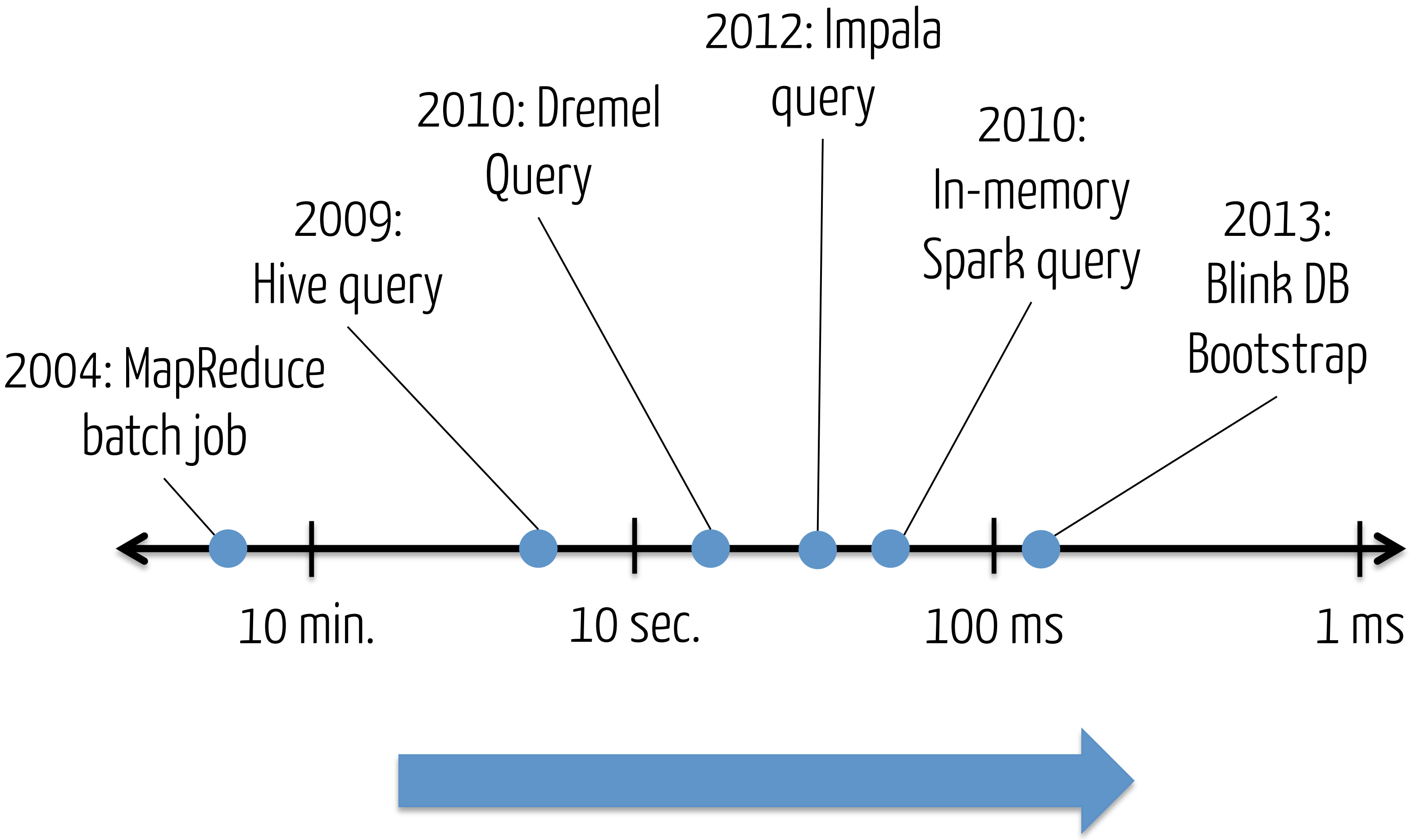
PCs, Internet

Data center Systems

P2P Systems

???

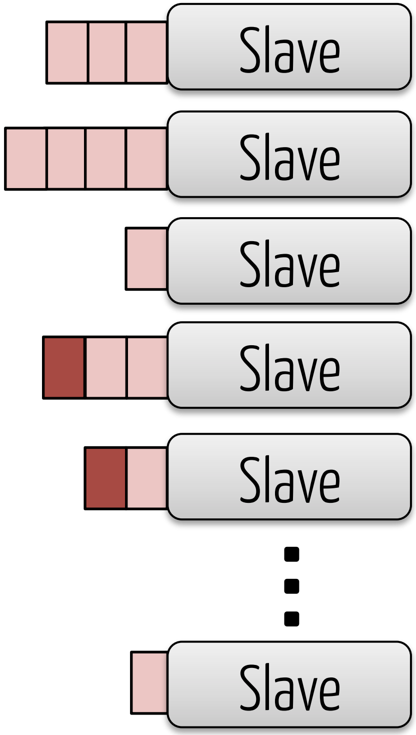
Problem Statement



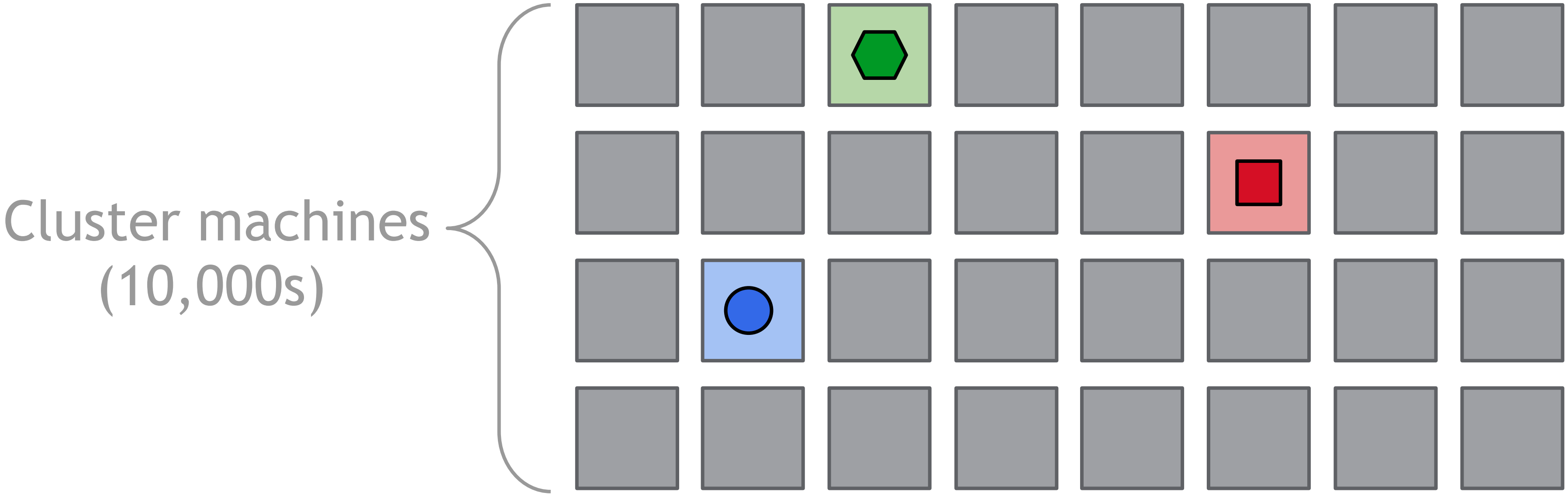
Real Problem Statement

~~Scheduling~~

State Management

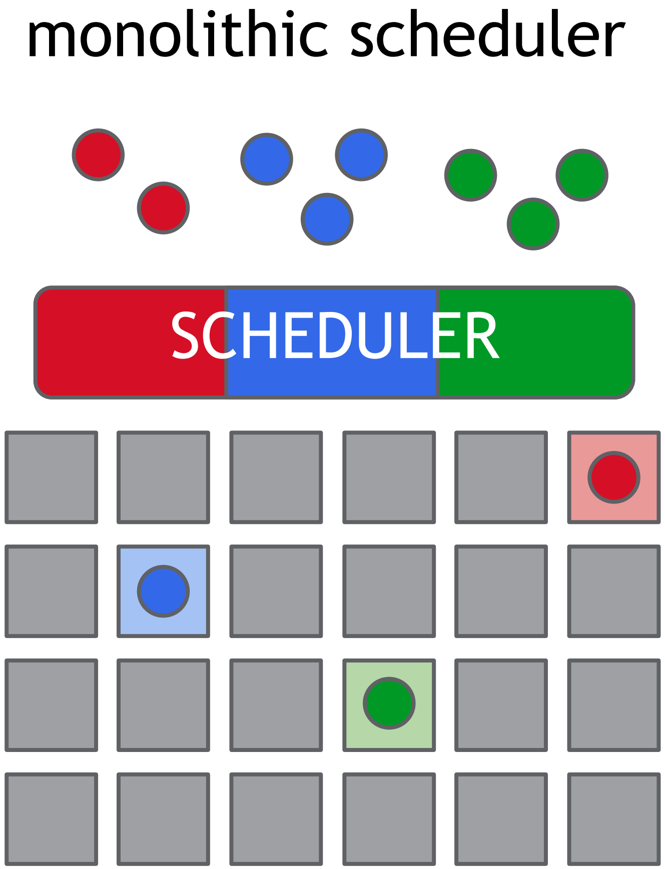


State Management



Scheduling: Add tasks to machine table

Ways to Schedule (1)

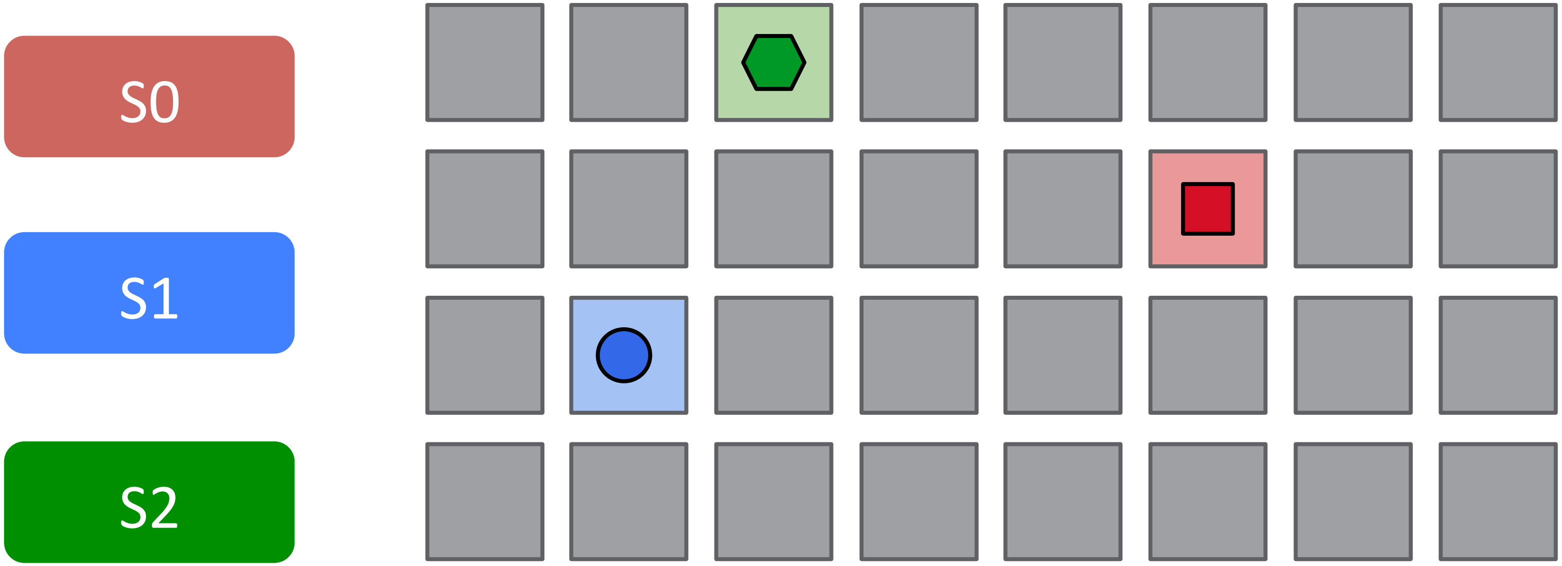


Monolithic Scheduler
(Serial updates to DB)

Easy to reason about
Support policies

Scalability
Fault tolerance

Ways to Schedule (2)



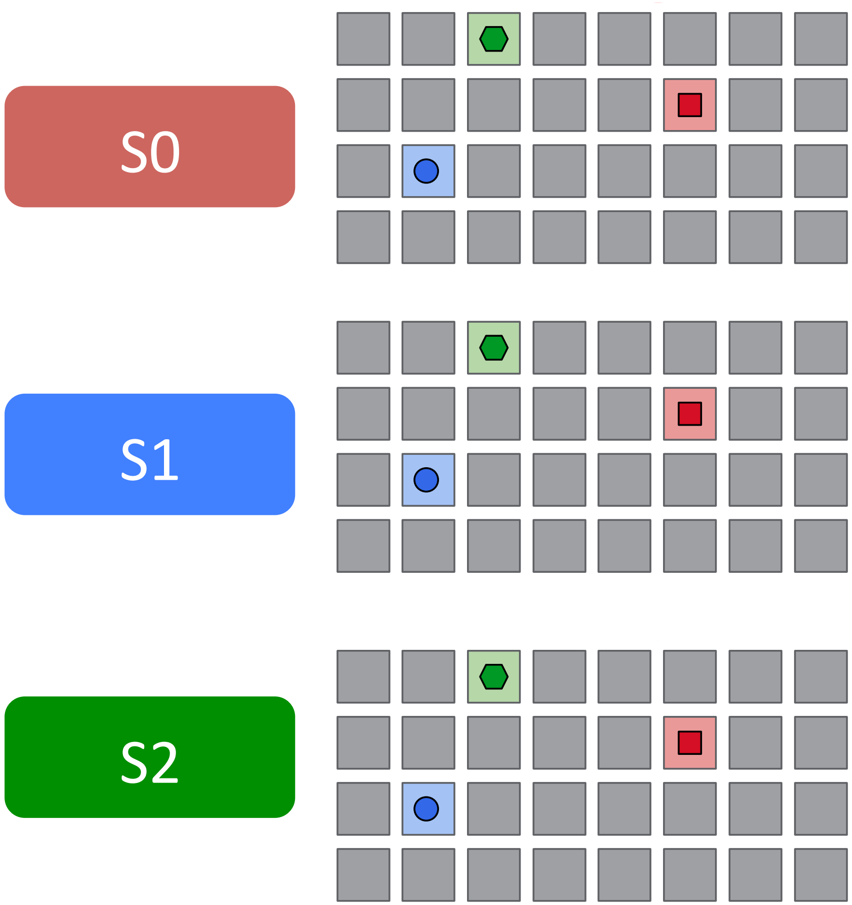
S0

S1

S2

Where is the state stored ?

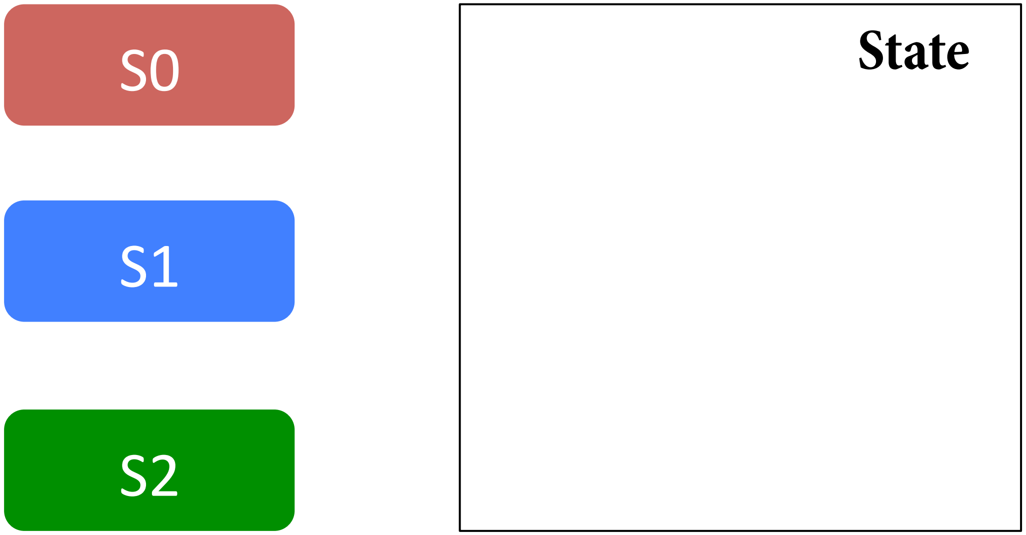
Ways to Schedule (2)



Replicate and Synchronize

Omega

Ways to Schedule (3)



S0

S1

S2

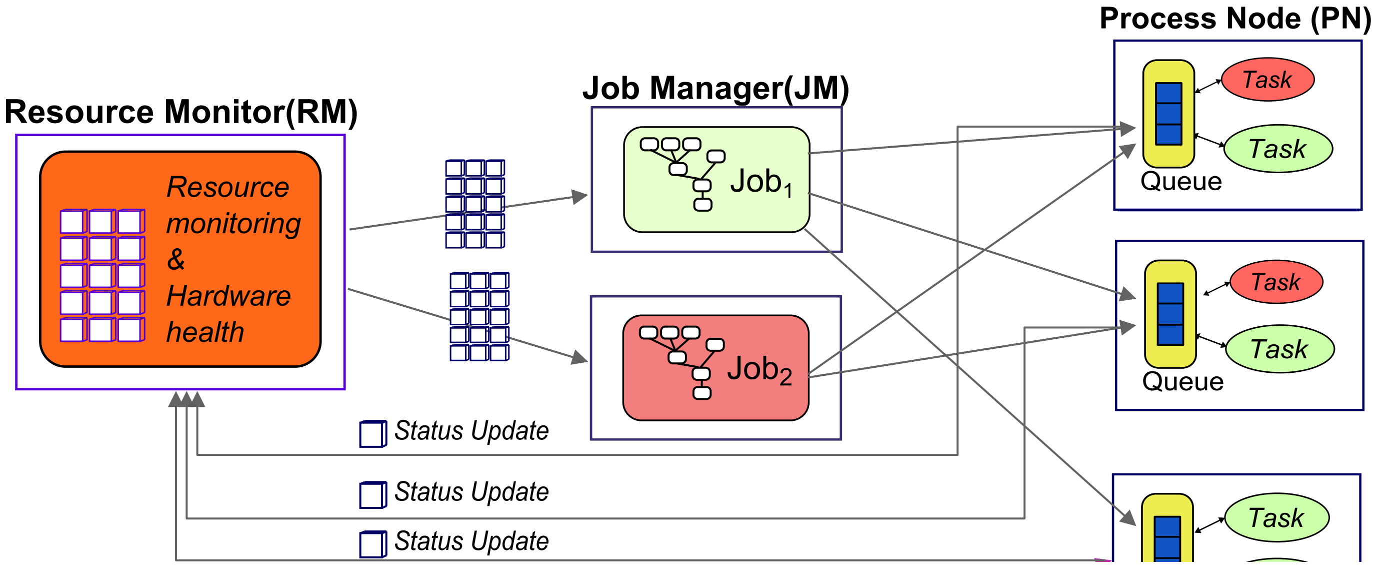
State

Sparrow:
Don't store
but *compute* it

Accurately is hard → Approximate it

Ways to Schedule (4)

Apollo (OSDI 2014): Collect state centrally
Information might be stale. Resolve conflicts



no-state schemes

Pros

- **Easy scalability, Fault tolerance**

  Similar to web *frontends*

Challenges

- **Accuracy of computed state**

  Batch sampling for *least loaded* worker

  What about other metrics ?

(1) Fundamental Trade-offs

Latency

- ***no-state:*** **Assured low latency (O(RTT))**
- ***shared-state:*** **Transaction Conflicts ?**

Question: Scalability as you add more schedulers ?

(1) Fundamental Trade-offs

Scheduling capabilities

- ***no-state:*** **Simple constraints, job, task-level**
- ***shared-state:***
  - **Across jobs: Bin packing, Complex policies**
  - **Within jobs: Dependencies across Stages**

(2) Insights from P2P systems

Routing:

- **Number of lookups (latency)**
- **Entries stored per node (state)**

Churn:

- **“...to reduce churn: add some randomness”**
- **Minimizing Churn in Distributed Systems**
- **[SIGCOMM 2006]**

(2) What is different now ?

Latency: Wide area vs. Datacenter

Trusted domains

- **No need for authentication, incentive schemes**
- **Need for fault tolerance vs. churn**

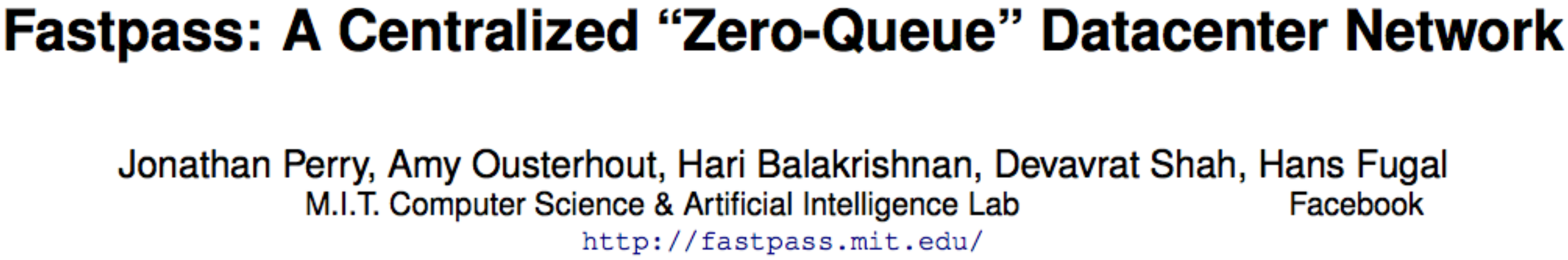
(3) One fast machine

What are the fundamental bottlenecks ?

Network Bandwidth or Latency ?

Multiple threads vs. Schedulers

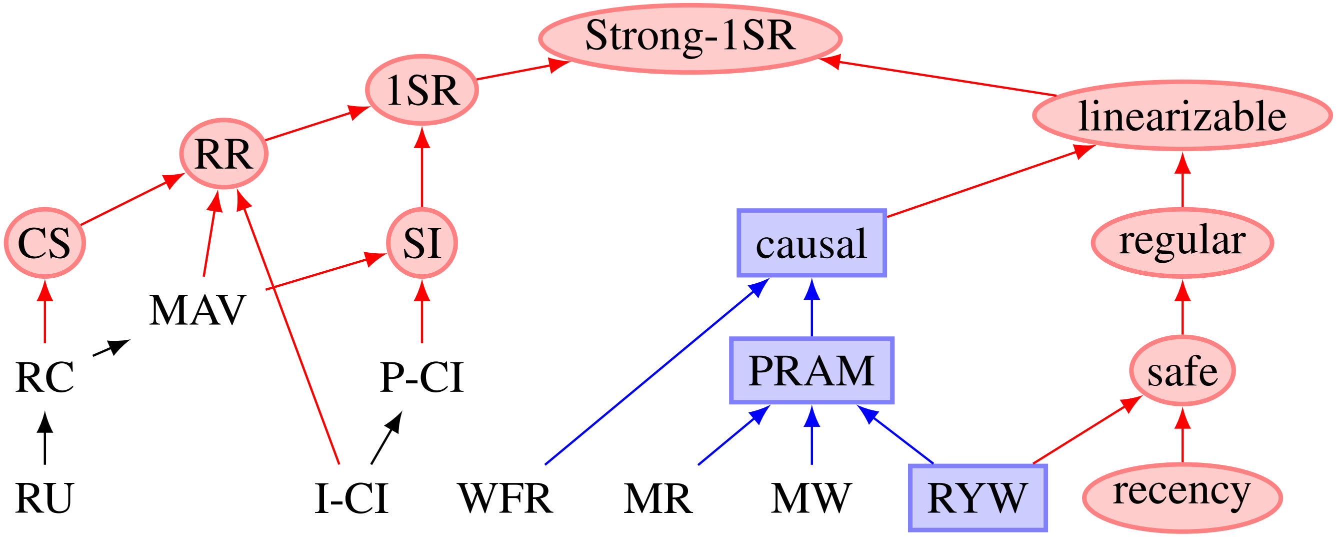
Lower number of RTTs



Fastpass: A Centralized "Zero-Queue" Datacenter Network

Jonathan Perry, Amy Ousterhout, Hari Balakrishnan, Devavrat Shah, Hans Fugal
M.I.T. Computer Science & Artificial Intelligence Lab Facebook
http://fastpass.mit.edu/

(4) Use other consistency models



Bounded latency vs. consistency (PBS)

Highly Available Transactions