Provenance-based Profiling of Swift Parallel and Distributed Scientific Workflows

Maria Luiza Mondelli, Fabrício Vilasbôas, Kary Ocaña, Marta Mattoso, Michael Wilde, Ana Tereza Vasconcelos, Luiz Gadelha

National Laboratory for Scientific Computing, Federal University of Rio de Janeiro, Computation Institute, Argonne National Laboratory/University of Chicago

Abstract

The demand for high-performance computing (HPC) resources has increased in recent time due the complex features of bioinformatics experiments and the biological big data that need to be processed. In general, these experiments execute a set of applications as a flow of activities in which one data is the entry of another activity, suggesting they can be modeled as scientific workflows. Scientific Workflow Management Systems (SWfMS) are used to manage the distribution and parallelism of scientific workflows in HPC environment. Swift is a SWfMS that follows the functional programming paradigm with implicit parallelism. However all benefits provided by Swift, managing provenance scientific data is still an open and challenging problem to be resolved in the next years. Swift creates by default a set of log files containing information of some environment statistics related to the workflow execution. Log files are created as the workflow finishes and contains information e.g., about the workflow execution time or the status of the activity/task execution, which are stored in a relational database. This information is not naturally reported to scientists, but it contains invaluable information about the performance of workflow execution. If scientists could access this provenance, they could be better positioned about their own experiments, for example to determine which data/parameter could generate possible executions errors or if any debugging process can be implemented. For assisting scientists at analyzing the performance of their workflow implementations, we designed a profiler tool called SwiftProfiler. It was implemented in Python and encapsulates a set of SQL queries to the provenance database, to extract resource usage behavior. With SwiftProfiler scientists are able to: (i) calculate execution time of e.g., tasks, activities or total workflow, (ii) report the CPU usage, (iii) trace the provenance of the workflow execution results to each workflow activity, and (iii) present statistical calculation as tables or graphics. As a future work, we will attempt to extend the profiler for processing more sophisticated queries, such as enabling scientists to profile read/write to filesystem behavior.