Proceedings of the 5th WSEAS Int. Conf. on SIMULATION, MODELING AND OPTIMIZATION, Corfu, Greece, August 17-19, 2005 (pp67-71)

A study on improving data transmission capability in grid using compression technology

ZHAO Hong , LIU Mi-xia, ZHANG Qiu-yu,YU Dong-mei
School of Computer and Communication, Lanzhou University of Technology
Lanzhou, 730050, P.R. China

Abstract: Network transmission capability is a key factor that affects the performance of grid. This paper presents an efficient data transmission method in which the serialized data is compressed before and decompressed after it is transmitted through networks. The SOAP extended intercept service is applied as a mean for the compression and decompression process, as a result, the data flow in networks is slimmed and the data transmission time is shortened. Experiments show the proposed method can apparently improve the transmission capability of the networks without additional cost on hardware or sophisticated algorithms.

Key words: grid; compress; Web Service; SOAP; intercept; transmission;

1 Introduction

The Internet technology realizes communication and information exchange among computers distributing all over the world. But the grid computing technology tries to make sharing all resources (computational resources, storage resources, communication resources, software resources, information resources, knowledge resources etc.) of the Internet and cooperating with others on the Internet possible, which makes the Internet become a giant super computer, and provides plug and play services for the users. The grid computing has technique advantages of high performance, integration and resources sharing. On the grid, its nodes have ability of heterogeneous autonomy; its resources evolve dynamically; and its tasks are distributed but cooperated with others. This eliminates resources and information isolated islands in the Internet. So the grid is considered as the next generation Internet technique.

There are two main architectures of grid: one is the layered hourglass model that focuses on protocol[1,2]; the other is OGSA (Open Grid Service Architecture) that focuses on service[3]. OGSA is the most important and new kind of grid architecture after five layers grid architecture considered next generation grid architecture. The layered hourglass architecture focuses on shared physical resources and their provided services. OGSA focuses on shared services. Service concept that includes all sorts of computing resources, memory resources, network, procedure and database is wide in OGSA. From resource to service, it unifies resources, information and data and is benefit for realizing flexible, consistent and dynamic share mechanism. This makes distributed system administration have standard interface and behavior. All are considered as grid service in OGSA. So grid is extensible set of Grid Services, namely grid={grid service}. In the Grid Services there are teams of interfaces that resolve problems of discovering services, creating services dynamically, managing life

period and notifying.

It is clear that Grid Services equal interfaces/behavior adding service data described as Figure 1.

From Figure 1 we can see that grid sends service data to request side or receives service request through Grid Services interface or other interfaces.

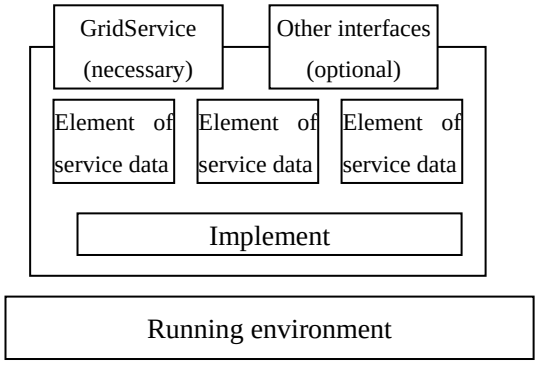

Fig. 1 Grid services description

2. The Process of Services Request and Services Respond in OGSA

OGSA accords with the standard of WebService frame. WebService is a software application that can be identified by URI (Uniform Resource Identifier); its interfaces and bindings can be described and discovered by XML (eXtensible Markup Language); and it can exchange data with other application instance based XML message through Internet protocols[1,4]. GridService is a spread of WebService, but it can support temporary service instance, and can be created and deleted dynamically.

After serializing service request and response become characters flow described by XML, and are transferred according to SOAP (Simple Object Access Protocol) standard in OGSA, as shown in Figure 2.

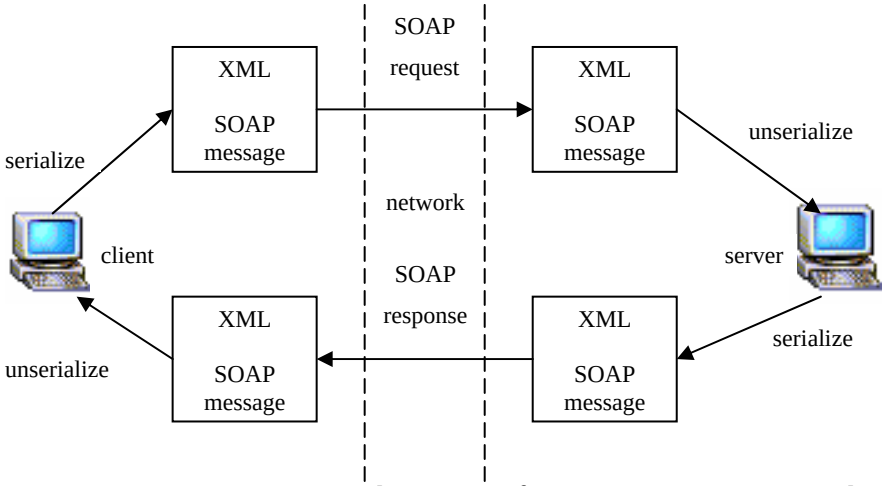

Fig. 2 Service request and response of transporting process in grid

In Figure 2, service request sent by client is serialized as XML characters sequence, and is transported according to SOAP standard. When arriving at server end, it is received by server after unserializing.

3. Improve data transmission capability using compression technology

All resources are distributed at different computers in grid. Network transport capability is one of the most important factors affecting grid performance using these distributed resources. The solution process of a problem is shown as Figure 3.

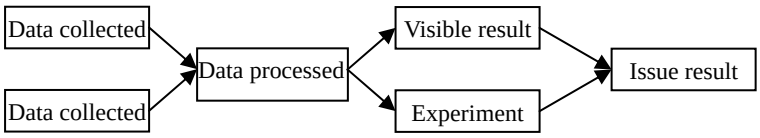

Fig. 3 The solution process of a problem

Figure 3 shows that there are six steps for the solution process of a problem. Each step may be different place that is remote from the other. There are lots of data transported among these different locations. It is clear that network transmission capability is the key factor affecting solution of this problem.

There are two methods usually used for improving network transport performance. One is to upgrade facility and widen network bandwidth; the other is to optimize network capacity, distribute flux reasonably and select the best way to transfer data through certain models and algorithms. Genetic algorithm is adopted in [5] for improving network transport performance.

The former needs to upgrade facility that is expensive; the latter needs to predict network flux, but the accuracy is low because of the randomicity of data transferring in network.

In OGSA, SOAP transports serialized XML character flow. Compression ratio of character is high, and using compression means can make its dimension reduce about 80 percent, which makes data flux between client and server reduce the similar percent. Compared with the two methods above, this is the cheapest and effective method for improving network transport performance undoubtedly. For instance, server sends 1M serialized data to client, after compressing, it is 0.2M. Then time of data transport is reduced, and network jam is avoided.

SOAP is a light-weight and extensible message exchange format. It is not tied to any specific programming language, platform, or transport mechanism, enabling the exchange of information across disparate run-time environments. Although HTTP is the most widely used transport layer for SOAP payload, other protocols such as FTP or SMTP can also be used. The use of XML and HTfP with the SOAP protocol makes it well suited to serve as an interoperable communication protocol on the Grid.

Using SOAP extensible intercepting mechanism condenses serialized XML character flow, which can control them before transferring request and response of SOAP. The principle figure is shown as Fig. 4.

After serializing the data, it is compressed before transferring through network, when arriving the data is extracted at the destination, and then it is adversely sequenced. Transferring data in the network is just as 20 percent as the original in this way. Though compressing data needs extra overhead, it is valuable compared with obtaining network performance.

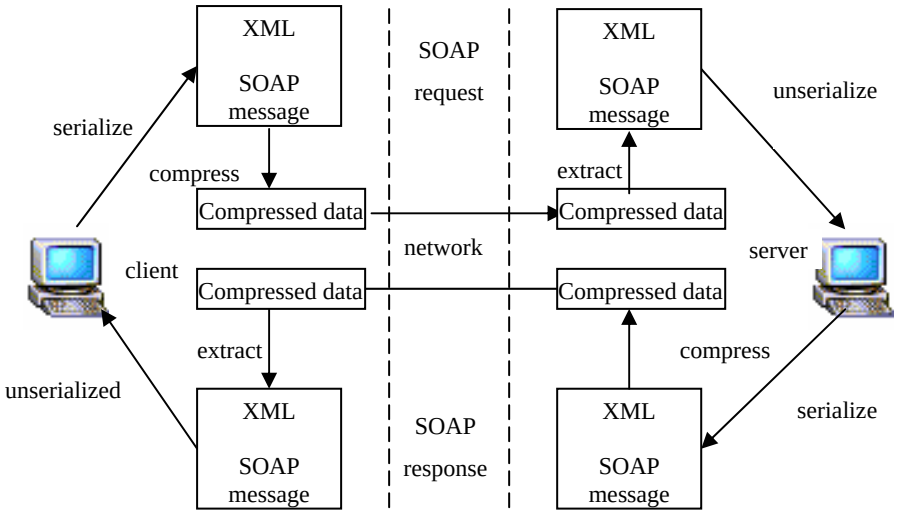

Fig. 4 The opportunity of data compression and decompression

4. Instance and Performance Analysis

Client asks for all records of orders table from SQL Server Northwind database. The data is 454K after being XML from server to client. It is 91K after compressing using free software NZIPLIB. The comparison between compression and none compression is shown in table 1.

Tab. 1 Transport data's comparison between compression and none compression

Adopted measure	Data dimension（KB）	Transferring time（ms）
None compression	454	23923
Compression	91	12143

Comparison between CPU using is shown in Figure 5 and Figure 6.

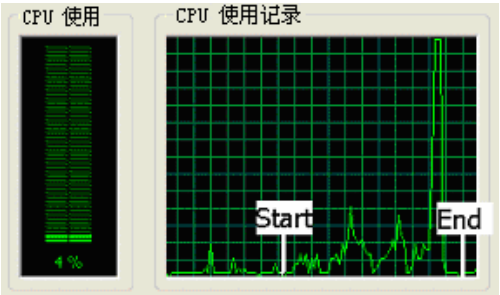

Fig. 5 CPU usage in non-compression

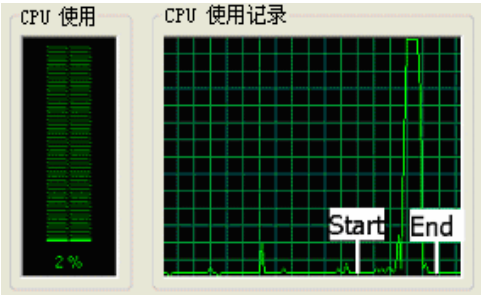

Fig. 6 CPU usage in compression

In this instance, service response is provided, that is data compression from sever to client; at the same time service request is also provided, because transferred data is small quantity from client to server, and performance is not improved obviously, so performance comparison is abridged here.

Experiments show that when transferred data is small quantity, the performance is poor, but when transferred data is exceed about 300KB, the performance is notability, the transferring time is about 1/2 of the no compressed

method.

5. Conclusion

Grid computing carries out among many kinds of heterogeneous computer systems, which forms high performance associated computing environment. However, transport performance of network is a main factor of effecting grid computing. Using SOAP extensible intercepting mechanism, compressing data before transport of SOAP request and response, and reducing data flux of network is a simple, cheap and high efficiency means for improving grid computing performance.

Acknowledgments

This research was supported by the Science Foundation, Gansu Province, P.R China under project No. 3ZS041-A25-020, this support is gratefully acknowledged.

References:

[1] Dou zhihui, Li lisan, Chen Yu, Liu Peng, Huang Zhenchun, He Chuan, Fang Xing. (2002). Grid Computing and its prototype execution research. Computer Science. 29(8):1~5

[2] I Foster, C Kesselman, S Tuecke. (2001). The anatomy of the grid: Enabling scalable virtual organizations. International Journal of Supercomputer Applications. 15(3): 200~222

[3] Tim Banks. draft_ggf_ogsi_gridserviceprimer-recent. http://www.gridforum.org/ogsi-wg/drafts, 2004-03-11

[4] Dou zhihui, Li Sanli, Chen Yu, Liu Peng. (2002). Grid Computing. Beijing: Tsinghua University Publishing Company.

[5] Xu Fushui, Lin Xiaohui. (2003). Optimized research of allocating capability and flux in network. Journal of Gansu University Technology, 29(2): 77~80

[6] Rainsoft. XML Web service compress. http://www.dotnetjunkies.com/Article, 2004-01-12

[7] J Frey, T Tannenbaum, I Foster et al. (2002). Condor-G: A computation management agent for multi-institutional grids. Cluster Computing, 5(3): 237~246

[8] I Foster, C Kesselman. (1997) Globus: A Metacomputing Infrastructure Toolkit. International Journal of Supercomputer Applications, 11(2): 115~128

[9] *Zhang Qiuyu, Yuan Zhanting, Xiao Haidong. (2003). Distributed computing policy analysis based on Grid nodes. Journal of Gansu University Technology,* 29(2): 86~89