

Grid Mon WG Meeting
25 January 2007
14:30

Date and Location -----

25 January 2007
14:00 - 17:30
It Aud

Attendees -----

Sonvane, Sergio, German, Ian N, Steve, Anthony, Emir Imamagic (SRCE), Julia, Dobrisa Dobrenic (SRCE),
Alessandra, Piotr

Subject -----

Gridview - Sonvane

Current gets data direct from nodes, plans to get it from fabric monitoring system at some point.

Also push back the data to the site current via exposing a HTTL/XML interface that the site pulls from.

Does summarization internally of the raw data
Currently only programmatic interface for Service Availability, will come for some of the other data including summarized data

And Visualization (see slides)

Service Availability - focused on Tier-1s. Can drill down into SAM results to see what is causing the availability. Can see everything that is produced by SAM, and the explanation

Also viewed via SAM Portal view, and XSQL view that exports as HTML/XLS. Plan is to merge all three views into Gridview

Also Gridftp log monitoring and job monitoring.

Ian: where does the information come from?

Sonvane: L&B for the RB.

Julia: where do you get it from?

Sonvane: R-GMA, but much information is missing.

Sonvane: Plan is to parse plain text log file on L&B and publish via the WS client

James: What does the RTM do - db?

Anthony: Yes

Emir: It'd be better if all used the same.

James: this is the idea, but remember we can have various higher level systems consuming the data.

Julia: ^ See differences in existence of job recording in different systems - RTM vs gridview.^

Julia: VOs are move and more submitting via Condor-G even on WCLG sites.

Emir: ^ Is there sensors on CE that connects info from batch system. APEL?

Julia: ^ It can tell you the snapshot state of the CE, but not on individual jobs.^
GridICE is an option here, but it's not running everywhere.

Emir: ^ Do you see differences between accounting DB and the job monitoring data.

Sonvane: ^ Not sure about the accuracy of data.

Emir: ^ You can see how many are not using the RBs- since accounting data is more.

Emir: ^ How do we know if our RB is publishing.

Sonvane: If it's publishing to R-GMA, then it will be there.

Future: ^ Plan is to visualize FTS statistics as well.

Obs: ^ Not all data is readily available as a metric, and need to process lots of log files.

Gridview has specified schema optimized for the type of data that is being processed.

Every info (aggregate/summary metric) needs to be visualized in a specific way depending on what it represents, and also summarization is specific for each type.

Steve: ^ So you do the summarization in real-time or regularly.

Sonvane: Regularly.

Steve: Can you download anything via the interface

Sonvane: ^ Yes.

Steve: you may want to be limiting this!

LEMON/SLS ^ - German

Front page similar to ganglia (deliberately !)

Emir: ^ If similar to ganglia, why didn't you use it?

German: ^ Ganglia used IP multicast (one reason).^ Also more flexible for adding metrics.^ Also, permanent repository for data, not just RRD graphs.

Emir: ^ Have you thought about using ganglia as backend - don't want to force people to use another fabric system.

James: ^ Definitely.

German: ^ LEMON has performance and exception monitoring.

Julia: ^ Do people look three years back?

German: no idea !

Alessandra: ^ You can write bids on it.

Emir: You can data mine it to see how things are changing

Julia: ^ but that's a lot of data to keep around.

German: It's only 2G/day at CERN.

German: ^ Some graphs are generic, some are very specific to given metrics e.g. reboot time by node.

SLS: We don't generate any data, only display it.

Julia: how do you calculate the weight for service when calculative metrics.

German: The user/VO will tell us.

Has configurable threshold per service between 'fully available', 'affected', 'degraded' and 'not available'.

James: Who is the target audience?

German: End users and Managers.

Information to be provided by MW developers - German

Steve: There are now some recommendations for JRA1 on log file formats.

Services should publish availability numbers

Ian N: Why do you think it's better if the service calculates its own availability, rather than a calculation on exposed metrics.

German: you know best how to do it, and it doesn't put load on the fabric system (or the need for a correlation/calculation engine)

NOTE: availability in this context is a point measurement, not the integral over time (as in SAM)

James: I think maybe this is only needed for availability going out to the grid, so could be in that layer.

James: Often we use log file data when the service could provide it easier e.g. LFC num. transactions per min.

Coffee

Three teams to work (with initial participants)

- Sensor description questionnaire (Ian N, Luigi, German,)
- Internal Sensor interface to Nagios/Iemon (James, Emir, German)
- Definition of format for publication of external metrics back to sites (Piotr, Sonvane, ...)

GridICE + sensors (Sergio)

Generic 'CheckDaemon' sensor which gathers a bunch of metrics.

Provide pre-defined daemon sets for nodes in gLite release

But not done before the release

James: Do you have to do it after the release.

Sergio: We get the code in the INFN release team before the release, so we can check.

Julia: Is the sensor a simple script for the CE?

Sergio: Not quite a simple script - two daemons that parse logfiles and put them in a cache.

Emir: Did you coordinate with the accounting people?

Sergio: Yes, with DGAS. But there are different needs - e.g. accounting can't lose any data. Also we're accounting the job while it's running or queued, not looking afterwards. The data it archived and published via the extended GRIS.

Emir: No problem with daemons, as long as it's the only thing getting the data.Â
Currently there are many sensors getting the same info on the CE on running jobs.

Anthony: For CE, need to consider interactions with GLUE2.0 publisher which also will query the LRMS.

German:Â Overlap between sensor sets in LEMON and GridICE.

Sergio:Â Historically, we wanted different one in a defined range, and in the new version of GridICE/LEMON, we'll move to the supplied sensors for LEMON.

Alessandra: What about the repository.

James:Â If you'd like to run it, that would be great.

Alessandra: Ok - it'll be in the UK, protected by certificates, using SVN.Â Is that ok?

James: Don't see any problem if you'll maintain it.

Nagios in CEE Region - Emir

Nagios doesn't need to run on the fabric, it can probe remotely.

A sensor checks one host, or one service.Â It returns a state.

There's a hierarchy:Â host, then host-ping, then service functionality.

Router, then host, then lightweight parent service (every 15min), then child services (heavyweight, every 1hr)

Return from a probe is State (OK, Warning, Unknown, Critical) and a description

States are similar to levels in SMS.

If NRPE is on host, that's active - scheduled by Nagios.

NCSA - it's passive, since it schedules, and sends to Nagios.

Not worth doing performance via Nagios - use ganglia instead.Â Nagios is really for Service Availability/Health!

Used in CEE Region.Â Single Nagios server. Nothing deployed on sites.

Use gatekeeper as 'passive probe'.. No automatic recovery, since we have no agents running.

Use GOCDB to generate Nagios config.Â They have tools to do this.

Currently data is stored in files.Â There is a plan to move to a DB (some extensions exist that can use MySQL - under investigation)

Have integrated GOCDB scheduled downtimes into Nagios as Nagios downtimes.

All sensors run as dteam.Â Can do it for other VOs, but need the information!

Added certificate-based Authentication + Authorization.

Can get the router, and then use it to isolate network problems.

Sonvane:Â Is there a programmatic interface for Nagios?

Emir: No, but it's possible.

For local sensors, we use basic Nagios sensors, and then locally run the global sensors. But if you run it locally, you can automatic recovery (since you have an agent there)

LEMON action. You have a **probe**, which is monitoring system agnostic, which is called by a **sensor**, with options to provide back the right bits. Then one sensor which can call a given service.

James: Seems quite similar to SAM, and the output they have.

German: Need a very specific impl, so there is a 'single' Nagios sensor, and a 'single' LEMON sensor.

Anthony: Might be hard to do -e.g. levels for warning.

James: Well, we just need a single place for the config - e.g. in Nagios, and passed to the script sensor.

Anthony: MW Example: For R-GMA, there's a HTTPS interface for raw data on everything going on the MON box - this could be integrated. Also a ping. And there's some interpretation done as well, in the GUI page.

LEMON: Operator screen as well. As can put priorities on alarms - but it's ++ on the alarm, i.e. it's very critical alarm...

German: Decide what information we need in the return - and then look at the

Anthony: does the sensor have to interpret the data.

James: It needs to 'understand' the service i.e. error or not. If it doesn't fit into a exception, return it as a performance metric.

Dobriza: Should we go for standardizing on the error messages.

James: Maybe an error scale 0-100, and services do a mapping

German: SLS does both.

James: Ok, then we should have a standard set, and a scale, if possible.

Discussion

AIMS: Subset of people to start writing a 'Sensor Description Questionnaire'.

AIMS: Subset to look at integrating remote probe data into fabric (what data, and what action)