Managing Risk of Inaccurate Runtime Estimates for Deadline Constrained Job Admission Control in Clusters

Chee Shin Yeo and Rajkumar Buyya

Grid Computing and **D**istributed **S**ystems (GRIDS) Lab.
Dept. of Computer Science and Software Engineering
The University of Melbourne, Australia
http://www.gridbus.org

Problem/Motivation: Computing as a Service

- User-specific Service Level Agreement (SLA)
  - Deadline QoS
- Deadline constrained job admission control in a cluster
  - Prevents workload overload, service degradation
  - Dependent on accurate runtime estimates
  - Focus: Managing inaccurate runtime estimates

Related Work

- Cluster Resource Management System (RMS)
  - Condor, LoadLeveler, LSF, OpenPBS, Sun Grid Engine
- Job admission control
  - [Irwin04][Popovici05]: Utility
  - [Islam04]: Soft Deadline
- Managing risk in computing jobs
  - [Kleban04]: Job delay
  - [Irwin04][Popovici05]: Penalty for job delay
- Job Scheduling with inaccurate runtime estimates
  - [Mu'alem01][Sabin04][Tsafrir05]

Deadline Constrained Job Admission Control in a Cluster

- Cluster RMS
  - Single interface for job submission
  - Non-preemptive job scheduling
- Job submission
  - No change in SLA after acceptance
  - User-defined parameters
    - Deadline QoS (Hard)
    - Runtime estimate
    - Number of processors

Libra: Deadline Constrained Job Admission Control in a Cluster

- Deadline-based Proportional Processor Share of a job *i* on node *j* (time-shared) [Sherwani04]

$$share_{ij} = \frac{remaining\_runtime_{ij}}{remaining\_deadline_i}$$

- Total share for *n* jobs on a node *j*

$$total\_share_j = \sum_{i=1}^{n_j} share_{ij}$$

- Suitable node if deadline of all jobs (with new job) met
- BEST FIT strategy (least available processor time after accepting new job)

LibraRisk: Modeling Risk of Deadline Delay

- Delay of job *i*

$$delay_i = (finish\_time_i - submit\_time_i) - deadline_i$$

- Deadline delay of job *i* [Kleban04]

$$deadline\_delay_i = \frac{delay_i + remaining\_deadline_i}{remaining\_deadline_i}$$

- Mean deadline delay of *n* jobs on node *j*

$$\mu_j = \frac{\sum_{i=1}^{n_j} deadline\_delay_{ij}}{n_j}$$

- Risk of deadline delay of *n* jobs on node *j*

$$\sigma_j = \sqrt{\frac{\sum_{i=1}^{n_j} (deadline\_delay_{ij})^2}{n_j} - (\mu_j)^2}$$

LibraRisk: Managing Risk of Deadline Delay

- Libra: Deadline-based Proportional Processor Share
- Different Admission Control
  - Determine delay of all jobs (previously accepted jobs and new job) on each node if new job accepted
  - Compute risk of deadline delay for each node
  - Suitable node if zero risk
  - Accept new job if sufficient number of suitable nodes as required by new job

Performance Evaluation: Simulation

- GridSim toolkit: Simulated scheduling in a cluster computing environment
  (http://www.gridbus.org/gridsim)
- Feitelson's Parallel Workload Archive
  (http://www.cs.huji.ac.il/labs/parallel/workload)
  - Last 3000 jobs in SDSC SP2 trace
  - Average inter arrival time: 2131 s (35.52 mins)
  - Average run time: 8880 s (2.47 hrs)
  - Average number of requested processors: 17
- SDSC SP2
  - Number of computation nodes: 128

Experimental Methodology: Performance Evaluation

- Modeling deadline QoS [Irwin04]
- High urgency jobs (Default is 20%)
  - Low deadline/runtime (Default mean is 4)
  - Values normally distributed in each deadline/runtime
  - Randomly distributed in arrival sequence
- Deadline high:low ratio (Default is 4)
  - Ratio of means for deadline/runtime of low and high urgency jobs

Experimental Methodology: Performance Evaluation

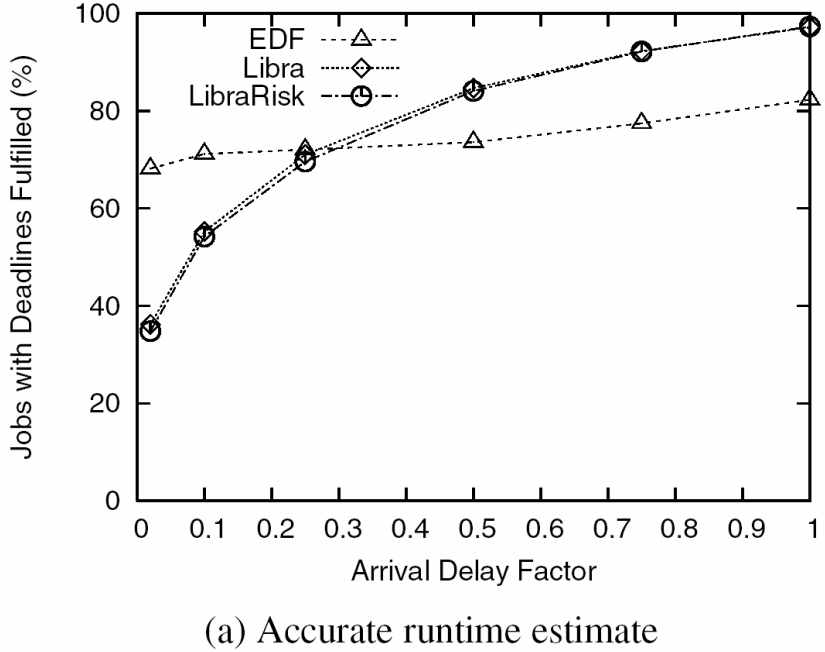
- Earliest Deadline First (EDF)
  - Space-shared
  - Reselect a new job with an earlier deadline that arrives later
  - Reject job prior to execution, not submission
- Libra
  - Time-shared (Deadline-based proportional processor share)
  - BEST FIT strategy (least available processor time after accepting new job)

Experimental Methodology: Performance Evaluation

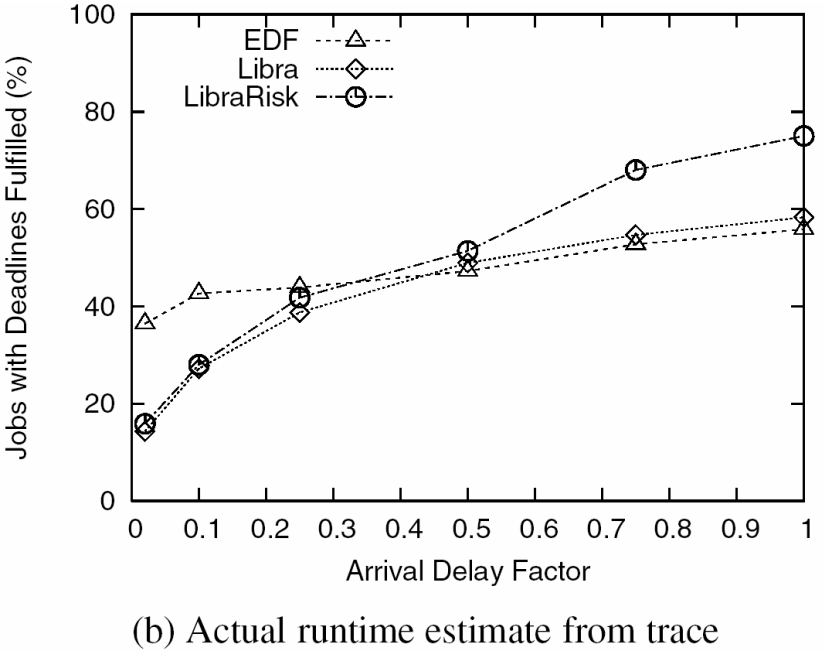
- Arrival delay factor (Default is 1 – from trace)
  - Model cluster workload thru job inter arrival time
- Inaccuracy of runtime estimates
  - 0% - accurate runtime estimate (runtime)
  - 100% - actual runtime estimate from trace
- Evaluation metrics
  - % of jobs with deadlines fulfilled
  - Average slowdown (jobs with deadlines fulfilled)

Impact of Varying Workload

Jobs with Deadlines Fulfilled (%)



(a) Accurate runtime estimate

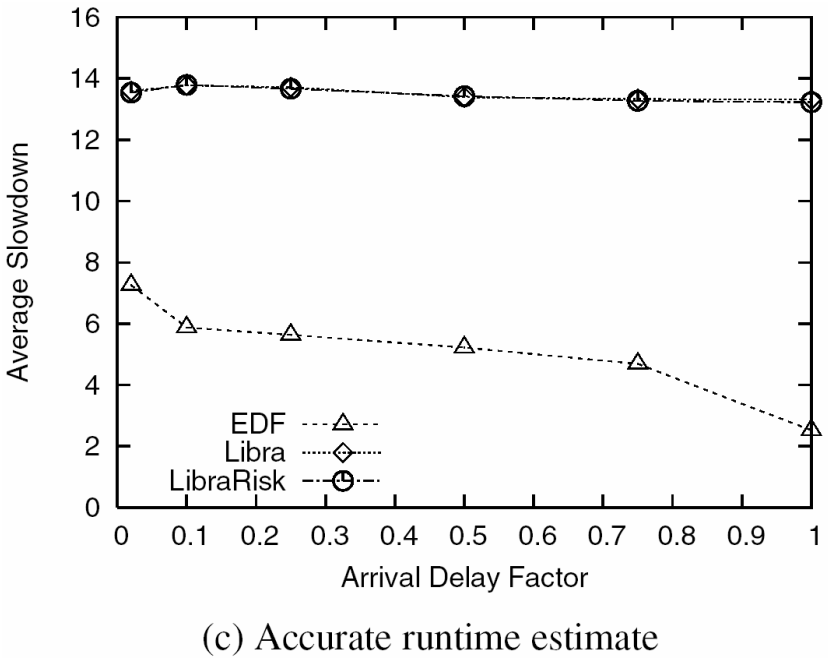


(b) Actual runtime estimate from trace

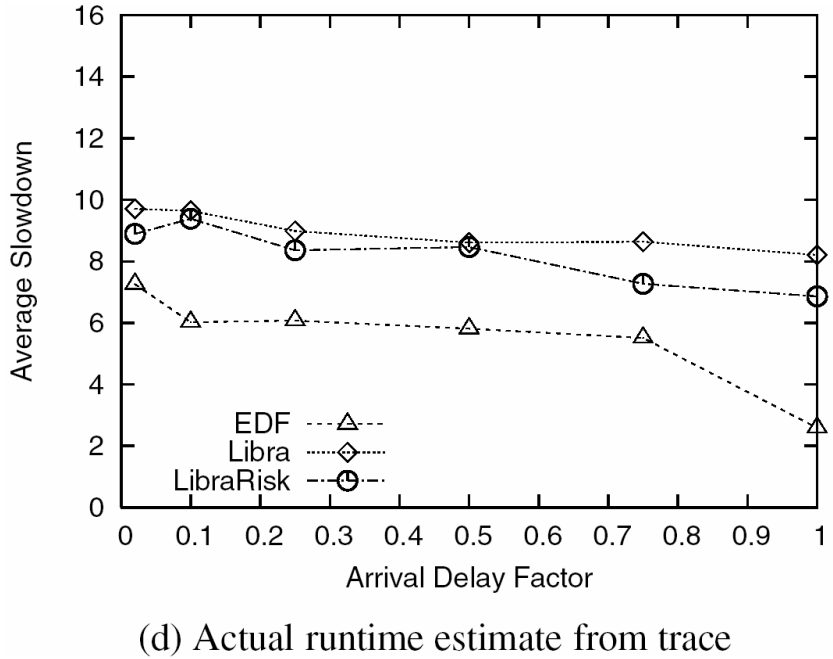
- Less jobs fulfilled for actual runtime estimate from trace
- More jobs fulfilled with higher arrival delay
- LibraRisk: More jobs fulfilled (higher arrival delay)

Impact of Varying Workload

Average Slowdown



(c) Accurate runtime estimate

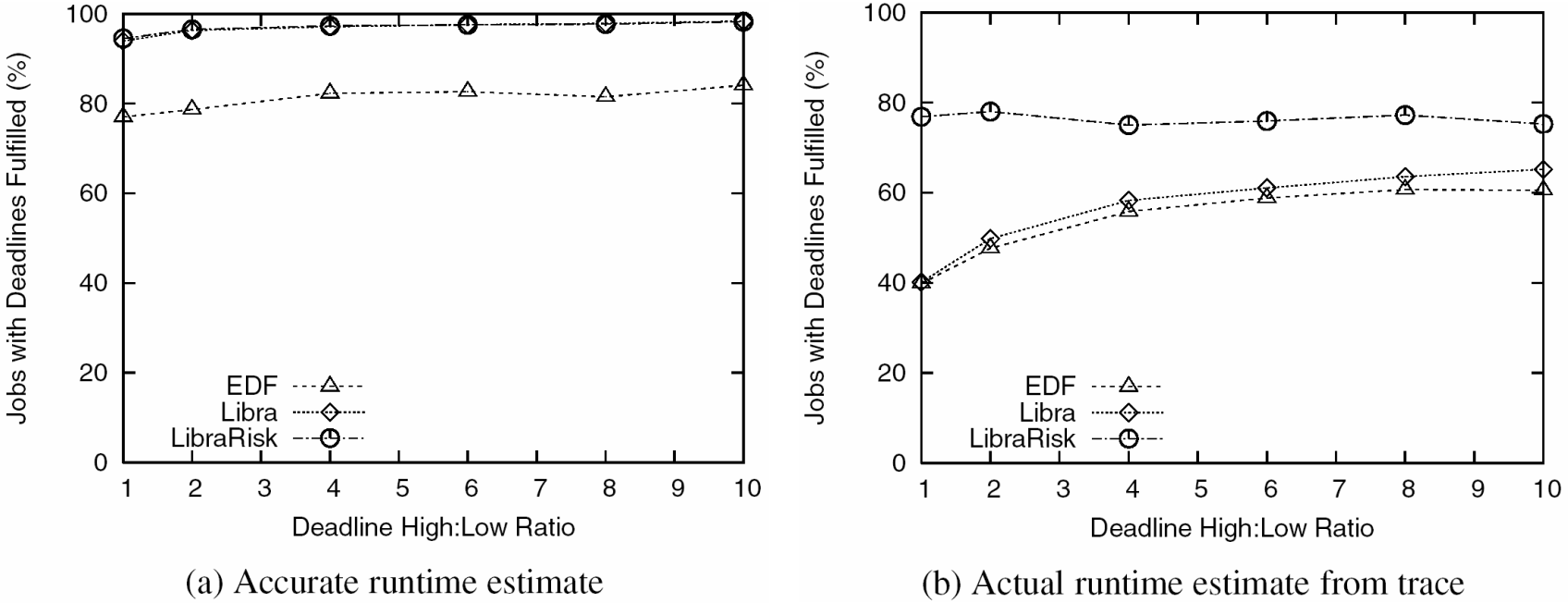


(d) Actual runtime estimate from trace

- Lower slowdown for actual runtime estimate from trace
- Lower slowdown with higher arrival delay
- LibraRisk: Lower slowdown than Libra

Impact of Varying Deadline High:Low Ratio

Jobs with Deadlines Fulfilled (%)



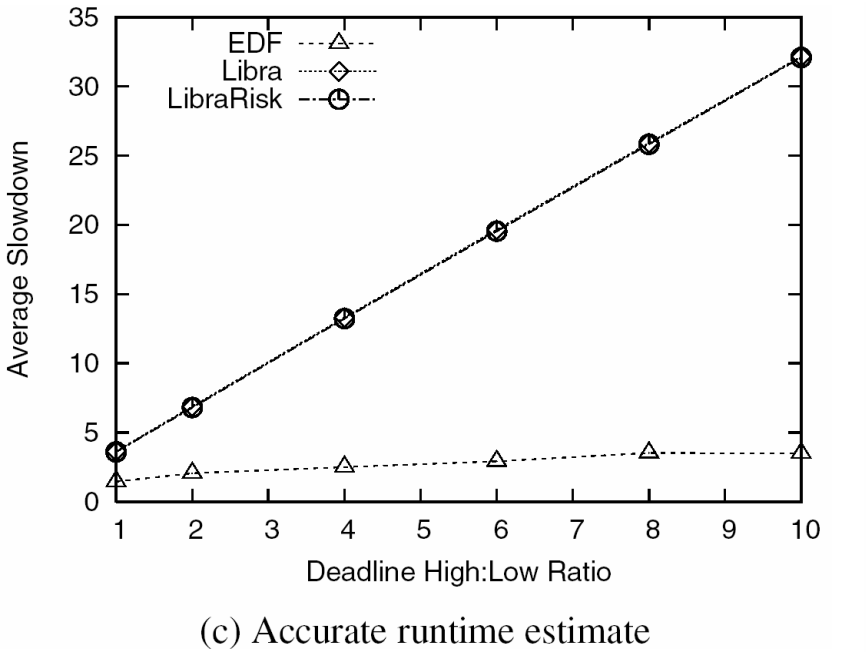
(a) Accurate runtime estimate

(b) Actual runtime estimate from trace

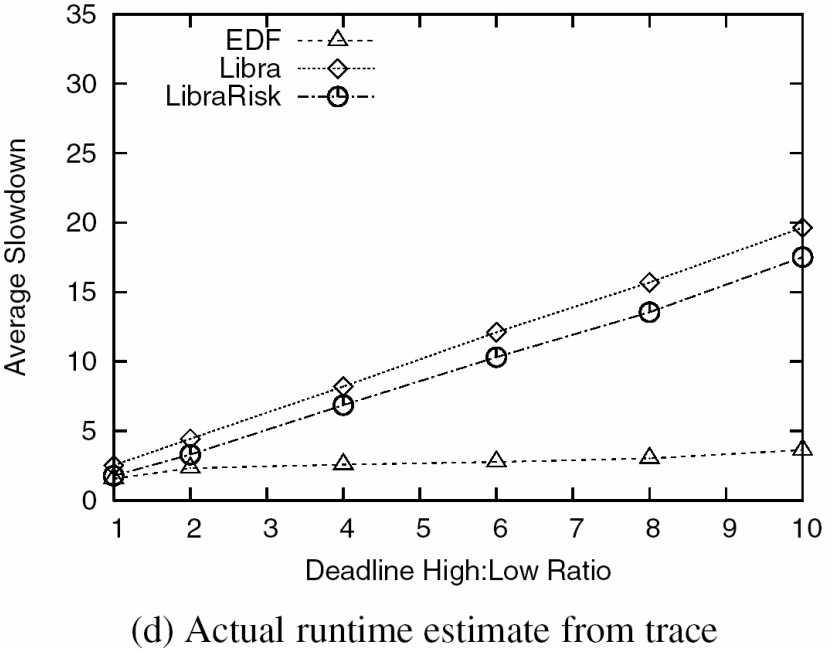
- More jobs fulfilled with higher deadline ratio
- LibraRisk: More jobs fulfilled (lower deadline ratio)

Impact of Varying Deadline High:Low Ratio

Average Slowdown



(c) Accurate runtime estimate

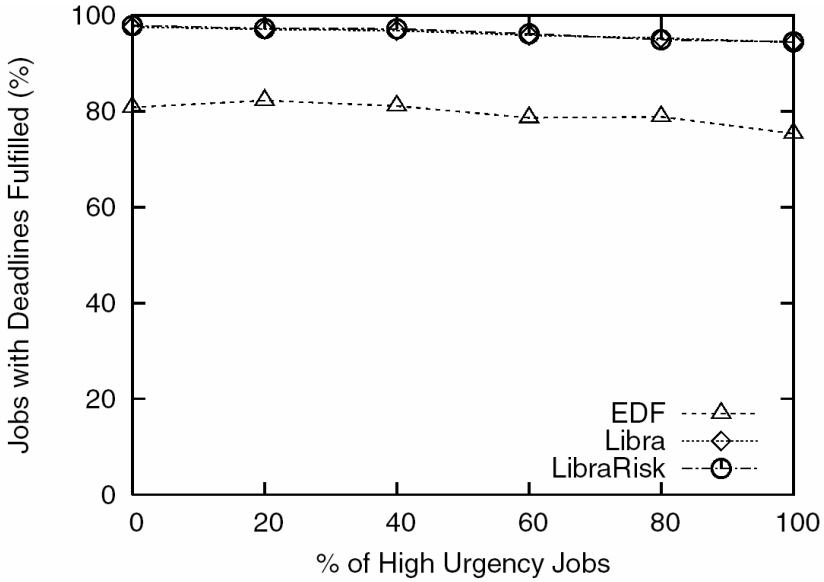


(d) Actual runtime estimate from trace

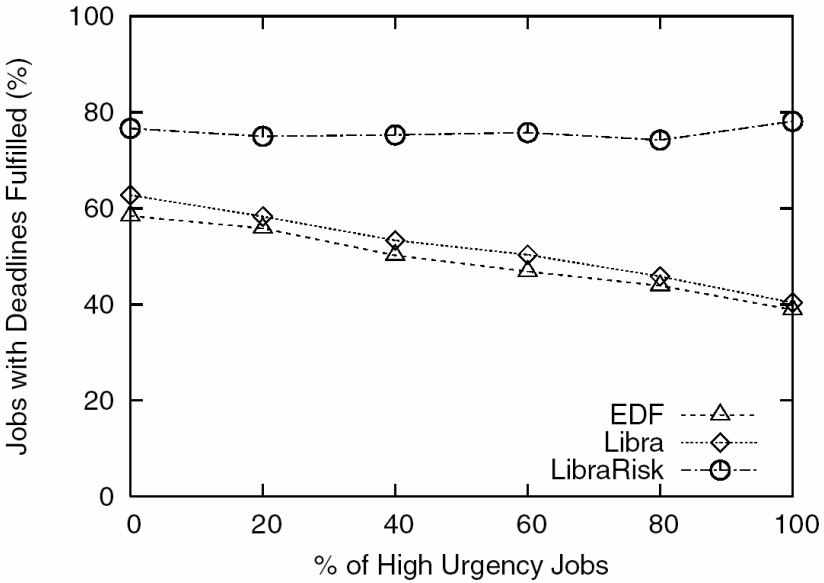
- Higher slowdown with higher deadline ratio
- LibraRisk: Lower slowdown than Libra

Impact of Varying High Urgency Jobs

Jobs with Deadlines Fulfilled (%)



(a) Accurate runtime estimate

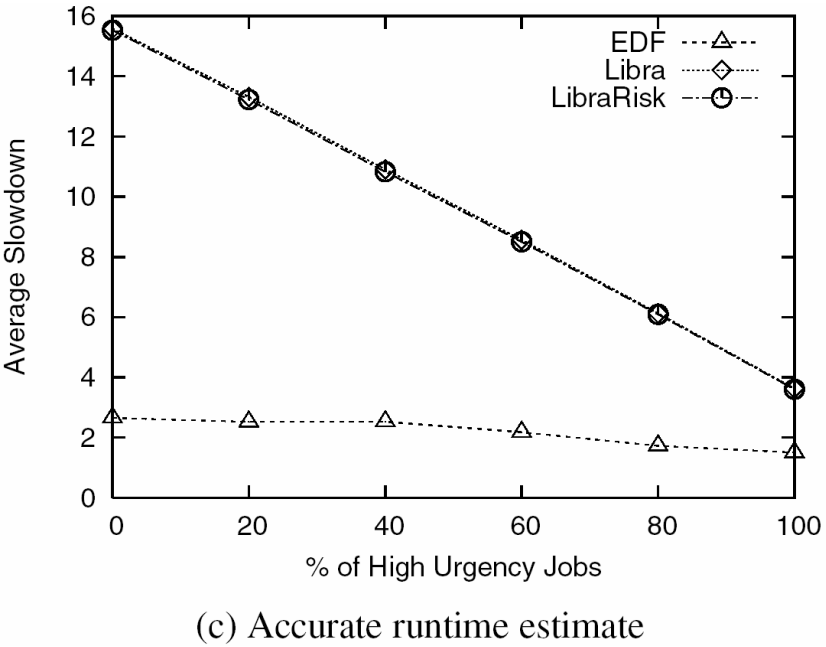


(b) Actual runtime estimate from trace

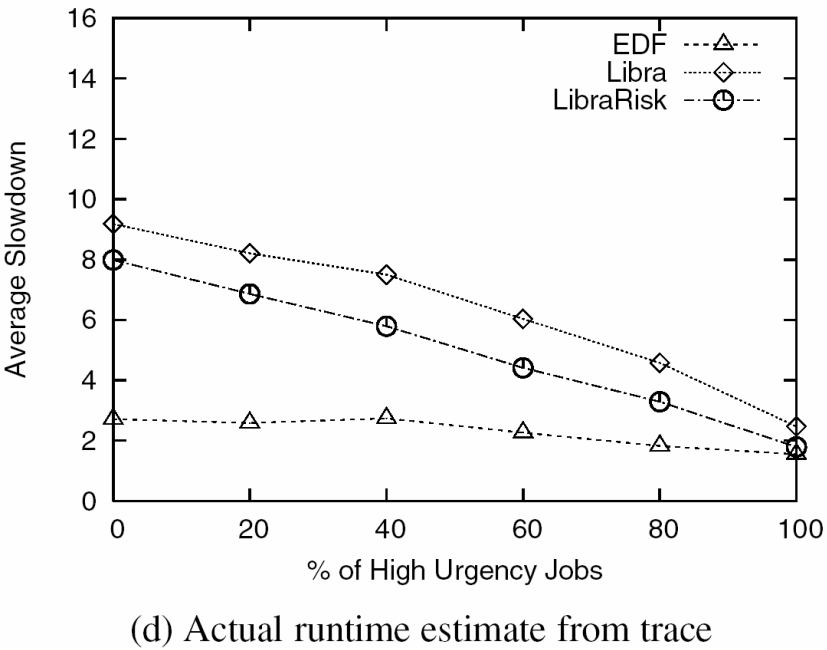
- Less jobs fulfilled with more high urgency jobs
- LibraRisk: More jobs fulfilled (more high urgency jobs)

Impact of Varying High Urgency Jobs

Average Slowdown



(c) Accurate runtime estimate

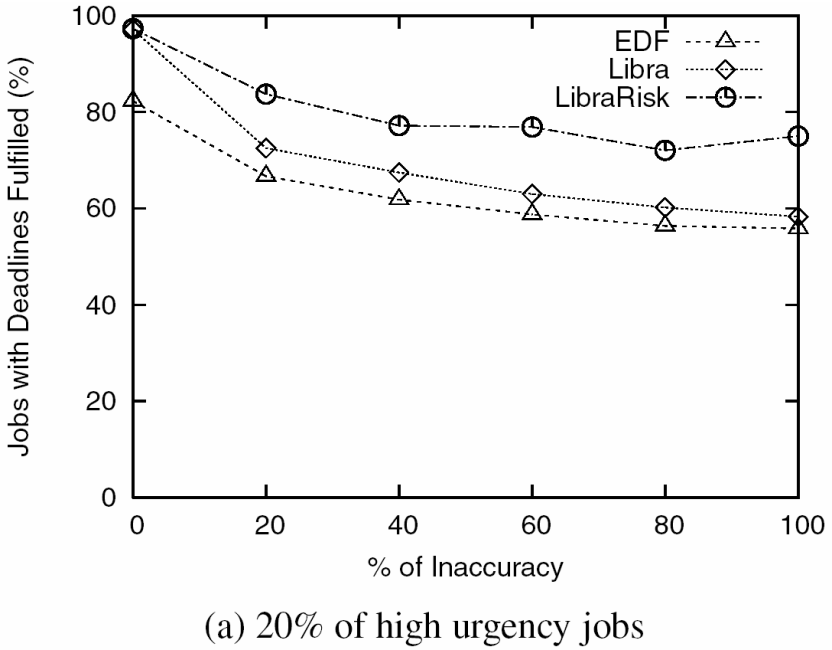


(d) Actual runtime estimate from trace

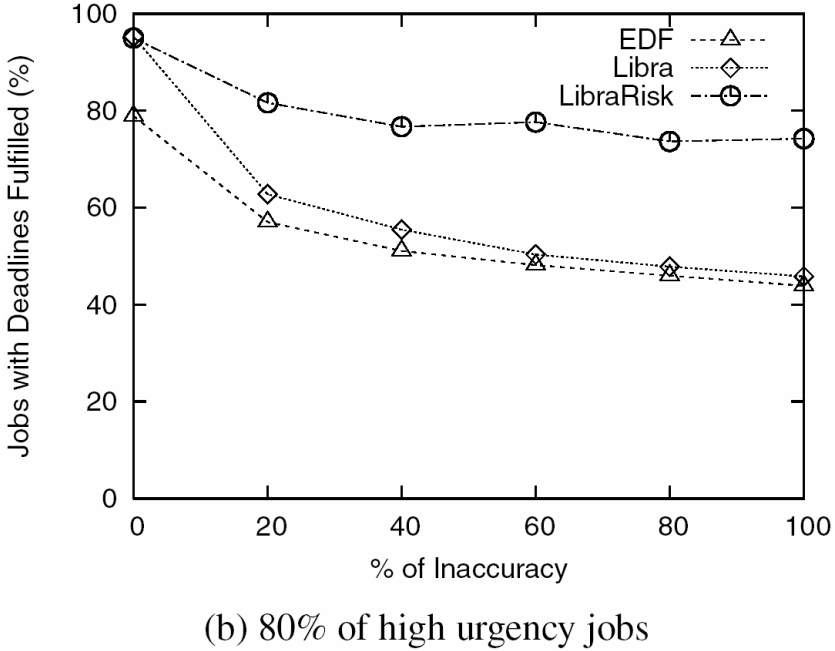
- Lower slowdown with more high urgency jobs
- LibraRisk: Lower slowdown than Libra

Impact of Varying Inaccurate Runtime Estimates

Jobs with Deadlines Fulfilled (%)



(a) 20% of high urgency jobs

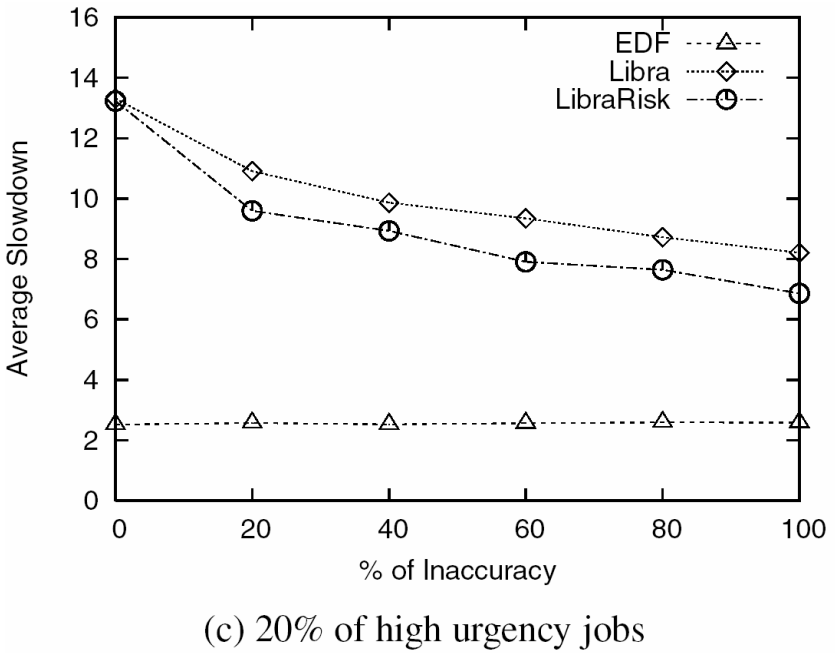


(b) 80% of high urgency jobs

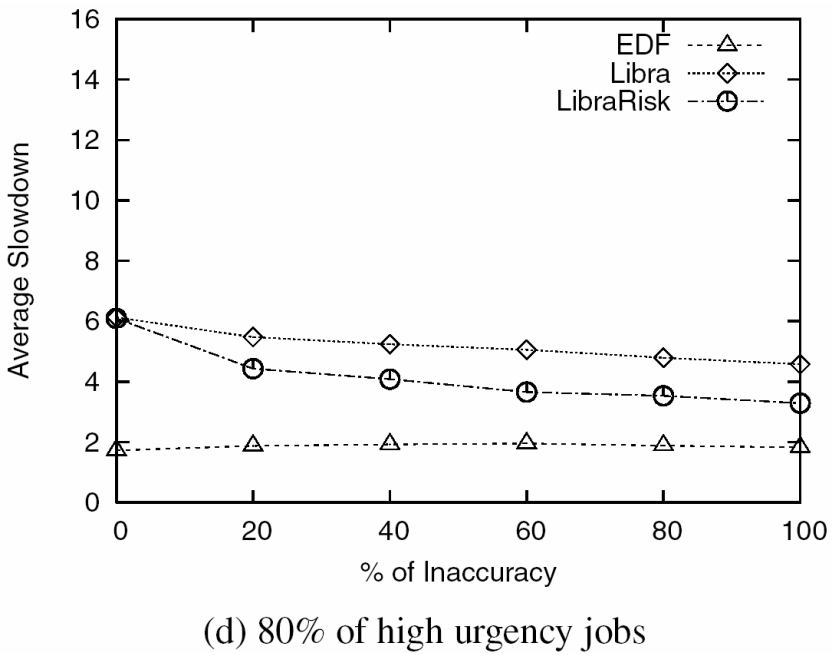
- Less jobs fulfilled with higher inaccuracy of estimates
- LibraRisk: More jobs fulfilled (higher inaccuracy of estimates)

Impact of Varying Inaccurate Runtime Estimates

Average Slowdown



(c) 20% of high urgency jobs



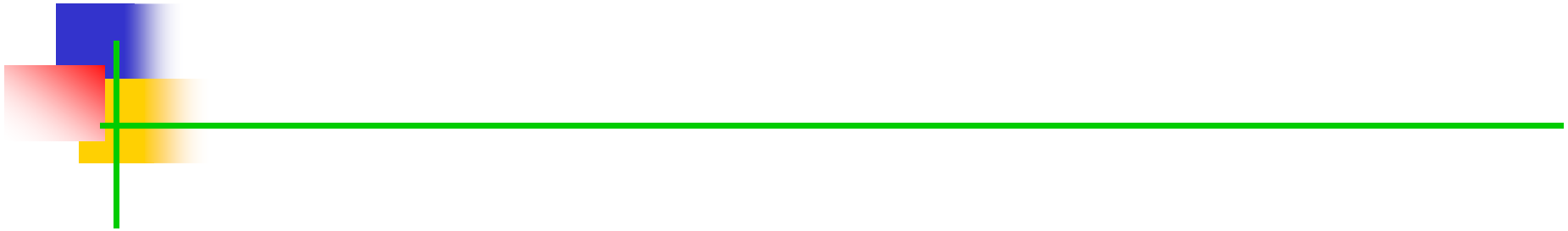
(d) 80% of high urgency jobs

- Lower slowdown with higher inaccuracy of estimates
- LibraRisk: Lower slowdown than Libra

Conclusion

- Actual runtime estimate from trace
  - Inaccurate and often over estimated
- LibraRisk
  - Manage risk of deadline delay
  - More jobs with deadlines fulfilled than EDF and Libra
    - Lower cluster workload (higher arrival delay)
    - More urgent jobs (shorter deadline)
    - Less accurate runtime estimates
  - Lower slowdown than Libra
- Future Work
  - Backfilling

End of Presentation



Questions ?