Geographical Workflow System over HPC Clusters Based on MPI

Anran Yang[1], Lu Liu[1], Luo Chen[1], Ning Jing[1]

[1]Department of Information Engineering, National University of Defence Technology
Yanwachi 137, Changsha, China
Telephone: (+86)73184573480
Email: {yanganran10, luliu, luochen, ningjing}@nudt.edu.cn

1. Introduction

High performance computing (HPC) offers possibility to overcome the hardness of geographical problems caused by big data, time-consuming algorithms and complicated geographical models. Message passing interface (MPI) is a clean, flexible, effective paradigm for HPC, which has been widely used in scientific researches. Nowadays, Geographical Information Science (GIScience) becomes one of the new fields that use MPI to speedup specific problem solving. Tarboton (2009) used MPI in tauDEM, a digital terrain analysis (DEM) library and reported significant performance improvement. Guan (2010) developed a parallel geospatial programming library built on MPI to help nonspecialist GIScientists develop high performance geospatial algorithms. Wang (2009) presented TeraGrid GIScience Gateway where MPI enables parallel processing within and across nodes in the cyberinfrastructure. However, most of the existing works are limited in terms of applying them to solve real-world problems. Although HPC is effective in reducing the execution time of algorithms, and provides opportunities to optimize the bottleneck of data IO, it worsens the problem caused by complicated scientific models since scientists have to concern with the parallel logic in addition to the complexity of models.

In scientific fields, workflow systems or more specifically scientific workflow systems have been widely used to build complex applications from simple components easily (Curcin 2008). Scientific workflow systems can introduce some parallelism implicitly (Zhao 2008). Tasks independent with each other can be executed in parallel naturally while necessary synchronizations preserve the processing logic defined by users. Since only parallel logic inside the simple algorithms should be handled, the complexity of development will be effectively reduced. Workflow systems specific to geographical computation were also presented. Chen (2008) developed a geographical workflow execution platform named MRGIS over MapReduce clusters to ease solution building and achieved significant performance advantages compared to non HPC solution. Ma (2009) proposed an architecture for high performance geocomputing, which contains a workflow engine to compose geographical algorithms mainly based on MPI. However, the details of requirements, issues and implementation concerns of geographical workflow systems over HPC clusters based on MPI have not been investigated in depth.

Several attributes are expected when building a geographical workflow system over HPC clusters based on MPI. It should be easy to combine simple MPI algorithms into complex solutions. The parallel efficiency should be optimized at the workflow level, to

fully exploit the potential of hardware and algorithms; Since HPC is costly, the system should offer some control abilities, like workflow cancelling and pausing/resuming, as well as a proper failure handling mechanism. In this paper, a workflow system over HPC clusters based on MPI is designed and developed targeted at these attributes.

2. Methods

A workflow as a complex processing procedure is split into a bunch of steps. Each step in the workflow refers to an action, mainly as a single algorithm and represented as an executable file. The action's interface is defined by extra descriptions besides the executable. The inputs of steps can be given either by users or former steps. Constraints can be given to define the relations of steps. Directed acyclic graph (DAG) is used to model the approach with nodes and edges corresponding to steps and relations between steps respectively. Recursions or loops are not modelled in the graph to avoid validating complexity.

A flexible framework integrated with basic geographical operations is also established. A bunch of workflow utility programs are developed in order to build reasonable geographical applications, including metadata extracting, data publishing, web map service (WMS) publishing and many more. These programs can be easily used as steps in workflow and execute in a transactional manner.

Geographical algorithms tend to be both IO and communicating intensive, which restricts their scalability. Within the same workflow, computing resources could be balanced between steps considering the scalability of corresponding algorithms to raise both the utilization ratio of resources and the overall execution efficiency. Suppose there are two algorithms to run on a 256-cores cluster. Executing them one by one with both given 256 cores may cause low parallel efficiency. For most geographical algorithms, the parallel efficiency of a 256-cores execution may be much lower than a 128-cores execution in practice. Consequently, the workflow engine may optimize the resource assignment and execute the two algorithms simultaneously with both given 128 cores, which may bring big performance boost.

Stable workflow control facilities are rather difficult to implement since many real-world workflows are fairly complex and have many states. Concurrency must be carefully handled to avoid inconsistent states or dead locks if synchronizations are heavily taken. Concurrent entities interact with each other all through explicit interfaces driven by an event-callback metaphor, in order to reduce uncertainty introduced by implicit interactions. Some flexibility is offered to handle failures in various ways. When a step fails, one may choose to either cancel all the unfinished steps in the workflow, or only cancel the steps dependent on the failed step. Both of the choices can avoid processing on the corrupted data.

3. Implementation

The core system that builds and executes workflows is written in C++, over RedHat Enterprise Linux (RHEL) 5.5 operating system. The algorithms are mainly written in C++ based on MPI, while geographical workflow utilities are written in various scripting languages like python, ruby, or bash. Such utilities can be any executables supplemented with corresponding interface description files. The system is open to non-MPI algorithms which, however, may not take advantages of special optimizations for MPI programs.

The TORQUE resource manager and the Maui scheduler are used to submit and execute MPI programs since they are open-source and rather solid. OpenMPI is chosen as the MPI implementation while in most cases it is not a heavy work to switch to other implementations, e.g. MPICH2 or Intel MPI. Several geographical workflows are built for experiments. The experimental results showed the feasibility and efficiency of the workflow engine. It provides varies functions including executing algorithms in parallel, taking several previous outputs to produce new output, registration to metadata storage then publish as Tile Map Service (TMS) automatically, and works well in a high-performance GIS.

4. Conclusion

This paper presents a geographical workflow system over HPC clusters based on MPI. The method takes advantage of high-performance geocomputation, and attempts to reduce the complexity when building efficient solutions to complex geographical problems. Future works include more strategies for scheduling workflows, more control operations like pause and resume, and experiments on practical geographical problems in larger scales.

5. Acknowledgements

This work is funded by National High-tech R&D Program of China (863 Program) (No. 2011AA120306). The development of the system takes resources from open-source projects in related fields, e.g. OpenMPI[1], TORQUE[2], Maui[3], etc. We sincerely appreciate the community and developers of these projects.

6. References

Tarboton, D. G., Watson, D. W., Wallace, R. M., Schreuders, K., & Tesfa, T. K. (2009). Hydrologic Terrain Processing Using Parallel Computing. *AGU Fall Meeting Abstracts* Vol. 1, p. 0867.

Q. Chen, L. Wang, and Z. Shang, MRGIS: A MapReduce-Enabled High Performance Workflow System for GIS. In: *IEEE Fourth International Conference on eScience, 2008.* eScience 08, 2008, pp. 646 – 651.

Guan, Q., Clarke, K.C., 2010. A general-purpose parallel raster processing programming library test application using a geographic cellular automata model. In: *International Journal of Geographical Information Science* 24, 695 – 722.

Wang, S., Liu, Y., 2009. TeraGrid GIScience Gateway: Bridging cyberinfrastructure and GIScience. In: *International Journal of Geographical Information Science* 23, 631 – 656.

Curcin, V., Ghanem, M., 2008. Scientific workflow systems - can one size fit all?. In: *Biomedical Engineering Conference, 2008. CIBEC 2008. Cairo International.* Presented at the Biomedical Engineering Conference, 2008. CIBEC 2008. Cairo International, pp. 1 – 9.

Zhao, Y., Raicu, I., Foster, I., 2008. Scientific workflow systems for 21st century, new bottle or new wine?. In: *Services-Part I, 2008. IEEE Congress* On. pp. 467–471.

Ma, Y., Liu, D., Li, J., 2009. A new framework of cluster-based parallel processing system for high-performance geo-computing. In: *Geoscience and Remote Sensing Symposium, 2009 IEEE International, IGARSS* 2009. pp. IV–49.

[1] Open MPI: Open Source High Performance Computing. http://www.open-mpi.org/

[2] TORQUE Resource Manager. http://www.adaptivecomputing.com/products/open-source/torque/

[3] Maui scheduler. http://www.adaptivecomputing.com/products/open-source/maui/