Trace-driven simulation for energy consumption in High Throughput Computing systems

Stephen McGough

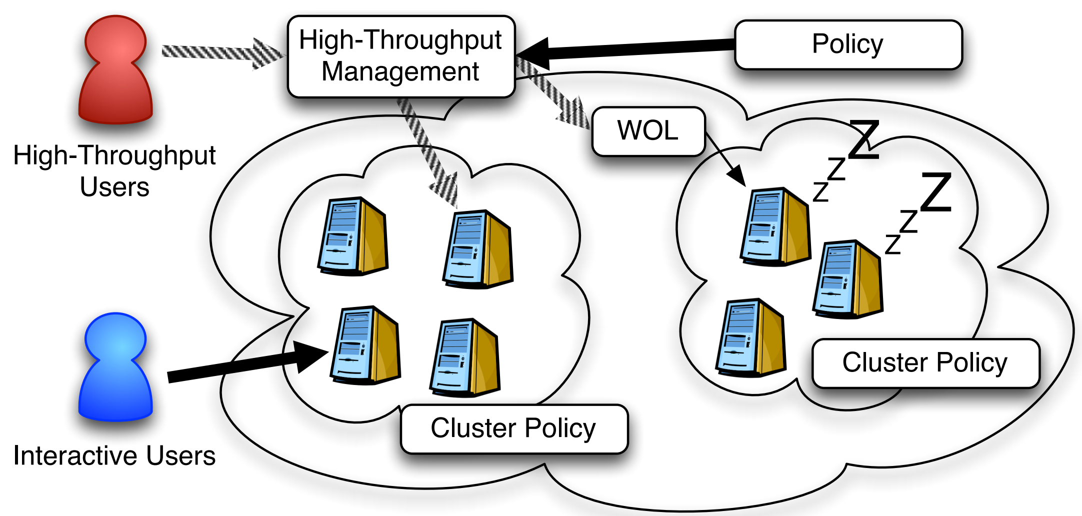
Durham University

Matthew Forshaw, Nigel Thomas

Newcastle University

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 (tasks)
  - Often used to exploit existing computing facilities
  - Resilient architecture
    - If a task fails to complete on one resource it will be reallocated to a different resource

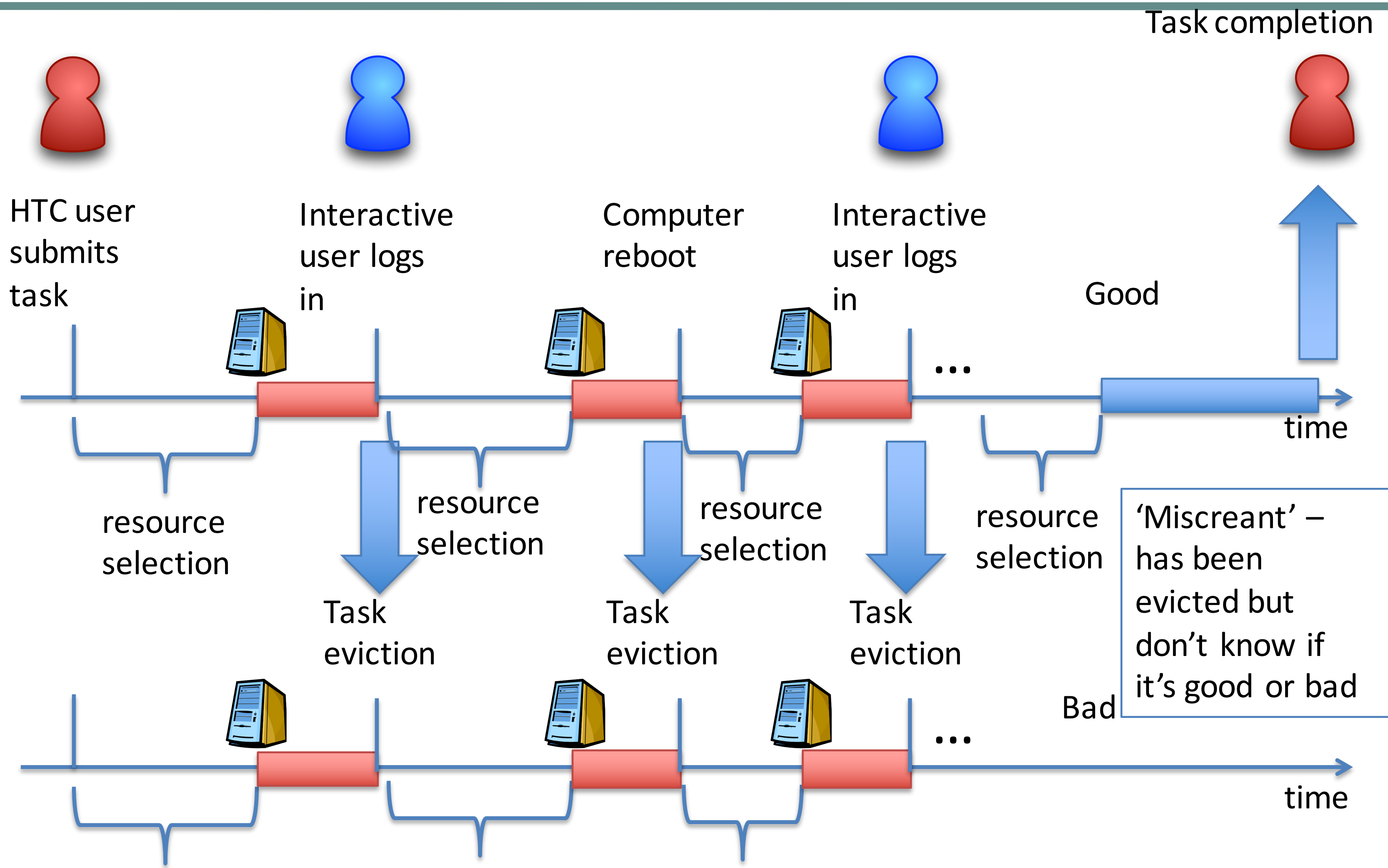


Dedicated

Cycle stealing

HTC

Task lifecycle

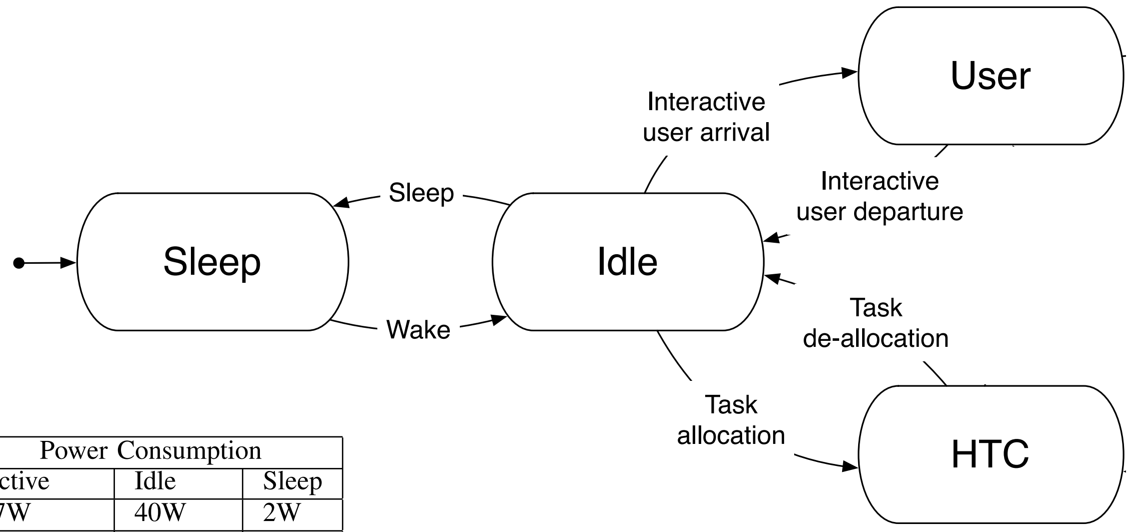


Motivation

- We have run a high-throughput cluster for ~6 years
  - Allowing many researchers to perform more work quicker
- Newcastle University has strong desire to reduce energy consumption and reduce CO_2 production
  - Currently powering down computer & buying low power PCs
  - "If a computer is not 'working' it should be powered down"
- Can we go further to reduce wasted energy?
  - Reduce time computers spend running work which does not complete
  - Prevent re-submission of 'bad' jobs
  - Reduce the number of resubmissions for 'good' jobs
- Aims
  - Investigate policy for reducing energy consumption
  - Determine the impact on high-throughput users

Cluster Simulation

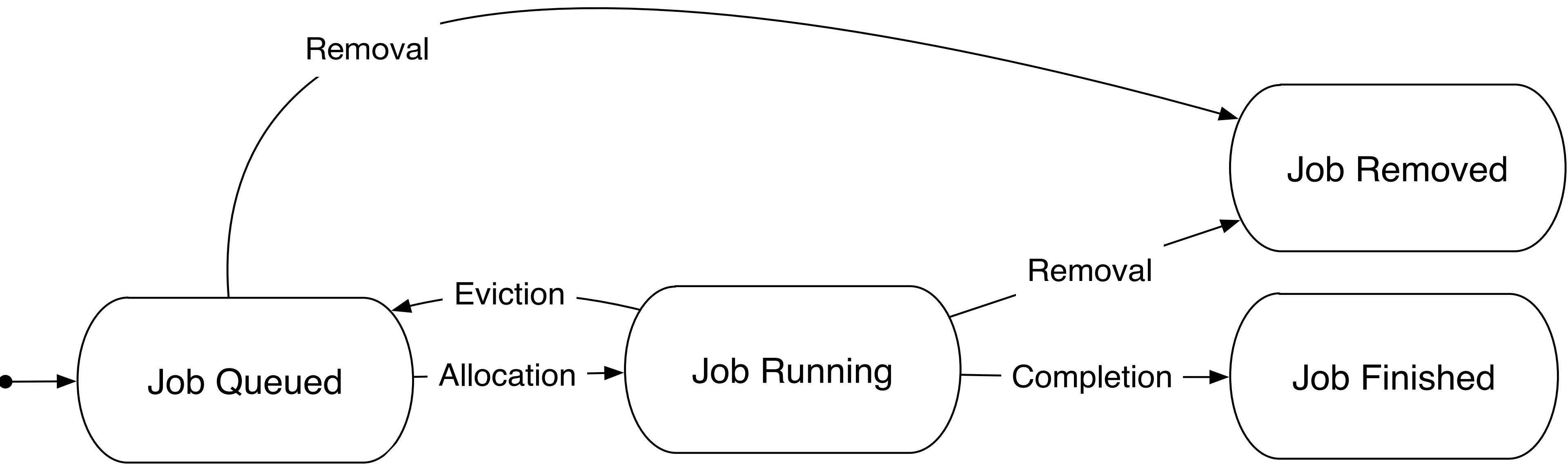
- High Level Simulation of a HTC system
  - Trace logs from a twelve month period are used as input
    - User Logins / Logouts (computer used)
    - Condor Job Submission times ('good'/'bad' and duration)



Type	Cores	Speed	Power Consumption		
			Active	Idle	Sleep
Normal	2	~3Ghz	57W	40W	2W
High End	4	~3Ghz	114W	67W	3W
Legacy	2	~2Ghz	100-180W	50-80W	4W

Cluster Simulation

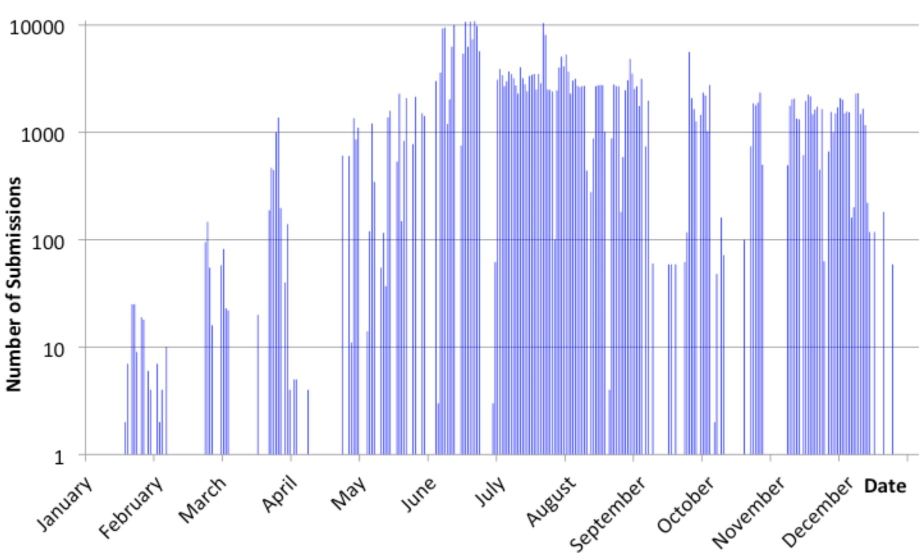
- Jobs can be in many states
  - Each having energy and performance impacts



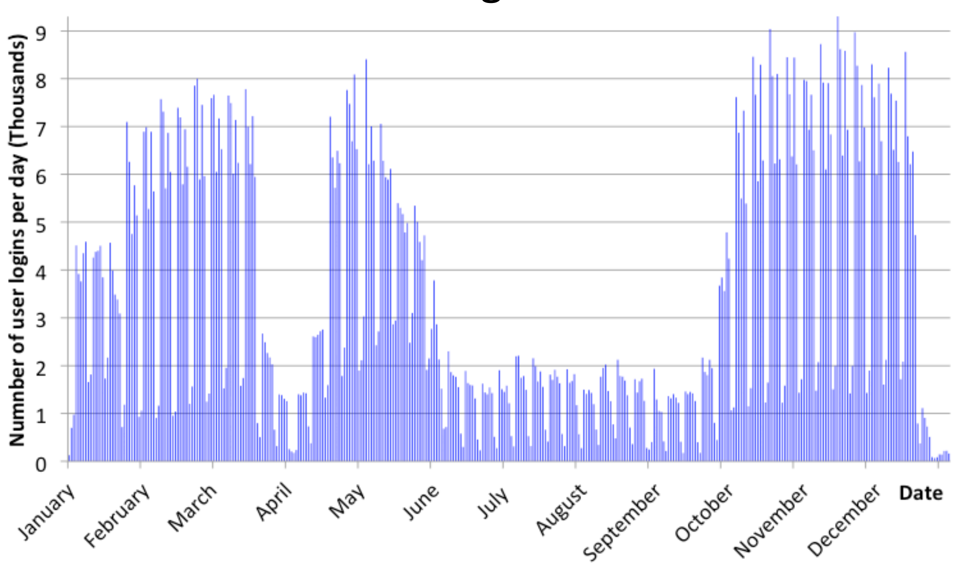
Condor At Newcastle

- Comprises of ~1300 open-access computers based around campus in 35 'clusters'
- All computers at least dual core, moving to quad / 8 core

Job Submissions



User Logins



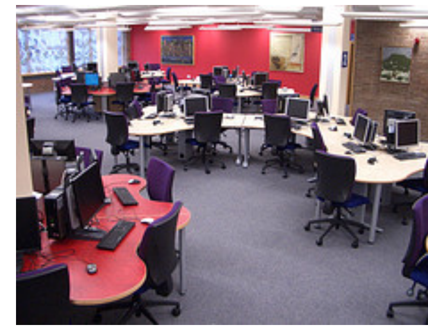
Locations

Old Library

Basement Cluster room
Needs heating all year (offset heat from computers against room heating) (Average idle time between users < 5 hours)

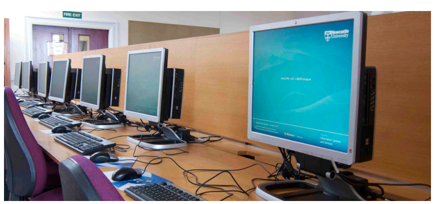
Robinson Library

Very high turnover and usage of computers room is hot and sunny (Average idle time between users < 2 hours)



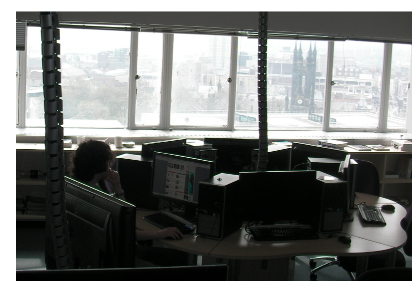
School of Chemistry (Chart)

Very low usage of Computers (Average idle time between users ~23 hours)



MSc Computing Cluster

South facing cluster room in High tower. (needs air-con all year) (Average idle time between users < 8 hours)

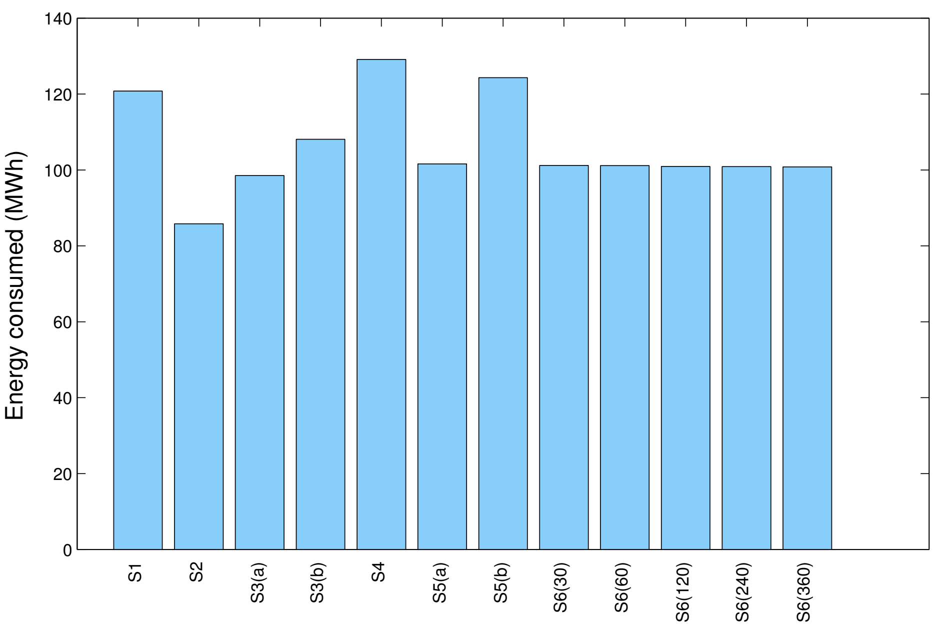


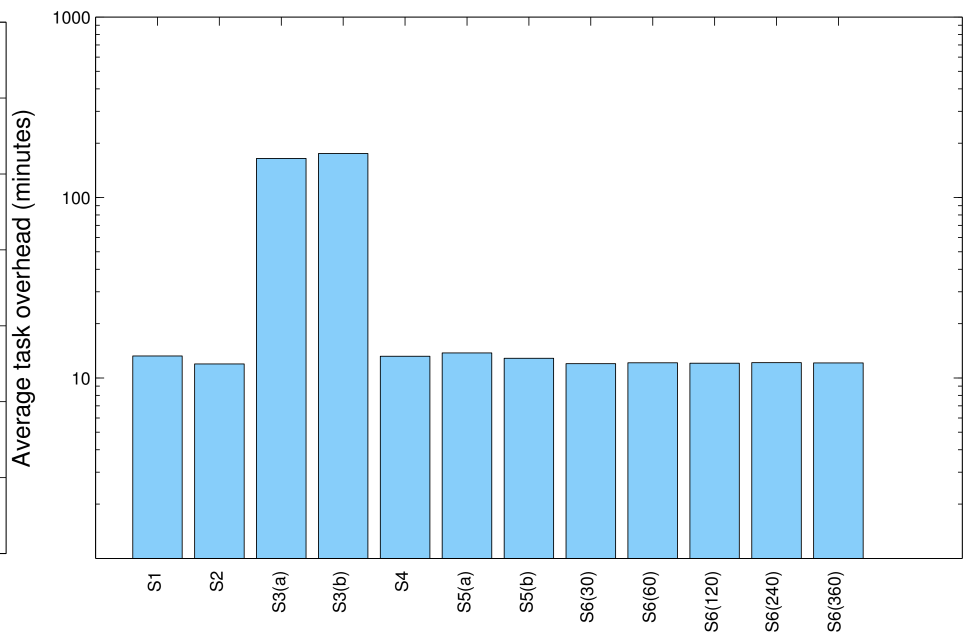
Policies For Saving Energy

- Selection of computer
- Started with simple Heuristics
  - S1: Random
  - S2: Most energy efficient computer
  - S3: Least interactive user activity
  - S4: Target closed clusters
  - S5: Less-used clusters
- More recent Heuristics
  - S6: Most likely to be idle computer based on monitoring of user activity over a window of recent activity

Policies For Saving Energy

- Can reduce energy consumption
  - By about 30%
  - Without significant impact on overheads
- But can we do better?



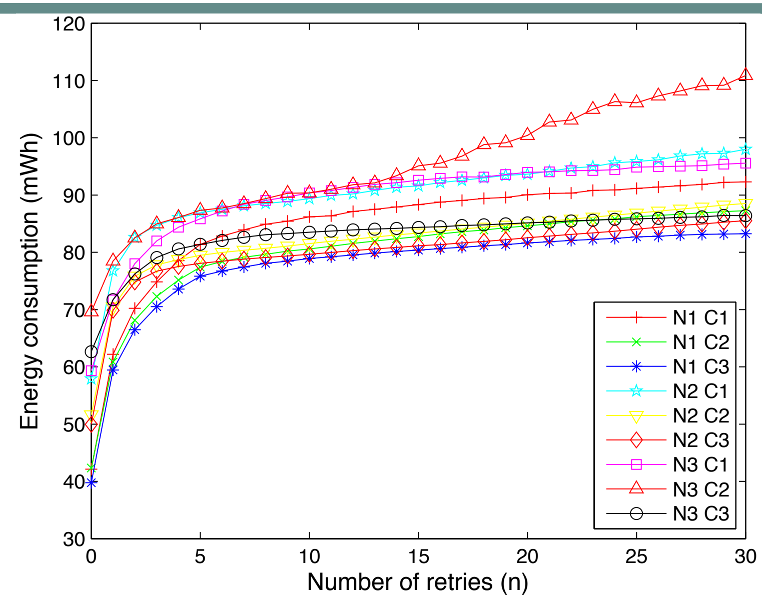


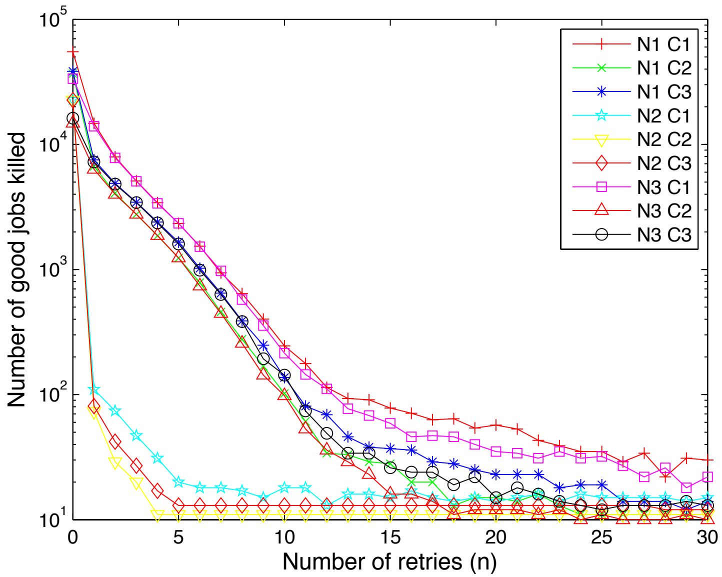
n reallocation policies

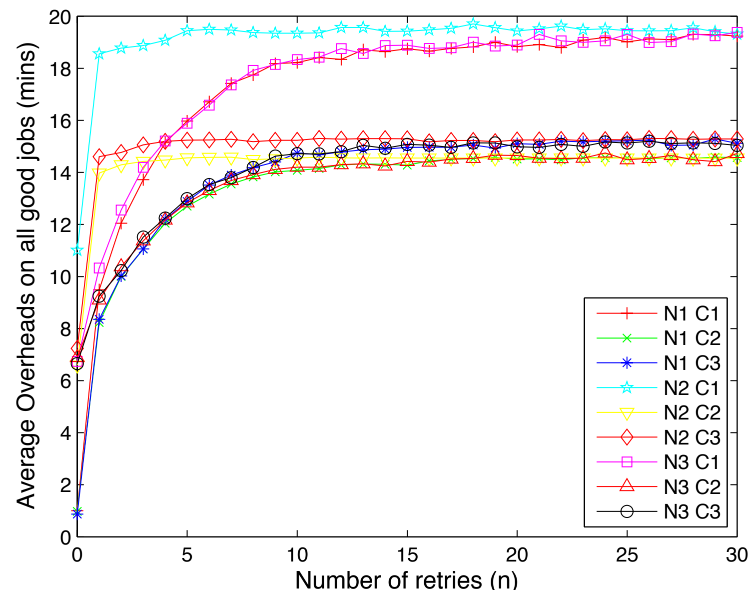
- Stop trying tasks after a number of resubmission attempts
  - **N1(*n*)**: Abandon task if deallocated *n* times.
  - **N2(*n*)**: Abandon task if deallocated *n* times ignoring interactive users.
  - **N3(*n*)**: Abandon task if deallocated *n* times ignoring planned machine reboots.
  - **C1**: Tasks allocated to resources at random, favouring awake resources
  - **C2**: Target less used computers (longer idle times)
  - **C3**: Tasks are allocated to computers in clusters with least amount of time used by interactive users

n reallocation policies

- Best policy N2 abandon after n retries ignoring user based evictions
- Energy saved ~37%
- But now we can have many 'good' jobs which are killed due to bad luck
  - Can still run all good jobs by having dedicated resources
  - Brings energy saving back to 30%
- Can we do better?







Reinforcement Learning

- Use Reinforcement learning to identify best resources to use
  - Or not to run a job at all
- Can save between 30% and 53% of the energy
  - 53% by doubling overhead
- No good jobs lost

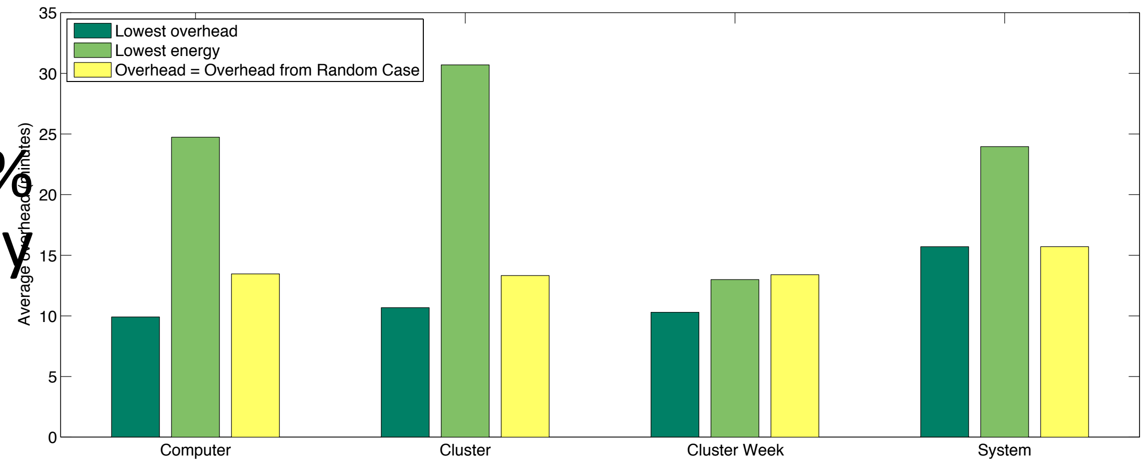
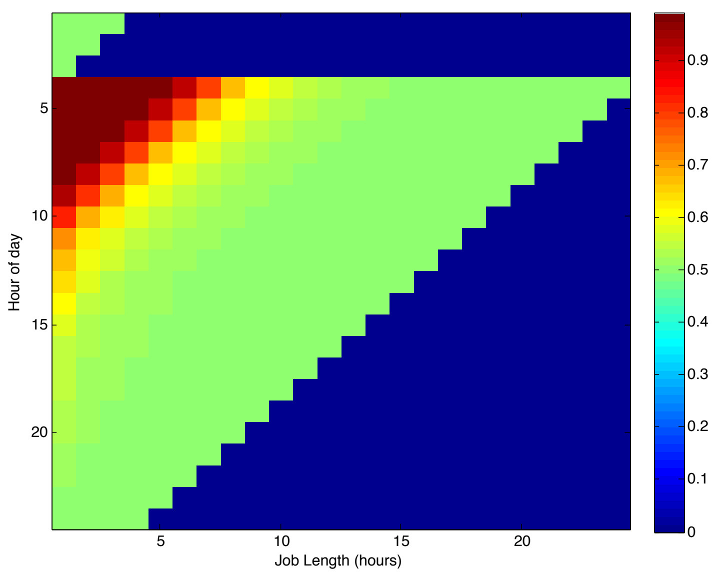
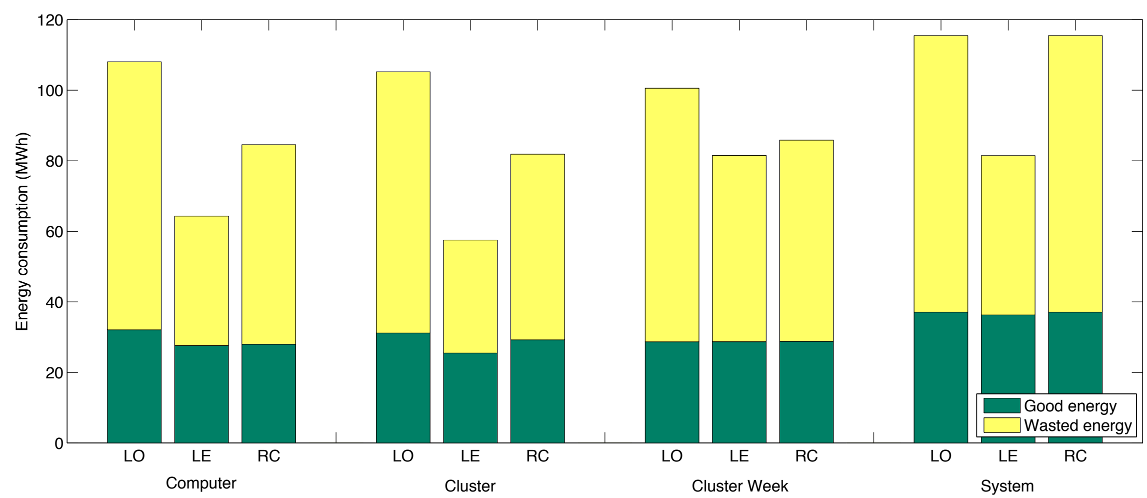


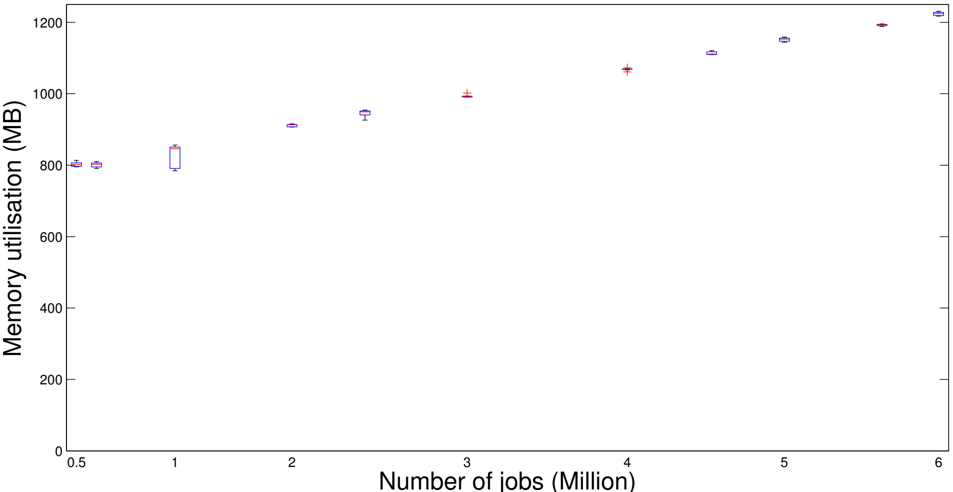
Figure 12: Comparison of the overheads for the different RL approaches

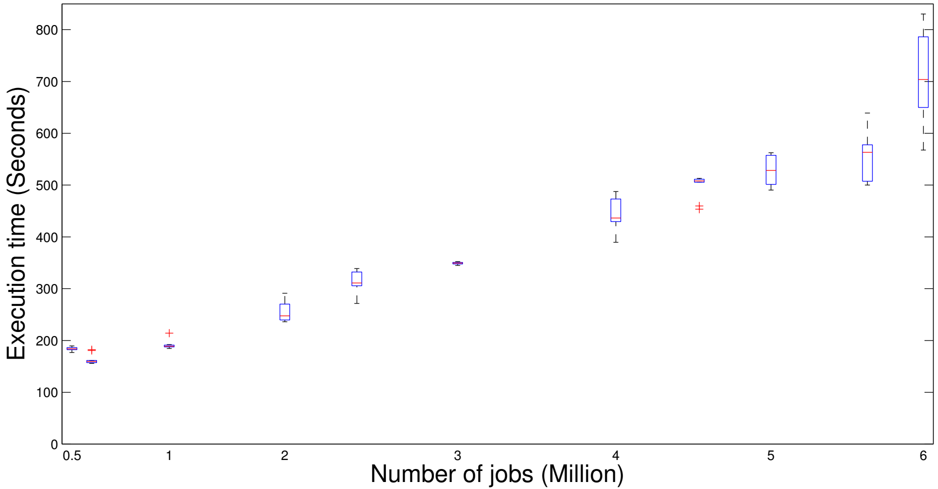




Scalability of the Simulation

- Simulation performance is linear with increase in number of jobs
  - Slight increase at ~6M jobs
  - Consequence of memory allocation





Conclusion

- HTC-Sim is a comprehensive simulator for HTC workloads on shared and dedicated resources
- With a focus on energy consumption of the system and overheads seen by the user
- Scales linearly with workload
- Future direction -> Cloud
  - We have a simple version for cloud cost
  - Cloud energy

Questions?

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