GlobalStat: A Statistics Service for Diverse Data Collaboration and Integration in Grid

Shaomei Wu, Zhihui Du
Department of Computer Science and Technology
Tsinghua University, Beijing, 100084, P. R. China
{wusm@mails, duzh@}tsinghua.edu.cn

Abstract

The rise of extensive computing has spawned an urgent need of information integration and collaboration. In this paper, we present a Grid Statistics Service (GlobalStat), a utility Grid service designed to provide an easy, safe, scalable and stable solution to integrate diverse data and get a global Grid view by statistics methodology. The GlobalStat between Grid nodes is organized in "semi-P2P" structure, so that it can be deployed in a wide-area, multi-institutional, heterogeneous and highly dynamic environment. The data hook (Statistics Investigation Program, SIP) and the elaborate policies designed for GlobalStat make it semantic-free to the statistics target, suitable for both application-origin or user-origin tasks.

1. Introduction

With a continuously development of Grid, more and more resources, applications and computing data are to be integrated with all shapes and sizes, from disparate organizations[2]. Much work has been done to address the Grid data integration and collaboration issues, such as OGSA-DAI [4] and Mobius[5].

Data integration is an important and necessary method to investigate and understand Grid system globally. Actually, traditional monitor technology can sometimes provide a global view of Grid as well. However, since the resources coordinated in Grid are various in types, it is still hard for current monitor to overcome the semantic limitation of the data it acquires. In most case, it is possible be cognized and process only the system information. While GlobalStat provides a wild semantics freedom, it can globally discover user-defined data oriented by human, application as well as system.

As a classical way to concentrate the scattered information, statistics is especially appropriate to be applied in highly dynamic Grid environment, for various resources, organization and application are usually unstable and irregular in Grid, they could only appear some macrostructure and global behavior patterns in statistics level.

In all, GlobalStat is a statistics based Grid data integration service, which enables a simple mechanism of data integration and knowledge discovery with the method of statistics. We have also designed a *semi-P2P* structure to distributed the kernel work of statistics over a group of Grid nodes so that the performance bottleneck is effectually avoided.

The rest of this paper is organized like this: section 2 describes the general functions of GlobalStat; section 3 presents the design details, including the system framework, composition, functional model, and protocols in GlobalStat; section 4 gives the description on a prototype of GlobalStat; finally, we conclude our work and outline future works in section 5.

2. The Role of GlobalStat

GlobalStat plays an important role in the Grid data

integration and knowledge discovery and it is also attractive to common users for its simple operation and strong adaptability for individual statistics requirements.

2.1 The Concept of GlobalStat

The Grid statistics referred here is a generic data collecting and analysis process. Similar with routine statistics, Grid statistics is organized by a statistics originator who designs the content of statistics and a group of statistics participants who fulfill the "statistics questionnaires" independently. In GlobalStat, these two jobs are performed by **SOM** (*Statistics Originator Module*) and **SPM** (*Statistics Participant Module*) separately. GlobalStat accepts two modes – manual or automatic – to create a statistics.

In Manual mode, statistics requirements are described by the user into GlobalStat with a interface called New Statistics Guide. Correspondingly, the Automatic mode of statistics generation is employed by writing code with GlobalStat APIs inside the program which needs statistics service, so, it is cater to experienced programmers, such as the developers of Grid application.

Meanwhile, the working mode of data collecting decides the way GlobalStat works in statistics participant ends. Manual mode is for people-oriented data, with questions that should be answered by the user; and Aotumatic mode, is required in computer-oriented statistics, in which the target data could be mined by GlobalStat automatically.

Determined by the characteristics of Grid environment, GlobalStat is a novel Grid data service with these important features: 1) large-scale and extensive data collaboration, especially statistics; 2) effective and real-time data-integration and knowledge discovery; 3) semantic freedom for statistics problem; 4) high security protection.

2.2 Example Applications

Before delving into the design of GlobalStat, we outline three classes of applications to motivate the system and its architecture.

Grid voting: Since the public opinion is playing a more and more important role in our lives, we need an easy way to originate and release a personal poll over Grid. Grid voting is an easy conducted-and-customized poll driven by GlobalStat, so that it can quickly show the most popular ideas on some special aspects.

Grid Application Monitor: So far, it is still difficult to observe the status of specific applications. A monitor is required to supervise what is going on in the distributed applications and even debug them just as we debug a local program. Now this desire is satisfied by GlobalStat with the operation to applications' middling data.

Large-scaled computing collaborating and data share: Most of data-intensive programs run in a rather long period with high computing load, and the tremendous amount of data accumulated decreases the data collaboration efficiency. With GlobalStat, we appoint an independent node to manage the run-time integration and analysis so as to help not only reduce the data redundancy but also speed up resource and computing cooperation between different computing centers.

3. Design of GlobalStat

In this part, we will give an overview of the architecture of GlobalStat, and show the details in working protocols as well as some special policies.

3.1 System Framework

The system structure of a running Grid statistics is roughly shown in Figure 1.

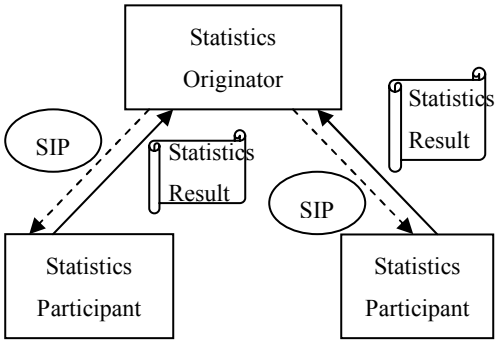


Figure 1. System Working Structure

Every statistics starts after it is initialized in a Grid node named *Statistics Originator* who calls for GlobalStat. And then GlobalStat will collect the scattered data by sending out the *Statistics Investigation Programs* (**SIP**s) to all the statistics participants. The statistics ends when object data are reported back and processed in Statistics Originator. Detailed description about each component will be present afterwards.

As described before, GlobalStat is composed by two independent main modules: SOM and SPM. Functionally, there is another important ingredient − SIP running in statistics participant to collect distributed data initiatively and give response instantly. It is platform-independent, working in a form like a java applet. However, as it is produced dynamically by SOM, we do not consider it as a stable part of GlobalStat.

Typically, a Grid Statistics functions among a SOM and a group of SPMs over the Grid. Figure 2 illustrates a simple scenario for GlobalStat. Here, SOM in Node C creates, organizes a Grid statistics by interacting with SPMs in Node A and Node B while Node A accepts and Node B refuses, so there will be no response in Node B after it checks the StatInvitation. But for Node A, SPM is going to finish a series of tasks, with collaboration from SOM in Node C.

The design of SIP will promoted the efficiency of data integration by actively looking for the inquire data but not passively waiting for the remote data response. On the other side, SIP solutes the syntax problem in application-specific data demanding, by abolishing remote statistics requirement compile.

With a separate SIP container, statistics participants can download the SIP initiatively and selectively. This mechanism not only helps the SIP to be transferred more steadily, but also guards against some evil programs' sudden invasion.

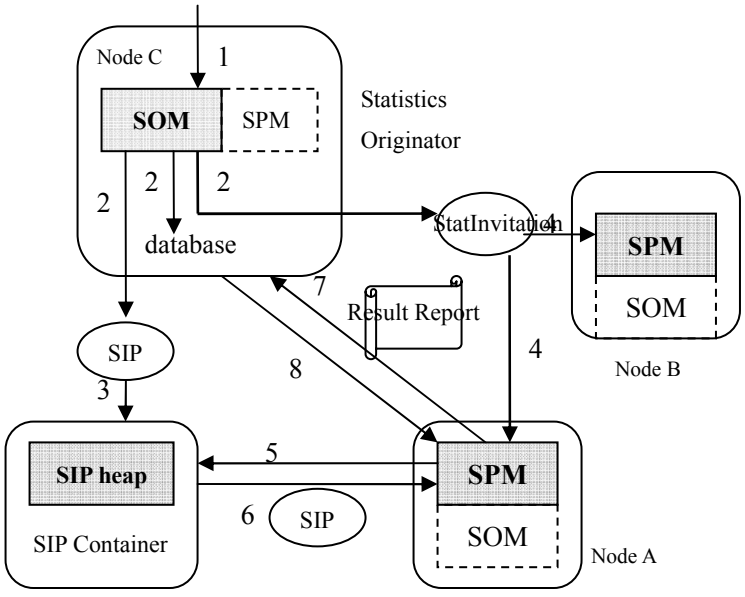
As mentioned before, GlobalStat can also work with a group of SOMs' cooperation for a single statistics, in order to keep away from SOM bottleneck. More details in section 3.3.

3.2 Components of GlobalStat

3.2.1 Statistics Initialization and Management: Statistics Creator Module (SOM)

Work of SOM is composed mainly by three parts:

- before statistics, SOM has to communicate with user to understand the statistics requirements, and then, create the SIP which will be distributed to participants as soon as statistics begins;
- during statistics, SOM function as the data center,



(1) User submits the statistics requirements
(2) SOM generates SIP, database, StatInvitation by StaRequirement
(3) SOM sends SIP to SIP Container
(4) SOM releases new StatInvitations
(5) SPM accepts new statistics, connects to SIP Container.
(6) SIP is delivered to corresponding SPM
(7) SPM returns the Result Report to the SOM of statistics originator.
(8) SOM get result, response ack to SPM.

Figure 2. A usage scenario for GlobalStat

taking charge of incepting the SIP reports for instant processing;

- after data collecting, SOM has to visualized the statistics results, and in some case response them to statistics participants again.

The first step of creating a new statistics is to acquire the requirements from GlobalStat client. In manual mode, it is realized by New Statistics Guide while in automatic mode, by Statistics Creation API Library.

The New Statistics Guide is applied to simplify the process of statistics creation and increase the validity of questionnaires. It is a template-based statistics setup interface, which generates a standard questionnaire by asking the user some questions about statistics subject (human, object or affair) and related specialties (which help to identify the subject). Then, after some modification made or limitation set by user, a formal statistics questionnaire is ready.

Automatic mode is used in programmable statistics initialization with GlobalStat APIs calls embedded in the program code. The data inquiry in this mode is constant, definite and well-predictive. So, the statistics requirements and rules can be explicitly expressed in the code of GlobalStat client application. The Statistics Creation API library includes some application interfaces for statistics orientation, SIP generation, data mining and allocation, and they will be linked into the client application during compiling.

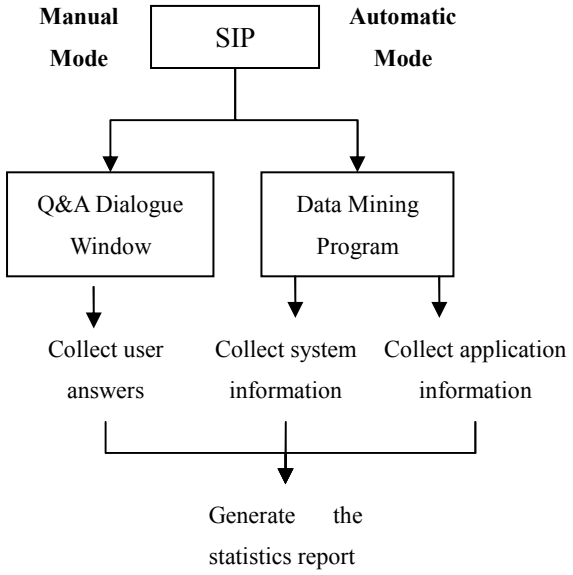


Figure 3. the function of SIP. The working mode of SIP is designated by statistics requirements. In manual data collecting mode, the SIP will form a questionnaire window to get the user's answers. While in automatic mode, SIP works without user's attention.

After statistics requirement expression, SOM will continue with other preparation of a Grid statistics, they are: creates a SIP, initializes the statistics database if needed and releases the *Statistics Invitations* (StatInvitation) to the nodes in statistics scope.

3.2.2 Statistics Response: Statistics Inquirer Program (SIP) and Statistics Participant Module (SPM)

The role of SIP is quite worth discussing. After a SPM downloads corresponding SIP from SIP container, it is added into local SIP_queue, and will be run under some policy. Figure 3 presents how SIP works on the node of statistics participant in different data collection modes. While manual data collecting is people-oriented, the automatic mode is designed to detect the status of system, collecting system information or current situation of target application.

Since it is dangerous to capture or monitor the running stack or memory buffer of a user application for the intermediate results, we employ a slightly tricky alternation here for the information inside programs: let the application itself decides what about it could be collected and how to export them. To do this, we design a Data Export API library in SPM which can be called and linked by the statistics participant program, which is to be investigated. Data Export API library provides the GlobalStat client applications an easy, standard and safe way to export their internal data timely and directionally. Such information revelation is under agreement with statistics originator for monitoring or better data collaboration, so, the SIP which is generated by statistics originator knows exactly how and where target data could be found.

The task of SPM is encapsulating the statistics re-

sult, sending them back and scheduling the life of SIPs. All SIPs will be destructed by SPM when it is useless. In a word, SPM is the host and SIP is implemented like a hook to grip information scattered in the remote node.

3.3 Functional Model

GlobalStat organizes the Grid statistics in an evolutional server-client model, in which, the SOM plays the role of "server" and all the participating SPMs act like "clients". Although the job of SPM is rather independent to SOM, it is SOM that the center of the whole statistics.

Since every GlobalStat has both SOM and SPM so that it could be a statistics originator and a participant at the same time. From the Grid view, GlobalStat has a semi-P2P working model, for example, when a GlobalStat in node A accepts a statistics, it may also acquire the Statistics Requirements file from the statistics originator so that clone this statistics to be a new originator. So the first statistics originator needs only broadcast the StatInvitation in a limited zone and it will spread over the Grid quite soon as shown in Fig 4.

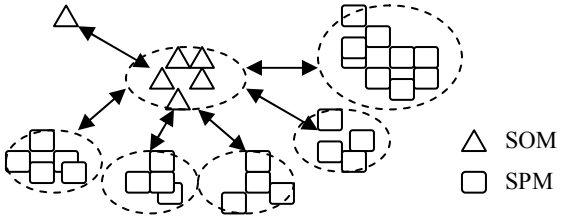


Fig 4. GlobalStat: working in semi-P2P structure. A little group of SOMs collaborate to organize a same statistics and finally they will concentrate all the results together to the original statistics starter.

The advantages of semi-P2P model are significant: 1) it is a good way of load-balancing, solving the performance bottleneck in unique SOM; 2) it boosts the efficiency of statistics notification and data transfer; 3) it enables a statistics to expand its participating scope easily, so that enhances the scalability of GlobalStat; 4) it avoids the concurrent message flooding in Grid, helps to reduce the network load.

3.4 GlobalStat Schema

We proposed some special schemas in GlobalStat, such as semantic solution, security, schedule policy and scope control.

As mentioned above, GlobalStat provides sufficient data **semantics freedom** to object raw data by the implementation of SIP and Data Export API library.

In **security** aspect, the Statistics Test and Verification Modula in SPM is a firewall of GlobalStat, which prevents disturbing from irrelevant or uninteresting statistics. Then, we also encrypt all the result response data with the encryption key along with StatInvitation, so that only the correct SOM would be able to decrypt and understand the data transmitted.

A good **schedule policy** is necessary to run SIPs efficiently and stably in a participant node. In SPM, we setup a *SIP_queue* storing all the living SIPs and schedule it with Weighted Round-Robin algorithm. The weight of each SIP could be associated with urgency or bonus of each statistics task. The schema to destroy outdated SIPs and response result is also well-designed.

Every statistics needs a range of participants as it could not annoy too many people/nodes over Grid. GlobalStat organizes participants by sending out StatInvitations, and we introduce the **statistics participant list** to control the area StatInvition distributed as well as invoke most potential participants farthest. A statistics participant list is like the contacts list in the mailbox. The StatInvitations will only be delivered to whom in this list with the new statistics.

3.5 API Design

As stated before, GlobalStat provides a set of synchronous java client application programming interface, which is consisted by a Statistics Initialization API library and a Data Export API library. For space reason, we do not include details of the API in this paper; instead, the operations of APIs will be described.

Statistics Initialization API library contains some APIs which help to build SIP and create the new statistics, they mainly are:

- Allocate a new statistics with a name specified.

- Set the subject of statistics.
- Set the data collection mode: Manual or Automatic.
- Set the location of object data.
- Create the local statistics database.
- Assign the function which will process the result reports from statistics participants.
- Specify a statistics participant list.
- Set the urgency weight and bonus of statistics.

Data Export API library is a standard channel for client applications outputting their internal data, such operations are offered by it:

- Set the associated statistics name.
- Set the mode of data export: onetime or periodical.
- Set the data exporting period.
- Output the data into the designated place, like the fprintf in C++ language.
- Export data by stream;
- Create a data output database.
- Output the data into an existing database.

4. Implementation

The Grid Meta Statistics Service uses a web service model. Its implementation uses an Apache web service front and a MySQL relational database backend. Up to now, we have implemented a GlobalStat prototype which only supports Manual Mode in statistics creation and data collecting. The service packet is written in Java. Initially, our prototype is installed in the Globus Toolkit 3.0.2 Grid environment and experimentally used for some applications such as “Grid Voting”.

5. Conclusion and Future Work

In this paper, we characterize the large-scale data collecting and integrating as a distinct Grid service component, GlobalStat, which supports extensive data integration and collaboration in dynamic, heterogeneous environment of Grid.

GlobalStat is under construction. This paper presented many of the design elements and algorithms of GlobalStat, several have been implemented. GlobalStat is believed to be very easy to apply to other distributed system, such as Internet. As a utility Grid service, there are still some other complements should be made for GlobalStat, such as authorization and encryption system, metadata description and effective data transmission.

Reference

1. Ian Foster, What is the Grid? A Three Point Checklist. July 20, 2002,
2. Ann Chervenak, Ian Foster, Carl Kesselman, Charles Salisbury, Steven Tuecke, The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets. *Journal of Network and Computer Applications: Special Issue on Network-Based Storage Services, vol. 23, no. 3, p. 187-200, July 2000.*
3. http://www.ogsadai.org.uk/
4. Stephen Langella, Shannon L. Hastings, Scott Oster, Tahsin M. Kurc, Umit V. Catalyurek, Joel H. Saltz, "A Distributed Data Management Middleware for Data-Driven Application Systems", *Proceedings of the 2004 IEEE International Conference on Cluster Computing*, 2005.
5. Gurmeet Singh, Shishir Bharathi, Ann Chervenak, Ewa Deelman, Carl Kesselman, Mary Manohar, Sonal Patil, Laura Pearlman, *A MetaData Catalog Service for Data Intersive Applications*. SC’03.
6. Bill Allcock, Joe Bester, John Bresnahan, An L. Chervenak, Ian Foster, Carl Kesselman, Sam Meder, Veronika Nefedova, Darcy Quesnel, Steven Tuecke, *Data Management and Transfer in High Performance Computational Grid Enviroments.*
7. W. Allcock, A. Chervenak, I. Foster, L. Pearlman, V. Welch, M. Wilde, *Clobus Toolkit Support for Distributed Data-Intensive Science.*