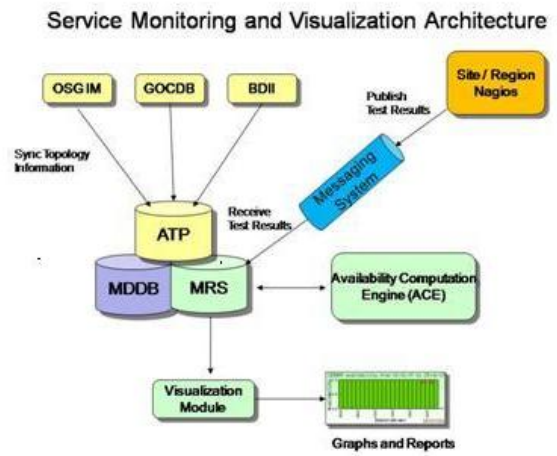
Table of Contents

Availability Computation Engine (ACE)

The main thrust in building the Availability Computation Engine (ACE) is providing flexibility at various levels and allowing definition of multiple algorithms for computing the Service Availability of Grid Resources. The engine design should be modular so that it facilitates future expansions with easy plug-ins.

Monitoring Architecture :



The new monitoring framework consists of following subsystems :

Aggregated Topology Provider (ATP) : ATP is a repository of grid topology containing information about projects (WLCG), grid infrastructures (EGEE, OSG, NDGF), Sites, Services, VOs and their groupings, site / service downtimes and its history. It synchronizes topology information from OSG IM, GOCDB and BDII.

Metric Description Database (MDDB) : Project-level component that contains meta-data about metrics and how those metrics can be combined to calculate different availabilities.

Metric Result Store (MRS) : A Repository of metric results containing Metric results for service end-points for the grid infrastructure.

Availability Computation Engine (ACE) : The Availability Computation Engine (ACE) module computes availability of Grid Resources using Test Results form MRS, topology data from ATP and profile definition from MDDB. It supports multiple algorithms for computation of Grid Resources.

Metric Status Definition :

In Metric Store we have 4 status definitions

- OK
- WARNING
- CRITICAL
- UNKNOWN

These status values are identical to those returned by Nagios probes.

CRITICAL is defined as metric is in ERROR state .

UNKNOWN: Here, UNKNOWN is not the same as missing test results as the case with Gridview. Nagios explicitly returns UNKNOWN as metric value in certain cases. e.g. if it tries to run lcg-cp to test a SE, but the command does exist on the Nagios machine, you d get UNKNOWN. Nagios sets the state to unknown if, e.g. a Timeout happens

Validity of Metric Results :

There may be many reasons for the non-arrival of test results, like outage of message bus or outage of nagios instance or some other problem. How long is the earlier metric result valid if a fresh result does not arrive for a given service?

Generally a VO would like to have different validity for different metrics depending on the cost of executing the metric and frequency at which the metric is scheduled. Certain inexpensive Metrics may be scheduled very often whereas some expensive (consume more resource, take more time) Metrics may be scheduled with less frequency, so they need to have different validity.

If validity of any of the metrics for a service in a profile is expired, the status of the service will be set to MISSING (to distinguish it from UNKNOWN).

Service Status Definition :

In Metric Store we have 5 status definitions for a service:

- OK
- WARNING
- CRITICAL
- UNKNOWN
- MISSING

The first 4 service status is same as metric status. The status of a service for a profile is set to MISSING only when validity of earlier metric result (including UNKNOWN) for a profile is expired.

The status of the service should indicate the actual state as per the results even if the service is in a scheduled downtime (scheduled down).

While visualizing service status, MyOSG displays both service status (from ACE) and downtime (from ATP) for the service if any.

Computation of Service Status

Metric Results to Service Status

i) Defining Metrics for a service flavour in a profile :

- Level of Flexibility:
  - Basic level Flexibility: ANDing of a set of metrics. (Current level)
  - Intermediate level of Flexibility : Two level Expression : OR of ANDs / AND of ORs
    Eg. OR of ANDs
    (Metric_A AND Metric_B AND Metric_C) OR (Metric_D AND Metric_E AND Metric_F) OR (Metric_A AND Metric_B AND Metric_C)
    Eg. AND of ORs
    (Metric_A OR Metric_B OR Metric_C) AND (Metric_D OR Metric_E OR Metric_F) AND (Metric_A OR Metric_B OR Metric_C)

- High level of Flexibility: Any Boolean expression with only 2 operators AND and OR.

By Metric_A here I mean a particular metric test run by a particular VO (as multiple VOs are running tests) A profile can define metrics for multiple service flavours.

ii) Defining Resource Domain for a profile :

For Resource definitions we may have multiple ATP groups to define a resource domain for a profile. Say a group of sites, then also a group of services, etc. Service status should be computed for all the services in the resource domain that support the profile VO. A profile may be associated with one or more such groups. The resource for the profile is the union of services belonging to all the groups associated with the profile. The groups may be created in ATP. The groups can be associated with the profile in MDDB. .

Computation of Aggregate Status for Service Groups

Service Group may be Standard (Site) or Custom (user defined). Custom Service Group may be created by creating a group of services. How to aggregate statuses of the services within a service group ?

1. Define a service sub-group
2. Define the aggregation operation on it

Definition of a service sub-group : A service sub-group can be defined implicitly as below :

1. All services of same flavour
2. Service of flavour x
3. All services of same type
4. Services of type x

Case (c) above, All services of same type would resemble our current aggregation for service type like CE or SE for a site.

Defination of the Aggregation Operation:

It can be one of the below

1. AND
2. OR
3. at least N (at least 1 is same as OR)
4. at least x %.

A service Group is a maximum level of status aggregation and status values may not be aggregated beyond this point in the hierarchy.

Computation of Availability / Reliability of a Service / Service Group

The Availability / Reliability metrics for a service / service Group or service sub-groups over a given period of time can be computed from the corresponding service status and scheduled downtime over the period.

Interval Definition: From and To Timestamps can be provided to define limited intervals over which availability/ reliability metrics should be computed. Infinite intervals can also be defined to always compute them.

Frequency Definition: The periodicities/frequency over which the Availability / Reliability Metrics are to be computed can be selected (multiple values can be selected) from below:

1. Hourly
2. Daily
3. Weekly
4. Monthly

.

Computation of Aggregate Availability / Reliability Metrics across Service Groups

A service Group is a maximum level of status aggregation and status values may not be aggregated beyond this point in the hierarchy. However Availability / Reliability metrics may be aggregated across Service Groups.

Aggregation Functions: Aggregation function can be one of

1. Max
2. Min
3. Average
4. Weighted Average
5. Median

This is like computing aggregate availabilities for Region / Federation / NGI / All Tier-1/0 sites

Definition of Availability Algorithm

The algorithm definition (configuration) should be kept in MDDB separate from the profile. An algorithm acts only on a single profile, not across profiles, that is an algorithm can use services statuses from a single profile, not from multiple profiles. On the other hand, the same algorithm may be used for computing Availabilities for multiple profiles. So aggregate statuses, availability / reliability metrics are computed for a given profile using a given Algorithm.

Resource Domain for an Algorithm :

How to define a resource domain for a profile is discussed earlier. Can we define resource domain for an algorithm in the same way ? In principle the domain for the algorithm can be any subset of the domain for the profile, shall we support defining such domain for the Algorithm ?

The algorithm should always act on the resource domain for the profile. Defining a separate domain for the Algorithm will not be supported.

Interface / Visualization of the Algorithm :

- Interface for user to visually define resource groups
- Visually define the algorithm, by using Resource Groups, Operators etc.
- One should be able to visualize / drill down the algorithm pictorially e.g. Tree format
- Also the same way user should be able to drill down the results.

We would have diagram showing the flow of information and the various steps to be performed, along with the inputs/outputs and responsible components. Interface to define resource groups may not be required as we

are using only standard resource groups and not supporting custom resource groups. We should have Interface to define the algorithm. Availability data will be visualized using MyEGI portal.

Recalculation of Service/Site Status and Availability/Reliability Metrics :

Re-calculation of Service/Site Status and Availability/Reliability metrics may be required due to various reasons like late arriving results, change in topology, SAM outages etc. ACE should be able to do the re-calculation on the fly. This topic will be handled separately after the implementation of the first version. We should think of various re-calculation strategies like the one based on deltas or checkpoints etc.

Requirements from VOs:

Here is a link to the discussions with the LHC VOs about their requirements for ACE.

- PhoolChand - 22-Jun-2010

This topic: LCG > ACE
Topic revision: r3 - 2010-06-23 - unknown