Data Analytics for Scheduling in Hybrid HPC Environments

Abel Souza, Johan Tordsson
Umeå University - Sweden
{abel,tordsson}@cs.umu.se

Mohamad Rezaei, Erwin Laure
KTH Royal Institute of Technology - Sweden
{mrez,erwinl}@kth.se

1. Background

Data Intensive (DI) applications require higher degree of integration with the system-level scheduler [2] than High Performance Computing (HPC) ones. Due to their dynamic nature, running DI applications in HPC clusters either wastes resources or limits their execution, as they are not always deadline bound. Traditional HPC schedulers are commonly used for coarse-grained resource allocation [2] and usually require both a job geometry (e.g., number of nodes, cores and memory) and a deadline. HPC users tend to make bad estimations about both these parameters, but DI jobs cannot be defined by such parameters due to their reactive characteristics. Furthermore, a majority of DI jobs in cloud data centers are long running jobs [1]. This is even more visible for jobs in the scientific community that tend to have very repeating patterns as researchers tend to focus on same problems for long periods of time. Long jobs give enough data in their lifetime for making robust models to understand their resource utilization.

2. Proposal

We propose to meet these differences in job requirements by a hybrid resource management architecture that combines schedulers for HPC and DI jobs. An Insight Engine (IE) will be used to in real time analyze performance traces, model these, and based on the derived models avoid interference between collocated jobs. Figure 1 shows the proposal architecture where Mesos (DI job scheduler) runs on top of the SLURM HPC scheduler to collocate HPC and DI jobs in same nodes. The IE ensures the isolation needed to guarantee predictability and scalability of HPC jobs, while improving resource utilization by collocating long HPC running jobs with DI tasks. The IE uses reactive sampling, i.e., each cluster node provides streamed data upon performance interference or requests from the IE. Worker nodes retrieve updated models of their running jobs to detect interference and outliers. Each node enforces isolation autonomously, by stopping DI jobs whenever needed.

3. Challenges

The herein proposed hybrid environment imposes research challenges within data analytics as well as cluster management. In order to have an updated view of resources, IE sampling must be performed real time and distributed in each node. The overhead imposed by sampling can impact the overall node performance, in particular if either the sampling resolution or the number of monitored micro-architectural metrics (cycles per instruction, cache misses, TLB misses, etc.) are high. The former can be handled by reactive profiling and streaming of task traces, and by controlled by using the Bag of Little Bootstraps [3] method that can lower the profiling overhead significantly. The latter depends on application characteristics and on how accurately the model captures the resource usage. Techniques such as multi-variate linear models, feature selection, and classification models can be used to evaluate and characterize different models for predictability of DI and HPC applications.

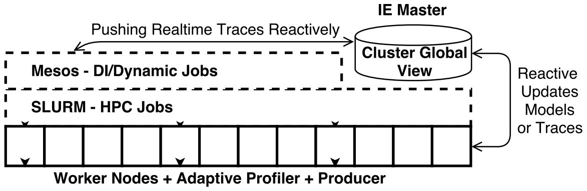
IE Master
Pushing Realtime Traces Reactively
Cluster Global View
Mesos - DI/Dynamic Jobs
SLURM - HPC Jobs
Reactive Updates Models or Traces
Worker Nodes + Adaptive Profiler + Producer

Figure 1. Worker nodes producing traces to the IE master.

Other concerns lie on integrating two different resource managers in the same cluster, as they have different objectives and isolation must be guaranteed. Enforcing this isolation can be the limiting factor since HPC clusters are usually bigger and more scalable than normal DI cloud platforms.

Finally, the rate of additional failures that Mesos will see compared to normal failures should be studied, as interferences may happen more frequently.

References

[1] P. Delgado, F. Dinu, A.-M. Kermarrec, and W. Zwaenepoel. Hawk: hybrid datacenter scheduling. In *USENIX ATC 15*, pages 499–510, 2015.

[2] S. Jha, J. Qiu, A. Luckow, P. Mantha, and G. C. Fox. A tale of two data-intensive paradigms: Applications, abstractions, and architectures. In *IEEE BigData Congress, 2014*, pages 645–652, 2014.

[3] A. Kleiner, A. Talwalkar, P. Sarkar, and M. Jordan. The big data bootstrap. *arXiv preprint arXiv:1206.6415*, 2012.