On Simulation and Design of Parallel-Systems Schedulers

Are We Doing the Right Thing?

Xinyu Chen

April 13, 2017

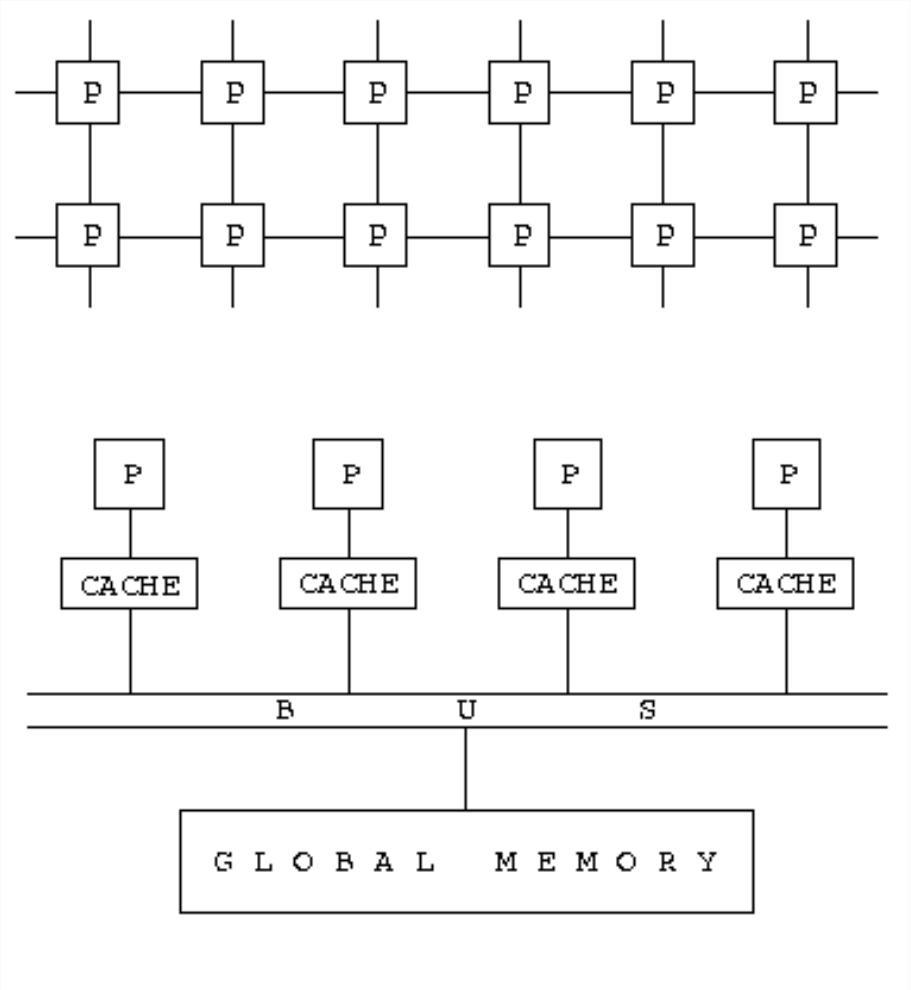
University of New Mexico

Outline

1. A Little Background
2. Scheduling Simulator
3. Experiments Result
4. Conclusion and Related Work
5. Questions

A Little Background

Parallel System Architecture

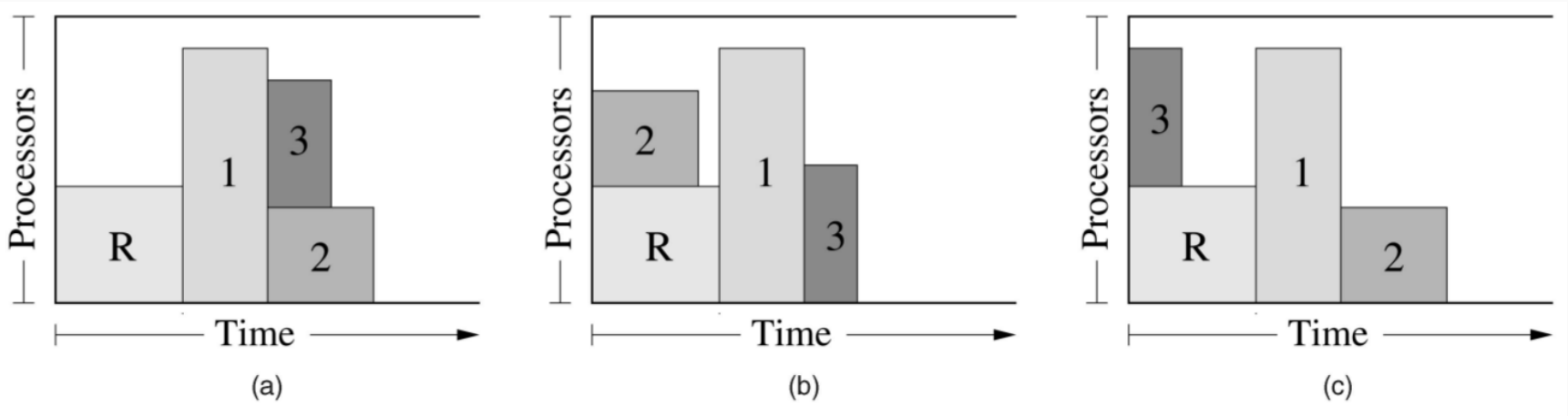


- Distributed-memory model
- shared-memory model

http://www.mcs.anl.gov/ itf/dbpp/

Scheduling Simulator

Conventional Schedulers and Simulations



Trace-driven Workload

- Close-System
- Open-System

Some Metrics

- Response Time = $t_{terminate} - t_{start}$
- Slowdown = $\frac{resp\_time}{actual\_runtime}$
- Thinking Time=$t_{new\_submission} - t_{last\_finish}$

Site-level Simulations

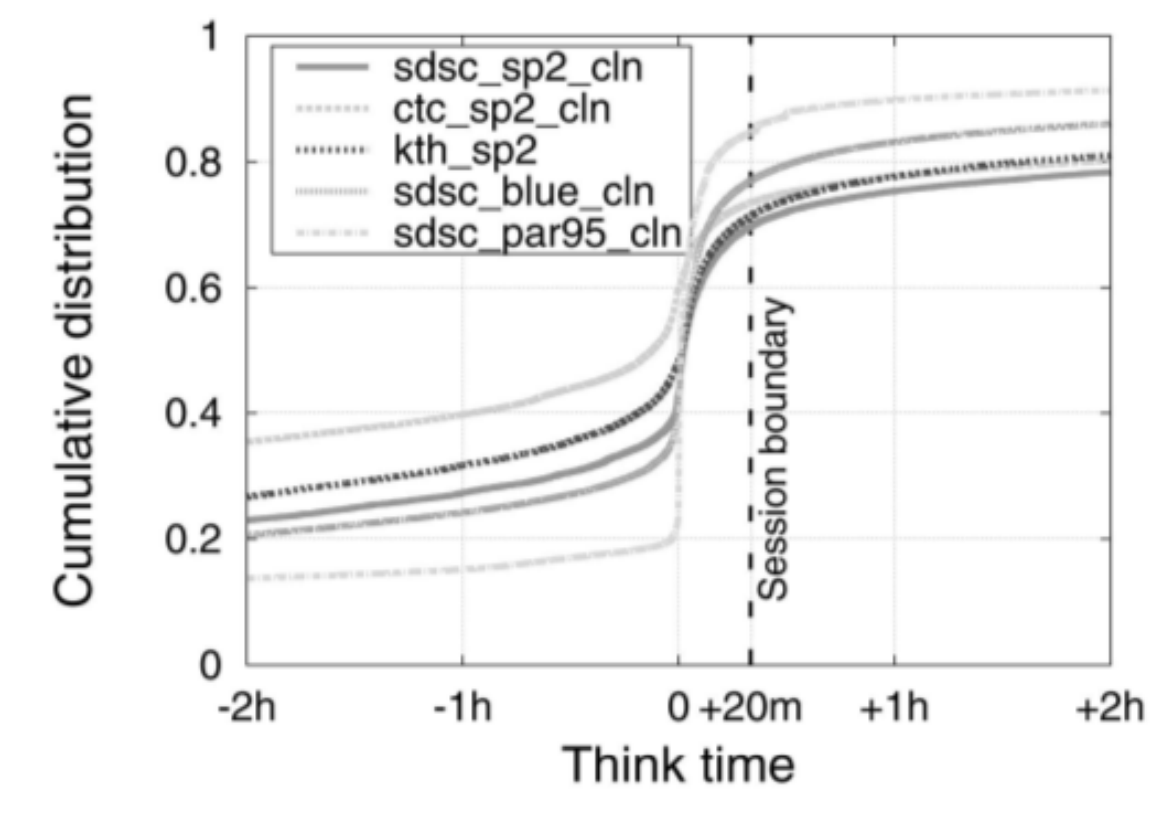
Table: Methodological Difference between Two Types of Simulation

Category	Conventional Simulations	Site-Level Simulations
Workload source	System traces	User models
Workload generation	Open-system model	User-scheduler interaction
Load scaling	Trace (de)-compression	Number of users
Performance Metrics	Response time, slowdown	Throughput, session length

User Model

- Session Dynamic
- Job Submission
- Activity Cycle

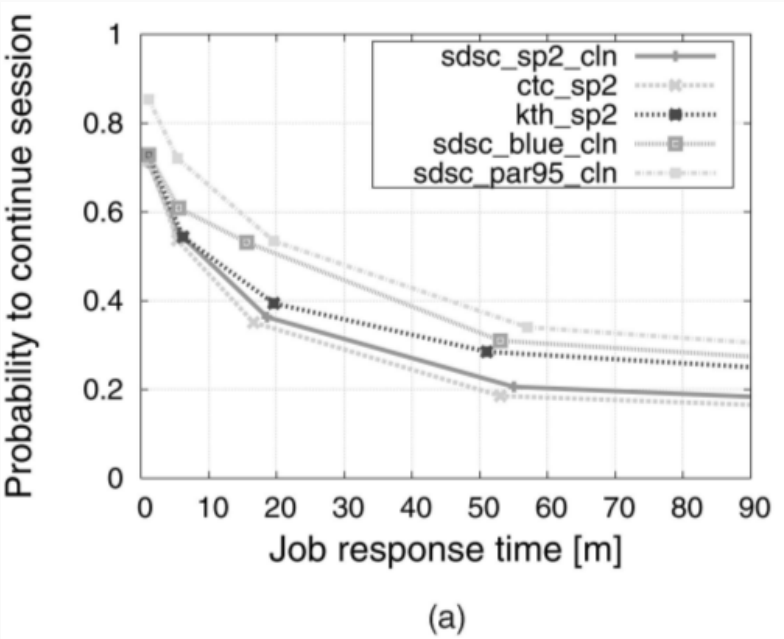
User Behavior - From Think Time to Sessions

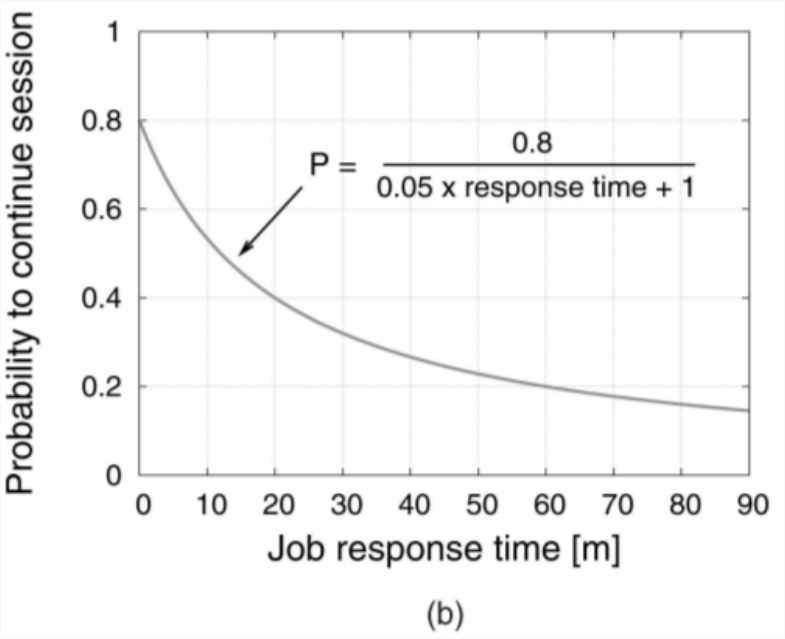


Potential Problems

The CDF of think time looks symmetric. The negative part indicates script submit.

User Behavior - Session Dynamics





Probability to continue session

$$p\_cont(j) = \frac{0.8}{0.05 \times resp\_time(j) + 1}$$

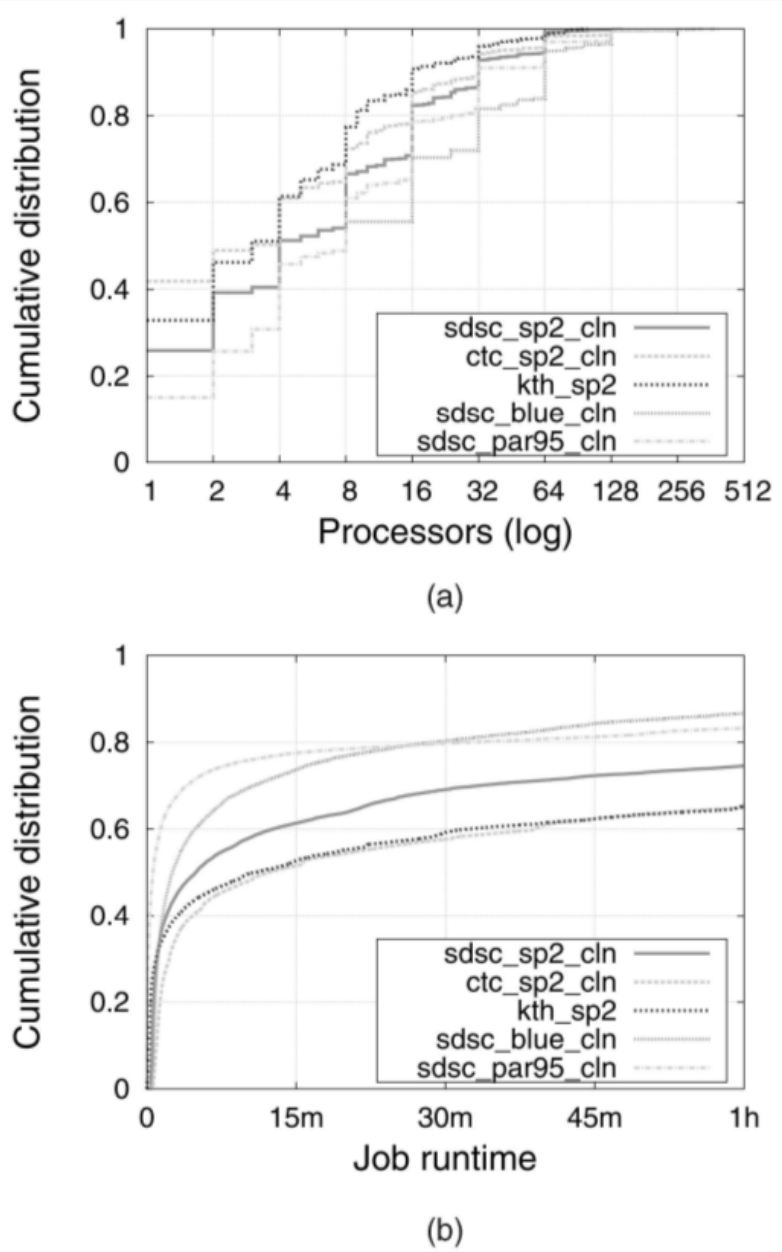
Potential Problems

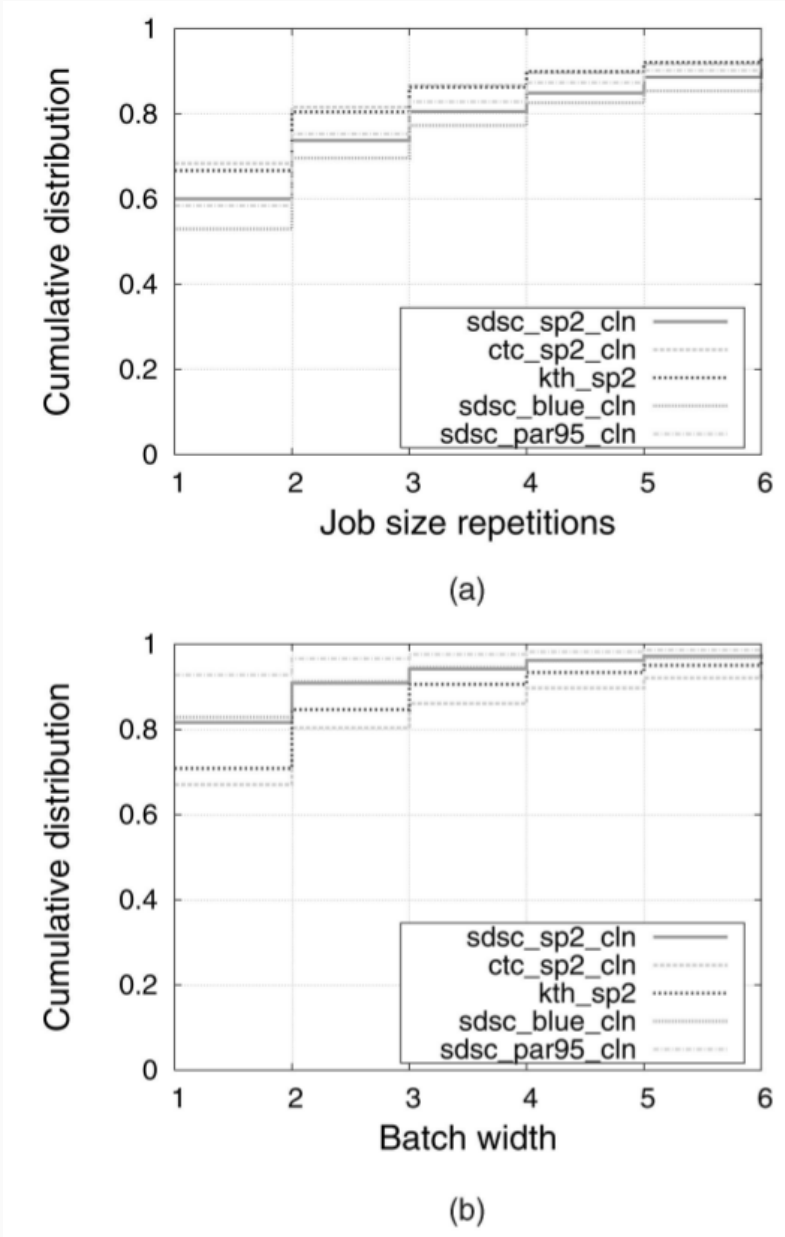
Regression?

Confounding Factors?

Causal Relationship?

User Behavior - Job Submission

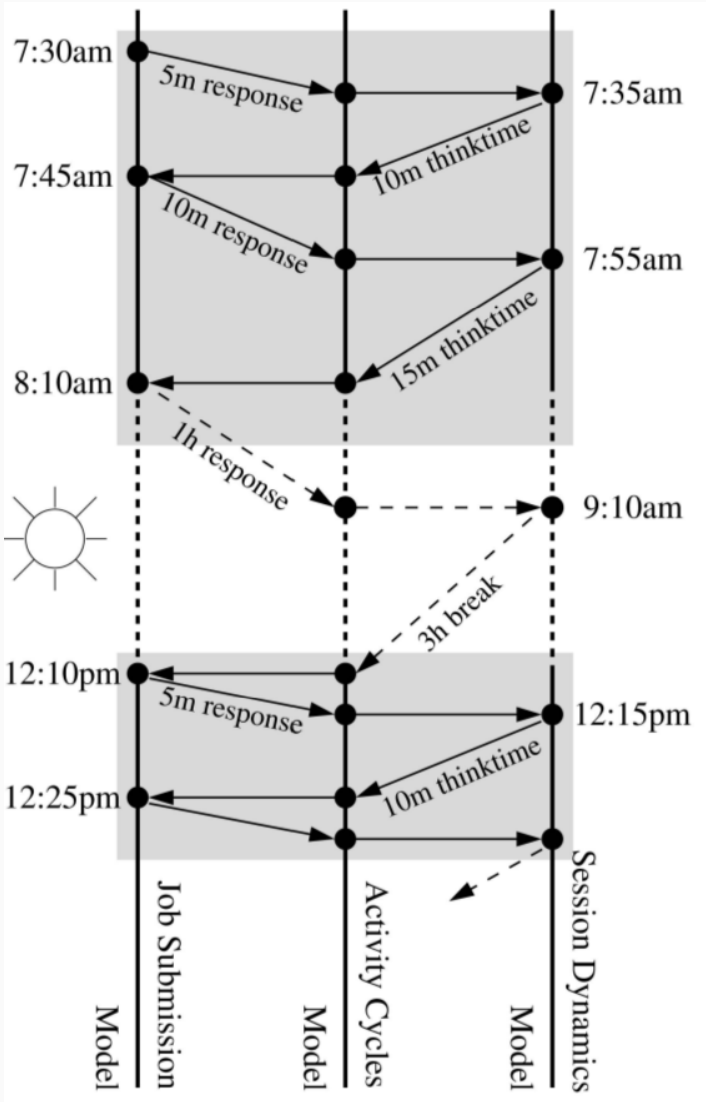




User Behavior - Activity Cycle

Table: Four user cycle classes

Daytime-weekdays	Daytime-weekend
Nighttime-weekdays	Nighttime-weekend



Recap User Model

- Workload source
- Workload Generation

User Aware Scheduling

Similar to EASY

$$criticality(j) = \frac{0.04}{(0.05 \times estimate\_resp\_time(j)+1)^2}$$

$$priority(j) = \alpha \times criticality(j) + seniority(j)$$

$$estimate\_resp\_time(j) = seniority(j) + runtime(j)$$

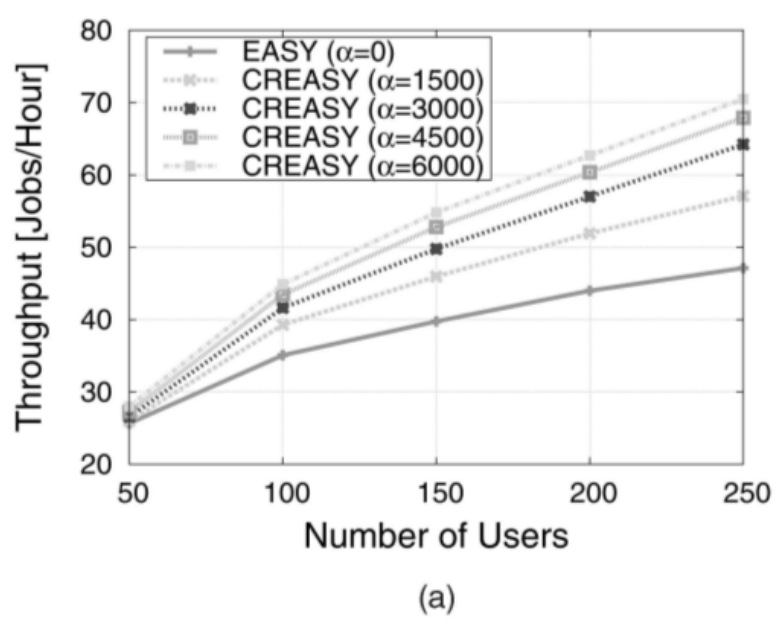
Potential Problems

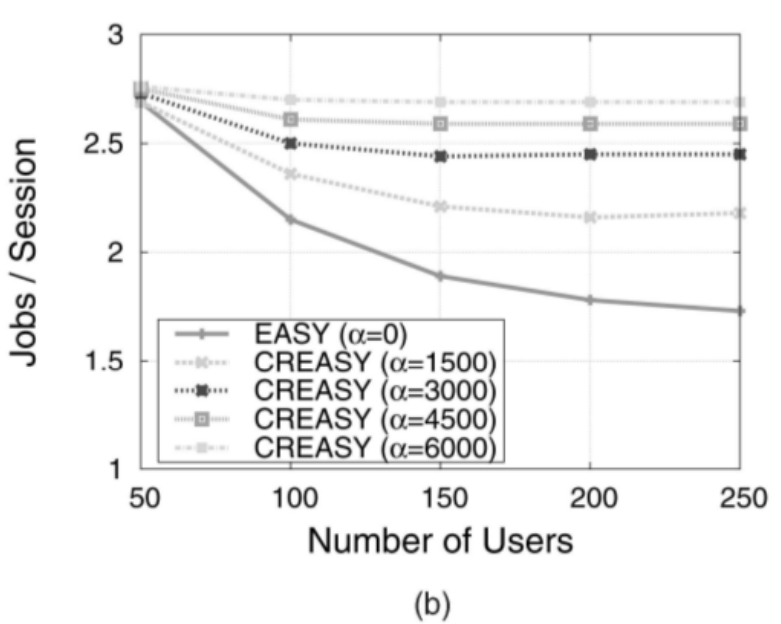
criticality tends to starve long jobs

How to decide and interpret α values

Experiments Result

Metrics: Throughput and Session Length





Potential Problems

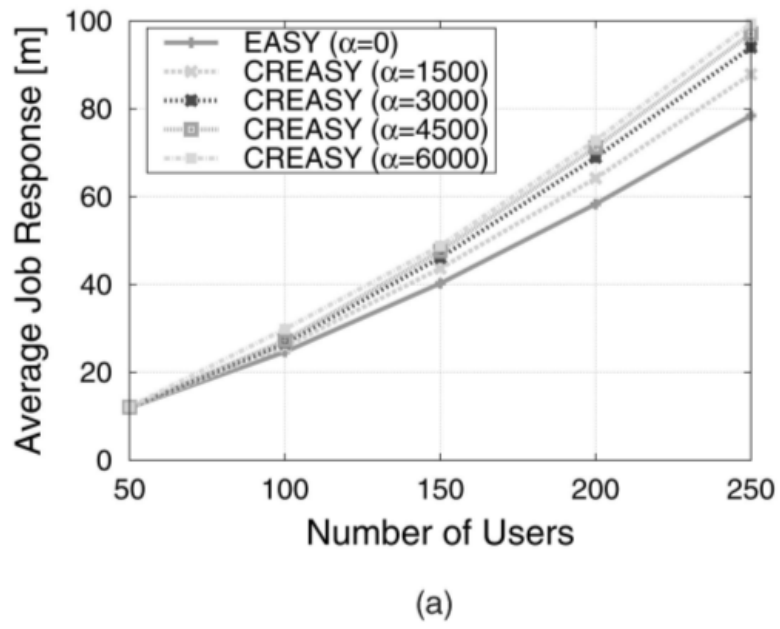
Is Number of Users related with Productivity?

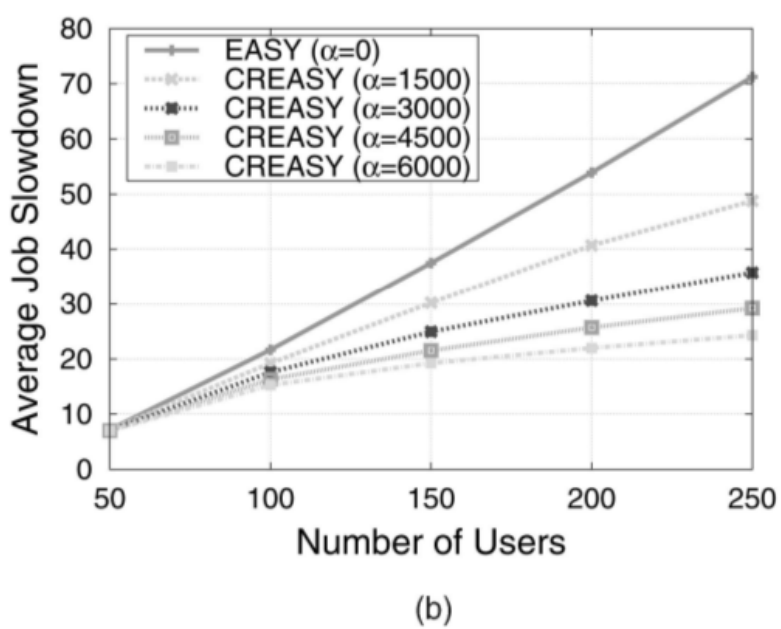
How many processors are used?

What are the job sizes?

Larger number of users favors large α, which favors short jobs.

Metrics: Response Time and Slowdown



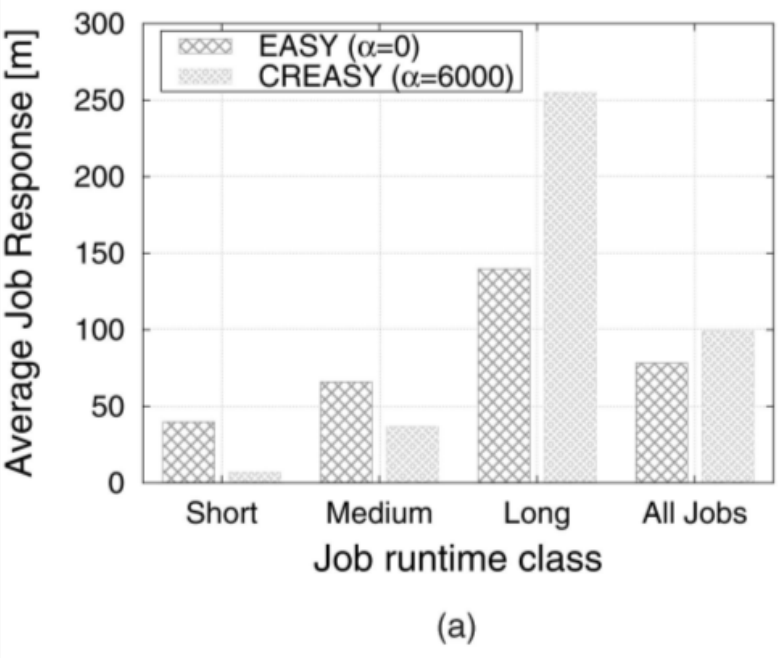


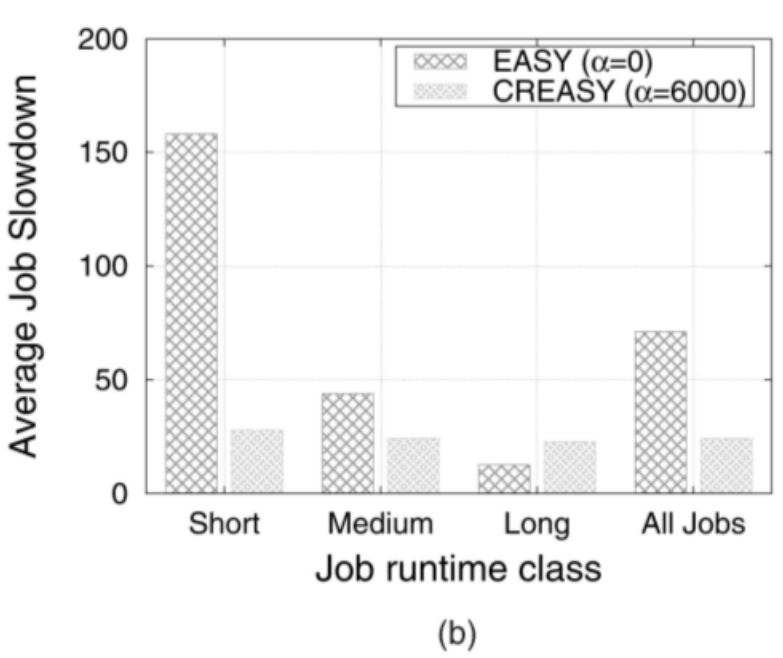
Potential Problems

Is Average Response Time and Slowdown useful? What is the standard error?

Slowdown should be greater than 100%.

Metrics: Response Time and Slowdown in Different Classes





Potential Problems

It's clear in the above two graphs that the scheduler favors small jobs and sacrifices big jobs.

Slowdown should be greater than 100%

Conclusion and Related Work

Conclusion

- Trace-driven Scheduling Simulation cannot reflect user-system interactions.
- Conventional metrics like Response Time and Slowdown cannot help improve user productivity
- The User Model is much simplified. Real user behaviors are more complicated.
- The interactive sessions is critical but only require less resources. Script submission need to be considered in the future.

Some Previous Discussed Papers

- Looking at Data by Dror G. Feitelson
- How Uber Uses Psychological Tricks to Push Its Drivers' Buttons

Questions