Towards a Unified Monitoring and Performance Analysis System for the Grid

Hong-Linh Truong, Thomas Fahringer
Institute for Software Science,
University of Vienna, Austria
{truong,tf}@par.univie.ac.at

http://www.par.univie.ac.at/project/calea

APART-2 Workshop on Grid Monitoring, Klagenfurt, August 25th, 2003
Outline

- Grid in our view
- SCALEA-G Architecture
- Sensor and Sensor Manager Service
- Instrumentation
- Data Subscription and Query
- Prototype
- Summary
**Grid Services**

- **Grid systems**
  - Collection of grid services

- **Grid services**
  - Web service that provides a set of well-defined interfaces (e.g. addressed discovery, dynamic service creation, lifetime management, notification, manageability) and that follows specific conventions (e.g. addressed naming, upgrading) in the Grid.

- **Types of Grid services**
  - Computational services (CS). E.g. computational hosts
  - Network services (NS). E.g. network connections
  - Software services (SS)
SCALEA & SCALEA-G

- **SCALEA**
  - Performance Instrumentation, Measurement, Analysis and Visualization for Parallel Applications
  - Main focus: Fortran OpenMP/MPI on Clusters

- **SCALEA-G (SCALEA Grid-enabled)**
  - Unified system of monitoring and performance analysis for Grid Services
    - Computational services, network services and software services
    - Based on GMA (Grid Monitoring Architecture) and OGSA (Open Grid Service Architecture)
  - Providing meaningful performance data to external tools/software
Combining GMA and OGSA

- Support both push (via subscribe) and pull (via query) model.
- Control operations: to control activities, to register information, to subscribe and query data.
  - Based on Grid services operations
  - Data Channel: to deliver real subscribed data, results of requests
  - Use a separate data stream connection.
- All are implemented as OGSA-Enabled Grid services
  - Deployed on different sites & shared by multiple users
  - Used by different external tools
Directory Service and Archival Service

- **SCALEA-G Directory Service**
  - Store information about Sensor Managers, sensors, properties of data provided by sensor instances, consumers
  - Employ a relational database (PostgreSQL)

- **Archival Service**
  - Extension of SCALEA Experiment Repository
    - Raw data provided by sensor instances
    - Analyzed data provided by analysis services

- **Open problem:**
  - Data is organized in distributed manner
  - Data has to be represented in a semantic way so that external tools/software can easily and automatically use the data: ontology?
SCALEA-G Sensor Manager Service

- **Components**
  - Service Administration
  - Data Subscription (push model)
  - Data Query (pull model)
  - Data Publication (publish data)
  - Instrumentation Request Mediator
  - Data Service
Sensor Manager Service: Data Service

- Data delivery is carried out via Data Service
- Data is cached and filtered at Sensor Manager Service (SMS)
- There is only one connection from SMS to consumer
Sensors

A sensor is a component that performs measurements

- **Classification**
  - *System sensors* are used to monitor Grid computational services and Network services
  - *Application sensors* are specific codes embedded in Grid software services to measure execution behaviors of code regions, to monitor events of these services, etc.

- **Static and dynamic properties**
  - Unique sensor identifier
  - Public XML Schema for measurements
  - Lifetime (start, end)
  - ...

...
System Sensors & Sensor Repository

- **System sensors**
  - Monitor computational services and network services
    - Networks link, hard disks, memory usage, CPU availability
  - Exploit existing tools: extracts information from existing providers, e.g. MDS, NWS
  - Network metrics
    - Based on work of Grid Network Measurements Working Group (http://www-didc.lbl.gov/NMWG/)
    - Close to applications, e.g. path metrics at transport layer (TCP, TSL), application protocol (HTTP, SOAP)

- **Sensor repository**
  - Collection of system sensors, add-on ability
  - Represented in XML
  - System sensors can be invoked by Sensor Manager Services
<table>
<thead>
<tr>
<th>Sensor Name</th>
<th>Monitored Object</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host.cpu.used</td>
<td>host</td>
<td>The average, over the last minute, of the amount of time that processors of this system were not idle</td>
</tr>
<tr>
<td>host.predict.cpu.used</td>
<td>Host</td>
<td>The prediction of CPU usage</td>
</tr>
<tr>
<td>host.mem.used</td>
<td>Host</td>
<td>Ratio of used memory</td>
</tr>
<tr>
<td>host.predict.mem.used</td>
<td>Host</td>
<td>The prediction of memory used ratio</td>
</tr>
<tr>
<td>host.system.loadavg</td>
<td>Host</td>
<td>System Load average</td>
</tr>
<tr>
<td>path.bandwidth.capacity.TCP</td>
<td>Network</td>
<td>TCP bandwidth capacity</td>
</tr>
<tr>
<td>path.predict.bandwidth.capacity.TCP</td>
<td>Network</td>
<td>Prediction of TCP bandwidth</td>
</tr>
<tr>
<td>path.delay.oneWay</td>
<td>Network</td>
<td>One way delay for IP packet</td>
</tr>
<tr>
<td>path.delay.roundTrip</td>
<td>Network</td>
<td>Round trip delay for IP packet</td>
</tr>
<tr>
<td>path.delay.roundtrip.TCP</td>
<td>Network</td>
<td>Round trip delay for TCP packet</td>
</tr>
</tbody>
</table>

The same work should be done for high-level network metrics e.g. (SOAP, HTTP)
<sensor name="host.mem.used">
    <impl>scaleag.sm.sensor.Mem</impl>
    <desc>Measure ratio used memory of a host</desc>

    <properties>
        <![CDATA[
            <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
                ...
                <xsd:element name="sensordata" type="SensorData"/>
                <xsd:complexType name="SensorData">
                    <xsd:sequence>
                        <xsd:element name="hostname" type="xsd:string"/>
                        <xsd:element name="eventtime" type="xsd:dateTime"/>
                        <xsd:element name="availmem" type="xsd:double"/>
                        <xsd:element name="usedmem" type="xsd:double"/>
                    </xsd:sequence>
                    <xsd:attribute name="name" type="xsd:string"/>
                </xsd:complexType>
            </xsd:schema>
        ]]>]
    </properties>

    <params>
        <param name="Interval" desc="second" dataType="int"/>
    </params>
</sensor>
Application Sensors

- How sensors are embedded into software services
  - Source code/byte code instrumentation service
    - Fortran (Source code), Java (byte code)
    - Investigate ARM (Application Request Management) standard for Grid service
  - Dynamic instrumentation:
    - Mutator service is created by application process
      - Created by user process
      - Number of mutators is controlled by user (via function calls, environment variables)
    - Mutator service runs as a separate service
      - Used by multiple users
      - One instance per node per user

- Data collected online
  - Profiling & tracing data
  - XML representation
  - Low level and high level metrics
Application Sensor Data

```
<sensordata name="app.trace">
    <coderegion> ... </coderegion>
    <processingunit> ... </processingunit>
    <events>
        <event eventname="FOO_CALL">
            <eventtime>1061567295288</eventtime>
            <eventdata attrname="CALLEE" attrvalue="ServiceB/FOO/>
        </event>
    </events>
</sensordata>

<sensordata name="app.prof">
    <coderegion> ... </coderegion>
    <processingunit> ... </processingunit>
    <metrics>
        <metric name="CTIME" value="8.0962703E7"/>
        <metric name="WTIME" value="2.61909657E8"/>
    </metrics>
</sensordata>
```
**Dynamic Instrumentation Request**

- **Instrumentation Request Language (IRL)**
  - XML based
  - C++/Java based on Xerces XML library

- Any tool that supports IRL can work with mutator service

```xml
<?xml version="1.0" ?>
<irl>
  <request name="instrument">
    <processingunit computationalNode="gescher" ... />
    <task coderegions="MPI_Reduce" metrics="WTIME,L2_TCA" />
  </request>
</irl>
```
SCALEA-G Client Service

**Consumer Service**
- Control activities of sensor manager services and sensors
- Register information to directory service
- Subscribe/unsubscribe and query data

**Instrumentation Mediator**: Act as intermediary agent in communicating between users/tools with
- Source Code Instrumentation Service (based on SCALEA Instrumentation Service)
- Dynamic instrumentation service

**Performance Analyzer**
- Analyze collected data provided by Consumer Service and provide the result to the user
Data Subscription and Query

Message Propagation uses simply tunnel protocol

Pull and Push Request

- **Consumer** has XML Schema specifying data provided by sensors
- **Consumer** builds Pull/Push request in XML based XPath/XQuery
Security Issues

- **Authentication & Authorization**
  - Performed in several actions such as registration, subscription, control of activities
  - Carried out by GSI (Globus) with user’s X.509 certificate

- **Shared SCALEA-G services**
  - The administration can define access control list which maps user information to data types/tasks which the user is allowed to access.

- **Subscription/Query data collected by application sensors**
  - Only the user who invokes the application is allowed
  - Sensor Manager Service records the information about the user who wants to subscribe/query data and the one who invokes applications
Summary

- **Design of SCALEA-G**

- **Current status**
  - Finishing the implementation of basic infrastructure
  - Very premature prototype

- **Future works**
  - Refine and improve design
  - Work on full implementation
  - Study representation of monitoring and performance data in Grids.