MONALISA

MONitoring Agents using a Large Integrated Services Architecture

Iosif Legrand
California Institute of Technology
Hierarchical structure of loosely coupled services which are independent & autonomous entities able to cooperate using a dynamic set of proxies or self describing protocols.

They need a dynamic registration and discovery & subscription mechanism

For an effective use of distributed resources, these services should provide adaptability and self-organization (aggregation and hierarchical orchestration)

Reliable on a large scale network distributed environment
  - Avoid single points of failure
  - Automatic re-activation of components and services

Scalable & Flexible for adding dynamically new services and automatically replicate existing ones to cope with time dependent load
“Traditional” Distributed Object Models (CORBA, DCOM)

The Stub is linked to the Client. The Client must know about the service from the beginning and needs the right stub for it.

The Server and the client code must be created together !!
The client can dynamically generate the data structures and the interfaces for using remote objects based on WSDL

Platform independent
Dynamic Code Loading
Less Protocols !!

Any service can be used dynamically

- Remote Services Proxy == RMI Stub
- Mobile Agents Proxy == Entire Service
- “Smart Proxies” Proxy adjusts to the client
Act as a true dynamic service and provide the necessary functionally to be used by any other services that require such information (Jini, UDDI - WSDL / SOAP)
- mechanism to dynamically discover all the "Farm Units" used by a community
- remote event notification for changes in the any system
- lease mechanism for each registered unit

Allow dynamic configuration and the list of monitor parameters.
Integrate existing monitoring tools (SNMP, LSF, Ganglia, Hawkeye …)

It provides:
- single-farm values and details for each node
- network aspect
- real time information
- historical data and extracted trend information
- listener subscription / notification
- (mobile) agent filters and alarm triggers
  algorithms for prediction and decision-support
A Service Registers with at least one Lookup Service using the same ID.

It provides information about its functionality and the URL addressed from where interested clients may get the dynamic code to use it. The Service must ask each Lookup Service for a lease and periodically renew it.

If a Service fails to renew the lease, it is removed from the Lookup Service Directory. When problems are solved, it can re-register.

The lease mechanism allows the Lookup Service to keep an up to date directory of services and correctly handle network problems.
Monitoring: Data Collection

Dynamic Thread Pool

SNMP get & walk
rsh | ssh remote scripts
End-To-End measurements

Farm Monitor

Other tools
(Ganglia, MRT...)

Configuration Control

PULL

PUSH

snmp trap

WEB Server

Dynamic loading of modules or agents

Trap Agent
(ucd – snmp) perl
The Muti-Threaded Execution Architecture

- Each request is done in an independent thread
- A slow agent / busy node does not perturb the measurements of an entire system
- Ex: Monitor 300 nodes @ 30 seconds interval → 10-15 Threads are running in parallel
Farm Monitor UNIT & Data Handling

Client (other service) Web client

WEB Service
WSDL SOAP

Monitor Data Stores
Config
Status
Data

Data Cache Service
Predicates & Filter Agents

Lookup Service
Registration

Discovery

data

Client (other service)
Java

Other tools

User defined loadable Modules to write /sent data

Configuration Control

MySQL
MDS

InstantDB
MySQL
Postgres…

Java Discovery

Client (other service)

Client (other service)
Data Handling

Data Model

- Configuration: Farm, Function (Cluster), Node, Module
- Monitored Values
  - (Automatic) Mapping of the Data Model in:
    - XML, SQL, SOAP, ...
  - Configuration & Results objects are stored in a DB
    - (dynamically configurable for InstanDB, Postgres, MySQL, Oracle ...)
- Subscription to results objects matching a template / predicate
- Clients can load filter objects into the Data Cache service and generate any derived (or aggregate) data structures and register to receive them.
- Monitored parameters have a life time

June 2003

Iosif Legrand
MonaLisa is a monitoring Framework:

- SNMP (walk and get) for computing nodes, routers and switches
- Scripts, dedicated application (programs) which may be invoked on remote systems
- Interface to Gangia
- Interface to LSF and PBS
- Interface to Hawkeye (Wisconsin)
- Interface to LDAP (MDS) (Florida)
- Interface to IEPM-BW measurements
- Specialized modules for VRVS
RC Monitoring Service

- Component Factory
- GUI marshaling
- Code Transport as “service attribute”
- RMI data access

Registration with several Lookup discovery services

July 2003

Iosif Legrand
Secure – Automatic Update Mechanism for Services and Clients

- All running services are updated using the discovery mechanism.
- At startup, each service checks if an update is done at a set of Web Servers.
- Clients use the Web Start mechanism.
Global Client / Dynamic Discovery

IEPM- BW Measurements @ SLAC

Load on the Farm Nodes @ CALTECH

Traffic from CERN into Geant

Production Traffic CERN-US Real-time

Traffic from CERN into DataTAG
### Global Views: CPU, IO, Disk, Internet Traffic...

<table>
<thead>
<tr>
<th>Regional Center</th>
<th>Free Nodes</th>
<th>RateOUT [KB/s]</th>
<th>RateIN [KB/s]</th>
<th>CPU [usr]</th>
<th>Free Disk [GB]</th>
<th>Load</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>WMAP</em></td>
<td><em>Table</em></td>
<td><em>Load</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Table</em></td>
<td><em>Load</em></td>
<td><em>WMAP</em></td>
<td>RateOUT [KB/s]</td>
<td>RateIN [KB/s]</td>
<td>CPU [usr]</td>
<td>Free Disk [GB]</td>
</tr>
<tr>
<td><em>Table</em></td>
<td><em>Load</em></td>
<td><em>WMAP</em></td>
<td><em>Table</em></td>
<td><em>Load</em></td>
<td>RateOUT [KB/s]</td>
<td>RateIN [KB/s]</td>
</tr>
<tr>
<td><em>Table</em></td>
<td><em>Load</em></td>
<td><em>WMAP</em></td>
<td><em>Table</em></td>
<td><em>Load</em></td>
<td>RateOUT [KB/s]</td>
<td>RateIN [KB/s]</td>
</tr>
<tr>
<td><em>Table</em></td>
<td><em>Load</em></td>
<td><em>WMAP</em></td>
<td><em>Table</em></td>
<td><em>Load</em></td>
<td>RateOUT [KB/s]</td>
<td>RateIN [KB/s]</td>
</tr>
<tr>
<td><em>Table</em></td>
<td><em>Load</em></td>
<td><em>WMAP</em></td>
<td><em>Table</em></td>
<td><em>Load</em></td>
<td>RateOUT [KB/s]</td>
<td>RateIN [KB/s]</td>
</tr>
</tbody>
</table>

**Load Distribution of Computing Nodes**

**WAN Traffic**
Regional Centers Discovery & Data access
Real-time Data for Large Systems
“Ixshare” cluster at cern ~ 600 ndoes

Load5

Load 5

Load 1

Load 0

June 2003

Iosif Legrand
Access to historical and real-time values

Past values are presented and the GUI remains a registered listener and the new values are added.

Real Time Histograms for various parameters
Filter Mobile Agents

Maximum Flow Data Replication
Path Agent Deployed to each RC and evaluates the best path for real-time data replication.
Monitoring VRVS Reflectors

File  Discovery  Groups  Security  Help

Nodes: No. of audio clients  ▼  8  Links: Internet RTT Time Quality  134.3  0.48  Show

Peer Mean quality / 2h  ▼  0  100%  Show

Show MST

Manage sets  Smooth sets  Analyze sets

Node = localhost

Traffic_IN  Traffic_OUT

Update  Clear  Data  Close  Plot

June  2003  Iosif Legrand
PDA 3D MonALISA Client

Developed by ERICSSON Research Lab
Performance Test: SNMP query (~200 metrics values) on a 500 nodes farm every 60 s.

~ 1600 metrics values collected per second.

Dell I8100 ~ 1GHz
Performance Test: High rate SNMP queries (every 15 s) on a large farm (500 nodes)

- Load
- CPU \textit{usr/sys}
- MANY Threads
- LONG RTT

Dell I8100 ~ 1GHz

June 2003

Iosif Legrand
SUMMARY

- MonaLisa is able to dynamically discover all the "Farm Units" used by a community and through the remote event notification mechanism keeps an update state for the entire system.
- Automatic & secure code update (services and clients).
- Access to aggregate farm values and all the details for each node.
- Selected real time / historical data for any subscribed listeners.
- Active filter agents to process the data and provided dedicated / customized information to other services or clients.
- Dynamic proxies and WSDL pages for services.
- Embedded SQL Data Base and can work with any relational DB. Accepts multiple customized Data Writers (e.g. to LDAP) as dynamically loadable modules.
- Embedded SNMP support and interfaces with other tools (LSF, Ganglia, Hawkeye…). Easy to develop user defined modules to collect data.
- Dedicate pseudo-clients for repository or decision making units.
- It proved to be a stable and reliable service.
MonALISA is NOT a monitoring or graphic tool. The aim is to provide a flexible and reliable MONITORING SERVICE for higher level services in distributed systems.

Code mobility paradigm provides the mechanism for a consistent, correct invocation of components in large, distributed systems. Filters and trigger agents can be dynamically deployed to any service unit to provide the required monitoring information to clients or other services.

MonALISA is a prototype for a dynamic distributed services. Suggestions to improve it, to better describe network elements and computing systems are welcome.

http://monalisa.cacr.caltech.edu