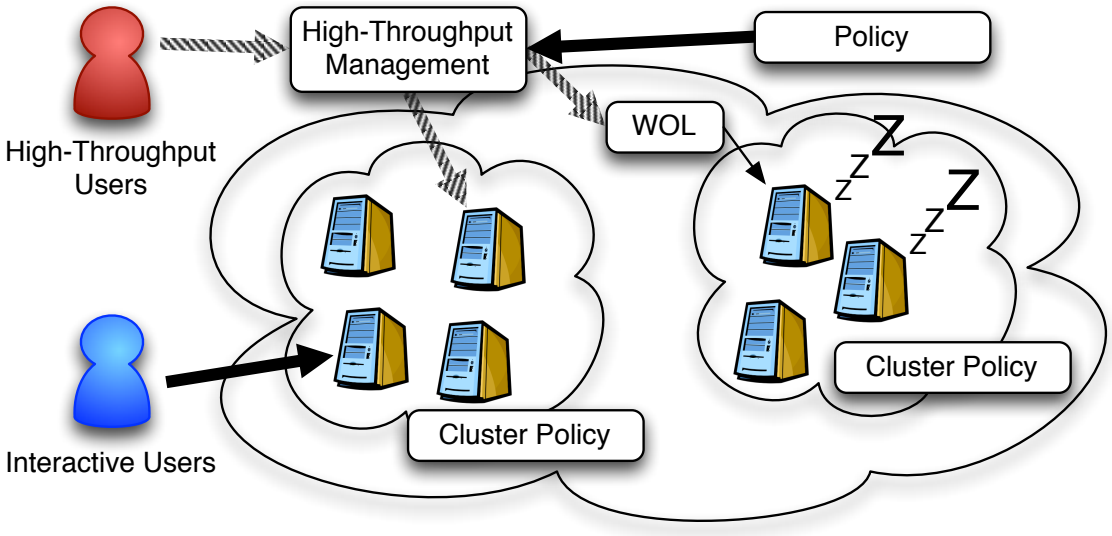
Energy-aware simulation of workflow execution in High Throughput Computing systems

Stephen McGough
Durham University
Matthew Forshaw
Newcastle University

14/10/2015

Opportunistic High-throughput cluster

- Using collections of distributed workstations and/or dedicated clusters as a distributed high-throughput computing (HTC) facility
  - manages both resources (machines) and requests (jobs)
  - Often used to exploit existing computing facilities
  - Resilient architecture
    - If a job fails to complete on one resource it will be reallocated to a different resource



Dedicated

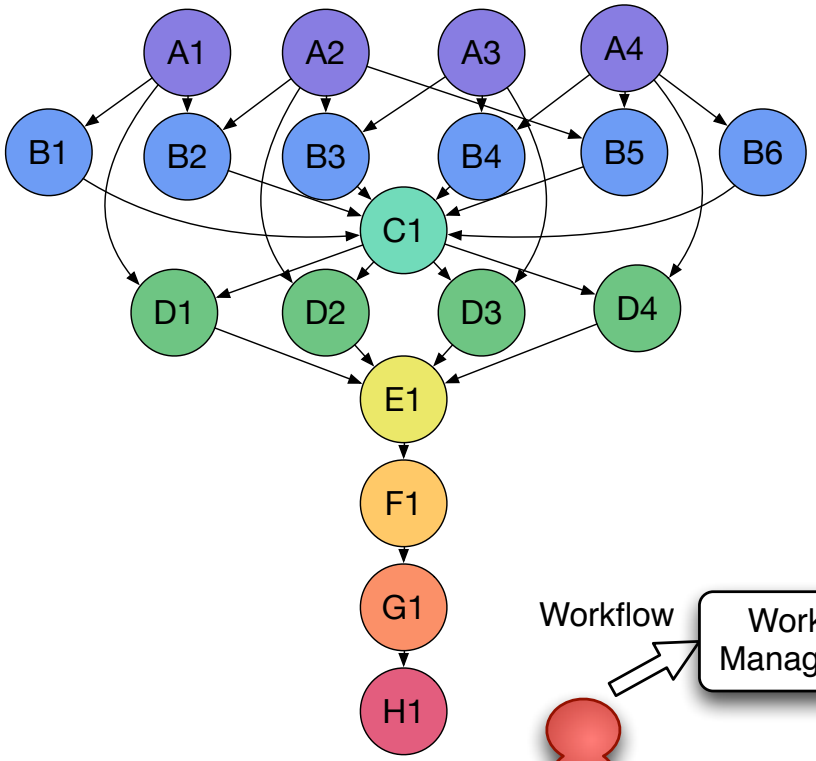
Cycle stealing

HTC

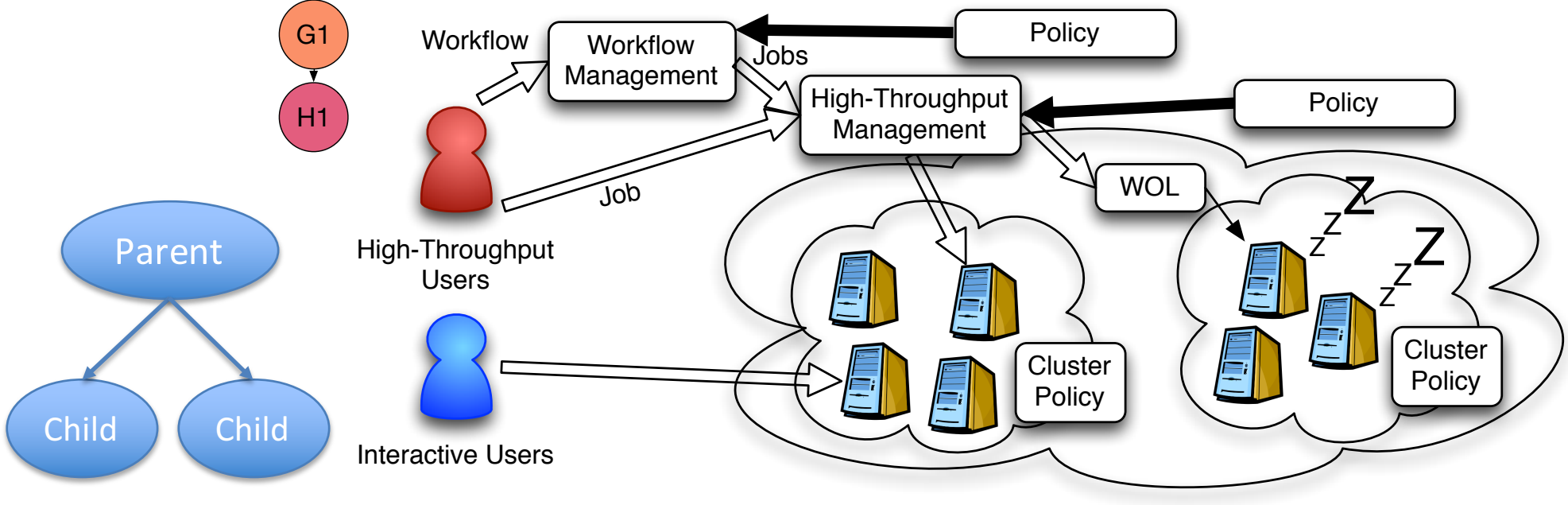
Motivation: Workflows

- Common way of performing complicated tasks
  - With dependencies between (sub) jobs
  - May contain many (sub) jobs
- Running on a dedicated system is well understood
- But what if resources can be lost?
  - How will this impact energy consumption?
  - How will this impact the time taken to perform work?
- Aims
  - Extend HTC-Sim to allow workflows
  - Evaluate workflow scheduling for energy / time impact

Workflow



- Constructed as a DAG
  - Point-wise dependencies
  - Job B2 depends on A1 & A2
- Extend model to support

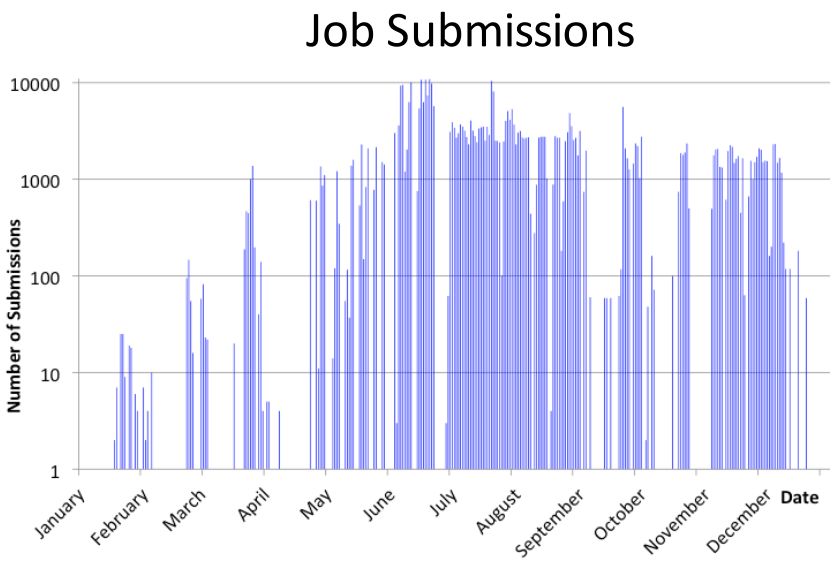


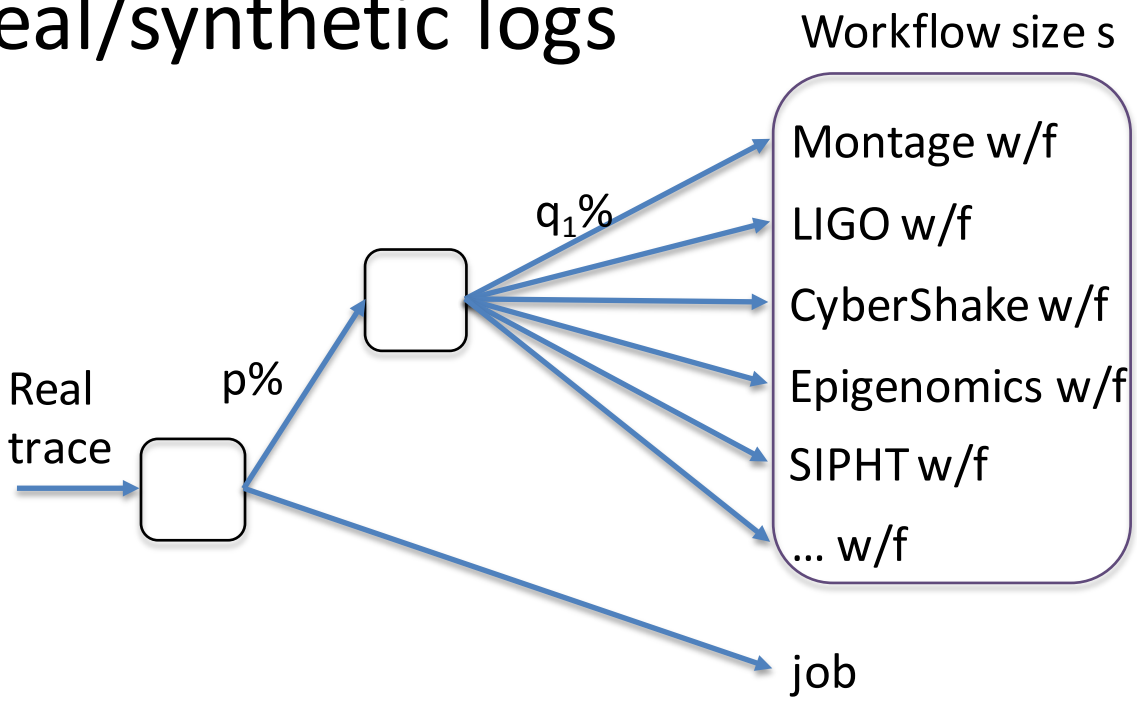
Workflow Management

- All ‘runnable’ (sub) jobs are sent to HTC manager
  - Runnable – all pre-dependencies are satisfied
- As each job completes:
  - If no child jobs – check if workflow is completed
  - If has children – submit all runnable children
- Workflows can be terminated at any time by killing any job in the workflow
- Evicted workflow jobs will be resubmitted by the HTC manager

Real/Synthetic Traces

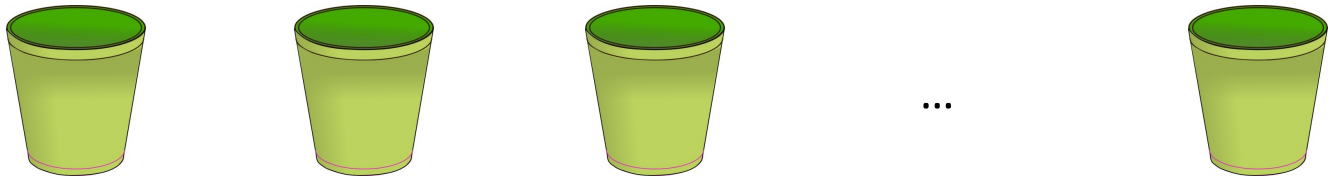
- We have real trace logs for 2010
  - But no workflow logs
- Generate hybrid real/synthetic logs



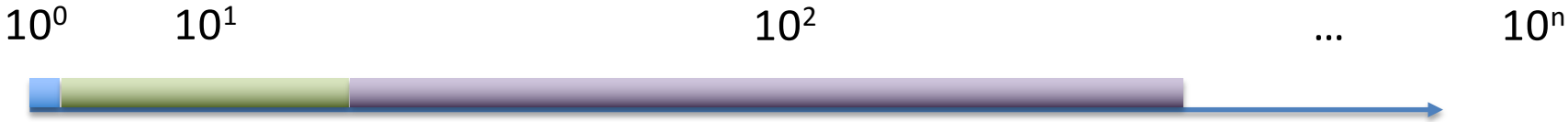


Estimating Execution time

- How long will each job in a workflow take?
  - **A priori:** Assume we have perfect knowledge
  - **Binned:** Users only need to estimate time into 'bins'



  - **Order of Magnitude:** Users only know time to the order of magnitude



Job selection policy

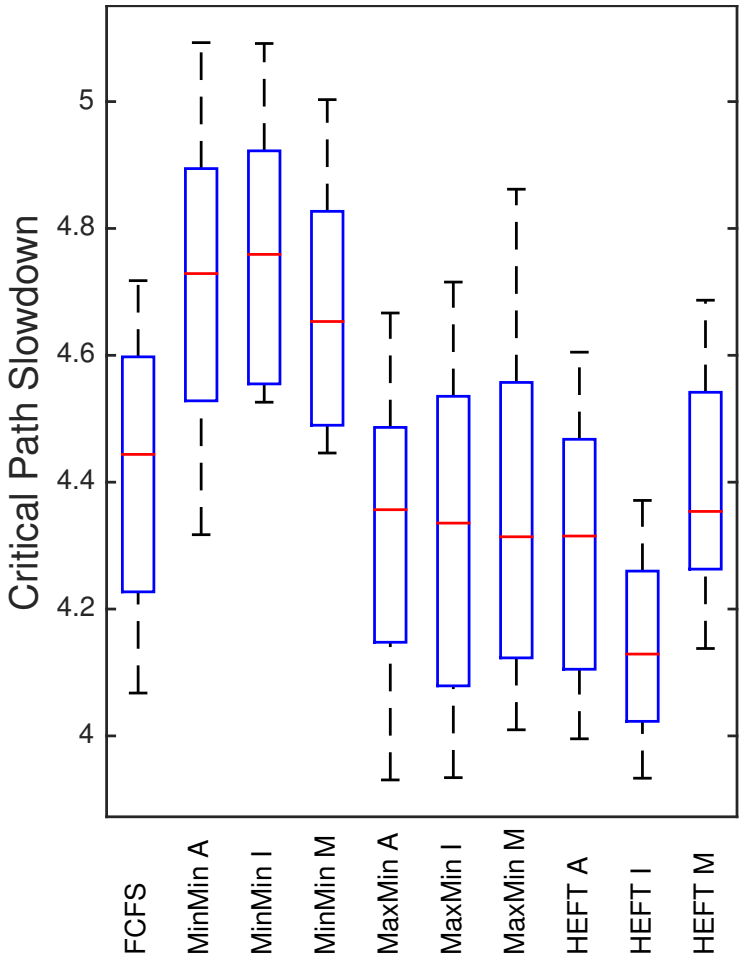
- Which job to run next?
  - **FCFS:** In order of arrival times
  - **Min-Min:** Short jobs are given priority
  - **Max-Min:** Longer jobs are given priority
  - **Heterogeneous Earliest Finish Time (HEFT):**
    - Based on future impact of job and its successors
    - Computed recursively

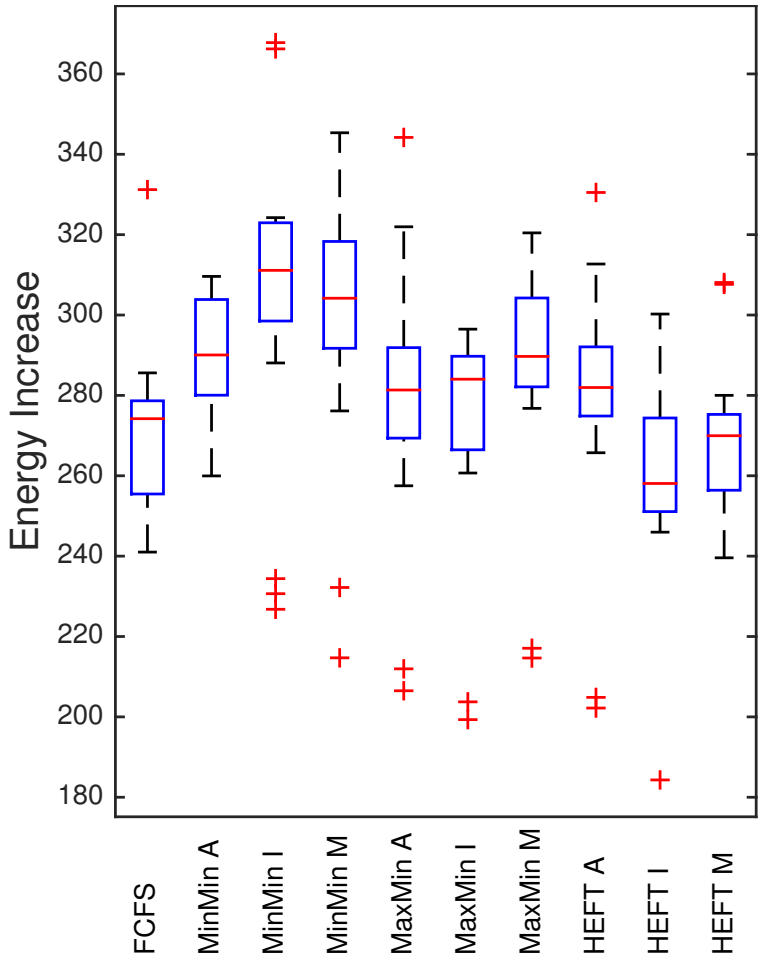
Resource Selection Policy

- Which Computer to use?
  - **S1:** Default HTCondor policy
    - random resource selection favoring powered up computers
  - **S2:** Target the most energy efficient computers
  - **S3:** Prioritize least interactive user activity
    - a: largest average inter-user interval
    - b: smallest number of interactive users
  - **S4:** Target clusters closed for use by interactive users

Impact of Job Selection Policy

- Workflow Size: 20

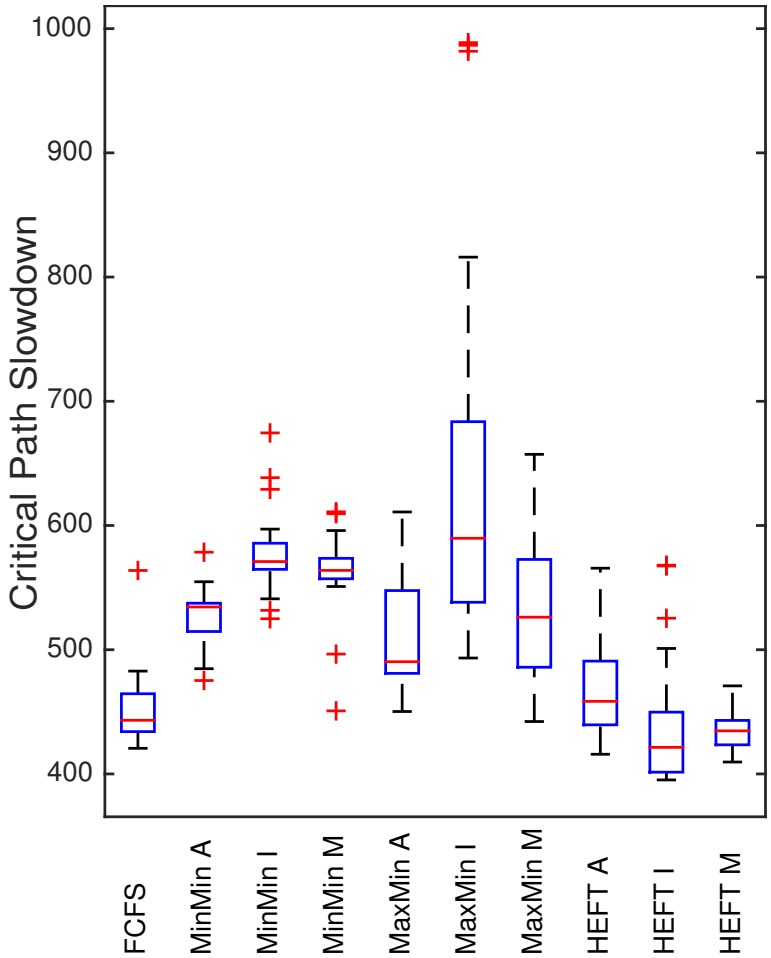


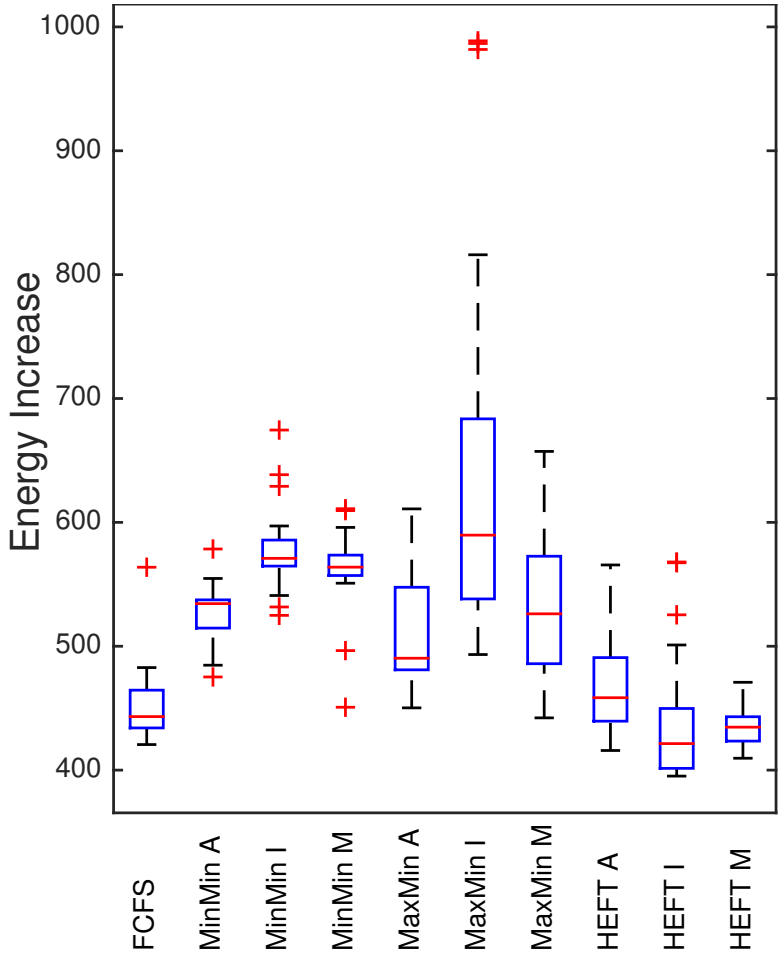


A – a priori I – binned M – order of magnitude

Impact of Job Selection Policy

- Workflow Size: 50

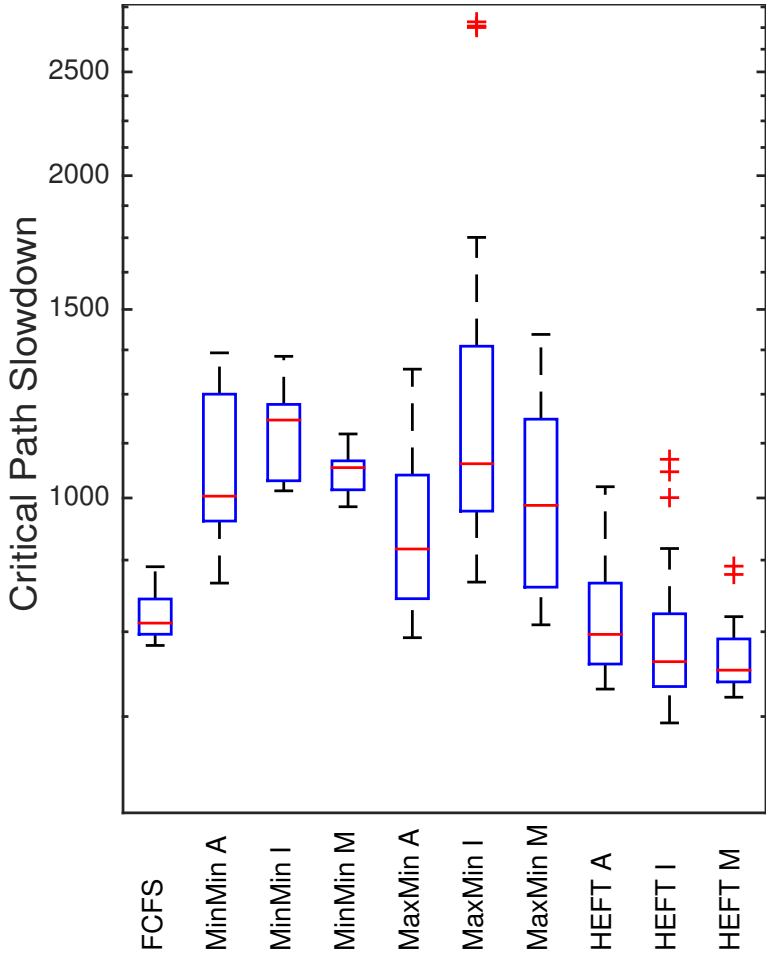


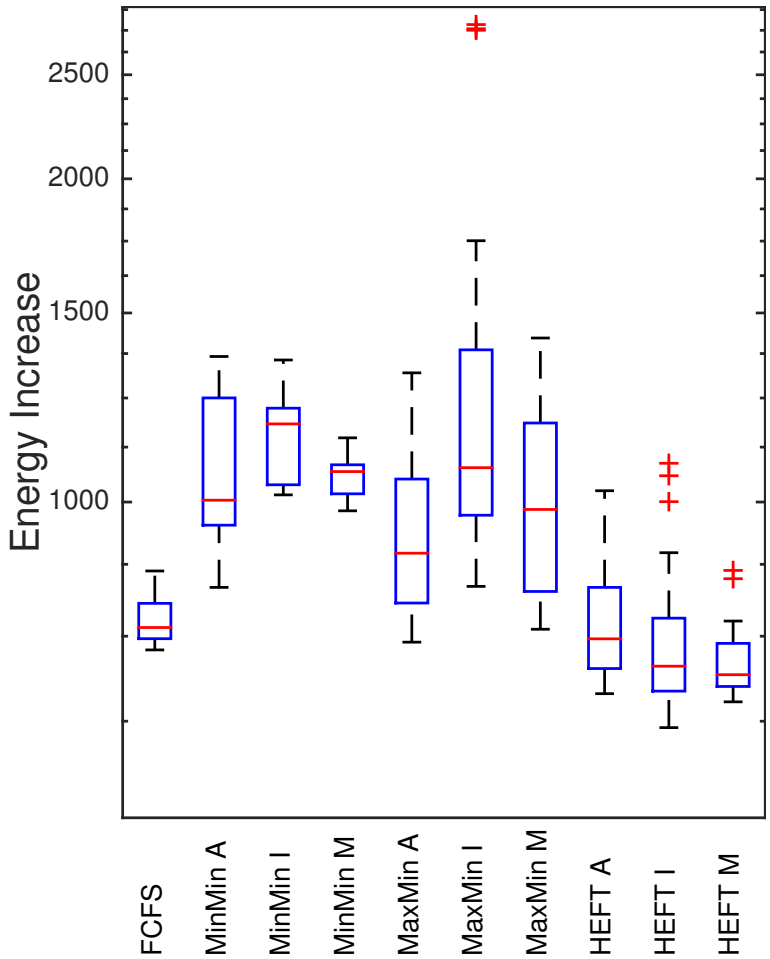


A – a priori I – binned M – order of magnitude

Impact of Job Selection Policy

- Workflow Size: 100

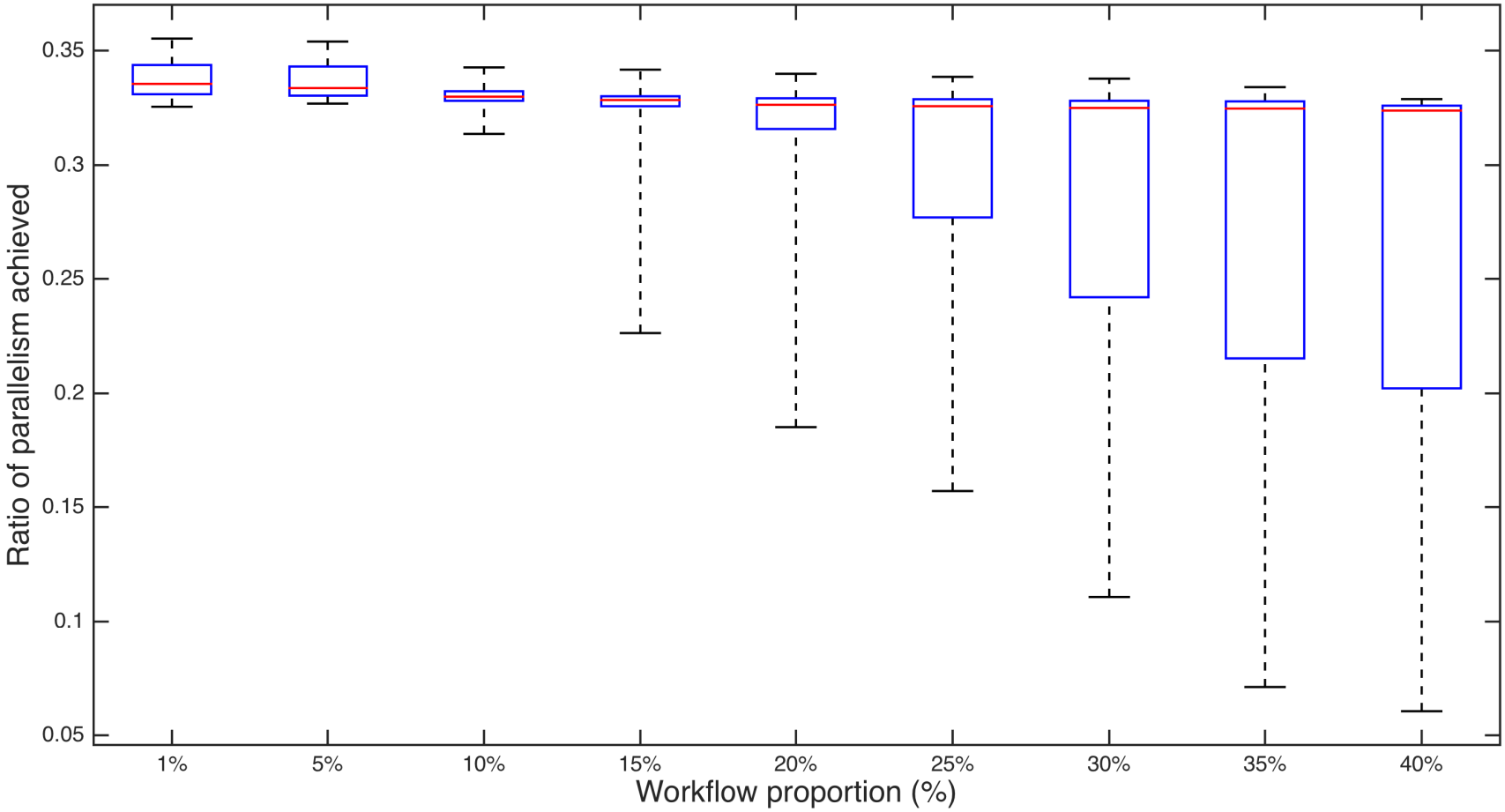




A – a priori I – binned M – order of magnitude

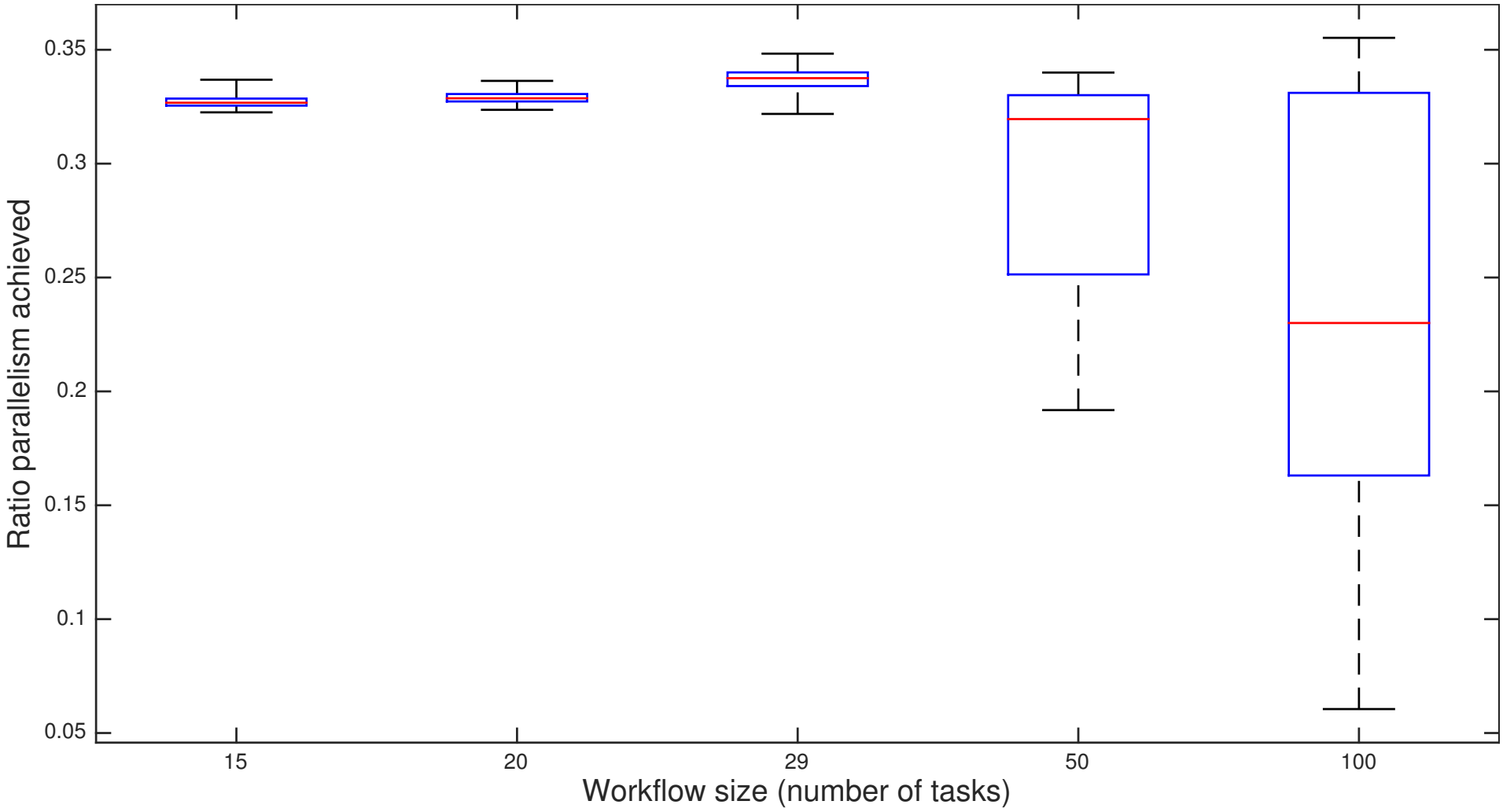
Parallelism achieved

- Varying workflow percentage p%



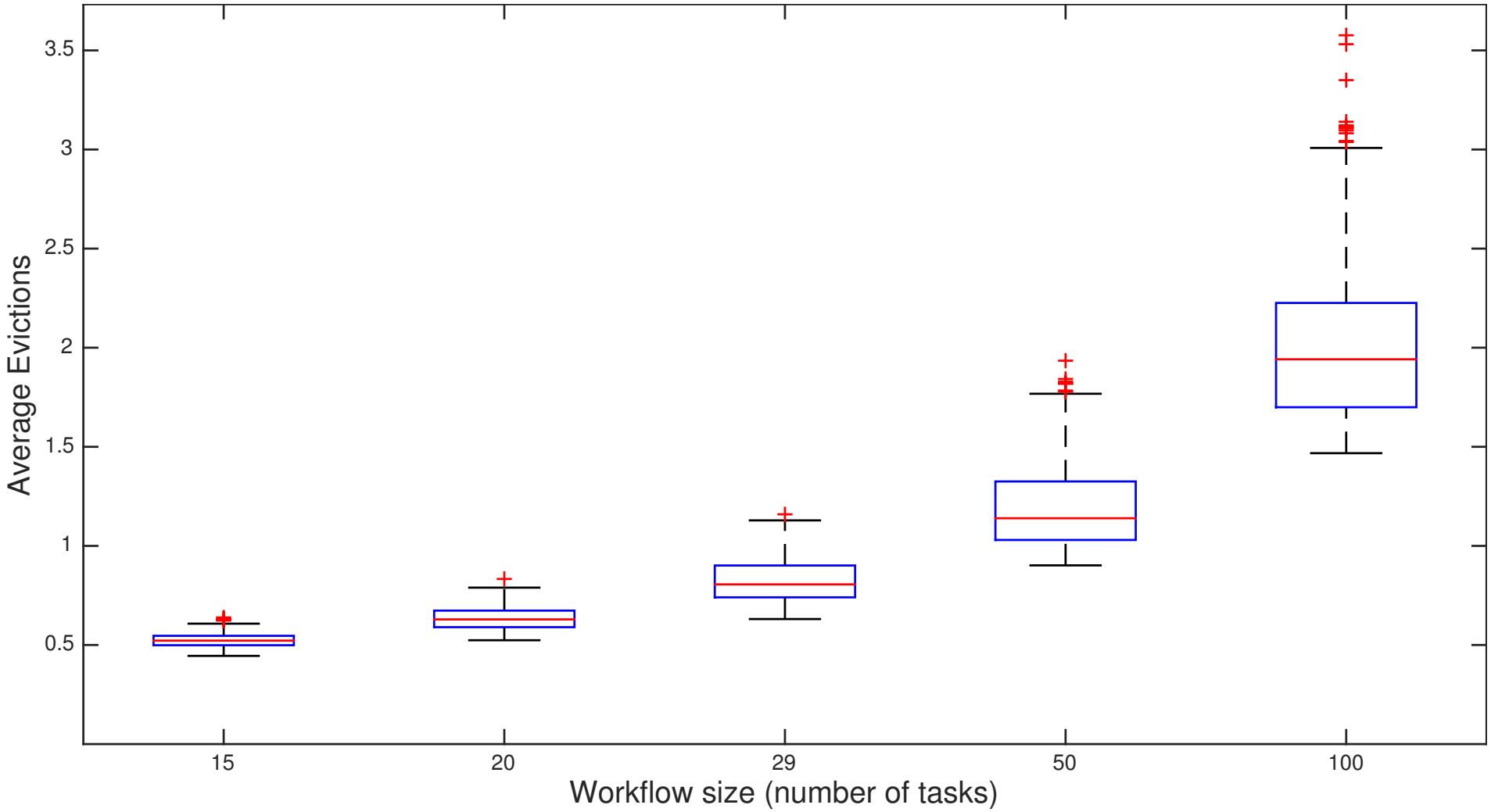
Parallelism achieved

- Vary workflow size

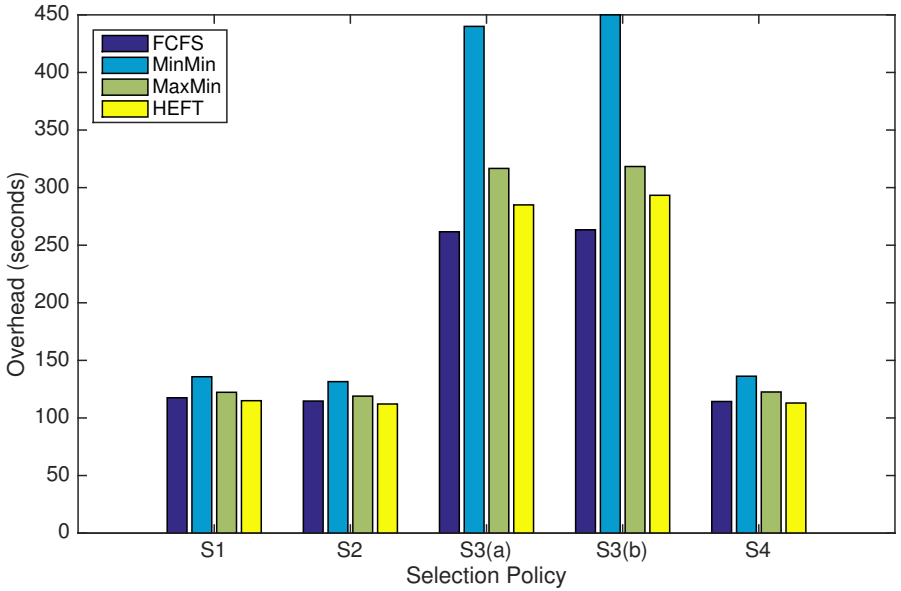


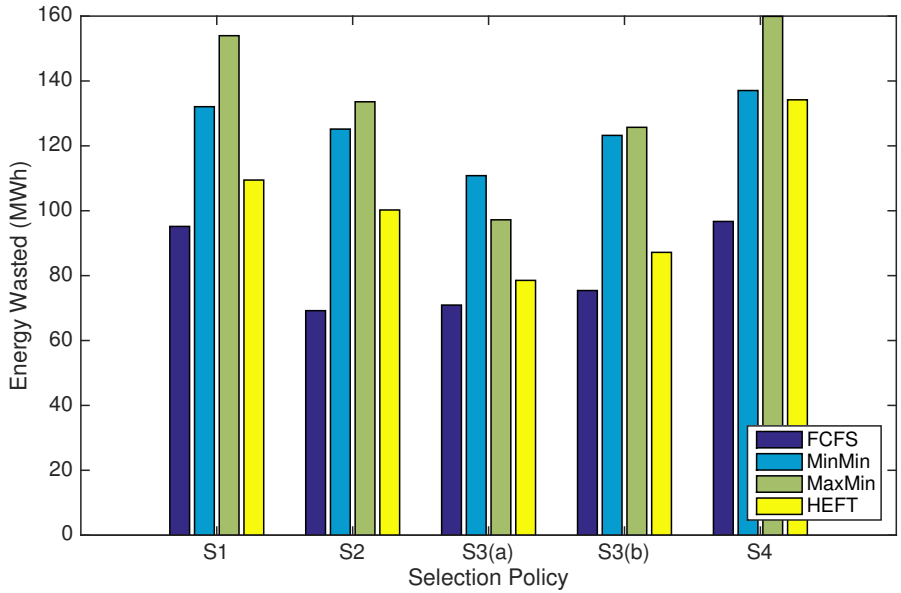
Workflow size and Evictions

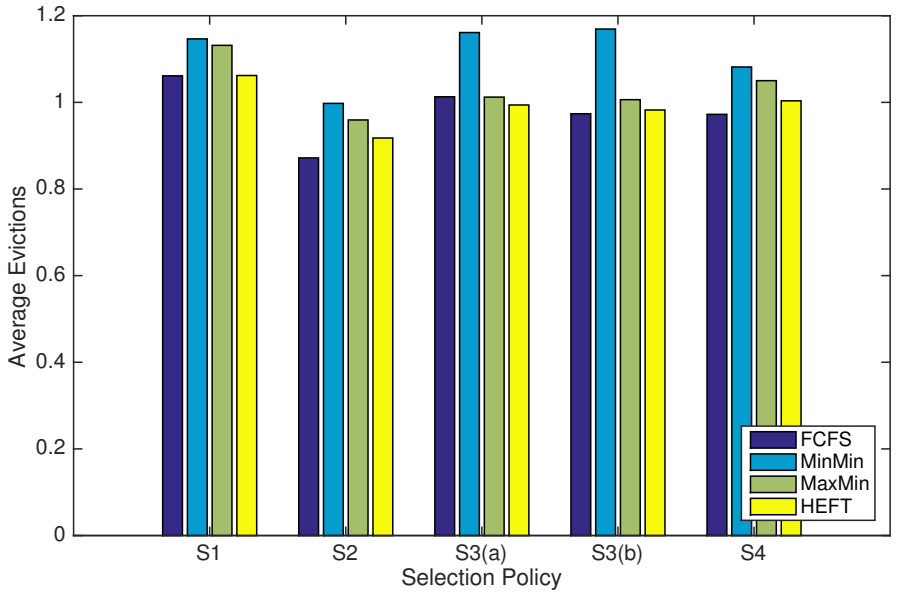
- Vary workflow size



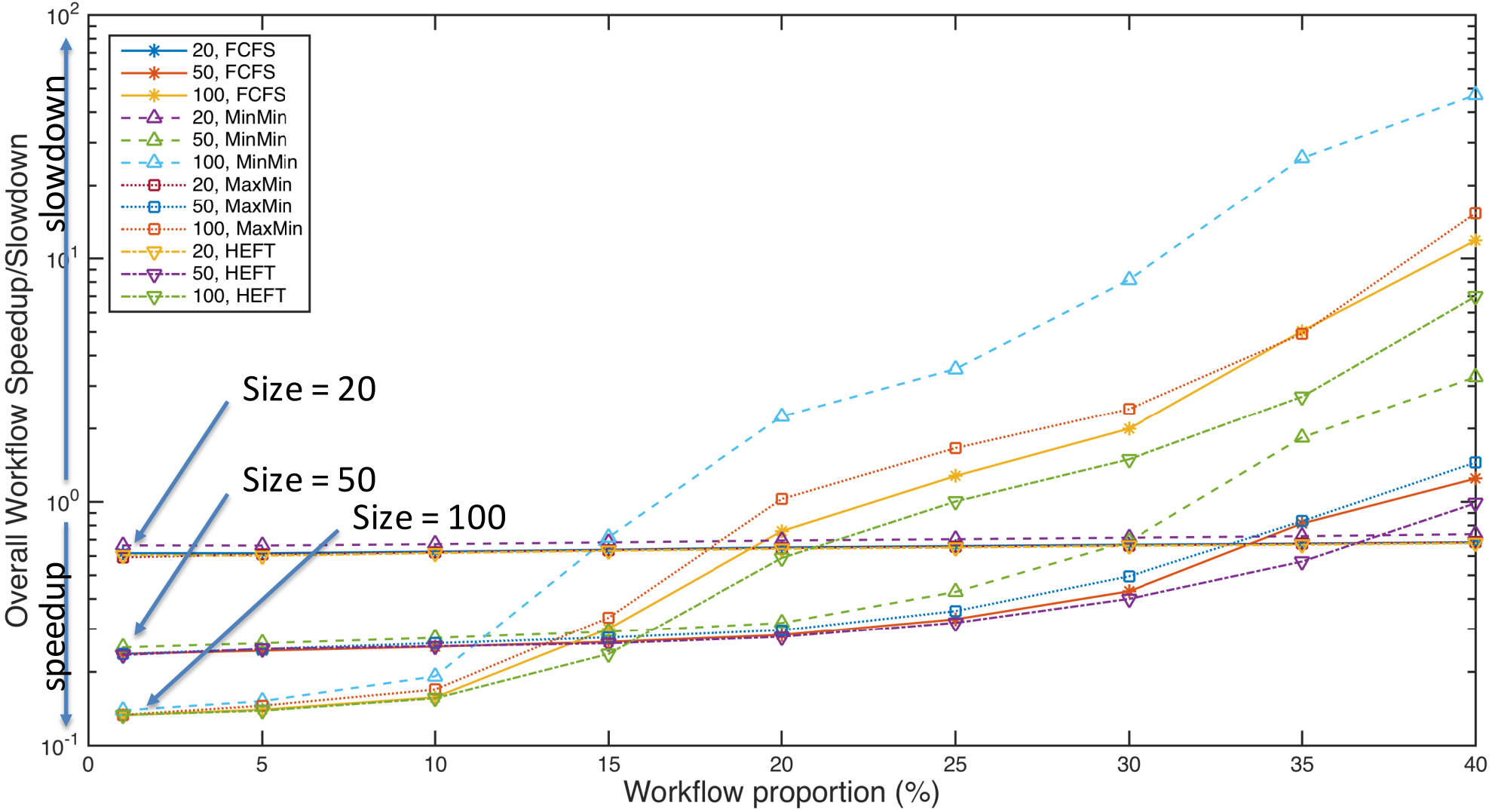
Impact on Energy, Overhead and Evictions







Workflow Speedup / Slowdown



Conclusion

- Extensions to HTC-Sim allowing us to run workflows across a non-dedicated set of resources
- With a focus on energy consumption of the system and overheads seen by the user
- Best results come with: HEFT and 'bucketing'
  - MinMin poor when resources aren't dedicated
- Can now use this to evaluate other scheduling strategies
  - Reservations, Check-pointing

stephen.mcgough@durham.ac.uk m.j.forshaw@ncl.ac.uk