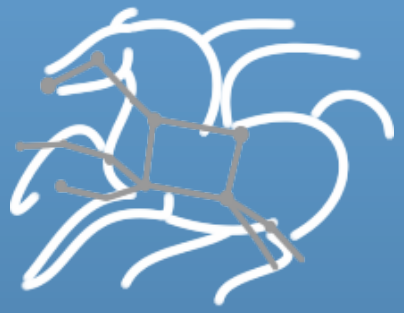


WorkflowSim: A Toolkit for Simulating Scientific Workflows in Distributed Environments

Weiwei Chen, Ewa Deelman

USC Viterbi
School of Engineering

Information Sciences Institute

Outline

- **Introduction**
  - Scientific workflows?
  - Distributed environments?
- **Challenge**
  - Large scale, system overhead
- **Solution**
  - Workflow Overhead and Failure Model
- **Validation and Application**
  - Overhead Robustness
  - Fault Tolerant Clustering

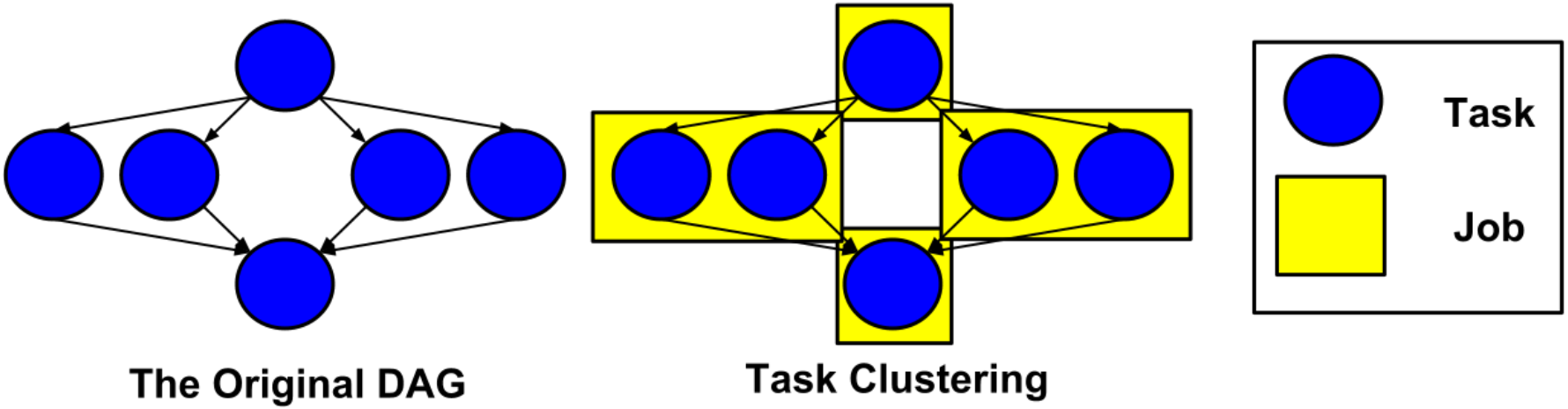
Scientific Applications

- **Scientists often need to**
  - Integrate diverse components and data
  - Automate data processing steps
  - Reproduce/analyze/share previous results
  - Track the provenance of data products
  - Execute processing steps efficiently
  - Reliably execute applications

Scientific Workflows provide solutions to these problems

Scientific Workflows

- **DAG model (Directed Acyclic Graph)**
  - Node: computational activities
  - Directed edge: data dependencies
  - Task: a process that users would like to execute
  - Job: a single unit for execution with one of more tasks
  - Task clustering: the process of grouping tasks to jobs



Workflow Management System

- **Common Features of WMS**
  - Maps abstract workflows to executable workflows
  - Handles data dependencies
  - Replica selection, transfers, registration, cleanup
  - Task clustering, workflow partitioning, scheduling
  - Reliability and fault tolerance
  - Monitoring and troubleshooting
- **Existing WMS**
  - Pegasus, Askalon, Taverna, Kepler, Triana

Workflow Simulation

- **Benefits**
  - Save efforts in system setup, execution
  - Repeat experimental results
  - Control system environments (failures)
- **Trace based Workflow Simulation**
  - Import trace from a completed execution
  - Vary workflow structures and system environments
- **Challenges (CloudSim, GridSim, etc.)**
  - System overhead and failures
  - Multiple levels of computational activities(task/job)
  - A hierarchy of management components

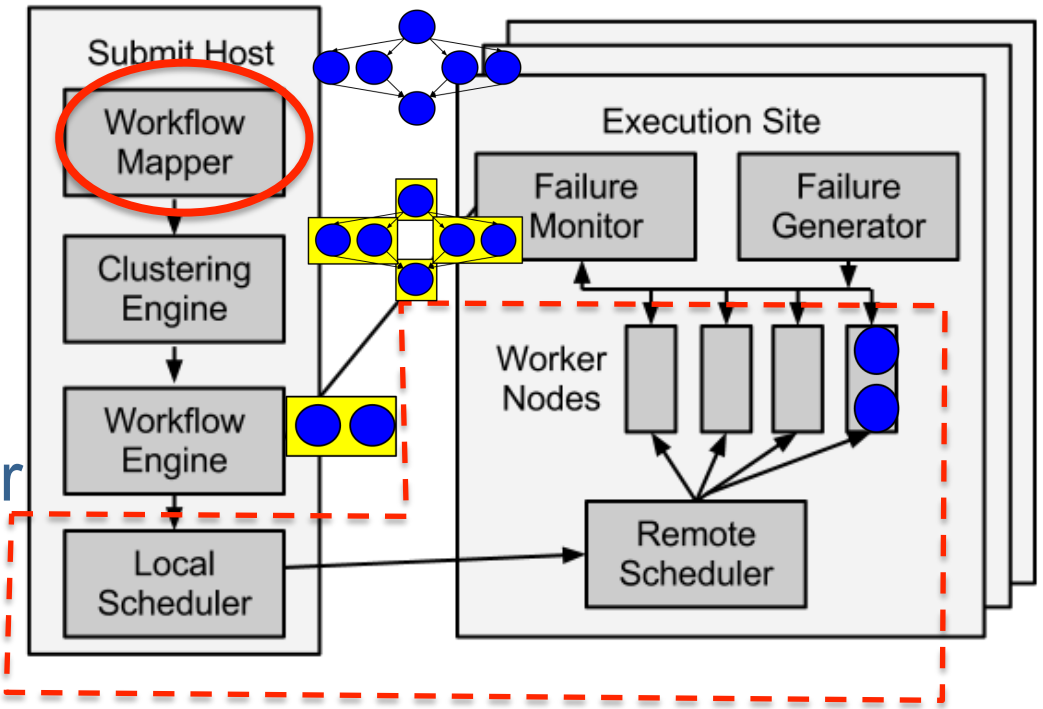
Comparison

	CloudSim	WorkflowSim
Execution Model	Task, Bag of Tasks	Task, Job, DAG
Failure and Monitoring	No	Yes
Overhead Model	Data Transfer Delay	Data Transfer Delay Workflow Engine Delay Clustering Delay …
Site selection	Single	Multiple
Optimization Techniques	Scheduling	Scheduling, Job retry Clustering, Partitioning …

WorkflowSim is an extension of CloudSim, but it is workflow aware

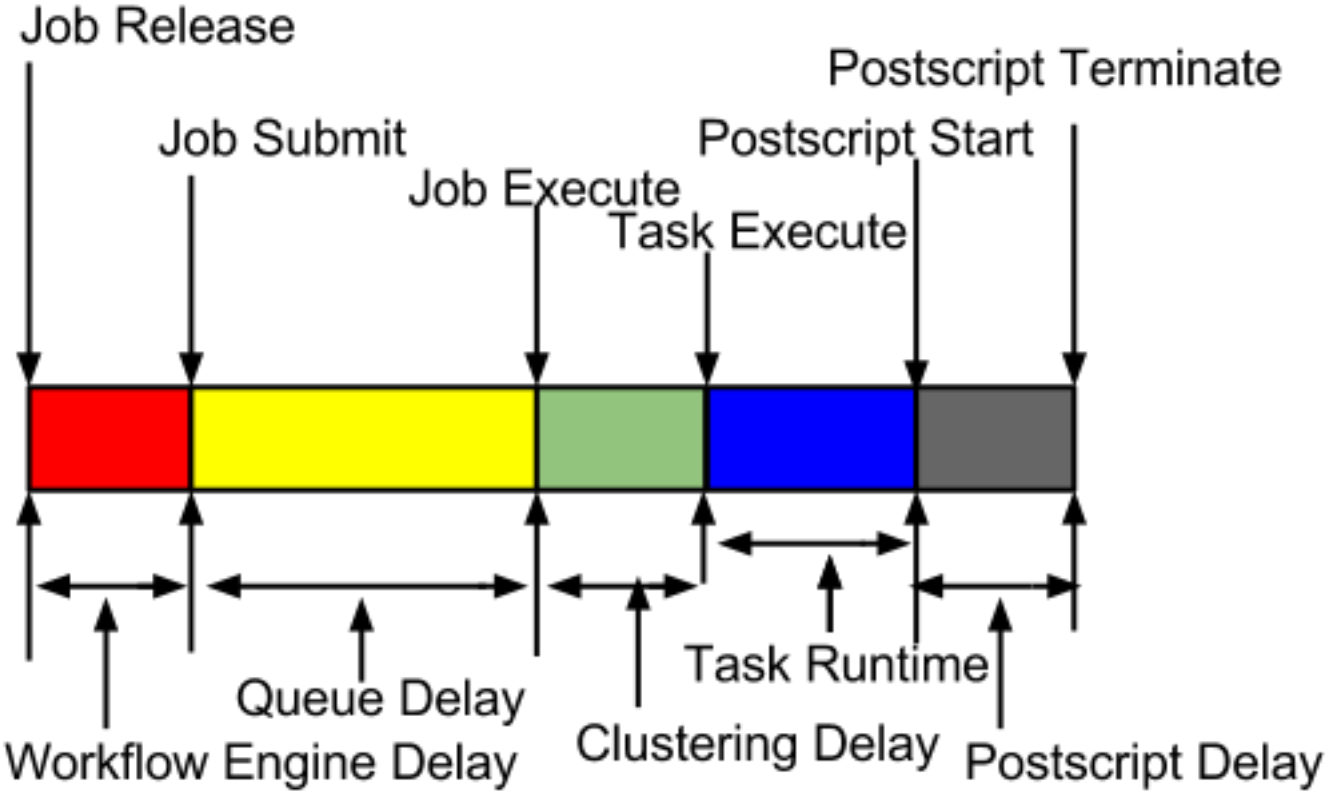
System Architecture

- **Submit Host**
  - Workflow Mapper
  - Clustering Engine
  - Workflow Engine
  - Local Scheduler
- **Execution Site**
  - Remote Scheduler
  - Worker Nodes
  - Failure Generator
  - Failure Monitor

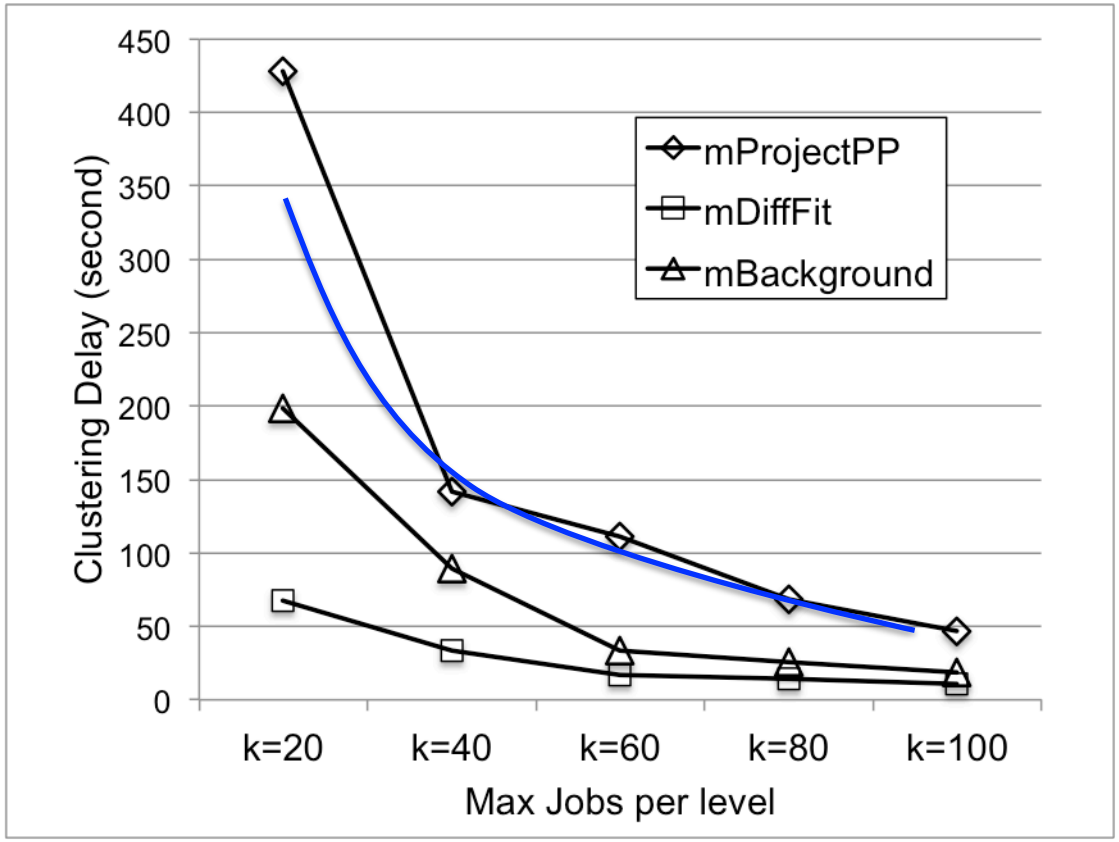


Workflow Overhead

- Workflow Engine Delay
- Queue Delay
- Postscript Delay
- Clustering Delay
- Data Transfer Delay



Example: Clustering Delay



n is the number of tasks per level. *k* is number of jobs per level.

$$\frac{Clustering\ Delay|_{k=i}}{Clustering\ Delay|_{k=j}} = \frac{n/_i}{n/_j} = \frac{j}{i}$$

mProjectPP, mDiffFit, and mBackground are the major jobs of Montage

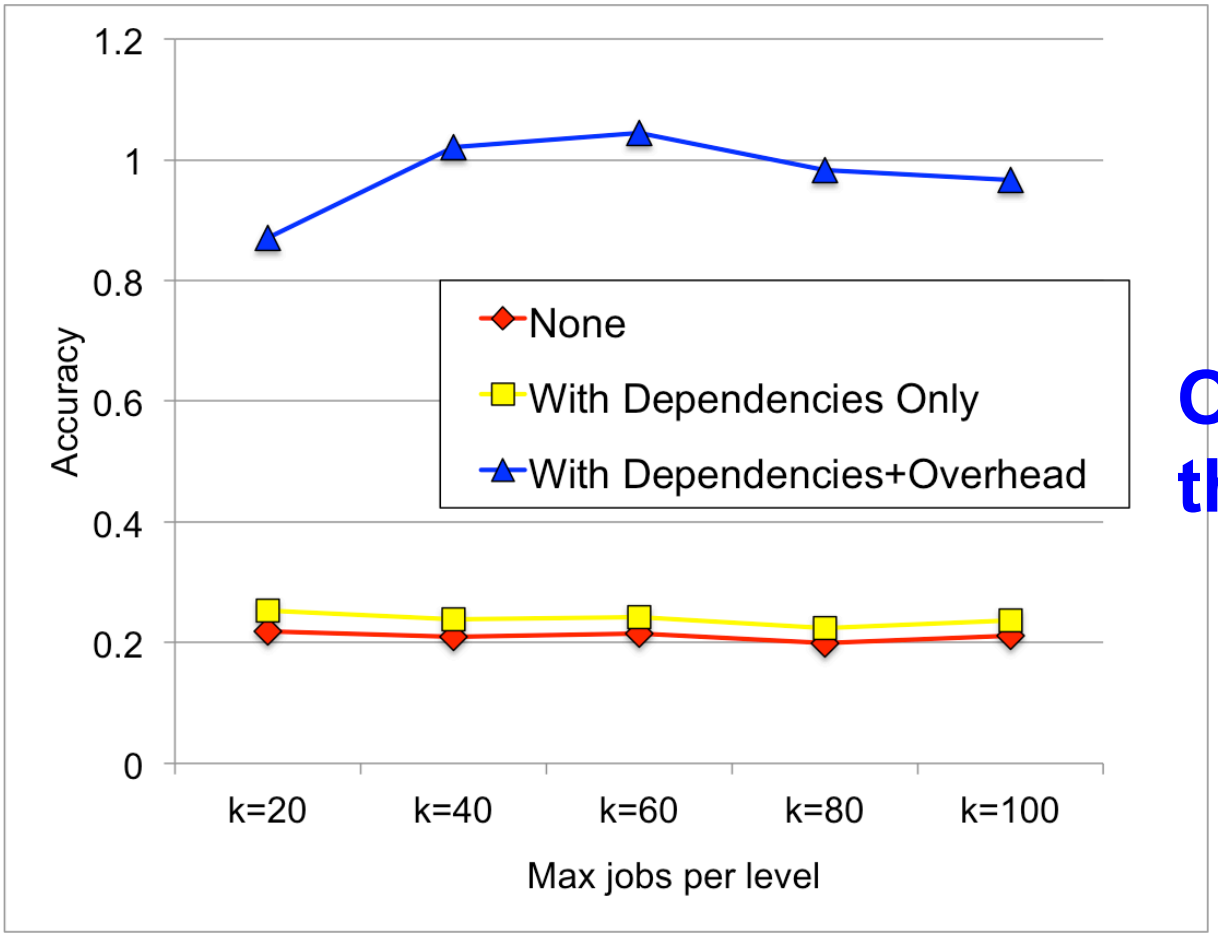
Overhead is not a constant variable.
It has diverse distribution and patterns

Validation

$$Accuracy = \frac{Predicted\ Overall\ Runtime}{Real\ Overall\ Runtime}$$

Ideal Case: Accuracy=1.0

k: Maximum jobs per level



Overheads have the biggest impact

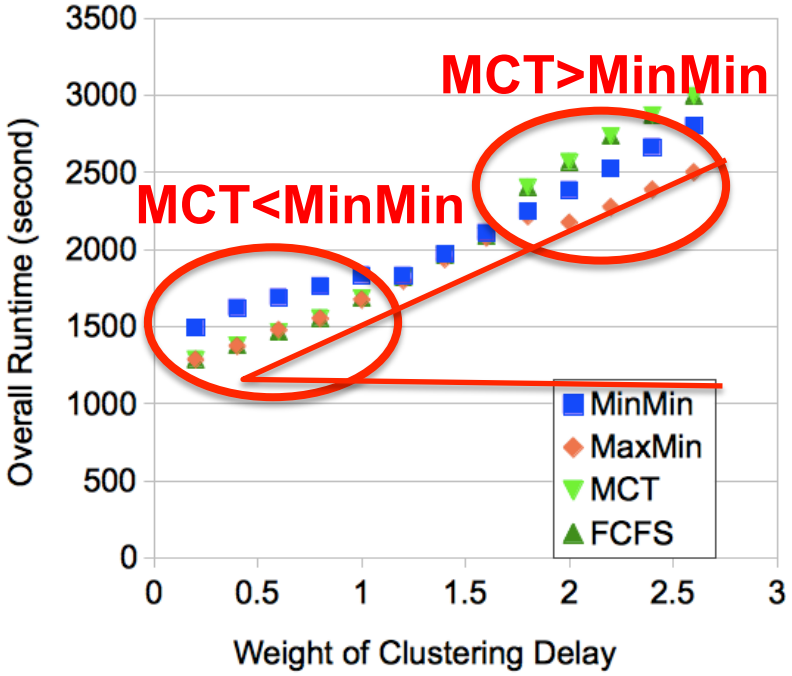
Application: Overhead Robustness

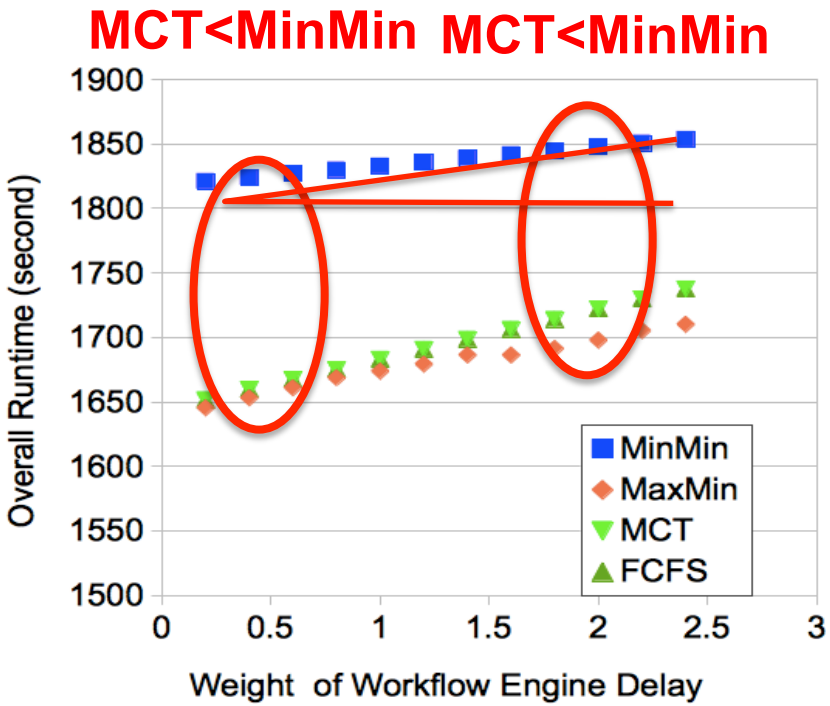
- Overhead robustness: the influence of overheads on the workflow runtime for DAG scheduling heuristics.
- Inaccurate estimation (under- or over-estimated) of workflow overheads influences the overall runtime of workflows.
- Research Merits:
  - Sensitivity of heuristics
  - Overhead friendly heuristics

Application: Overhead Robustness

- **Increase or Reduce Overhead by a Factor (Weight)**
  - Under estimation and over estimation
- **Heuristics Evaluated**
  - **FCFS**: First Come First Serve
  - **MCT**: Minimum Completion Time
  - **MinMin**: The job with the minimum completion time is selected and assigned to the fastest resource.
  - **MaxMin**: The job with the maximum completion time and assigns it to its best available resource.

Application: Overhead Robustness

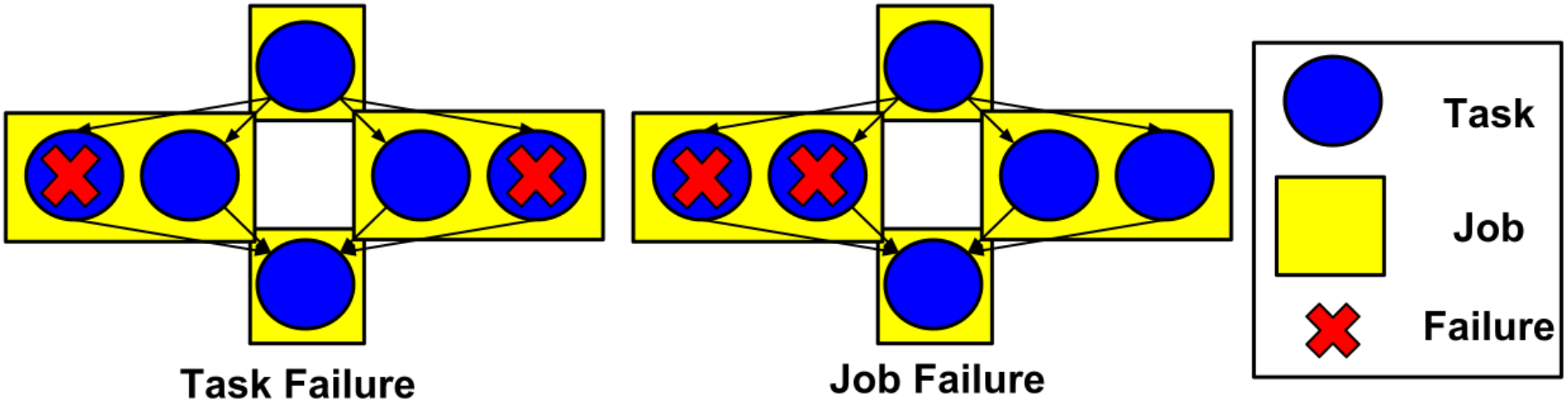




Accurate estimation of Clustering Delay is more important

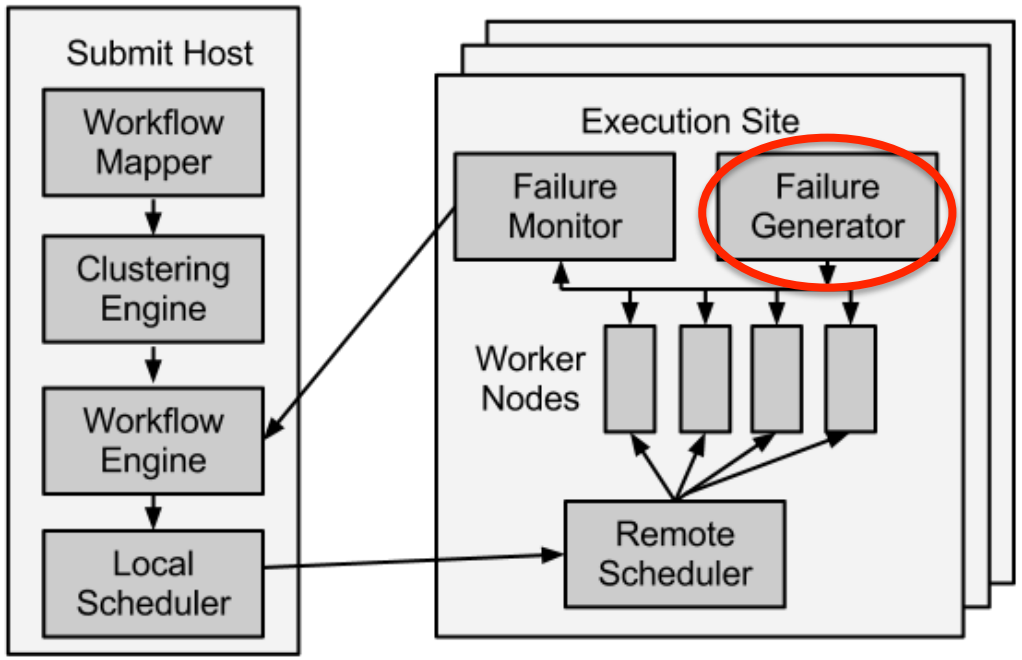
Workflow Failure

- Failures have significant influence on the performance
- Classifying Transient Failures
  - Task Failure: A task fails, other tasks within the same job may not fail
  - Job Failure: A job fails, all of its tasks fail



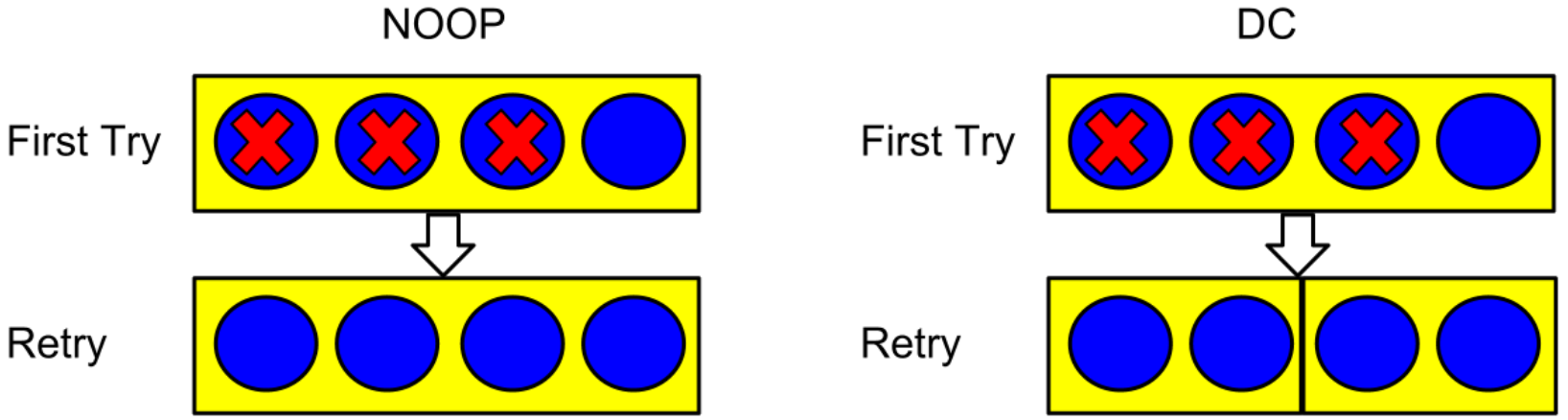
System Architecture

- **Submit Host**
  - Job Retry
  - Reclustering
- **Execution Site**
  - Failure Generator
  - Failure Monitor



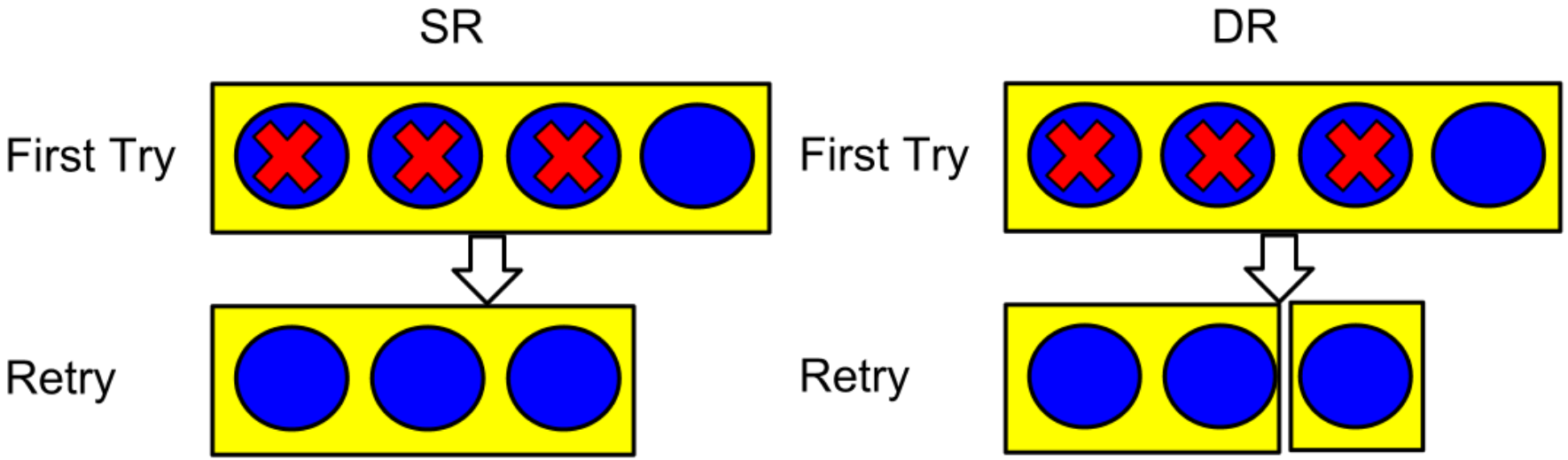
Application: Fault Tolerant Clustering

- Task clustering can reduce execution overhead
- A job composed of multiple tasks may have a greater risk of suffering from failures
- Reclustering and Job Retry are proposed
  - *No Optimization* (NOOP) retries the failed jobs.
  - *Dynamic Clustering* (DC) decreases the *clusters.size* if the measured job failure rate is high.



Application: Fault Tolerant Clustering

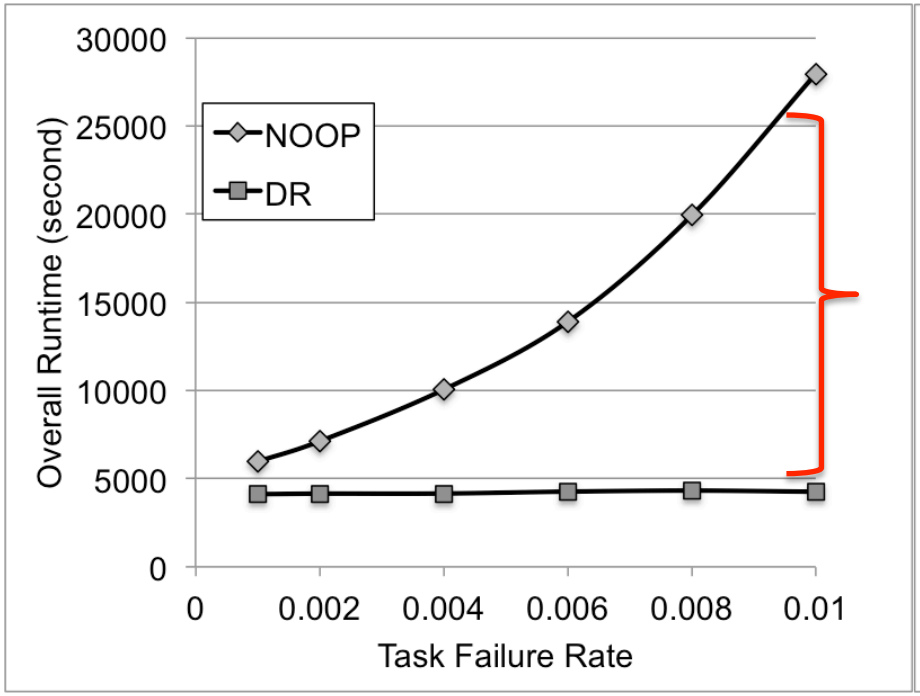
- *Selective Reclustering* (SR) retries the failed tasks in a job
- *Dynamic Reclustering* (DR) retires the failed tasks in a job and also decreases the *clusters.size* if the measured job failure rate is high.

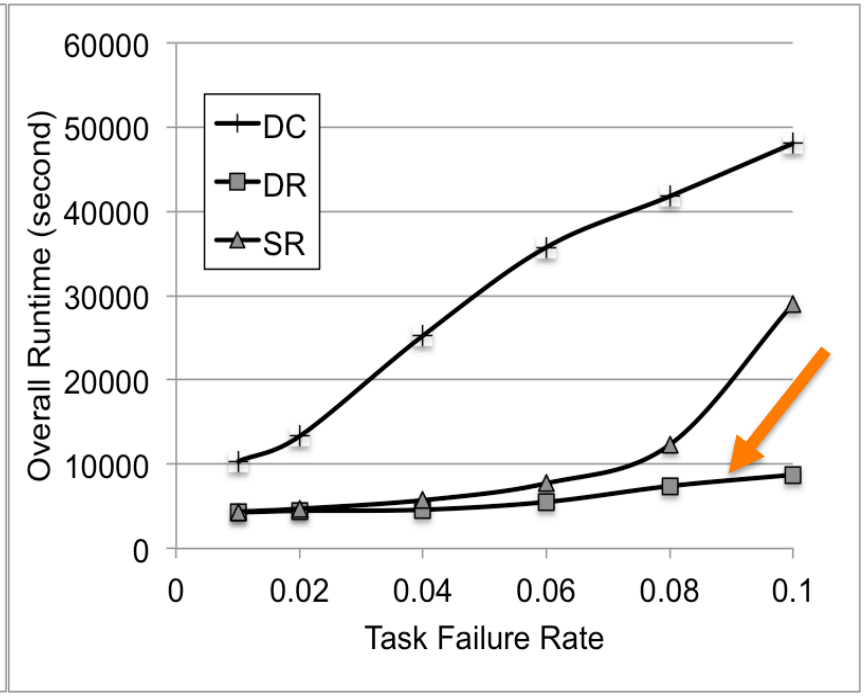


Performance

- Reclustering (DR/DC/SR) reduces the influence of failures significantly compared to NOOP
- DR outperforms other techniques.

The lower the better





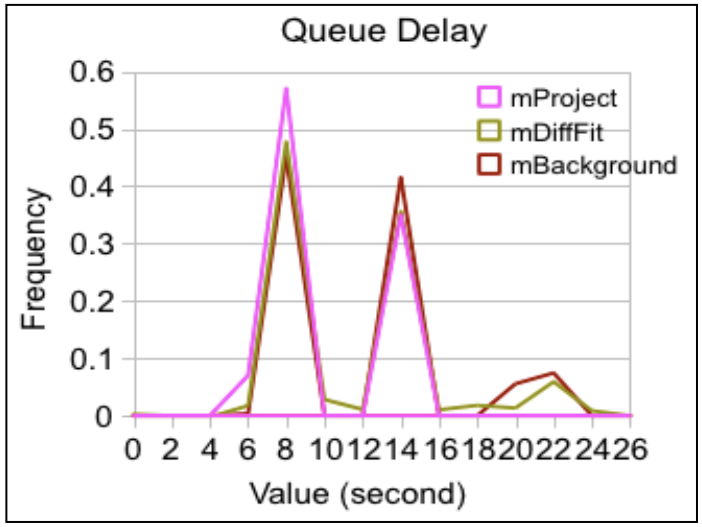
Conclusion

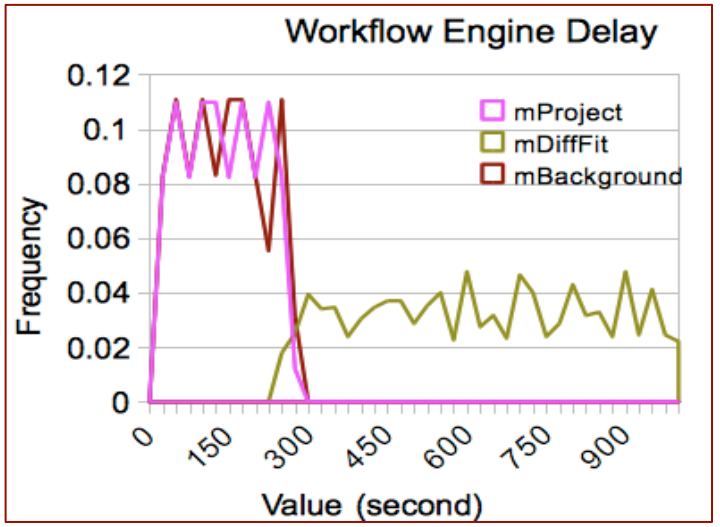
- WorkflowSim assists researchers to evaluate their workflow optimization techniques with better accuracy and wider support.
  - Modeling Overhead and Failures
  - Distributed and Hierarchical components
  - Workflow Techniques
- It is necessary to consider both data dependencies, workflow failures and system overhead.

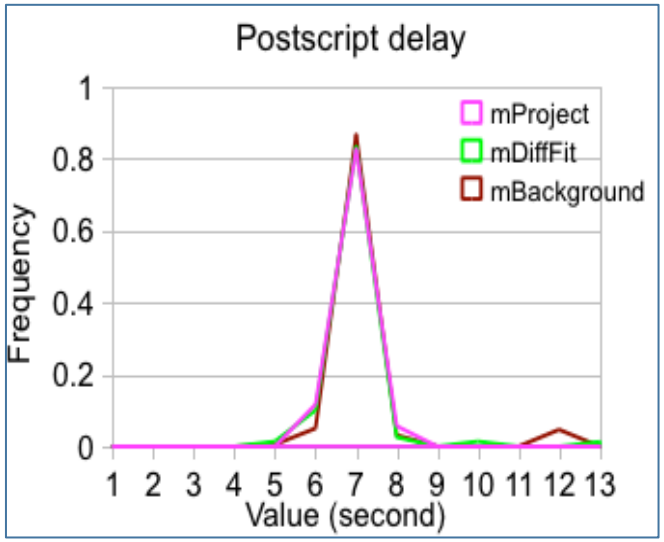
Acknowledgements

- Pegasus Team: http://pegasus.isi.edu
- FutureGrid & XSEDE
- Funded by NSF grants IIS-0905032.
- More info: wchen@isi.edu
- Available on request
- Q&A

Overhead Distribution

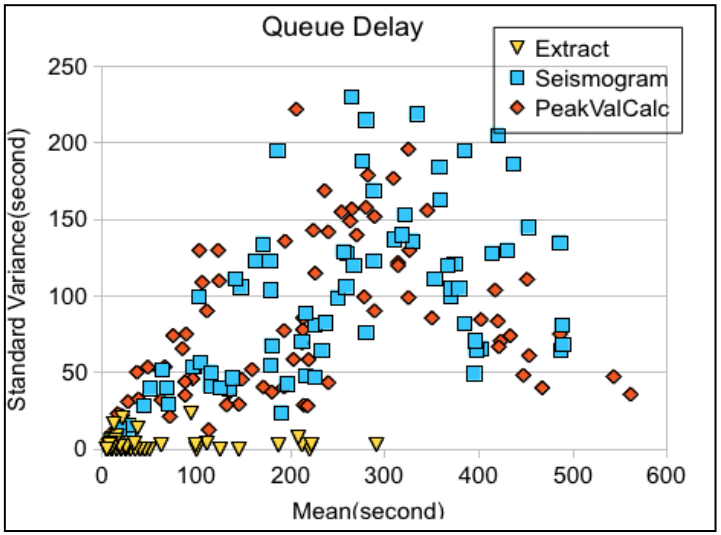


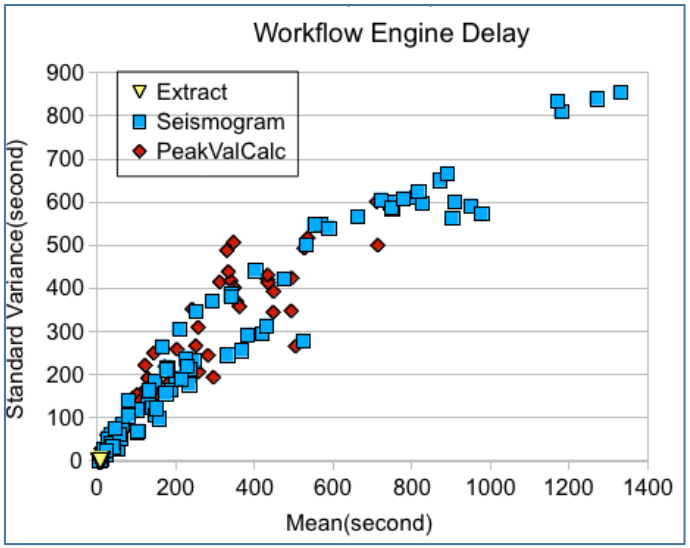


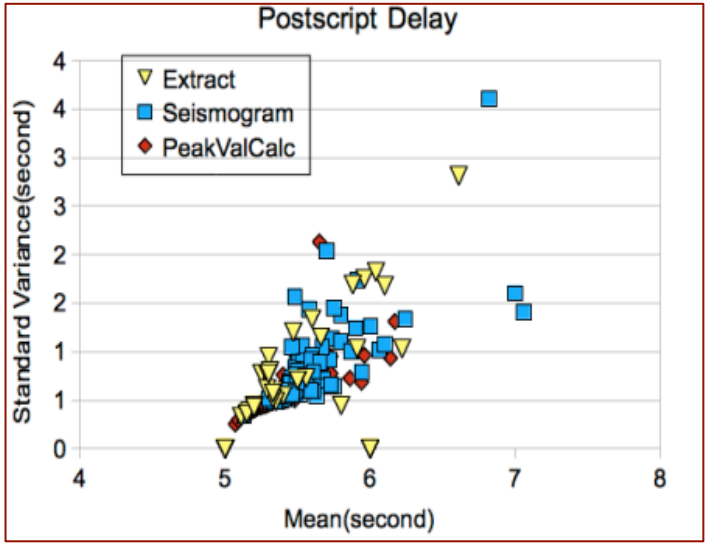


Montage

Overhead Distribution

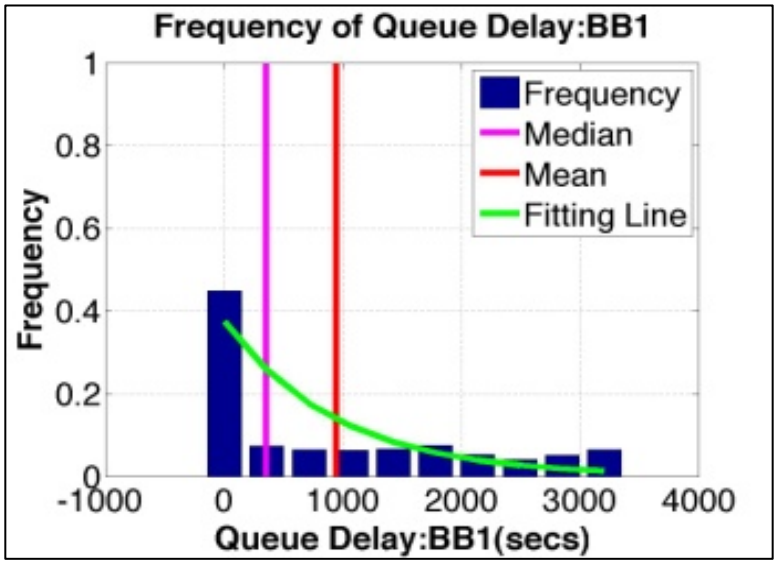


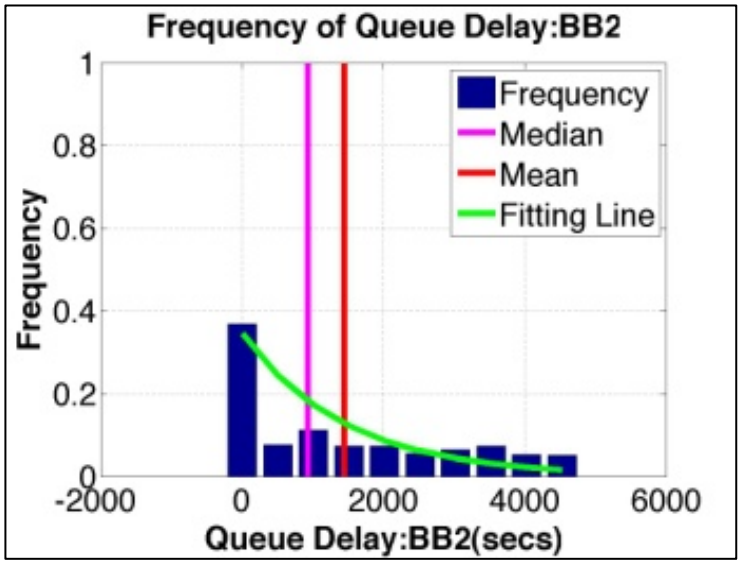




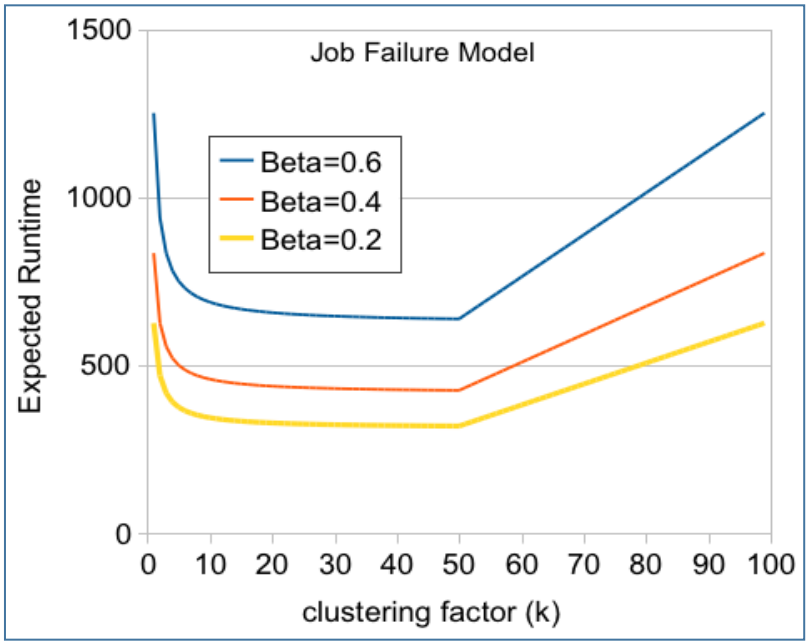
CyberShake

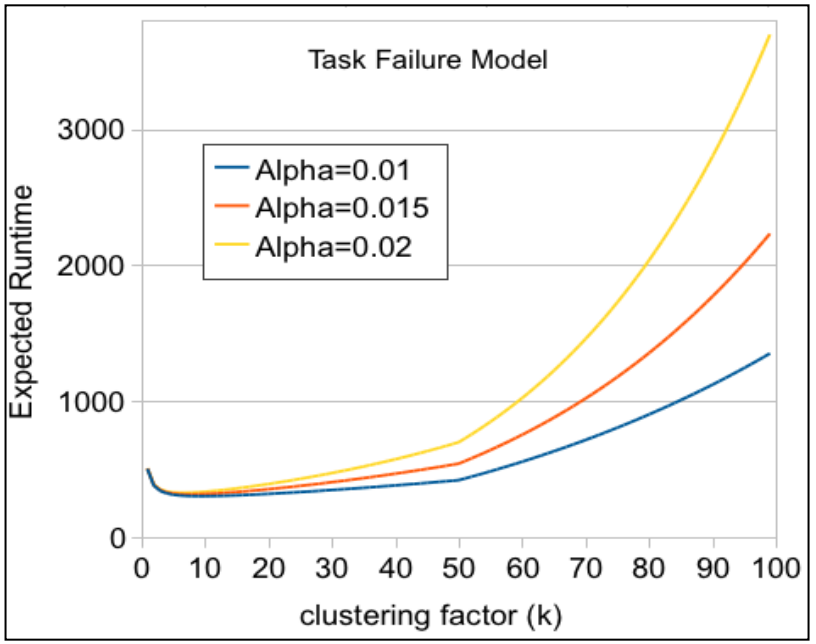
Broadband


Frequency of Queue Delay:BB1
Frequency
Median
Mean
Fitting Line
Frequency
1
0.8
0.6
0.4
0.2
0
-1000
0
1000
2000
3000
4000
Queue Delay:BB1(secs)


Frequency of Queue Delay:BB2
Frequency
Median
Mean
Fitting Line
Frequency
1
0.8
0.6
0.4
0.2
0
-2000
0
2000
4000
6000
Queue Delay:BB2(secs)

Task Failure Model and Job Failure Model





$$k^* = \frac{n}{r}$$

$$t^*_{total} = \frac{(kt+d)}{1-\beta}$$

$$k^* = \frac{-d + \sqrt{d^2 - \frac{4d}{\ln(1-\alpha)}}}{2t}, \quad if \quad n >> r$$

$$t^*_{total} = \frac{n(k^*t+d)}{rk(1-\alpha)^{k^*}}$$

Related Papers

- Integration of Workflow Partitioning and Resource Provisioning, CCGrid 2012.
- Improving Scientific Workflow Performance using Policy Based Data Placement, IEEE Policy 2012.
- Fault Tolerant Clustering in Scientific Workflows, SWF, IEEE Services 2012
- Workflow Overhead Analysis and Optimizations, WORKS 2011.
- Partitioning and Scheduling Workflows across Multiple Sites with Storage Constraints, PPAM 2011
- Pegasus: a Framework for Mapping Complex Scientific Workflows onto Distributed Systems, Scientific Programming Journal 2005