Service Level Agreement Based Scheduling

We describe a recently funded project that aims to establish a fundamental new infrastructure for efficient job scheduling on the Grid, based on Service Level Agreements (SLAs). These are negotiated between the client (user, superscheduler, or broker) and the scheduler, contain information on acceptable job start and end times, and may be re-negotiated during runtime.

Motivation

The Grid community wants to be able to schedule complex workflows onto compute resources which have their own schedulers. Obviously, dependences between parts of the workflow have to be honoured. Also, the user will want to have some assurances about time to completion. This means that it is not practical to schedule these workflows a piece at a time, waiting until one component comes back to start thinking about scheduling the next component. This technique also produces inconsistent overall execution times for the workflow, this being dependent on the queuing time of each workflow component.

The prediction problem is exacerbated by the fact that the underlying schedulers on compute resources, e.g. batch queue systems, do not provide assurances about when a specific job will run. Jobs can wait in a queue for an unpredictable amount of time. It would be far more efficient if the compute jobs of a workflow could be scheduled more accurately, hiding any such latencies.

Currently Available Solutions

Currently, the only existing mechanism for increasing predictability of execution time is Advance Reservation (AR). This forces the start time of the job to be fixed precisely. Often, we're only interested in fixing bounds (soonest start and latest end time), so AR can be a sledgehammer being used to crack a nut.

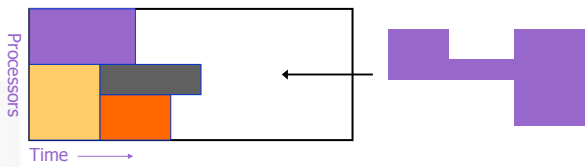
Also, advance reservation doesn't fit very well into the batch processing model. It causes expensive gaps in the schedule that can't be effectively plugged. This is often attempted with checkpointing (especially by the OS) which takes a lot of time, e.g. 12 minutes for a 64 processor Unified Weather Model job, with a memory footprint of only 3GB; at Manchester, some capability jobs have footprints of 300GB. Suspend/Resume is a quicker alternative, but impacts available resources and/or performance of the incoming AR job (due to the cost of swapping out the suspended job). These gaps cause machine utilisation to decrease rapidly (worse than linear) as the percentage of AR jobs in the mix increases.

We believe that the central problem is that batch schedulers only offer two levels of service: **run this when it gets to the head of the queue**; and **run this at a specific time**.

Our Approach

We propose that Service Level Agreements (or SLAs) need to be pushed from the Grid middleware, down into the batch schedulers. We will develop new scheduling algorithms for a scheduler which has an SLA associated with each and every job; these SLAs are negotiated at the point of job submission. The schedule will be calculated based upon these SLAs. There is no queue, and jobs don't have a "priority". The current set of SLAs will determine what new SLAs the scheduler can commit to – saying "no" must be an option.

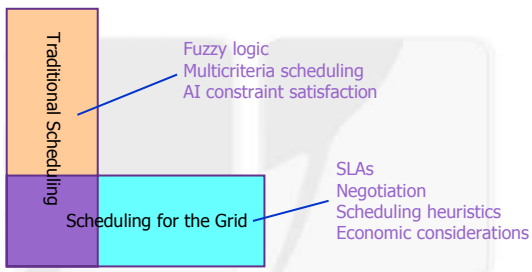
To make things more complicated, we will address requirements from the UK e-Science Pilot Project, RealityGrid, which calls for re-negotiation of compute resources **at runtime**. Introducing renegotiation makes the problem far more complex and dynamic.



"Breaking the Box" with renegotiable resource requirements

Combining Techniques

The scheduling problem no longer looks like a batch scheduling problem. It looks more like a "traditional scheduling" optimisation problem, e.g. optimising workflow in a factory. So in addition to applying compute scheduling techniques, we also need the traditional scheduling community, and their techniques. So this project will be addressed by two groups, the University of Manchester, with our Grid computing expertise, and the ASAP "traditional scheduling" group from Nottingham, both of whom are world leaders.



Our solution lies in an intersection of two research areas...

The University of Nottingham

asap automated scheduling optimisation & planning

the Inter-disciplinary Scheduling Network