Resource requirement specification for novel data-aware and workflow-enabled HPC job schedulers

E. Farsarakis, I. Panourgias, A. Jackson, J.F.R. Herrera, M. Weiland, M. Parsons
EPCC, The University of Edinburgh, UK

ABSTRACT

Technological advancements in computer architectures and the ever-increasing scope of what may be described as a hybrid HPC architecture have not been followed by relevant changes in the way jobs are described to HPC job schedulers. In this WiP, we aim to introduce an augmented job resource request (JRR) specification to be adapted by existing job scheduler implementations and used to more accurately describe the resource requirements of a job to HPC schedulers. The ultimate aim of this work is to both improve the performance of individual applications by improving the utilization of novel resources as they become available, as well as to enable the more efficient scheduling of jobs and workflows on future HPC systems.

CCS CONCEPTS

• **Hardware → Memory and dense storage**; • **Software and its engineering → Massively parallel systems**;

KEYWORDS

Storage class memory, Workflows, Scalability, NEXTGenIO, Resource allocation, SLURM, Portable Batch Scheduler, HPC, Systemware

1 MOTIVATION

We will demonstrate the motivation behind this work through the use of three representative theoretical use cases which highlight the issues faced by current job scheduling systems, followed by three widely used HPC applications which demonstrate some of these functionalities in their behavior.

1.1 Use cases

In the following three subsections we will look at three types of situations where storing files in high performance local storage might be beneficial to an application.

1.1.1 Frequently used, read-only data. Use case: A job reads from a moderately sized, read-only file throughout execution.

1.1.2 On-node files. Use case: A job indicates a group of files which it will access and modify on a per process or per node basis. Such files could be output files, independently generated by each process (i.e. MPI process).

1.1.3 Workflows. Use case: A newly submitted job depends on a previous job which is still running or is waiting to run. It will use the output files of that job as input. These files are indicated to the scheduler as node specific input files.

1.2 HPC Applications

We will present three example HPC applications that could benefit from such functionality, and are in line with the use cases we have already described.

COSA is a structured multi-block Navier-Stokes (NS) code featuring a steady, a time domain (TD) and a harmonic balance (HB) solver, all using a finite volume space-discretisation and an efficient multigrid integration.

OpenFOAM is a framework that provides a set of numerical solvers and tools for solving continuum mechanics problems, including computational fluid dynamics (CFD).

CASTEP is a density functional theory code that can be used to calculate the properties of materials from first principles.

2 THE JRR SPECIFICATION

Existing job descriptions for job schedulers such as SLURM and PBS focus mainly on the computational requirements of a job, such as number of nodes, number of CPU cores and the type of processing units required. They usually offer limited functionality for workflows in the form of a series of interdependent job ids. Finally, there is limited functionality for memory requirements such as the size of DRAM required by the job or files and locations to be placed in high speed off-node storage such as Burst Buffer. It is clear however from the use cases presented previously, that these specifications are insufficient to fully exploit the performance potential of complex, hybrid HPC systems with fast on-node storage, or shared SCM throughout the system.

The Job Resource Requirements specification we are proposing will augment existing job descriptions in order to incorporate information about a jobâĂŹs data and data movement requirements. Our initial investigation of possible use cases has resulted in the following additional vocabulary to be considered for the augmentation of standard existing vocabulary of existing scheduler implementations.

- “IN", “READ_ONLY", “LOCAL", “OUT", “INOUT" to describe files
- Process mapping file

ACKNOWLEDGMENTS

The NEXTGenIO project received funding from the EU Horizon 2020 research and innovation programme under grant agreement No 671591.