

Storage Aware Resource Allocation for Grid Data Streaming Pipelines

Wen Zhang¹, Junwei Cao^{2,3*}, Yisheng Zhong^{1,3}, Lianchen Liu^{1,3}, and Cheng Wu^{1,3}

¹Department of Automation, Tsinghua University, Beijing 100084, China

²Research Institute of Information Technology, Tsinghua University, Beijing 100084, China

³Tsinghua National Laboratory for Information Science and Technology, Beijing 100084, China

*Corresponding email: jcao@tsinghua.edu.cn

Abstract

Data streaming applications, usually composed with sequential/parallel tasks in a data pipeline form, bring new challenges to task scheduling and resource allocation in grid environments. Due to high volumes of data and relatively limit storage capability, resource allocation and data streaming have to be storage aware. In this paper, Genetic Algorithm (GA) is adopted for task scheduling of pipelines, based on on-line measurement and prediction with Gray Model (GM). On-demand data streaming is introduced to avoid data overflow using repertory strategies. Experimental results show that balance among task executions with on-demand data streaming is required to improve overall performance, avoid system bottlenecks and backlogs of intermediate data, and increase data throughput of pipelines as a whole.

1. Introduction

Grid data streaming applications require *bring-data-to-program* supports. Storage availability has to be considered during task scheduling and resource allocation. A grid data streaming application is decomposed into several tasks which are executed in a form of pipelines. Genetic Algorithm (GA) [1] is adopted for task scheduling, based on on-line performance measurement and prediction with Gray Model (GM) [2]. On-demand data streaming is introduced to avoid data overflow using repertory strategies, and implemented using GridFTP [3] of the Globus Toolkit [4] by tuning the data transfer parallelism. Experimental results included in this work show that our approach makes better use of CPU cycles as well as improving data throughput of overall pipelines with storage constraints.

2. System Implementation

The system architecture for grid data streaming pipelines is shown in Figure 1.

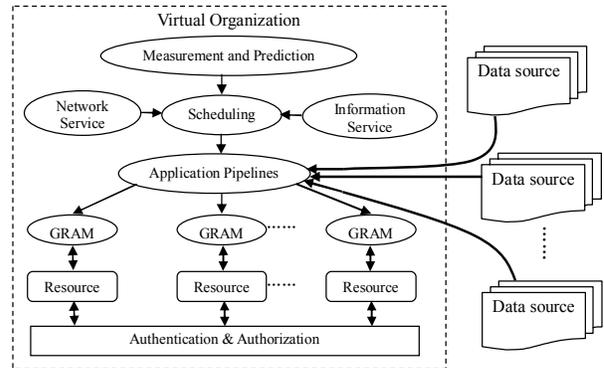


Figure 1. System Architecture

GM(1,1) is applied to make a prediction of performance of tasks, which provides information for heuristic task scheduling with genetic algorithm.

Data supply plays a key role in achieving high throughput, and it must be storage aware to avoid data overflow while guaranteeing no task will starve for data. Repertory policy is introduced to control the start and end of data transmissions, and then storage usage will just be reasonable. To cooperate with the consuming speeds, data transmission rates will be adjusted by tuning parallelism (p) in GridFTP.

3 Performance Evaluation

A case study is illustrated in Figure 2 and some experimental results are provided.

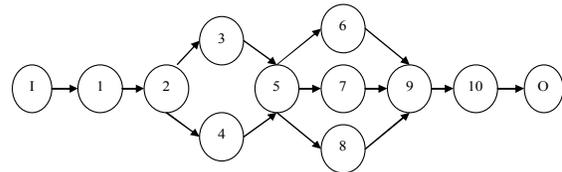


Figure 2. A Case Study

Job mapping is made with GA to keep balance among tasks as much as possible and avoid performance bottleneck. Data supply is another key

factor to gain high throughput, which is designed to guarantee just enough data while avoiding redundant backlogs of interim products of each tasks, in order to make full use of computational capacity with just reasonable storage and bandwidth. Actually, job mapping and data supply must be storage aware, for available storage is usually small compared with the high volumes of data to be processed. Some experimental results are provided as follows.

3.1. Large storage without repertory strategies. In this case, data are supplied continuously and this scheme is not storage aware. With two different parallelism of GridFTP, the same throughputs are obtained, with too many backlogs for there are some bottlenecks in the pipelines.

Table 1. Experimental Result I

p	Backlog of Data									Data Throughput
	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	
8	0	334	335	0	332	0	0	0	0	165
2	0	84	85	0	85	0	0	0	0	164

3.2. Bottlenecks are eliminated. When the pipeline is kept balanced, the throughput will be increased dramatically with enough data supply while inadequate data supply will also hinder the throughput.

Table 2. Experimental Result II

p	Backlog of Data									Data Throughput
	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	
8	0	167	168	0	166	83	84	0	0	248
2	0	0	0	0	0	123	123	0	0	124

3.3. Limit storage with repertory strategies applied. In this case, data supply is not continuous but intermittent for it is controlled by the repertory strategy. It is storage aware for data supply will be stopped or resumed according to storage usage. The same throughput is achieved with much less intermediate backlogs, i.e., the same amount of data can be processed with much less storage.

Table 3. Experimental Result III

p	Backlog of Data									Data Throughput
	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	
8	0	12	13	0	11	0	0	0	0	165
2	0	11	12	0	10	0	0	0	0	164

Backlogs of data at each task do not increase linearly over time as illustrated in Figure 3. It is obvious that in a long run, the storage usage will not increase linearly, which is very important in the data streaming applications.

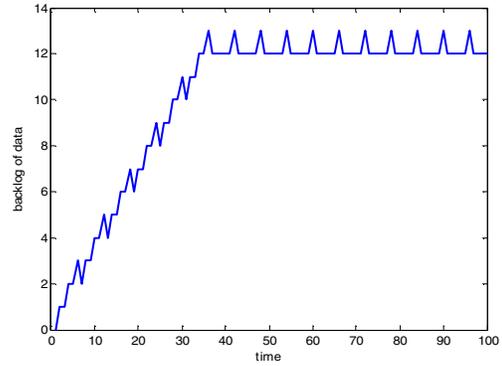


Figure 3. Backlogs of Data with Repertory Strategies Applied at Task s_2 ($p=8$)

4. Conclusions

The *the-faster-the-better* principle does not apply to data streaming applications in some situations. Balances among dependent tasks of a pipeline become essential to achieve high performance. Implementation of data streaming applications not necessarily requires large data storage capability. Storage aware resource allocation using repertory strategies is proposed in this work so that applications can gain high data throughput with relatively small storage.

Actually, job mapping and data supply are of the same importance for the throughput of data streaming applications in the form of pipelines. Bottlenecks in the pipeline must be eliminated if possible; data supply must coordinate with the processing capacity of the pipeline while being storage aware. In this way, high throughput can be achieved

References

- [1]. J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975.
- [2]. L. Man, "An Application of GM(1,1) Model: the Prediction of flight safety", *J. Gray System*, Vol. 1, No. 1, pp. 99-102, 1989.
- [3]. B. Allcock, J. Bester, J. Bresnahan, A. L. Chervenak, I. Foster, C. Kesselman, S. Meder, V. Nefedova, D. Quesnal, and S. Tuecke, "Data Management and Transfer in High Performance Computational Grid Environments", *Parallel Computing*, Vol. 28, No. 5, pp. 749-771, 2002.
- [4]. I. Foster and C. Kesselman, "Globus: A Metacomputing Infrastructure Toolkit", *Int. J. Supercomputer Applications*, vol. 11, No. 2, 1997, pp.115-128.