



Digital Institute

Making HTCondor Energy Efficient by identifying miscreant jobs

Stephen McGough, Matthew Forshaw

& Clive Gerrard

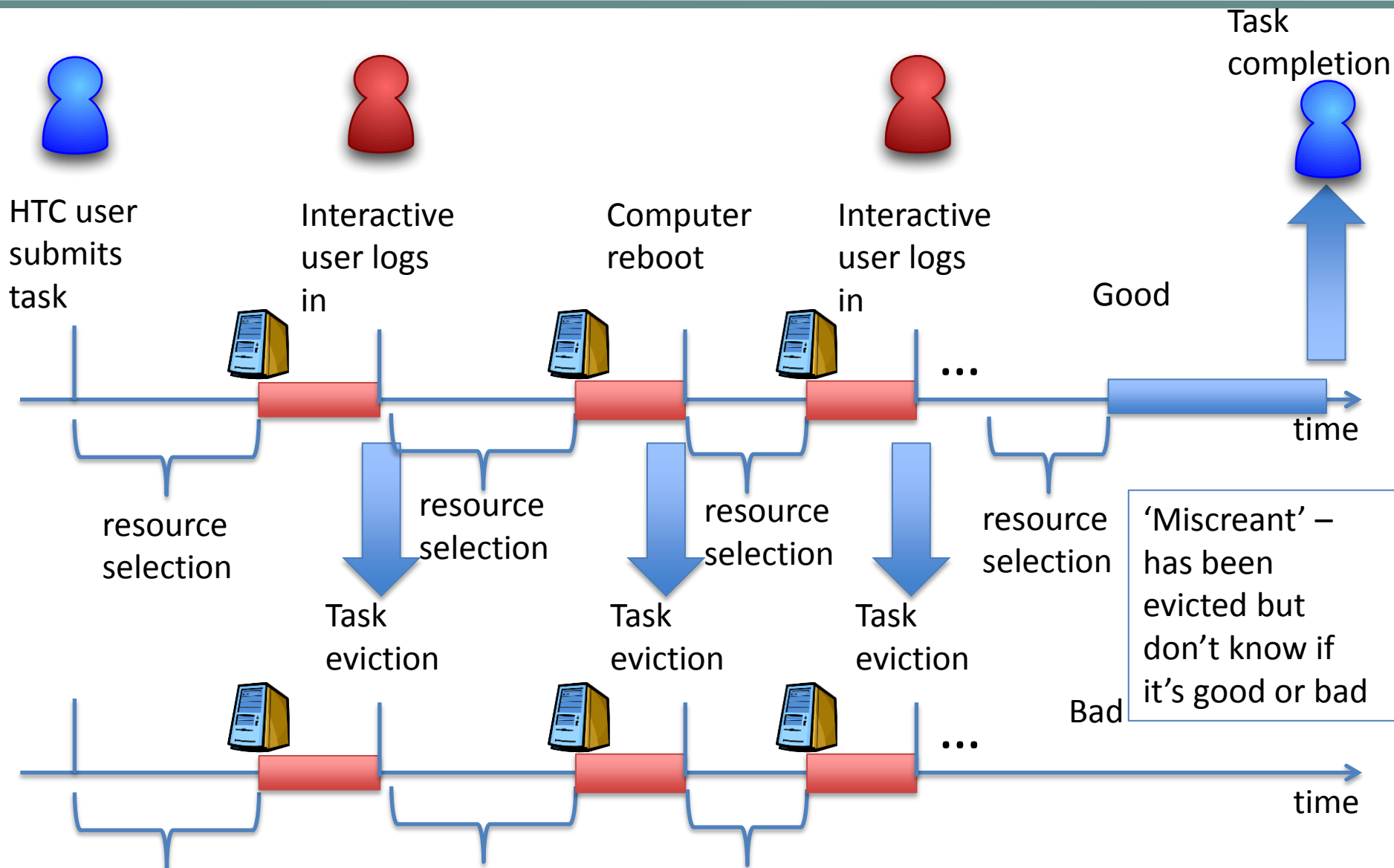
Newcastle University

Stuart Wheeler

Arjuna Technologies Limited



Task lifecycle

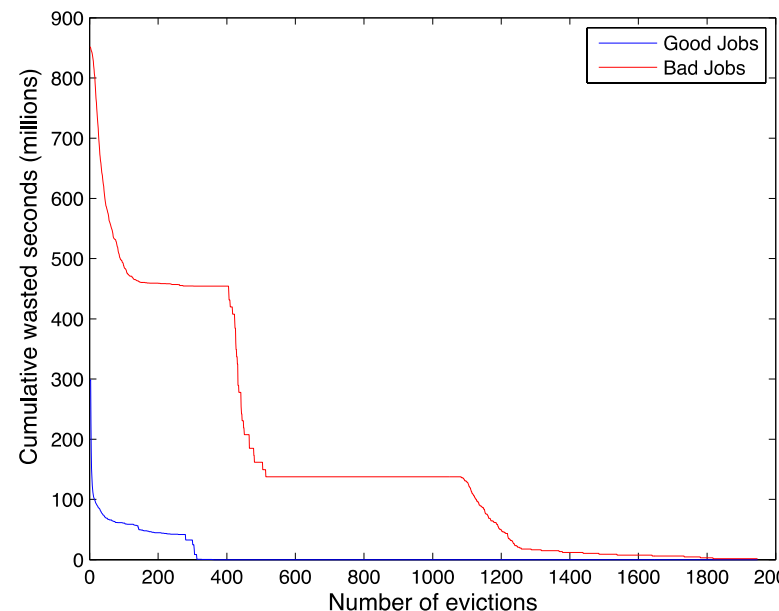
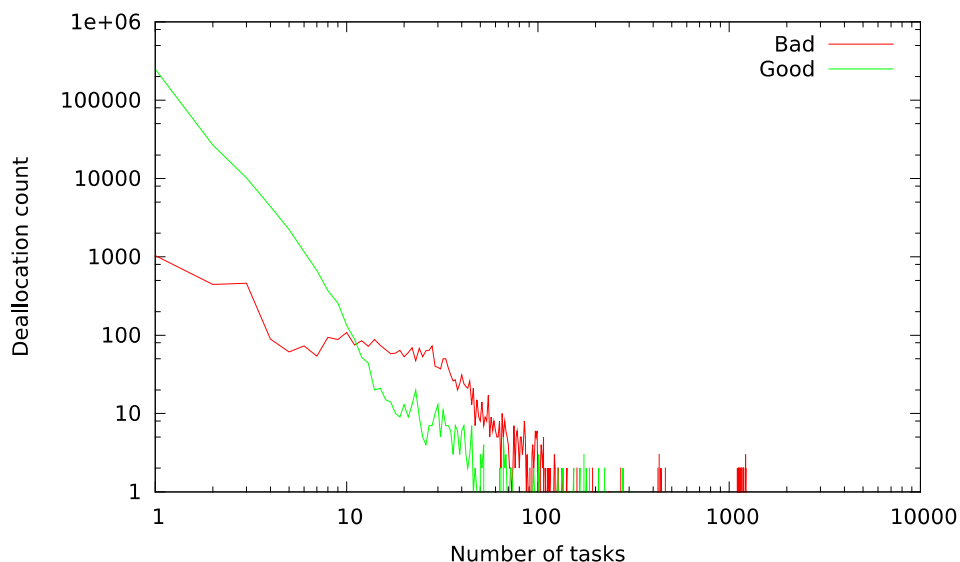


Motivation

- We have run a high-throughput cluster for ~6 years
 - Allowing many researchers to perform more work quicker
- University has strong desire to reduce energy consumption and reduce CO₂ 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

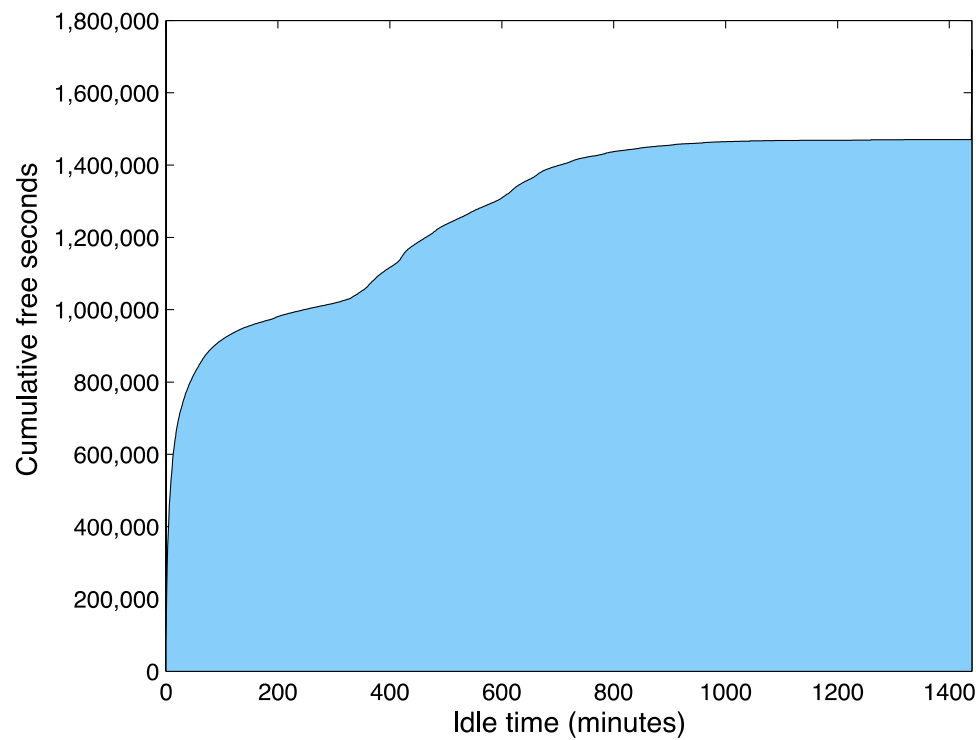
Can we fix the number of retries?

- ~57 years of computing time during 2010
- ~39 years of wasted time
 - ~27 years for 'bad' tasks : average 45 retries : max 1946 retries
 - ~12 years for 'good' tasks : average 1.38 retries : max 360 retries
- 100% 'good' task completion -> 360 retries
 - Still wastes ~13 years on 'bad' tasks
 - 95% 'good' task completion -> 3 retries : 9,808 good tasks killed (3.32%)
 - 99% 'good' task completion -> 6 retries : 2,022 good tasks killed (0.68%)



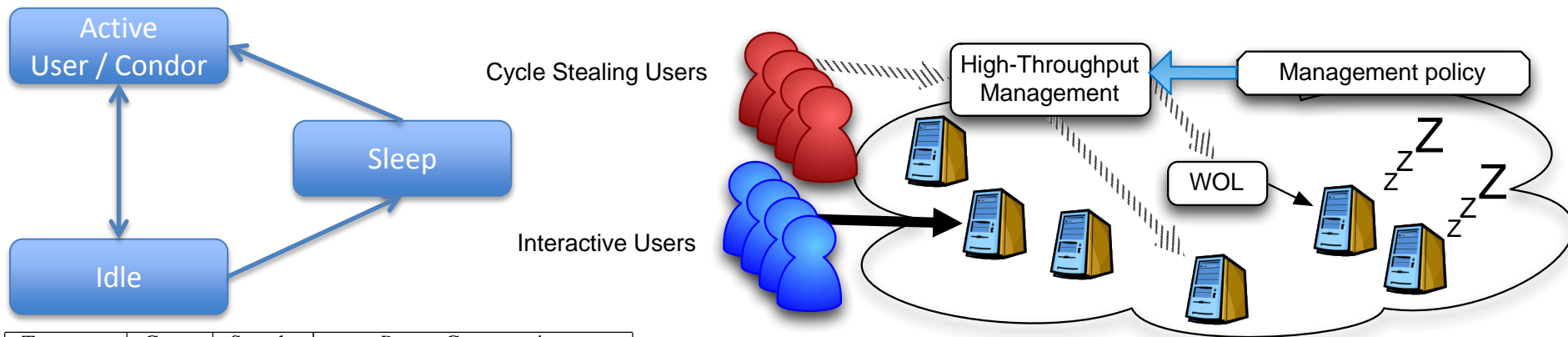
Can we make tasks short enough?

- Make tasks short enough to reduce miscreants
- Average idle interval – 371 minutes
- But to ensure availability of intervals
 - 95% : need to reduce time limit to 2 minutes
 - 99% : need to reduce time limit to 1 minute
- Impractical to make tasks this short



Cluster Simulation

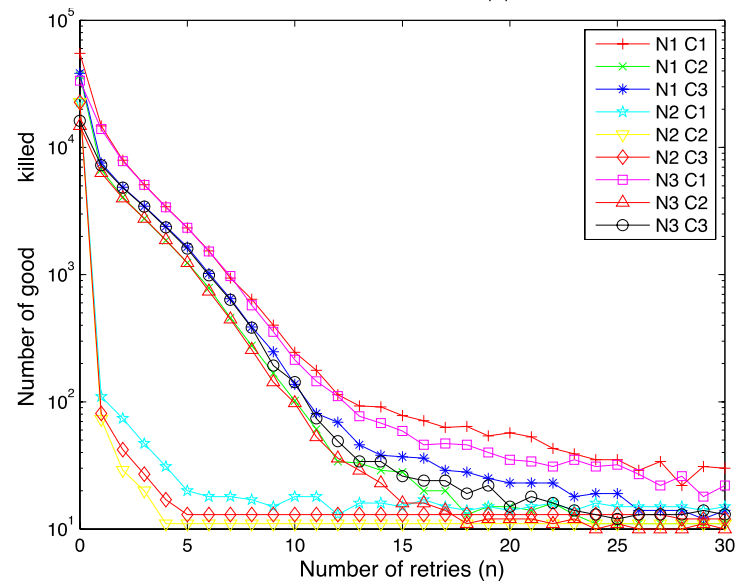
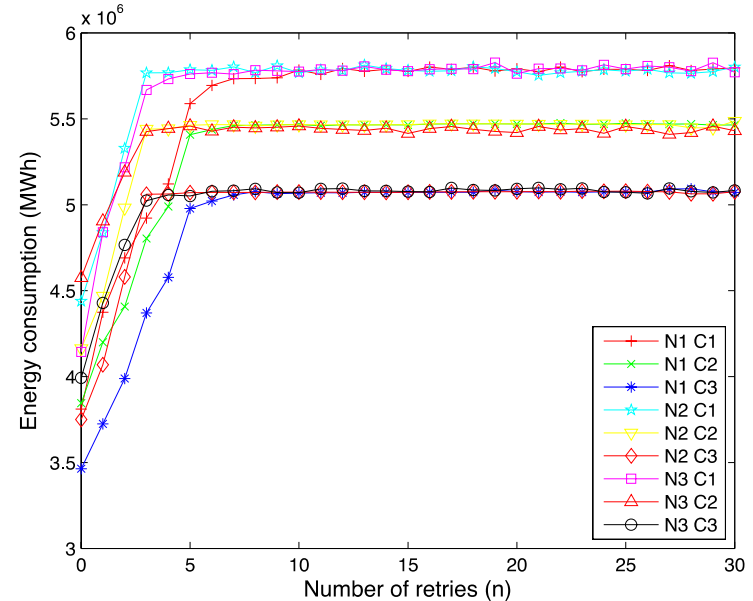
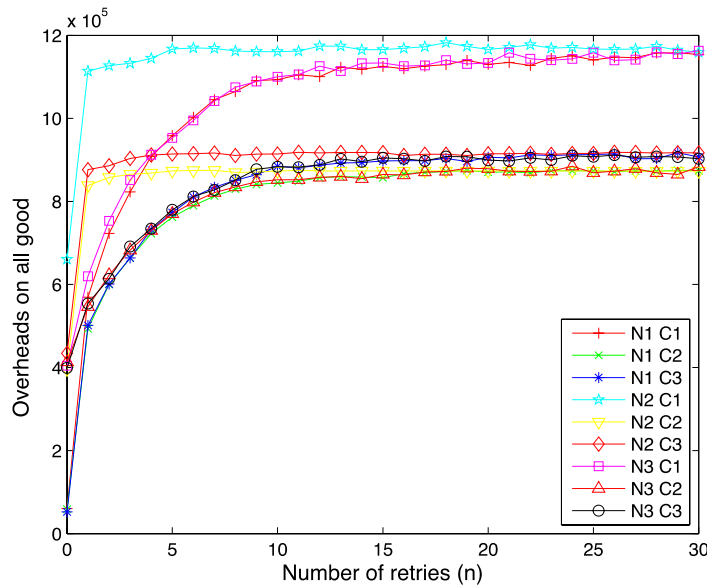
- High Level Simulation of Condor
 - Trace logs from a twelve month period are used as input
 - User Logins / Logouts (computer used)
 - Condor Job Submission times ('good'/'bad' and duration)



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

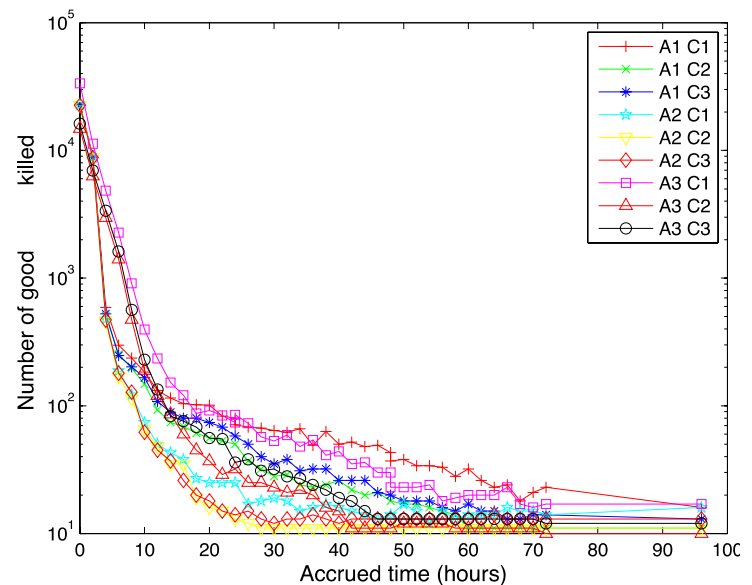
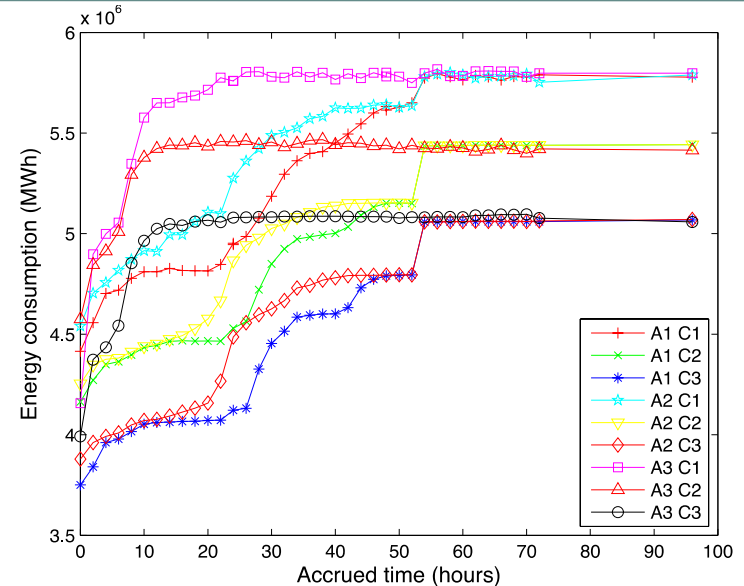
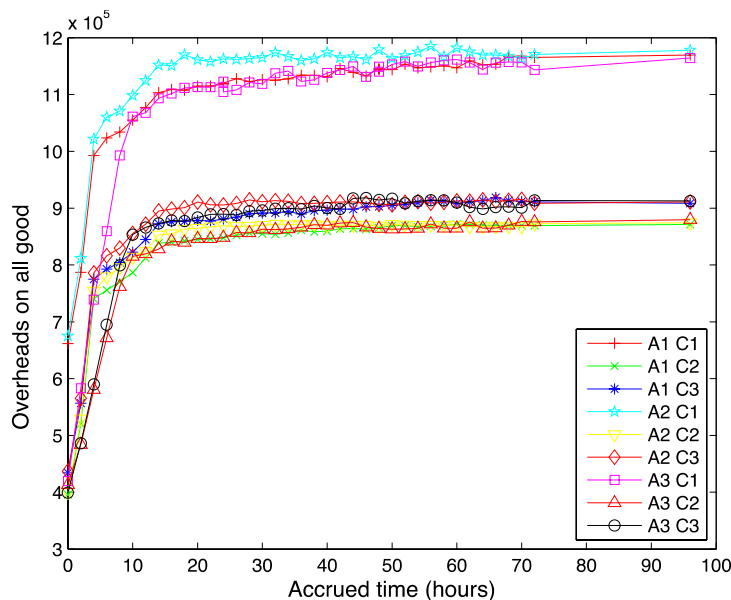
n reallocation policies

- **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



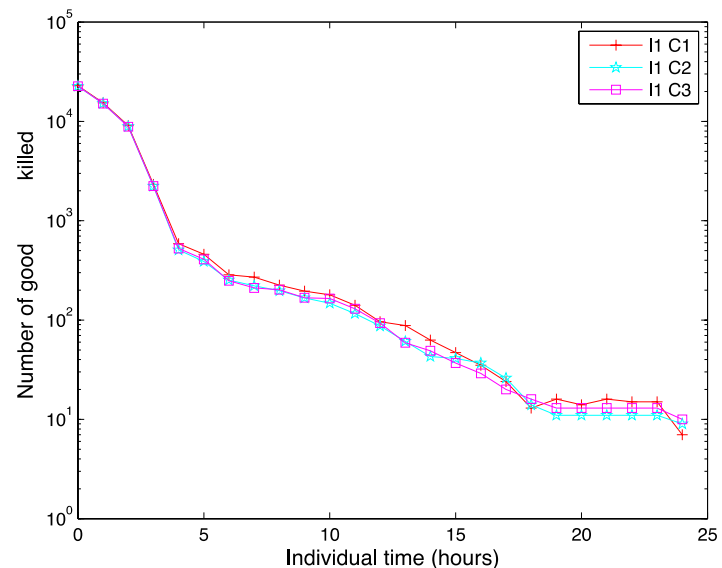
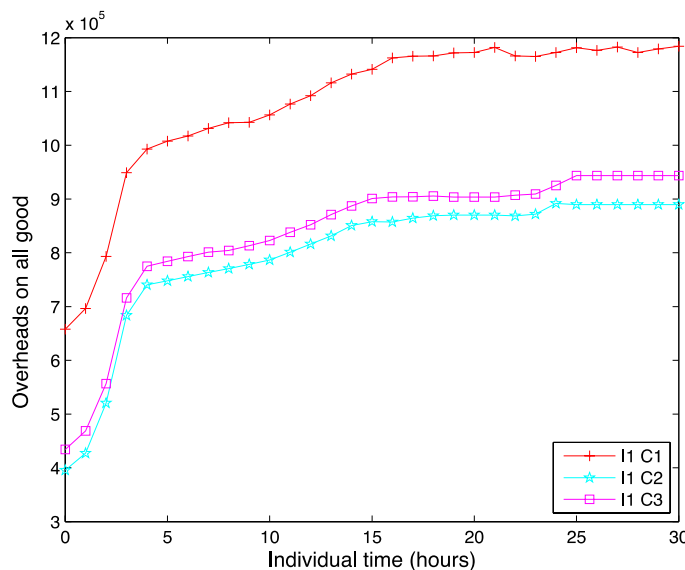
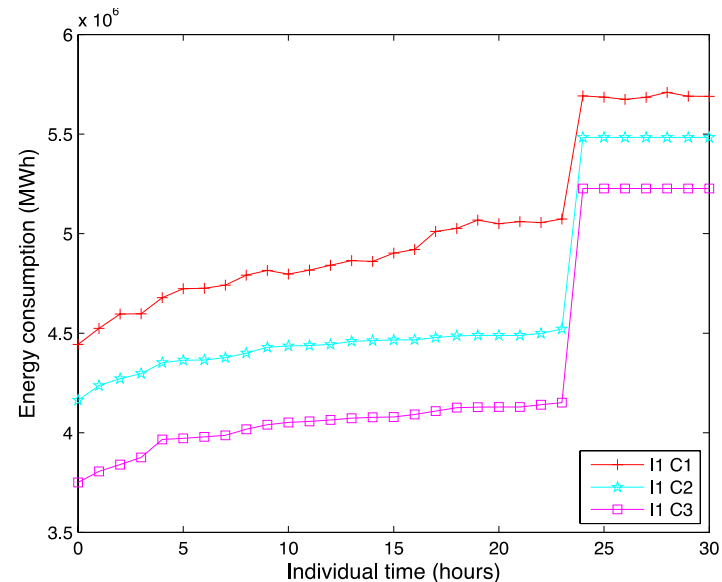
Accrued Time Policies

- Impose a limit on cumulative execution time for a task.
- **A1(t)**: Abandon if accrued time $> t$ and task deallocated.
- **A2(t)**: As A1, discounting interactive users..
- **A3(t)**: As A1, discounting reboots.



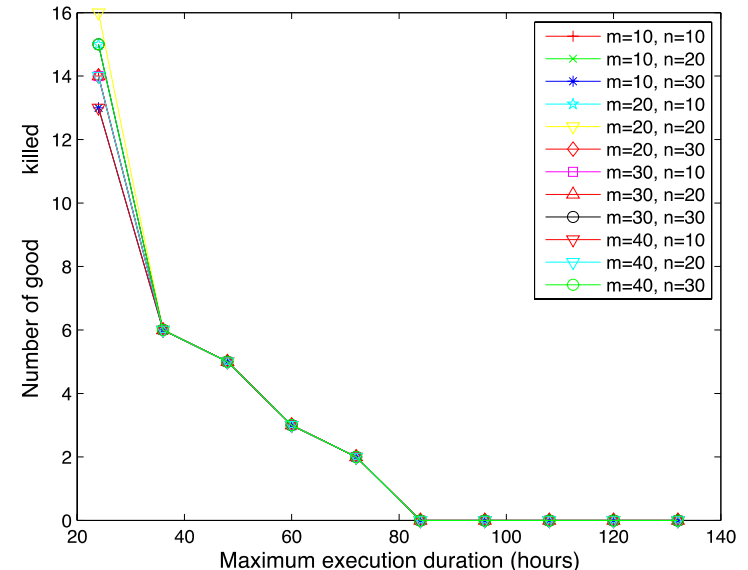
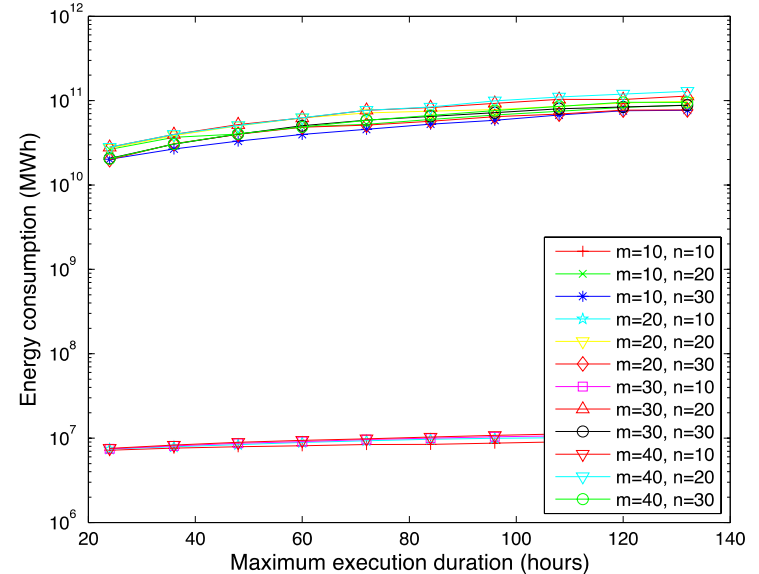
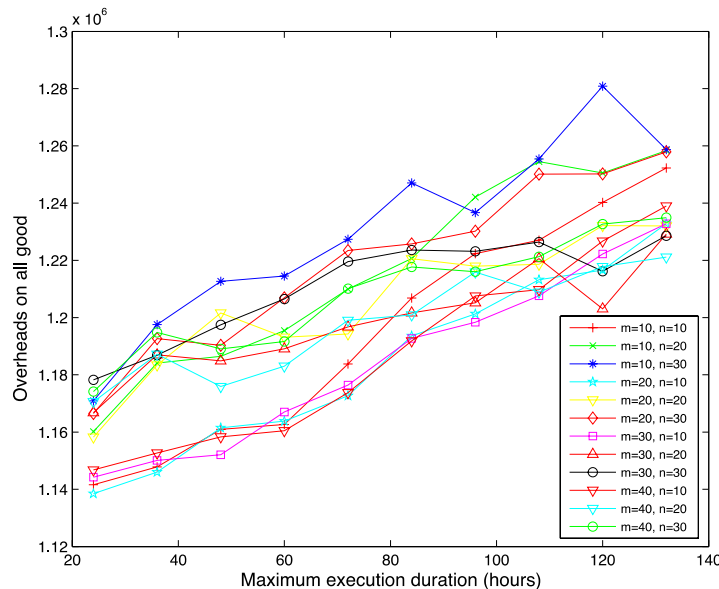
Individual Time Policy

- Impose a limit on individual execution time for a task.
 - Nightly reboots bound this to 24 hours.
 - What is the impact of lowering this?
- $I1(t)$: Abandon if individual time $> t$.



Dedicated Resources

D1(m, d): Miscreant tasks are permitted to continue executing on a dedicated set of m resources (without interactive users or reboots), with a maximum duration d .



Conclusion

- Simple policies can be used to reduce the effect of miscreant tasks in a multi-use cycle stealing cluster.
 - N2 (total evictions ignoring users)
- Order of magnitude reduction in energy consumption
 - Reduce amount of effort wasted on tasks that will never complete
- Policies may be combined to achieve further improvements.
 - Adding in dedicated computers

Questions?

stephen.mcgough@ncl.ac.uk

m.j.forshaw@ncl.ac.uk

More info:

McGough, A Stephen; Forshaw, Matthew; Gerrard, Clive; Wheeler, Stuart;
Reducing the Number of Miscreant Tasks Executions in a Multi-use Cluster,
Cloud and Green Computing (CGC), 2012