I/O-bound and CPU-bound tagging

Abstract

Sites have used general purpose batch queues (LRMS) for many years to provide highest throughput and fairest allocation of resources between the communities that they serve. Many sites have also expressed interest with scheduling based on Resource Constraints, particularly storage, we should like to give sites more options to schedule resources as they assert that higher throughput could be achived if jobs requirements are known better to the LRMS scheduler.

Introduction

I/O throughput on sites storage systems have an optimum Load, and global throughput can be reduced when a load above a certain threshold is reached. Excessive Storage load could also lead to Storage system instability. For example a site my decide that by spreading users jobs on different nodes, the start up time is more staggered and so preventing resource contention on a single file. The introduction of pilot jobs has prevented scheduling decisions based on user name. Although user name only gave an indirect interpretation of job load, I/O-bound and CPU-bound tagging intends to better scheduling based on user name, by giving more useful information to the LRMS.

Definitions

- LRMS or Local Resource Management Systems

LRMS are systems that schedule to execution of jobs in clusters, this covers both Cloud and Batch queue.

- VO or Virtual Organisation

A VO is a collection of scientists working in a related field doing similar processing.

- SE or Storage Element.

A SE is a storage resource usable by one or more VO's.

Assumptions

- Sites may support more than one VO.
- Pilot Jobs presented to the LRMS may not give any scheduling hints to LRMS except VO.
- Sites are aware of load on local site resources and would like to schedule jobs based on this knowledge.
- Some jobs use less IO resources than others (Monte-Carlo jobs V Analysis jobs)
- Sites can contribute to higher job efficiency by optimising scheduling of jobs from multiple VO communities.
- Sites Storage systems have an optimum throughput above this threshold of requests their aggregate throughput may reduce.
- DESY have found it beneficial to schedule user jobs per node, and fill the remaining slots with Monte-Carlo jobs.
- Grid worker nodes can support more CPU bound jobs than I/O bound jobs.
- LRMS have rich configuration for scheduling.
- Jobs submitted by the same user at the same time are often of the same type.
- Jobs of the same type submitted at the same time often access the same files at the same stage of processing.

Use cases

Identifier	Actors	Pre-conditions	Scenario	Outcome	(Optional) What to avoid
1	Site	IO load on SE is optimal or higher	Only Run more CPU bound jobs and no more IO bound jobs	SE remains at opimal throughput	SE does not overload
2	Site	Job Slots are free	Next 100 Jobs all want to access the same file	Other job types can be interleaved	All jobs accessing the same file at the same time
3	Site	SE is underused	Jobs with high IO load should be scheduled to run in preference.	SE is optimally loaded	

Discussion on Use cases.

The questions was raised, whether cpusets or affinities (see the relevant part of the WM TEG wiki) could solve this issue at a site level. It turns out these are quota tools, preventing one job from starving another job of resources; each works at a Single Worker Node level. We believe these techniques are complimentary to job tagging and do not remove the use case for it.

Requirements

Identifier	Originating Use Cases	Actor	Details
1	1,3	VO jobs submission service	Job information presented to the CE for jobs type
2	1,3	CE service	CE Job information presented to the LRMS for jobs type
3	1,3	Pilot job framework	Pilot jobs must run only jobs of the same type which where sent to CE to start pilot job

Impact

WMS and Computing elements

These Constraint Tags should be honored by the CE and passed to the LRMS in an agreed way, allowing sites to customize scheduling if they so desire. Sites that do not want to use to make use of Resource Constraint Tags will not need to use them.

Pilot job frameworks

Pilot frameworks should honour the Resource Constraints between the pilot and the jobs the pilot executes.

Negative Impact

We believe that the number pilot jobs will increase if their are more tags as pilots can only run jobs which match the requirements they where submitted with. For this reason we cannot extend the number of job tags indefinitely.

Positive Impact

We believe many sites will optimise job load to better reflect site resources, so making more efficient use of available resources.

Conclusions

The original proposal was to tag jobs and have a boolean list of constraints, i.e. a job could be flagged as either CPU or I/O bound.

Unfortunately time did not permit VO's tagging jobs by more abstract tags to be discussed. Such as that the Job was an "Analysis Job" or a "Monte-Carlo Job". This has the advantage that the VO knows this job type accurately, it has the disadvantage that these abstract tags would be very VO specific in terms of their IO or CPU load.

The WLMTEG decided that the proposed changes to the JDL to include flagging jobs as either CPU or I/O bound, should be more extensible. A weighted value between 0 and 1 (with 0 being CPU-bound and 1 I/O-bound); initially, just the values 0 and 1 will be used to gain experience with the idea. Further Tags may be added at a later date.

As knowledge of job characteristics improve, a VO can specify different values; at a later stage we may decide to define an exact metric. This parameter does not translate into a specific resource to be allocated to the job, but may help sites distributing the jobs in a convenient way. It is up to the experiments to decide which of their jobs will be tagged one way or the other.

There was a long discussion on the accuracy of constraint tagging. One first conclusion was that users adding constraint tagging is to be avoided, since users are commonly unaware of such details and inconstant in their actions. In general, accuracy in the scalar value was not seen as important, and sites must expect user communities will not be consistent in weighted values. On the other hand, experiments do know some jobs will be CPU or I/O bound, and tagging these jobs as such will help some sites scheduling them more efficiently. It is recognized that jobs often change constraints during execution, and precise measurement is not seen as easy.

Recommendations

We propose that Resource Constraint Tags should be agreed and added to the JDL as a set of weighted values between 0 and 1; we expect Resource Constraint Tags may not be limited to I/O and CPU bound constraints. We propose that Specifications should be agreed between sites and CE and VO developers. -- OwenSynge - 08-Mar-2012 -- OwenSynge - 02-Mar-2012 -- DavideSalomoni - 03-Feb-2012

This topic: LCG > WMTEGIOvsCPUjobs
Topic revision: r7 - 2012-03-08 - OwenMillingtonSyngeExCern