Dask-Enabled External Tasks For In Transit Analytics

Amal Gueroudji*†, Julien Bigot*+, Bruno Raffin§

* Universite Paris-Saclay, UVSQ, CNRS, CEA, Maison de la Simulation, 91191, Gif-sur-Yvette, France

† Email: amal.gueroudji@cea.fr

+ Email: julien.bigot@cea.fr, ORCID: https://orcid.org/0000-0002-0015-4304

§ Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LIG, 38000 Grenoble, France

Email: bruno.raffin@inria.fr, ORCID: https://orcid.org/0000-0002-7980-4946

Most In situ data analytics tools are based on the MPI programming model inherited from the host simulation. While the MPI+X model suits many high-performance simulations well, higher-level programming models that promote programmer productivity, like map-reduce or task models, are preferred for parallel data processing algorithms characterized by irregular communication and control patterns. In a previous work [1], we have proposed to use Dask for in situ analytics via DEISA to take advantage of MPI performance for simulation and Dask productivity for analytics. In this previous work, data and metadata were exchanged synchronously at each timestep, and a new task graph was submitted to process that step every time. Our current work replaces this with a new solution that introduces asynchronicity and reduces the traffic to the scheduler. We avoid metadata fetch and allow submitting time-independent task graphs. This offers better performance by reducing the load of the centralized scheduler. It also makes analytics easier to write by considering global Dask arrays, including the time dimension, thus letting Dask manage dependencies and schedules better.

To this end, we have introduced three main concepts: ***deisa virtual arrays***, ***contracts,*** and ***external tasks*** in Dask distributed. A ***deisa virtual array*** describes the spatio-temporal domain decomposition of a generated data array. It contains the global sizes in each dimension, including the time dimension, the size of each block (size of generated data by each MPI rank), and the starting indexes of each block. Describing the data in this way allows us to have a global view of the generated data. ***Deisa virtual arrays*** are used to create the Dask arrays in the analytics client. They are sent to Dask through contracts. A ***contract*** is concluded between the simulation and Dask at the beginning of the simulation: MPI rank 0 builds the ***deisa virtual arrays*** descriptors using data from the simulation and sends them to Dask. The analytics client in Dask then selects the data it is interested in from the available ***deisa virtual array***s, and sends back a selection request to the simulation identifying the data it is actually interested in. It also creates matching Dask arrays for the analysis. The created Dask arrays are collections of ***external tasks,*** with the specific state ***'deisa'*** and particular keys. They refer to tasks computed by other applications outside from Dask, and can be used as other Dask arrays, thus integrated into Dask task graphs transparently. In Dask, data is considered as a specific task called pure data task, and the ***external tasks*** belong to that category. By introducing the new state ***'deisa',*** we prevent the scheduler from sending those tasks to the workers. They only become schedulable when the simulation sends the generated data with that specific keys to a worker. The tasks that depend on that specific ***external task*** can then be scheduled.

We have implemented these improvements on top of the work presented in [1]. We have added a new deisa plugin in the PDI Data interface and included our ***external tasks*** contribution into a forked version Dask distributed repository. We have tested our approach using a heat equation mini-app with several analytics, such as temporal derivative and incremental PCA. We are currently working on using our work with Gysela[2] nuclear fusion code.

[1] A. Gueroudji, J. Bigot and B. Raffin, "DEISA: Dask-Enabled In Situ Analytics," 2021 IEEE 28th International Conference on High Performance Computing, Data, and Analytics (HiPC), 2021, pp. 11-20, doi: 10.1109/HiPC53243.2021.00015.

[2] V. Grandgirard, J. Abiteboul, J. Bigot, T. Cartier-Michaud, N. Crouseilles, G. Dif-Pradalier, Ch. Ehrlacher, D. Esteve, X. Garbet, Ph. Ghendrih, G. Latu, M. Mehrenberger, C. Norscini, Ch. Passeron, F. Rozar, Y. Sarazin, E. Sonnendrücker, A. Strugarek, D. Zarzoso January 2016