Smart Data Analytics Workflows for High Performance Simulations

Advisors

Advisor:	Julien Bigot	julien.bigot@cea.fr	CEA researcher	Tel: +33 (0)1 69 08 01 75
Director:	Bruno Raffin	bruno.raffin@inria.fr	Inria research director	Tel: +33 (0)4 56 52 71 39

Context

The increasing performance gap between computing elements and permanent storage in supercomputer reshuffles the cards for optimization of simulation codes. Peripheral concerns such as I/O and post-processing data analytics that used to be developed independently with little care for performance are becoming more and more critical to optimize. Their integration in the main code can however come at a high cost in term of maintainability and performance portability. Simulations are also more and more commonly requested to run associated with data analytics processes when coupled with other codes, scientific instruments or sensors, as for Digital Twins applications. These analytics, like deep learning, can require significant compute resources leading to complex multi-level parallelizations and data stagings. Today's approaches are mostly based on explicit and manual code assembly that are time consuming to write, intrusive in the simulation code and lack flexibility.

Building workflows that achieve high performance executions while keeping programming simple requires a framework enforcing modularity through a good separation of concerns, and automatic assembly processes for negotiating task placement and data extraction.

Post-mortem data analytics is an approach where the simulation writes *raw* results to permanent storage that is later analyzed with independent software. By construction, this ensures a good separation of concern between the main simulation code and data analytics. This approach does however lead to severe performance issues where simulation is limited by the disk bandwidth and can not leverage supercomputers computing power.

In-situ processing consists in using the same computing resources for data analytics and simulation. This reduces the performance impact as the data can be reduced before it is written to disk. Many variants of in-situ data analytics exist ranging from **a)** sequential where the simulation code is interrupted for data processing, **b)** dedicated core where some cores of each node are reserved for analytics or even **c)** task-based where the runtime dynamically interleaves fine grain simulation and analytics tasks[7]. Depending on the approach chosen, memory overhead and cache trashing can however still badly impact performance. The approach is also typically rather invasive in the code and can be very complex to implement if the data distribution used in the analysis differs from that of the simulation proper.

In-transit processing consists in using dedicated nodes, often called staging nodes, distinct from the simulation nodes. This approach remove load from the simulation nodes that do not need to share their resources (compute units, cache, memory) with the analytics, but these node may become underused and it may require costly data transfers towards these staging nodes. Some frameworks like ADIOS exclusively support one option. Other frameworks like FlowVR, Damaris or Decaf support in-situ and in transit processing, but task placement and data movements need to be manually defined. The choice is often the results of a multi-criteria tradeoff. The in transit storage system DataSpace has recently investigated several approaches for the automatic data analytics placement, but on the staging nodes only [14, 2].

Goals

The goal for this Ph.D. thesis is to investigate a modular workflow design and automatic assembly processes capable of achieving high performance executions while keeping programming simple. This approach will have to minimize the impact on simulation code both in term of performance and software complexity by ensuring a good separation of concerns.

Based on some constraints expressed by the user, the system should be able to automatically assemble the different requested data processing steps, with in-situ or in transit task placement and efficient data extraction and redistributions.

The proposed approach will have to take into account the specificity of data processing workloads for HPC: strong memory coupling of some operations, new *deep memory* hardware, different requirements (memory, CPU, ...) of different operations, etc. It will have to handle new types of workload expected to appear, including machine-learning based ones for example. It will have to handle workflow including logic based on the data produced by simulation to activate or disable some analytics steps. To reach these goals, it will likely have to extend approaches like lazy data copies, contract based data extraction [13], $N \times M$ data redistributions [4].

Work-plan

During the first phase of the work, the candidate will study the related bibliography (see an extract bellow) to get a good understanding of the involved concepts, understand the limitations of current approaches, and size the benefits of various existing software. In a second step the work will focus on the workflow scripting design, building on existing solutions in particular the PDI and FlowVR libraries developed by the two teams co-advising this PhD. A goal for this first step will be to design a platform that clearly separates the specification of the various post-processing tasks from their assembly and placement on computing resources with required communication methods using multiple approaches (sequential in-situ, tasks, dedicated cores or nodes, etc.) Finally, the focus will turn towards an automatic assembly and placement process, likely the more challenging part of the work.

Proposed solutions will be prototyped and experimented on supercomputers for validation, and the results published at international conferences and journals like Supercomputing, IPDPS or JPDC. The candidate will have access to several supercomputers. as well as to large-scale production simulation codes including codes developed and used at CEA such as the Tokamak plasma simulation code GYSELA5D [10] (http://gyseladoc.gforge.inria.fr/), or standard molecular dynamics simulation code Gromacs (http://www.gromacs.org/).

Involved Research Teams

Maison de la Simulation (MdlS), http://www.maisondelasimulation.fr/ is a joint laboratory of CEA, CNRS, Inria, Univ. Versailles Saint-Quentin and Univ. Paris-Saclay located on the Saclay plateau. The laboratory activities revolve around high performance computing (HPC): research in computer science and applied mathematics, engineering and development of simulation applications. Of specific interest at MdlS are software engineering aspects of HPC, especially regarding separation of preoccupations.
✉ Maison de la Simulation, bât. 565, CEA Saclay, 91191 Gif-sur-Yvette CEDEX

DataMove, https://team.inria.fr/DataMove/ is a joint research team in between Inria and LIG from Univ. Grenoble-Alpes. DataMove focuses on on optimizing data movements for large scale computing mainly at two related levels: resource allocation, and integration of numerical simulation and data analysis.
✉ LIG, Batiment IMAG, 700 Avenue Centrale, Domaine Universitaire de Saint-Martin-d'Heres, CS 40700, 38058 Grenoble CEDEX 9

Available Existing Software

PDI (https://gitlab.maisondelasimulation.fr/jbigot/pdi) is a library to decouple high-performance simulation codes from peripheral concerns such as input/output or post-processing. It offers a declarative API for codes to expose the buffers in which they store data and to notify PDI of significant steps of the simulation. A YAML file is used to specify what to do with this data instead of interleaving this logic with the simulation code. PDI offers both **a)** a plugin system to access existing third-party libraries from the YAML file and **b)** enable users to call arbitrary code from the YAML file.

FlowVR (http://flowvr.sf.net) is an open source middelware to augment parallel simulations running on thousands of cores with in-situ processing capabilities and live steering. FlowVR offers a very flexible environment while enabling high performance asynchronous in-situ and in transit processing. FlowVR, puts the emphasis on the programmability. FlowVR is based on a component model. The scientist designs an analytics pipeline by first developing processing components that are then assembled in a dataflow graph through a Python script. At runtime, the graph is instantiated according to the execution context, the framework taking care of deploying the application on the target architecture, and of coordinating the analytics workflows with the simulation execution [6]. FlowVR is not bound to MPI. It supports different transport layers and enables to couple components that rely on different parallelization paradigms. FlowVR is particularly interesting for building heterogenous in-situ pipelines, for instance, integrating monitoring and steering [5].

References

[1] Leonardo Bautista-Gomez, Seiji Tsuboi, Dimitri Komatitsch, Franck Cappello, Naoya Maruyama, and Satoshi Matsuoka. Fti: High performance fault tolerance interface for hybrid systems. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '11, pages 32:1–32:32, New York, NY, USA, 2011. ACM.

[2] J. Y. Choi, J. Logan, M. Wolf, G. Ostrouchov, T. Kurc, Q. Liu, N. Podhorszki, S. Klasky, M. Romanus, Q. Sun, M. Parashar, R. M. Churchill, and C. Chang. Tge: Machine learning based task graph embedding for large-scale topology mapping. In *2017 IEEE International Conference on Cluster Computing (CLUSTER)*, pages 587–591, Sept 2017.

[3] Matthieu Dorier, Matthieu Dreher, Tom Peterka, Justin M. Wozniak, Gabriel Antoniu, and Bruno Raffin. Lessons Learned from Building In Situ Coupling Frameworks. In *Workshop on In Situ Infrastructures for Enabling Extreme-scale Analysis and Visualization (ISAV'15)- Held in conjunction with SC15*, Austin, November 2015. ACM.

[4] Matthieu Dreher and Tom Peterka. Bredala: Semantic data redistribution for in situ applications. In *CLUSTER - IEEE International Conference on Cluster Computing*. IEEE, September 2016.

[5] Matthieu Dreher, Jessica Prevoteau-Jonquet, Mikael Trellet, Marc Piuzzi, Marc Baaden, Bruno Raffin, Nicolas Ferey, Sophie Robert, and Sébastien Limet. ExaViz: a Flexible Framework to Analyse, Steer and Interact with Molecular Dynamics Simulations. *Faraday Discussion*, 169:119–142, 2014.

[6] Matthieu Dreher and Bruno Raffin. A Flexible Framework for Asynchronous In Situ and In Transit Analytics for Scientific Simulations. In *14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, Chicago, United States, May 2014. IEEE Computer Science Press.

[7] Laurent Colombet Estelle Dirand and Bruno Raffin. Tins: A task-based dynamic helper core strategy for in situ analytics. In *SupercomputingAsia (SCAsia) 2018*, Singapore, 2018.

[8] Wolfgang Frings, Felix Wolf, and Ventsislav Petkov. Scalable massively parallel i/o to task-local files. In *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, SC '09, pages 17:1–17:11, New York, NY, USA, 2009. ACM.

[9] Yuankun Fu, Feng Li, Fengguang Song, and Zizhong Chen. Performance analysis and optimization of in-situ integration of simulation with data analysis: Zipping applications up. In *Proceedings of the 27th International Symposium on High-Performance Parallel and Distributed Computing*, HPDC '18, pages 192–205, 2018.

[10] Virginie Grandgirard, Jérémie Abiteboul, Julien Bigot, Thomas Cartier-Michaud, Nicolas Crouseilles, Guilhem Dif-Pradalier, Charles Erhlacher, Damien Esteve, Xavier Garbet, Philippe Ghendrih, Guillaume Latu, Michel Mehrenberger, Claudia Norscini, Chantal Passeron, Fabien Rozar, Yanick Sarazin, Eric Sonnendrücker, A. Strugarek, and David Zarzoso. A 5D gyrokinetic full-f global semi-lagrangian code for flux-driven ion turbulence simulations. *Computer Physics Communications*, 207:35–68, 2016.

[11] Cyrus Harrison, Brian Ryujin, Adam Kunen, Joe Ciurej, Kathleen Biagas, Eric Brugger, Aaron Black, George Zagaris, Kenny Weiss, Matt Larsen, Todd Gamblin, George Aspesi, Justin Bai, Rupert Deese, and Linnea Shin. Conduit: Simplified data exchange for hpc simulations. http://llnl-conduit.readthedocs.io. Accessed: 2018.

[12] Jay F. Lofstead, Scott Klasky, Karsten Schwan, Norbert Podhorszki, and Chen Jin. Flexible io and integration for scientific codes through the adaptable io system (adios). In *6th international workshop on Challenges of large applications in distributed environments*, CLADE '08, pages 15–24, New York, 2008. ACM.

[13] Clement Mommessin, Matthieu Dreher, Bruno Raffin, and Tom Peterka. Automatic data filtering for in situ workflows. In *IEEE Cluster*, Hawaii, 2017.

[14] Qian Sun, Tong Jin, Melissa Romanus, Hoang Bui, Fan Zhang, Hongfeng Yu, Hemanth Kolla, Scott Klasky, Jacqueline Chen, and Manish Parashar. Adaptive data placement for staging-based coupled scientific workflows. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '15, pages 65:1–65:12, New York, NY, USA, 2015. ACM.

[15] The HDF Group. Hierarchical Data Format, version 5. http://www.hdfgroup.org/HDF5/. Accessed: 2018.

[16] F. Zheng, H. Zou, G. Eisnhauer, K. Schwan, M. Wolf, J. Dayal, T. A. Nguyen, J. Cao, H. Abbasi, S. Klasky, N. Podhorszki, and H. Yu. FlexIO: I/O middleware for Location-Flexible Scientific Data Analytics. In *IPDPS'13*, 2013.