ACAT 2021

Contribution ID: **633** Contribution code: **contribution ID 633** Type: **Poster**

Adding multi-core support to the ALICE Grid Middleware

Abstract

The Large Hadron Collider's third run poses new and interesting problems that all experiments have to tackle in order to fully exploit the benefits provided by the new architecture, such as the increase in the amount of data to be recorded.

As part of the new developments that are taking place in the ALICE experiment, payloads that use more than a single processing core have become a reality. These new types of jobs vary from production jobs that just spawn a certain number of threads, based on the number of cores of the slot, to analysis jobs that spawn as many processes as tasks.

This paper presents how the ALICE Grid middleware has been changed in order to support and manage multi-core jobs. The additional job isolation and the scheduling requirements for multicore jobs are also presented along with how they can be implemented in different site configurations, in particular for sites shared with other experiments or providing High Performance Computing resources.

Introduction

The ALICE Grid is an effort to pool resources together from different institutions that have the common goal of analysing the events captured by the ALICE detector. By combining the resources physicists can use more resources at a time than their own institutions have at their disposal, thus leading to faster advancement and iteration in the researchers'goals.

The software that enables running jobs on the ALICE Grid is the JAliEn Middleware. It is made up of JAliEn Computing Elements services that run on dedicated nodes on each site (called VOBox-es), which connect to their infrastructure and submit generic jobs from the Grid to the cluster. Another component of the framework is the Central Services, located at CERN, that work as a centralized point of management which dispatches payloads to run on sites, manages the input and output files required by the jobs, catalogues them and steers the clients to the appropriate Grid storage nodes to write their output to.

As of now, the average job that runs on the Grid is expected to use a single processing core, 2 GigaBytes (GB) of physical memory, and up to 8 GB when including swapped memory and 10 GB of disk space. With these assumptions the experiment has operated in Run 1 and 2 with the sites having deployed their resources accordingly. In the Large Hadron Collider's third run ALICE has upgraded the detector and changed the data acquisition model from a triggered mode to a trigger-less /

streaming mode with a sharp increase in bandwidth and storage requirements.

The paradigm shift from processing events to looking at 10ms long "movies"required a complete rewrite of the experiment software, from simulation and reconstruction to the analysis framework. For efficient processing of the new data type the enhanced framework requires larger physical memory allocations per job, in the order of 10 to 20 GB of memory. Swapping them out is not an option as the entire data file content is accessed and thus the CPU efficiency would be dramatically impacted.

The new experiment software is multi-threaded or multi-process which allows for efficient use of multiple core slots on the sites that also satisfy the new memory requirements while maintaining the existing 2 GB per core ratio demand from the resource providers. This paper addresses the modifications that have been made to the JAliEn software in order to allow it to run multi-core jobs.

Another requirement of the Grid is an overall increase in resource allocations from all sites. This paper will also present how the modifications that have made to the JAliEn code allow it to run efficiently on supercomputers, overcoming issues that the experiments have to face in their workload management system for the integration of HPC resource deployments.

State of the Art

CERN's main experiments working with the Large Hadron Collider have made efforts to enable the execution of multi-core jobs on their Grid sites. The approach each of them has taken is different, as are the strategies for integrating HPC resources into its resource pool. In this paper, the strategies taken by the other three main LHC experiments (ATLAS, CMS and LHCb) are analyzed and compared.

Concerning multi-core job scheduling, ATLAS relies on their Grid sites for the scheduling of both single and multi-core jobs. In contrast, the CMS and LHCb experiments have provided their framework with the ability to prioritize and schedule their jobs. The two approaches have a direct impact on the predictability of execution times, granting CMS and LHCb a higher and more accurate prediction capability than ATLAS, which is entirely site-dependent.

Implementing support for multi-core jobs and inclusion of HPC resources

Having to deal with the strict requirements of the LHC third run involves the revision of the framework to adapt it to increase its throughput while using the available resources more efficiently. In order to manage the jobs running using the new framework, the middleware has to be adapted for dealing with its stringent demands. For this purpose, we have worked on the implementation of the logic that allows the framework to be able to execute multi-core jobs, besides continuing to offer support for single-core jobs. This has involved a refactoring in several domains, such as job scheduling, resource management and job launching.

In addition, the framework has been adapted to make use of HPC resources which has enabled their inclusion in the ALICE Grid. The modifications made to the job scheduling, together with the notion of whole-node scheduling on supercomputers have led to an increase of the efficiency of the amount of memory consumed by the running jobs. Moreover, the newly introduced management components have improved the control over the scheduling. This paper addresses one of the issues raised by this new approach, the lack of guaranteed isolation between resources used for the execution of the running jobs, with a focus on the analysis of the CPU cores isolation.

Results

We have been able to run multi-core jobs on several experimental sites on the ALICE Grid and we also deployed the software on the Cori supercomputer, based at the Lawrance Berkeley National Lab. The ALICE sites had to prepare special queues that allocate a certain number of cores per slot. For the moment, all the multi-core queues have been configured to allocate eight cores, while to Cori supercomputer an entire node was allocated that was further partitioned to run multiple single or multi-core jobs. In order to test our implementation, the analysis and production teams created multi-core job prototypes that could be run on the new queues. These jobs do not reflect the real world resource usage of the jobs that will be run in production, but they helped us test the functionality of the new mechanism. As a consequence, aspects such as resource consumption and resource interference could not be tested. This is a subject that will be approached when production ready jobs are received.

In this paper we will analyze the difference in error rates between the jobs running using the old mechanism, that supported running only single core jobs, and the jobs running with the new mechanism.

Significance

We implemented a mechanism capable of managing multi-core slots allocated by ALICE Grid sites and of using the resources in order to launch single and multi-core jobs from the Grid with a user-defined granularity. This new feature allows the experiment to run the new generation of jobs, but it also allows us to deploy the software on new sites, such as supercomputers. Furthermore, this paper proposes a solution for slot fragmentation on the nodes and for CPU resources isolation.

References

Speaker time zone

Compatible with Europe

Authors: BERTRAN FERRER, Marta (Aalto University); Mr WEISZ, Sergiu (University Politehnica of Bucharest (RO))

Presenter: Mr WEISZ, Sergiu (University Politehnica of Bucharest (RO))

Session Classification: Posters: Crystal

Track Classification: Track 1: Computing Technology for Physics Research