A Service-Based SLA (Service Level Agreement) for the RACF (RHIC and ATLAS Computing Facility) at Brookhaven National Lab

Mizuka Karasawa¹ (mizuki@bnl.gov)
Tony Chan¹ (tony@bnl.gov)
Jason Smith¹ (smithj4@bnl.gov)

¹ Physics Department, Brookhaven National Laboratory, Upton, NY 11973-5000, USA

Abstract: The RACF provides computing support to a broad spectrum of scientific programs at Brookhaven. The continuing growth of the facility, the diverse needs of the scientific programs and the increasingly prominent role of distributed computing requires the RACF to change from a system to a service-based SLA with our user communities. A service-based SLA allows the RACF to coordinate more efficiently the operation, maintenance and development of the facility by mapping out a matrix of system and service dependencies and by creating a new, configurable alarm management layer that automates service alerts and notification of operations staff. This paper describes the adjustments made by the RACF to transition to a service-based SLA, including the integration of its monitoring software, alarm notification mechanism and service ticket system at the facility to make the new SLA a reality.

1. The SLA Concept and the Elements:

A Service Level Agreement (SLA) is a key component of an IT organization's service level management (SLM) strategy. It's commonly seen in companies, whether it's an IT department that is performing services for internal customers or that offers IT services to their external customers. A SLA is a clear procedure and is used as the benchmark for how well the services are provided. In the RACF at Brookhaven National Lab, due to the prominent role of distributed computing and the continuing growth of the facility, a new model with service-based SLA has to replace the old model (the system-based service) in order to achieve a good quality of service with the user communities.

The SLA has few important elements:

Description of Service:

Specify the catalog of services that are provided and corresponding components, including applications, infrastructure and other business functions. This requires a good understanding of what services the facility offers to the user communities and affirms to the user communities that the facility understand its responsibilities.
Service Standards:

Once the services are established, the standards have to be considered include concepts such as availability and reliability, as well as response and resolution times. It also has to be determined if the facility can offer the services if a disaster or emergency happens, and if the same hours of operations can be provided during one of those scenarios.

Roles and Responsibilities:

Designate the responsibilities cross the groups in the facility for the SLA.

Service Level Management:

Define the person who is responsible for managing each service, responsible for taking actions when incidents occur and are due and who is responsible for communicating with the user communities about issues pertaining to the service and its performance.

Evaluation Criteria:

The material/tools that are used to determine how well the facility is performing.

2. How the SLA Model is Applied to the RACF:

Previous Model at BNL:

• Sub-groups are responsible for systems within the facility (tape storage, disk storage, linux cluster, grid-computing, network etc.).
• Sub-groups may have their own monitoring systems, different monitoring mechanisms.
• Facility systems were classified into three categories: non-essential, essential and critical. Critical components were covered 24*7 year-round, essential components have built-in redundancy/duplication, addressed in the next business day. Non-essentials were addressed in the next business day.
• Staff provide primary coverage during normal business hours, operators contact on-call person during off-hours and weekends.

New Model:

• Introduce the concept of SLA, define the new service categories with service components mapped out with each category as well as service dependencies defined.
• Classify the services into different service levels: Critical, High availability, Medium availability, Low availability. Each of these have standards such as availability, response time, resolution time defined.
• Designate the roles and responsibilities of each service component including service level managers who coordinate the operations and take further actions when incidents are due.
• Merge the monitoring within sub-groups into single monitoring system.
• Set up the evaluation criteria to determine how well the facility is performing.

3. How the SLA Inter-operates With the Monitoring System and Automates the
Notification & Escalation Actions

Previous System Architecture With Existing Model:

The open source monitoring system (Nagios) monitors services including network, servers/hosts, applications within the facility, sends the alerts to the responsible technicians and alerts ticketing system (RT) when an incidence occurs, RT opens the ticket correspondingly and associate the incident with the proper queue.

New Architecture With SLA Integrated:

First have services categories and associated services defined, with each service, the service level is defined with availability, response time and resolution time. This is written to a text format rule file which will be processed by rules engine implemented on SLA system. The following is an example of the rule being defined for a service (“dCache Thumper Pools and Space”) for service category (“Access to mass data archive – Atlas”) as seen in following screen-shot of the Matrix:

[Rule-name]

host: hostname
service: servicename
hostgroup: #RE compatible field (default ‘any’)
queue: #RT Queue
owner: #RT ticket owner (default queue owner)
priority: #RT ticket priority (default ‘none’)
auto_up: #0/1 to indicate if paging should occur (default ‘0’)
firstContact: #First person to page, can be pager #, or rt name (default disabled)
secondContact: #Second person to page, can be pager #, or rt name (default disabled)
importance: #0-10, only used if more than one rule applies to a host & service (default 0)

Examples:
[Thumpers ]
**RACF SLA Status Matrix**

| Thu Feb 4 17:23:17 2010 | Access to mass data archive - RHIC | Access to mass data archive - ATLAS | Access to NFS storage | Access to AFS storage | SMB File Service (Samba) | Data Catalogs | Us Data |
|-------------------------|----------------------------------|----------------------------------|-----------------------|-----------------------|-------------------------|--------------|--------|
| HPSS                    |                                   |                                  |                       |                       |                         |              |        |
| dCache                  | XXXX                             | XXXX                             |                       |                       |                         |              |        |
| DQ2                     |                                   |                                  |                       |                       |                         |              |        |
| FTS                     |                                   |                                  |                       |                       |                         |              |        |
| PANDA                   |                                   |                                  |                       |                       |                         |              |        |
| Gatekeepers             |                                   |                                  |                       |                       |                         |              |        |
| Gateways                |                                   |                                  |                       |                       |                         |              |        |
| Farms                   |                                   |                                  |                       |                       |                         |              |        |
| Condor                  |                                   |                                  |                       |                       |                         |              |        |
| User Databases          |                                   |                                  |                       |                       |                         |              |        |
| Facility Databases      |                                   |                                  |                       |                       |                         |              |        |
| NFS                     |                                   |                                  |                       |                       |                         |              |        |
| AFS                     |                                   |                                  |                       |                       |                         |              |        |
| Backup                  |                                   |                                  |                       |                       |                         |              |        |
| Kerberos                |                                   |                                  |                       |                       |                         |              |        |
| MyProxy                 |                                   |                                  |                       |                       |                         |              |        |
| VOHS                    |                                   |                                  |                       |                       |                         |              |        |
| GUPS                    |                                   |                                  |                       |                       |                         |              |        |
| DNS/LDAP/NTP            |                                   |                                  |                       |                       |                         |              |        |
| Gratia                  |                                   |                                  |                       |                       |                         |              |        |
| RT                      |                                   |                                  |                       |                       |                         |              |        |
| Nagios                  |                                   |                                  |                       |                       |                         |              |        |
| Ganglia                 |                                   |                                  |                       |                       |                         |              |        |
| Web Server              |                                   |                                  |                       |                       |                         |              |        |

Screen-shot of RACF SLA Status Matrix

SLA layer is added as the intelligence layer between monitoring system (Nagios) and ticketing system (RT). The rule engine is capable of processing rules which can be written in the text format and have service level associated with the services that are monitored in monitoring system. When an incidence occurs, monitoring system sends the alert to SLA system, SLA reads the rules and makes corresponding actions such as emailing/paging the technician who is responsible for that service, also escalating the incidents to the next service level manager if the incident is near or past due, once the service is recovered, it automatically closes the ticket with the RT ticketing system.
4. Future Plan & Improvements:

4.1 Merge the SLA rules/escalation code with ticketing system code due to the close relationship between SLA & RT.

4.2 Change the Nagios to SLA notification method from the Nagios notification handler to an independent SLA script which will read the Nagios object cache directly in order to process more internal Nagios information and keep the consistency and accuracy of SLA monitoring data.

4.3 Enhance the rule engine in order to deliver more efficient and informative alerts.

4.4 Enhance the current Web UI to present user communities with recent services status and how well the services are provided.

4.5 Enhance the evaluation material to be capable of generating service-based reports to help measure how well the facility is performing.

5. External links:

[1] Service level agreement – Wikipedia: http://en.wikipedia.org/wiki/Service_level_agreement

[2] Build you SLA with these five points in mind: http://articles.techrepublic.com.com/5100-10878_11-1049067.html

[3] A Service-Based SLA Model – Tony Chan: http://indico.cern.ch/contributionDisplay.py?contribId=36&sessionId=9&confId=27391
[4] Nagios: The Leader and Industry Standard in Enterprise System, Network and Application Monitoring:  
http://www.nagios.org/

[5] RT: Request Tracker:  
http://bestpractical.com/rt/