MONALISA

MONitoring Agents using a Large Integrated Services Architecture

Iosif Legrand
California Institute of Technology

Distributed Dynamic Services Architecture

- 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)

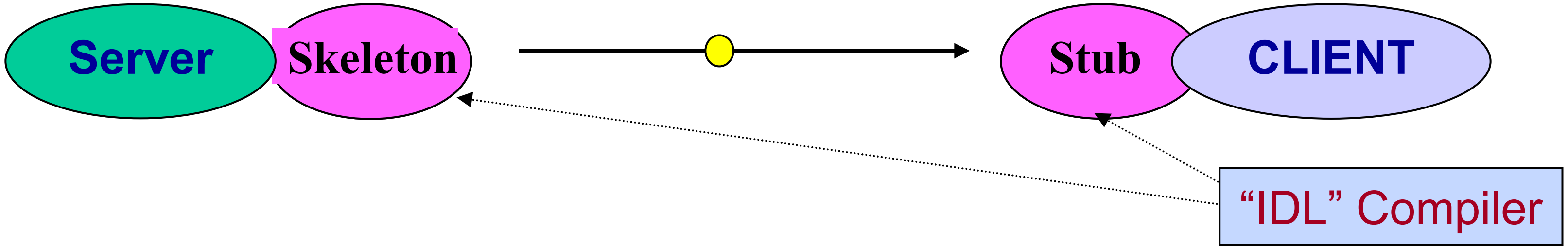
- Reliable on a large scale network distributed environment
  - *Avoid single points of failure*
  - *Automatic re-activation of components and services*

- Scalable & Flexible for adding dynamically new services and automatically replicate existing ones to cope with time dependent load

Distributed Object Systems
CORBA , DCOM

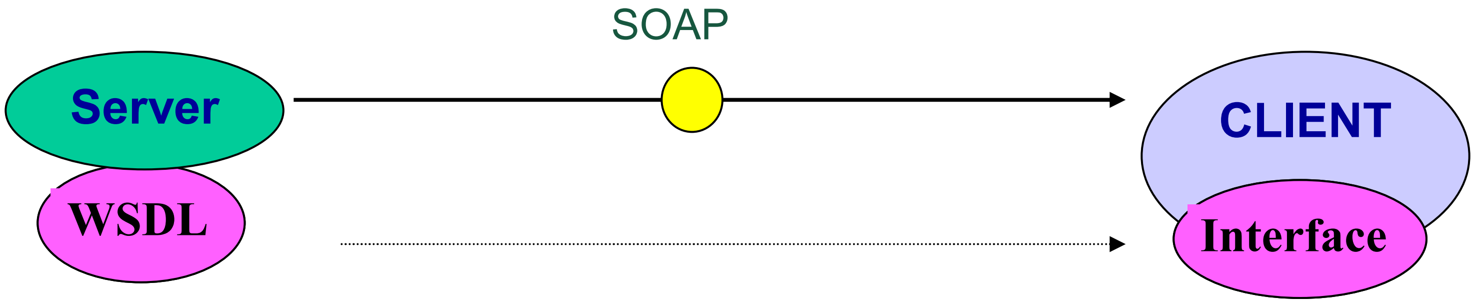
“Traditional” Distributed Object Models
(CORBA, DCOM)

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 !!

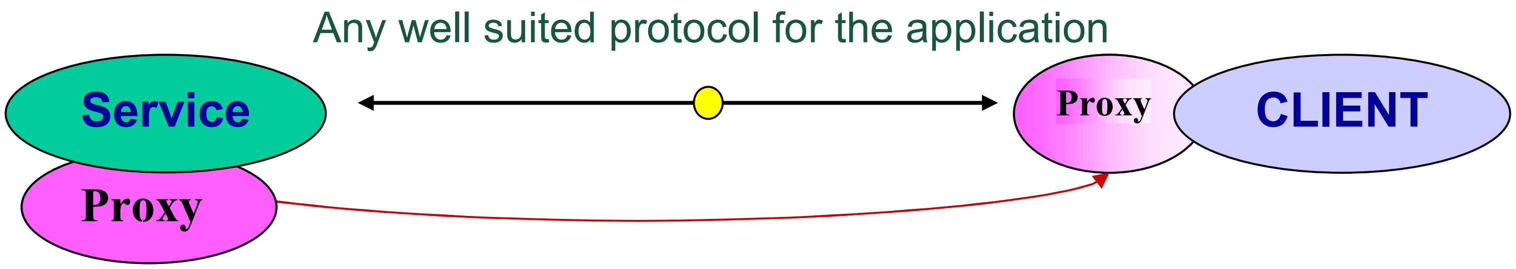
Distributed Object Systems
Web Services WSDL/SOAP



The client can dynamically generate the data structures and the interfaces for using remote objects based on WSDL

Platform independent

Distributed Object Systems
Java / JINI



Dynamic Code Loading
Less Protocols !!

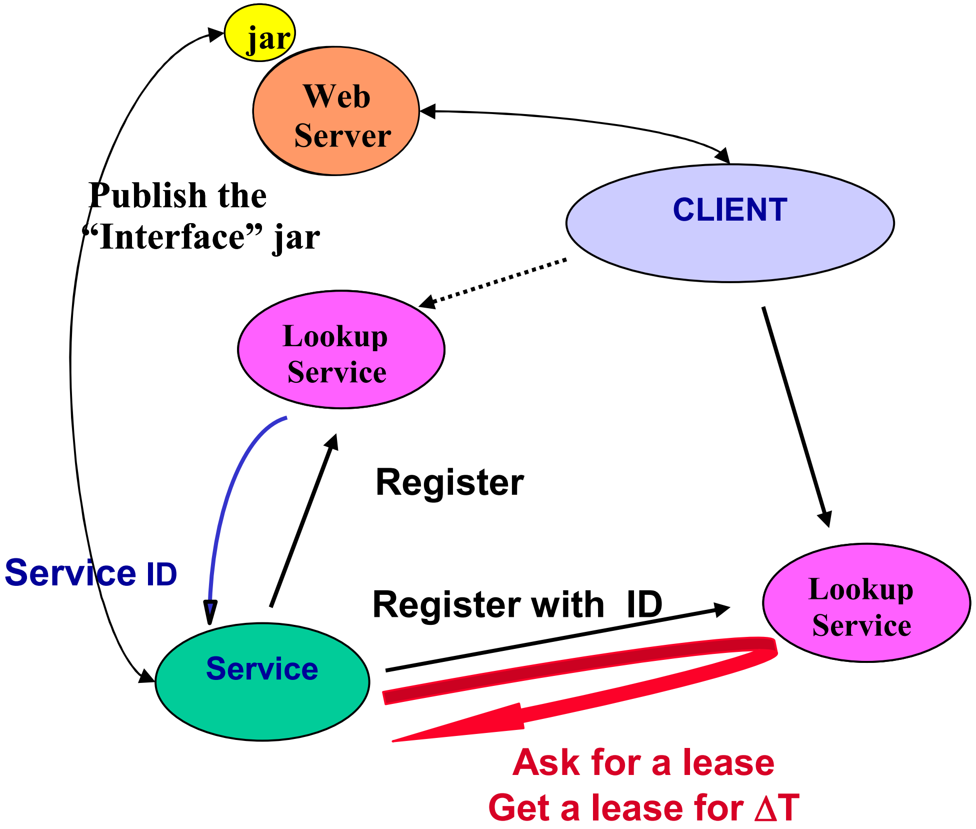
Any service can be used dynamically

•Remote Services	Proxy == RMI Stub
•Mobile Agents	Proxy == Entire Service
•“Smart Proxies”	Proxy adjusts to the client

MonALISA
Design Considerations

- 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

JINI – Network Services



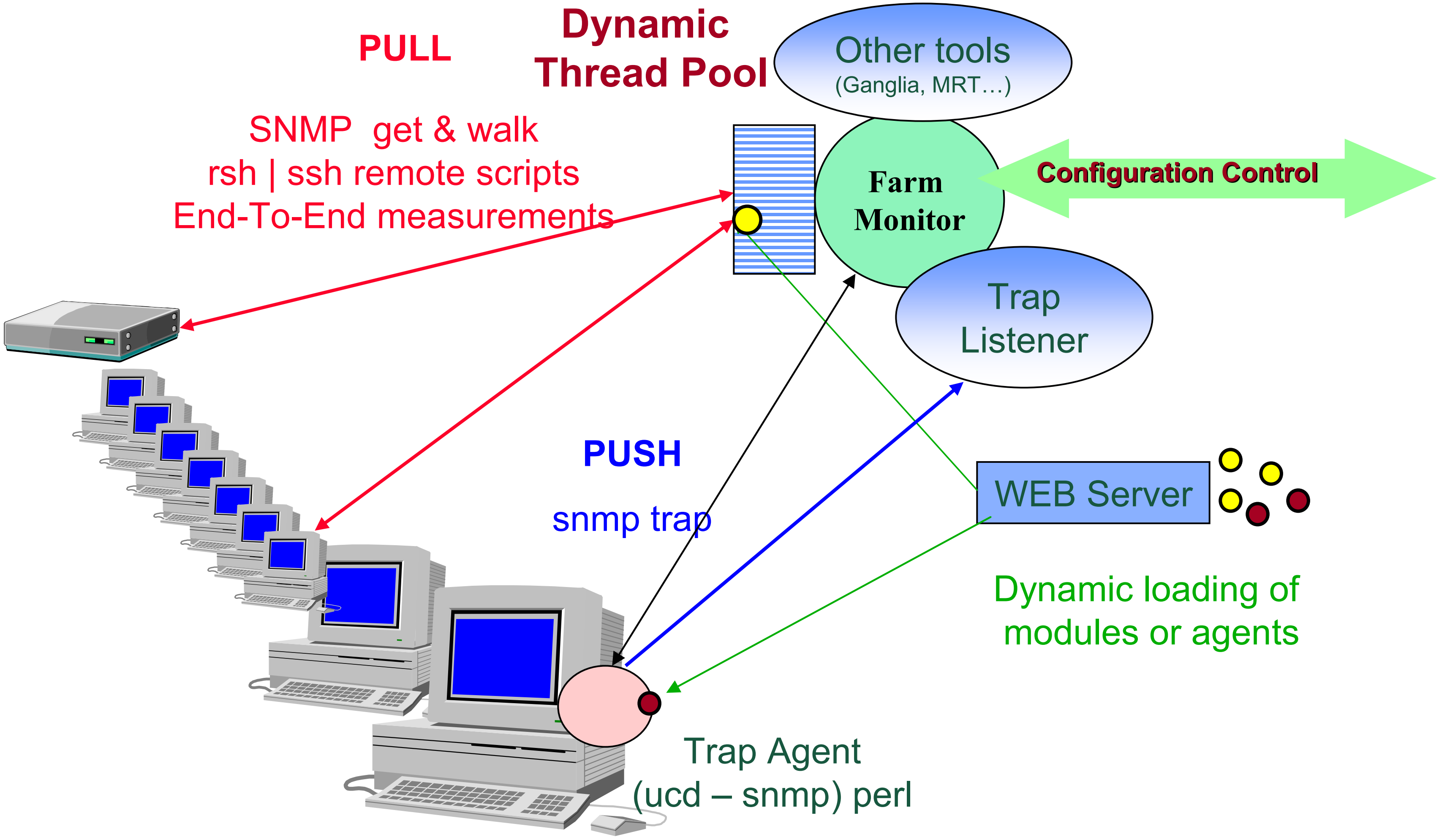
A Service Registers with at least one Lookup Service using the same ID.

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 form 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



PULL

Dynamic Thread Pool

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

Other tools
(Ganglia, MRT…)

Farm Monitor

Configuration Control

Trap Listener

PUSH

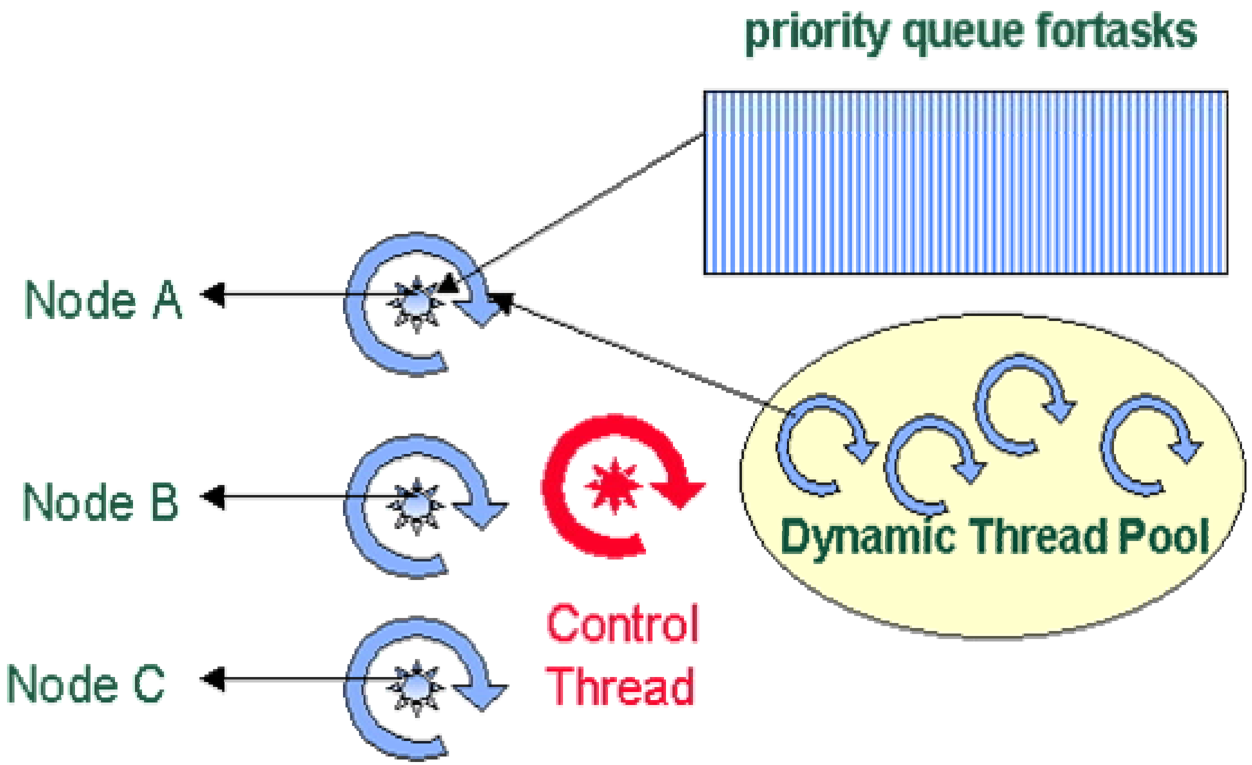
snmp trap

WEB Server

Dynamic loading of modules or agents

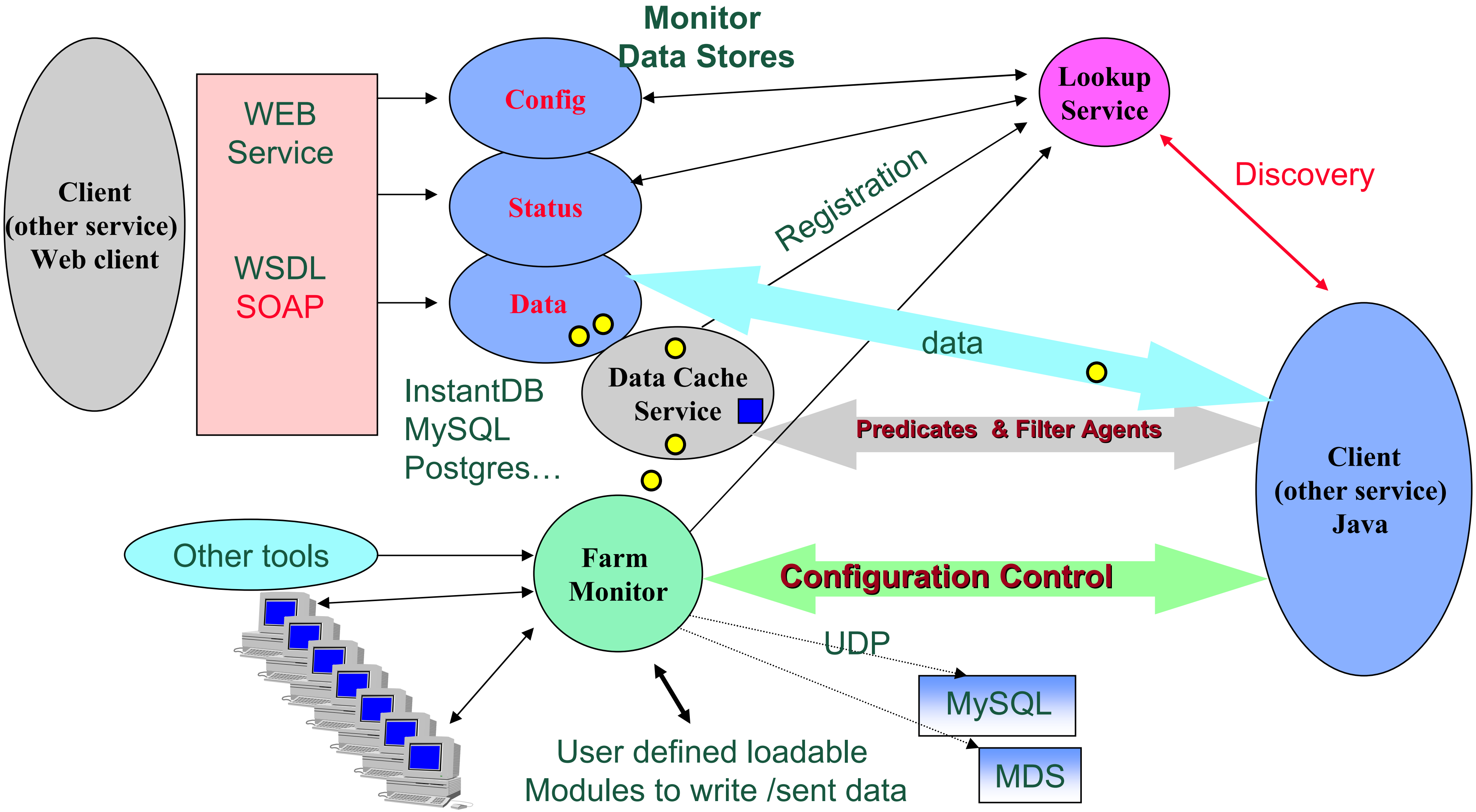
Trap Agent
(ucd – snmp) perl

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

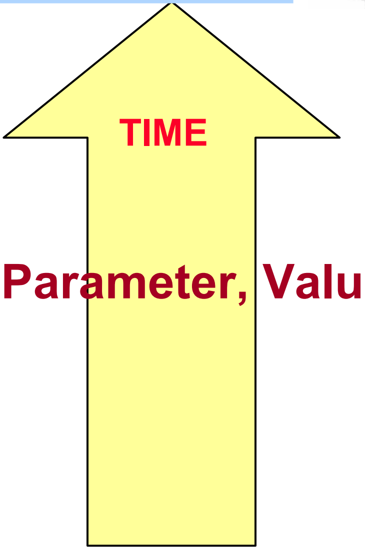
Farm Monitor UNIT & Data Handling



Data Handling

Data Model

- Configuration Farm , Function (Cluster), Node, Module
- Monitored Values {Parameter, Value}



- (Automatic) Mapping of the Data Model in :

  XML, SQL, SOAP, …

- Configuration & Results objects are store in a DB

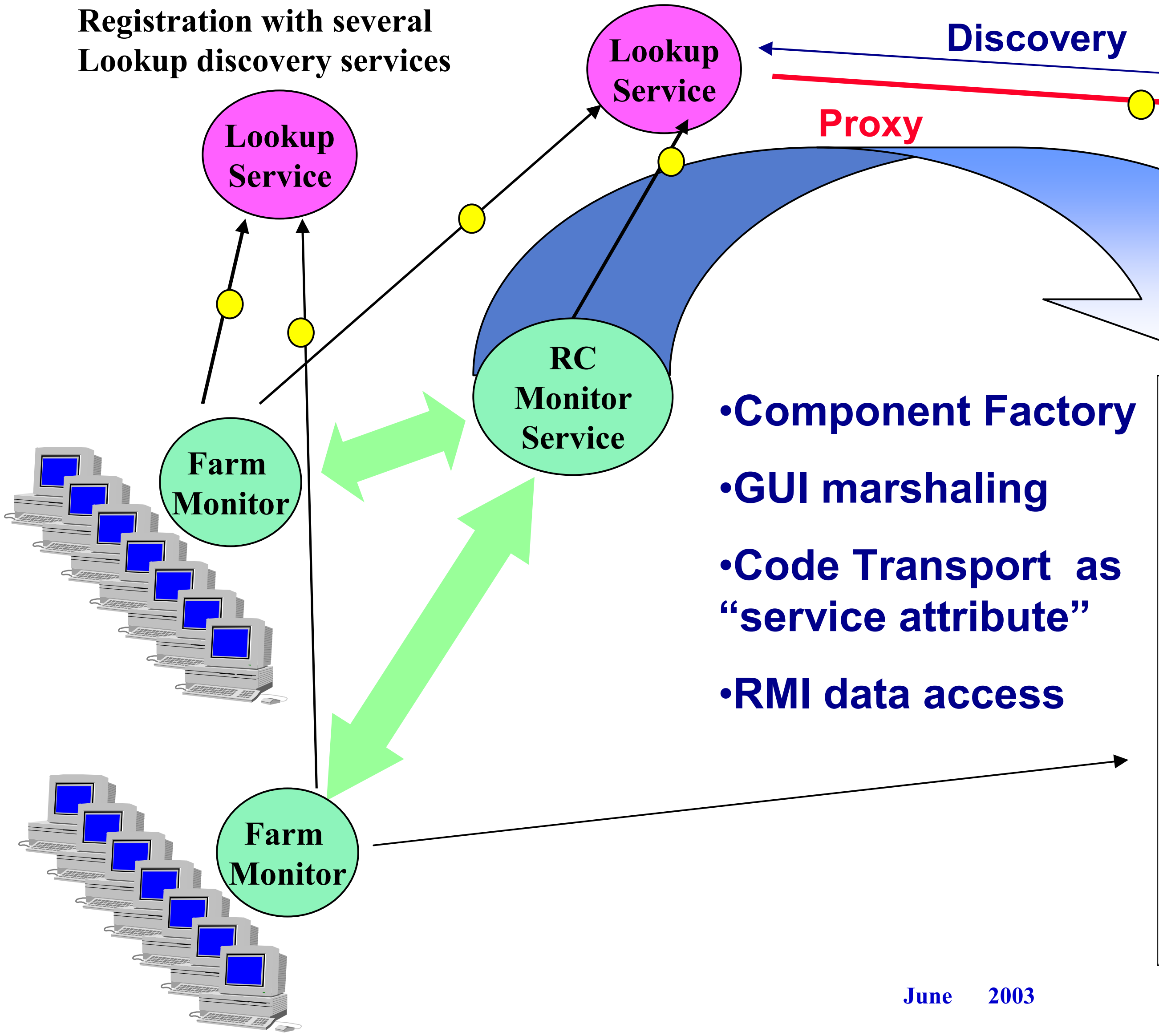
  (dynamically configurable for InstanDB, Postgres, MySQl, Oracle …)
- Subscription to results objects matching a template / predicate
- Clients can load filter objects into the Data Cache service and generate any derived (or aggregate) data structures and register to receive them.
- Monitored parameters have a life time

Data Collection Modules

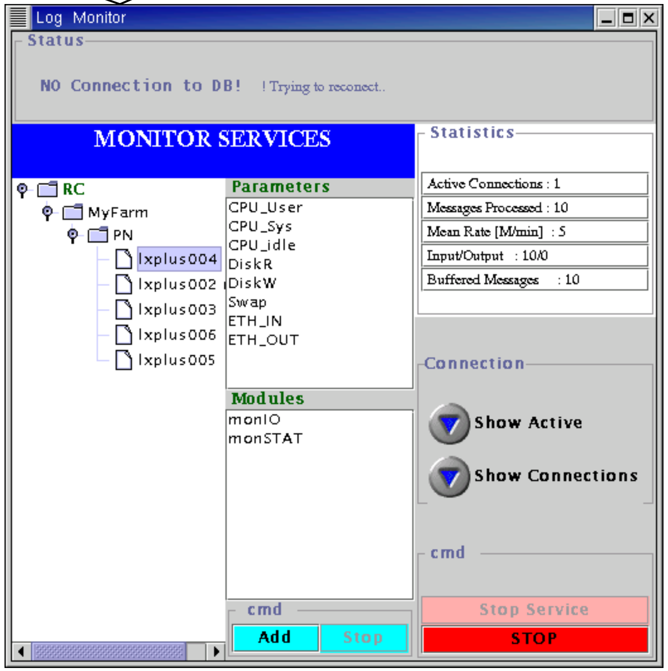
MonaLisa is a monitoring Framework:

- 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



NO Connection to DB! ! Trying to reconect..

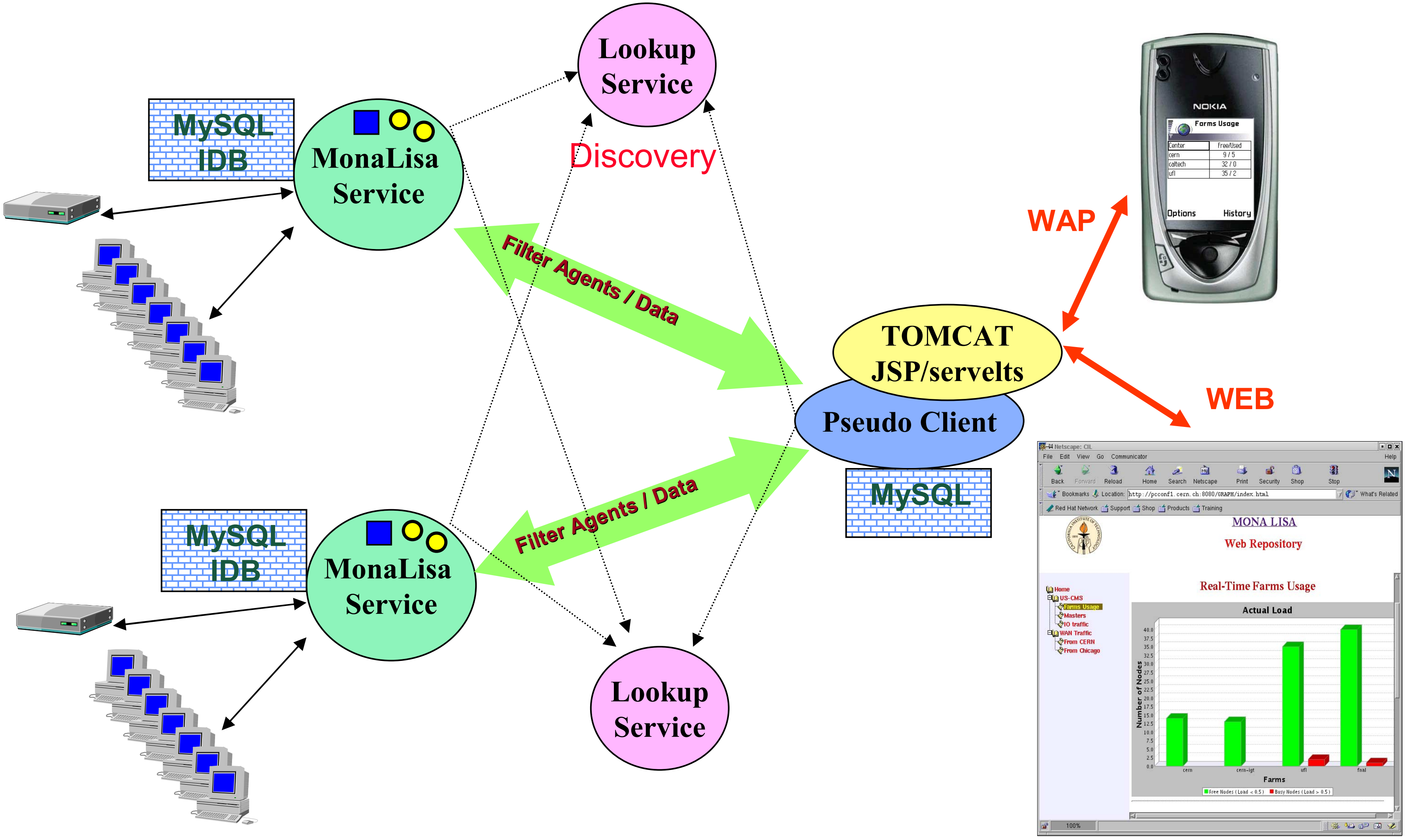
RC
MyFarm
PN
lxplus004
lxplus002
lxplus003
lxplus006
lxplus005

CPU_User
CPU_Sys
CPU_idle
DiskR
DiskW
Swap
ETH_IN
ETH_OUT

monIO
monSTAT

Active Connections : 1
Messages Processed : 10
Mean Rate [M/min] : 5
Input/Output : 10/0
Buffered Messages : 10

Pseudo – Clients & Dedicated Repositories



Farms Usage

Center	Free/Used
cern	9 / 5
caltech	32 / 0
ufl	35 / 2

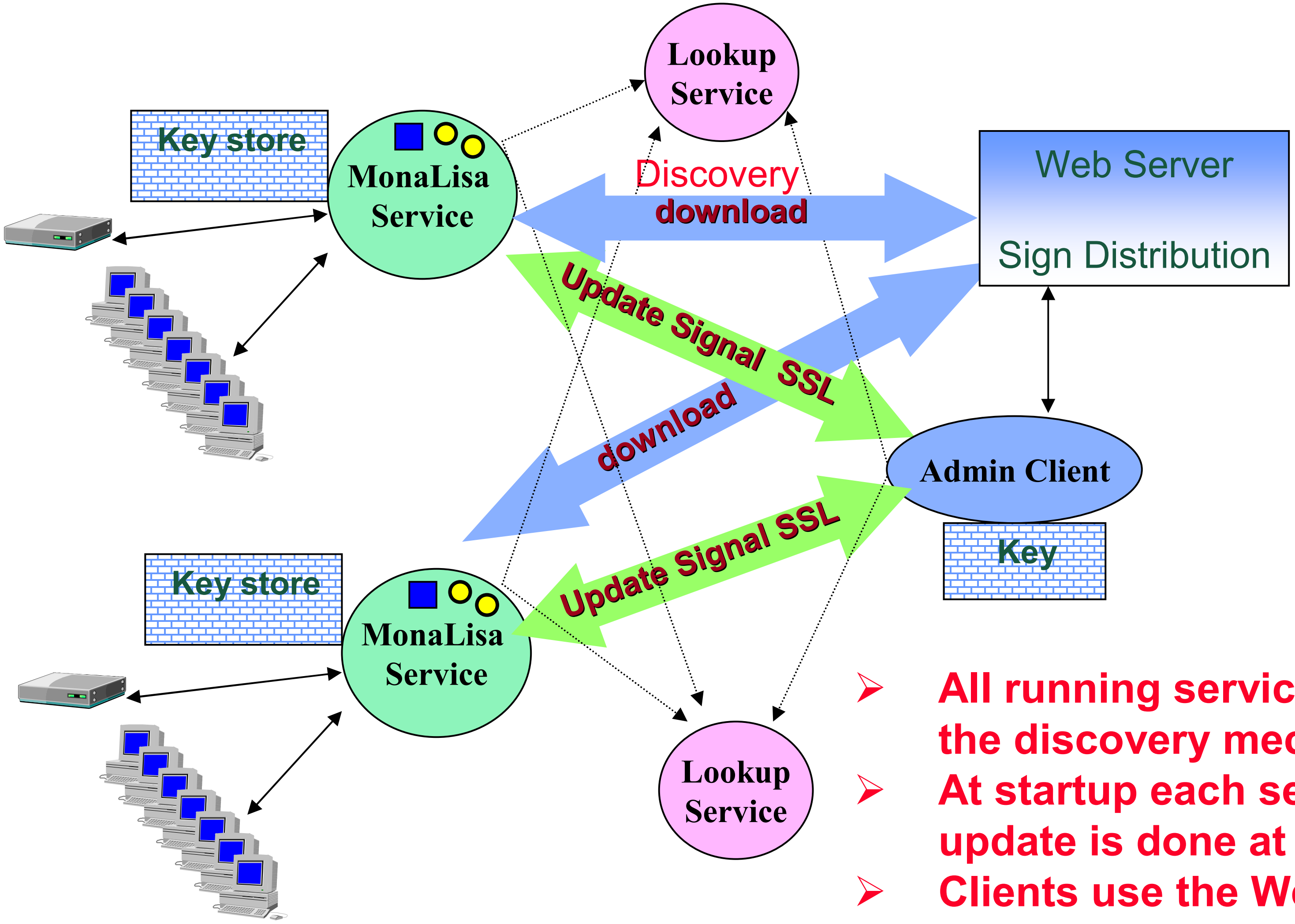
http://pcconf1.cern.ch:8080/GRAPH/index.html

MONA LISA

Web Repository

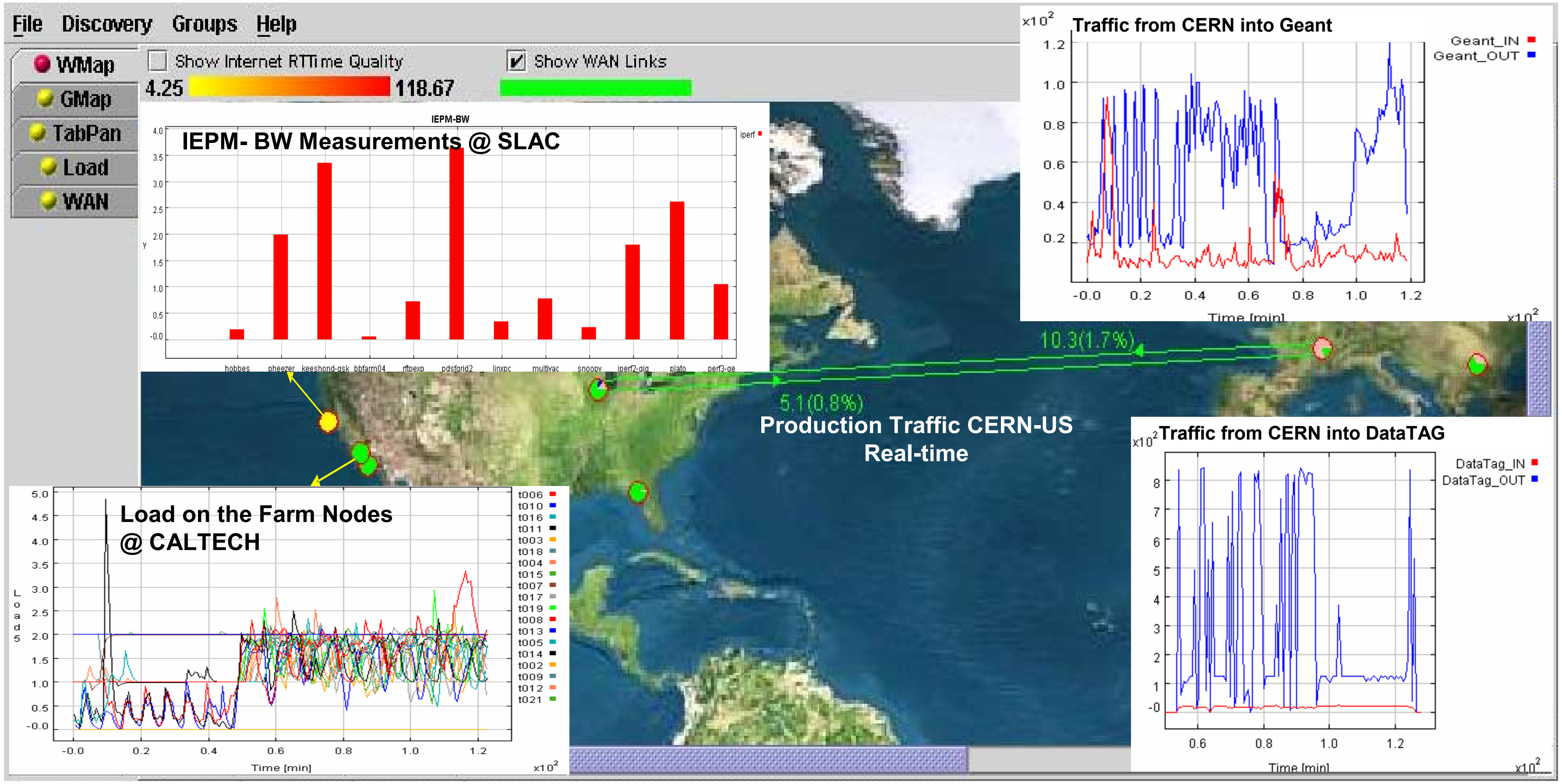
Real-Time Farms Usage

Secure – Automatic Update Mechanism for Services and Clients



- ➢ All running services are update using the discovery mechanism
- ➢ At startup each service check if it an update is done at a set of Web Servers
- ➢ Clients use the Web Start mechanism

Global Client / Dynamic Discovery



☐ Show Internet RTTime Quality

☑ Show WAN Links

IEPM- BW Measurements @ SLAC

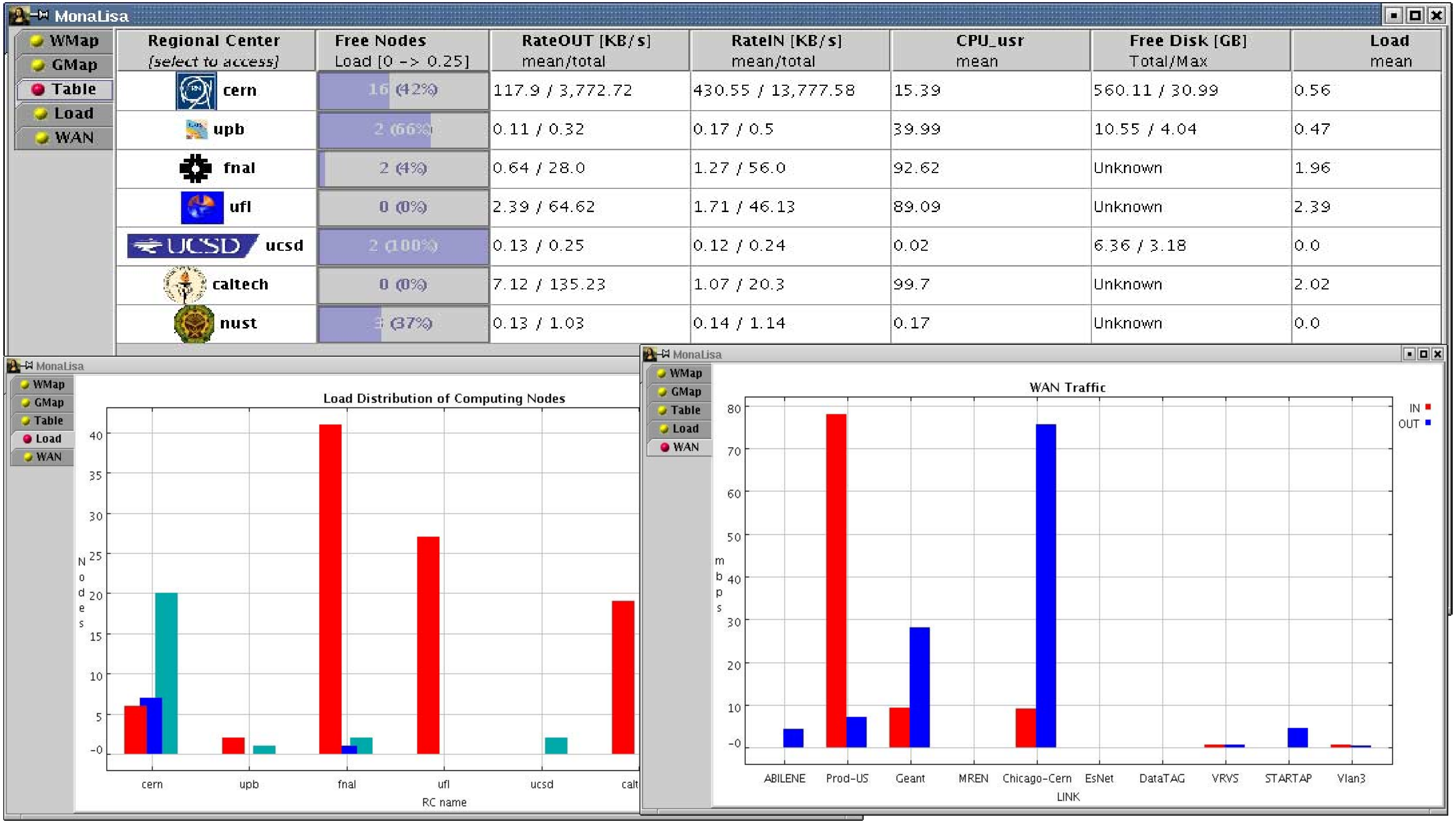
Traffic from CERN into Geant

Production Traffic CERN-US
Real-time

Traffic from CERN into DataTAG

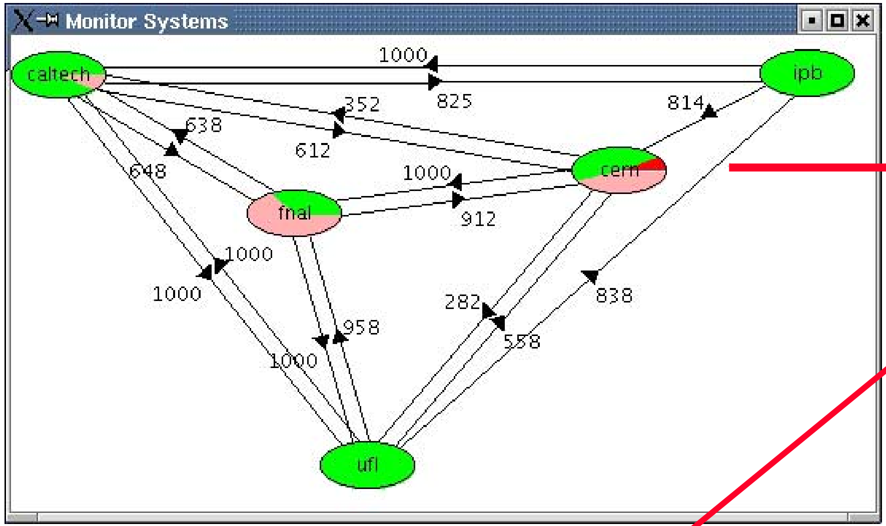
Load on the Farm Nodes
@ CALTECH

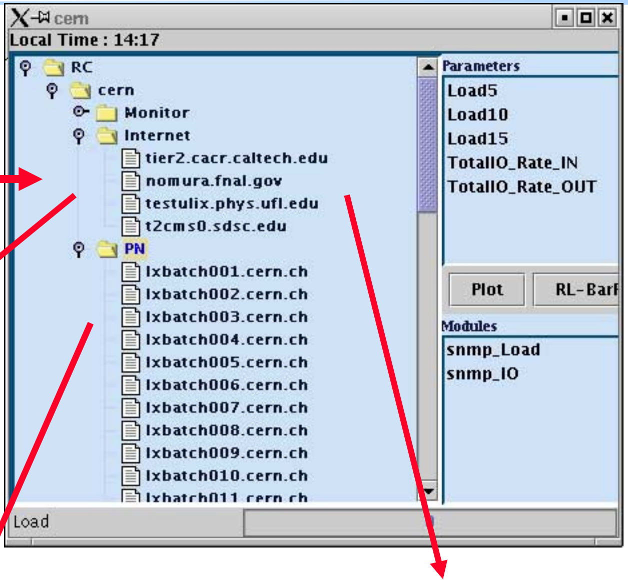
Global Views : CPU, IO, Disk, Internet Traffic …



Regional Center (select to access)	Free Nodes Load [0 -> 0.25]	RateOUT [KB/s] mean/total	RateIN [KB/s] mean/total	CPU_usr mean	Free Disk [GB] Total/Max	Load mean
cern	16 (42%)	117.9 / 3,772.72	430.55 / 13,777.58	15.39	560.11 / 30.99	0.56
upb	2 (66%)	0.11 / 0.32	0.17 / 0.5	39.99	10.55 / 4.04	0.47
fnal	2 (4%)	0.64 / 28.0	1.27 / 56.0	92.62	Unknown	1.96
ufl	0 (0%)	2.39 / 64.62	1.71 / 46.13	89.09	Unknown	2.39
ucsd	2 (100%)	0.13 / 0.25	0.12 / 0.24	0.02	6.36 / 3.18	0.0
caltech	0 (0%)	7.12 / 135.23	1.07 / 20.3	99.7	Unknown	2.02
nust	3 (37%)	0.13 / 1.03	0.14 / 1.14	0.17	Unknown	0.0

Regional Centers Discovery & Data access




cern

Local Time : 14:17

- RC
  - cern
    - Monitor
    - Internet
      - tier2.cacr.caltech.edu
      - nomura.fnal.gov
      - testulix.phys.ufl.edu
      - t2cms0.sdsc.edu
    - PN
      - lxbatch001.cern.ch
      - lxbatch002.cern.ch
      - lxbatch003.cern.ch
      - lxbatch004.cern.ch
      - lxbatch005.cern.ch
      - lxbatch006.cern.ch
      - lxbatch007.cern.ch
      - lxbatch008.cern.ch
      - lxbatch009.cern.ch
      - lxbatch010.cern.ch
      - lxbatch011.cern.ch

Parameters

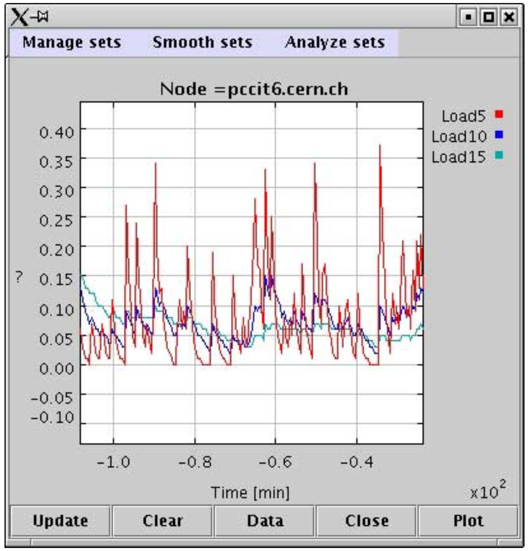
- Load5
- Load10
- Load15
- TotalIO_Rate_IN
- TotalIO_Rate_OUT

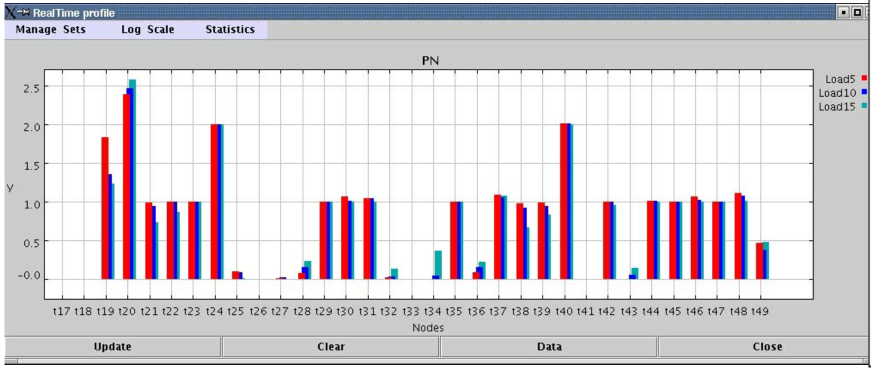
Plot | RL-Barl

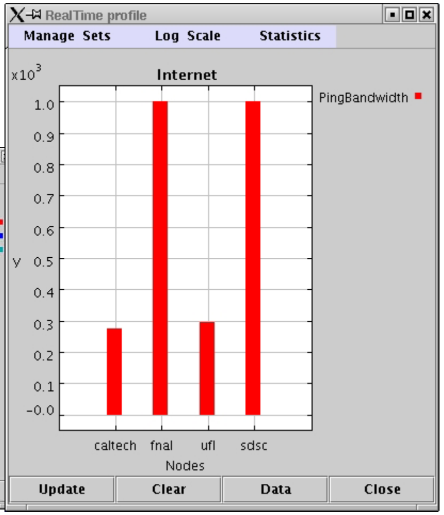
Modules

- snmp_Load
- snmp_IO

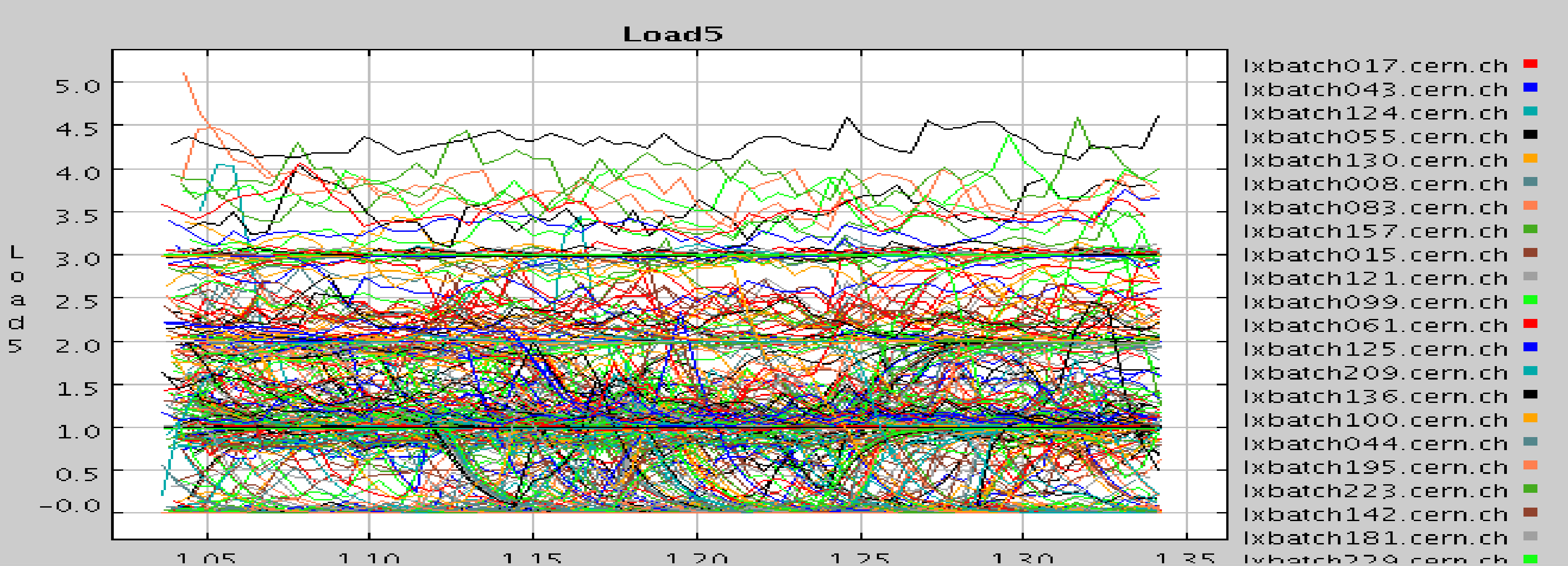
Load

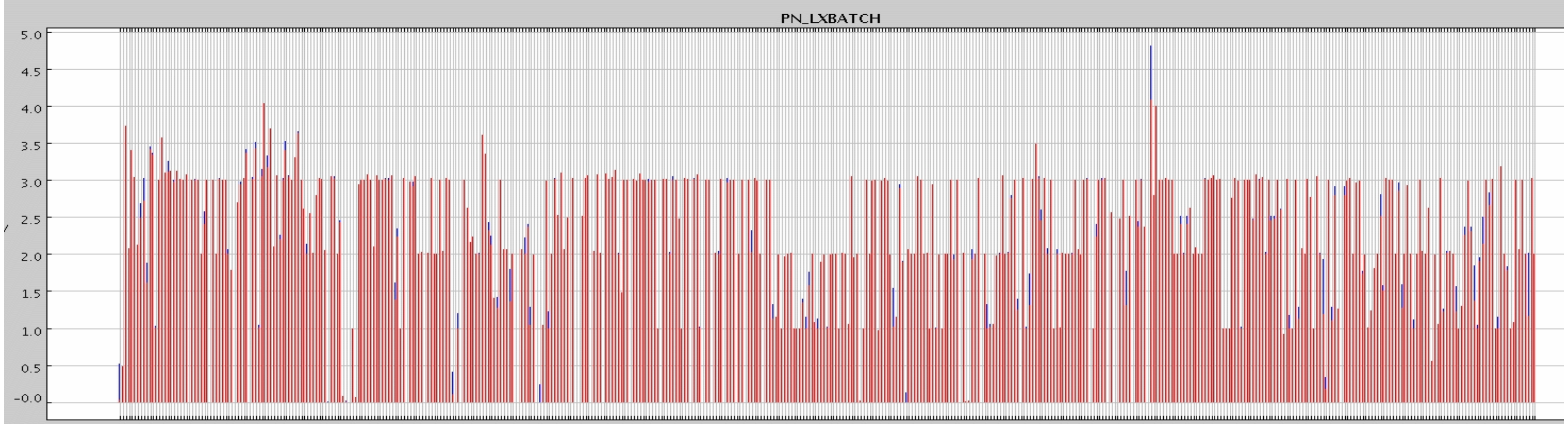




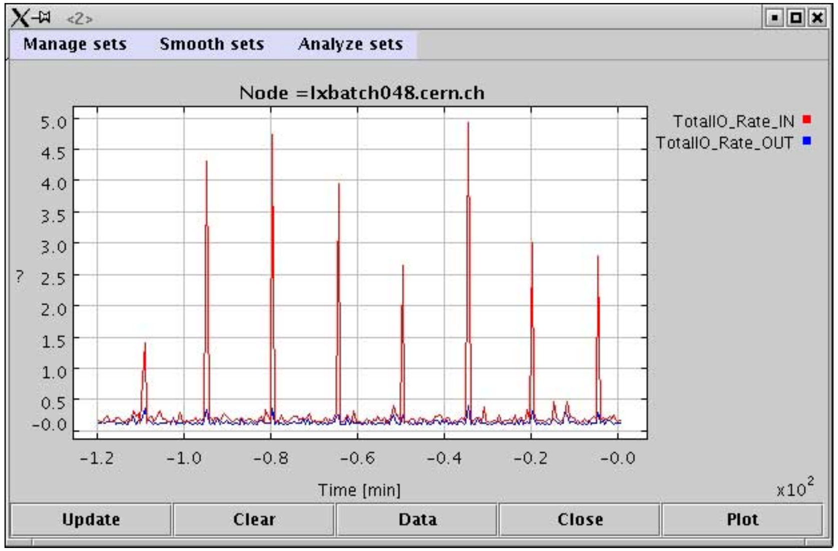


Real-time Data for Large Systems
"lxshare" cluster at cern ~ 600 ndoes

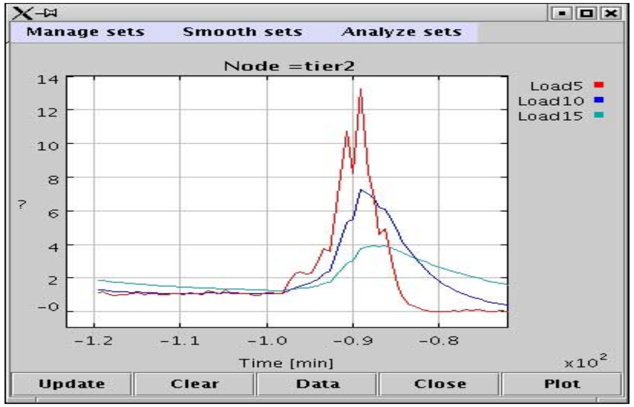




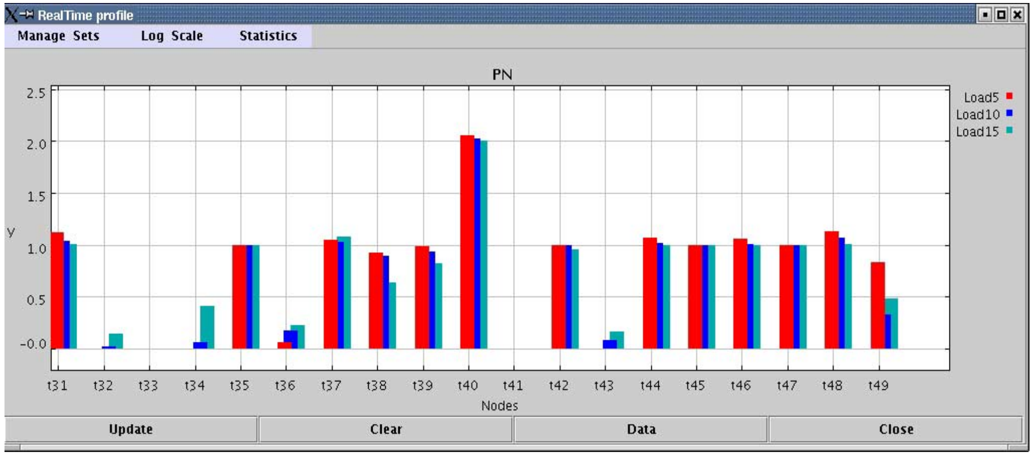
Access to historical and real-time values

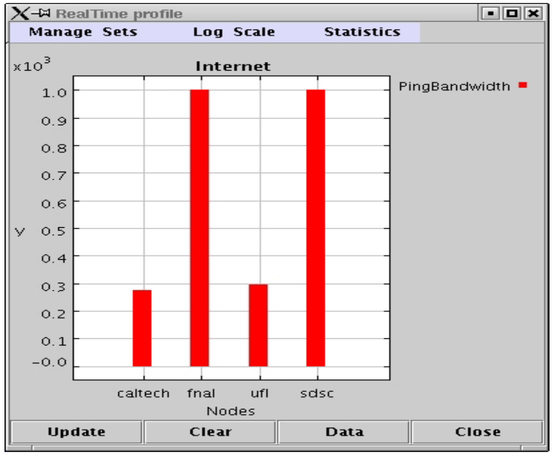


Past values are presented and the GUI remains a registered listener and the new vales are added

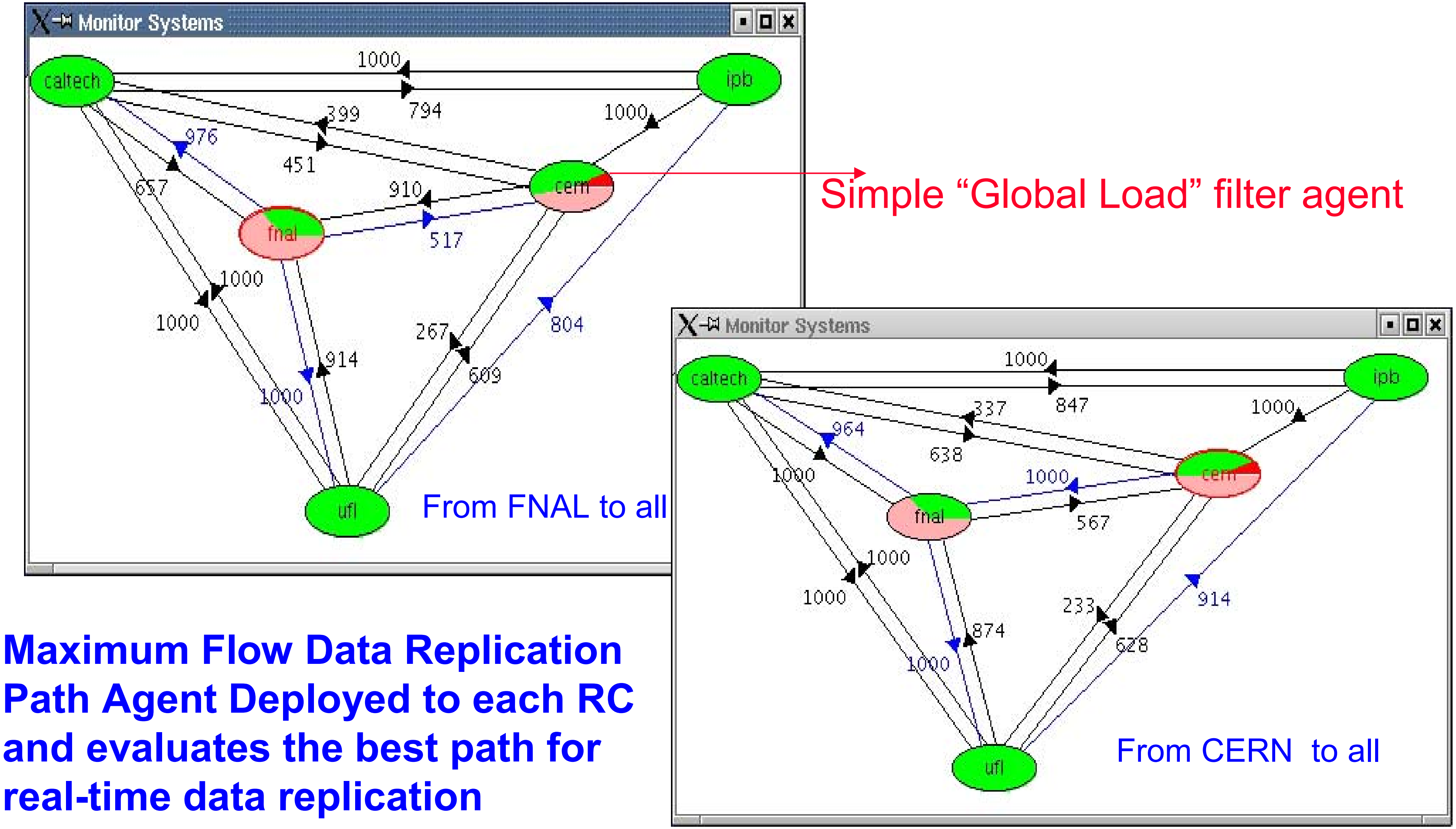


Real Time Histograms for various parameters





Filter Mobile Agents



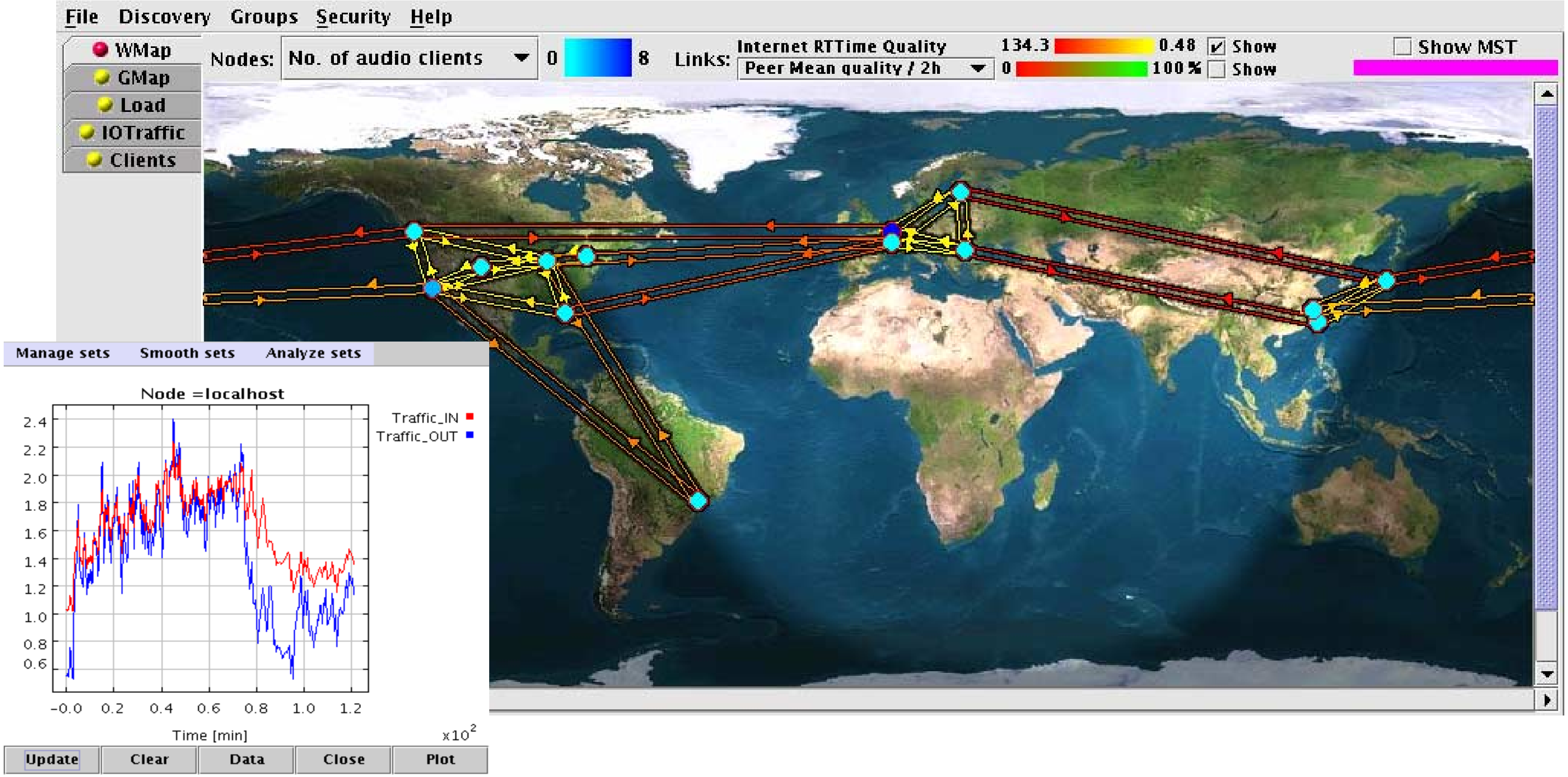
From FNAL to all

From CERN to all

Simple “Global Load” filter agent

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

Monitoring VRVS Reflectors



Nodes: No. of audio clients 0 8

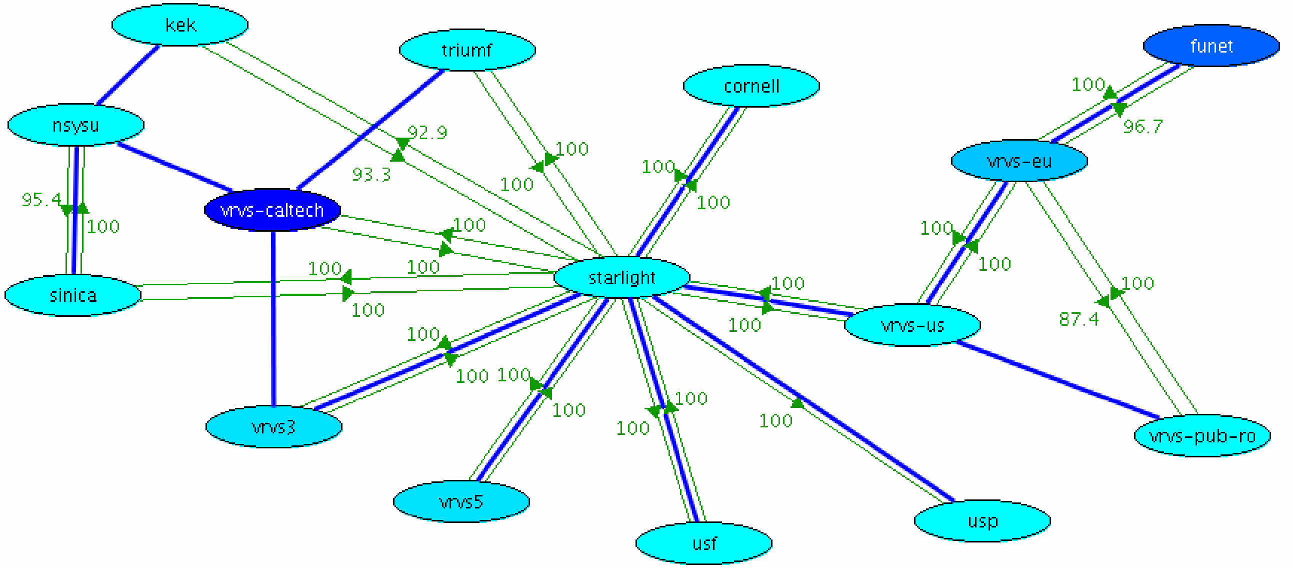
Links: Internet RTTime Quality 134.3 0.48 ☑ Show

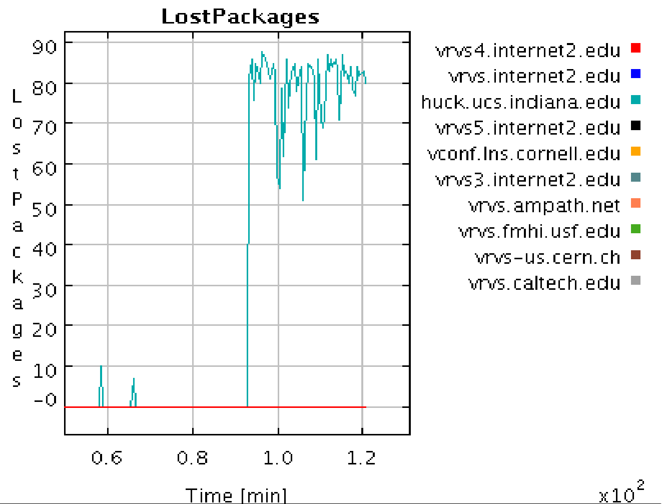
Peer Mean quality / 2h 0 100 % □ Show

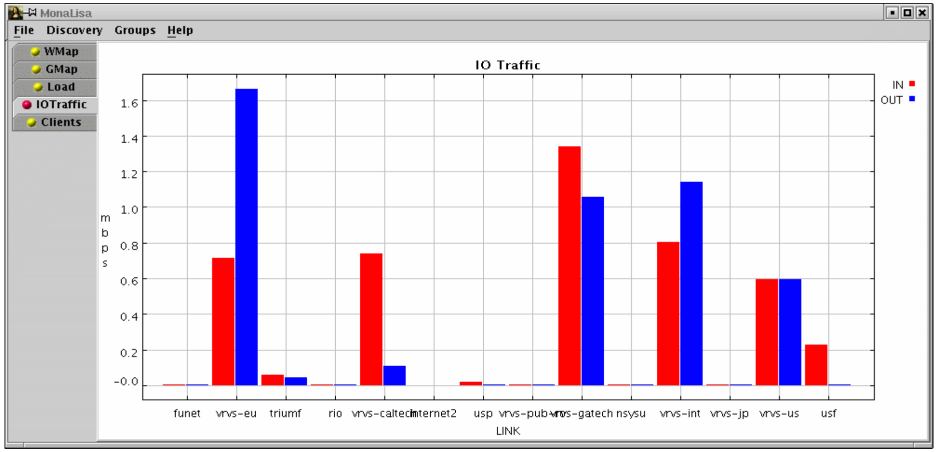
□ Show MST

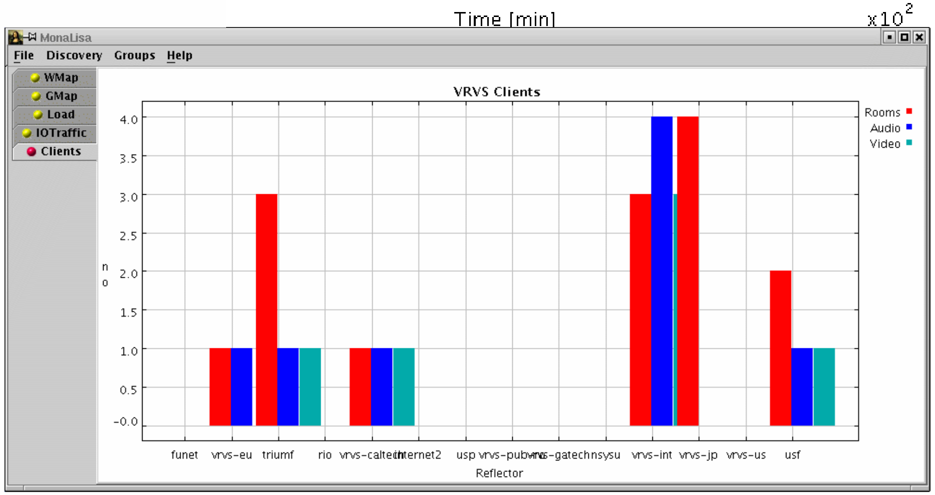
Node =localhost

Monitoring VRVS Reflectors (2)



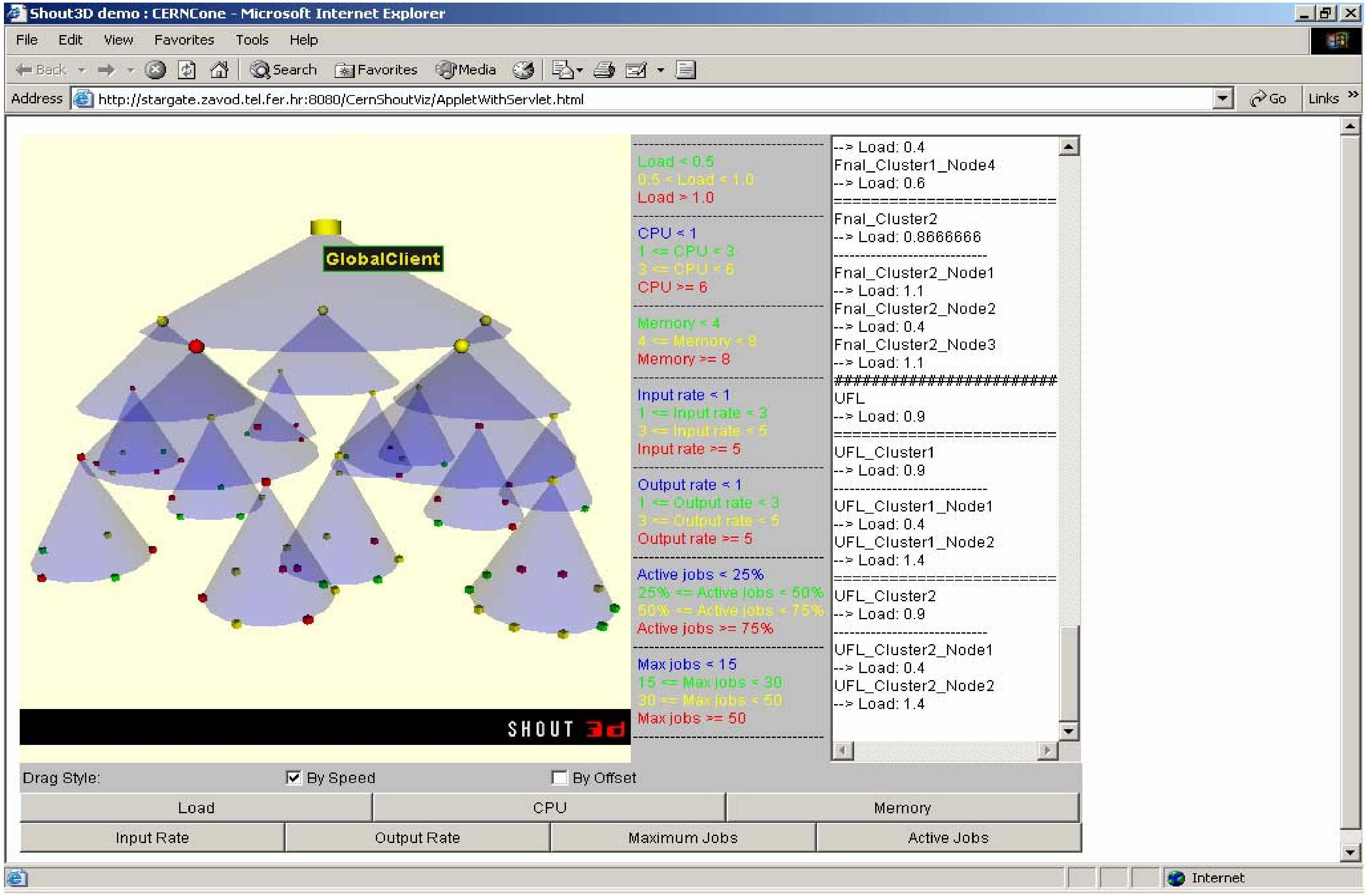






PDA 3 D MonALISA Client

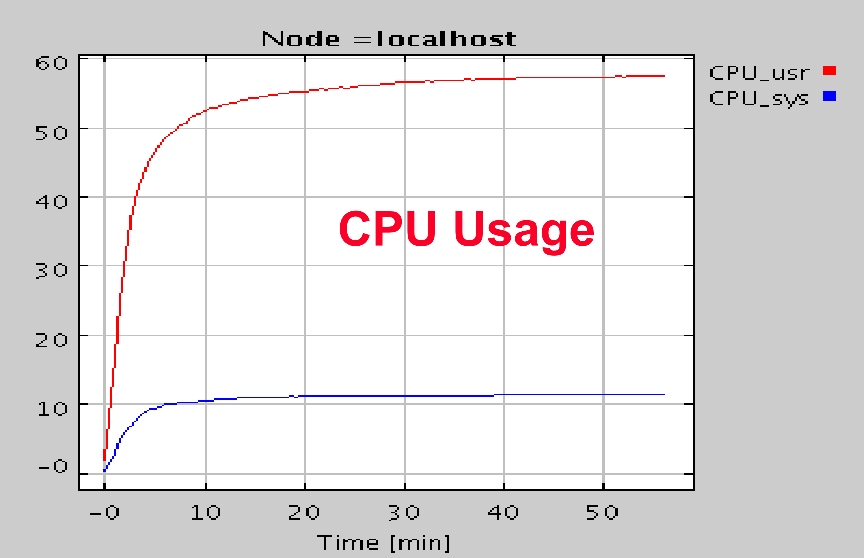
Developed by ERICSSON Research Lab

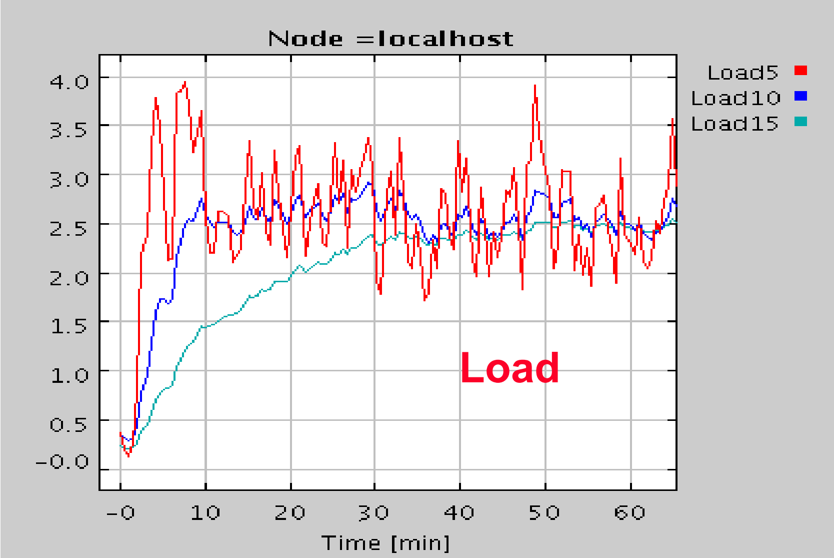


```
--> Load: 0.4
Fnal_Cluster1_Node4
--> Load: 0.6
==========================
Fnal_Cluster2
--> Load: 0.8666666
----------------------------
Fnal_Cluster2_Node1
--> Load: 1.1
Fnal_Cluster2_Node2
--> Load: 0.4
Fnal_Cluster2_Node3
--> Load: 1.1
##########################
UFL
--> Load: 0.9
==========================
UFL_Cluster1
--> Load: 0.9
----------------------------
UFL_Cluster1_Node1
--> Load: 0.4
UFL_Cluster1_Node2
--> Load: 1.4
==========================
UFL_Cluster2
--> Load: 0.9
----------------------------
UFL_Cluster2_Node1
--> Load: 0.4
UFL_Cluster2_Node2
--> Load: 1.4
```

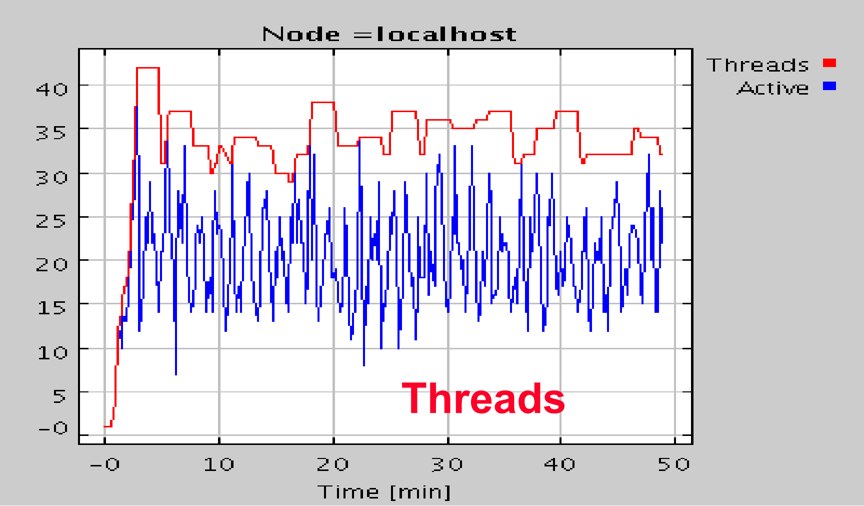
Performance Test: snmp query (~200 metrics values) on a 500 nodes farm every 60 s.

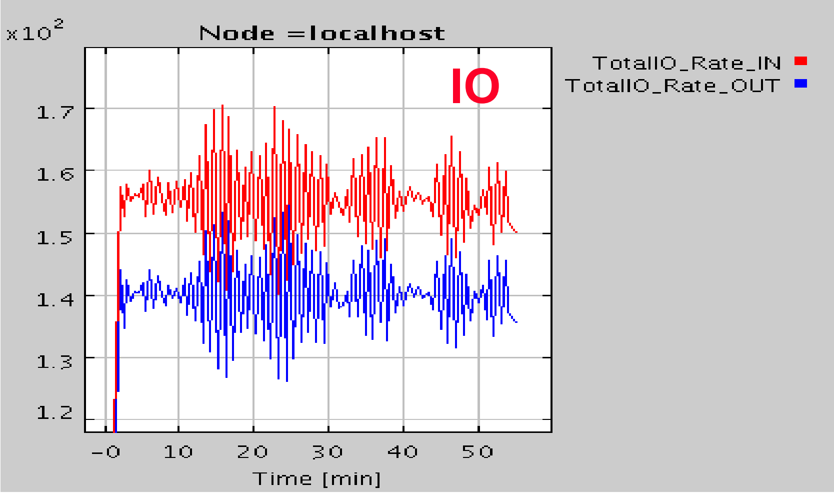
~ 1600 metrics values collected per second.



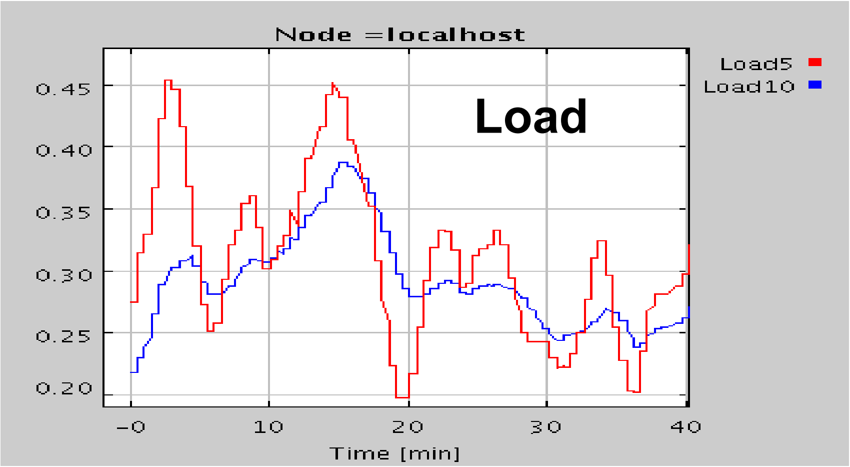


Dell I8100 ~ 1GHz

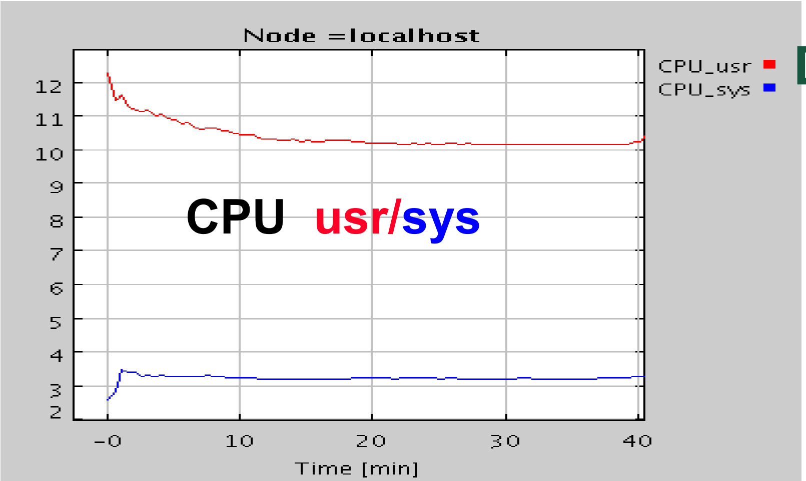




Performance Test: High rate snmp queries (every 15 s) on a large farm (500 nodes)

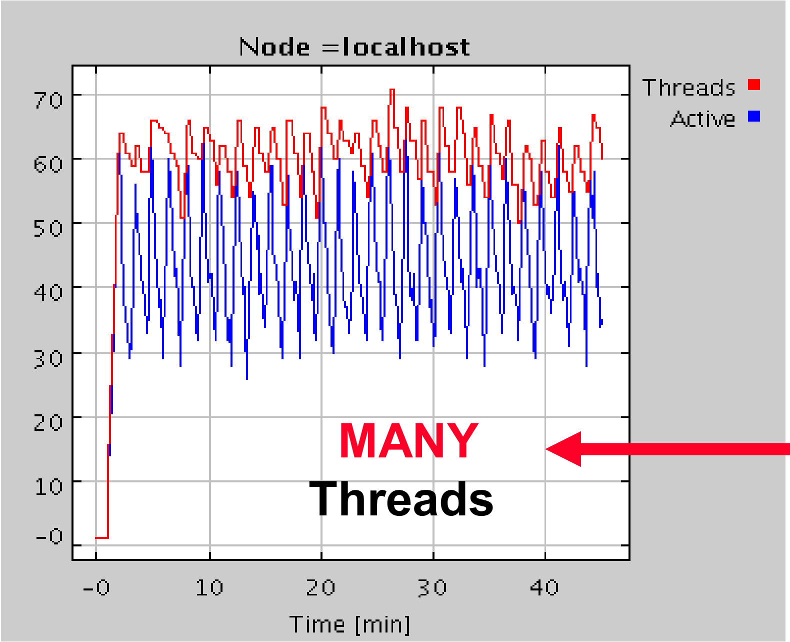


Load

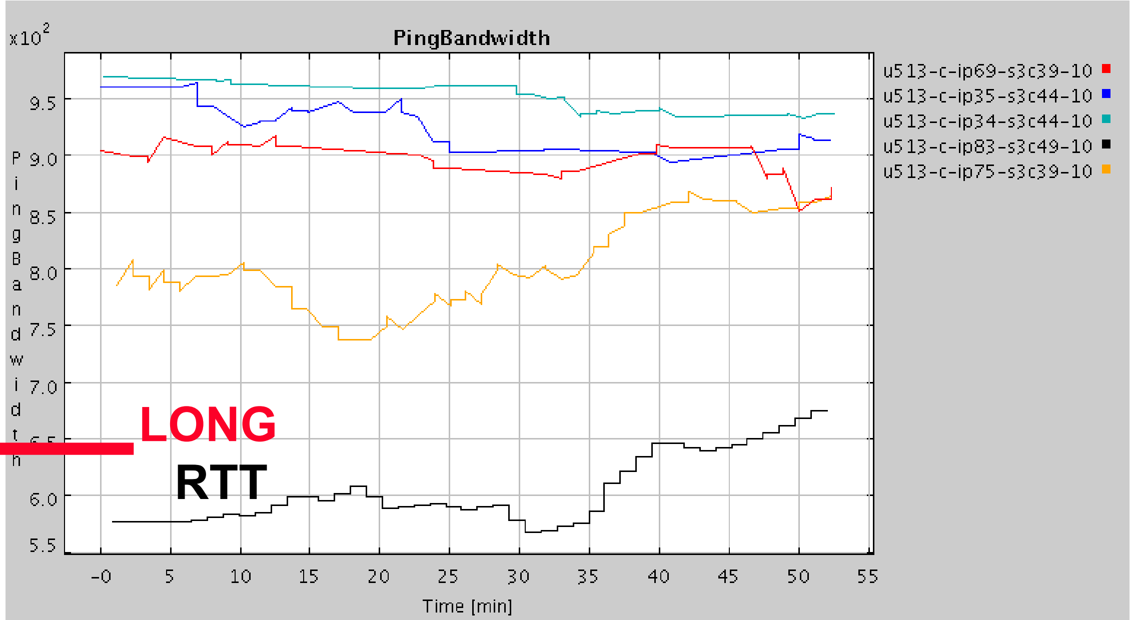


CPU usr/sys

Dell I8100 ~ 1GHz



MANY Threads



LONG RTT

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) .
- Dynamic configuration for farms / network elements. Secure Admin interface.
- 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

CONCLUSIONS

- 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