



Contribution ID: 72

Type: **Short talk**

Enhancing iteration performance on distributed task-based workflows

Thursday 23 March 2023 15:00 (10 minutes)

Task-based programming models have proven to be a robust and versatile way to approach development of applications for distributed environments. The programming model itself feels natural and close to classical algorithms; the task-based distribution of tasks can achieve a high degree of performance. All this is achieved with a minimal impact on programmability. However, execution on this paradigm can be very sensitive to the granularity of tasks –i.e., the block size, or equivalently, the quantity and execution length of tasks. This is manifested during the iteration of the distributed datasets, a procedure that will yield tasks across the distributed computing resources. Identifying and setting this optimal block size is not trivial, requires inner knowledge of the computing environment, and is not an easy task for the domain expert –i.e. the application developer. Having the programming model performance be highly dependent on this block size is undesirable and a challenge to overcome.

Our proposal is to enhance the distributed iterations by including a new mechanism –a procedure that we call *split*. At its core, the *split* mechanism provides a transparent way to get *partitions* (which are logical groups of blocks, obtained without any transfers nor data rearrangement) of blocks. By doing so, performance is improved as the system produces fewer tasks, there is a cutback on the scheduling cost, and the invocation overhead is reduced. Our proposed implementation of the *split* goes one step further and also integrates with the storage framework, thus being able to attain those benefits while guaranteeing data locality.

The evaluation we have conducted shows that *split* mechanism is able to achieve performance improvements of over one order of magnitude. We have chosen different applications covering a wide variety of scenarios; those applications are representatives of a broader set of applications and domains (both memory-intensive and CPU-intensive applications, for applications widely used in Machine Learning, Data Analytics, etc.). The changes required in the source code of a task-based application are minimal, preserving the high programmability of the programming model.

JLESC topic

Primary author: BARCELO, ALEX (Barcelona Supercomputing Center)

Presenter: BARCELO, ALEX (Barcelona Supercomputing Center)

Session Classification: Short Talks on Tasking

Track Classification: Programming languages and runtimes