

Operations Manager Performance Monitoring

Prior to upcoming upgrades or a new installation of System Center Operations Manager (SCOM), it is of high value to keep statistics on how performance is affected over time. It may also be that you make changes to the infrastructure where you want to understand if you get a positive or negative impact.

As a result, we have developed a free Management Pack for SCOM 2012 R2 and SCOM 2016 that simulates a number of user calls using powershell commands - as they communicate with SCOM in a similar way as the console through the SDK.

Powershell may not be entirely right for simulations of the console, but it at least it gives us an anchor point as to whether performance improves or deteriorates over time.

This also gives you as an platform administrators an opportunity to compare your values to other environments to get an idea of whether your environment works well or suffering from performance issues.

Operations Manager Performance Monitoring

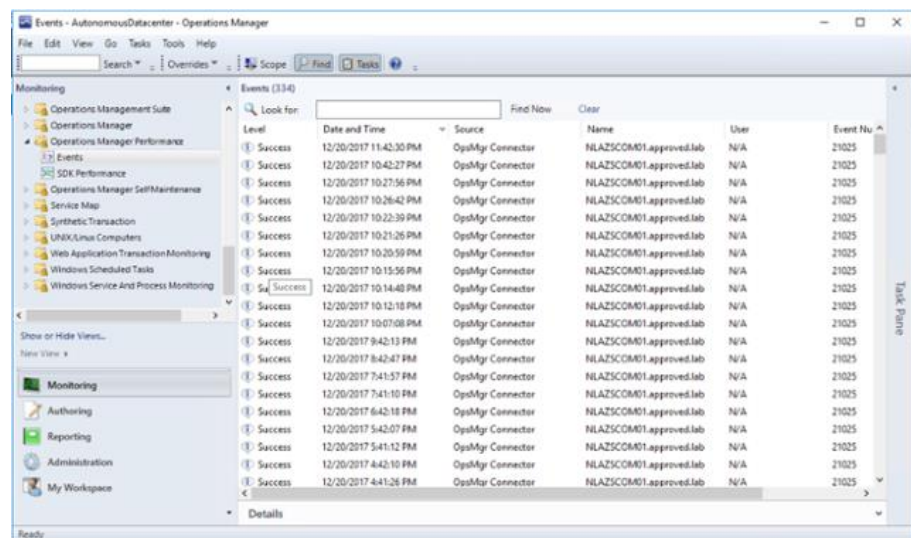
Our Management Pack consists of four different types of rules that run at different intervals and against two different classes for data collection. All rules are disabled on import, so selected rules need to be activated afterwards.

Internally at Approved, we use this in all our projects to compare the performance with previous versions of SCOM, but primarily to ensure the quality of the projects we work in, as we continually add more agents or management packs.

Views

You will find two different views under the "Operations Manager Performance" folder after the management package has been imported into SCOM.

- Events
- SDK Performance



Events

The rule that collects Event ID 21025 from the Operations Manager log is used to identify if there are abnormal changes in the environment that force SCOM to process new configuration files. The rule is called "OpsMgr Connector Event New Config Rule" and has "Root Management Server Emulator" as a target.

Details

Date and Time: 12/20/2017 11:42:30 PM
Log Name: Operations Manager
Source: OpsMgr Connector
Generating Rule: [OpsMgr Connector Event New Config Rule](#)
Event Number: 21025
Level: Success
Logging Computer: NLAZSCOM01.approved.lab
User: N/A

Description:
OpsMgr has received new configuration for management group AutonomousDatacenter from the Configuration Service. The new state cookie is "bac5dcb071164a8109ad198656403bbb,8851bf3d72f943d0b39a7f011a9523aa:000428E6"

In earlier environments, a phenomenon called "Config Churns" could be encountered, and was a major problem when you only had a "Root Management Server" that handled this kind of calculations. In later versions of SCOM, this load can be divided by several servers, but you do not want to change too much environment, especially daytime when you have operators who may work in the console.

More information about this can be found [here](#).

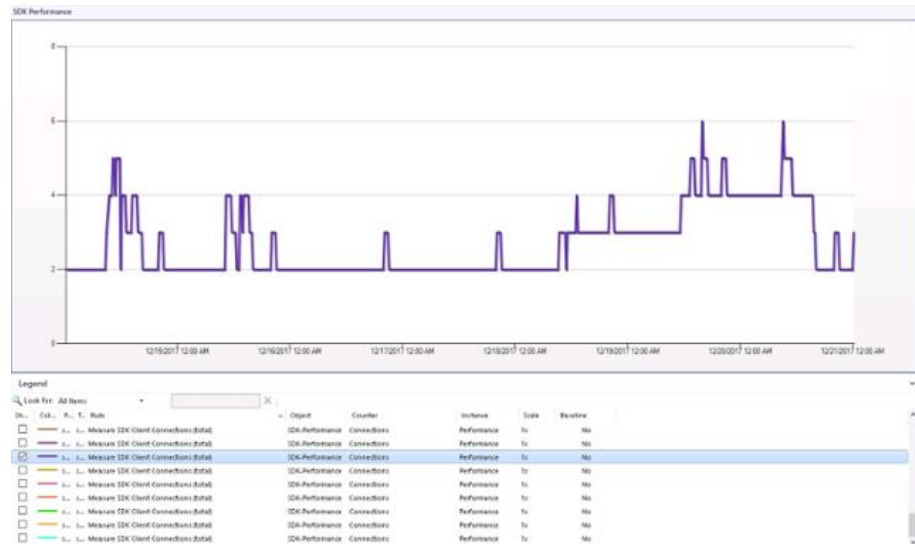
We analyze the trends of this event with the help of our [IT Service Analytics](#) tool. This gives us a whole new dimension to work with this kind of data and to help us identify patterns such as time of day or weekdays.



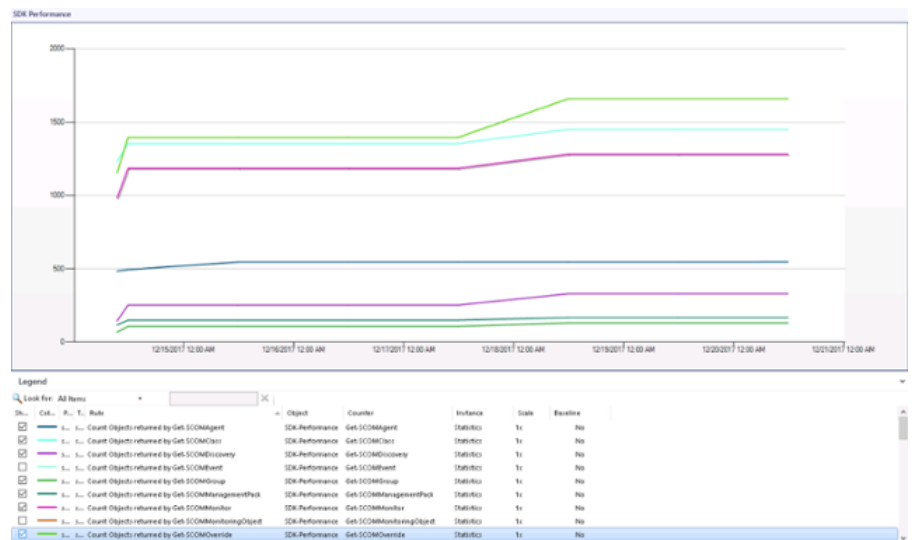
SDK Performance

To get a good picture of how performance is affected over time, we have three different rules available.

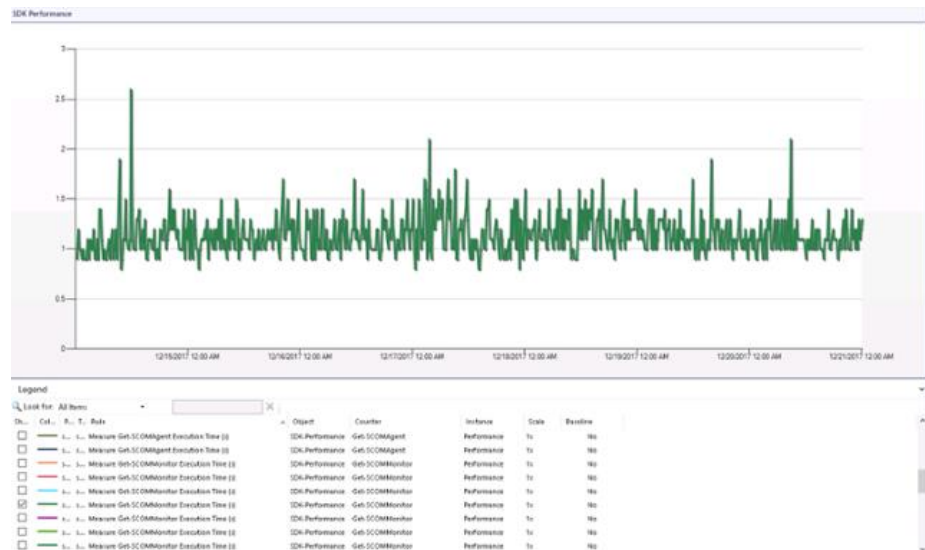
Measure SDK Client Connections (total). Collection of number of console connections on each management server. This differs from the built-in as it counts the total number of processes that are connected in one single rule. This rule has "Management Server" as target and runs as a default every 15 minutes.



Count objects returned by Get- (command). Collection of number of items returned on each execution. As an example, Get-SCOMAgent returns the number of agents installed in the environment and stores the result in each operations manager database. This rule has the "Root Management Server" as target and runs as a default once a day.



Measure Get- (Command) Execution Time (s). Measures the time it takes for the script to run and returns this value in seconds. This rule has "Management Server" as target and runs as a default every 15 minutes.



The above values can be compared in retrospect to see if changes made in the environment get a percentage improvement or deterioration and you can also see if it is due to too many console users against a specific server or too many changes that might occur in the environment.

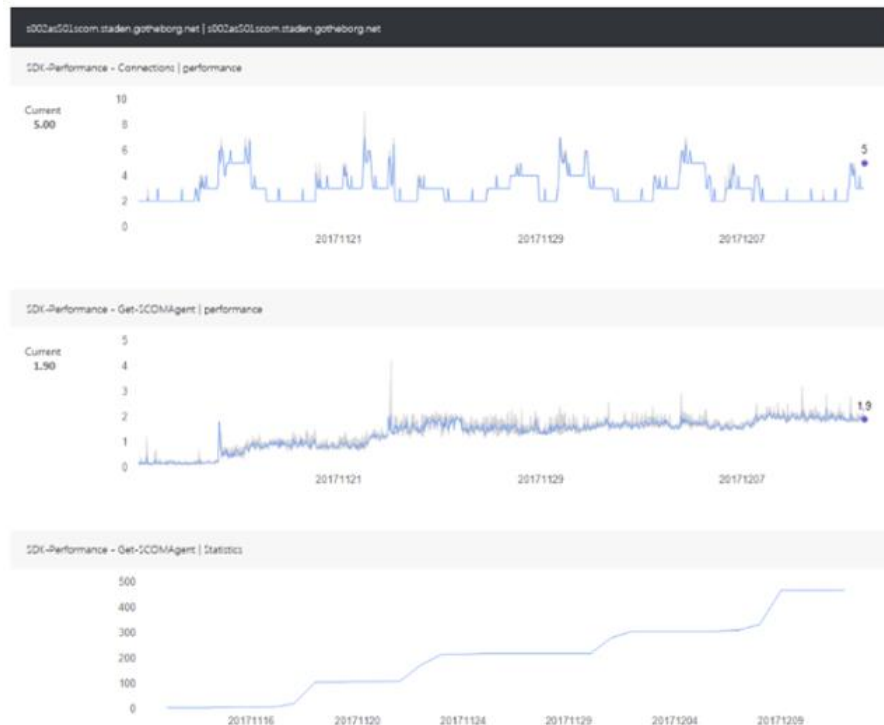
The Powershell commands, which are currently in use, are as follows:

- Get-SCOMAgent
- Get-SCOMClass
- Get-SCOMGroup
- Get-SCOMManagementPack
- Get-SCOMMonitor
- Get-SCOMMonitoringObject
- Get-SCOMOverride
- Get-SCOMRule
- Get-SCOMDiscovery
- Get-SCOMEvent

Please be careful before activating the rules and be sure that they do not affect your environment negatively. If you have not trimmed it well and, for example, collect a lot of events, the **Get-SCOMEvents** rule can result in long response times that may adversely affect performance. All scripts have a 120-minute timeout, which can be noticed in the error logs when changes occur in the environment that prevent the script from being executed. This is unfortunately quite normal and this is how SCOM works.

Follow-up of performance

To provide a good overview of performance, we can easily run a performance report through [IT Service Analytics](#) that shows an overview of the collected data.



To have a more detailed view at specific dates, click on a desired day to see more details on selected counters +/- 1 day.

