The Importance of Complete Data Sets for Job Scheduling Simulations

Dalibor Klusáček, Hana Rudová

Faculty of Informatics,
Masaryk University,
Brno, Czech Republic

{xklusac, hanka}@fi.muni.cz

15th Workshop on Job Scheduling Strategies for Parallel Processing
Atlanta, GA
23 April 2010

Introduction

- Both production or experimental scheduling algorithms have to be heavily tested
- Usually, through a simulation using synthetic or real-life workloads as an input
- Popular real-life based workloads
  - **Parallel Workloads Archive (PWA)**
    - Data usually coming from 1 cluster
  - **Grid Workloads Archive (GWA)**
    - Data coming from several clusters that constitute the Grid

PWA and GWA workloads

- Both provide variety of different workloads
- Job description typically contains
  - job_id, submission time, start time, completion time, # of requested CPUs, runtime estimate, ...
- GWF (GWA) extends SWF (PWA) format with "Grid features", e.g.:
  - ID of the cluster (site) where the job comes from
  - ID of the cluster (site) where the job was executed
  - Additional job requirements (OS, OS-version, CPU-arch, site restriction, ...)

What do we miss in GWA?

- Resource description
  - Missing (Grid'5000)
  - Incomplete (e.g., Sharcnet, NorduGrid, DAS-2)
- Changing state of the system (the dynamics)
  - Installation time of each cluster
  - Machine failures
  - Dedicated machines, background load
- Additional constraints (specific job requirements)
  - Fields are empty in the GWF files
  - Corresponding parameters of the machines are not known

Specific job requirements

- **In real life, not every cluster can execute every job**
  - Long jobs (runtime > 24h) have dedicated clusters
    - Long jobs can not run where short jobs run
  - Scientific applications need software licenses
    - Job needs Gaussian – cluster must support Gaussian
  - Job needs fast network interface – cluster must support e.g. Infiniband
  - Only some users (group) can use given cluster
  - Suspicious users want to use only "known clusters"
- All these requests and constraints can be combined together
- User/Admin may prevent jobs from running on some cluster(s).

Are these features important?

- **Intuition**:
  - Failures and restarts require appropriate reactions of the scheduler (job is killed, job restarts, job can start earlier, …)
  - Cluster installations, failures and restarts or background load change the amount of available computing power, thus the load of the system
  - Specific job requirements limit the choices that the scheduler has when allocating jobs to clusters
  - Specific job requirements can locally increase machine usage or even cause local overload
- **Experimental evaluation needs truly complete data set**

Complete data set from MetaCentrum

- MetaCentrum is the Czech national Grid infrastructure
- We were able to collect complete data set
  - **Jobs** – 103,656 jobs from January – May 2009
    - No ignored background load
    - **Specific job requirements included**
  - **Machines** – 14 clusters (806 CPUs)
    - Detailed description of each cluster including specific properties
  - **Failures and restarts**
    - Time periods when machines were available or not
  - **Queues** – priorities and time limits (long, normal, short, …)

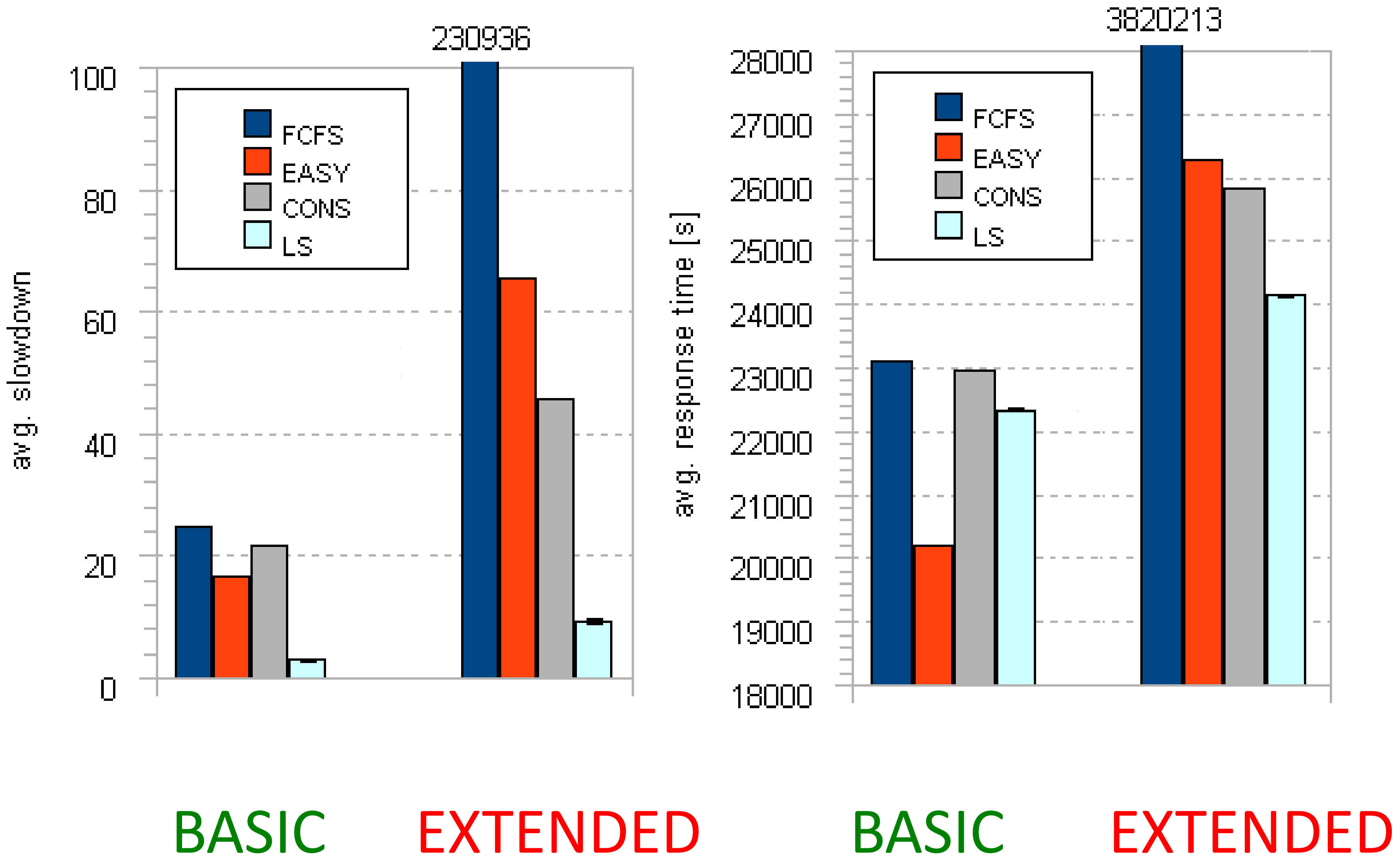
Experiments using MetaCentrum data set

- **Question**: Do the additional information and constraints such as machine failures or specific job requirements influence the quality of the solution?
- **BASIC problem**:
  - **No machine failures**
  - **No specific job requirements**
  - Similar to the typical amount of information available in GWA
- **EXTENDED problem**:
  - **Includes both machine failures and specific job requirements**

Scheduling algorithms

- **FCFS**, EASY backfilling (**EASY**), Conservative backfiling (**CONS**)
- Local Search (**LS**) based optimization of CONS
  - Periodical optimization of the schedule of reservations
  - Randomly moves existing reservations
  - Accepts move if the parameters of the new schedule are better
    - Detailed description is in the paper
- **Criteria**: slowdown, response time, wait time, number of killed jobs

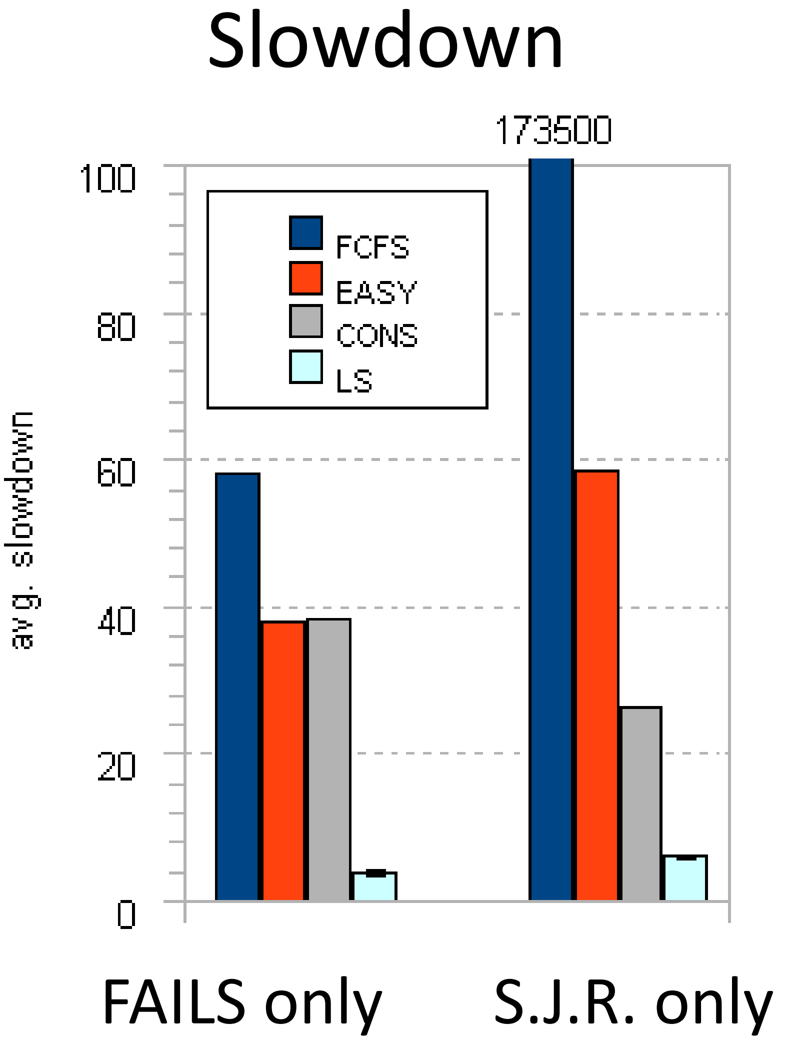
MetaCentrum: BASIC vs. EXTENDED

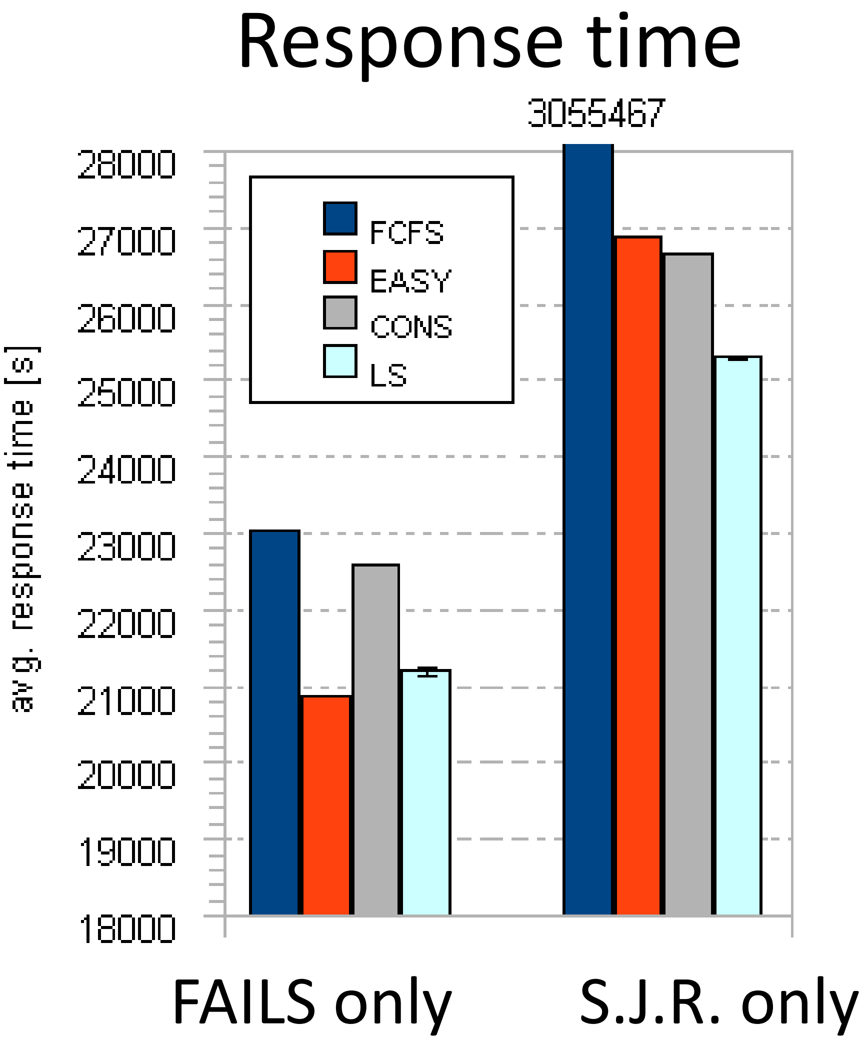


Slowdown

Response time

MetaCentrum: Failures vs. Specif. job. req.





- Machine failures has usually smaller effect than specific job requirements
- It is easier to deal with machine failures than with specific job requirements when the overall system utilization is not extreme (43% here).

Summary

- In MetaCentrum, complete and "rich" data set influences the quality of the generated solution (**EXTENDED problem**)
- **BASIC problem** ignores important real-life features so the results are less interesting

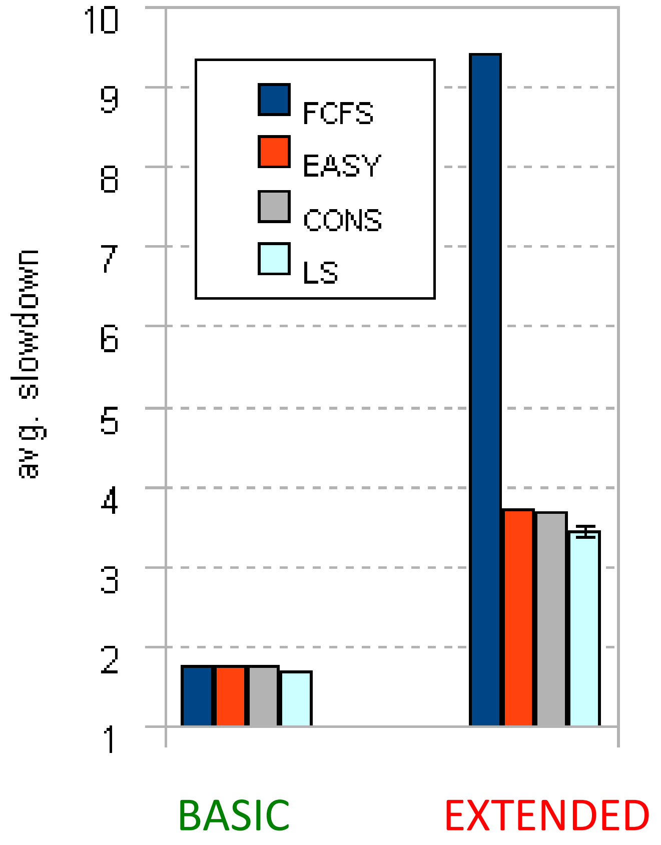
- **Question**: Are similar observations possible also for the existing GWA workloads?
  - PWA workloads cover mostly homogeneous clusters (specific job requirements are less probable here)

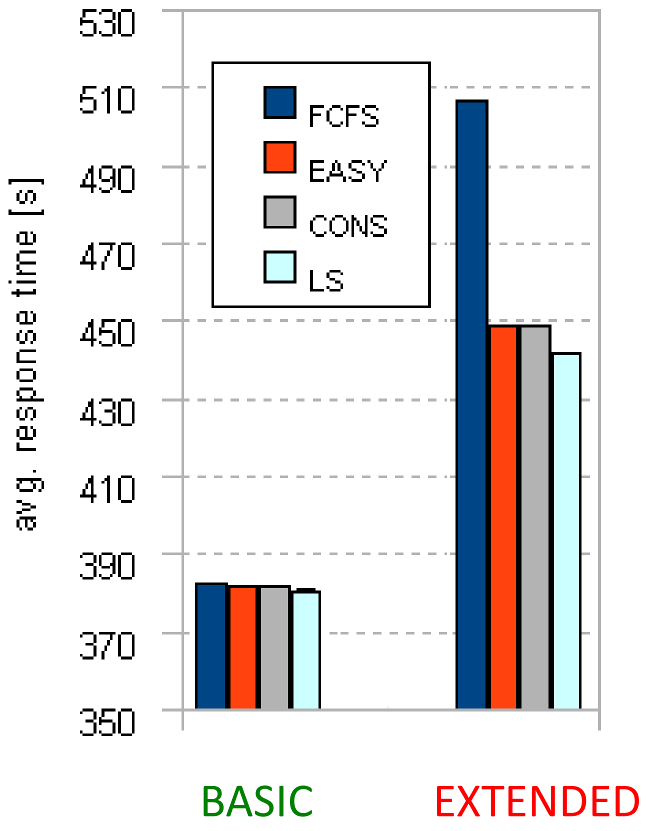
Extending the GWA

- We have extended DAS-2 and Grid'5000 workloads
- Failures
    - DAS-2: synthetic failures using model of Zhang et al. (JSSPP'04)
    - Grid'5000: using known data from Failure Trace Archive
- Specific job requirements
    - Synthetically generated by the analysis of the original workload
    - Each job has an "application code" → ID of the binary/script
    - More jobs can have the same application code
    - Cluster(s) used to execute jobs with the same application code were taken as "required" simulating specific job requirements

DAS-2: BASIC vs. EXTENDED

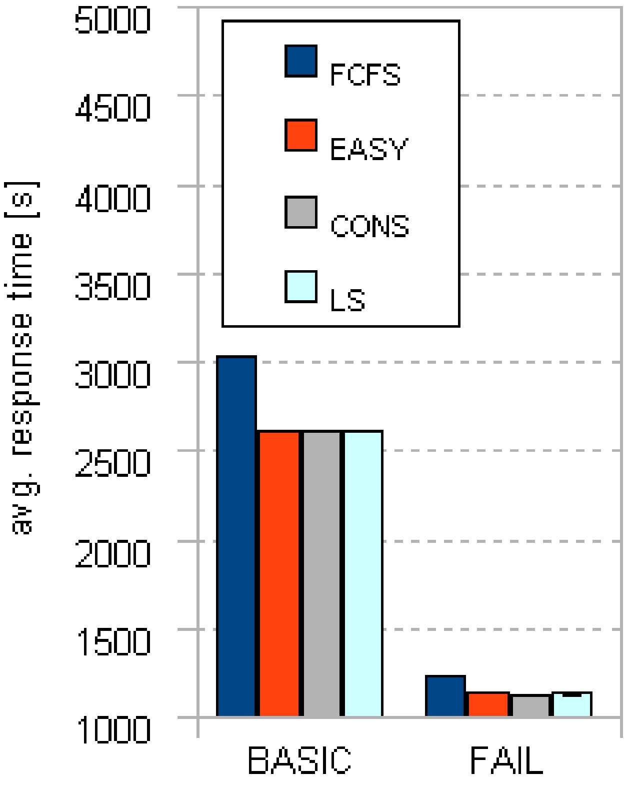
- DAS-2 has a very low utilization (10%)
  - Differences between algorithms are small
- Otherwise similar to MetaCentrum
  - EXTENDED problem is "harder" than BASIC, machine failures less demanding than sp.j.req.





Grid'5000: BASIC vs. EXTENDED

- Exhibits different behavior than MetaCentrum or DAS-2
- **Response time is always much lower when failures are used** (which is weird at the first sight)



- Why? – high frequency of machine failures
  - # Failures per machine per month = 12.6
- **Frequent failures kill especially long jobs**
  - Killed jobs had average duration of 17 hours
  - Average duration of all jobs was just 43.5 minutes
- **Such behavior influences especially the response time**

Pros and Cons of Complete Data Sets

- **Pros**
  - Otherwise "easy" data sets may become demanding
  - Algorithms are no more "equal" wrt. performance
  - Optimization techniques start to make sense
  - More realistic scenarios (users' reqs., system dynamics)
- **Cons**
  - Collecting and publishing such data is very complicated
  - Raw data often contain many errors, duplicates (e.g. mach. failures)
  - Popular objective functions can be misleading (resp. time)
  - Simulation results have to be carefully interpreted
  - It is harder to identify problems and understand algorithms' behavior

Conclusion

- Complete and "rich" data sets may significantly influence algorithms' performance
- Especially "**specific job requirements**" are interesting
- If possible, complete data sets should be collected and used to evaluate algorithms under harder conditions
- May narrow the gap between "ideal world" and "real-life experience"
- Our workload is freely available for further open research:

  http://www.fi.muni.cz/~xklusac/workload
- I am looking forward to answer your questions at Skype:
  user name = **dalibor.klusacek**