A Hierarchical Model for Distributed Monitoring Data Collection

Rafael Keller Tesser, Phillipe O. A. Navaux
Instituto de Informática, Universidade Federal do Rio Grande do Sul
Av. Bento Gonçalves 9500, Porto Alegre, RS, Brazil
{rktesser, navaux}@inf.ufrgs.br

Abstract

Distributed system monitoring tools provide information which can be used to perform a wide set of tasks. The Grid also can take advantage of monitoring. However, specific tools are needed in order to satisfy its special requirements. DIMVisual is a monitoring data integration model which receives data originated by a set of tools and generates standardized data for visualization. Unfortunately it doesn't include ways of gathering monitoring data from distributed resources. For this reason, we propose a distributed model of monitoring data collectors in order to extend the DIMVisual with the goal of easing its use in distributed systems. The proposed model consists of local data collectors and hierarchically structured aggregators. These aggregators receive monitoring data from collectors and lower level aggregators and send it to a client application. To decrease the transmission of unnecessary monitoring data, the model establishes a subscription mechanism that enables the sending of data from a collector only to components which need it. A prototype is being implemented in order to validate the model through the evaluation of its execution in a Grid environment.

1. Introduction

Monitoring systems can be used to perform several tasks in distributed systems. Among these are fault detection, scheduling, debugging, performance analysis and application execution adaptation. Historical monitoring data can be stored in order to examine the behavior of the system along a period of time.

In a Grid system some inherent characteristics like large scale, dynamic resource availability, geographic distribution, and resource heterogeneity must be taken into account. Therefore, we need specific monitoring tools which should be able to deal with these characteristics.

DIMVisual [11] is a model which aims to integrate and normalize data received from various monitoring tools. The result of the application of this model is data in a format which can be used to visualize the information from the different sources together. This way it's easier to examine correlations between the behavior of the monitored resources.

A monitoring system performs the generation of monitoring data, its processing, distribution and presentation. However, the DIMVisual doesn't deal with the distribution monitoring information, because it doesn't specify a way to collect data from distributed systems. In these systems the data is usually scattered through various places. For this reason the user needs to gather the data previously to the application of the model.

We propose a distributed monitoring data collection model with the goal of easing the use of DIMVisual in distributed systems. This model is composed by collectors and aggregators. The collectors are responsible for gathering the information from different monitoring tools and resources. The aggregators are organized in a hierarchy. They receive data from collectors and lower level aggregators and send it to higher level aggregators and clients. These clients are responsible for the data integration. The model includes a subscription mechanism which allows the data originated from one collector to be sent only to interested components.

Some existing monitoring tools, like the Globus MDS-2 [3] and Ganglia [9], form hierarchical structures but use a pull model for data propagation, not supporting subscription. The Globus MDS-4 [5] provides subscription but is not suited to our use of the data. Other tools support subscription but don't use a hierarchical structure. For example, in MonALISA [8] and NetLogger [7] data is transmitted directly from local producers to consumers.

A prototype of the proposed model is being implemented. We intend to test it in a Grid environment in order to evaluate the model.

2. The DIMVisual Data Integration Model

DIMVisual is a data integration model for visualization. This model receives monitoring data generated by different monitoring tools. It integrates and unifies this data in order to visualize them together, using a visualization tool.

The data integration process has three steps: synchronization, unification and standardization. The first step is necessary because the data comes from different origins. The second is the unification of the identifiers used by the monitoring tools according to a entity hierarchy (cluster, node, process, etc). The last is the conversion from the format used by each tool to one that can be visualized.

There is a DIMVisual prototype which receives files containing data collected by monitoring tools and generates a file which can be used as input for the Pajé [4] visualization tool.

3. Distributed Monitoring Data Collection Model

The DIMVisual doesn't specify a data collection mechanism for distributed systems. This means that the data must be gathered together previously to the application of the model. We propose a distributed monitoring data collection model in order to ease the use of the DIMVisual in distributed systems. The goal of this model is to make it possible to the DIMVisual to access data collected from various components of a distributed system.

The structure of the proposed model is shown in the figure 1. It has two main component types: collectors and aggregators. The collectors gather the information from the monitoring tools. The aggregators are structured hierarchically and receive monitoring data from a set of collectors or lower level aggregators. The received information is forwarded to higher level aggregators or to clients. The clients perform the integration steps of the DIMVisual model, being very similar to the original DIMVisual.

The transmission of data uses a push model. This means that the transmission is started by the component who sends the data. In addition to that, a subscription schema is used to allow the sending of the data provided by a collector only

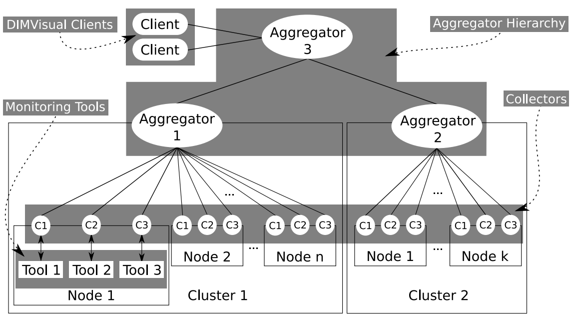

Figure 1. Components of the Distributed Monitoring Data Collection Model

to components which are interested in it. The way the subscription works is explained in the section 3.4.

3.1. Collectors

A collector is a component which gathers data obtained by a monitoring tool. Therefore, there must be one kind of collector for each monitoring tool. The collected data must be periodically sent to the aggregators which are subscribed to them. The collectors must register themselves before they send data to an aggregator. This register allows the aggregators to identify the collectors and the origin of the data they receive. It also allows them to subscribe to the data provided by a collector. The section 3.3 shows how this registration is made.

3.2. Aggregators

The aggregators form a hierarchical structure for monitoring data transmission. They receive data from lower level aggregators and from collectors and forward it to higher level aggregators interested in it. This is done in a way that a client can obtain data about the whole system through the aggregator in the top of the hierarchy.

3.3. Collector Registration

When a collector is initialized it has a set of aggregators in which it must register itself. The aggregator processes the registration by creating an entry in a registry for the collector and forwarding the request to higher level aggregators.

Because of this registration the aggregators don't need to have previous knowledge of the components which are bellow them in the hierarchy. In this way collectors can be added and removed without restarting the aggregators. Moreover, aggregators can be added to the structure in the same way. However, there's the restriction that components must be started before those who intend to send data to them.

3.4. Subscription To Monitoring Data

In order to decrease the transmission of unneeded monitoring data, the model includes a subscription mechanism. This allows clients to select the collectors from which they are interested in receiving data instead of receiving it from every collector in the system.

Clients and aggregators can subscribe to monitoring data. When an aggregator receives a subscription request it registers the requester as interested in the data from the specified collector. If the aggregator hasn't a subscription for the requested data, then it requests a subscription for a lower level aggregator or to a collector in case it's already in the lowest level. The collectors also register the requester in a set of components interested in the data generated by them.

When a client or aggregator doesn't need the data anymore it must cancel the subscription. The cancel request makes the aggregator to remove the requester from the set of interested components for the specified collector. If there's no other component subscribed to receive data from this collector through the aggregator, then it also requests the canceling of its own subscription. This way the cancel subscription requests are propagated toward the bottom of the hierarchy. In case there are no other subscriptions related to the collector in the path, the collector will be reached. Then it will remove the subscription of the proper aggregator from its registry.

With the subscription mechanism, a collector sends the data it generates only to the aggregators registered as its subscribers. When an aggregator receives data it checks the identification of the collector who sent it. Then this data is sent only to the components subscribed to receive it through him.

3.5. The DIMVisual Client

The DIMVisual client is in charge of the monitoring data integration process. It works in a similar way to the original DIMVisual. The greatest difference is that the user doesn't need to put together all the data before beginning the process. Instead, the data is provided by the monitoring data collection model we are proposing. The result of the processing made by the client is integrated and standardized data in a visualizable format.

The client must ask an aggregator for the registry of the collectors it gives access to. This way the client discovers the available data sources, the resources from which data can be obtained and the collector identifiers. These identifiers are used in the subscription requests and to distinguish the origin of received monitoring information.

4. Related Work

The Open Grid Forum (former Global Grid Forum) produced a specification called Grid Monitoring Architecture (GMA) [12]. It's composed by producers, consumers and directory service, as shown in figure 2. The producers register themselves in the directory service. The consumers query this service to find the producer of the data they want. In the GMA the consumers receive data directly from the producers.

There are several monitoring tools which can be used in a Grid environment. Some of them where developed specifically for the grid, others were not. Some of these were developed to monitor other kinds of system or the interconnection network.

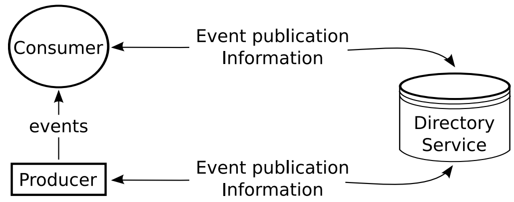

Figure 2. Components of the Grid Monitoring Architecture (GMA)

Ganglia [9] is a monitoring tool initially aimed for clusters. Although it is also used in the Grid [10]. Inside an administrative domain Ganglia transmits data using multicast. This way every node has all the monitoring data from that domain. To provide access to data from different administrative domains ganglia uses a hierarchical tree of republishers. Unlike our proposed model, which uses a push model, polling is used by the services which compose the hierarchy to discover the availability of new data.

Monitoring and Discovery Service (MDS) is a Globus Toolkit component whose main concerns are resource discovery and decision making based in monitoring information. The MDS version 2 [3] uses a hierarchical structure do discover and transmit data. The MDS version 4 [5] uses a simpler structure and relies on web services. This version also supports subscription for receiving data.

MonALISA [8] is a monitoring system whose main focus is to provide information for system management. Despite it being capable of receiving data from various monitoring tools, the data is not normalized. This makes difficult the correlation of information from different sources. As our model, MonALISA supports subscription to data. However, the consumers connect themselves directly to the data providers.

GridICE [2, 1] is a monitoring tool whose architecture is structured in five layers: measurement, publication, data collection, detection/notification and data analysis and presentation. It uses the MDS version 2 in the publication layer. Because of this it doesn't support subscription to data. Other difference from our proposed model is that only registration information is propagated through the hierarchy. This information is used by clients to access the data source directly.

NetLogger [6] is a tool for instrumenting software in order to generate monitoring events. This tool allows the definition of one host to receive the monitoring data. The NetLogger Activation Service [7] allows the control of the instrumentation level of NetLogger. It also adds the ability to subscribe to data. However, a way of forming a hierarchical structure for transmitting the data is not specified.

Network Weather Service (NWS) [13] is a tool whose goal is to provide historical information and performance forecasting of the network and the computers from a distributed system. Its architecture is centralized in relation to the directory service and it doesn't support subscription to data.

Resource Monitoring Framework (RMF) [14] is a tool which makes scheduling decisions based in monitoring data and in the results of a performance predictor. Its structure is very centralized because of the database and the directory service.

Zanikolas and Sakellariou [15] proposed a taxonomy of Grid monitoring systems. According to their taxonomy the DIMVisual extended by our model is classified as a level 3 system because it has producers (collectors) and republishers (aggregators) which can be distributed in an arbitrarily formed hierarchy.

5. Conclusion

Distributed systems can use monitoring tools to perform several tasks related to system management or to the development and execution of applications.

DIMVisual is a monitoring data integration model for visualization which integrates data received from various tools. The result is standardized in a format which can be used for visualization. However, this model doesn't specify a way to collect distributed monitoring data.

In order to ease the use of DIMVisual in distributed systems we propose its extension through a distributed monitoring data collection model composed by local collectors and by hierarchically distributed aggregators. The collectors gather the data generated by monitoring tools and send it to a set aggregators. The aggregators receive data from collectors or lower level aggregators and send it to higher level aggregators or to clients. A push model is used to transmit the data, that is, the communication is started by the sender. There is also a DIMVisual client which receives data from the collector model and does the integration in a similar way the original DIMVisual does. The model also includes a subscription mechanism to transmit data from one collector only to the components which are interested in it.

A prototype implementation for evaluation of the model is being made. We intend to test it in a Grid environment.

References

[1] C. Aiftimiei, S. Andreozzi, G. Cuscela, G. Donvito, V. Dudhalkar, S. Fantinel, E. Fattibene, G. Maggi, G. Misurelli, and A. Pierro. Recent evolutions of gridice: a monitoring tool for grid systems. In *GMW '07: Proceedings of the 2007 workshop on Grid monitoring*, pages 1–8, New York, NY, USA, 2007. ACM.

[2] S. Andreozzi, N. D. Bortoli, S. Fantinel, A. Ghiselli, G. L. Rubini, G. Tortone, and M. C. Vistoli. Gridice: a monitoring service for grid systems. *Future Gener. Comput. Syst.*, 21(4):559–571, 2005.

[3] K. Czajkowski, C. Kesselman, S. Fitzgerald, and I. Foster. Grid information services for distributed resource sharing. In *HPDC '01: Proceedings of the 10th IEEE International Symposium on High Performance Distributed Computing*, page 181, Washington, DC, USA, 2001. IEEE Computer Society.

[4] J. C. de Kergommeaux and B. de Oliveira Stein. Pajé : an extensible environment for visualizing multi-threaded programs executions. *Euro-Par 2000 Parallel Processing, Proc. 6th In-ternational Euro-Par Conference*, 1900:133–140, 2000.

[5] I. Foster. A globus toolkit primer, 2005.

[6] D. Gunter, B. Tierney, B. Crowley, M. Holding, and J. Lee. Netlogger: A toolkit for distributed system performance analysis. In *MASCOTS '00: Proceedings of the 8th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, page 267, Washington, DC, USA, 2000. IEEE Computer Society.

[7] D. Gunter, B. L. Tierney, C. E. Tull, and V. Virmani. On-demand grid application tuning and debugging with the netlogger activation service. In *GRID '03: Proceedings of the Fourth International Workshop on Grid Computing*, page 76, Washington, DC, USA, 2003. IEEE Computer Society.

[8] I. C. Legrand, H. B. Newman, R. Voicu, C. Cirstoiu, C. Grigoras, M. Toarta, and C. Dobreand. Monalisa: An agent based, dynamic service system to monitor, control and optimize grid based applications. In *Computing in High Energy and Nuclear Physics conference (CHEP2004)*, 2004.

[9] M. L. Massie, B. N. Chun, and D. E. Culler. The ganglia distributed monitoring system: Design, implementation, and experience. *Parallel Computing*, 30, 2004.

[10] F. D. Sacerdoti, M. J. Katz, M. L. Massie, and D. E. Culler. Wide area cluster monitoring with ganglia. In *2003 IEEE International Conference on Cluster Computing. Proceedings.*, pages 289–298, 2003.

[11] L. M. Schnorr, P. O. A. Navaux, and B. de Oliveira Stein. Dimvisual: Data integration model for visualization of parallel programs behavior. *ccgrid*, 00:473–480, 2006.

[12] B. Tierney, R. Aydt, D. Gunter, W. Smith, M. Swany, V. Taylor, and R. Wolsky. A grid monitoring architecture, 2002. Disponível em: <http://www.ogf.org/documents/GFD.7.pdf>. Acesso em: Maio 2008.

[13] R. Wolski, N. T. Spring, and J. Hayes. The network weather service: a distributed resource performance forecasting service for metacomputing. *Future Generation Computer Systems*, 15(5–6):757–768, 1999.

[14] R. Yang and M. D. Theys. Rmf: resource monitoring framework for integrating active and passive monitoring tools in grid environments. *J. Parallel Distrib. Comput.*, 65(11):1419–1428, 2005.

[15] S. Zanikolas and R. Sakellariou. A taxonomy of grid monitoring systems. *Future Generation Computer Systems*, 21(1):163–188, January 2005.