SLA Monitoring for service providers

White Paper

Turning SLA monitoring into a source of income for service providers

Introduction

When a new service is offered to the world, early adopters are usually more forgiving when checking the service availability. As time passes, new customers which are usually more demanding adopt the new service. At that point customers expect the service to be of high quality and to have high availability.

When talking about computer & network based services, many services are critical to the operation of the customer, and their lack of availability can be very costly to the customer even if they are not available for only a few minutes. These customers will choose their supplier by the service level agreement they are willing to offer.

Service level agreement

A service level agreement (SLA) is usually a negotiated agreement between two parties where one is the customer and the other is the service provider. This can be a legally binding formal or informal "contract".

The SLA records a common understanding about the services, priorities, responsibilities, guarantees, and warranties. Each area of the service scope should have a "level of service" defined.

The SLA may specify the levels of availability, serviceability, performance, operation, or other attributes of the service, such as billing. The "level of service" can also be specified as "target" and "minimum," which allows customers to be informed what to expect (the minimum), whilst providing a measurable (average) target value that shows the level of organization performance.

In some contracts, penalties may be agreed upon in the case of non-compliance of the SLA.. It is important to note that the "agreement" relates to the services the customer receives, and not how the service provider delivers that service.

SLAs are more common as part of a contract with corporate customers. However this practice is spreading and now it becomes more common even for a home customer to engage a service provider by including a service-level agreement in a wide range of service contracts in practically all industries and

markets. Internal departments (such as IT) in larger organization have adopted the idea of using service-level agreements with their "internal" customers — users in other departments within the same organization. One benefit of this can be to enable the quality of service to be benchmarked with that agreed to across multiple locations or between different business units. This internal benchmarking can also be used to market test and provide a value comparison between an in-house department and an external service provider.

Service-level agreements are, by their nature, "output" based — the result of the service as received by the customer is the subject of the "agreement." There could be different levels of SLA agreements, depending on the type of the customer, and his willingness to pay for the service. For example Platinum customers will be offered higher SLA then a Gold or a Silver customer.

Proactive SLA measurement

Measuring SLA depends on the type of service supplied. In this paper we will focus on services that can be monitored from remote. Examples of such services can be:

- An availability of a network based application:
  - A web server
  - A mail server
  - A banking service or any application accessed via the network
- An availability of a physical component:
  - A communication line
  - A router
  - A cable modem
  - A UPS device
  - An ATM

To monitor such components, there is a need for a tool that will be able to:

- Have the ability to assign a component to a given service level based on its status.
  - A component might have a range of statuses that might be assigned to different service levels.
    - The statuses must have an order so that when a higher status for a component is discovered, the component is moved to the right service level.
- Query the components at a predefined frequency,

- The tool should be scalable to support monitoring any size of components, at whatever frequency needed

- Extract the needed information to determine each of the the status of the monitored components
- Generate reports on the components needing attention based on their service level.
  - There should be a query based mechanisms to view the problematic components
- Generate historical reports of statuses of given components.
- Have an ability to define what is the service-level agreement of a component, and when it is bridged
  - What percentage of the time a component be in a given status, before the agreement is bridged
- Allow the user change the status of a component manually & automatically
  - This is used in order to change the status of a component to indicate it is being handled
- Have a historical report for the states of a given component

Monitoring needs

An SLA monitoring tool should first be a strong monitoring platform. The number of SLA monitored components is growing in an exponential rate, when such components can be either hardware devices or a software applications. Those components get more and more complicated.

Increasing number of monitored parameters

If in the past it was enough to know if a component is up or not, however today we can not suffice with only this information. SLA is determent by much more detailed information on each component. We want to know for example:

- What its performance & response time
- What resources does it use, and how much of each resource does it use
- What is it's response time
- whats it's jitter
- And so on.

Multi-Protocol parameter collection

More intelligent components, do not lead to more standardization in accessing the required parameters. Any SLA monitoring solution should support multiple methods for collecting data.
Examples to such protocols are:

- snmp
- icmp

- wmi
- http
- sql
- telnet
- ssh
- and more

Simple installation and operation

As the SLA monitoring operations get more and more complex, the need to simplify its installation and configuration grows. It is not very practical to have a monitoring agent for every component or small group of components, when the number of monitored components grows rapidly. This makes the operability and manageability almost impracticable. The ideal solution could have been a centralized SLA monitoring product, which is able to perform its required actions from a single point, however it is clear that this approach will fail on scale.

It looks like the optimal solution will be a centralized SLA monitoring platform with as less agents as possible, when every agent monitors as many components as possible. This will simplify the installation and operation of this kind of platform and enable scale.

Complex correlations and conditions on the monitored parameters

The requirements for conditions that are needed to be tested on the monitored components have also changed dramatically in the past few years. If in the past, events were based on status changes and threshold passing, today there is a need to correlate multiple parameters' behavior over time, in order to detect anomalies in the components, and alert or true problems and not false alarms.

Flexibility

As the number of possibly monitored parameters increase exponentially, and the number of the possible correlations that make sense increase exponentially too, it is clear that out of the shelf products can not prepare in advance an answer to all the possible needs of all enterprises. As such it is needed that a monitoring solution will be flexible in a way that will enable with ease to add more monitoring parameters and more correlation rules, and conditions.

Scalability

Scale can be measured by this relation:

monitored components x
average parameters count x
average monitoring frequency x
average affective correlation rules

of course, this equation, gives only a ruff estimate to the scale required by a monitoring solution. As can be seen from the previous equation there are 4 major aspects of scale that should be taken into consideration:

Component count scale

As said before, the number of intelligent components in the enterprise increases exponentially both software and hardware. Any monitoring solution must address this increase in a way that will not increase the monitoring cycle time.

There should not be also any effect between the monitored nodes (i.e, if a given node does not respond it should not effect the time the other node is monitored.

Collected parameters count scale

Each component has multiple parameters that can be collected from it. Every monitoring solution should make it possible to collect the increasing number of parameters from the different components.

Monitoring frequency

Once it was good enough to make a certain checks every day or hour, now it is required to know of changes to parameters, on line, as they occur. A monitoring solution must support short monitoring cycles.

Correlation rules

Now days it's not enough just to collect the data, checks must be made on the collected data, and some of the checks can be quite complicated. A monitoring solution must scale well with the number of checks performed.

Monitoring methods

There are many methods for monitoring the components that compose the enterprise. For example:

1. An agent located on the component, collecting the requested information and passing accumulated data to a central point.
2. A remote agent collecting data from multiple components accumulating the collected parameters and passing them to a central point
3. A central collector which collects data from all components, and processes them.

For different components there are different optimal data collection methods. A SLA monitoring platform should support all methods

Central monitoring data repository

No matter what monitoring method is taken, it is recommended that there will be a central repository holding all the monitoring data. This repository can be used for:

1. History data collection
2. Trends analysis, and pattern discovery
3. A central location for data used for events correlations and compound conditions

and many more uses.

Monitoring Data format

The monitored parameters data can be kept in many formats however there are 2 recommended methods to hold the data:

1. Summary data should be held in a relational database, so it will be easily queried and reported.
2. History data should be held in an RRD format (Round Robin Database). This method is a recommended because it puts bounds to the amount of storage used for keeping the data.

Collecting monitoring parameters and keeping them in a raw data format, allows the maximum flexibility, but it also becomes a liability, specially in large organizations when the size of the data collected can increase to huge amounts, and since there is hardly any critical need to know what was the exact response time for a given transaction a few months ago.

Reporting tools

SLA products should be able to display and priorities the handling of problems based on the SLA agreement with the customers. This mechanism is not trivial, as some times a bronze customer, how is near the end of the permitted handling window, will have higher priority that a gold customer who just got a problem. The product should be familiar with the concept of SLA, as defined by the customer, and should act accordingly.

It is recommended that the product will have topological views that will show problems based on SLA too. For example a view that at its top is divided by SLA groups (Platinum, Gold, Silver,...) and each group should be dived to areas, and there to companies up to the failing device.

SLA Monitoring platform Genie

SLA Monitoring Platform Genie from Jilroy Software addresses the issue of on-line proactive SLA monitoring by enabling real time monitoring, of large number of parameters in large organizations using a single central collection station or a relatively small number of agents to collect data from a large number of components in the enterprise. The product addresses all the existing monitoring needs of enterprises today, in a way that focuses on scale.

Addressing the Monitoring needs

Discovering the SLA monitored elements

SLA Monitoring Platform Genie comes with a very powerful discovery tool that can extract data from almost any source (from network devices, to databases and files). The product can discover the full network of the enterprise, or discover selected nodes from the full network.

Defining SLA parameters

SLA Monitoring Platform Genie enables the user to define for each monitored element, what is its SLA parameters. The product will use this information to determine the recommended priority of a failing node.

Increasing number of monitored parameters

SLA Monitoring Platform Genie enables the user to determine which parameters will be monitored, and in which frequency, on which components. The selection is very robust, and very simple. The user can define which parameters will be collected from which components, using masks on discovery information on the components. The product will know based on the accepted result what is the status of the element, and determine the priority required for handling an error event.

Multi-Protocol parameter collection

SLA Monitoring Platform Genie is built to support multiple collection protocols. It currently supports the following protocols:

- ICMP
- SNMP
- SQL
- HTTP
- CSV
- Telnet
- SSH
- WMI
- TCP Ports monitoring
- and others.

Its support of the monitoring protocols integrates with all its other capabilities, like support in increasing number of monitored parameters, ease of installation and scale.

Simple installation and operation

SLA Monitoring Platform Genie, is built as a centralized monitoring product. It is centrally managed and controlled. It can operate on a single machine monitoring large numbers of parameters on large number of components in high frequency. When scale requires, it supports distribution of its components to additional machines. It's installation and use out of the box are very simple, however it enables the advanced user to tailor it to its exact needs.

Complex correlations and conditions on the monitored parameters

SLA Monitoring Platform Genie has a built in mechanism for defining correlation rules and conditions on the monitored data, including the historical data, so that events can be scheduled and event handlers launched, based on these correlation rules and conditions.

Flexibility

SLA Monitoring Platform Genie was designed for flexibility. It has several levels where it becomes obvious:

Monitoring Parameters selection

With SLA *Monitoring Platform Genie* it is easy to define what parameters will be monitored on which nodes, in a simple and robust way that uses masks and information discovered on the monitored components. It is not needed to specify for each node specifically what parameters will be collected. It is possible, and recommended to use generic rules to define those values.

Graphical User Interface flexibility

The product was designed with SLA in mind. It has reports that understand the SLA concept, and has a mechanism to determine the priority that should be given to a failing component. The product has also support of topological maps with color propagation, that are showing SLA related information. The product has built in simple tools that enables the user, to tailor it's User interface to its exact needs. The user can change the product's menu, and add or remove query reports & topological maps. All that does not require any programing skills, and can be done at the site level.

The product has also a permission system that can control what each user will see. This enables customers to view SLA reports only for their components.

There is a Web based GUI designed for specially for customers to see SLA reports for their monitored comonnents.

Product's components location

In order to support scale the product was designed to allow flexibility in the location of the product's components. These components can be moved from the server's central point to near by or remote computers/sites.

There is an option to locate monitoring agents in different organizations and to support any connection mode (Connection from the agent to the data collector or vice versa, according to the organizations security policy).

Scalability

SLA Monitoring Platform Genie was designed for scalability, in all the aspects mentioned above.

Component count scale

The product, out of the box, was built in a way that the monitoring components can collect data from a large number of components concurrently from a single machine, in parallel. However the product supports having multiple collection processes, located either on the same machine or distributed in the enterprise. This capability enables scale, as the monitoring of different components can be distributed over the different collection processes.

The product is built in a way that collecting data from one component does not affect the collection of data from other components.

Collected parameters count scale

Each collection process can collect multiple parameters from any given number of components. As described before, the scale in the number of monitored parameters is handled by adding more collection processes.

Monitoring frequency

As the product was designed in a way that monitoring of a given node, can be done in parallel with monitoring of others, there is no theoretical limit to the monitoring frequency, and the only actual limits are CPU, memory, and bandwidth, which are addressed by the capability of adding more collection processes, and distributing the monitored components between them.

Correlation rules

The correlation rules analysis is performed in a special process. This process can also be multiplied and distributed if scale is required.