BiDAl: Big Data Analyzer for Cluster Traces

Alkida Balliu, Dennis Olivetti, Ozalp Babaoglu,
Moreno Marzolla, Alina Sirbu

Department of Computer Science and Engineering
University of Bologna, Italy

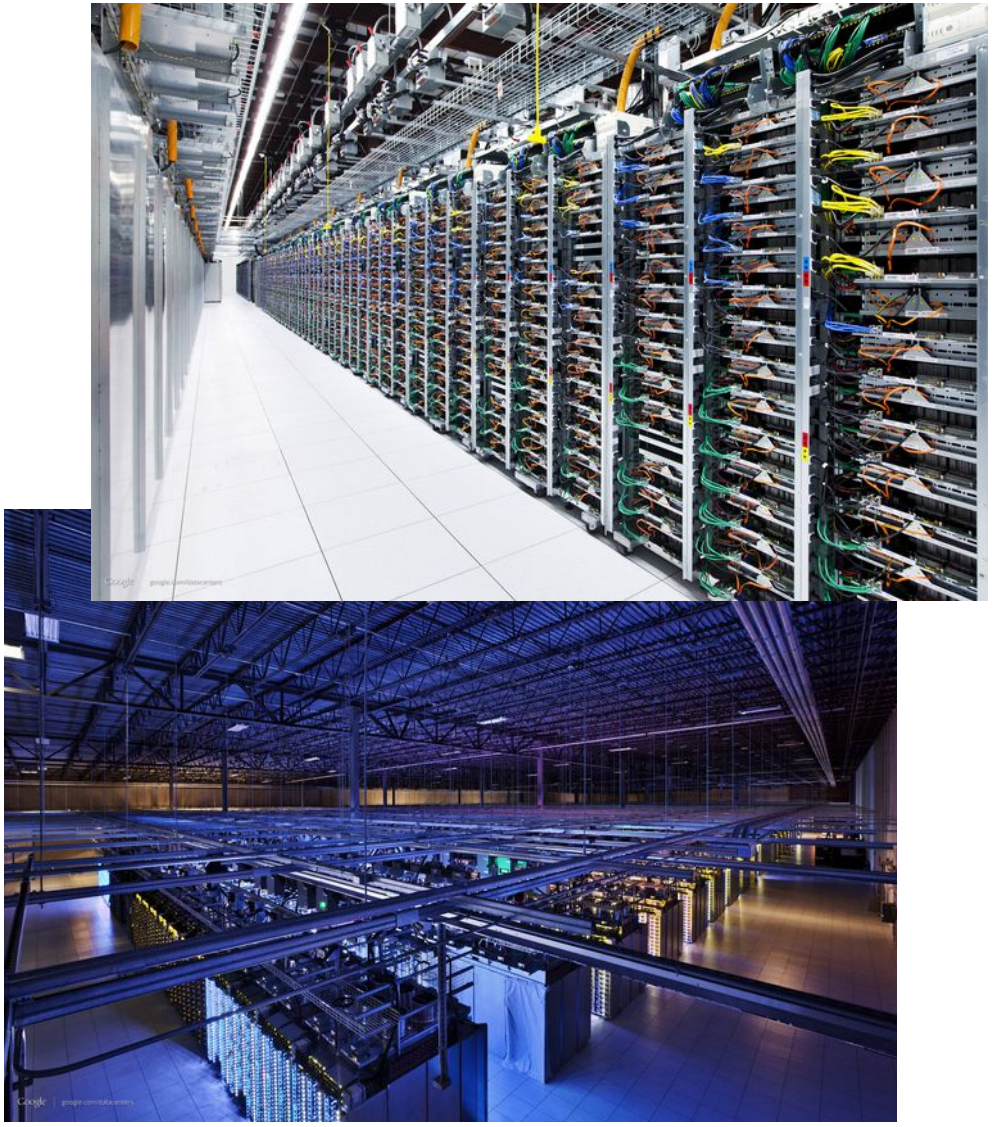
BigSys 2014 – System Software Support for Big Data
September 25-26 2014, Stuttgart, Germany

Talk Outline

- Motivation
- BiDAl
- Case Study
- Conclusions

Motivations

- Modern datacenters produce huge amounts of data in the form of event and error logs
- Understanding the logs is essential to identify problems or improve efficiency
  - Understand and exploit hidden patterns and correlations
  - First step towards self-managing, self-healing datacenters



Challenges

- Huge size of logs
  - A 2010 study [Thusoo et al, SIGMOD'10] reports that Facebook data centers produced 60TB of logging information daily
- Log analysis falls within the class of Big Data applications
  - Data sets are so large that conventional storage and analysis techniques are not appropriate to process them

BiDAl
Big Data Analyzer

- Java application (with GUI)
  - Proof-of-concept
- Typical workflow:
  - Instantiation of a storage backend
  - Data selection and aggregation
  - Data analysis

BiDAI
Big Data Analyzer

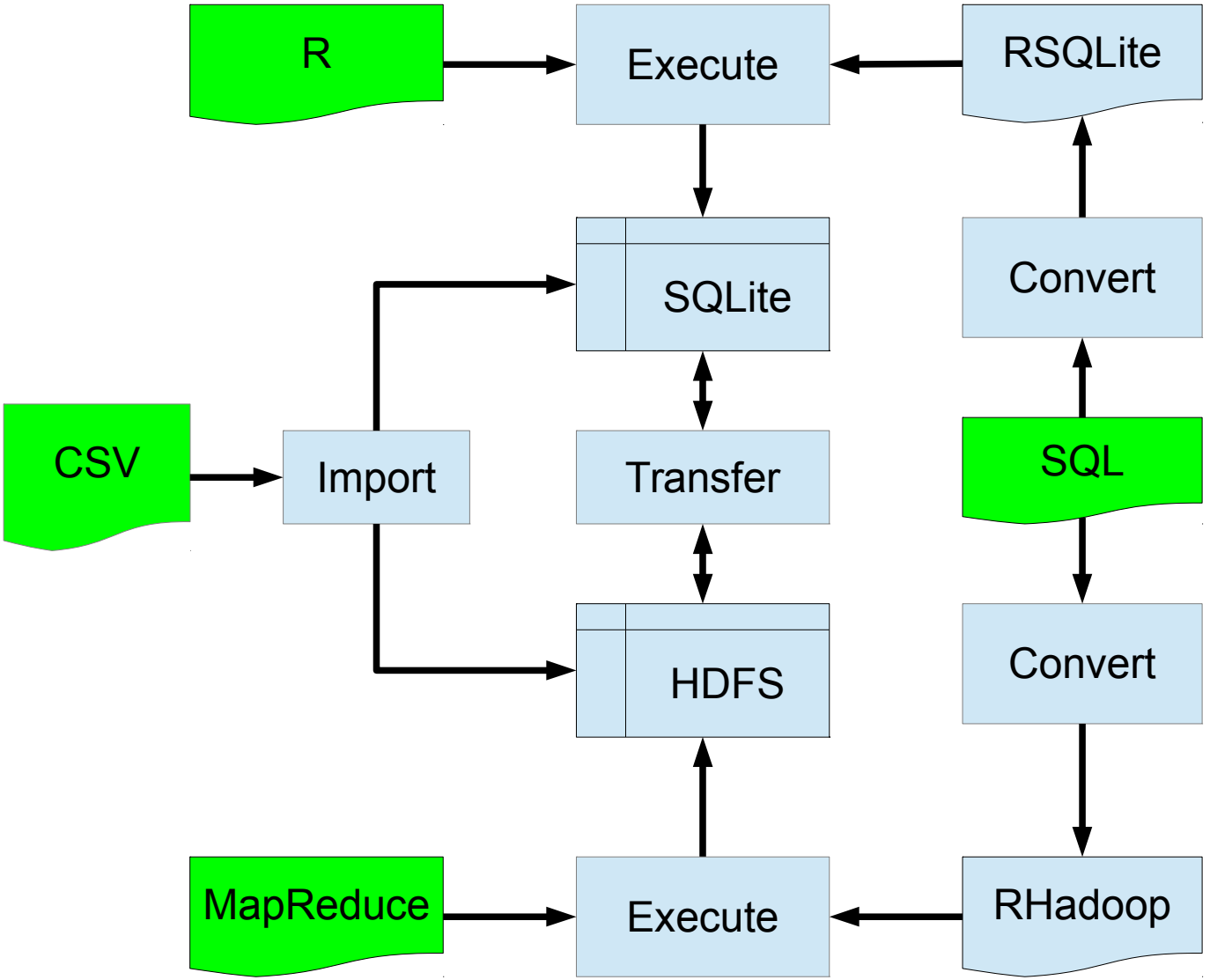
- Can import raw data in .CSV format
- Uses SQLite or Hadoop File System (HDFS) as storage backends
  - Additional storage types can be added
  - Although the current storage backends are based on the concept of “table”, other backends could be used too, e.g., HBase for <key, value> pairs
- Uses (a subset of) SQL as the query and data manipulation language
  - Translates SQL to the language understood by the storage backend – currently RSQLite or RHadoop

BiDAI

Big Data Analyzer

- Statistical computations can be performed using either R or Hadoop MapReduce
  - R commands are usually applied to the SQLite storage, while MapReduce commands are usually applied to the HDFS storage
  - BiDAI can transfer data automatically and transparently between the backends, to allow both languages to operate on both backends
- Computations can be concatenated
  - Usually, a data reduction is followed by the computation of some statistics

Data flow in BiDAI



Tables:

Machine_Events @ /home/me/Desktop/Storages/Google Storage @ SQLi
Machine_Downtime @ /home/me/Desktop/Storages/Google Storage @ S

Table preview:

A	B	C
0	5	0
0	6	0
0	7	0
0	10	0
0	13	0
0	14	0
0	19	0
0	21	0
0	23	0
0	25	0

Commands:

export
r utils

spline @ /home/me/D
exponential_distribut
aggregate @ /home/m
filter @ /home/me/Des
get column @ /home/m
lognormal_distributio
polynomial regressio
spline_ecdf @ /home/m
ecdf @ /home/me/Des
histogram @ /home/m
spline function expor
spline function expor

Selected commands:

aggregate @ /home/me/
filter @ /home/me/Deskt
ecdf @ /home/me/Deskt

v

^

Parameters:

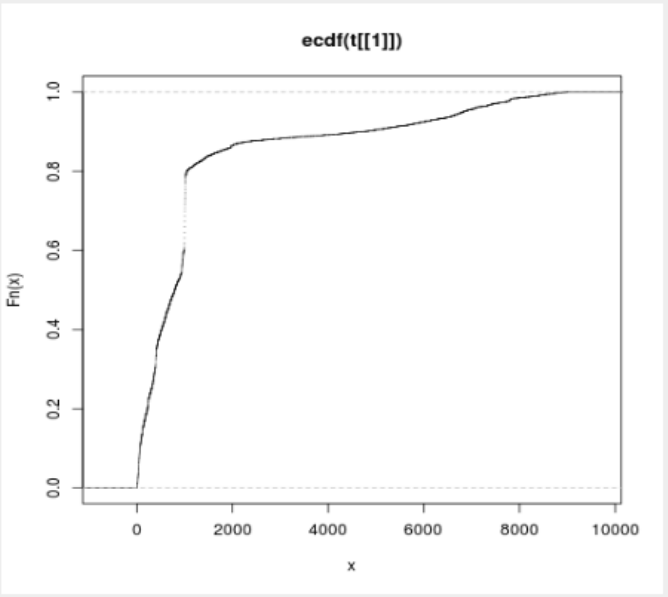
1

Temp result:

A
336
1011
136
694
5462
4957
7553
153
1032
397

Scripts:

spline_task_cpu
task_duration_kill
testhadoop
task_duration_end
spline_tasks_per_job
spline_task_prio
job_arrival
machine_down_arrival
machine_events_ram
spline_job_arrival
tasks_per_job
spline_task_ram
machine_downtime
machine_events_cpu



Script name:

Load

Save

Delete

Current Table:

Temp result

Cancel

Current parameter:

Current command:

```
png(filename="temp.png");
plot(ecdf(t[[1]]));
dev.off();
```

Exec

Case Study: Google Traces

- The development of BiDAl was initially motivated by the need to analyze Google traces
  - https://code.google.com/p/googleclusterdata/
- Goal:
  - Extract workload parameter
  - Instantiate a simulation model of the Google cluster
  - Validate the simulation with respect to the observed data

Google traces

- Contain 29 days of information from May 2011, on a cluster of about 11k machines
  - Machine event (e.g., new machine is added to the pool, ...)
  - Machine attribute (e.g., OS is updated to a newer version, ...)
  - Jobs and Tasks (requirements, submit/completion time...)
  - Resource usage (sampled at some fixed intervals)
- Total size of the compressed trace is ~40GB
- https://code.google.com/p/googleclusterdata/

Entities of the Simulation Model

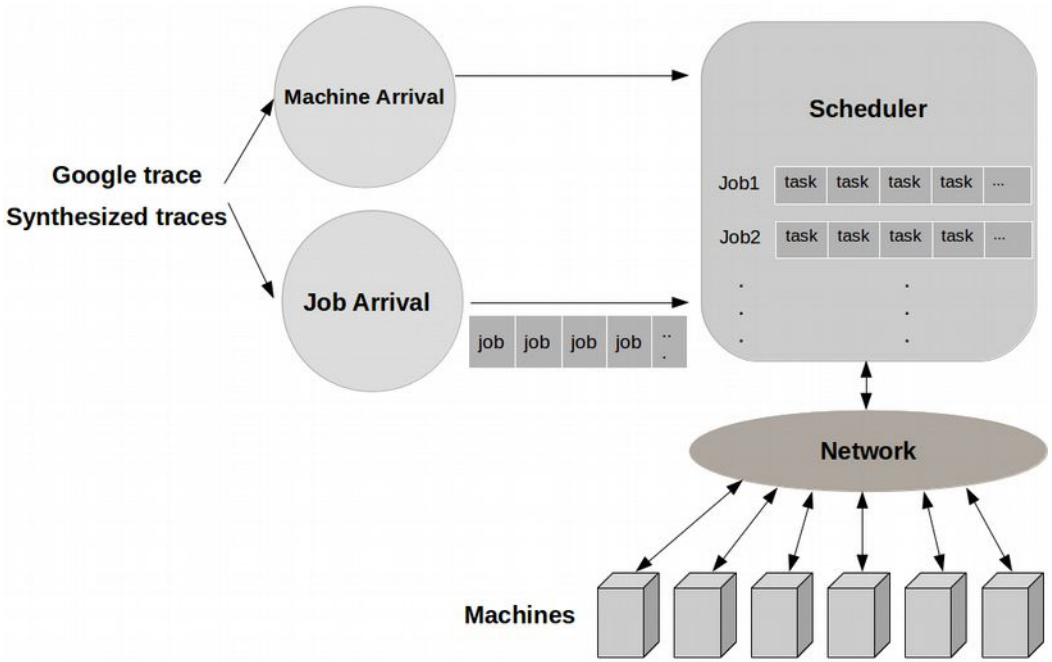
- Tasks and Jobs
- Arrival

  Process that generates new events (new job, new machine, machine removal...)
- Scheduler

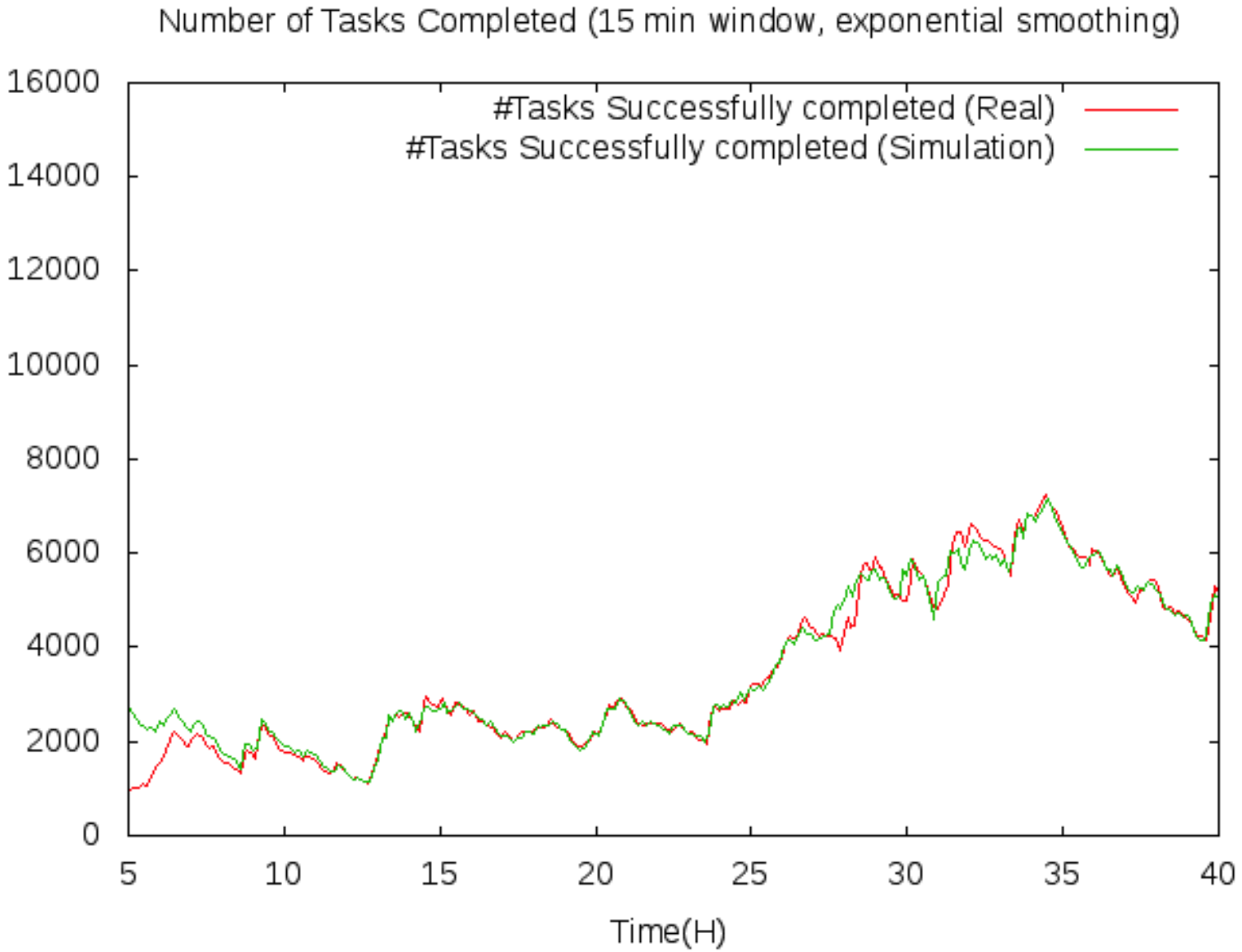
  Decides where the tasks of a job can be executed
- Machines

  Execute tasks; notify the scheduler when a task terminates; send status updates to the scheduler
- Network

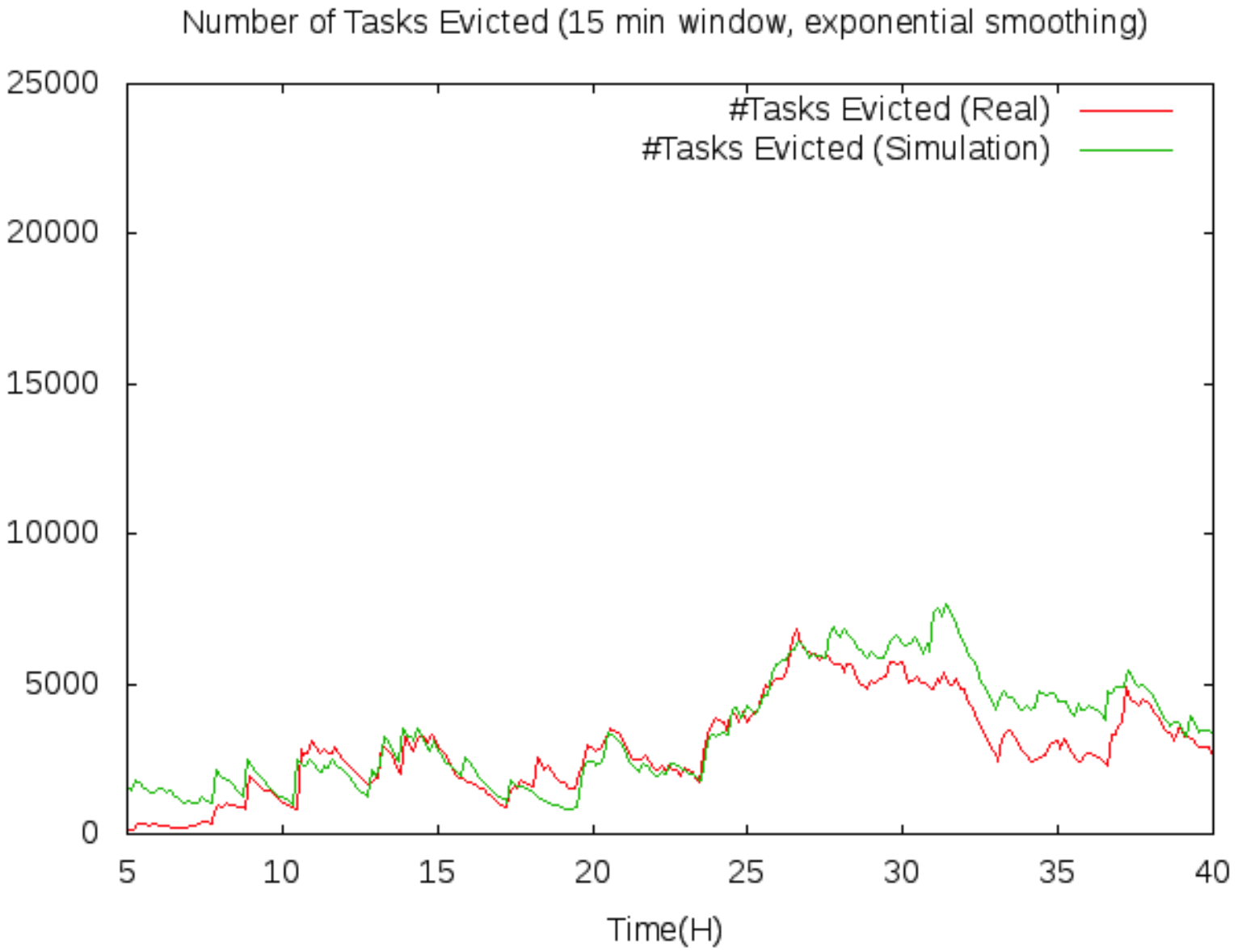
  Allows other entities to communicate



Trace-Driven Simulation Results



Trace-Driven Simulation Results

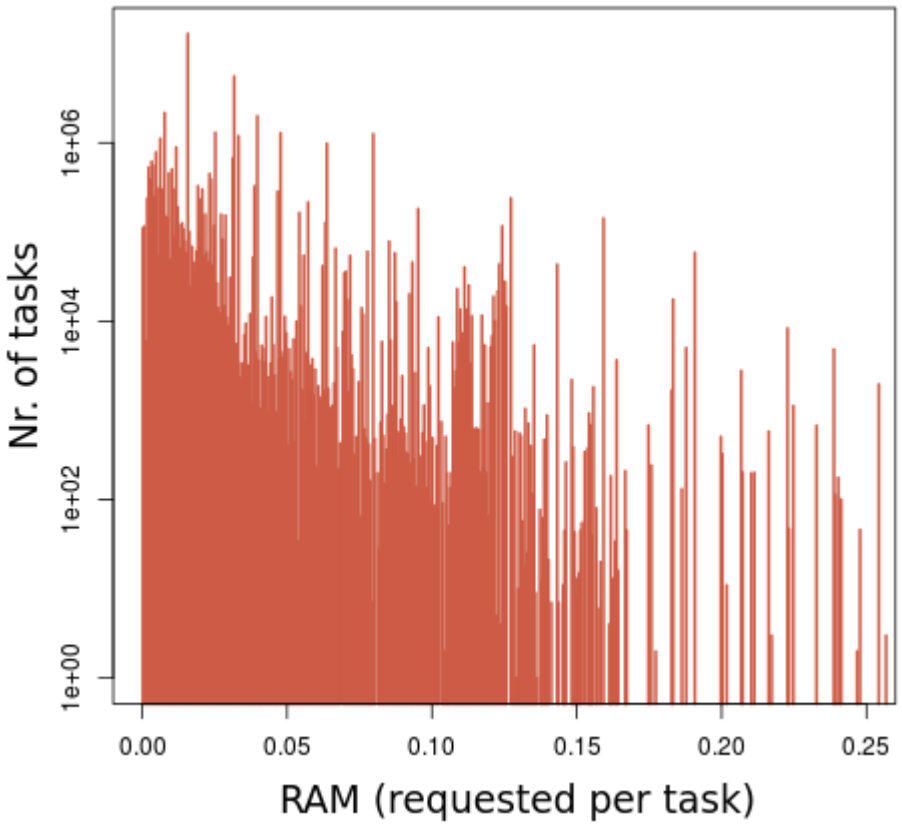


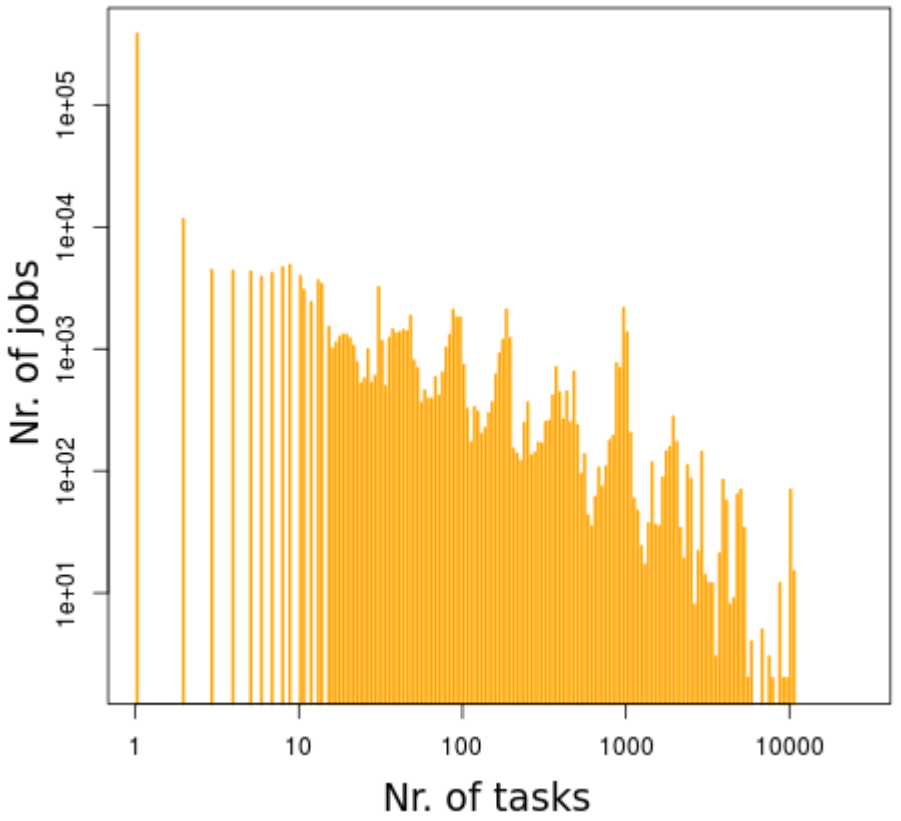
Workload Characterization

- We used BiDAl to extract workload parameters from the traces
  - Jobs Inter-arrival time distribution
  - Number of tasks per job
  - Distribution of execution times of different types of jobs (e.g., jobs that terminate successfully, jobs that are aborted by the user, …)
  - ...

Examples

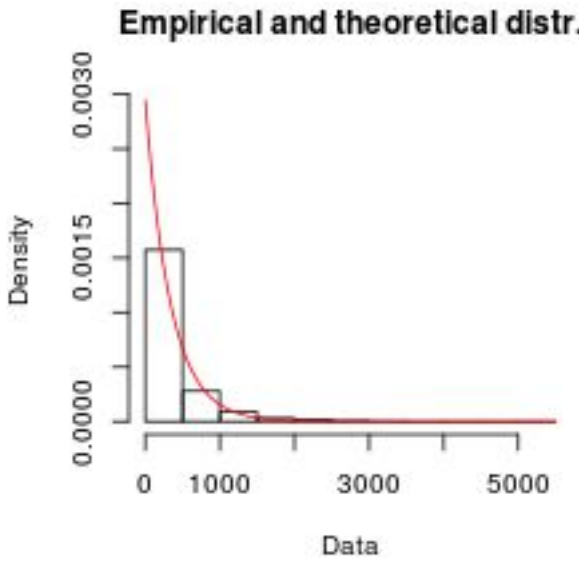
Frequencies of the amount of RAM used by tasks (left) and number of tasks per job (right)

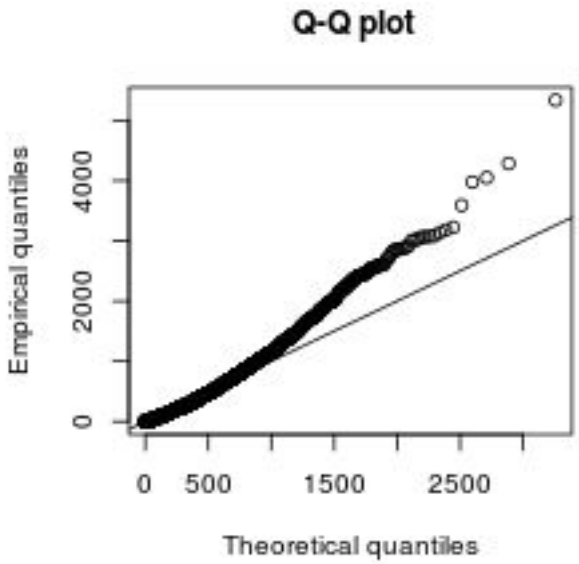


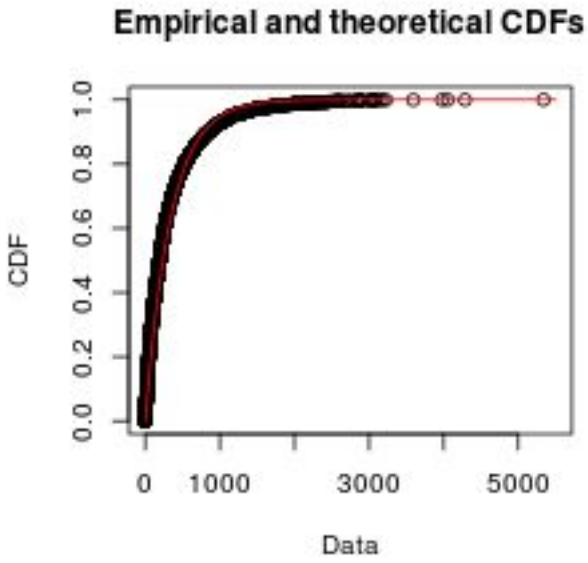


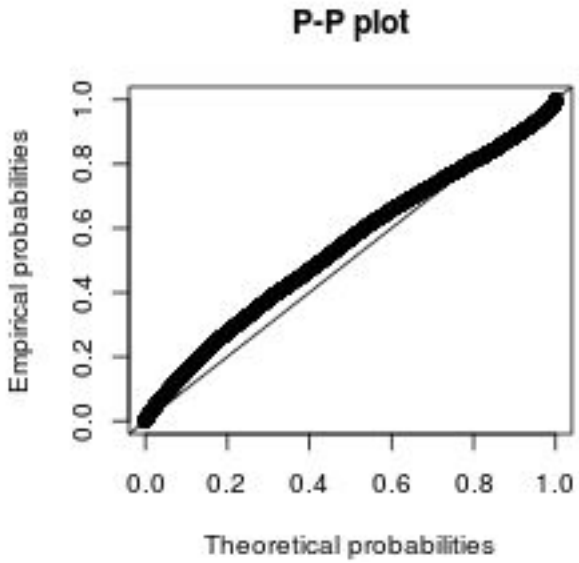
Examples

- Machine update events
- Left
  - Density and CDF with lines representing exponential fitting
- Right
  - Goodness of fit in Q-Q and P-P plots (straight lines denote perfect fit)



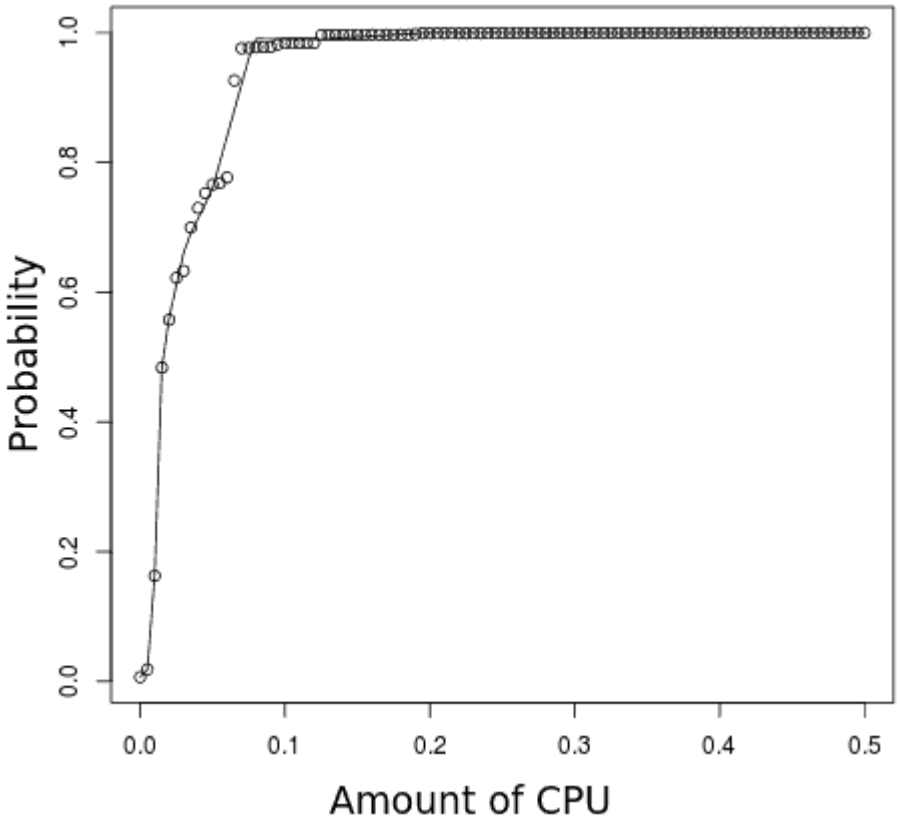


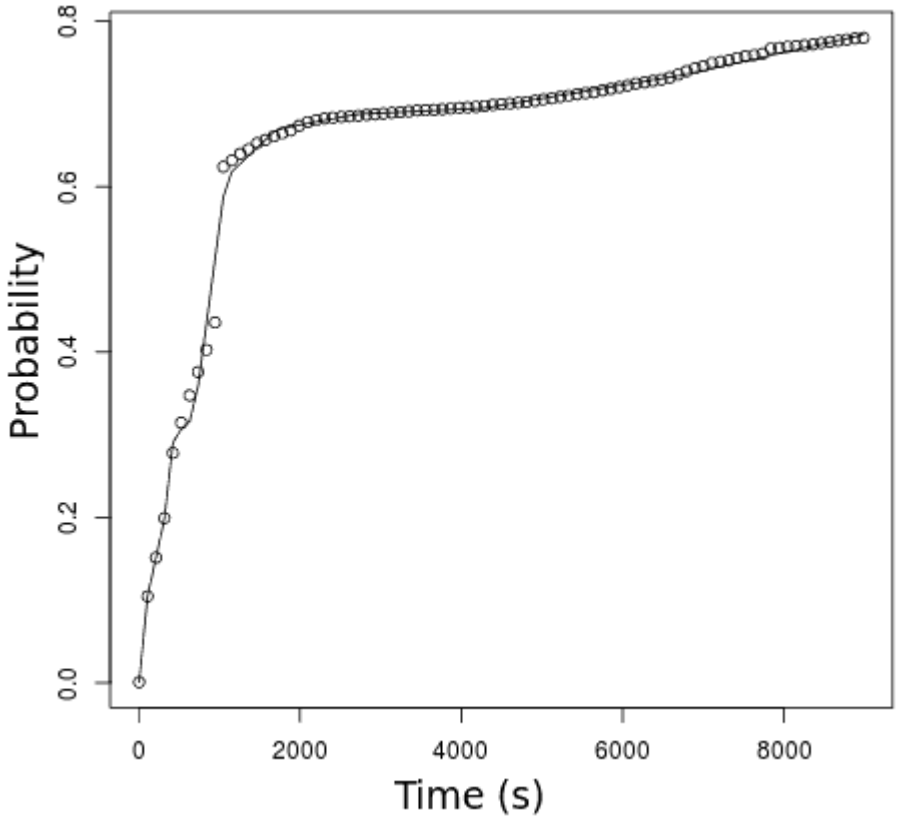




Examples

CDFs fitted by a sequence of splines: CPU task requirements (left) and machine downtime (right)





Results using synthetic traces

	Real	Simulated	Rel. dif.
Running	124217	136037	0.09
Ready	5987	5726	0.04
Completed	3277	2317	0.29
Evicted	1057	2165	1.04

Can be explained by the high variance of real data

Conclusions and Future Works

- Big Data Analyzer (BiDAl) is a prototype data analysis tool that can handle large datasets
  - SQL, R, Hadoop/MapReduce
  - Extensible
- We used BiDAl to analyze the Google traces dataset
- Future works
  - Support additional storage backends
  - Include additional analysis algorithms (e.g., predictive algorithms, machine learning)
  - Live log analysis

Thanks for your attention!

http://www.cs.unibo.it/~sirbu/bidal.zip