Monitoring the Grid

Costin Grigoras <costin.grigoras@cern.ch>

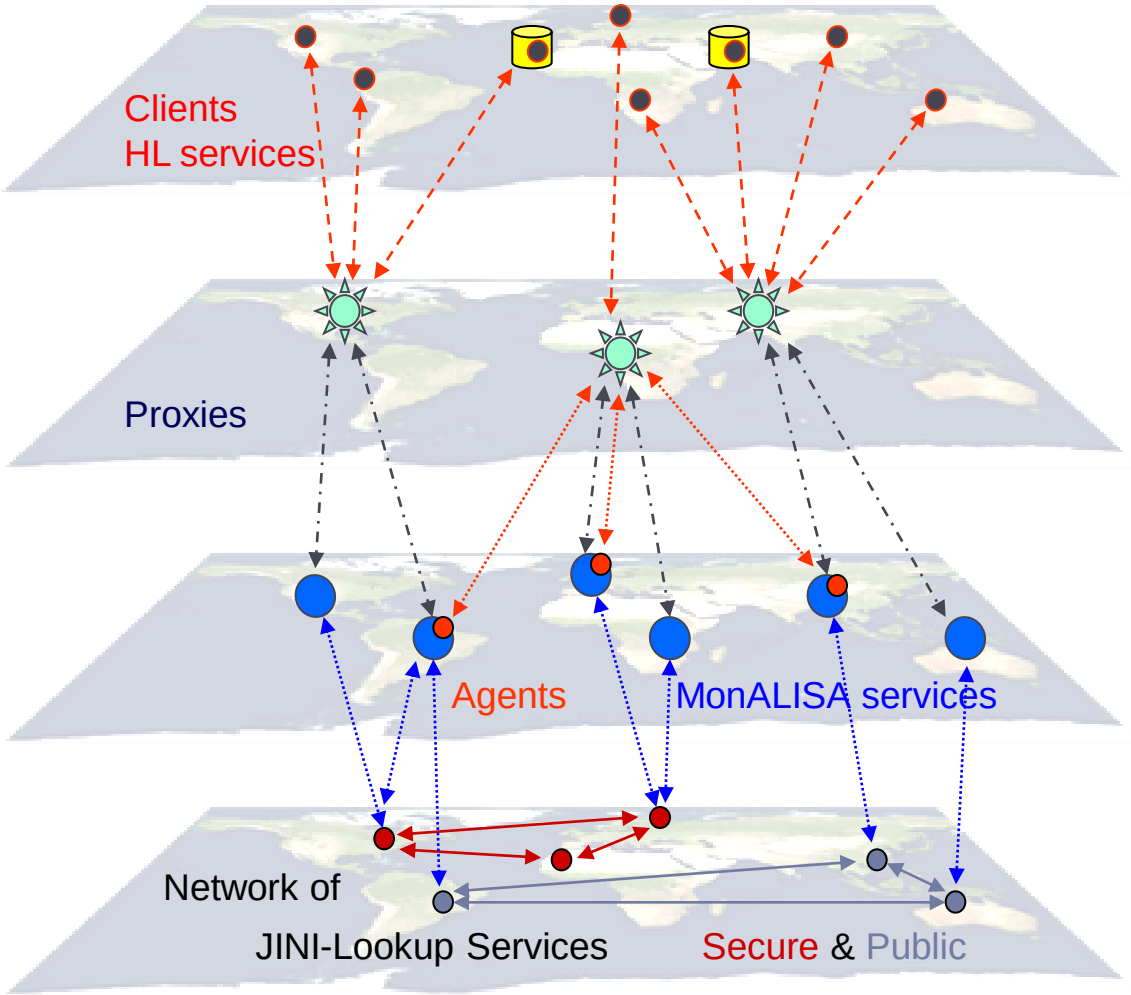
Challenge

- ALICE Grid means
  - 80 Computing Elements on 5 continents
    - More than 35000 CPU cores
  - 60 Storage Elements (8 tape-backed)
    - 12PB in use (72% is stored on tape SEs)
  - 350 users active on Grid
- … and we have to understand the system and optimize it
- For this we are using MonALISA

What is MonALISA ?

- Caltech project started in 2002
  http://monalisa.caltech.edu/
- Java-based set of distributed, self-describing services
- Offers the infrastructure to collect any type of information
- All data can be processed in near real time
- The services can cooperate in performing the monitoring tasks
- Can act as a platform for running distributed user agents

MonALISA software components and the connections between them



Data consumers

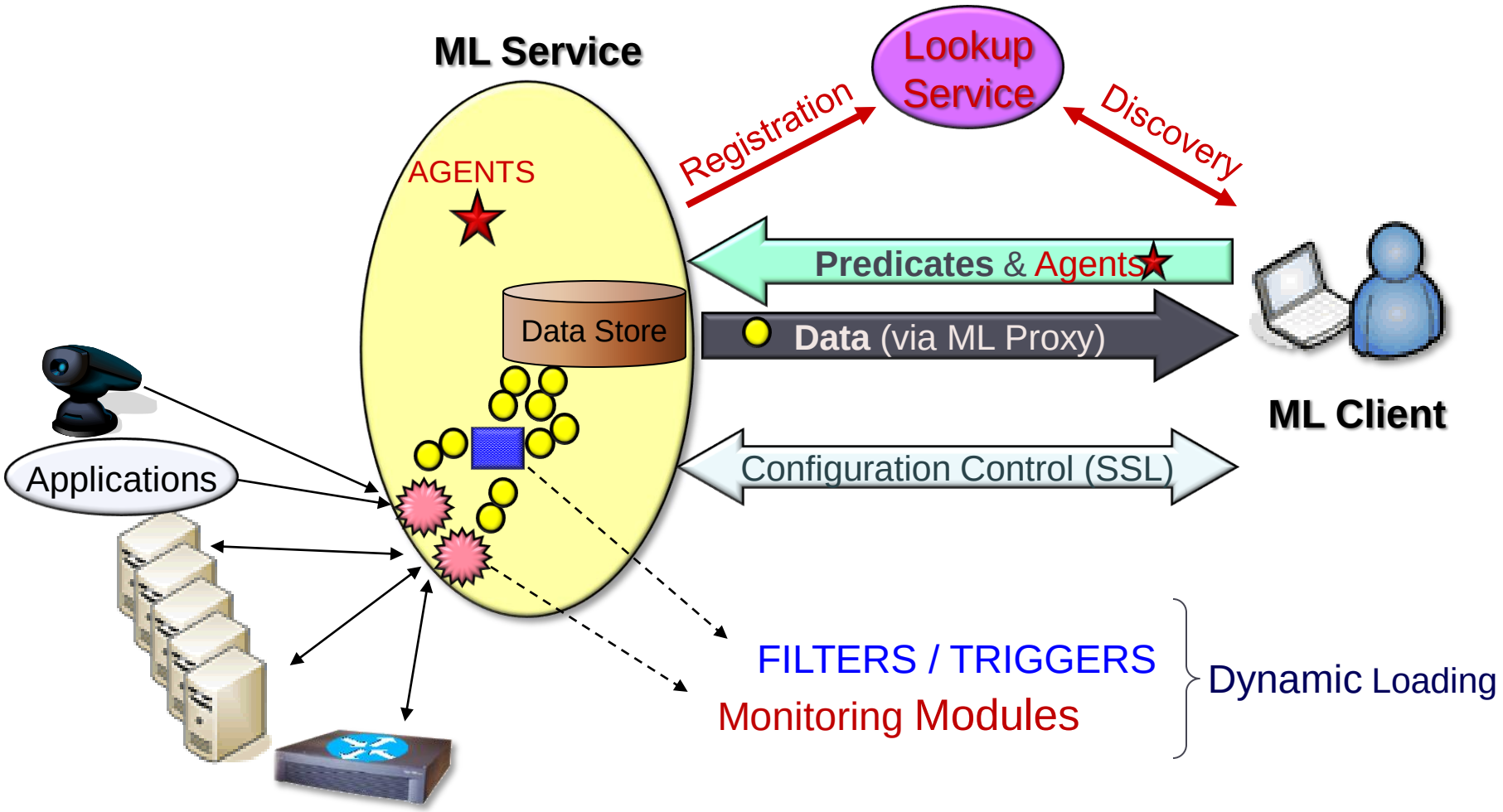
Multiplexing layer
Helps firewalled endpoints connect

Data gathering services

Registration and discovery

Fully Distributed System with no Single Point of Failure

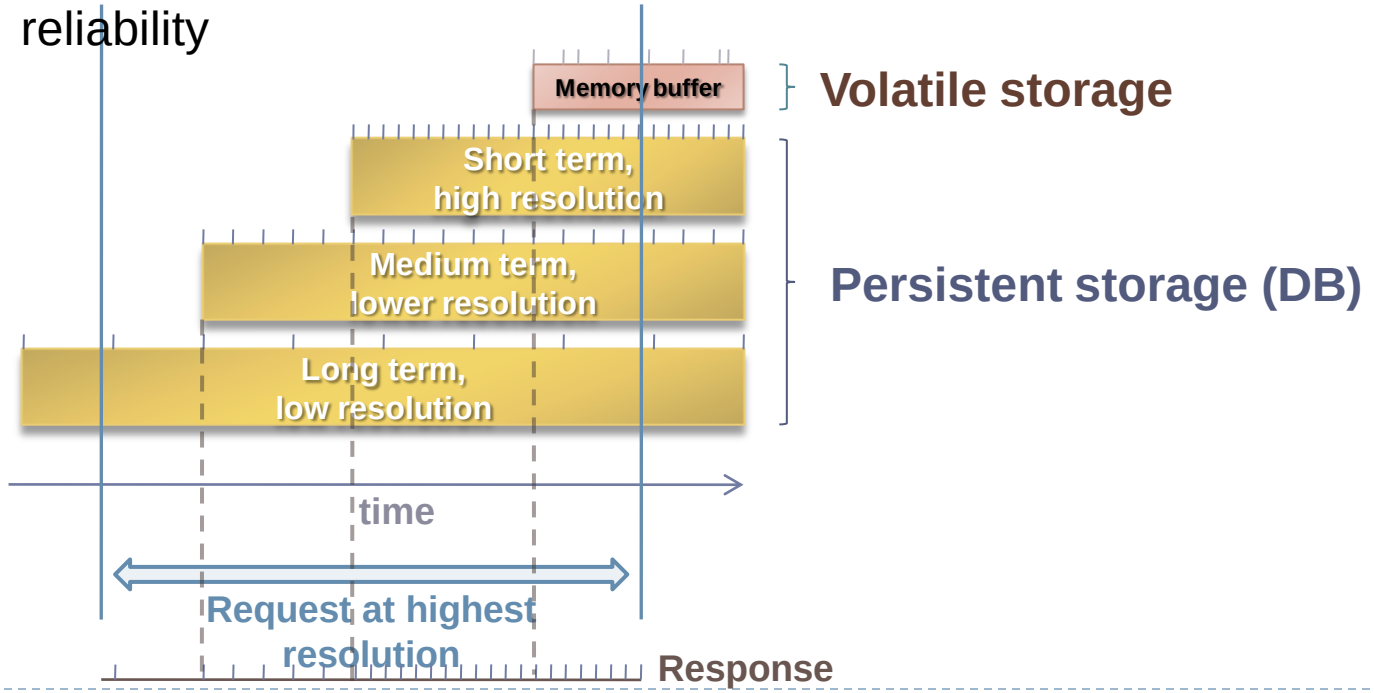
Subscriber/notification paradigm



Push or Pull, depending on device

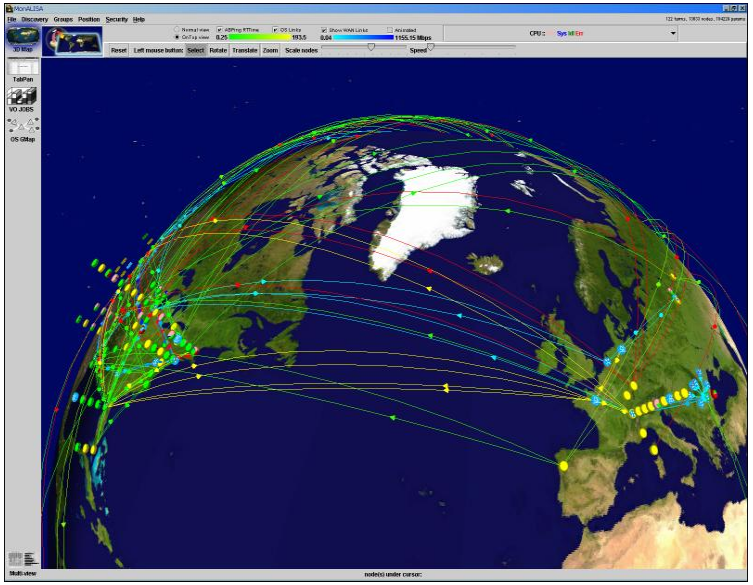
Data storage model

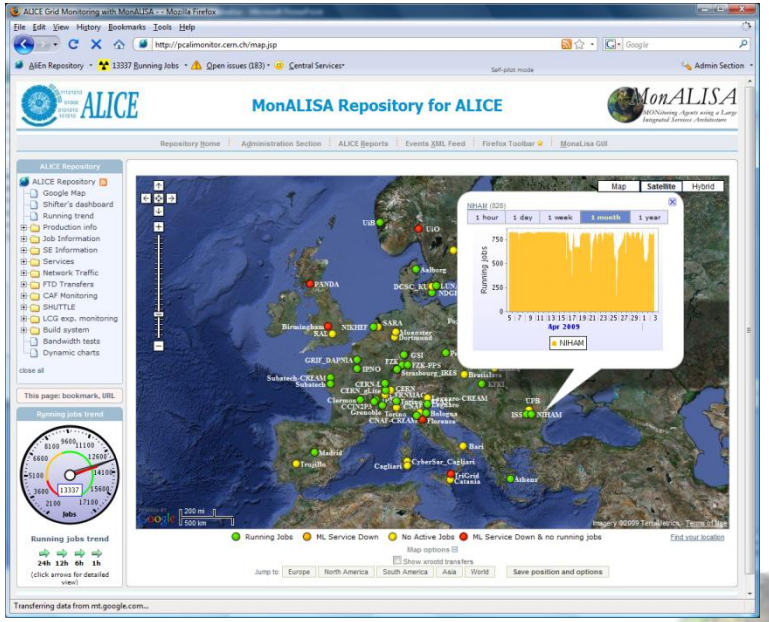
- MonALISA keeps a memory buffer for a minimal monitoring history
- In addition, data can be kept in configurable database structures
- Default configuration is adapted to the different use cases
  - the service keeps one week of raw data and one month of averaged values
  - the repository creates three averaged structures (2mo @ 2m, 1y @ 30m, 10y @ 2.5h)
- Parallel database backends can be used to increase performance and reliability



Clients

- **GUI client**
  - Interactive exploring of all the parameters
  - Can plot history or real-time values
  - Customizable history query interval
  - Subscribes to those particular series and updates the plots in real time
- **Storage client (aka Repository)**
  - Subscribes to a set of parameters and stores them in database structures suitable for long-term archival
  - Is usually complemented by a web interface presenting these values
  - Can also be embedded in another controlling application
- WebServices & REST clients
  - Limited functionality: they lack the subscription mechanism





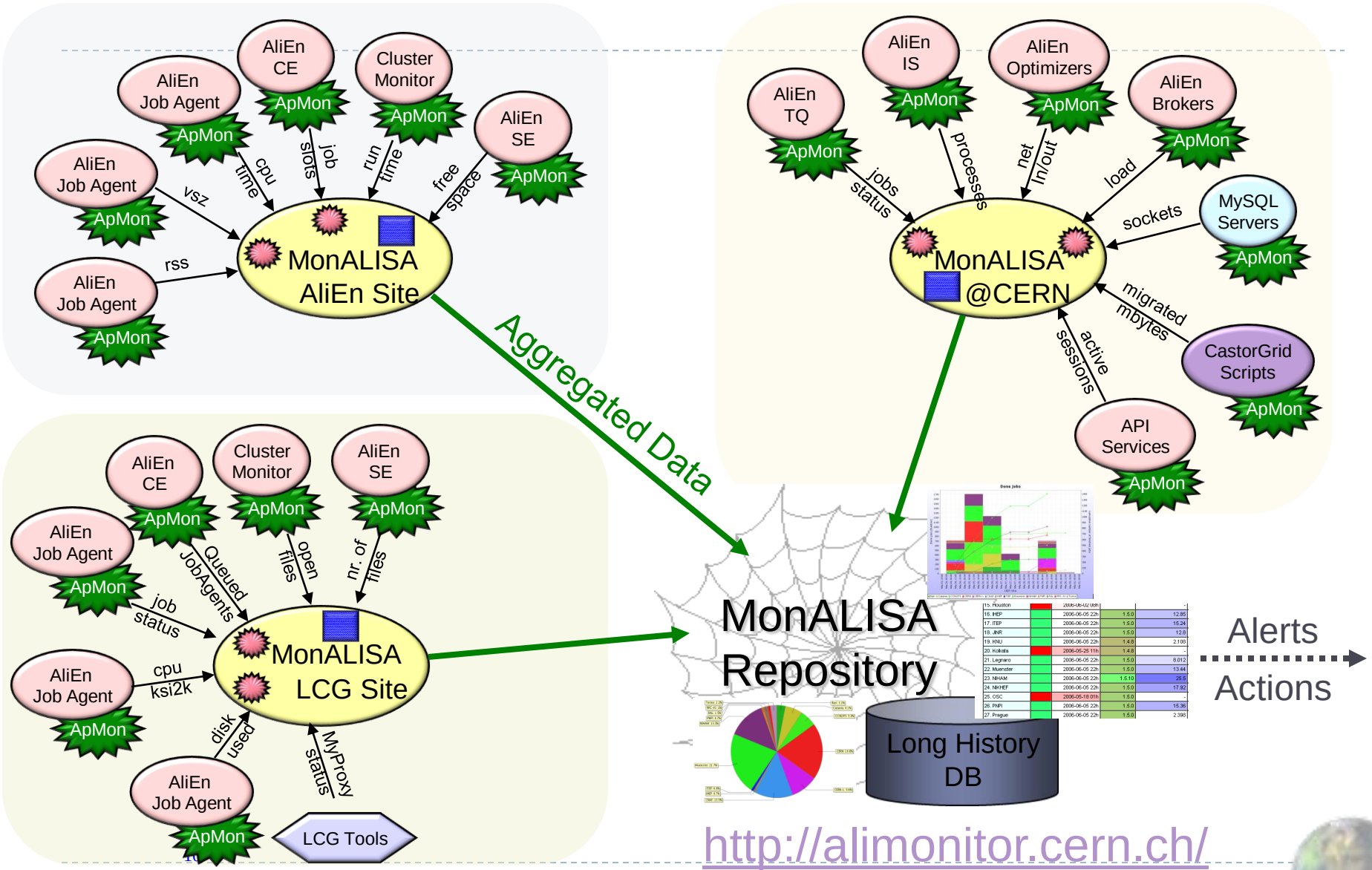
MonALISA service includes many modules; easily extendable

- The service package includes:
  - Local host monitoring (CPU, memory, network traffic , processes and sockets in each state, LM sensors, IPMI, APC UPSs), log files tailing
  - SNMP generic & specific modules;
  - Condor, PBS, LSF and SGE (accounting & host monitoring), Ganglia
  - Ping, tracepath, traceroute, pathload, xrootd
  - Ciena, Optical switches (TL1); Netflow/Sflow (Force10)
  - Calling external applications/scripts that output the values as text
  - XDR-formatted UDP messages (ApMon)
- New modules can be added by implementing a simple Java interface.
- Filters can also be defined to aggregate data in new ways
- The Service can also react to the monitoring data it receives through the action framework

Embeddable APlication MONitoring library (ApMon)

- Lightweight library of APIs (C, C++, Java, Perl, Python) that can be used to send any information to MonALISA Service(s) over UDP
- Flexible configuration (hardcoded / configuration file / URL)
- Background system monitoring (optional)
  - Load, CPU, memory & swap usage
  - Network interfaces (in/out/ip/errs)
  - Sockets in each state, processes in each state
  - Disk IO, swap IO
- Background application monitoring (optional)
  - Used CPU & wall time, % of the machine CPU
  - Partition stats, size of workdir, open files
  - Memory usage (rss, virtual and %), page faults
- Very high throughput (O(10K msg/s) on a regular machine)

AliEn monitoring architecture



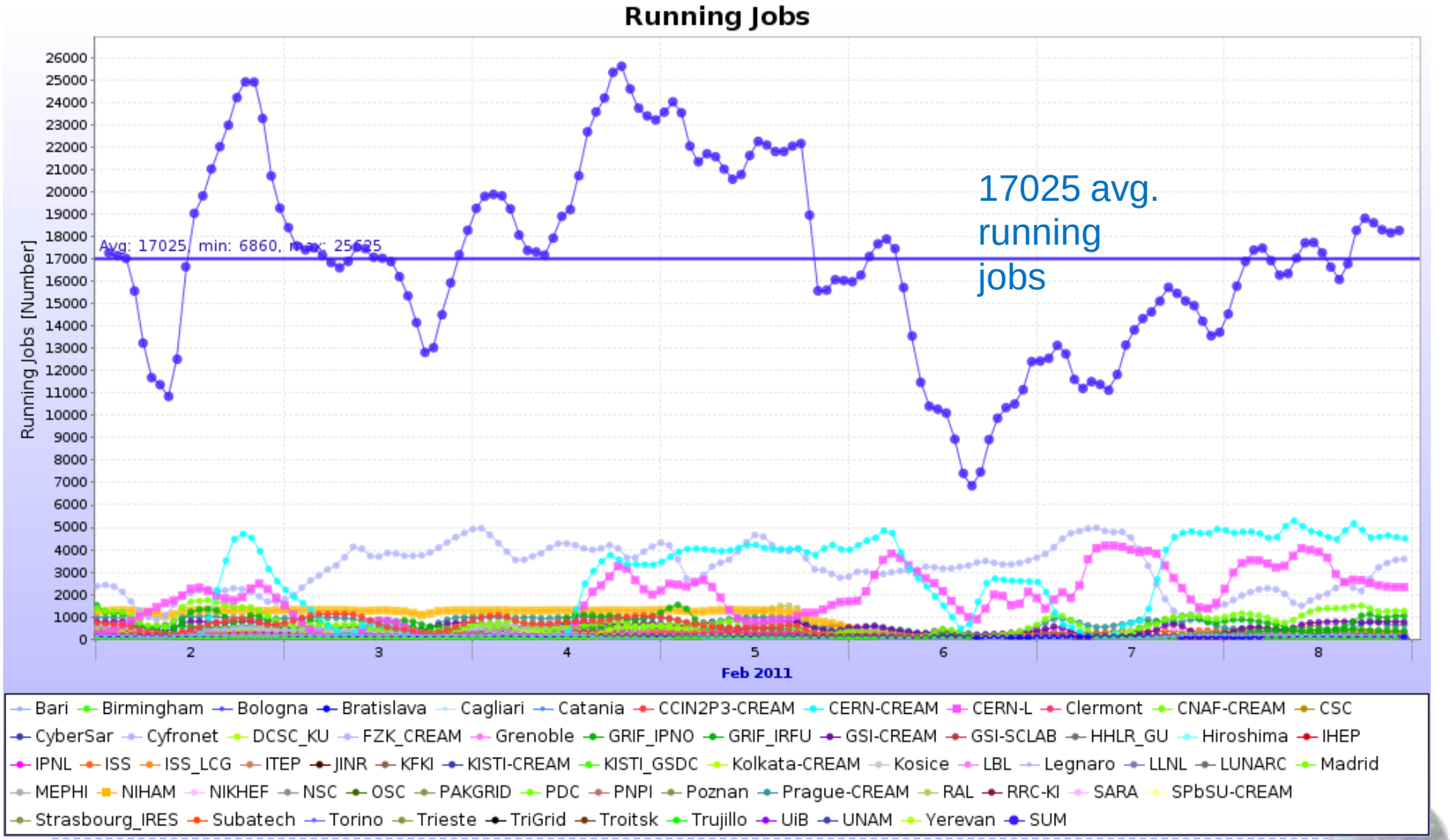
Monitoring statistics

- 116 active services (site + central services)
- 2.6M published parameters @ 20.5KHz
- Central repository subscribes to only 150K parameters
  - Mostly aggregated values
  - Storing at 450Hz
  - 7-10K dynamic pages / hour
    - Average generation time is below 0.1s
  - 320GB database size (with the data compaction scheme shown before)
    - 5 years of history

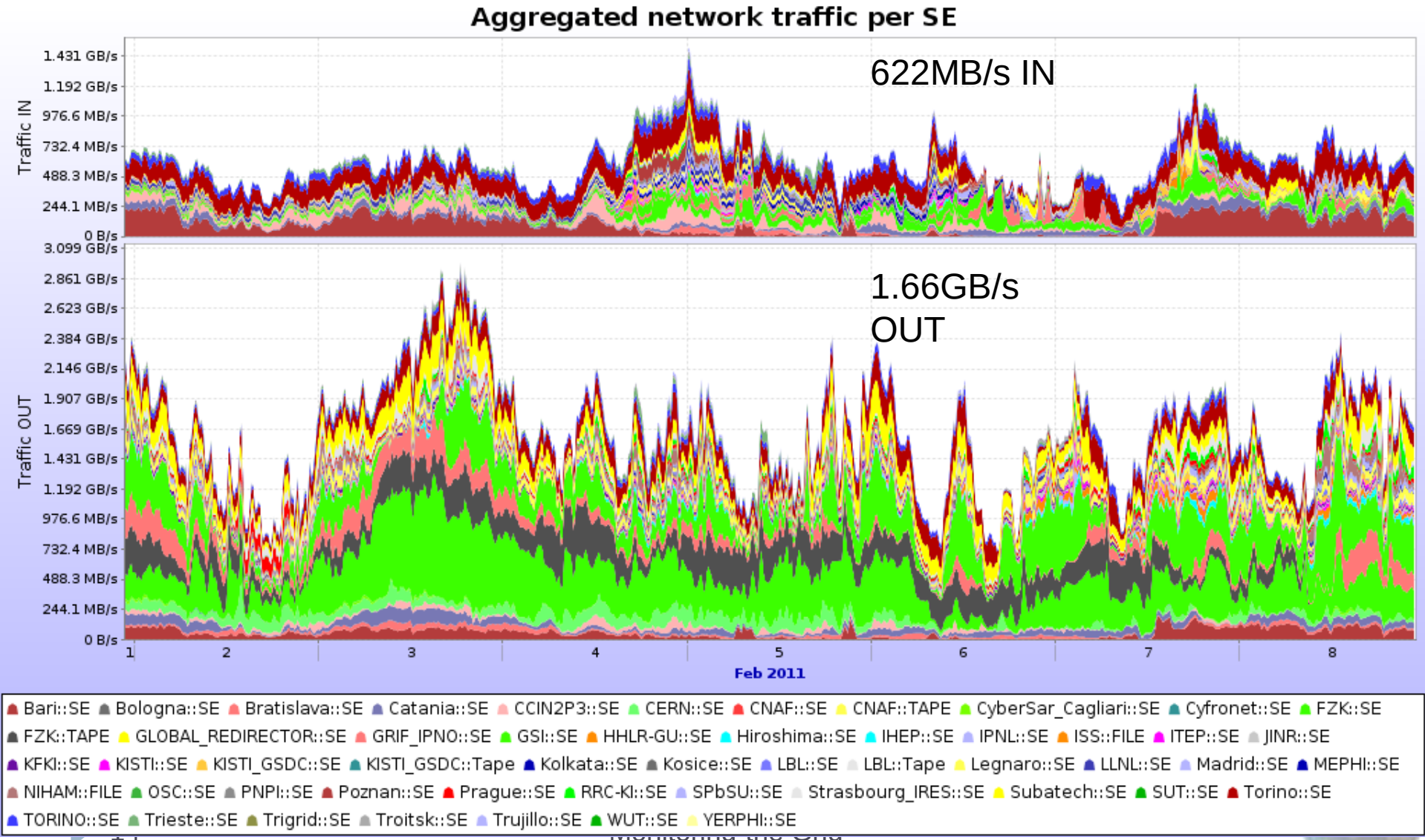
Site monitoring

- The sites can collect the local monitoring information to keep more details
  - Full host monitoring history (ApMon, Ganglia, snmp)
  - Fabric monitoring (snmp to network equipment, UPSs)
  - Job accounting information for the site
    - Users that run jobs on the site, CPU time, memory profiles, IO …
- Extending it with custom filters and alarms
- An example: GSI http://lxgrid2.gsi.de:8080/
- Network monitoring repository: http://repository.uslhcnet.org/

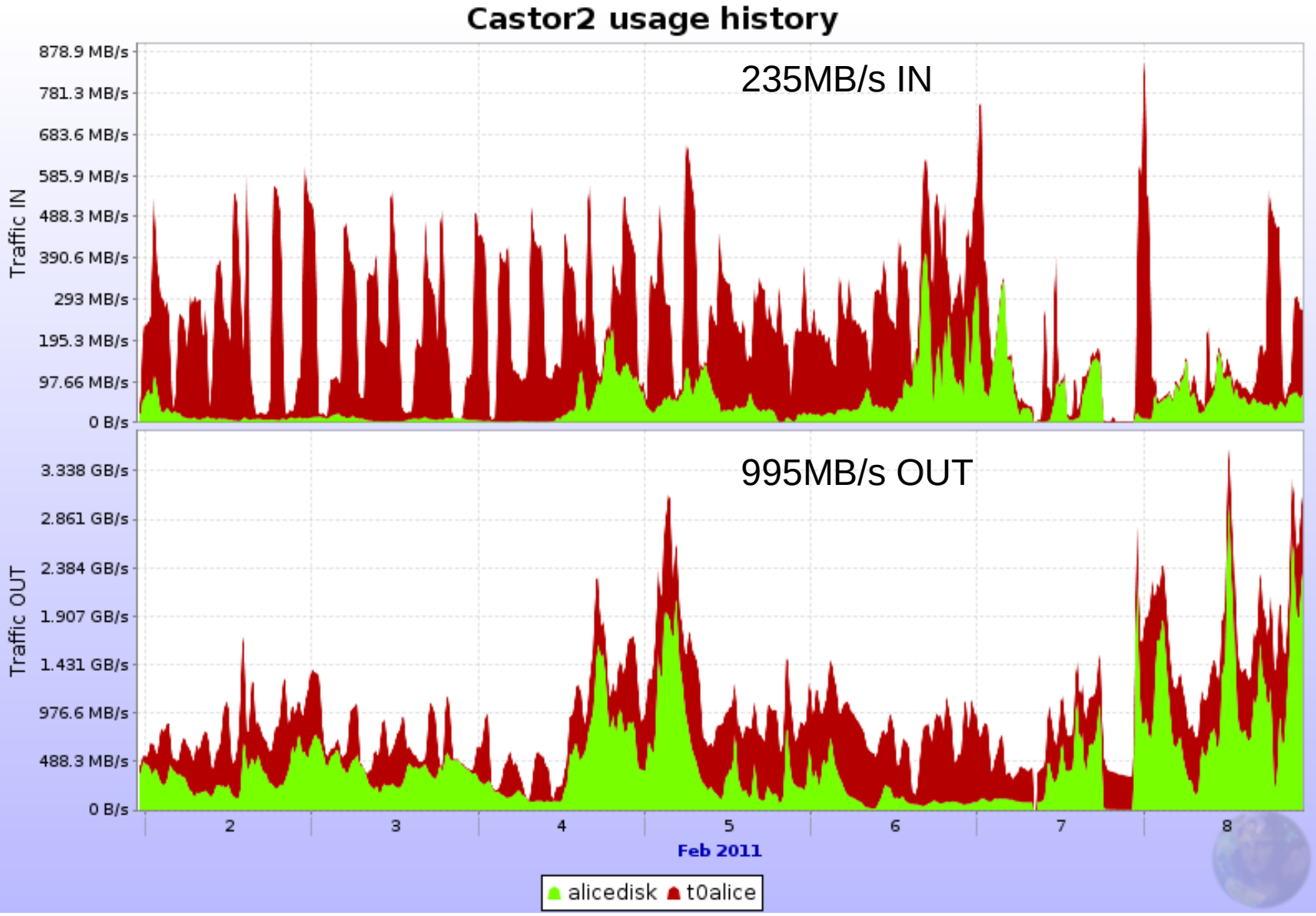
Last week activities



Xrootd servers monitoring



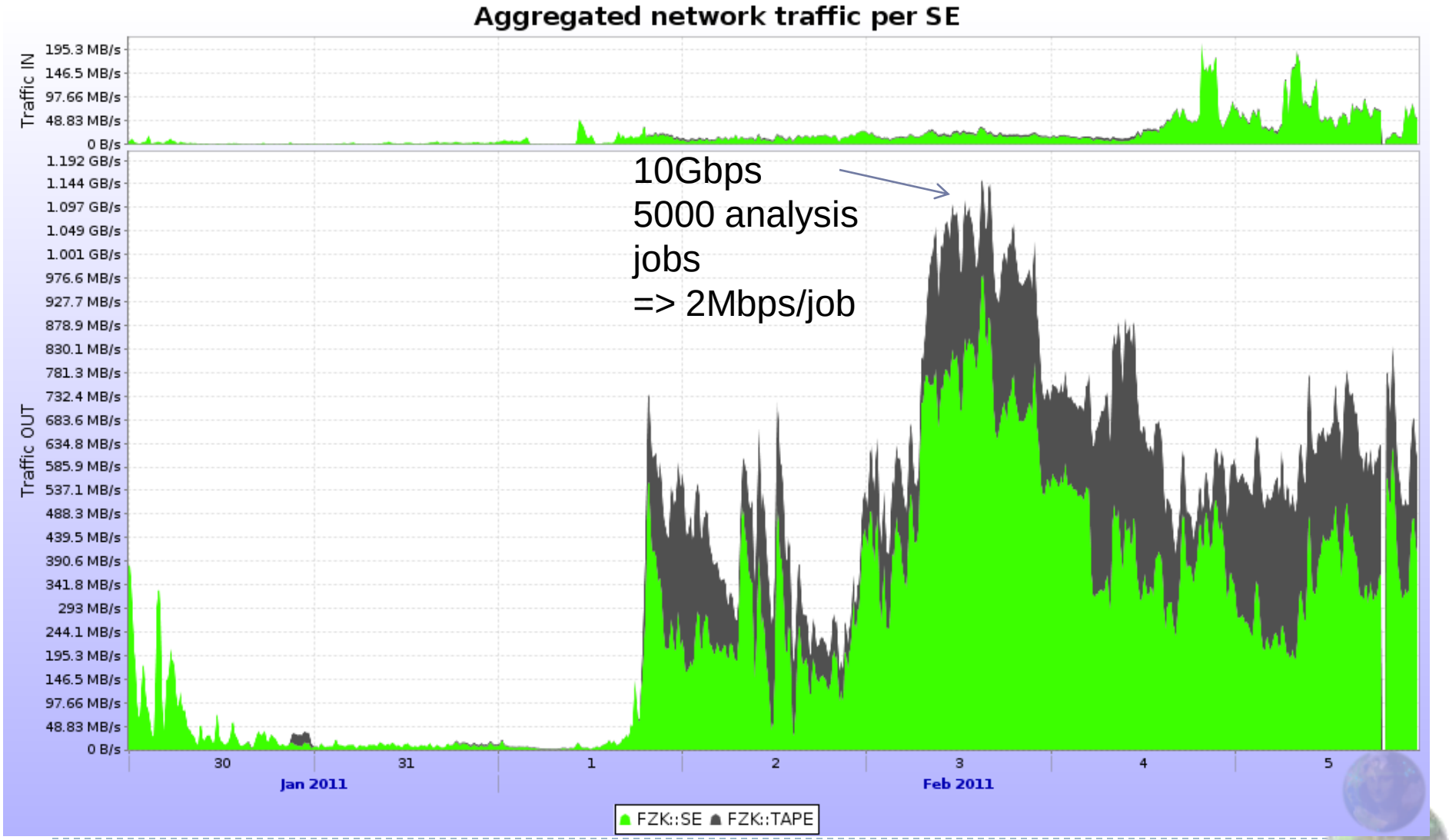
CERN Castor2 servers



Average job I/O activity over 1w

- 17025 average running jobs
- Xrootd: 1.66GB/s out, 622MB/s in
- Castor2: 995MB/s out, 235MB/s in
- So each job (worker node core) consumed about
  - 1.25 Mbps in
  - 0.4 Mbps out
- Not taking into account DPM and dCache storages
- With spikes of up to 3x this
- The jobs could process more if it would be available

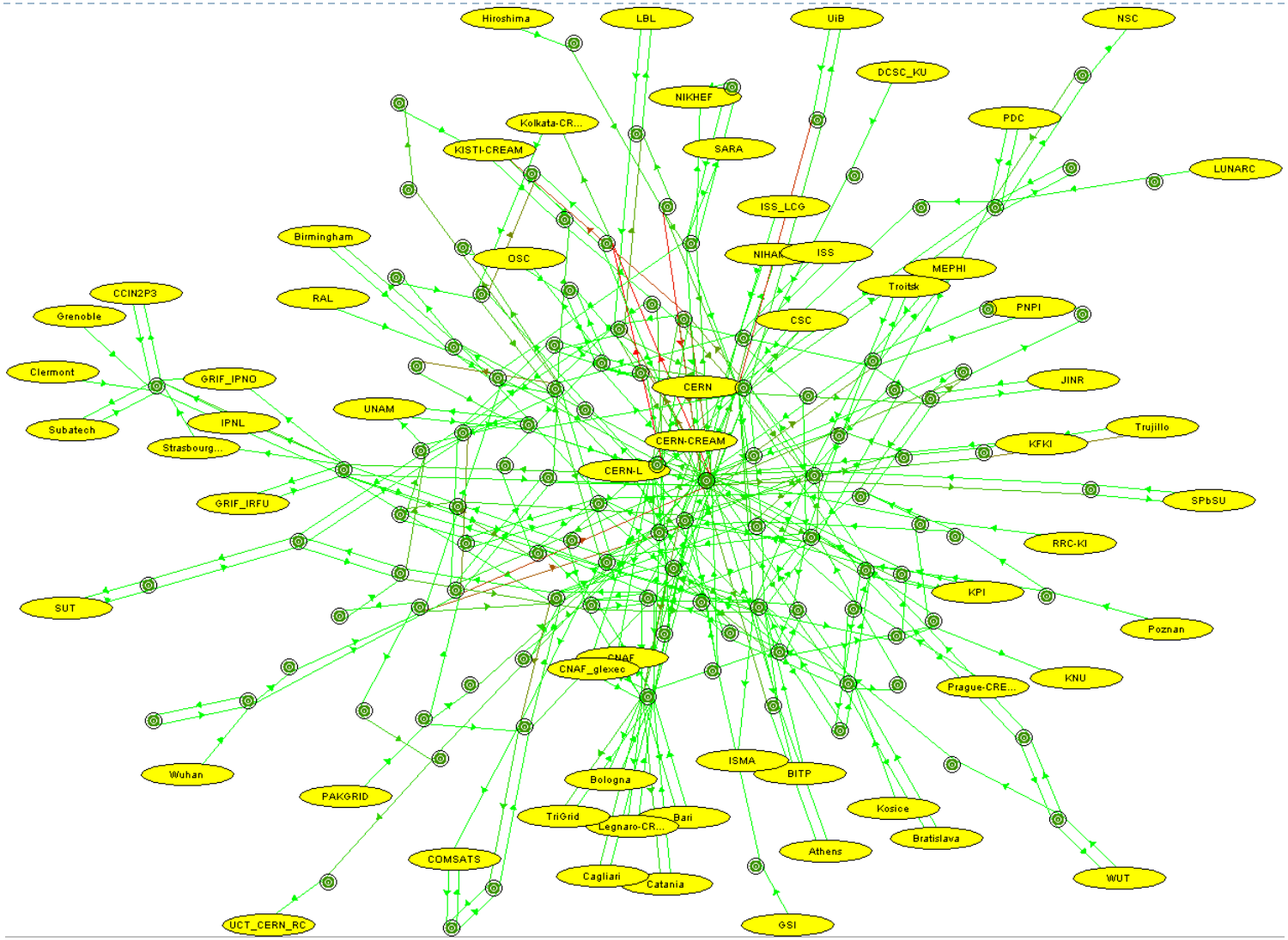
A T1 example


Aggregated network traffic per SE
Traffic IN
195.3 MB/s
146.5 MB/s
97.66 MB/s
48.83 MB/s
0 B/s
Traffic OUT
1.192 GB/s
1.144 GB/s
1.097 GB/s
1.049 GB/s
1.001 GB/s
976.6 MB/s
927.7 MB/s
878.9 MB/s
830.1 MB/s
781.3 MB/s
732.4 MB/s
683.6 MB/s
634.8 MB/s
585.9 MB/s
537.1 MB/s
488.3 MB/s
439.5 MB/s
390.6 MB/s
341.8 MB/s
293 MB/s
244.1 MB/s
195.3 MB/s
146.5 MB/s
97.66 MB/s
48.83 MB/s
0 B/s
10Gbps
5000 analysis
jobs
=> 2Mbps/job
30
31
1
2
3
4
5
Jan 2011
Feb 2011
FZK::SE
FZK::TAPE

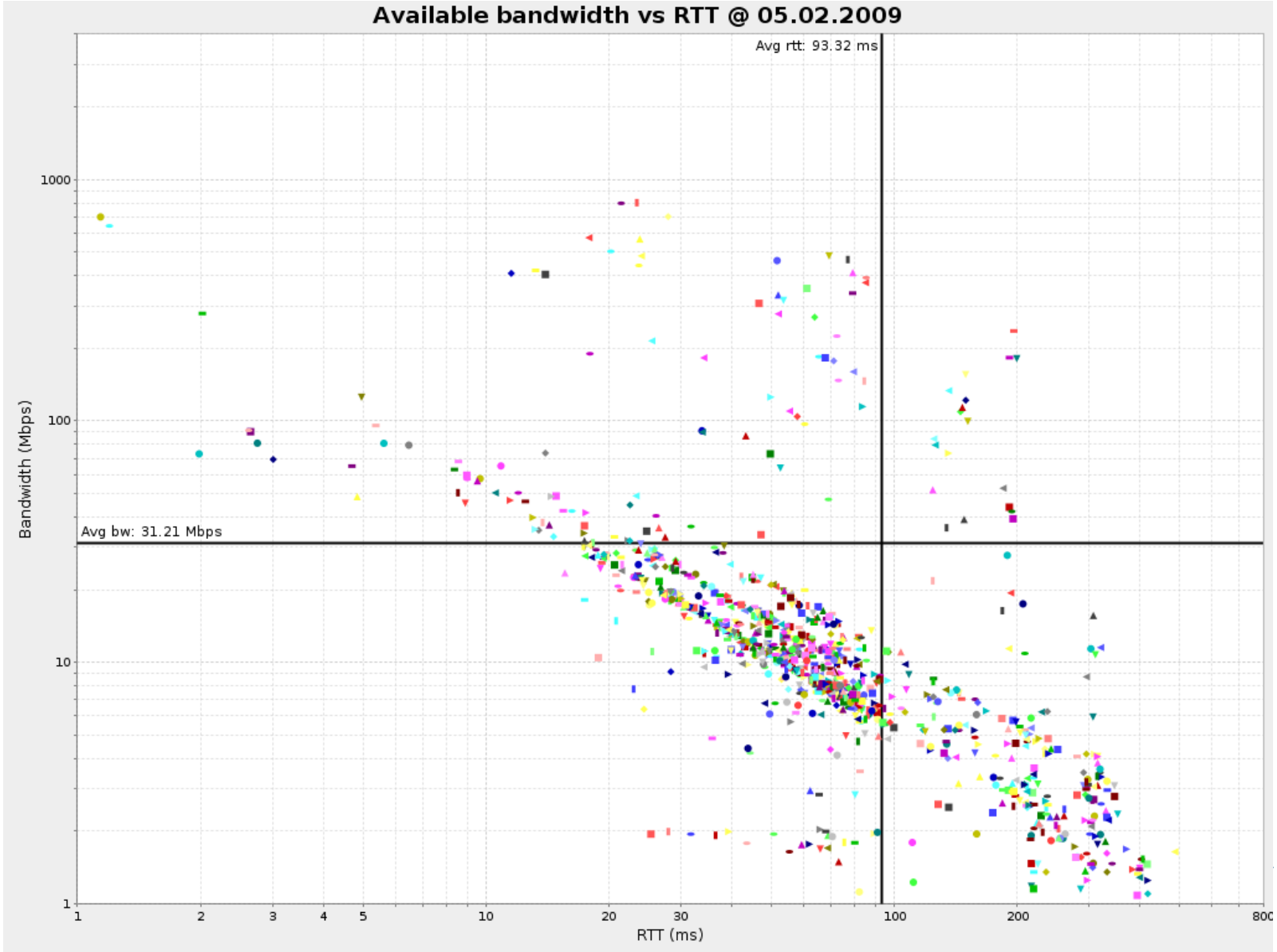
Network topology discovery

- Site MonALISA services continuously perform traceroute between them and publish the results
- Central repository coordinates bandwidth tests between VoBoxes
- With each test the traceroute and machines' configuration is stored
- Together with other monitoring information (storage element functional test results, occupancy) a "closest SE" metric is computed for any client IP address
  - Jobs read/write from/to the closest working storage
  - Clients benefit from the same features

Network topology (AS level)



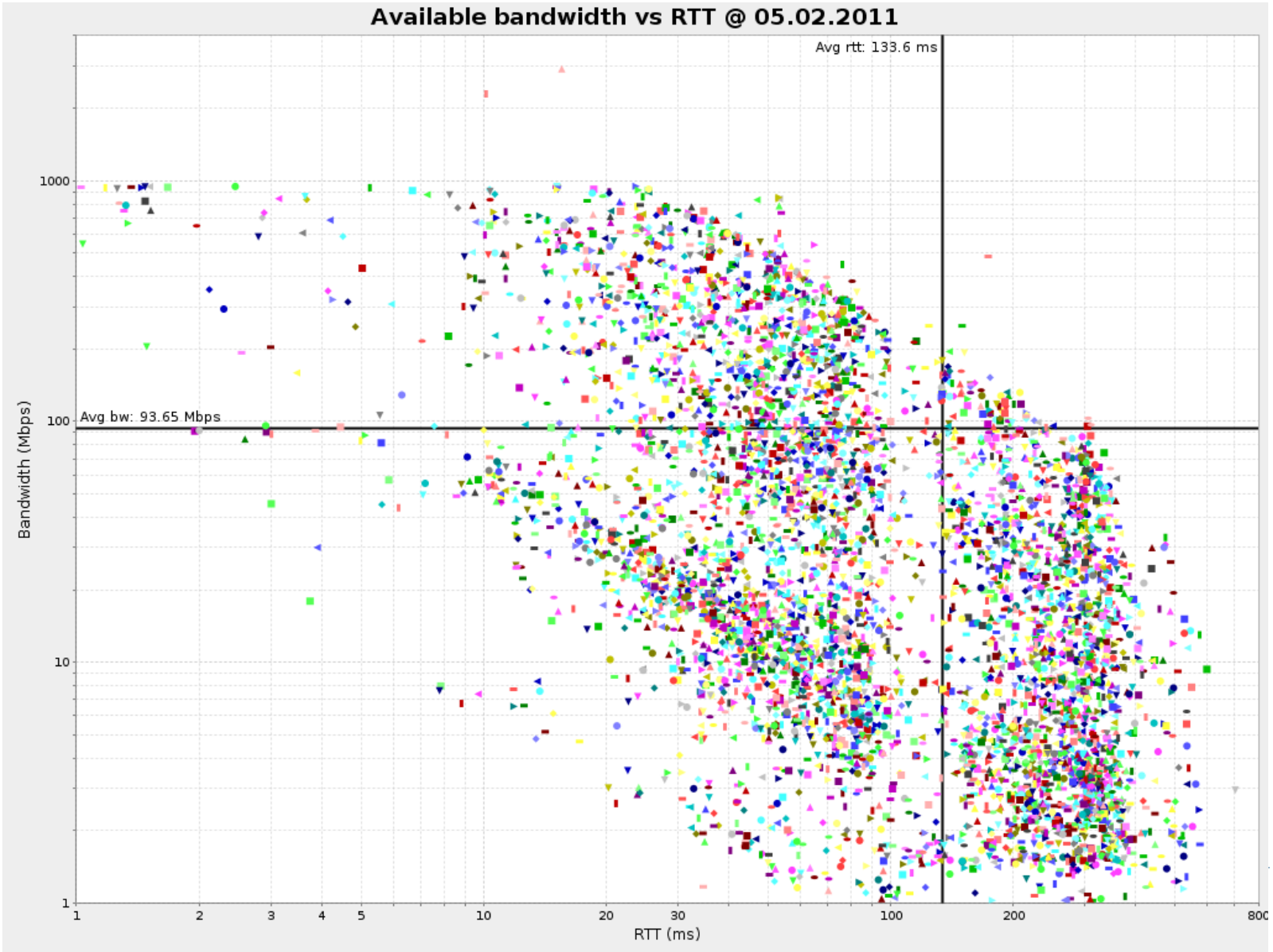
Available bw and buffer sizes



SLC4
1MB
default
buffer size

Avg:
30 Mbps
93ms

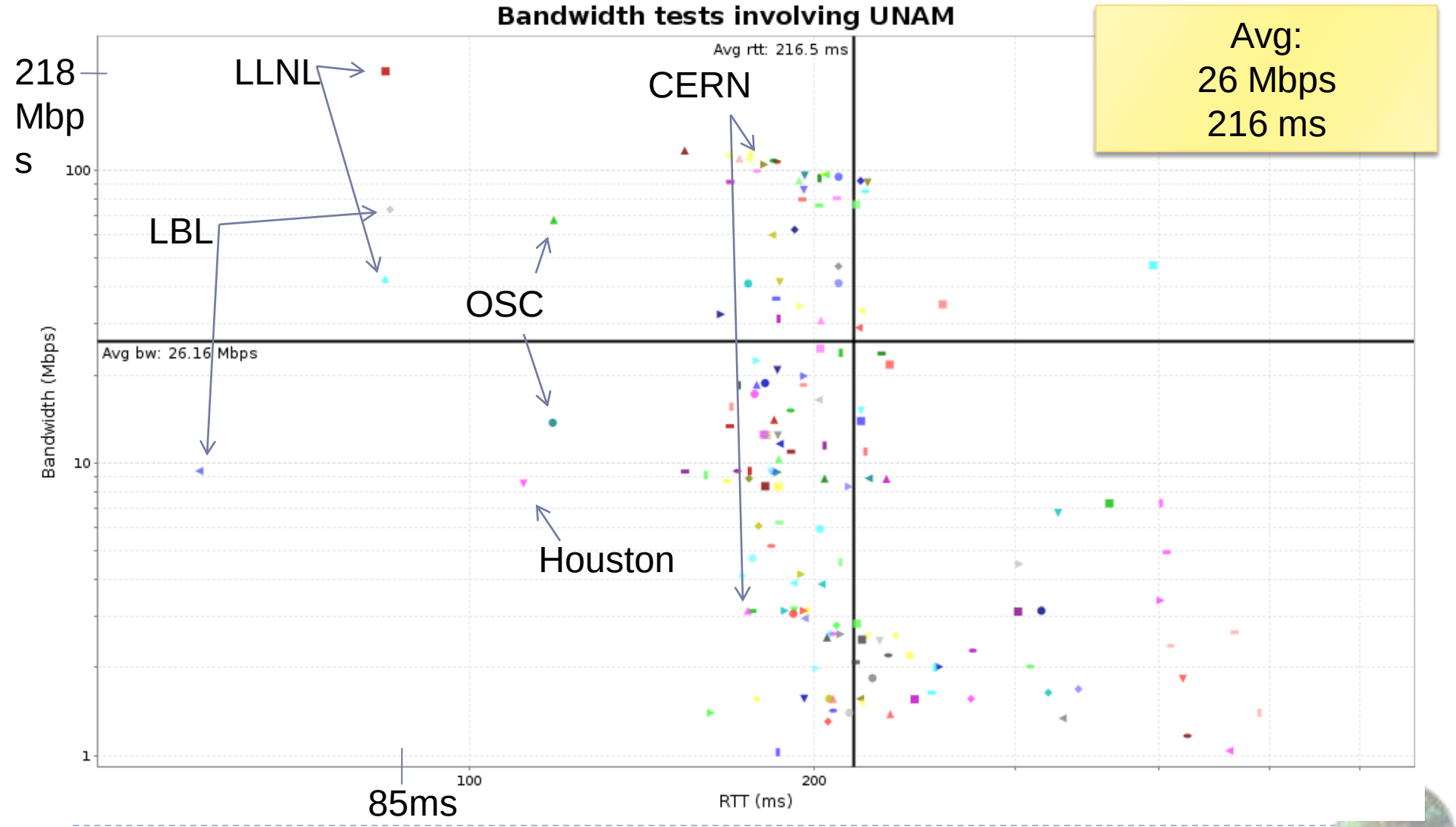
Available bw and buffer sizes



SLC5
4x
increase
of default
buffer size

Avg:
93 Mbps
133ms

UNAM connectivity with the Grid



Firewall requirements (VoBoxes)

‹UNAM›

Chart view »

IN from

No.	ID	Site	Speed (Mbps)	Hops	RTT (ms)	Streams
1.	737060	SARA	116.92	14	154.31	1
2.	735359	LBL	81.84			1
3.	731451	CyberSar	75.89	18	202.25	1
4.	733253	GSI-CREAM	69.44			1
5.	735432	OSC	67.69	19	118.60	1
6.	736884	Bratislava	60.17	17	183.89	1
7.	734816	IHEP	56.72			1
8.	735079	Catania	47.01	19	210.14	1

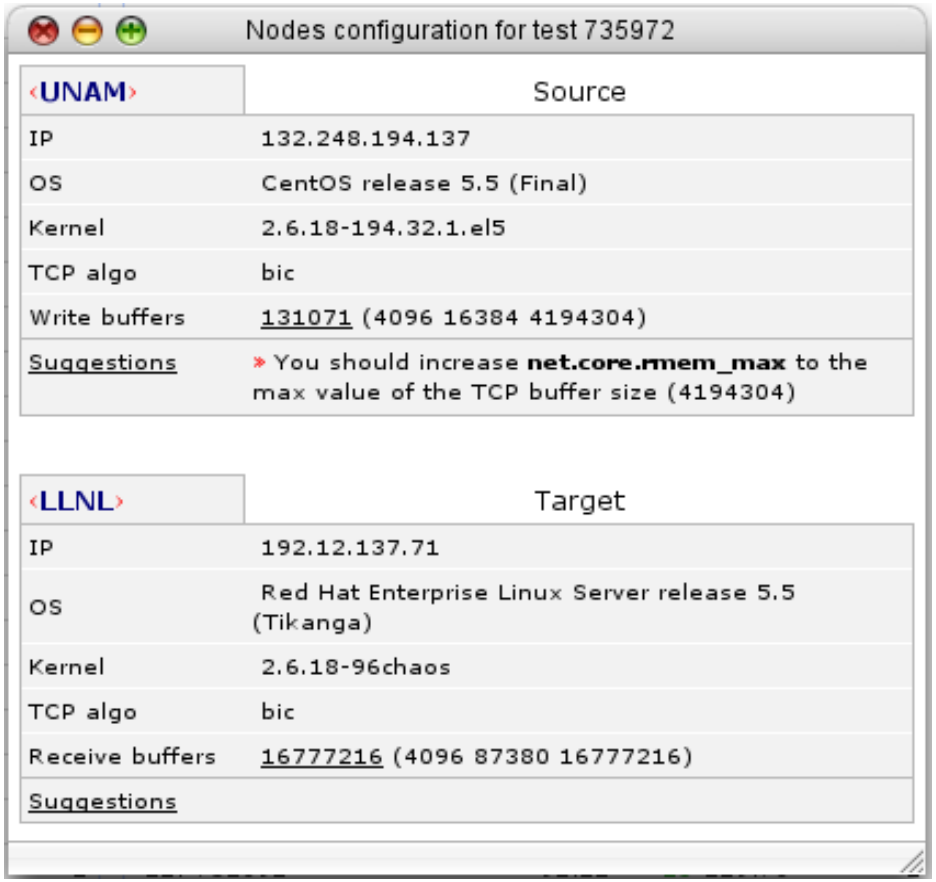
OUT to

No.	ID	Site	Speed (Mbps)	Hops	RTT (ms)	Streams
1.	735972	LLNL	218.12	14	84.58	1
2.	736713	SARA	122.82			1
3.	732975	CSC	114.56	13	176.23	1
4.	733107	HHLR_GU	112.82	17	168.94	1
5.	734790	NSC	109.86	16	172.24	1
6.	735782	Subatech	107.98	20	184.50	1
7.	737437	CNAF_glexec	107.05	17	185.71	1
8.	734109	GRIF_IPNO	106.78			1

- All columns on this page should be filled for your site
  - http://alimonitor.cern.ch/speed/
- Incoming and outgoing
  - TCP/1093 – bandwidth estimation
  - ICMP
  - UDP/33434..33534 (traceroute/tracepath)

Machines' configuration

- On the web interface the configuration for each test is displayed, along with tuning suggestions



Machines' configuration

- Considering the large RTT the following values should be applied at UNAM too:
  - net.core.rmem_max = 8388608
  - net.core.wmem_max = 8388608
  - net.ipv4.tcp_rmem = 4096 87380 8388608
  - net.ipv4.tcp_wmem = 4096 65536 8388608
  - net.core.netdev_max_backlog = 250000
- All nodes should have this configuration
  - VoBox, storage servers, worker nodes …
- More information here:
  - http://monalisa.cern.ch/FDT/documentation_syssettings.html

Thank you for your attention!

- http://alimonitor.cern.ch/speed/videos
  - How the available bandwidth and RTT evolved in time