Computation of Service Availability Metrics in Gridview

Digamber Sonvane, Rajesh Kalmady, Phool Chand, Kislay Bhatt, Kumar Vaibhav

Computer Division,

BARC, India

Computation of Availability Metrics

- Gridview computes Service Availability Metrics per VO using SAM test results
- Computed Metrics include
  - Service Status
  - Availability
  - Reliability
- Above Metrics are computed
  - per Service Instance
  - per Service (eg. CE) for a site
  - per Site
  - Aggregate of all Tier-1/0 sites
  - Central Services
  - over various periodicities like Hourly, Daily, Weekly and Monthly

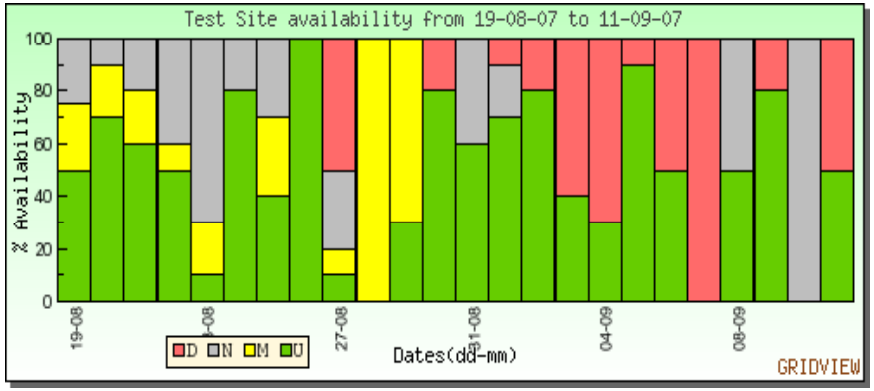
Metric Definitions

- Status of a service instance, service or a site is the state at a given point in time
  - For eg. UP, DOWN, SCHEDULED DOWN or UNKNOWN
- Availability of a service instance, service or a site over a given period is defined as the fraction of time the same was UP during the given period
- Reliability of a service instance, service or a site over a given period is defined as the fraction of time the same was UP (Availability), divided by the scheduled availability (1 – Scheduled Downtime – Unknown Interval) for that period
  - Reliability = Availability / (1 - Scheduled Downtime – Unknown)
  - Reliability = Availability / (Availability + Unscheduled Downtime)
- Reliability definition approved in LCG MB Meeting,13th Feb, 2007

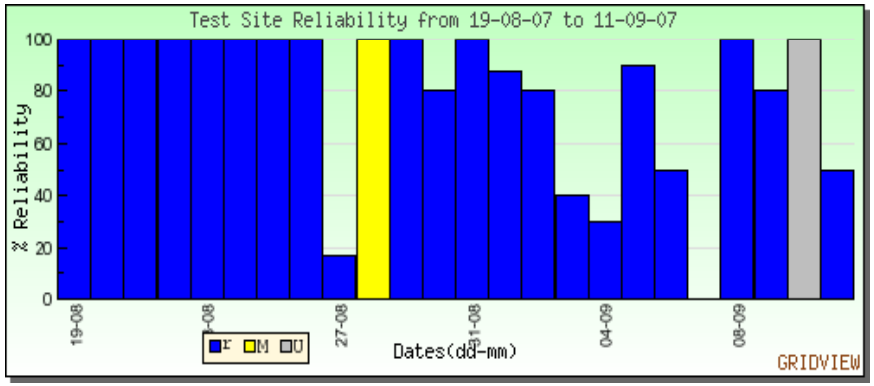
Metric Computation Example

- Consider Status for a day as
  - UP – 12 Hrs
  - Scheduled Down – 6 Hrs
  - Unknown – 6 Hrs
- Availability Graphs (1st bar in Graph) would show
  - Availability (Green) – 50 %
  - Sch. Down (Yellow) – 25 %
  - Unknown (Grey) – 25 %
- Reliability Graph (1st bar in Graph) would show 100%
- Reliability = Availability(Green) / (Availability(Green)+ Unscheduled Downtime(Red))
- Reliability not affected by Scheduled Downtime or Unknown Interval

Sample Availability Graph



Sample Reliability Graph



New Algorithm

- Old (Current) algorithm
  - Computes Service Status on Discrete Time Scale with precision of an hour
  - Test results sampled at the end of each hour
  - Service status is based only on latest results for an hour
  - Availability for an hour is always 0 or 1
- Drawbacks of Discrete Time Scale
  - Test may pass or fail several times in an hour
  - not possible to represent several events in an hour
  - loss of precise information about the exact time of occurrence of the event
- New Algorithm
  - Computes Service Status on Continuous Time Scale
  - Service Status is based on all test results
  - Computes Availability metrics with higher accuracy
  - Conforms to Recommendation 42 of EGEE-II Review Report about introducing measures of robustness and reliability
  - Computes reliability metrics as approved by LCG MB, 13 Feb 2007

Major Differences

- Major differences between old (current) and new algorithm
  - Service Status computation on Continuous time scale
  - Consideration of Scheduled Downtime (SD)
    - Service may pass tests even when SD
    - Leads to Inaccurate Reliability value
    - New algorithm ignores test results and marks status as SD
  - Validity of Test Results
    - 24 Hours for all tests in old case
    - Defined by VO seperately for each test in New method
    - Invalidate earlier results after scheduled interruption
  - Handling of UNKNOWN status

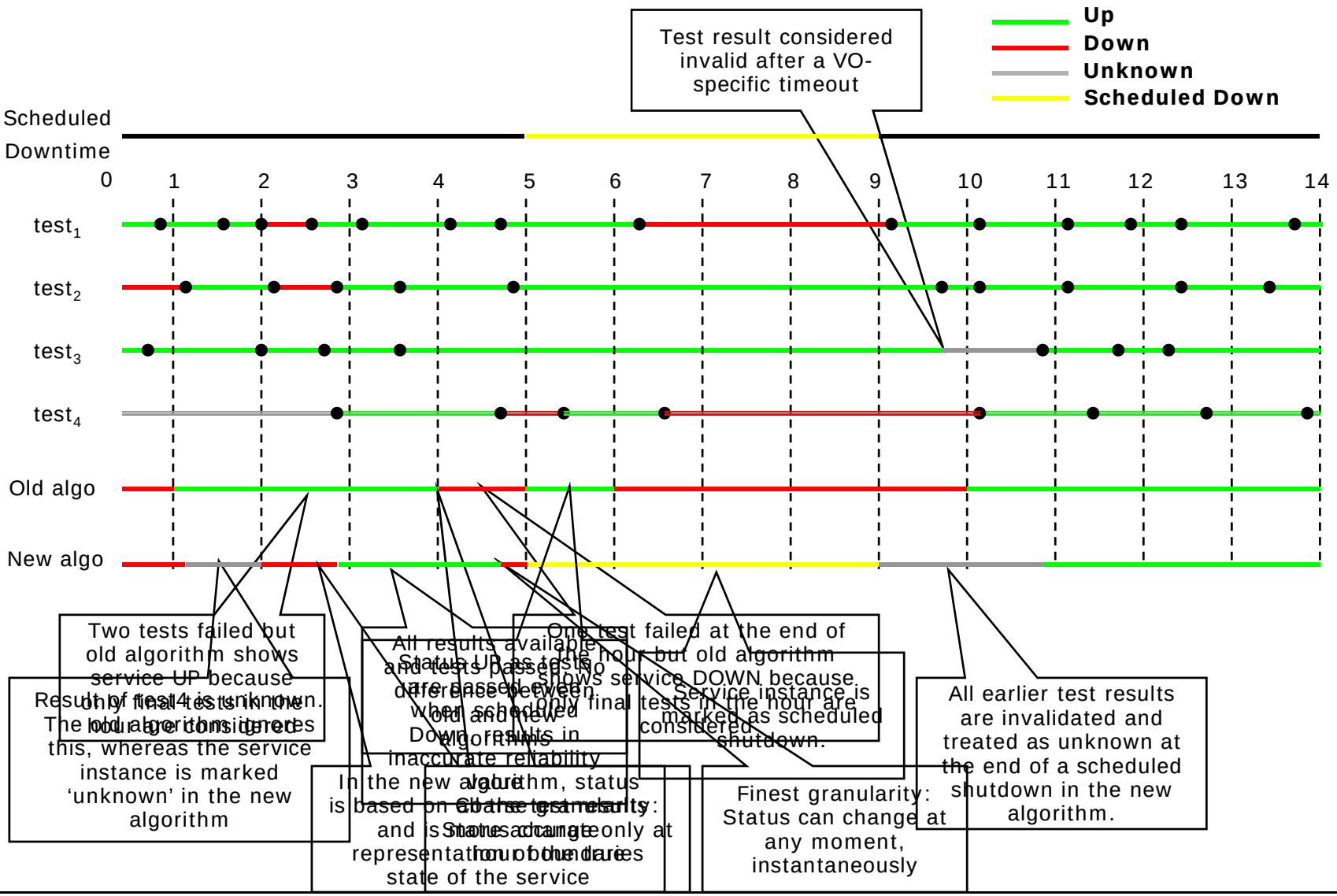
Handling of UNKNOWN Status

- Old (Current) Algorithm
  - Computations are based merely on available parameters
  - Parameters for which values are not known are ignored
- New Algorithm
  - Takes all essential parameters into consideration
  - Status explicitly marked “UNKNOWN” when values of some essential parameters are not known
- Affects
  - Service Instance Availability
  - Site Availability

Computation of Service Instance Status

- VO Marks critical tests for each service
- Service should be considered UP only if all the critical tests are passed
- Old (Current) Algorithm
  - Marks status as UP even if test result of only one of the many critical tests is available and UP
  - Ignores the status of those critical tests whose results are not known
- New Algorithm
  - marks status as UNKNOWN even if the status of all known critical tests is UP and one of the critical tests is UNKNOWN
  - When any of the known results is DOWN, status is marked as DOWN anyway

Computation of service instance status : Difference between old (current) and new algorithm



Discovery of Service Instances

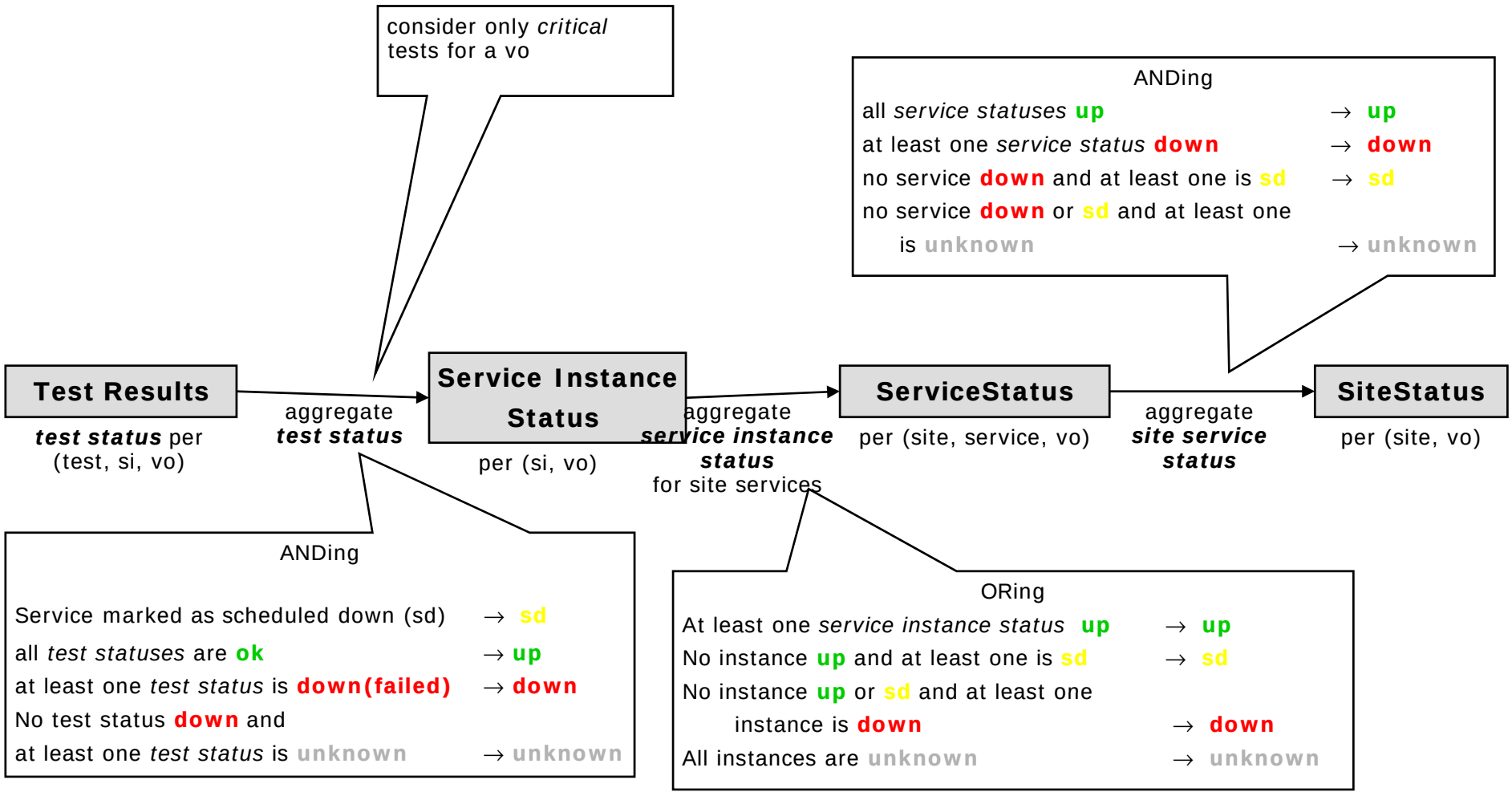
- Old (Current) Algorithm
  - identifies service instances based on available test results
  - computes status of only those services for which at least one critical test is defined and result of at least one critical test is available
  - leads to unexpected behavior like a service instance suddenly disappearing from Gridview's monitoring page
- New Algorithm
  - tracks service instances based on their registration in Information System, either in BDII or GOCDB
  - computes status of all service instances supporting the given VO irrespective of whether critical tests are defined or results available

Computation of Site Status

- Status of a service Eg. CE is computed by ORing statuses of all its instances
- Status of the Site is computed by ANDing statuses of all registered site services
- Status of a site should be UP only if the status of all registered site services is UP
- Old (Current) Algorithm
  - Marks status of site as UP even if status of only one of the many registered services is known and UP
  - Ignores those services for which the status is not known
- New Algorithm
  - Marks status of site as UNKNOWN even if the status of all known services is UP and one of the services is UNKNOWN
  - When any of the known services is DOWN, status is marked as DOWN anyway

Service & Site Service Status Calculation

(new algorithm)



Service = *a service type (e.g. CE, SE, sBDII, ...)*

Serviceinstance (si) = (*service*, *node*) combination

Availability and Reliability Computation

- Hourly Availability, Scheduled Downtime and Unknown Interval for a service instance, service and site are computed from the respective status information
- Hourly Reliability is computed from the Availability, Scheduled Downtime and Unknown Interval for that hour
- Daily, Weekly and Monthly Availability, Scheduled Downtime and Unknown Interval computed by averaging Hourly figures
- Daily, Weekly and Monthly Reliability figures are computed directly from the Availability, Scheduled Downtime and Unknown Interval figures over the corresponding periods

Conclusions

- Old (Current) Algorithm
  - coarse grain representation of service status
  - arrives on a conclusion based on incomplete data
  - can be misleading, giving the impression that everything is ok when the fact is not known
  - may adversely affect actual availability of the service as it might be hiding some of the service breakdowns rather than generating alerts
- New Algorithm
  - resolves ambiguities present in the old (current) algorithm
  - generates true status of the service, sincerely stating it as UNKNOWN whenever it doesn't have adequate data to establish the state of the service
  - Fine Grain Model, Computes service availability metrics with more accuracy
  - Implementation is ready, we are awaiting management nod for deployment

Thank You

Your comments and suggestions please