Profiler Tool

Repository

The tool is stored on svn at the address:
svn://serverurl//groundwork-professional/trunk/load-test-tools/profiler/profiler-console

The tools is also provided as a tar file, Profiler.tar.gz, which can be downloaded from
http://texas/build/index.cgi

Configuration

Untar Profiler.tar.gz , get the war file from:
profiler-java/profiler-console/target/groundwork-profiler.war
Copy the war file under the webapps directory of your j2ee container, or of foundation:
/usr/local/groundwork/foundation/container/webapps/groundwork-profiler.war

Also, the context file profiler-java/profiler-console/resources/groundwork-profiler.xml must be copied in the context directory of your container or in the context directory of foundation:
/usr/local/groundwork/foundation/container/contexts/groundwork-profiler.xml

The /usr/local/groundwork/foundation/container/etc/foundation.xml must be edited and following line (the bold one) must be added to:

```
<Set name="redirectURL">http://localhost</Set>
        <Set name="allowTargets">
                <Array type="java.lang.String">
                        <Item>/foundation-webapp/performanceDataPost</Item>
                        <Item>/foundation-webapp/services</Item>
                        <Item>/foundation-webapp/fwsmodel.wsdl</Item>
                        <Item>/groundwork-console/</Item>
                        <Item>/groundwork-profiler/</Item>
                </Array>
        </Set>
<Set name="debug">false</Set>
```

Database

The Profiler Tool needs a database, the GWProfilerDB.
Therefore, mysql server must be installed.
Copy
profiler-java/database/schema/mysql/create-db.sql
profiler-java/database/schema/mysql/GWProfilerDB.sql
to any directory on your machine and invoke mysql in that directory:
> mysqladmin create GWProfilerDB -- this will create the GWProfilerDB database.
> mysql GWProfilerDB < create-db.sql
> mysql GWProfilerDB < GWProfilerDB.sql

Foundation

Start foundation:

/etc/init.d/gwservices start

J2EE Container

If you installed the war file in another container you must start the container, which must be configured to start on another port than 8080 (like port 8081) because of conflicts with foundation.

Jetty for example allows you to change the port in

jetty/etc/jetty.xml

by adding the port at:

```
<Call name="addConnector">
    <Arg>
        <New class="org.mortbay.jetty.nio.SelectChannelConnector">
            <Set name="host"><SystemProperty name="jetty.host" /></Set>
            <Set name="port"><SystemProperty name="jetty.port" default="8081"/></Set>
            .....
    </Arg>
</Call>
```

Web Browser

Access the tool at:

http://localhost:8080/groundwork-profiler/

or

http://localhost:8081/groundwork-profiler/

depending where you installed the war file (foundation or another j2ee container).

Command Line

The Profiler Tool can be invoked also from the command line.
After untaring the Profiler.tar.gz invoke:

```
profiler-java/deploy/foundation-profiler.sh -config configName.xml -session <name of session>
```

where configName.xml is a test configuration file, configName being assigned by the tester.

By invoking the profiler shell without any parameters:

profiler-java/deploy/foundation-profiler.sh

it defaults into:

profiler-java/deploy/foundation-profiler.sh -config foundation.xml

where foundation.xml comes together with the installation, as a model.

Using the Profiler Tool

1. Using the Profiler Web Application to create workloads

After starting foundation or any other container it is invoked at the url:
http://localhost:8080/groundwork-profiler/

The main page describes the first steps to do. There are four links:
a) About
b) SystemAdmin
c) Help SystemAdmin
d) Foundation 1.5
e) Help Foundation 1.5

The databases and foundation can be installed on different machines.
SystemAdmin is a simplified version of Foundation 1.5 and relies only on the four SystemAdmin message types :

SystemAdminInitMessage	- initializes the system
SystemAdminToggleHostStatusMessage	- host checks
SystemAdminServiceStatusMessage	- service checks
SystemAdminLogMessage	- log messages

SystemAdmin and Foundation 1.5 allows the user to configure:
A. Test System
a) url of the Profiler database
b) create a new configuration by specifying a filename (filetype must be xml) and clicking on the save button.
c) choose an existing configuration to be executed/deleted/changed
d) clean the database
e) run reports
f) also the user is allowed to run the tests in three differend modes:
ALL – Metrics are captured and written to both the log and database
LOG – Metrics are captured and written to the log only
OFF – NO metrics are captured

B. Target Groundwork Server
a) url of the Foundation database
b) JMS socket (4913)
c) address (IP Address or domain name) of foundation

C. Workloads
The difference between SystemAdmin and Foundation 1.5 pages are due to the fact that SystemAdmin provides a simplified version. Instead of going through the detail of build workload, it ask the user only following information:

Initialize the system with following

Number of Hosts:	Toggle the Service Status in percentage:	Interval between batches:
Number of Services:	Toggle the Host Status in percentage:	Number of batches:
Threshold:	Specify in percentage the consolidation rate:	

On the other side, Foundation 1.5 allows the user to build from scratch workloads.

A workload can be a single message or a collection of messages.
Workloads is a set of one or several messages.
A workload can run once (numBatches=1) or multiple times (numBatch=n)
The time between two consecutive runs is called interval.

2. Run tests and create Reports

The simplest mode to create tests is by invoking SystemAdmin .
This allows to configure a basic SystemAdmin test which should be saved and thereafter executed. After each test execution the Profiler tool allows you to see a report, which can also be invoked from the SystemAdmin or from Foundation 1.5 pages by pressing the Report button.

Tests run in batches. A batch is made of one or more workloads. A workload can be one or more messages. The user can specify the number of batches, and the interval between batches.

The user specifies the number of Hosts, the number of Services, the precentage of time the hosts or services are up or down. Also, the case of consolidation/no consolidation can be specified.

3. Using batch mode to run an existing workload.

The installation comes with a default configuration file, at:

profiler-java/deploy/foundation.xml

Start the test from the command line:

profiler-java/deploy/foundation-profiler.sh -config configName.xml -session <name of session>

The output can be seen in the console or in the log4j files.

4. Tips and tricks for creating workloads.

Originally the code had following message types:

ToggleServiceStatusMessage
ToggleHostStatusMessage
HttpRequest
HostCreateMessage
SysLogMessage
SNMPMessage
EventMessage

We added the SystemAdmin message types:

SystemAdminInitMessage	- initializes the system
SystemAdminToggleHostStatusMessage	- host checks
SystemAdminServiceStatusMessage	- service checks
SystemAdminLogMessage	- log messages

which correspond to the intial messages but in the SystemAdmin format:

SystemAdminInitMessage	-	HostCreateMessage
SystemAdminToggleHostStatusMessage	-	ToggleHostStatusMessage
SystemAdminServiceStatusMessage	-	ToggleServiceStatusMessage
SystemAdminLogMessage	-	EventMessage

The purpose of the tool is to simulate a real situation. Depending on the simulation needs various parameters must be changed. In the workloads section it was described how to configure the workloads. Messages vary be number of hosts/devices/services and the percentage in changes of UP/DOWN state of the host/service. Also, in the case of log messages there is the option to configure the consolidation percentage 0% = no consolidation, 100% = all messages are consolidated.