Teligent
Pre-study Report

SMART
SEGv2
SMART
SEGv2 pre-study report

Table Of Content

1 Introduction ...............................................................................................................3
  1.2 Referenced documents ..........................................................................................3
  1.3 Definition and acronyms .......................................................................................3
  1.4 Details ....................................................................................................................3
2 Revision history ........................................................................................................4
3 Test Environment Overview .....................................................................................5
  3.1 Hardware...............................................................................................................5
  3.2 Software ...............................................................................................................5
4 Test Setup Overview .................................................................................................6
5 Test Case ...................................................................................................................7
  5.1 Unmonitored Call..................................................................................................7
6 Test Results ...............................................................................................................8
  6.1 Test 1: SEGv1.....................................................................................................8
  6.2 Test 2: SEGv2.....................................................................................................8
7 Conclusion .................................................................................................................8
1 Introduction

The SEG component is the interface between TeliGent’s Service Capability Server (SCS) and Smart’s Service Engines (SE). A first version of the SEG (SEGv1) was created by Smart. However, since the SEG component will run in TeliGent’s SCS it was considered a good idea that TeliGent become responsible for it. TeliGent made an assessment of SEGv1 which showed that it would be better to make a new SEG (SEGv2). Reasons for this decision were:
1. The initial version of the SEG had some performance and stability problems.
2. SEGv1 was not implemented according to TeliGent conventions.
3. Transactions were not delivered in FIFO order.

This report will present the test setup, test results and suggest how to proceed.

1.2 Referenced documents


1.3 Definition and acronyms

| CEVN | Call Event Notify |
| CPS  | Calls per second  |
| CPU  | Central Processor Unit |
| CTC  | CEVN Test Component |
| FIFO | First In First Out |
| RREQ | Route Request |
| SCS  | Service Capability Servers |
| SE   | Service Engine |
| SE-Sim | Service Engine Simulator |
| SEGv1 | Service Engine Gateway version 1.6.0 implemented by SMART |
| SEGv2 | The new version of Service Engine Gateway implemented by TeliGent |
| SEP  | Service Engine P90 Protocol |
| TM   | Transaction Manager in P90/E platform |

1.4 Details

All tests conducted by Erik Waling.
SMART
SEGv2 pre-study report

<table>
<thead>
<tr>
<th>Id</th>
<th>Document Responsible</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id-17779</td>
<td>Per-Olof Åström</td>
<td>2007-01-26</td>
</tr>
<tr>
<td>Revision</td>
<td>Author/Prepared</td>
<td></td>
</tr>
<tr>
<td>A1</td>
<td>Erik Waling/Tommy Öström</td>
<td>4 of (8)</td>
</tr>
</tbody>
</table>

2 Revision history

A1. Initial version.
3 Test Environment Overview

The available Spectra were not able to produce enough load to stress test SEGv2. We decided to write a simple test component instead (CTC).

During the tests we could see that the available Smart SE simulator sometimes changed the order of output to the SEG. We decided to create our own SE simulator.

Two hosts were used to simulate the setup of the production environment, SCS on one host and SE simulator on the other host.

This section specifies the test environment in which the tests were conducted. It should provide a detailed overview of the test equipment and provide repeatability to the tests.

3.1 Hardware

<table>
<thead>
<tr>
<th>Item</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>SE simulator Host</td>
<td>SUN X4100, see [X4100] for details.</td>
</tr>
<tr>
<td></td>
<td>Two dual-core AMD Opteron 200 2.6GHz processors.</td>
</tr>
<tr>
<td></td>
<td>4GB Memory</td>
</tr>
<tr>
<td>SCS Host</td>
<td>SUN X4100, see [X4100] for details.</td>
</tr>
<tr>
<td></td>
<td>Two dual-core AMD Opteron 200 2.6GHz processors.</td>
</tr>
<tr>
<td></td>
<td>4GB Memory</td>
</tr>
</tbody>
</table>

3.2 Software

<table>
<thead>
<tr>
<th>ITEM</th>
<th>VERSION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating system</td>
<td>RHEL4 Update 3, for SCS hosts.</td>
</tr>
<tr>
<td>P90/E kernel</td>
<td>p90e_kernel-1.6.3RC1</td>
</tr>
</tbody>
</table>
4 Test Setup Overview

The CTC, SE-simulator and SEGv2 are all single-threaded and written in C to minimize the overhead.

For the pre-study only limited functionality was implemented in SEGv2:

- The code needs to be restructured; although this will not notably affect the performance.
- Only one SE-simulator is allowed to connect. i.e. no functionality for load sharing and failover was implemented.
- No fault handling is implemented.
- Some parts that are now hard coded should be made configurable.

The CTC component spreads the calls evenly over time. No bursts are produced.

Figure 1 Test setup
5 Test Case

The “Unmonitored call” test case was selected since it is the least complicated case. This is enough to measure the message transport.

On the SCS host the CTC component sends a CEVN message to the TM which forwards the transaction to the SEG. The SEG component transforms the message into the SEP protocol and forwards it to the SE simulator, see Figure 2.

The CTC measures the time it takes from that a CEVN message is sent until the corresponding RREQ message is received, step 1 through 9 in Figure 2.

5.1 Unmonitored Call

Figure 2 Unmonitored call sequence
6 Test Results
The aim of the tests was to find the maximum load where the system was stable.

6.1 Test 1: SEGv1
Number of calls: 1,000,000.
Maximum load: 4,000 CPS at 100% CPU load.
Maximum load: 1,500 CPS at 68% CPU load.

6.2 Test 2: SEGv2
Number of calls: 1,000,000.
Maximum load: 8,200 CPS at 35% CPU load.

7 Conclusion
We recommend that the maximum component CPU usage should not exceed 70%. Exceeding this limit jeopardizes the stability of the platform.

SEGv2 handled 447% more calls than SEGv1 at recommended CPU load.
SEGv1 did not maintain FIFO order which is unacceptable.
SEGv1 crashed during high load.

We estimate that SEGv2 will have five times less code than SEGv1. SEGv2 is single threaded while SEGv1 is multi-threaded. All-in-all the SEGv2 will be much easier to understand and maintain.
We suggest that the development of SEGv2 continue so that SEGv1 can be replaced.