The Distributed Monitoring Framework

Brian Tierney, LBNL

Summary

The goal of the Distributed Monitoring Framework is to provide the ability to do performance analysis and fault detection in a Grid computing environment. This monitoring framework will provide accurate, detailed, and adaptive monitoring of all distributed computing components, including the network. Analysis tools will be able to use this monitoring data for real-time analysis, anomaly identification, and fault recovery. Services for network-aware applications will use this monitoring infrastructure in order to provide past, present, and future predictions of the network conditions, which the applications can then use to adapt their behavior. Applications such as widely distributed workflow management systems can then use the DMF as a generalized real-time event service for tracking workflow progress.

The next generation of high-speed networks will allow DOE scientists unprecedented levels of collaboration. The focus of the DMF is to unify existing monitoring tools, provide seamless integration with a Grid environment, and to use these tools with scientific Grid applications. Experience gained will help the longer-term goal of automatically detecting, and correcting for, anomalies in scientific application workflows.

The DMF consists four main components: instrumentation, event publication, sensors, and event archiving. The term “event” is used here to mean a single time-stamped unit of monitoring data. The current state of each component is described briefly below.

Instrumentation

Precision, real-time instrumentation of Grid applications and middleware is essential to understanding high performance data intensive applications. We have extended our previous work on the NetLogger Toolkit to provide the basis for this component. The NetLogger Toolkit included language APIs, data collection tools, and a visualization tool called NLV. To this we added a new, very efficient, binary logging format. We also added a fault tolerance mechanism, so that NetLogger is robust in the face of network instability. Finally, we added a "trigger interface" to NetLogger, which allows NetLogger to be turned on, off, or the logging level changed, in a running application.

We have added a flexible URL-based means of specifying where and what NetLogger should be logging.

Event Publication

The DMF bases its event publication on the architecture known as the Grid Monitoring Architecture (GMA), which was developed in coordination with other researchers in the Global Grid Forum (GGF). The GMA defines event “Producers”, event “Consumers”, and a distributed metadata “Registry” that allows Producers and Consumers to find each other. The event data goes directly from Producer to Consumer instead of through a central bottleneck.

We have written a prototype implementation of the GMA in the high-level Python language, called pyGMA, that uses standard Web Services protocols (the same protocols used in the newest release of the Globus Toolkit) for communicating control information between components, but which can use NetLogger as an efficient "data channel" for the monitoring event data.

We have combined the pyGMA and NetLogger "trigger" mechanism to allow remote activation of NetLogger instrumentation in running applications. We call this the NetLogger Activation Service, and have demonstrated it with an instrumented version of the PyGlobus toolkit and AMBER biochemistry application at the IEEE Supercomputing conference in November 2003. This demonstration produced truly end-to-end monitoring of the entire "Grid workflow" including the file staging, authentication, job submission, application run, and transfer of results. We used a Grid workflow identifier (GID) inserted into all the running application logs to differentiate between concurrent workflows.

Sensors

For sensors, the focus has been on integrating existing monitoring and sensors into the framework. Most of this work so far has been in standardization efforts.

We co-chaired the Network Monitoring Working Group, which is applying some of the principles of the DAMED work to the difficult problem of a precise vocabulary for network measurements. We participated in the first revision of an XML schema specifying a flexible, standard format for reporting network measurements with the NM-WG classification scheme. We also co-chaired the GGF Discovery and Monitoring Event Descriptions (DAMED) working group, which defined a common naming convention.

Event Archiving

Historical analysis tools are necessary to understand why, for example, application performance "this week" is so much worse than it was "last week". Condensing the data may mask important details, so this task requires an archive that can handle streams of detailed monitoring events. We designed and implemented a prototype of such an event archive using an open source relational database back-end (mySQL), and extensive buffering on disk to ensure that the database is never overloaded by a large chunk of incoming events. The archive supports the GMA Consumer and Producer interfaces and has a web-based interface.

For real-time queries of large datasets, we have implemented a prototype that integrates NetLogger data streams with of a back-end database technology called "continuous queries" (CQ) to provide real-time relational query capabilities. The back-end we are using for this is Telegraph CQ from the University of California at Berkeley.

Next Steps

We will implement a Web Services front-end to our archive that uses the NM-WG XML work to provide interoperable queries. We will explore ways to integrate more efficient NetLogger transports for out-of-band transfer of large sets of NM-WG measurements. We will also develop a prototype system that uses the same front-end for our streaming query engine.

We plan to expand support for Grid workflow identifiers (GIDs) in all NetLogger components.

For further information on this subject contact:

Brian Tierney, Staff Scientist

Data Intensive Distributed Computing, DSD

Phone: 510-486-7381

blltierney@lbl.gov