Adding multi-core support to the ALICE Grid Middleware

Sergiu Weisz[1] Marta Bertran Ferrer[2]

[1]*University POLITEHNICA of Bucharest* sergiu.weisz@upb.ro [2]*CERN* marta.bertran.ferrer@cern.ch

INTRODUCTION

As part of the new developments that are taking place in the ALICE experiment [1], multi-core payloads have become a reality. There are two common payload types: production with number of threads proportional to the core count and analysis with as many processes as tasks.

We present the modifications that have been made to the JAliEn framework[2] for enabling it to run multi-core jobs while satisfying the memory per core requirements imposed by resource providers. Moreover, the framework has been equipped with the ability to be integrated in new resource types such as supercomputers, thus increasing resource allocations.

ARCHITECTURE FOR SUPPORT OF MULTI-CORE JOBS

The framework has been updated to increase its flexibility through the management of multi-core slots.

- The Computing Element deploys Job Runners, which spawn Job Agent threads.
- A single JVM is created per Job Runner, shared among the spawned Job Agents.
- The Job Runner can distribute an arbitrary amount of memory and CPU resources within the limits of the underlying system, even controlling the entire node.
- Job Agents advertise the available resources and receive compatible payloads, which are kept in a common pool.
- Job Agents can run jobs with arbitrary core requirements.

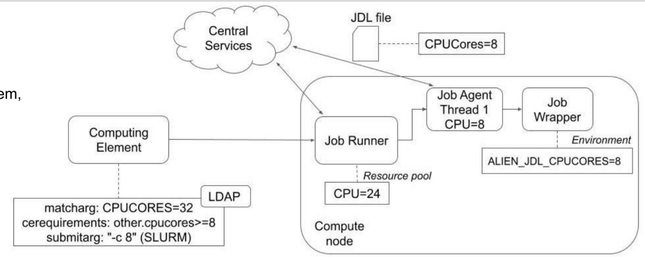

Figure 1: Diagram of the CPU Cores parameter definition and mechanism used for the execution of multi-core jobs.

MULTI-CORE JOBS ON THE GRID

We present two major workflows that have run on different sites:

- Analysis tasks, with multiple processes that access and process data simultaneously, requiring a high speed connection to the Storage Elements.
- Reconstruction payloads, which load the data in RAM and convert it to specific format, requiring a large amount of memory.

The decrease of RAM usage when running eight core jobs is shown in the Table 1.

Figure 2: Ratio of multi-core analysis tasks to single core tasks on the KFKI site

	Average RAM usage per core
1 core	3.17 GB
8 core	1.13 GB

Table 1: Average memory usage per core

SCAVENGING RESOURCES FROM NEW QUEUES

The JAliEn framework allows for integration of new resource types:

- Supercomputers: The Cori supercomputer located at the NERSC's facilities.
- Preemptable queues: The scavenging queues of the HPCS cluster located at LBNL. The utilization of the nodes can be preempted by local applications, thus the ALICE payloads may be interrupted and should be re-submitted.

These systems are used in opportunistic regime and can be an important contribution to the already allocated computing resources.

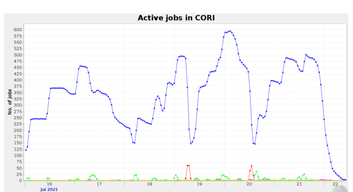

Figure 3: Fluctuation of running jobs on the Cori supercomputer

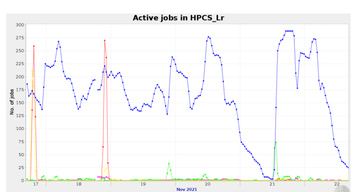

Figure 4: Fluctuation of running jobs on preemptable queues

CPU RESOURCES ISOLATION

While resource providers enforce a strict control of the memory consumption through detailed measurements, we found that more than 80% of the Grid nodes do not constrain the CPU usage. To assure a reasonable CPU use by the payload, the JAliEn framework implements isolation through the *taskset* tool.

The graphs below display the difference between non-isolated and isolated jobs, proving how CPU usage is reduced to the originally requested amount.

Figure 5: CPU usage in jobs non-isolated jobs

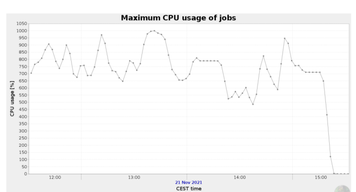

Figure 6: CPU usage in jobs isolated jobs

RESULTS INTERPRETATION

- We observe decreased memory usage per core on multi-core payloads, compared to single core.
- Backfilling and preemptable queues on Supercomputers show promise as additional source of computing resources.
- The CPU usage control achieved through *taskset* is essential to avoid resources contention.

CONCLUSIONS

- The new Grid framework for the ALICE Run3 and beyond implements an important new paradigm - multicore scheduling and payload support.
- For a more efficient resource usage it incorporates methods to isolate and limit the payload CPU, disk and memory consumption.
- Supercomputers and preemptable queues are added as a source of opportunistic resources.

FUTURE WORK

- Complete the conversion of the ALICE Grid to the new JAliEn framework.
- Provide methods for integration of supercomputers with limited network access.
- Further automatization of the framework deployment to facilitate the deployment on new sites and computers with exotic architecture, including accelerators.
- Reduce the impact on payload efficiency from isolation tools.
- Enhance and refine the accounting mechanisms.

REFERENCES

[1] K. Aamodt, et al., The ALICE experiment at the CERN LHC, JINST 3 (2008) S08002. doi:10.1088/1748-0221/3/08/S08002.

[2] M. Martinez Pedreira, C. Grigoras, and V. Yurchenko. "JAliEn: the new ALICE high-performance and high-scalability Grid framework". In: EPJ Web Conf. 214 (2019). Ed. by A. Forti et al., p. 03037. Doi: 10.1051/epjconf/201921403037.