Service Availability Monitoring (SAM)

Piotr Nyczyk

SAM Introduction

- Monitoring system that probes all service nodes in the grid on regular basis by testing different aspects of functionality
- Tests from user’s perspective with grid credentials
- Not a fabric monitoring! But: very useful for debugging sites
- Main source of information for Grid Operators (COD, ROCs)
- The only source of monitoring information used by FCR tool to exclude/include sites based on VO policy (critical tests, blacklisted resources)
- Additional (but important) source of information for site admins

SAM Architecture I

- Client server architecture
- SAM Server:
  - Oracle database that contains infrastructure description and test results
  - Tools to build and update infrastructure description based on GOC DB and BDII
  - Tools for results processing: calculating availability metric
  - User interface
    - WEB portal providing view of all results (current, history)
    - WEB tool for exporting availability metric data (Excel)
  - Application interface for SAM Client and other tools

- SAM Client (a.k.a. SAM Submission Framework)
  - framework for executing tests from a central UI
  - communicates with SAM Server:
    - get infrastructure description (list of nodes per service type)
    - publish test results
  - two level hierarchy of tests:
    - sensor - usually related to a single service type, contains a number of tests
    - test - single script (executable) that checks selected aspect of functionality and produces a single result record

SAM Architecture III

- Test result record identification:
  - node - host providing the service, linked to site, region, etc.
  - VO - on behalf of which the tests are performed
  - test - selected aspect of functionality being tested
  - timestamp
- Test result record contents:
  - status - one of predefined severity scale codes (OK, WARNING, ERROR ...)
  - summary - short string describing the result, for example: number of channels in FTS, version of middleware installed
  - details - text blob with full log of test procedure for debugging

Types of sensors in SAM

- Integrated with submission framework
  - Test jobs: currently CE and gCE sensor which replaces SFT submission framework
  - Remote probing: SE, SRM, LFC, FTS (!)
- Standalone sensors - existing monitoring tools publishing to SAM DB:
  - RB job monitor - maintained at RAL, UK
  - GStat - site-BDII and toplevel-BDII maintained at Sinica, Taiwan

Integrated sensors - test jobs

- Jobs submitted to all CEs and gLite-CEs listed in GOC DB and/or BDII through a selected RB
- Test job is executed on one (selected by the batch system) WorkerNode at the site
- Performs a number of tests and returns the results to SAM Server
- Delay between submission time and results: about 5 minutes to few hours when site is under heavy load
- Frequency: currently every 2 hours for OPS VO

Integrated sensors - remote probing

- Set of tests executed on a central UI where SAM Client is installed
- Each test gets the host name of the service node to test, and checks the node remotely by utilising grid command line tools
- Results returned immediately (exception: FTS)
- Possibility of delayed results - example: FTS sensor that submits test transfer jobs
- Frequency: currently once per hour

Existing tests - CE, gCE

- job submission - UI->RB->CE->WN chain
- version of CA certificates installed (on WN!)
- version of software middleware (on WN!)
- broker info - checking *edg-brokerinfo* command
- UNIX shells environment consistency (BASH vs. CSH)
- replica management tests - using lcg-utils, *default SE* defined on WN and a selected “central” SE (3-rd party replication)
- accessibility of experiments software directory - environment variable, directory existence
- accessibility of VO tag management tools
- other tests: R-GMA client check, Apel accounting records

Existing tests - SE, SRM

- storing file from the UI - using *lcg-cr* command with LFC registration
- getting file back to the UI - using *lcg-cp* command
- removing file - using *lcg-del* command with LFC de-registration

Existing tests - LFC, FTS

- LFC
  - directory listing - using *lfc-ls* command on /grid
  - creating file entry in /grid/<VO> area
- FTS
  - checking if FTS is published correctly in the BDII
  - channel listing - using *glite-transfer-channel-list* command with ChannelManagement service
  - transfer test (in development):
    - submitting transfer jobs between SRMs in all Tier0 and Tier1 sites (N-N testing)
    - checking the status of jobs
    - Note! The test is relying on availability of SRMs in sites

Existing standalone sensors

- GStat:
  - site-BDIIs: accessibility (response time), sanity checks (partial Glue schema validation)
  - top-level BDIIs: accessibility (response time), reliability of data (number of entries)
- RB:
  - jobs submitted through all important RBs to selected “reliable” CEs
  - measuring time of matchmaking

Adding new tests

- SAM is designed to be easily extended by adding new sensors or tests for existing sensors
- Examples:
  - additional sensors and tests for certification testbed
  - experiment specific tests - LHCb already added their own test to the framework
- Sensors for multiple service types - example: host certificate validity test (ready but not used)

Ongoing work

- VO specific submission:
  - LHCb already submits jobs (not frequently)
  - Atlas - will be submitted from central SAM UI
- Jobwrapper tests:
  - simplified set of tests executed on WNs by the CE wrapper with each grid job
  - will potentially reach all WNs in relatively short time (fixing the hole in SAM/SFT)
- Alarm system - basic development done in SAM, however development of the interface in COD dashboard is in progress