Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 45 (2015) 815 – 822
Procedia Computer Science

International Conference on Advanced Computing Technologies and Applications (ICACTA-2015)

Study on Grid Resource Monitoring and Prediction

Sreeja S R[a], Sangita Chaudhari[b,*]

[a]*Department of Computer Science and Engineering, A.C Patil College of Engineering, Kharghar, Navi Mumbai*
[b]*Department of Computer Science and Engineering, A.C Patil College of Engineering, Kharghar, Navi Mumbai*

Abstract

Grid is a collection of heterogeneous systems which share the computing power and storage capacity over the internet. Grid applications usually involve large amounts of data and/or computing and often require secure resource sharing across organizational boundaries. Thus they cannot be easily handled by todays Internet and Web infrastructures. Monitoring the resources in a grid system is necessary to determine any performance problems and for tuning the system for the purpose of optimizing its performance. Prediction data about the resources and applications are needed for managing the various grid resources more effectively. This will also help in scheduling the jobs more efficiently. Various prediction and monitoring systems are available which perform either prediction or monitoring or both for grid systems. Since the grid systems they aim at are different, they have very different characteristics, architectures, and use different techniques. Predicting the dynamic availability of resources such as CPU time, network bandwidth, and disk bandwidth is vital to load-balancing. In this paper, a comparison of various prediction and monitoring systems for grid environment is done.

Peer-review under responsibility of scientific committee of International Conference on Advanced Computing Technologies and Applications (ICACTA-2015).

Keywords: Grid Computing; Grid Resource Monitoring; Grid Resource Prediction.

1. Introduction

A distributed system can be considered as a single virtual machine built by an interconnected collection of heterogeneous systems which share their resources. Grid is a large-scale generalized distributed system that can scale to internet-sized environments with machines distributed across multiple organizations and administrative domains. Grid is defined as "coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations"[1]. A grid system requires managed sharing of its resources. In grid environment, resources

Peer-review under responsibility of scientific committee of International Conference on Advanced Computing Technologies and Applications (ICACTA-2015).
doi:10.1016/j.procs.2015.03.161

may dynamically join and leave. This may result in the unavailability of resources or sometimes the resources allocated will not be the best for performing a particular task. Grid enables organizations to transparently integrate, streamline, and share dispersed, heterogeneous pools of resources into one combined system in order to deliver the service at the agreed-upon levels of application efficiency and processing performance. Grid helps to do resource sharing. Resources include computers, storage, sensors and networks. Sharing in grid environment is conditional and it is based on some factors like trust, policies and negotiation.

2. Grid Monitoring and Prediction

Grid consists of a large number of components and users which increase the vulnerability to faults and failures. Resources, users and applications in the grid should be monitored appropriately. The mechanisms used for this should be capable of detecting conditions which will lead to bottlenecks, faults or failures. Grid monitoring provides a robust, reliable and efficient grid environment. It focuses on measuring and publishing the state of grid resources at a particular instant. Grid monitoring services should make sure that monitoring is taking place in an end-to-end manner so that all the components in the environment are monitored. The components which are getting monitored include software like applications, services, processes and operating systems, host hardware like CPUs, disks, memory and sensors and networks which include routers, switches, bandwidth and latency. Basic grid Monitoring Architecture is shown in figure. It consists of three types of components: [1]

- Producer: any process that produces performance data (the event source)
- Consumer: any process that receives performance data (the event sink)
- Directory Service: a service supports information publication (for the producer) and discovery (for the consumers).

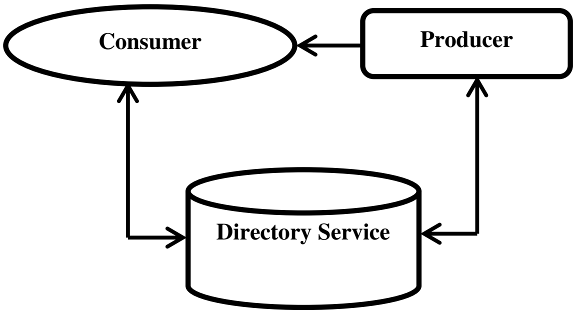

Fig. 1 GMA Architecture

Monitoring of resources is required to understand the performance, to identify the problems and to tune the system for better performance. Fault detection and recovery mechanisms make use of the monitoring data to determine whether different parts of an environment are functioning correctly or not. Monitoring data is also used to decide whether a component has to be restarted or a service request has to be redirected. Many systems have been developed for performing the monitoring task in grid systems. Some of them are visPerf, Ganglia, GridICE, GHS etc.

Although monitoring plays a very important role in increasing the performance of the system, frequent monitoring will consume network bandwidth and CPU time. Prediction plays a very important role in such a scenario. Prediction is a service that can forecast performance using monitoring data as input. These predictions could in turn be used by a scheduler to determine which components to select for allocation. Prediction process will help to perform the tasks of scheduling and allocation more effectively. Monitored historical data will be used and analysed by the prediction systems in order to predict the future values. Prediction systems use different prediction models to generate a prediction. They may make use of existing historical data or they may have their own methods for collecting the training data. Some systems were developed which perform both monitoring and prediction for grid systems. It is helpful to study the behaviour of various monitoring and/or prediction systems in order to get an overview of them. Section 3 explains two Prediction Systems. This is followed by systems which perform monitoring. Some of them have prediction service also integrated into them. Finally a table which compares all these tools is provided.

3. Prediction Systems

Predicting and analyzing the performance levels of grid resources play a key role in the Computational grid environment. Performance predictions made at run time can be used to do effective application scheduling. Grid resources provide widely varying performances to applications. This variation occurs because of the contention between applications. So schedulers must be able to predict the performance that a resource can deliver to a particular application. These predictions will help the scheduler to choose the best combination of resources from the resource pool so that application can provide maximum level of performance. There are various prediction systems available like Prophesy, RPS (Resource Planner System), adaptive scheme for prediction etc. In this section, the characteristics, advantages and limitations of Prophesy Prediction Tool and Adaptive Scheme for Prediction are explained.

3.1. Prophesy

It is a performance forecasting tool based on a prediction modelling. It was developed as a part of RPS (Resource Planner System) that was used with a gravitational-wave physics experiment. Prophesy consists of three parts: automatic instrumentation of applications, databases for archival of information, and automatic development of performance models. There exists a model builder in Prophesy which will produce predictions based upon performance data. Historical performance data will be collected and stored in databases. Prophesy predicts the best site to execute an application. It requires several data such as application name, the semantic parameter names and their values, and the list of available resources etc. Query will be sent to the Prophesy server. Prophesy will rank the given resources and will return that information. If the historical data of the application executed at the particular site is unavailable, then that site will be ranked as the lowest. Ranking is done based on predicted execution times. Sites with less execution time will get higher ranks.[2].

Performance of Prophesy when compared with random site selection provided the best site selection with less overhead. Prophesy works for a dedicated parallel system instead of a non-dedicated distributed environment. It uses only the curve fitting method of prediction. Pre-existing data is required for doing the prediction. Since the prediction is done statically, it may not be accurate always.

3.2. Adaptive Scheme

Adaptive scheme requires less storage and computational power and provides continuous prediction values even in the absence of periodic patterns. It gives long-lasting prediction by adjusting the confident level of patterns. Analysis of time series data and a smoothed algorithm are used for performing the prediction. Time series analysis is done by the dynamic detection of periodicity. The dynamic detector part of this tool analyses the historical information of the grid resources. This pattern detector requires less system resources. Prediction system receives the resource data sent by other grid nodes. This data is processed by the model and slight variations that influence the pattern detection process are ignored. In the absence of patterns, the tool will switch to another scheme. This scheme generates prediction information such as CPU usage, memory utilization, and network saturation [3].

This prediction model can process the statistics of different resource identically. On the occurrence of large vibrations in the time series, the Time Series Predictor and the Smoothed Predictor may not work properly. Prediction process is done based on real time series data. Computing nodes will send resource information periodically and the prediction results are sent to the scheduler or the resource administrator. This data transmission consumes considerable bandwidth. Also the accuracy of prediction depends on the collected information.

4. Monitoring Systems

In a grid system, the execution of long running- applications needs to be managed and monitored by a grid resource management system. There are various monitoring tools that have been developed for resource monitoring. These tools sometimes may contain components for performing the prediction also. Some of them are visPerf, Ganglia, GridICE, and GHS. In this section, the characteristics, benefits and drawbacks of these systems are discussed.

4.1. visPerf

VisPerf is a distributed grid monitoring tool. Grid middleware maintains status information about the system. But this information may not be sufficient. visPerf consists of a distributed monitoring sensor (visSensor) which contains various sensors to monitor the grid middlewares visSensor uses indirect and direct interfaces to retrieve the status. The indirect interface makes use of the run-time log information of the middleware. The direct one uses an interface of the grid middleware to collect the internal information of the grid system. visPerf monitoring system allows users to refine the log which is later used by visPerf. visPerf uses the interface provided by a grid middleware to display the internal information provided by the system. It has a monitor proxy which does the remote sensing of networks. This helps visPerf to adapt to various network configurations including a firewalled system which accepts only legal network ports. Sensors of visPerf share the information through a peer-to-peer network[4].

visPerf does local as well as remote monitoring. Monitored data can be used for scheduling, prediction or tuning of the system. visSensor keeps track of the performance trends of the machine. visPerf can operate only over homogeneous infrastructures and cannot work on systems which share resources with distinct grid computing middlewares. visPerf is not equipped with a standardised information model to represent the statistics provided by different middlewares which are given in their proprietary formats. Because of this, it cannot provide a global view of the system to the grid administrators. Redundancy of monitored data which is required in case of failures is also not provided.

4.2. Ganglia

Ganglia is a monitoring system designed for high performance distributed systems like clusters and grids. It is a scalable monitoring system. It has a hierarchical design that is based on aggregation of clusters. It makes use of a multicast-based listen/announce protocol for monitoring the states of the resources within the clusters. A tree of point-to-point connections amongst the cluster nodes is used to combine them and to aggregate their states. For data representation, Ganglia uses XML (Extensible Mark-up Language), for portable data transport, Ganglia makes use of XDR (External Data Representation) and for data storage and visualization, it uses RRD (Round Robin Database) tool. It achieves very low per-node overheads and high concurrency through the use of efficient algorithms and appropriate data structures. There are currently three classes of distributed systems where Ganglia is being used: clusters, grids, and planetary-scale systems[6]. This tool does not fulfill all the requirements because the choice of a round-robin database for persistent storage is not suitable for retrospective analysis, correlation, or aggregation of monitored data. Ganglia's monitoring daemon is based on a distributed redundant database, which can imply more overhead than necessary [5].

4.3. GridICE

GridICE is a monitoring service architecture satisfying the requirements of real life scenarios giving particular attention to the different categories of consumers of monitoring information. It is based on three use cases which refer to three levels of abstraction in a grid system: the VO (Virtual Organization) level, the site level, and the operations domain level. On the basis of these use cases, a set of requirements is inferred which form the basis for the design of GridICE. GidICE has a layered architecture. Two important aspects of this architecture are the modularity of its components and the easy integration that it allows with current production grid middleware. The ability to deal with the multi-dimensional concepts of a Grid system, by representing data in different dimensions is an important advantage of GridICE. Easy integration with current middleware and a modular architecture are also there which make GridICE a good candidate for monitoring different types of grid systems. It gives both a detailed and a general overview of the grid. It cannot be used for custom grid systems[7].

4.4. Grid Harvest Service (GHS)

GHS is a tool which does both monitoring and prediction based on a novel performance prediction model and a set of task scheduling algorithms. Passive and active monitoring techniques are used to evaluate the end-to-end bandwidth availability and neural networks are used to predict the available bandwidth and latency. GHS consists of the components for performance monitoring, performance prediction, task partition and scheduling, and user interface. The prediction component is based on a new performance model, which is derived from a combination of stochastic analysis and direct-numerical simulation. GHS uses an adaptive measurement methodology to monitor resource usage pattern, where the measurement frequency is dynamically updated according to the previous measurement history. Since the design is based on several components, GHS can be easily integrated into other scheduling systems and vice versa. GHS only considers the workload in distributed systems but not the communication and synchronization costs. It does not provide long-term network bandwidth modelling and prediction. More works need to be done to seamlessly integrate it into the grid environment and with other existing grid tools[8].

4.5. Grid Monitoring Framework Based on Mobile Agents (GMF-BMA)

This is implemented using a mobile agent platform. It has a layered architecture and consists of four layers mainly. The bottom-most layer consists of sensors. The infrastructure layer consists of Resource Monitoring Region. The middle layer consists of monitoring center and servers for providing information. The top layer is a client interface based on web service. Sensors capture performance data from resources. These data are collected by the mobile agents. Additional overhead produced by grid resource monitoring is reduced since a mobile agent platform is used. Grid resource monitoring based on mobile agents can save more bandwidth and can decrease the extra cost caused by grid resource management. Interface between different layers causes overhead. Creating and dispatching mobile agents also result in overhead [9].

4.6. Grid Resource Information Monitoring (GRIM)

This focuses on resource monitoring trade-off and provides a new information retrieving protocol, called the Grid Resource Information Retrieving (GRIR) protocol, to obtain the precise resource status. To get the latest information without unduly increasing the maintenance load in grids, an efficient Grid Resource Information Monitoring (GRIM) protocol is used. There are three basic layers: the query layer, mediation layer, and information provider layer. In the query layer, there are various users, and these users inquire information of interest about some resources in the grids. The mediation layer consists of Mediators to mediate the queries from users and the information from hosts. Each Mediator records the registration of resource information and caches the distinct monitoring conditions of every monitored host. The information provider layer consists of monitored hosts. The resources providing grid services are regarded as hosts in the grid system. These hosts have sensors to detect the state of a resource continuously, and advertise the changed status to the designated Mediator in the mediation layer. GRIR protocol is based on the push data delivery model[10].

4.7. Performance-Forecast and Resource-Autonomy Grid Monitoring Architecture (PFRA-GMA)

PFRA-GMA is a tool which provides both prediction and monitoring services and it works based on GMA (Grid Monitoring Architecture) and SOA (Service Oriented Architecture). It has the capability to do dynamic performance forecast and closed-loop resource autonomy. Prediction service of PFRA-GMA gets the performance information of distributed and networked resources periodically. It makes use of various numerical models to generate predictions about future performance dynamically. The PFRA-GMA has three-layer architecture. Bottom most layer consists of local resources. The layer on top of that will handle the monitoring of resources and the top-most layer interacts with external domains. This three-layer architecture helps PFRA-GMA to provide very good scalability and flexibility. In addition to that, this architecture separates local resource management from global system management. Dynamic performance adaptation and distributed optimization in the grid environment are not achieved by PFRA-GMA which is a drawback of this scheme. Prediction is based on management rules in the Rule Base. Instead of that, adaptive scheduling services based on forecast service are to be implemented [11].

4.8. Online System for Grid Resource Monitoring and Machine Learning-Based Prediction

This is a distributed system for grid resource monitoring and prediction. This system includes two parts: a resource monitoring part for monitoring the resources and a resource state prediction part which makes predictions based on the online data. The resource monitoring portion of the system is situated inside the computing environment and the resource prediction portion is situated outside the computing environment. The system is supported by a collection of services like monitoring service, prediction service, evaluation service and information service. Several service containers are placed in the nodes and above said services are deployed inside these service containers. The service containers and the services are collaborated dynamically to realize the functions of the system. Since the system has a hierarchical structure, management of the resource information can be easily achieved. Prediction is machine-learning based. An optimization model is also integrated with the system for optimizing the performance of the prediction system. In this, the monitored data is more relevant since it is done online. Prediction accuracy is more since it is done based on the online monitoring data. Presence of optimization model also increases the efficiency of the overall system[12].

5. Comparison

Various tools are available for performing Resource Prediction and Monitoring for grid systems. Some of them do only prediction, some of them do only monitoring and some do both monitoring and prediction. Each one makes use of different techniques for performing their tasks. Prediction Accuracy is one of the parameters used to evaluate a prediction system. Table 1 shows a comparison of most of the different prediction and/or monitoring systems that are available today.

Table 1. Comparison of different Grid Resource Monitoring and Prediction Systems

Name	Prediction/ Monitoring	Prediction Data	Features	Techniques used	Design	Evaluation (Prediction Accuracy)
Prophesy (2004)	Prediction	Execution Time	Part of RPS, Uses performance data, Based on execution time, less overhead, less Accurate	Predicted execution time, curve-fitting	Integral part of RPS	----
Adaptive Scheme (2006)	Prediction	Various Resource Information	Time series data, smoothed algorithm, time series analysis, less computational power and storage, lot of bandwidth	Time-series analysis, smoothed algorithm	Sub-modules	Near to 100% if small amount of vibration
visPerf (2002)	Monitoring	Various Resource Information	visSensor, direct & indirect data collection, keeps track of long-term performance trends, no redundancy, middleware-Dependent	visSensor, direct &indirect interfaces	Components	NA
Ganglia (2004)	Monitoring	Various Resource Information	Hierarchical design, cluster based, listen/announce protocol, XML, XDR and RRD tools	multicast-based listen/announce protocol, ,XML, XDR & RRD	Hierarchical	NA

GridICE (2005)	Monitoring	Various Resource Information	Three use cases, recognizes faults, CIM (Common Information Model)	Sensors, CIM	Layered	NA
GHS (2006)	Both	Various Resource Information	Monitoring, Prediction ,scheduling, neural network for prediction, component-oriented Design	Sensors, Passive & active monitoring, neural Networks	Component-oriented	-----
GMF-BMA (2008)	Monitoring	Various Resource Information	Based on mobile agents, Layered, monitoring is done by mobile agents, saves cost and bandwidth, overhead	Mobile agents, sensors, RRMB (Region Resource Monitor Broker)	Layered	NA
GRIM (2008)	Monitoring	Various Resource Information	GRIR protocol, GRIM protocol, layered, Constant data updation	GRIR,GRIM	Layered	NA
PFRA- GMA (2010)	Both	Various Resource Information	Performance forecast, resource autonomy, MA & SOA, numerical models, closed-loop control, Layered	GMA,SOA, numerical models	Layered	----
Online System (2012)	Both	Various Resource Information	Two subsystems, resource info is hierarchical ,supporting services, accurate, efficient	Resource sensor, SVR (Support Vector Machine),PH-PSO (Parallel Hybrid Particle Swarm Optimization)	Supporting Services	-----

6. Summary

Grid is a distributed system in which resources join and leave dynamically. This makes managing them a difficult task. So it is necessary to monitor the resources properly. Prediction is based on monitored data. Both Prediction and Monitoring are required in a grid system. Among all the monitoring systems compared above, Online System does both monitoring and prediction. Since the system collects data online, the data is more relevant and since prediction is done based on this data, it gives more accuracy. Optimization model helps to improve the efficiency of the prediction model which in turn improves the efficiency of the overall system.

References

1. Li, Maozhen, and Mark Baker. The grid: core technologies. John Wiley & Sons, 2005.
2. Jang, S., et al. "Using performance prediction to allocate grid resources." Texas A&M University, College Station, TX, GriPhyN Technical Report 25 (2004).
3. Lee, Liang-Teh, Der-Fu Tao, and Chieh Tsao. "An adaptive scheme for predicting the usage of grid resources." Computers & Electrical Engineering33.1 (2007): 1-11.
4. Lee, DongWoo, Jack J. Dongarra, and Rudrapatna S. Ramakrishna. "Visperf: Monitoring tool for grid computing." Computational Science—ICCS 2003. Springer Berlin Heidelberg, 2003. 233-243.
5. Massie, Matthew L., Brent N. Chun, and David E. Culler. "The ganglia distributed monitoring system: design, implementation, and experience."Parallel Computing 30.7 (2004): 817-840.
6. "Rocks Monitoring Grid Report for Thu, 11Nov 2013 02:27:49 - 0800", Internet: http://meta.rocksclusters.org/ganglia/, Nov. 11,2013
7. Andreozzi, Sergio, et al. "GridICE: a monitoring service for Grid systems."Future Generation Computer Systems 21.4 (2005): 559-571
8. Wu, Ming, and Xian-He Sun. "Grid harvest service: a performance system of grid computing." Journal of Parallel and Distributed Computing 66.10 (2006): 1322-1337.
9. Wu, Libing, et al. "GMF-BMA: A New Grid Resource Monitoring Model."Computational Intelligence and Industrial Application, 2008. PACIIA'08. Pacific-Asia Workshop on. Vol. 2. IEEE, 2008.
10. Chung, Wu-Chun, and Ruay-Shiung Chang. "A new mechanism for resource monitoring in Grid computing." Future Generation Computer Systems 25.1 (2009): 1-7.
11. Yang, Weiqing, Xuebin Chi, and Honghai Zhang. "Performance-Forecast and Resource-Autonomy Grid Monitoring Architecture

(PFRA-GMA)." Distributed Computing and Applications to Business Engineering and Science (DCABES), 2010 Ninth International Symposium on. IEEE, 2010.

12. Hu, Liang, Xi-Long Che, and Si-Qing Zheng. "Online System for Grid Resource Monitoring and Machine Learning-Based Prediction." *Parallel and Distributed Systems, IEEE Transactions on* 23.1 (2012): 134-145.