Monitoring Platform Genie

Testing Large-Scale ICMP Monitoring sample

Table of Contents

Preface

Welcome to the “Testing Large Scale ICMP monitoring sample” document . This chapter provides an introduction to the structure and assumptions of this guide.

The Purpose of This Guide

This guide contains information needed to run the test for Large Scale ICMP monitoring sample.

Who Should Use This Guide

This guide is intended for network managers familiar with network management and its fundamental concepts.

Prerequisites

You need to install and get familiar with the Monitoring Platform users guide.

Organization of This Guide

This guide is structured to reflect the following conceptual divisions:

- Preface – A description of the guide’s purpose, intended audience, organization, and conventions.
- Introduction section – A general description of the test and how to start it.
- Running the test section – Information about the steps that need to be executed in order to run the test, and analyzing the results.
- Discussion section – Contains information about the results and additional issues needed to be taken into consideration.
- Final Page – Information about contacting Jilroy Software.

Conventions

The manual uses the following conventions:

- Names of dialog boxes, windows, and unnamed screen areas are displayed in *italics*.
- Names of buttons, tabs, check-boxes, and other screen elements are displayed in **bold**. For example, click **OK** or type the **Start date**.
- **`This font`** is used for text that you enter.
- `This font` is used for code, directory names, file names, and system activity.

- UPPERCASE is used for keys and acronyms.
- Steps that involve two or more selections from a menu may be presented as a combination of selections separated by an > angled bracket.

For example, when you see **File > New**, click the **File** menu on the menu bar. This will open a drop-down menu. Then select the **New** command.

- Cross-references are underlined. For example, see Chapter 2.
- Hyperlinks are underlined and blue.
- The ⓘ symbol signifies notes, which are used to provide extra or special information regarding the preceding topic.
- The *Italic* font style is used to *emphasize* words and phrases in special cases.

Introduction

Writing large scale monitoring platforms requires a lot of consideration when designing and writing code. Things that take relatively small time when dealing with 10s or 100s of objects, become inefficient and impractical when working with 10,000s of objects and more. For example operations that could have been done serially now must be done either in parallel or in an asynchronous way.

Large scale monitoring is not an easy thing to test too.

- First you must have a large number of objects that you can monitor.
- Those objects must be accessible to the monitoring agent using the desired protocol.

This document describes a large scale test that can be performed anywhere, from any computer that has access to the Internet.

The document explains:

- The test scope & environment
- How to run the test
- How to analyze the output

The test scope

In the described test we are planning to test the status of 128K IP addresses in the Internet. Their status will be checked using the ICMP protocol.

We have chosen 2 class B ranges in which we know there are some web servers, and we are evaluating their status in a periodic way every 5 minutes.

The test can be run from almost any station that have a connection to the Internet, but it should be clear that the quality of the line will determine how many ICMP requests will be discarded along the path to the target and how many ICMP responses will be discarded on the way back.

It is not clear how many devices will respond to the ICMP requests in this environment, and it should also be clear that routers and switches throw ICMP either by definition or when in congestion.

The state of the monitored elements should

How to run the test

Install the product

Before running the test you have to install the product.

In windows the setup program will perform all needed actions, and in Unix after you tar the installation file to the right destination you will have to run the bin/install script.

There is a possibility that all the monitoring processes will run, however if you have a weak computer that might cause slowdowns.

If you want to stop all the monitoring processes you can run the command:

bin/monitoringplatform_stop

Required services

In order to run the test you have to start the following processes:

In unix.

- The database using bin/startdb
- The datacollectionserver using **bin/datacollectionserver.sh**
- The icmpdatacollectionagent using ***bin/icmpdatacollectionagent.sh***

for the GUI we need

- The tomcat application server using ***bin/runtomcat***
- The gui using ***/bin/monitoringplatformgui***

In windows the monitoring services, the database and the web server are windows services and the the GUI is started from Start->Programs

Loading the information on the monitored elements

The monitoring process requires an inventory database containing the monitored elements.

Prior to running the Inventory process, start the database.

In order to fill the inventory database you will have to run an already prepared discovery rule named:

FastIpRange128K

Run the following command: ***bin/inventory FastIpRange128K***

which will fill tblObject with information about the monitored addresses. The process assumes an empty tblObject and generates 128K addresses.

This operation should be done once, and may take a few minutes to complete.

Checking configuration files

After having the inventory information in place, and prior to starting the required services go over the configuration files to understand how you monitor things.

Config.prop

This file contains the default definitions for the monitored elements monitoring parameters

We will only focus on the default values:

```
[Default_monitor_parameters]

Timeout_in_sec=10

Retry_count=5

Monitoring_cycle_timeout_in_sec=300

Monitoring_count_in_cycle=1

Inter_cycle_monitor_delay=1

Start_monitoring_time=NOLIMIT

Stop_monitoring_time=NOLIMIT

Backbone_failure_operational=0
```

A detailed explanation on the parameters can be found in the Monitoring Platform users guide. In this section we will focus on the important ones.

Timeout_in_sec

Defines how long do we wait for a reply. 10 sec is a good value and it does not disturb the monitoring of other nodes if the “Max_concurrent_requests” parameter In the agent prop file is large enough.

Retry_count

The retry count together with the timeout_in_sec determines how long it will take to decide that an element is not responding. The value will be Timeout_in_sec * Retry_count second.

Note:it does not disturb the monitoring of other nodes if the “Max_concurrent_requests” parameter In the agent prop file is large enough the “Monitoring_cycle_timeout_in_sec” is large enough, otherwise optimizations should be made.

Monitoring_cycle_timeout_in_sec

This parameter determines how long we will wait for the next monitoring cycle after figuring what was the status the previous round. This parameter should be set according to the needed frequency.

Notes:

- A too small value will cost more bandwidth but will be more accurate.
- The default we gave was 5 minutes.

Inter_cycle_monitor_delay

This parameter is set for Jitter measurements. In this test, set it to 1.

Start_monitoring_time

We did not give any limit to the start monitoring time.

Stop_monitoring_time

We did not give any limit to the stop monitoring time.

Backbone_failure_operational

We did not activate the backbone failure process.

Note

You can override these parameters on a monitored element level by altering the relevant SQL filter file (see later an explantion on that)

ICMPSQLFilteringFile.txt

This file selects the monitored elements. We selected all the addresses in the inventory database.

The SQL command was:

select nObjectId,strIpAddr,strDNSName from tblObject where strType = 'Address' and strMonitoringMethod = 'ICMP';

look at the file and the documentation for additional parameters selection. As mentioned before you can use this method to change default monitoring parameters on the element level.

configICMPDataCollectionAgent.prop

This file determines how the monitoring agent operates.

```
[General]

Verbose=FINEST

ServerURL=http://127.0.0.1:54346/

[DataCollection]

Push_data_timeout_in_sec=60

[ICMP]

Send_delay_count=1000

Max_concurrent_requests=5000

Only_send_change_status_information=yes
```

The important parameters that are related to scale are:

Send_delay_count

This parameter specifies how many monitoring requests will be sent in a given second. We found that when working with an ADSL connection and a simple ADSL router than a value of 100 is reasonable

an a value of 1000 packets/sec is too big and causes the system not to stabilize, however in a strong connection to the Internet and a powerful machine & fast lan, values of 5000 are working fine.

Max_concurrent_requests

Specifies how long open requests can be concurrently pending. We have set the value to 10Ks and even more, and in this test it had little effect.

Only_send_change_status_information

This parameter determines if the agent sends information about response time on each monitored element. As we did not start the log collectors we disabled this option.

Additional parameters

In order to reduce the CPU costs, allow running this product on a single workstation, and avoid unneeded operations we disabled the status change job scheduling by not defining jobs that need to be executed.

Running the monitoring processes

After building the inventory database and setting the right parameters in the configuration files we can now start the monitoring services.

Start the ICMP monitoring service. Note that under a unix platform it must be executed with root permissions as sending ICMP requests (RAW sockets) requires root permission.

Start the Data Collection server.

Stabilizing the system

After starting the monitoring processes, there is a time period that will take the system to stabilize. In this period status records will be written for each monitored element. This can take a few minutes.

You can see if the system stabilized looking at the pending SQL commands queue in the System counters report.

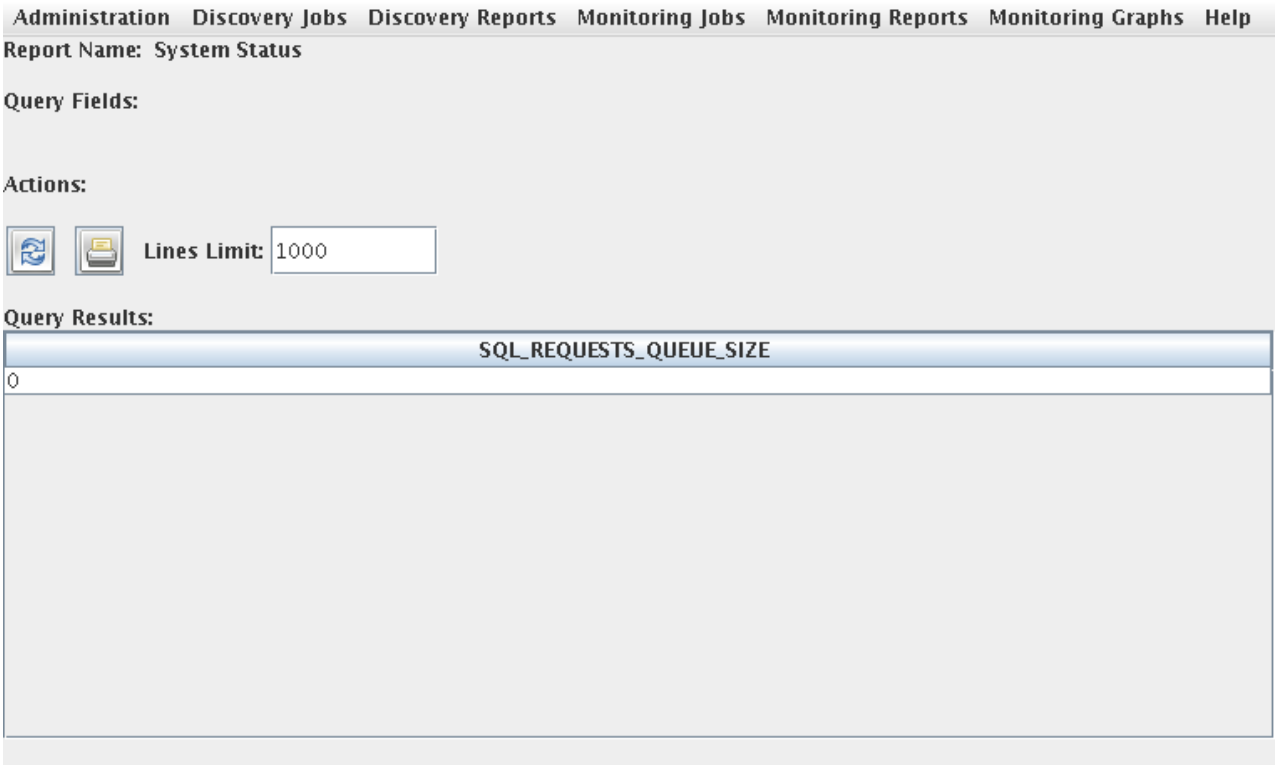
We have set the default status for all nodes to be “DOWN” assuming that most of the addresses selected will be down, however this is controlled by the configuration files.

After the stabilizing period we expect to have relatively little changes over time, and at that time the

system will give accurate values immediately. In the instability period there might be inconsistencies between the database representation of the elements status and their actual status.

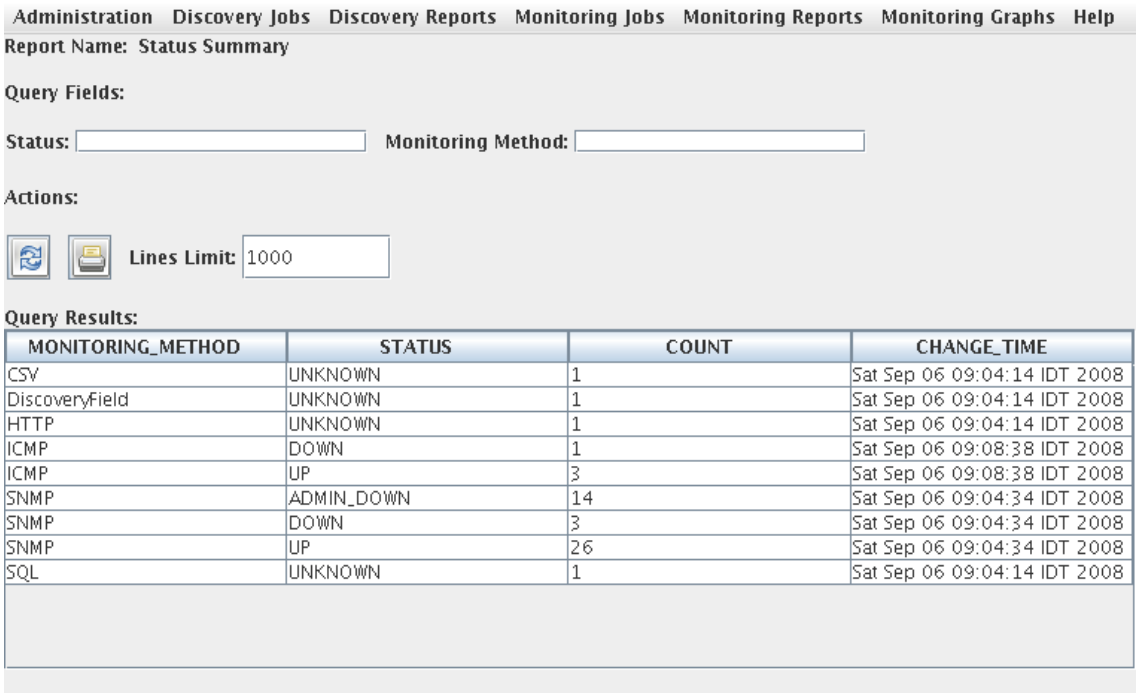
You can identify that the system is stabilized by looking at the following reports.

Under Administration->System Status report you will see a counter of pending SQL commands. This counter indicates how many update commands are waiting. When this number is zero or low, then the system is stabilized.



Another way to see if the system is stabilized is by looking at the

Monitoring Repots ->Status Summary



MONITORING_METHOD	STATUS	COUNT	CHANGE_TIME
CSV	UNKNOWN	1	Sat Sep 06 09:04:14 IDT 2008
DiscoveryField	UNKNOWN	1	Sat Sep 06 09:04:14 IDT 2008
HTTP	UNKNOWN	1	Sat Sep 06 09:04:14 IDT 2008
ICMP	DOWN	1	Sat Sep 06 09:08:38 IDT 2008
ICMP	UP	3	Sat Sep 06 09:08:38 IDT 2008
SNMP	ADMIN_DOWN	14	Sat Sep 06 09:04:34 IDT 2008
SNMP	DOWN	3	Sat Sep 06 09:04:34 IDT 2008
SNMP	UP	26	Sat Sep 06 09:04:34 IDT 2008
SQL	UNKNOWN	1	Sat Sep 06 09:04:14 IDT 2008

When this value does not change rapidly, it means that the monitoring process has stabilized, however it does not mean that the interface status report is accurate, because as we explained in the previous paragraph we have put the update of the database in a set of worker threads, not related to the actual monitoring process, and this allows the monitoring to perform its actions without any delay caused by the I/O to the database. We write the summary information regardless to the update of the information on the monitored element level.

Note: We can can control the size of the SQL writers thread pool, in the config.xml parameters.

How to analyze the output

We explained how to identify when the monitoring process has stabilized, in the previous section.

In this chapter we will talk about how to interpret the outputs of the product.

Send Delay parameter and the devices buffers

The send delay parameter defines how many requests will be sent within a second. This parameter has a lot of effect when working with different connections of the monitoring platform to the network and to the Internet.

Send & Receiver Buffer sizes

Application and network devices have buffers that hold data pending their sending to the network or their retrieving by the application. These buffers have limited size.

If we send or plan to receive to much data, than it is possible that these buffers will overflow and we will loose data.

By default we have set the send/recv buffer sizes of the application to a large value of a few mega bytes (it is controlled in the config.xml files), and in linux environment we have given instructions on how to set the operating system buffers, however there are buffers that are controlled by us, such as the router buffers.

The Send Delay determines the rate of putting packets out. You can see that its value is not right (too big) if you see that you do not get a lot of responses when the system first starts from scratch, via the “Status Summary” report.

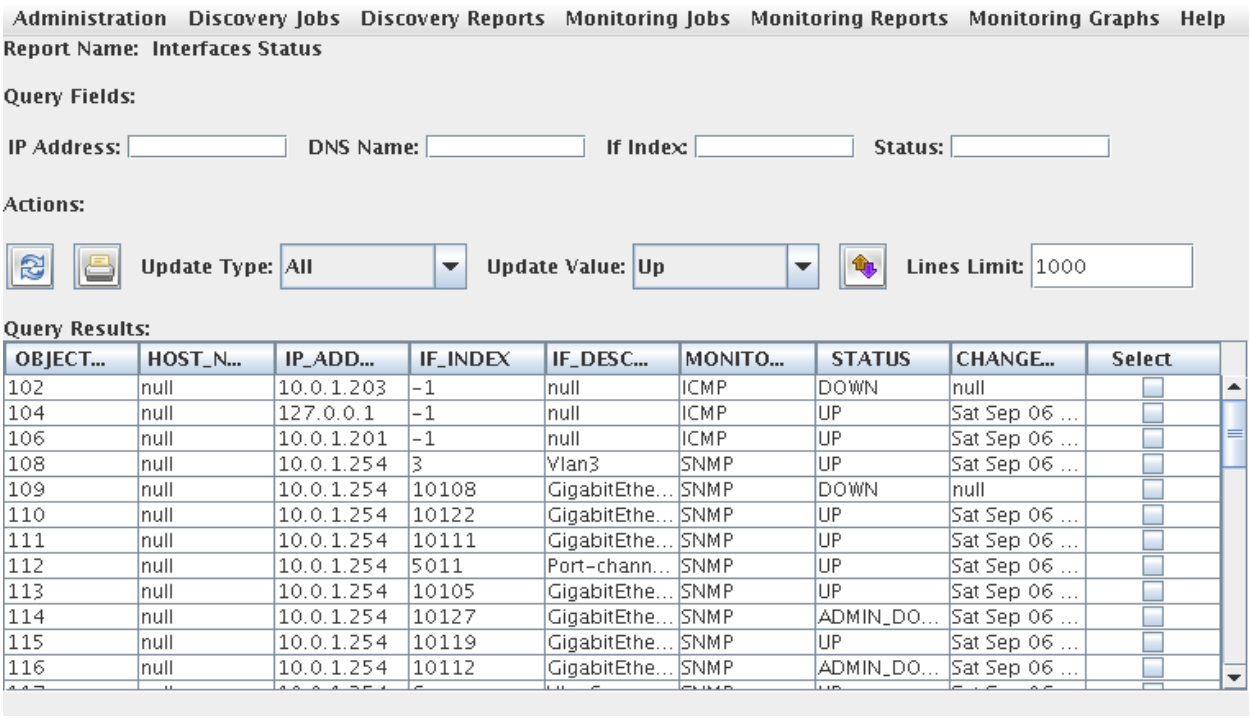
Note that this parameter will determine the time it takes to finish a cycle of sending requests to the target. Note that giving a low value to this parameter will cause the finishing of a send cycle to all monitored elements to take along time. For example for 100K elements and a send rate of 100 requests per second, it will take 1000 sec to finish a single cycle.

Dropping ICMP requests in congestion

Routers are configured by default to drop ICMP echo requests and UDP requests in a case of congestion. This will have an impact on our product. Again you can see it in the rate you get responses in the beginning of the monitoring process.

How fast do we discover that a node is down

The best way to discover how fast the product discovers if a node is down, is to add a node close to us to the monitored elements list and then after the system has stabilized, disconnect it from the network and see when its status is changes using the reports in the: Monitoring Reprts->Interfaces status report.



OBJECT...	HOST_N...	IP_ADD...	IF_INDEX	IF_DESC...	MONITO...	STATUS	CHANGE...	Select
102	null	10.0.1.203	-1	null	ICMP	DOWN	null	
104	null	127.0.0.1	-1	null	ICMP	UP	Sat Sep 06 ...	
106	null	10.0.1.201	-1	null	ICMP	UP	Sat Sep 06 ...	
108	null	10.0.1.254	3	Vlan3	SNMP	UP	Sat Sep 06 ...	
109	null	10.0.1.254	10108	GigabitEthe...	SNMP	DOWN	null	
110	null	10.0.1.254	10122	GigabitEthe...	SNMP	UP	Sat Sep 06 ...	
111	null	10.0.1.254	10111	GigabitEthe...	SNMP	UP	Sat Sep 06 ...	
112	null	10.0.1.254	5011	Port-chann...	SNMP	UP	Sat Sep 06 ...	
113	null	10.0.1.254	10105	GigabitEthe...	SNMP	UP	Sat Sep 06 ...	
114	null	10.0.1.254	10127	GigabitEthe...	SNMP	ADMIN_DO...	Sat Sep 06 ...	
115	null	10.0.1.254	10119	GigabitEthe...	SNMP	UP	Sat Sep 06 ...	
116	null	10.0.1.254	10112	GigabitEthe...	SNMP	ADMIN_DO...	Sat Sep 06 ...	

Note that it should take at most the time specified in the parameter: Monitoring_cycle_timeout_in_sec specified either in the default settings of the config.prop, of the data collector or in the specific settings for the node.

Note also that the monitoring process is fair, and if the parameters you have set (send_delay) create a queue for sending, than those who wait longer will be sent before new arrivals to the queue. This might cause the system more time to detect changes on the selected node, if the parameters you have set are not reasonable to the size of the monitored elements group.

Final Page

More Information

More information about Jilroy Software, the Monitoring Platform product, and our other products can be found on our web site. www.jilroy.com

Contact Us

For any information or problem, request for information or extension idea related to the Monitoring Platform or any other product, please contact one of the following

email addresses.

Sales

sales@jilroy.com

Product Management

discovery pm@jilroy.com

FTP Site

All jilroy products can be downloaded from our Web site.