DGP monitoring is focused on proactively collecting various metrics data from systems as they are being used. Together with logging, they provide observability into the functioning of each tier in a distributed system, while other tools are used to collect lower-level data from host operating systems, app servers, networks, etc. The most important capability is to be able to capture metrics from systems in production under normal workloads without adversely affecting the performance or scalability of the system.

Requirements

1. Collect end-to-end and server performance data from production systems during their normal use. Depends on message-based APIs.

What: *Measurements of the end-to-end performance data should be collected from native applications in production environments during their normal use, without adversely affecting the performance or scalability of the system.*

Why: *End-to-end performance experienced by users of client applications is the single best way to monitor a production system. Some "gray failures" of subsystems may not show up as errors or exception events, but will result in slower performance. Also, load or stress tests of systems are very difficult and expensive when trying to generate large volumes of synthetic traffic. Capturing real performance data from production systems is far better, as long as the collection of the data does not interfere with the operation of the production system. The RBAC security system is used to define a RemoteMonitor role for data collection, and limit the number of users that are members of the role to control the volume of performance data collected.*

Testing: *The API Tester test harness can test the API methods that collect performance data, as well as users that are members of the RemoteMonitor role.*

2. Collect end-to-end and server performance data from DGP automated processes. Depends on message-based APIs.

What: *Measurements of the end-to-end performance data should be collected from the AutoWork automated processing subsystem in production environments during their normal use.*

Why: *In DGP, all automated processes are implemented as API methods. The AutoWork service acts as a scheduler and a client application that calls the API process methods. Therefore, the same type of measurements such as end-to-end and server performance can be collected and logged by the AutoWork service..*

Testing: *The API Tester test harness can test the API methods that collect performance data, as well as test runs of the AutoWork service.*

3. Collect server-side resource utilization data from production systems during their normal use: Depends on the AutoWork subsystem.

What: *Measurements of various types of resource utilization (CPU, memory, network IO, storage IO, .NET working set, etc.) should be captured at regular intervals from each server in the system, without adversely affecting the performance or scalability of the system.*

Why: *The various performance metrics only provide part of DGP observability. It is just as important to collect measurements of efficiency in order to be able to extrapolate the total capacity of each server from the workload handled + resource utilization during actual use.*

Testing: *The API Tester test harness can test the API methods that collect resource utilization data, as well as test runs of the AutoWork service.*