The Palantir Grid Meta-Information System

F. Guim, I. Rodero, M. Tomas, J. Corbalan, J. Labarta

Barcelona Supercomputing Center, Universitat Politècnica de Catalunya, Computer Architecture Department

(francesc.guim, irodero, julita.corbalan, jesus.labarta) @ bsc.es

miquelt@ac.upc.edu

Abstract— Grids allow large scale resource-sharing across different administrative domains. Those diverse resources are likely to join or quit the Grid at any moment or possibly to break down. Grid monitoring tools have to adapt supporting access information to these heterogeneous and not reliable environments. There is a wide range of types of resources to be monitored, with different nature, characteristics and so on. For instance Grid users want to monitor from which is the state of the jobs that submit till how many processors has a given resources of a given centre. These issues make the task of monitoring complex to treat, and it is difficult to provide a general way for accessing to all this information.

In this work we propose a set of functionalities that a Grid Information System should provide. We describe the Palantir meta-information system that has been designed for uniform the access to different monitoring and information systems and that implements all the discussed functionalities. This meta-information system tries to abstract the resource information that underlies on the local systems, and homogenize the access to all this. We present the architecture of Palantir and its main characteristics.

I. INTRODUCTION

As other Grid middleware applications, information services have to adapt to the characteristics of Grid architecture. They should provide uniform access to information and resources, ways to discover which capabilities it has and how to access it. However it is not easy, because Grid environments are volatile, heterogeneous and not reliable. We define a resource as any Grid entity, software or hardware that is able to provide information, for instance a resource can be from a host till a user job.

In the eNANOS[1] architecture we needed a new component that would allow accessing to all the information related to the entities that are involved in our system, i.e: Grid jobs, local processes, resources and so on. This component would integrate and merge information from the bottom part of the Grid, coming from the local components of the centres, such as the NANOS scheduler, with information coming from the top components of our architecture, in our case from the eNANOS Broker. The information nature that it would provide would be much diversified, not only providing monitoring information from the monitoring system, or job monitoring. On the other hand, this new component had to be extensible, in sense it should be easy to extend it with new information features.

For achieve this goal, our first approach consisted on having a deeper study of the available monitoring tools that have been deployed till the moment and try to adapt the more appropriate one. However, we realized that any of them matched all the requirements that we had. Using our recent experience in grid monitoring and information systems obtained in [2][3], we designed a kind of meta-information system that implements all the needed functionalities.

This information system provides a generic access to different information producers in a uniform way. Similar as the Globus does with the GRAM manager uniforming the job submission, it abstracts the data models and the different access mechanisms (API, RPC, command line etc) that each of this producers have. Something important to remark is that the presented system is not intended to substitute of other existing monitoring or information tools. It provides easy and powerful mechanisms for gathering data from different sources, and providing mechanism for implement new information modules in case that it is needed.

The goal of this poster is to provide a description and characteristics that our meta-information system has in terms of functionalities and architecture.

II. PALANTIR DATA MODEL

The cores of its data model are the entities. They represent the conceptual parts of the systems that can contain information suitable to be requested. They are not required to be physical resources. For example, hosts, jobs or applications are considered to be entities. Each entity has associated a set of Metrics. They contain specific information about the entity to who are linked, for instance the metric \textit{elapsed time} is a metric that can be associated to the entity job.

Each entity type has a set of instantiations: for example the entity host may the instantiations “\textit{host1.bsc.es}”, “\textit{host2.bsc.es}” etc.

III. PALANTIR SYSTEM ARCHITECTURE

This section provides a general description of what is the architecture of the Palantir meta-information system that we designed. In [4], the Global Grid Forum Performance Working Group developed a model for Grid monitoring tools, the Grid Monitoring Architecture (GMA), we have based the design of our system on such proposal. Figure 1 provides a general view of the system architecture. As can be observed there are three top architectural components: Palantir Access Point, the Palantir Gateways and finally the information modules. In the following subsections their main characteristics are presented.
A. The Palantir Access Point

The first layer is the Palantir Meta-Information System access point. It manages the queries that are done by the users or applications redirecting them to the appropriate Palantir Gateway. This redirection is based on two different facts:

- If monitoring information is queried, then it is based on where the queried entity remains. The unique requirement is that the client must provide which information has to be gathered, for example: "uptime" for the host host1.bsc.es.
- If discovery information is required, for example, the entities of type host available on the whole system. The query may be split to many queries to different gateways.

The client has not to be aware if the final queries are done to the MDS or to the NWS system, the API provides a generic authentication and information access methods.

B. The Palantir Gateways

The second layer of components that are located on the local centers, are the Palantir Gateways. They are responsible to carry out the queries to the information systems and monitoring system modules.

The main task of the gateway is choosing the appropriate module that will process the query. It stores a data base with all the entities that are available on the systems, and when entity information is required it searches in such database where it can be retrieved.

The gateway distinguishes two kinds of entities:

- Persistent entities. Entities that are non-volatile or that in case that there are no problems (network, a host is down etc.) will remain available for a long time. Examples of this kind of entities are: host, network, cluster etc. For these entities the database stores among other data in which information system are located, for example in Figure A the Palantir Gateway of the Center A will know that the entity "Host A" can be queried in the module with id "MDS1". Regularly the data base is updated with the creation or destruction of entities in the information systems.
- Non-persistent entities. Entities that are volatile or with a limited amount life time. For example: jobs. These kind of entities are identified with a composed key that contains information about where they can be found. Basically, a subset of its key must identify a persistent entity that will be used to find out in which module the query has to be done. For example entity "job1@hostD" has the key job1+hostD, then the gateway knows that the persistent subkey is hostD and can found in which module the information about the entity can be found. For this kind of entities no information is based on the data base.

The gateways have a cache mechanism to avoid unnecessary communications with the local modules. The TTL of such data can be tuned by the system administrators, big TTL can be useful for entities whose data is not likely to change for a while, for instance static information for the entity broker, such as policy, ip, name etc.

C. Monitoring and Information Systems modules

They are the components present in the bottom layer of the architecture. Modules are responsible of carry out the final queries to the information or monitoring systems that they represent. The modules receive queries based on the generic format presented in the following section and they take the appropriate actions for answer them.

Obviously, at this level modules must know how the queries have to be done to the systems. For each monitoring or information system that can be queried through the Palantir, a module must be implemented.

CONCLUSIONS

In this poster we have introduced the main characteristics and design goals of the Grid information system Palantir. It has been presented it architecture, and in a very high level, how it provides functionalities for accessing to a very wide range of Grid information. It has also been shown how this information system abstracts the user to the complexity of the underlying information and monitoring systems.

ACKNOWLEDGMENT

This work has been supported by the Spanish Ministry of Science and Education under contract TIN2004-07739-C02-01.

REFERENCES

[1] Ivan Rodero, Francesc Guim, Julia Corbalán and Jesus Labarta. eNANOS: Coordinated Scheduling in Grid Environments. Parco - Parallel Computing 2005

