Presentation about: Rodrigo Álvarez, G.P, Elmroth, E., Östberg, P.O., Ramakrishnan, L. ScSF: A Scheduling Simulation Framework. 21th Workshops on Job Scheduling Strategies for Parallel Processing (JSSPP 2017)

ScSF: A **Sc**heduling **S**imulation **F**ramework

JSSPP'17

Gonzalo P. Rodrigo Álvarez
gprodrigoalvarez@lbl.gov

ScSCF is open and **available at:**
http://frieda.lbl.gov/download

June 2nd, Orlando, Florida

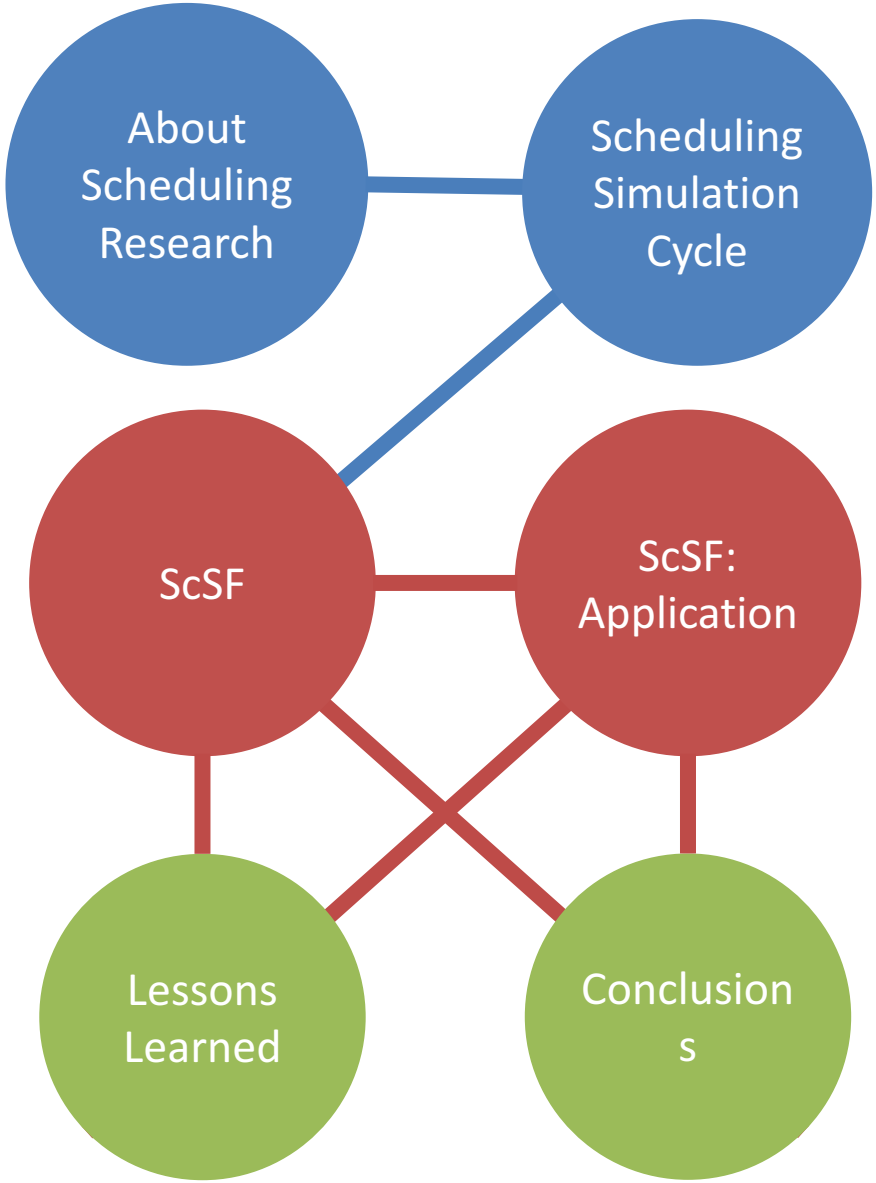
Outline

Introduction

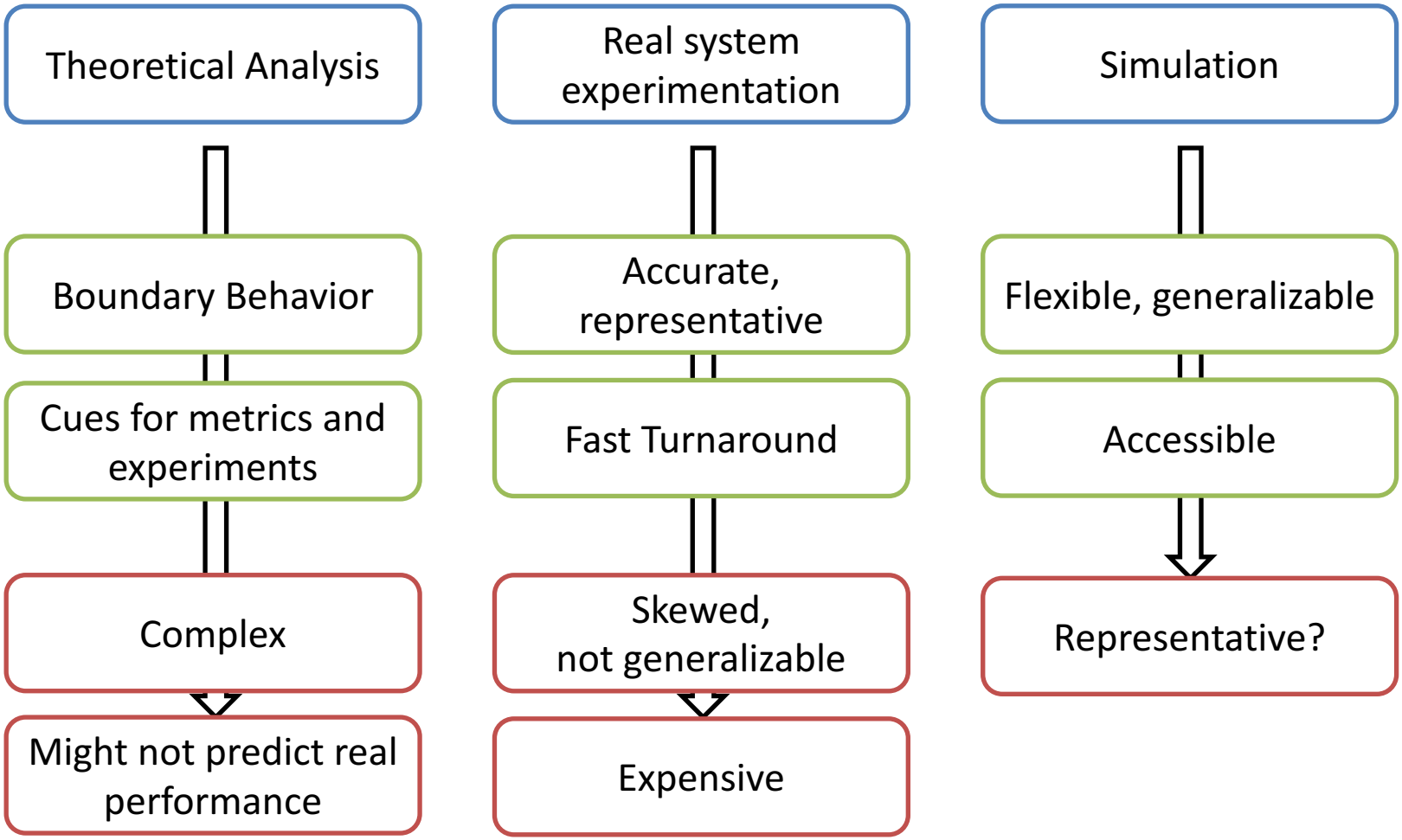
Contributions

The System

Concepts

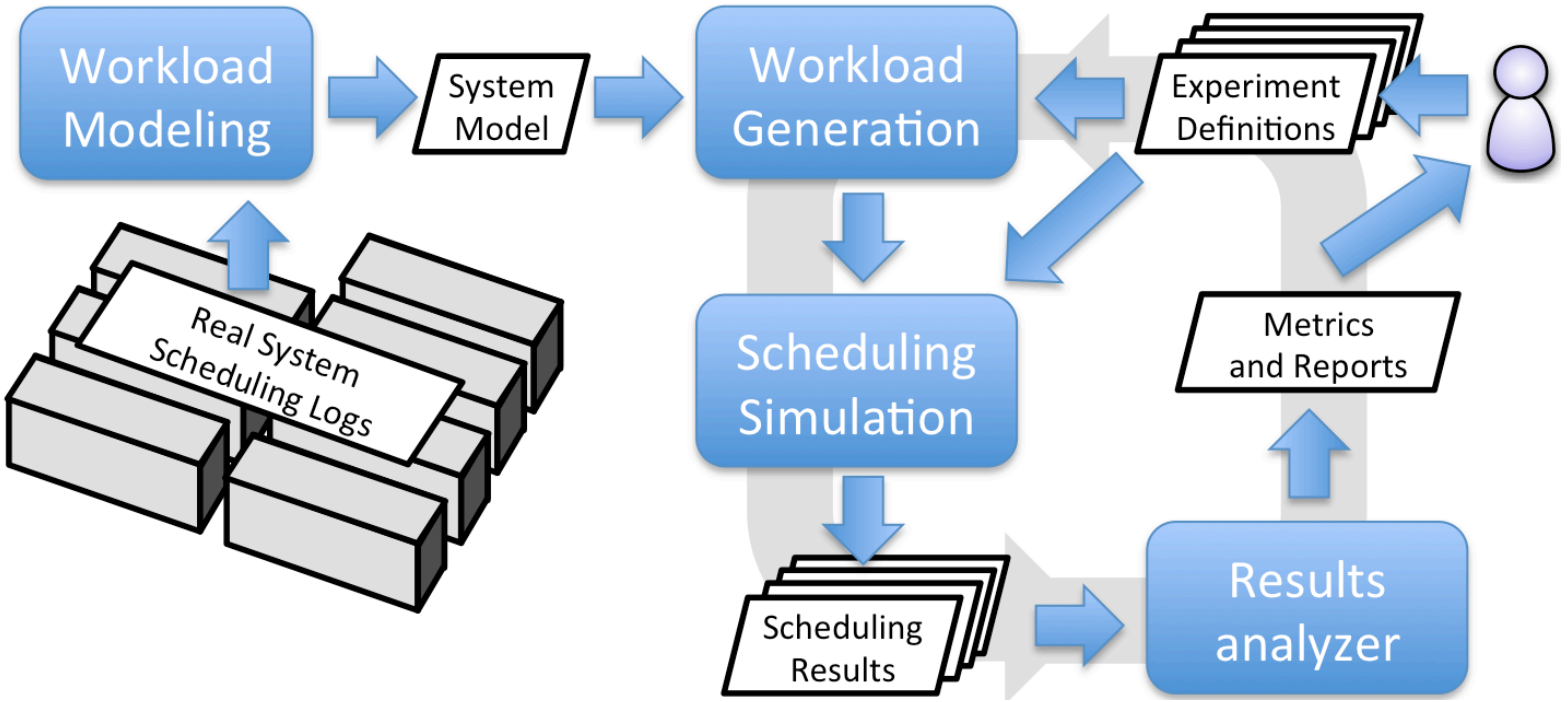


HPC Scheduling Research approaches



Schwiegelshohn, U.: How to design a job scheduling algorithm. In: Workshop on Job Scheduling Strategies for Parallel Processing. pp. 147–167. Springer (2014)

HPC Scheduling Simulation: Research cycle



	Modeling	Generation	Simulation	Analysis
Alea/BatSim	✗	✗	✓ Not prod. scheduler	✗
Slurm Simulator	✗	✗	✓ Slow	✗
Parallel Archive	✗	✓ Old Small	✗	✗

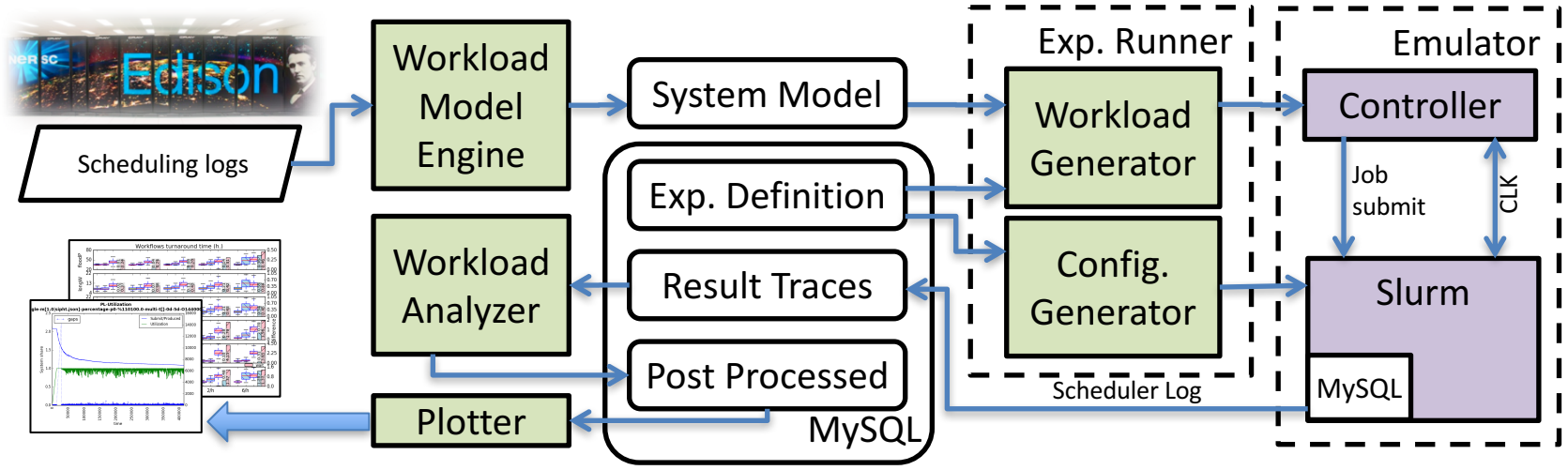
HPC Scheduling Simulation: Research cycle

Complete HPC Scheduling research tools

Not publicly available!

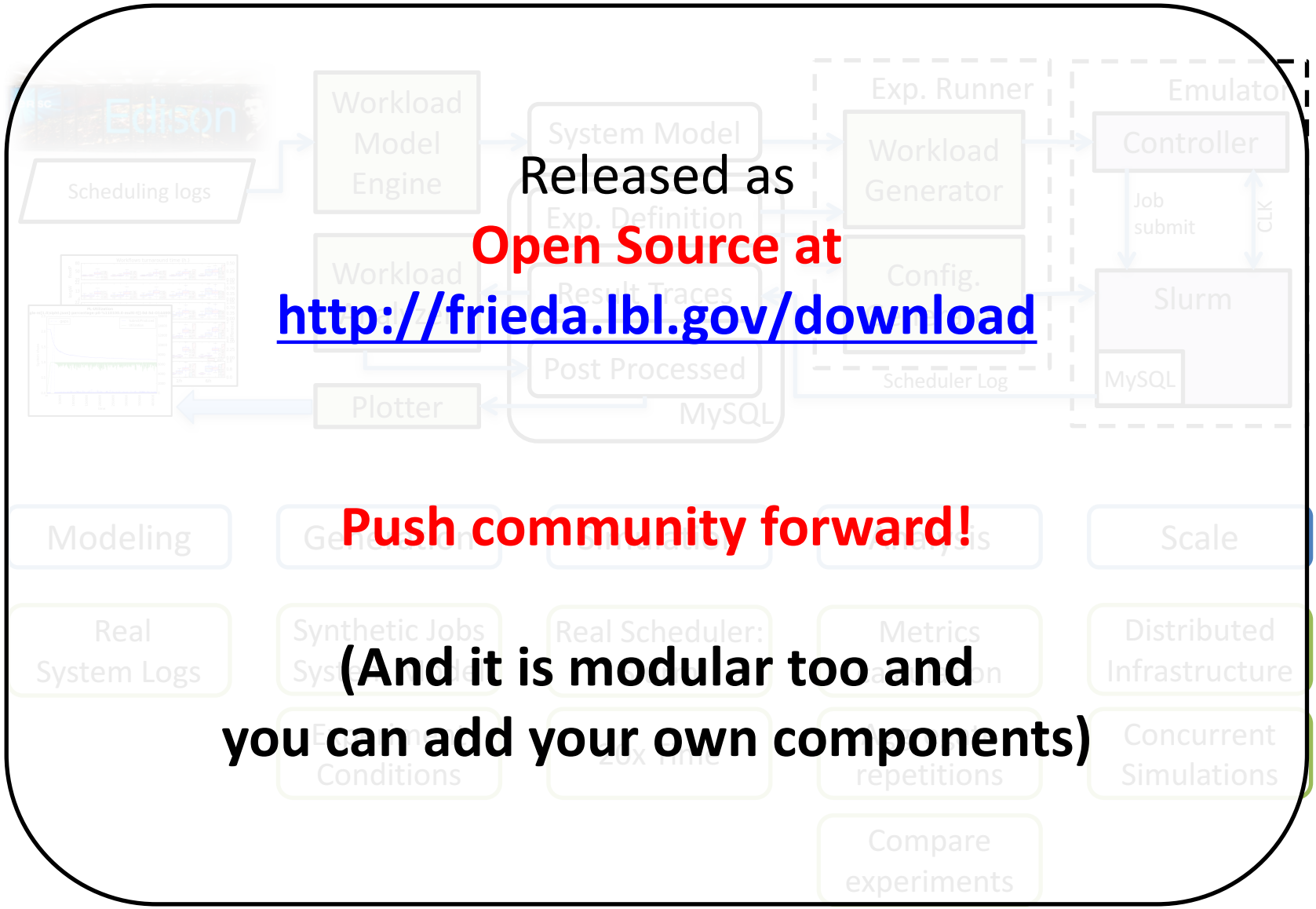
(At least not in 2015-2016 ☺)

ScSF: Scheduling Simulation Framework



Modeling	Generation	Simulation	Analysis	Scale
Real System Logs	Synthetic Jobs System Model	Real Scheduler: Slurm	Metrics calculation	Distributed Infrastructure
	Experiment Conditions	20x Time	Aggregate repetitions	Concurrent Simulations
			Compare experiments	

ScSF: Scheduling Simulation Framework



Released as

Open Source at

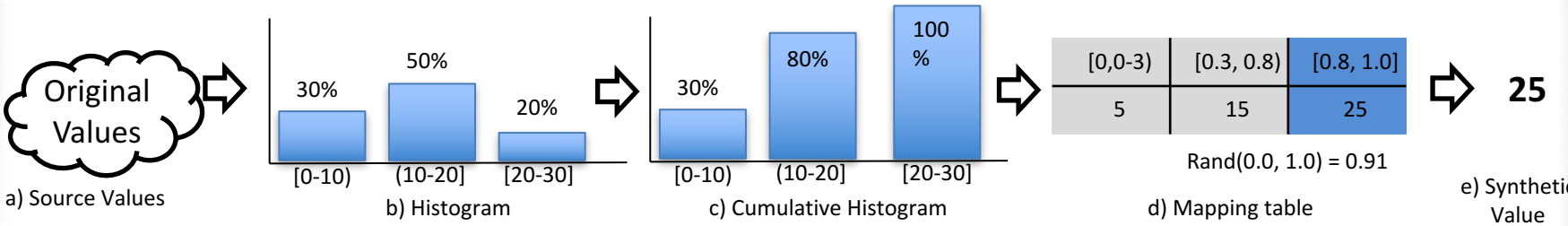
http://frieda.lbl.gov/download

Push community forward!

(And it is modular too and you can add your own components)

ScSF: Workload Modeling & Generation

Empirical Distribution



Gonzalo P. Rodrigo – gprodrigoalvarez@lbl.gov

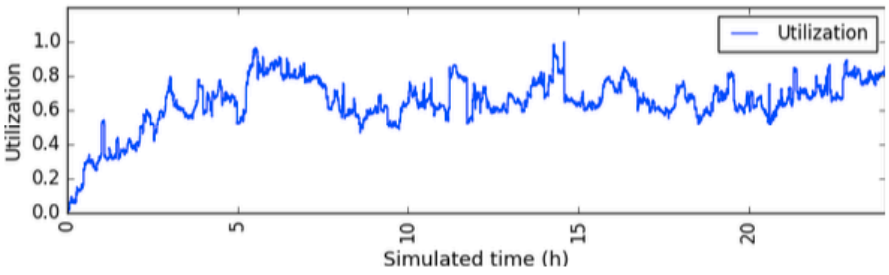
ScSF: Workload Modeling & Generation

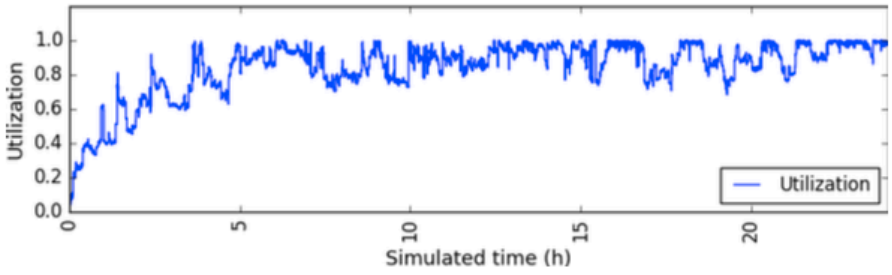
Empirical Distribution

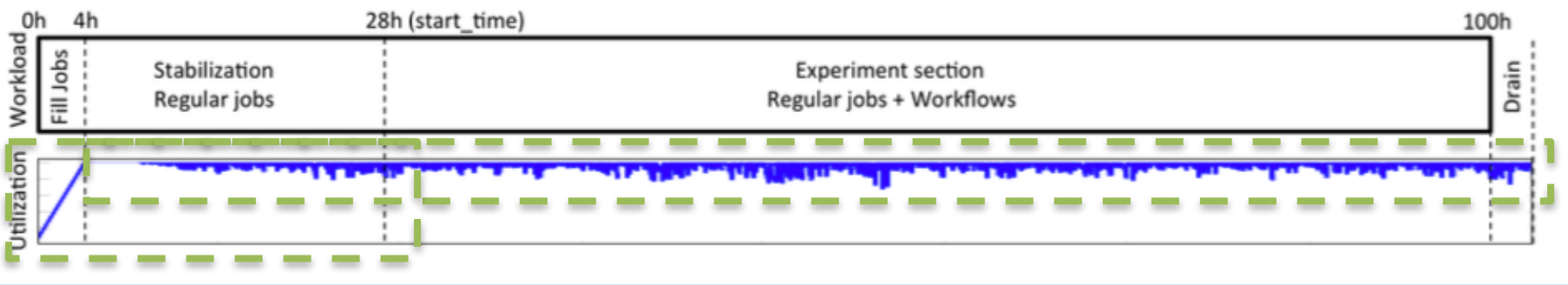
Open Loop

Job Pressure Control

Cold start stabilization

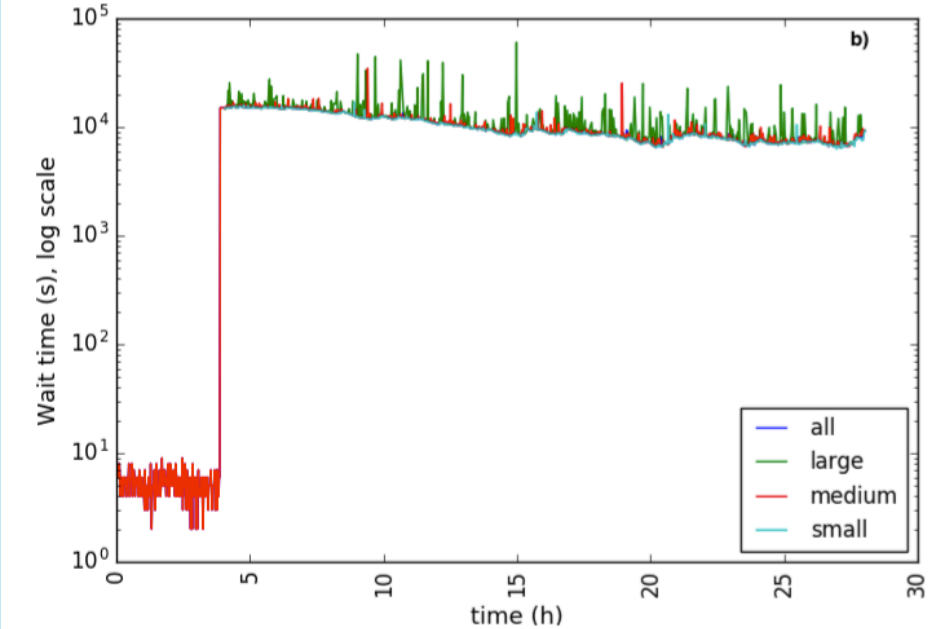






ScSF: Workload Modeling & Generation

- Empirical Distribution
- Open Loop
- Job Pressure Control
- Cold start stabilization
- Baseline wait time



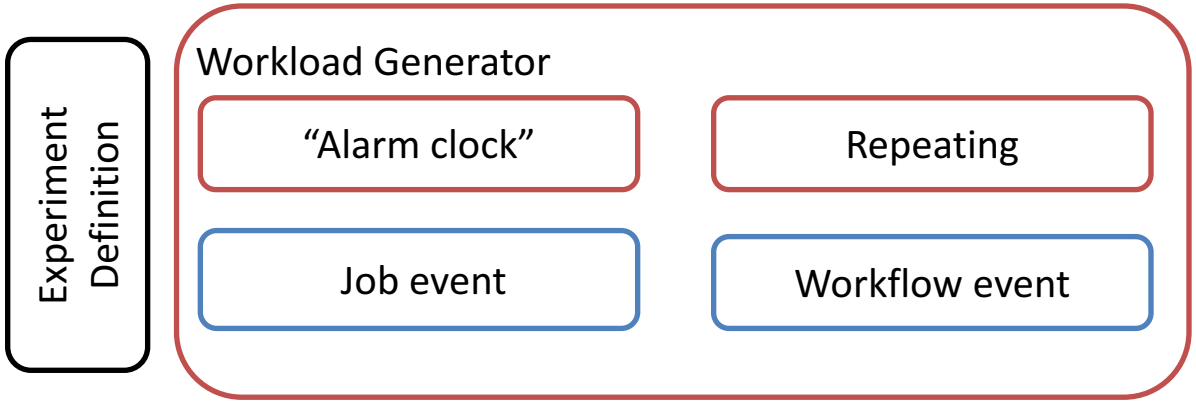
ScSF: Workload Modeling & Generation

- Empirical Distribution
- Open Loop
- Job Pressure Control
- Cold start stabilization
- Baseline wait time
- Workflows

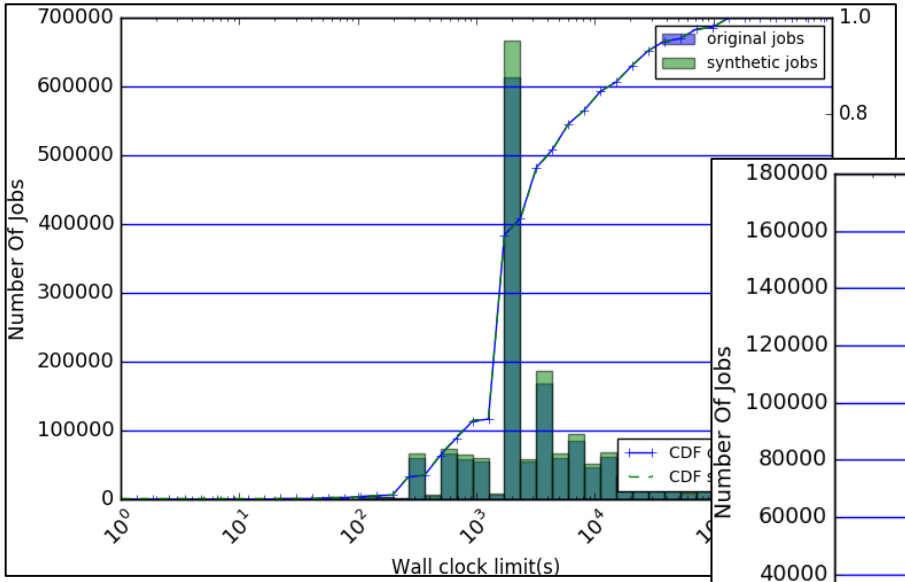
```
{"tasks": [
  {"id":"SWide", "cmd":"./W.py", "cores":480, "rtime":360.0},
  {"id":"SLong", "cmd":"./L.py", "cores":48, "rtime":1440.0,
    "deps": ["SWide"]}]}
```

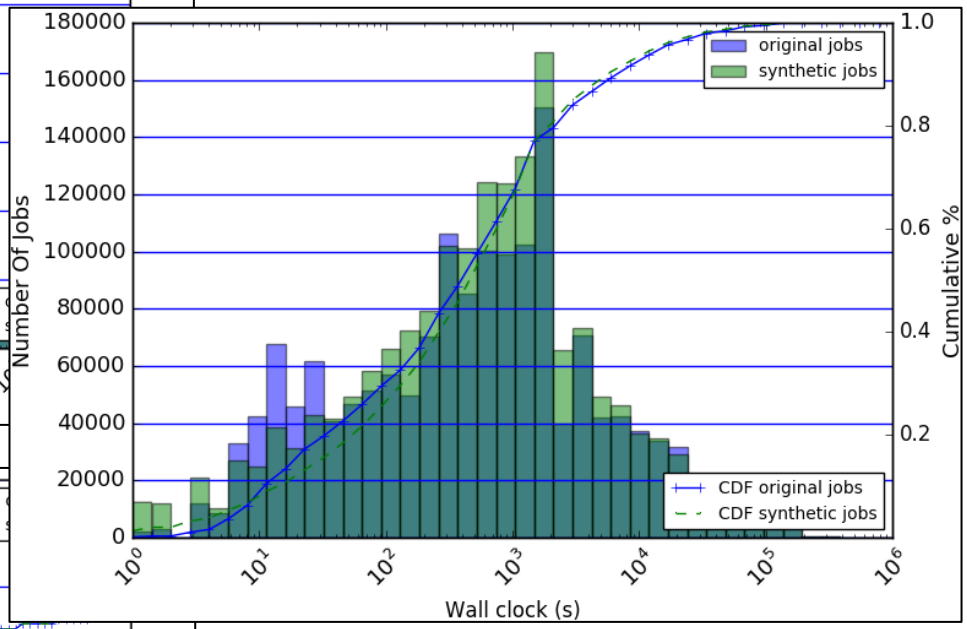
ScSF: Workload Modeling & Generation

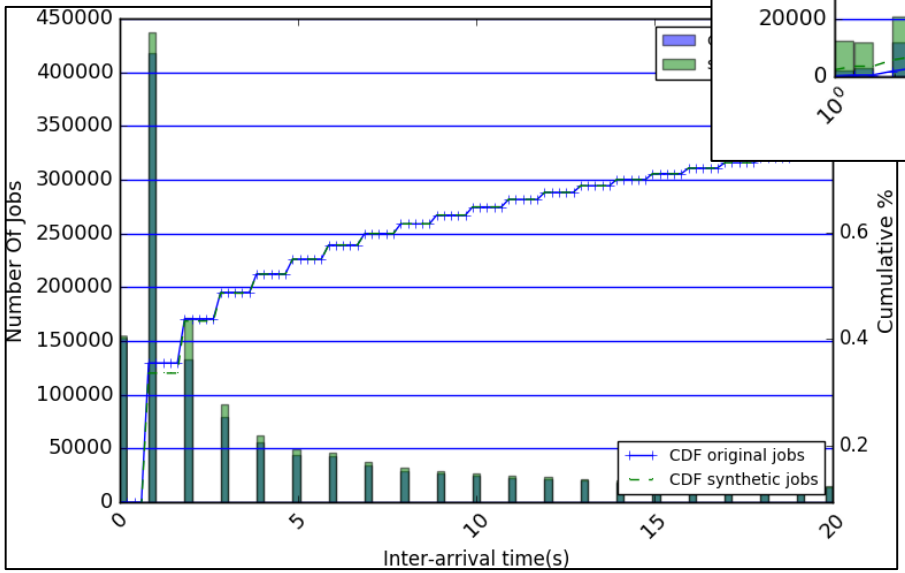
- Empirical Distribution
- Open Loop
- Job Pressure Control
- Cold start stabilization
- Baseline wait time
- Workflows
- Programmable API



ScSF: Workload Modeling & Generation







Peak: **2.57 Petaflops/s**.

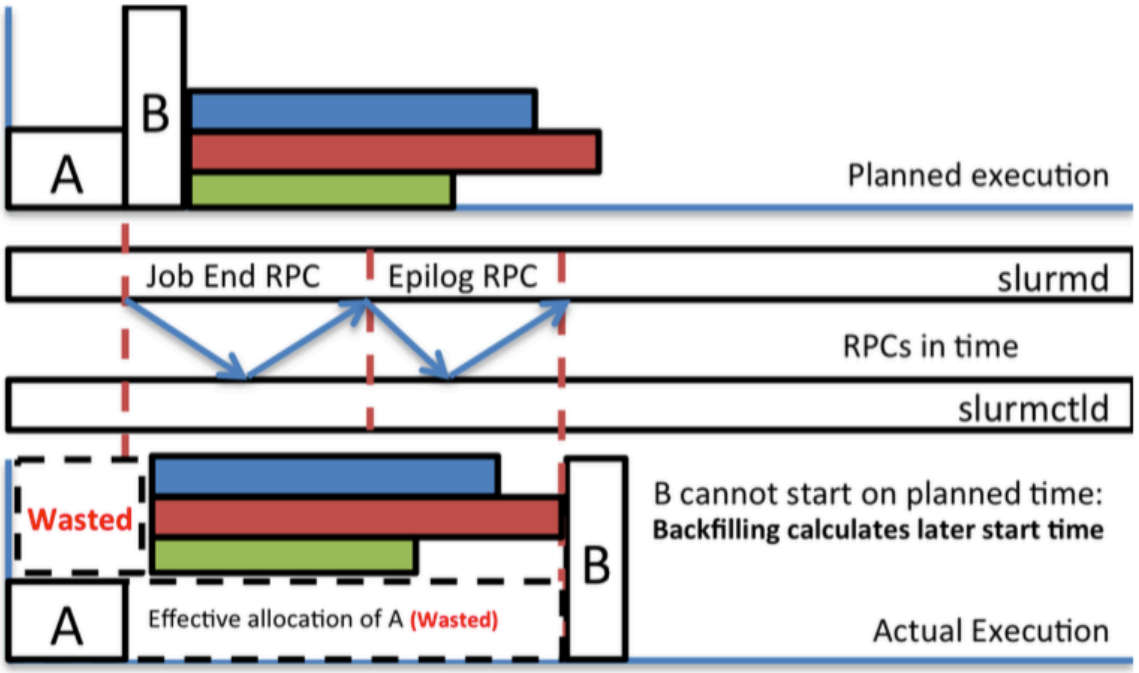
ScSF: Slurm Simulator

Wraps real Slurm Scheduler

Emulates system and job execution

Emulates job submission (replay)

Original Implementation: Slow (1 to 1), no determinism



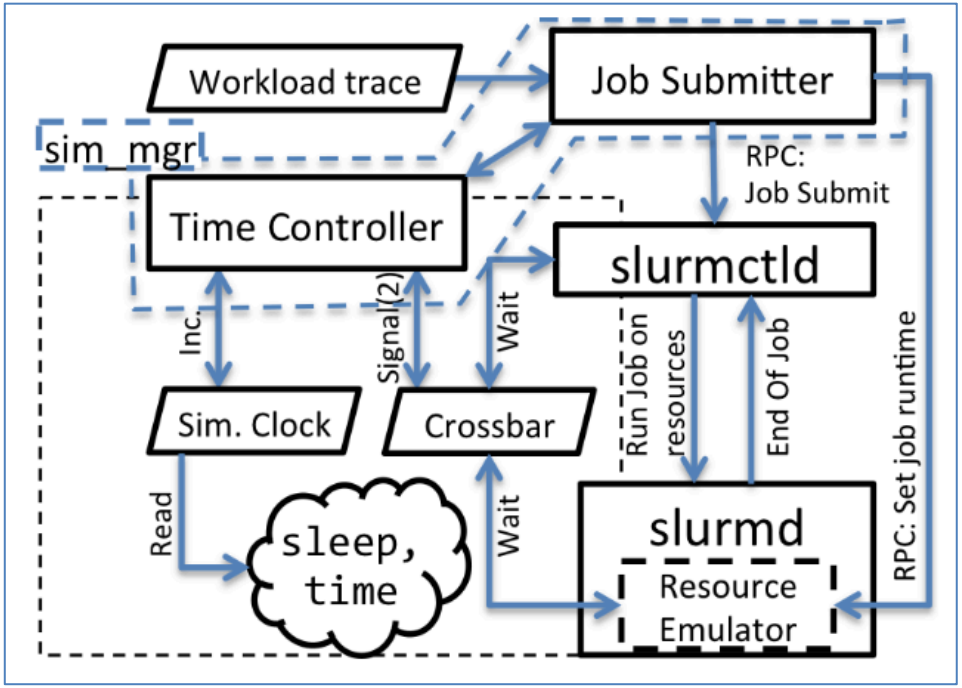
ScSF: Slurm Simulator

Wraps real Slurm Scheduler

Emulates system and job execution

Emulates job submission (replay)

Original Implementation: Slow (1 to 1), no determinism



Slurm simulator improved by synchronizing scheduling threads

Faster (20x speed-up)

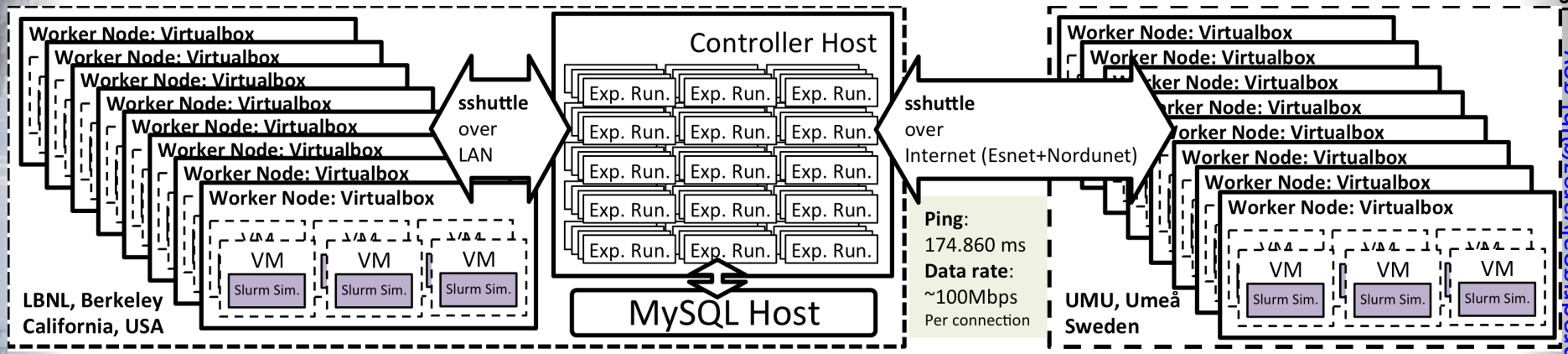
Time consistent

Achieve good utilization with out-of-the-box scheduler

ScSF: Running experiments in scale

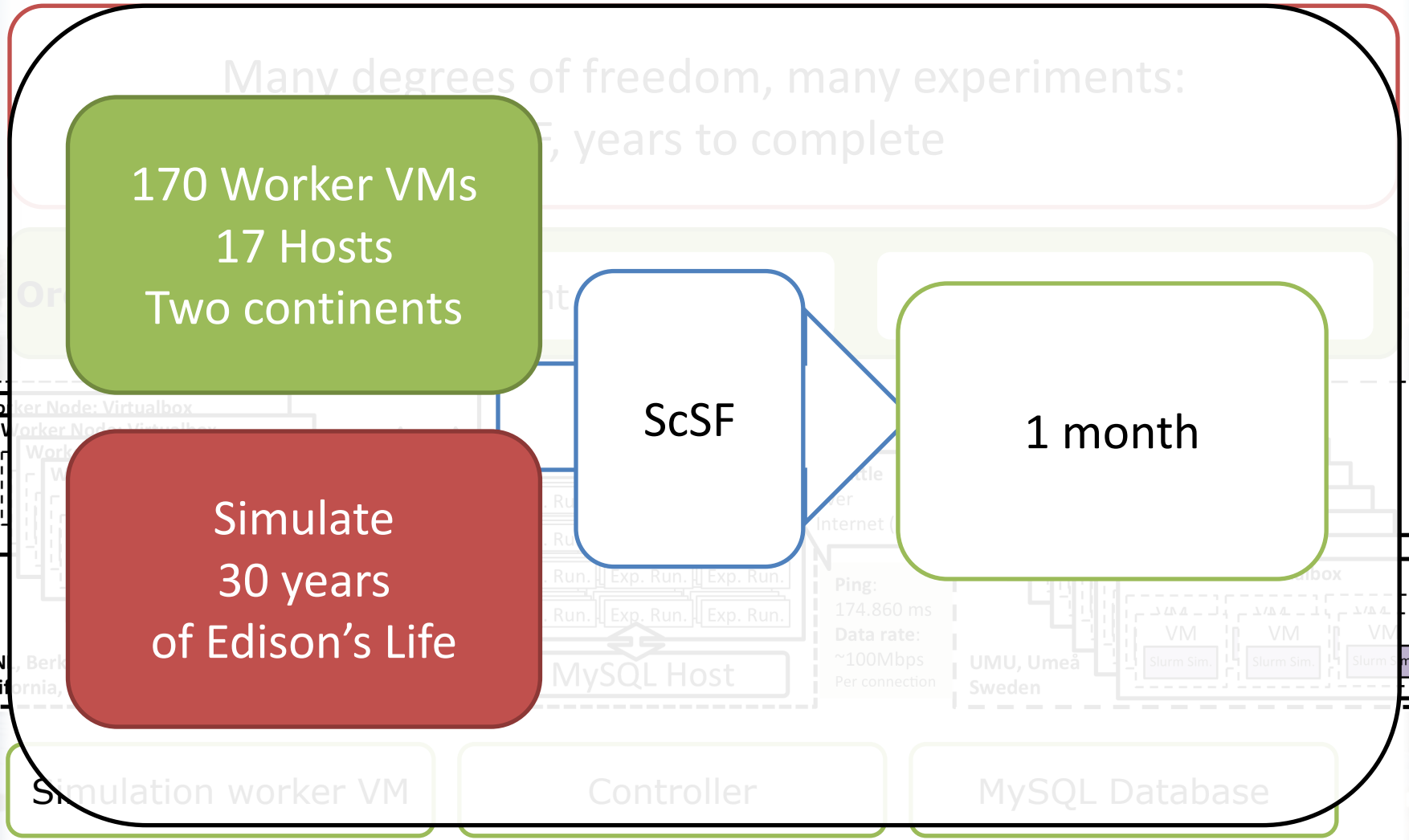
Many degrees of freedom, many experiments:
ScSF, years to complete

Orchestrator | Concurrent experiments | Distributed resources



Simulation worker VM | Controller | MySQL Database

ScSF: Running experiments in scale



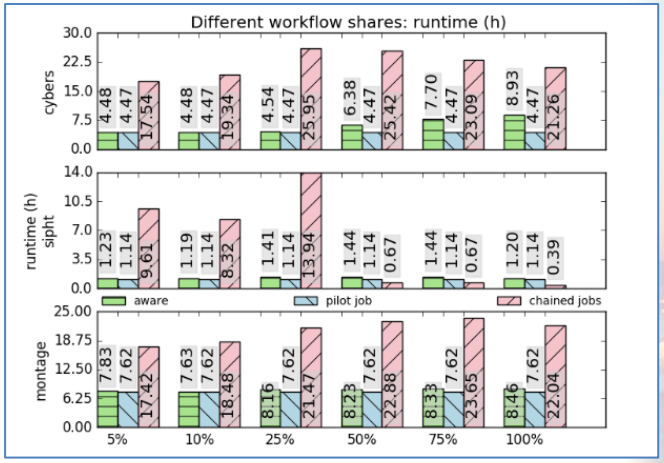
ScSF: Analysis Capabilities

Job Variables

Workflow variables

Histogram+ CDF

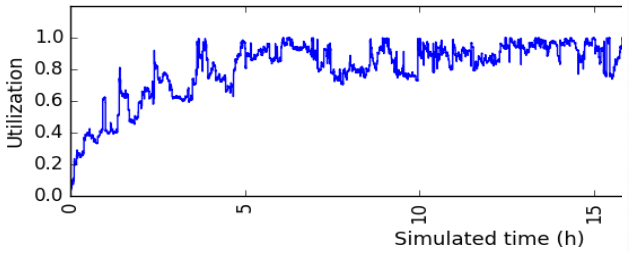
Average, Median, Percentiles



Utilization

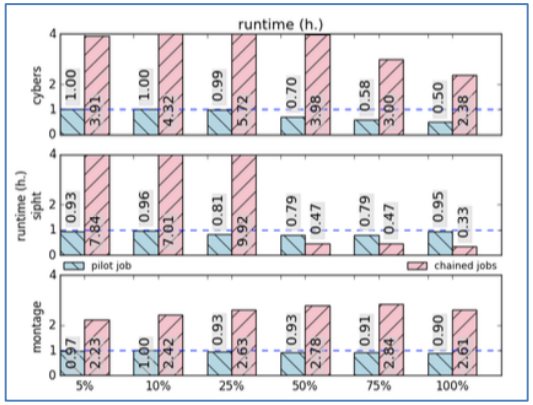
Aggregated

Per Minute



Aggregating results from repetitions

Compare experiments with similar configuration



Gonzalo P. Rodrigo – gprodrigoalvarez@lbl.gov

ScSF: Research Use case, WoAS*

WoAS: An scheduling technique to minimize workflows turnaround time without over allocating resources

Questions to answer with ScSF

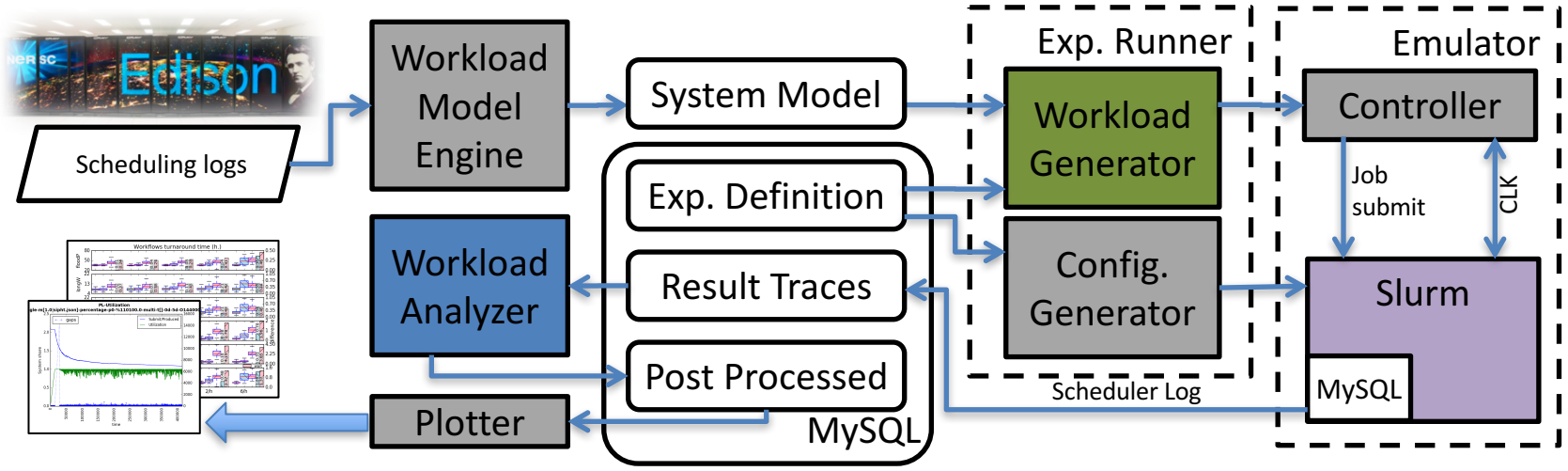
Can the WoAS scheduling technique minimize workflows turnaround time without over allocating resources?

How much shorter is turnaround time with WoAS? (Compared to existing techniques)

Does WoAS impact negatively on non-workflow jobs? (Slowdown)

* Results to be published in **HPDC'17**: "Enabling Workflow-Aware Scheduling on HPC systems"

ScSF: Getting ready for WoAS *



1: Generate workload with WoAS workflows

2: Implement WoAS in Slurm

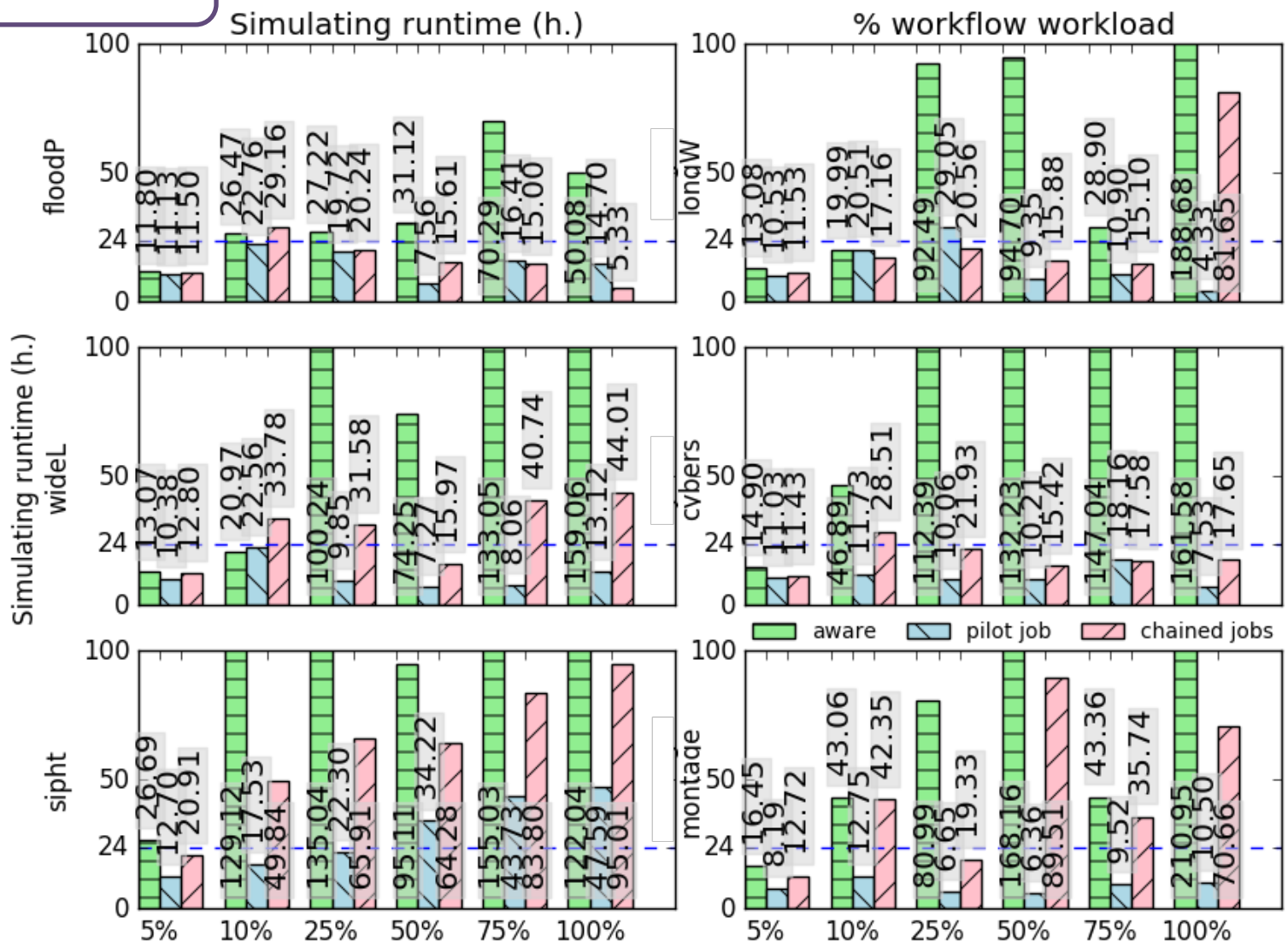
3: Identify WoAS workflows

271 scenarios,
1728 experiments
33 years of supercomputer time!
38 Billion Core-hours (simulated)

* Results to be published in **HPDC'17**: "Enabling Workflow-Aware Scheduling on HPC systems"

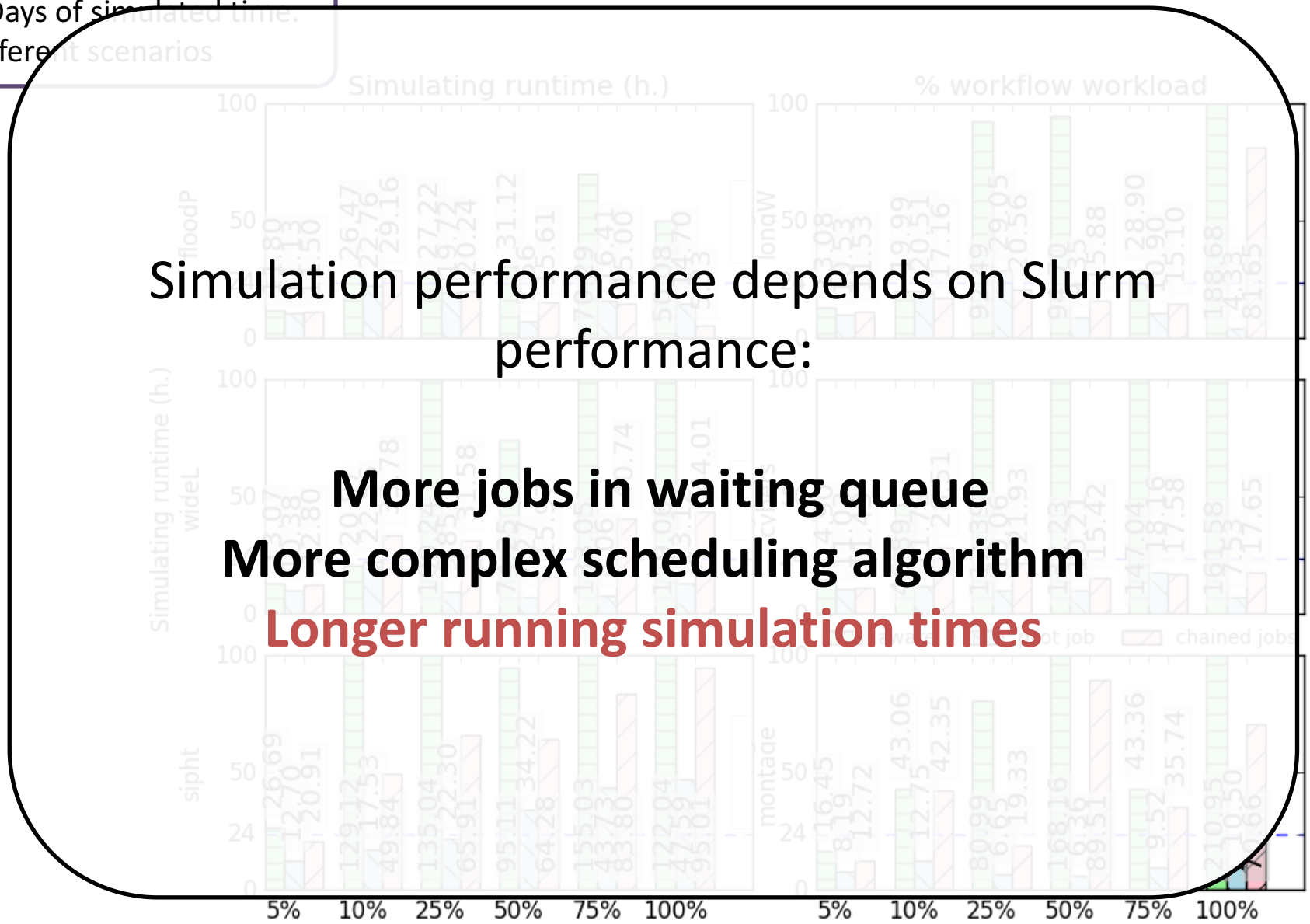
ScSF: Simulation performance on WoAS

6 Days of simulated time.
Different scenarios



ScSF: Simulation performance on WoAS

6 Days of simulated time.
Different scenarios



Simulation performance depends on Slurm performance:

More jobs in waiting queue

More complex scheduling algorithm

Longer running simulation times

ScSF: Lessons learned

Slurm is a complex old-fashion-SWE package:
expensive to modify

Loss-less experiment restart is needed
Specially if experiment runtime are long (e.g. 5 days)

Monitoring is important
To debug why something fails (and things will fail)

Loaded systems network fail
So harden your comms

The system is as weak as its weakest link
Single point of failure

ScSF: Lessons learned

Slurm is a complex old-fashion-SWE package:
expensive to modify

Loss-less experiment restart is needed
Specially if experiment runtime are long (e.g. 5 days)

HPC scheduling requires a lot of simulation

Think big from the beginning!

To debug at large scale things will fail)

Loaded systems network fail
So harden your comms

The system is as weak as its weakest link
Single point of failure

ScSF: Summary

HPC Scheduling research cycyle:
Model/generate workloads -> scheduling emulation -> analysis

Tools to run experiments in scale

Slurm simulator in its core: A production HPC simulator

Open Source! **Use it!** (**http://frieda.lbl.gov/download**)
And **modular:** free to replace any of the parts

(Keep scale in mind!)

THANKS

For any questions, please contact:
gprodrigoalvarez@lbl.gov

To know more, read: Rodrigo Álvarez, G.P, Elmroth, E., Östberg, P.O., Ramakrishnan, L. ScSF: A Scheduling Simulation Framework. 21th Workshops on Job Scheduling Strategies for Parallel Processing (JSSPP 2017)

ScSF is open source and **available at:**
http://frieda.lbl.gov/download

Supported by U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research (ASCR). the National Energy Research Scientific Computing Center, a DOE Office of Science User Facility, is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. Financial support has been provided in part by the Swedish Government's strategic effort eSSENCE and the Swedish Research Council (VR) under contract number C0590801 (Cloud Control).