OCSS7 Benchmarks

TAS-073-Issue 1.1.0-Release 1

February 2018

metaswitch

Notices

Contents

1 OCSS7 Benchmarks

This book contains performance benchmarks using OpenCloud's OCSS7.

It contains these sections:

- Benchmark Scenarios on page 6 — descriptions of each of the benchmark scenarios, and notes on the benchmark methodology used
- Hardware and Software on page 11 — details of the hardware, software, and configuration used for the benchmarks
- Benchmark Results on page 13 — summaries of the benchmarks and links to detailed metrics.

Other documentation for OCSS7 can be found on the OCSS7 product page .

2 Benchmark Scenarios

This page describes the scenarios and methodology used when running the benchmarks.

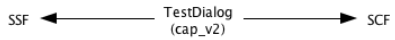
Benchmarks are run using two scenarios. Each scenario is run with the OCSS7 acting as the initiator and as the responder.

2.1 CAP call monitoring scenario

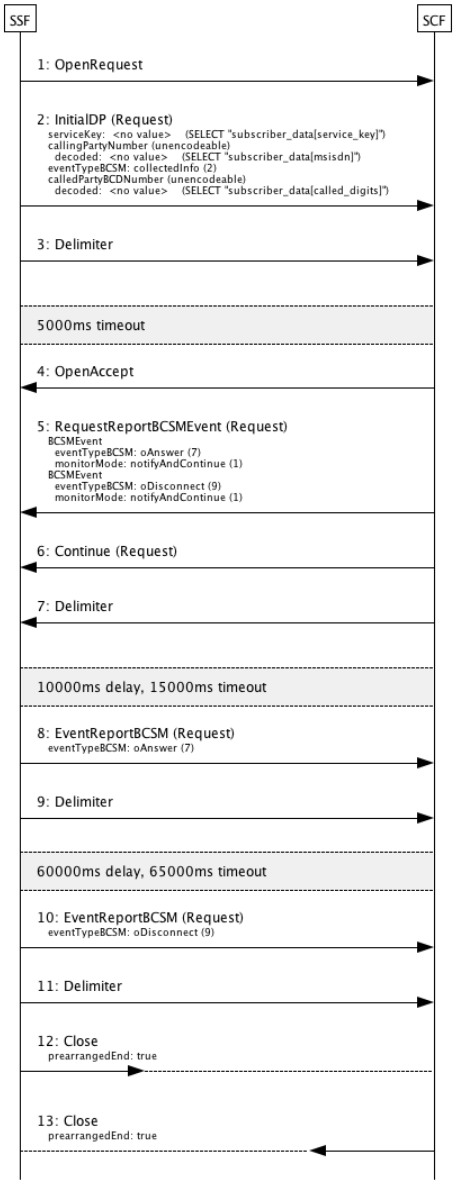
These scenarios consist of dialogs following the message flow:

- The initiator sends a TC-BEGIN initiating a CAP v2 dialog, containing an CAP InitialDP operation invoke
- The responder sends a TC-CONTINUE containing two operation invokes: CAP RequestReportBCSM(oAnswer,oDisconnect), and CAP Continue
- The initiator delays for 10 seconds
- The initiator sends a TC-CONTINUE containing a CAP EventReportBCSM(oAnswer) operation invoke
- The initiator delays for 60 seconds
- The initiator sends a TC-CONTINUE containing a CAP EventReportBCSM(oDisconnect) operation invoke
- The dialog is ended with prearranged end on both the initiator and responder

Dialogs



Message flows



2.1.1 OCSS7 as the Initiator

In this configuration, the test system generates load, and is the initiator of the dialog. The responder is the external support system.

Response time is measured on the test system by the scenario simulator infrastructure, from immediately before submission of the TC-BEGIN to the TCAP stack, to immediately after receipt of the TC-CONTINUE containing the CAP Continue invoke from the TCAP stack.

2.1.2 OCSS7 as the Responder

In this configuration, the support system generates load, and is the initiator of the dialog. The test system responds to the dialog.

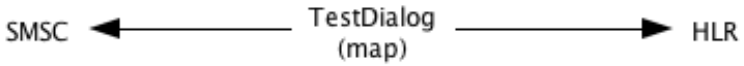
Response time is measured on the support system by the scenario simulator infrastructure, from immediately before submission of the TC-BEGIN to the TCAP stack, to immediately after receipt of the TC-CONTINUE containing the CAP Continue invoke from the TCAP stack.

2.2 MAP SRI for SM scenario

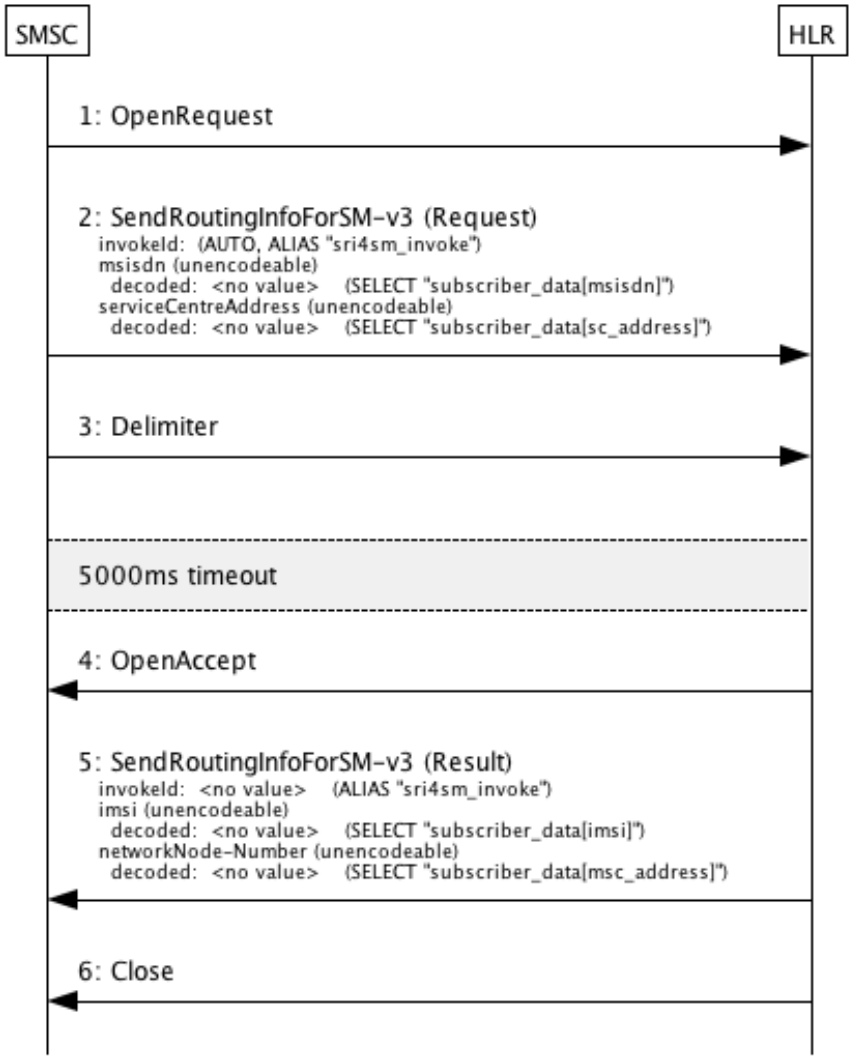
These scenarios consist of dialogs following the message flow:

- The initiator sends a TC-BEGIN initiating a MAP dialog, containing SEND-ROUTING- INFO-FOR-SM invoke
- The responder immediately responds with a TC-END containing a SEND-ROUTING- INFO-FOR-SM result

Dialogs



Message flows



2.2.1 OCSS7 as the Initiator

In this configuration, the test system generates load, and is the initiator of the dialog. The responder is the external support system.

Response time is measured on the test system by the scenario simulator infrastructure, from immediately before submission of the TC-BEGIN to the TCAP stack, to immediately after receipt of the TC-END from the TCAP stack.

2.2.2 OCSS7 as the Responder

In this configuration, the support system generates load, and is the initiator of the dialog. The test system responds to the dialog, sending a TC-END.

Response time is measured on the support system by the scenario simulator infrastructure, from immediately before submission of the TC-BEGIN to the TCAP stack, to immediately after receipt of the TC-END from the TCAP stack.

2.3 Test setup

Each test run consists of a 10 minute ramp-up period where load is increased from zero to the target rate; then a 30 minute measurement period at peak load; then a 2 minute drain period during which no new dialogs are initiated.

The ramp-up period is included as the Oracle JVM provides a Just In Time (JIT) compiler. The JIT compiler compiles Java bytecode to machinecode, and recompiles code on the fly to take advantage of optimizations not otherwise possible. This dynamic compilation and optimization process takes some time to complete. During the early stages of JIT compilation/optimization, the node cannot process full load. JVM garbage collection does not reach full efficiency until several major garbage collection cycles have completed.

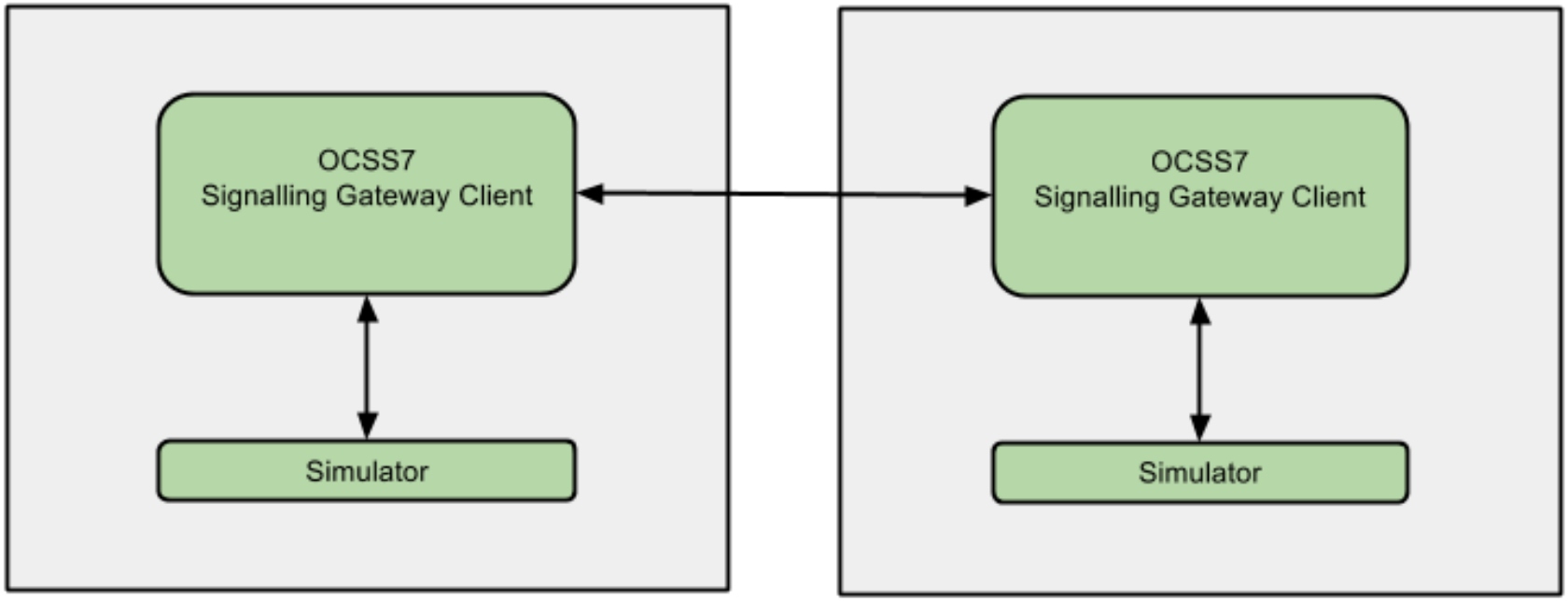
Only latency measurements during the measurement period are used; latency measurements during the ramp-up period are ignored.

Load is not stopped between ramp up and starting the test timer.

3 Hardware and Software

This page describes the hardware and software used when running the benchmarks.

3.1 Hardware



HP ProLiant DL380 G7
2 x 2.8GHz (12 cores, 24 threads) Xeon X5650
24G RAM
CentOS 6.4

HP ProLiant DL380 G7
2 x 2.8GHz (12 cores, 24 threads) Xeon X5650
24G RAM
CentOS 6.4

3.2 Software

Vendor	Software	Version
Oracle	Java	JDK 1.7.0_45 (64 bit)
OpenCloud	OpenCloud OCSS7	1.0.1.8
OpenCloud	Scenario Simulator	2.3.0.5
OpenCloud	IN Scenario Pack	1.5.2.8

3.3 Configuration

Parameter	Value
Heap size	2048M
com.cts.ss7.commsp.server.sendQueueCapacity	1024
sgc.ind.pool_size	20000
sgc.req.pool_size	20000
sgc.worker.threads	32
Parameter	**Value**
net.sctp.rto_min	80
net.core.rmem_max	2000000
net.core.wmem_max	2000000

4 Benchmark Results

A summary of the benchmark results follows. Click on a benchmark name for detailed results.

<table>
<tr><th>Benchmark</th><th>Rate</th><th>CPU Usage</th></tr>
<tr><td>CAP Call Monitoring scenario on page 14</td><td>6000 calls per second (24000 messages per second)</td><td>
<table>
<tr><th>Configuration</th><th>Across 12 cores</th></tr>
<tr><td>OCSS7 as Initiator</td><td>209.9%</td></tr>
<tr><td>OCSS7 as Responder</td><td>252.3%</td></tr>
</table>
</td></tr>
<tr><td>MAP SRI for SM scenario on page 24</td><td>20000 calls per second (40000 messages per second)</td><td>
<table>
<tr><th>Configuration</th><th>Across 12 cores</th></tr>
<tr><td>OCSS7 as Initiator</td><td>355.7%</td></tr>
<tr><td>OCSS7 as Responder</td><td>376.0%</td></tr>
</table>
</td></tr>
</table>

5 CAP Call Monitoring scenario

This page summarises the results for benchmarks executed with the **CAP Call Monitoring** scenario. Detailed metrics on page 16 follow the summary tables.

The benchmarks are run in two configurations:

1. OCSS7 as the Initiator
2. OCSS7 as the Responder

The CAP call monitoring scenario description on page 6 has more information on how these configurations are defined.

Identical call rates was used for both Initiator and Responder benchmarks

5.1 Benchmarks

<table>
<tr><td>Call rate on page 16</td><td>6000 calls per second (24000 messages per second)</td></tr>
<tr><td>CPU usage on page 18</td><td>
<table>
<tr><th>Configuration</th><th>Across 12 cores</th></tr>
<tr><td>OCSS7 as Initiator</td><td>209.9%</td></tr>
<tr><td>OCSS7 as Responder</td><td>252.3%</td></tr>
</table>

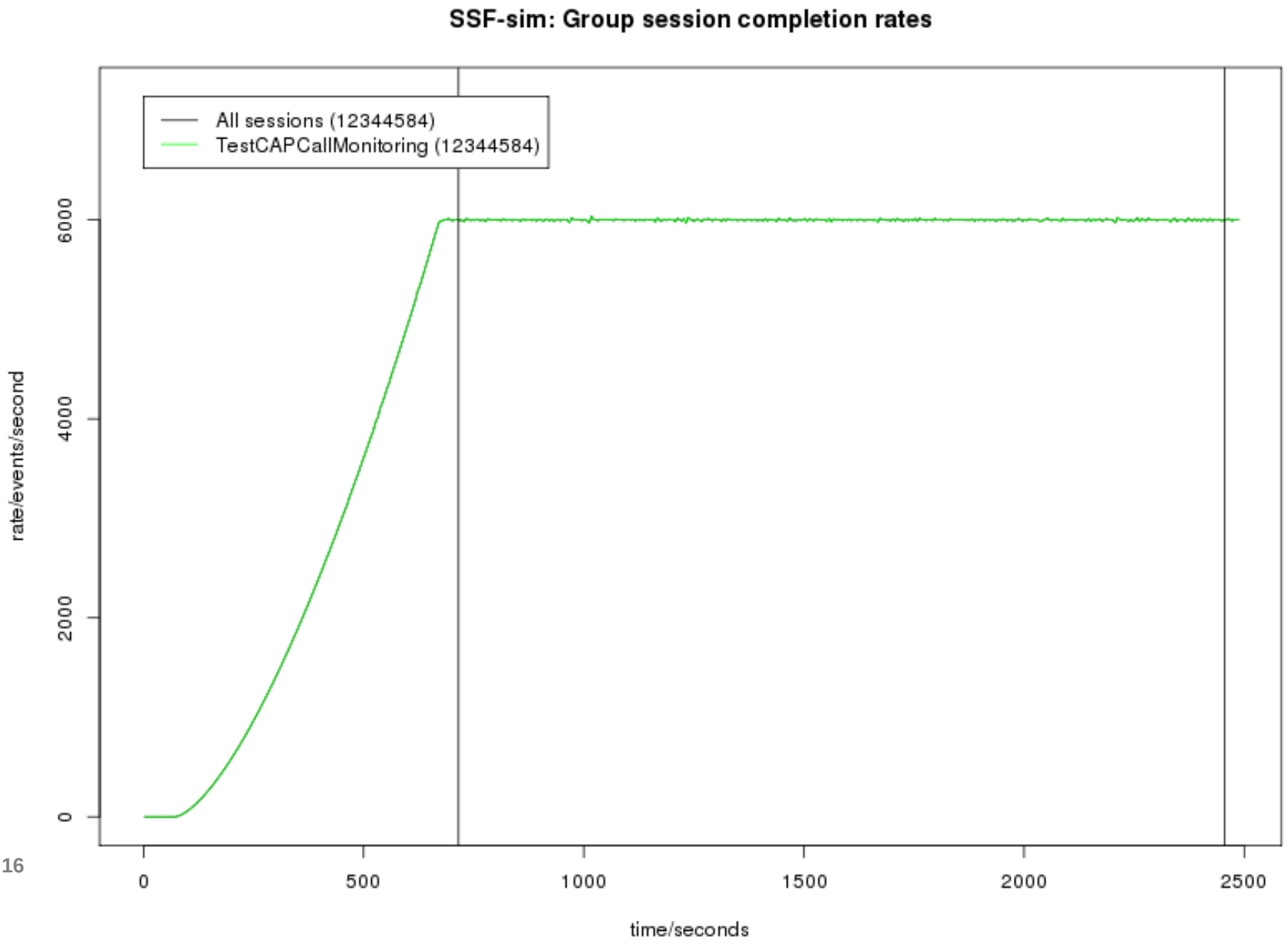
Maximum theoretical CPU usage is 1200%.
</td></tr>
<tr><td>Heap usage on page 20</td><td>
<table>
<tr><th>Configuration</th><th>Average heap</th></tr>
</table>
</td></tr>
</table>

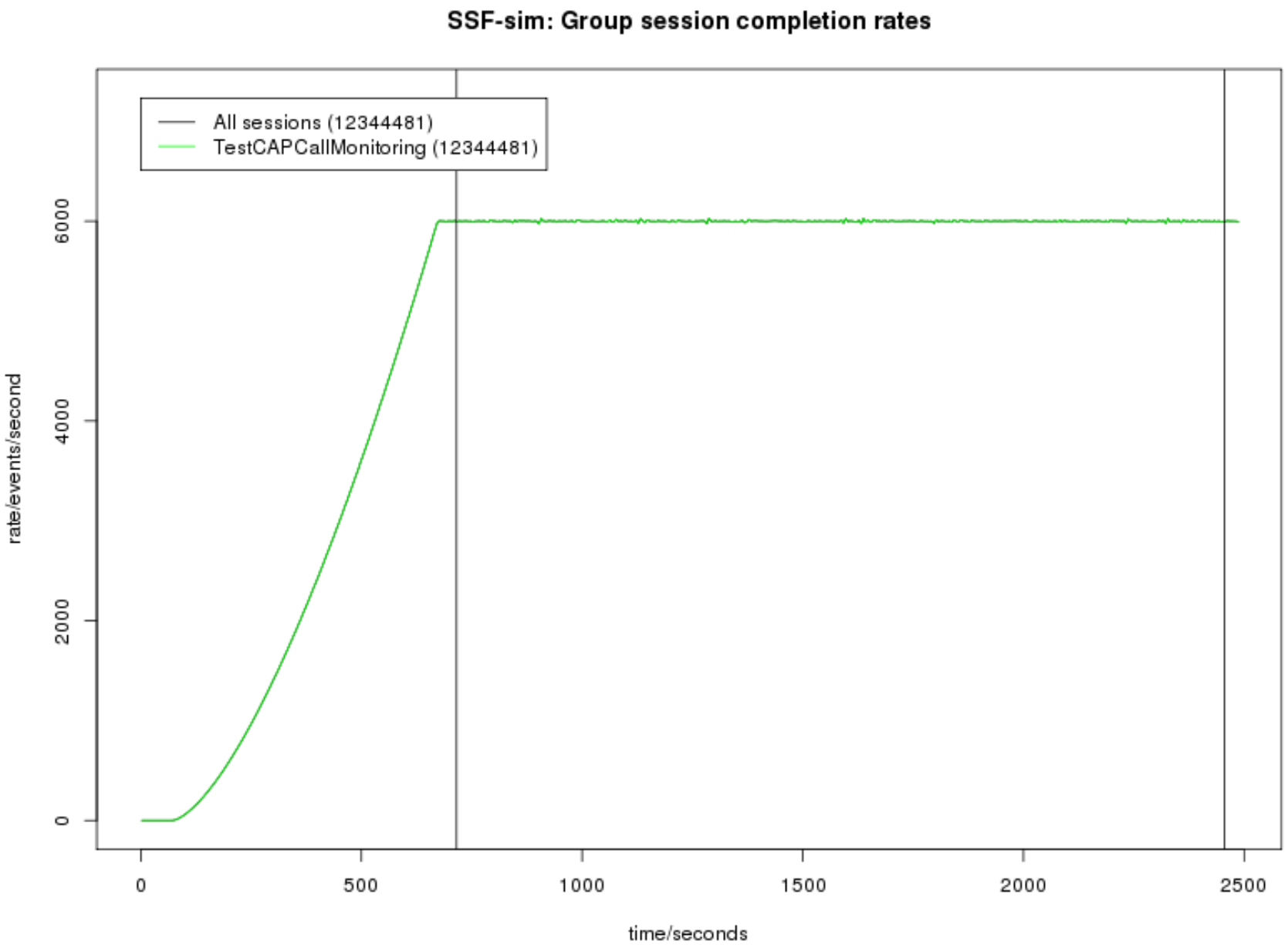
	OCSS7 as Initiator: 750MB OCSS7 as Responder: 1200MB
Scenario Latencies on page 22	

Configuration	50th	90th	95th	99th
OCSS7 as Initiator	1.6ms	6.5ms	11.0ms	31.16ms
OCSS7 as Responder	1.5ms	6.1ms	10.5ms	30.6ms

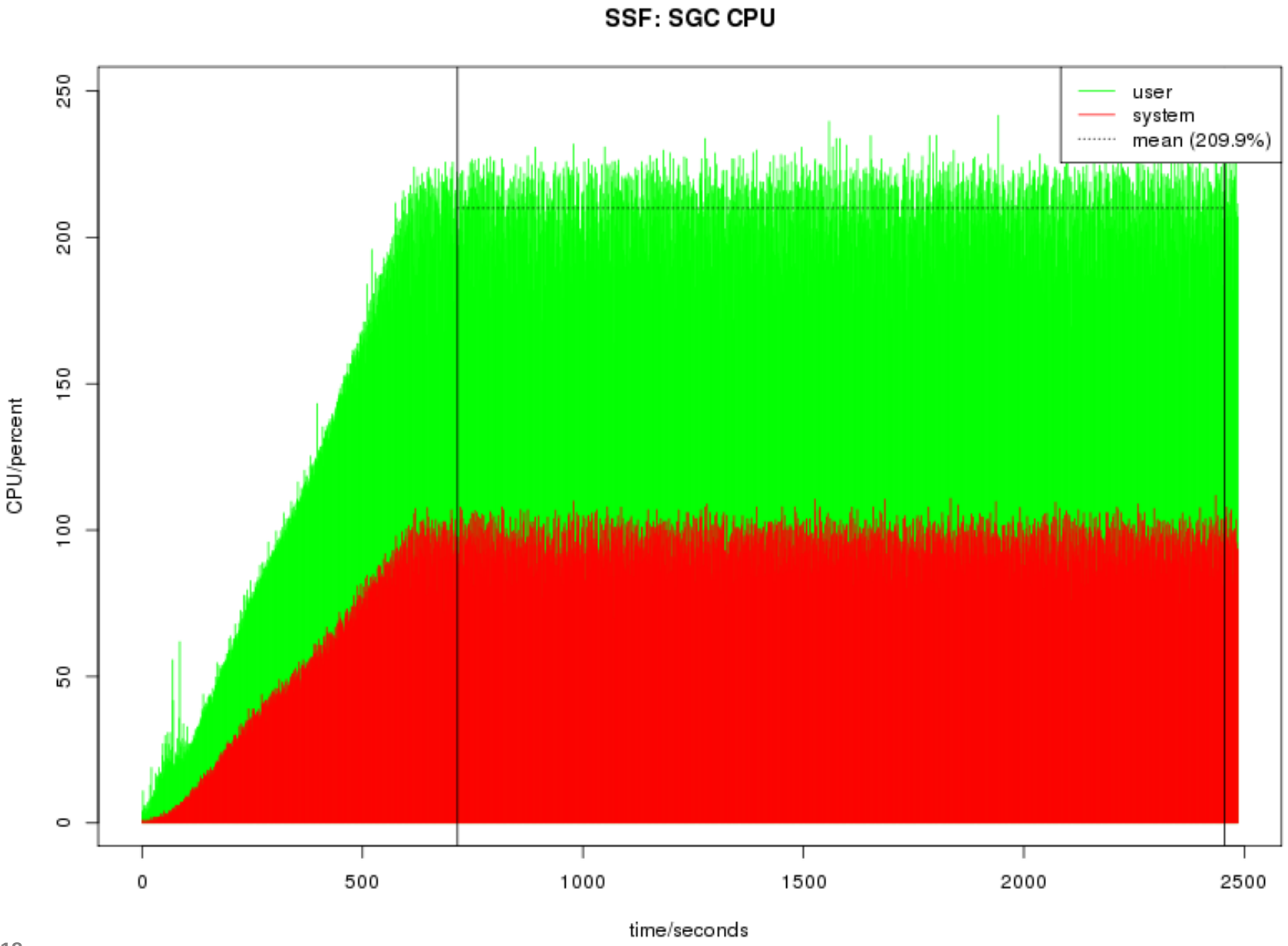
5.2 Detailed metrics

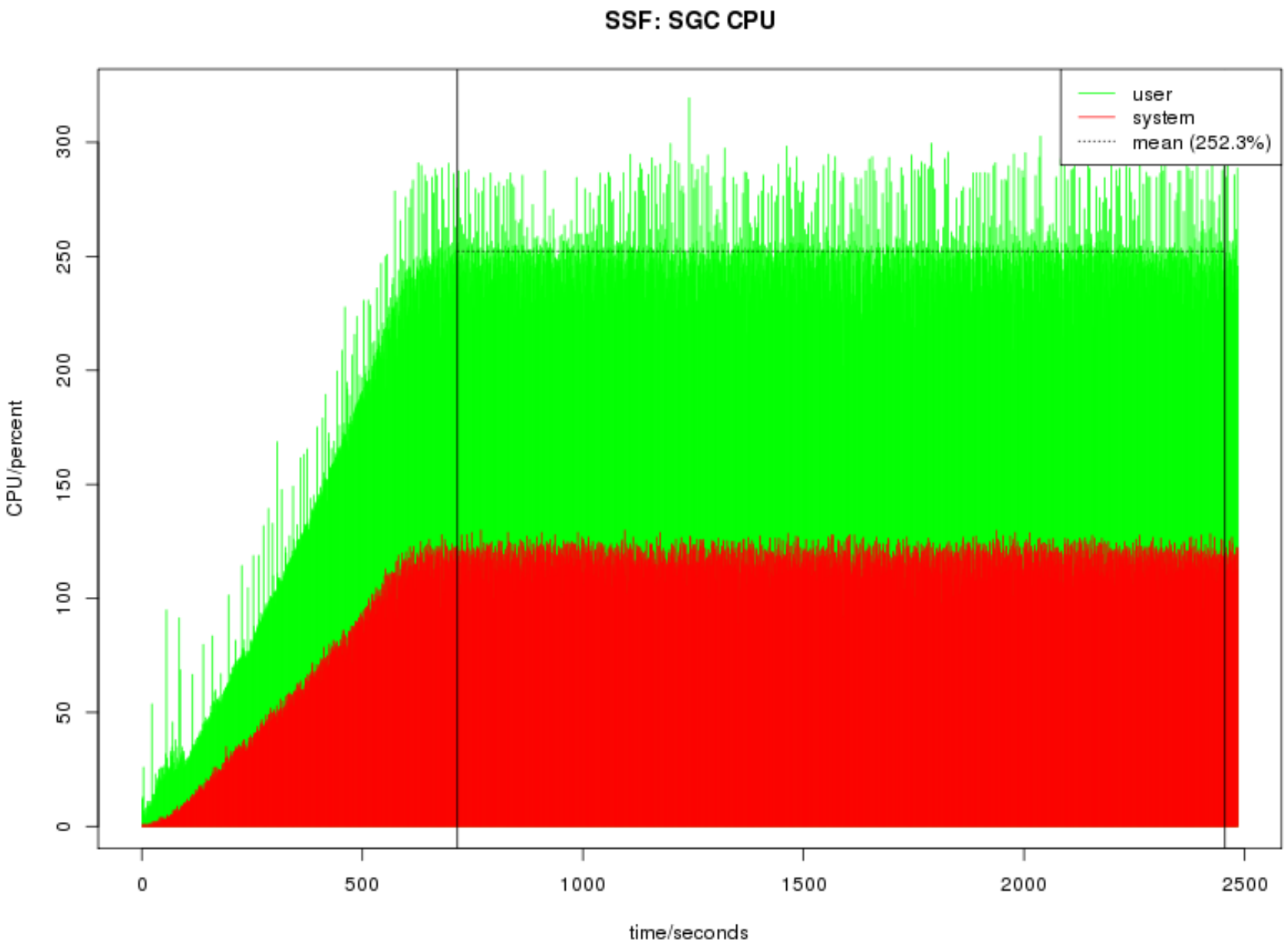
5.2.1 Call Rate



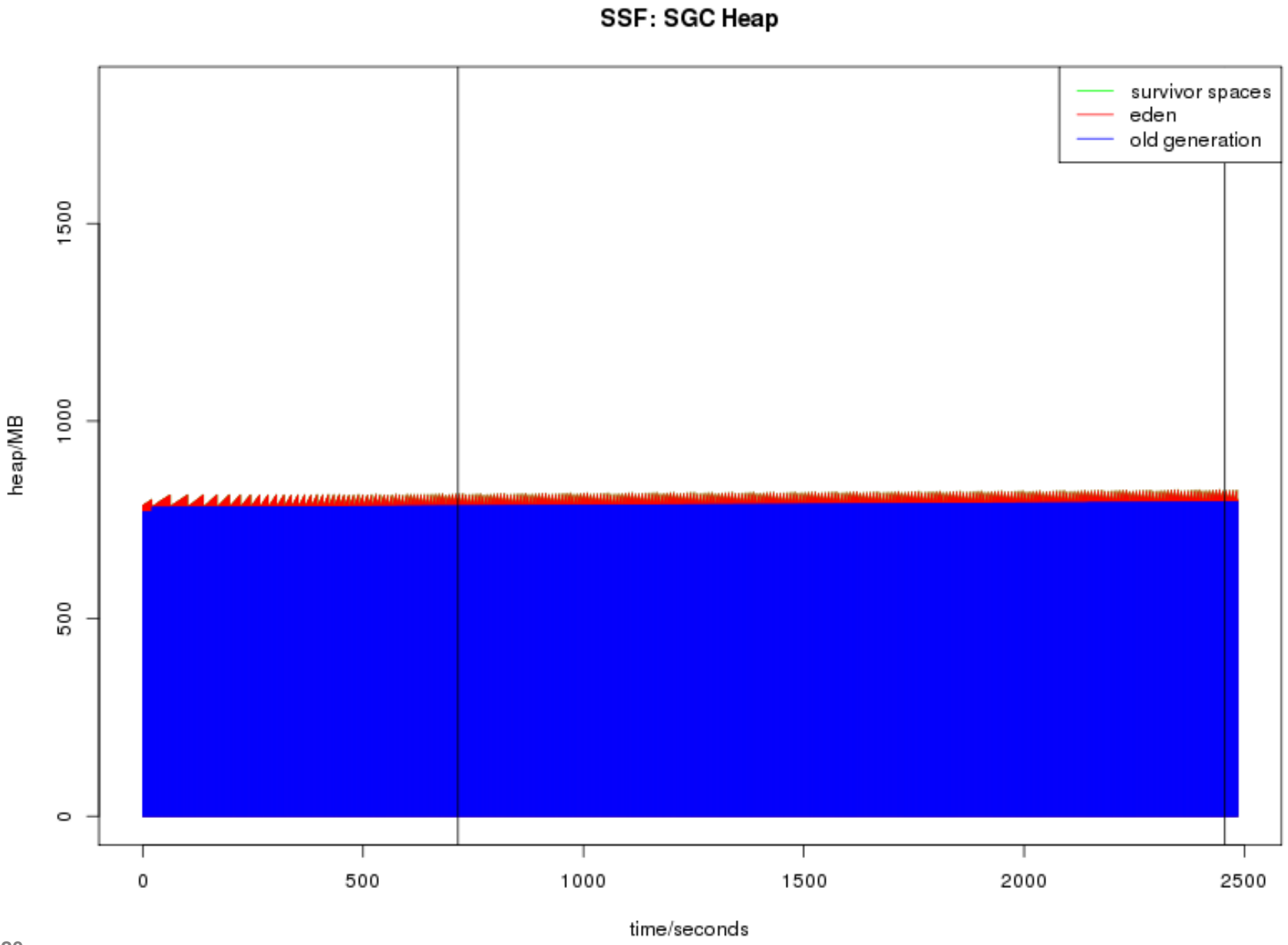

SSF-sim: Group session completion rates
All sessions (12344481)
TestCAPCallMonitoring (12344481)
rate/events/second
0
2000
4000
6000
0
500
1000
1500
2000
2500
time/seconds

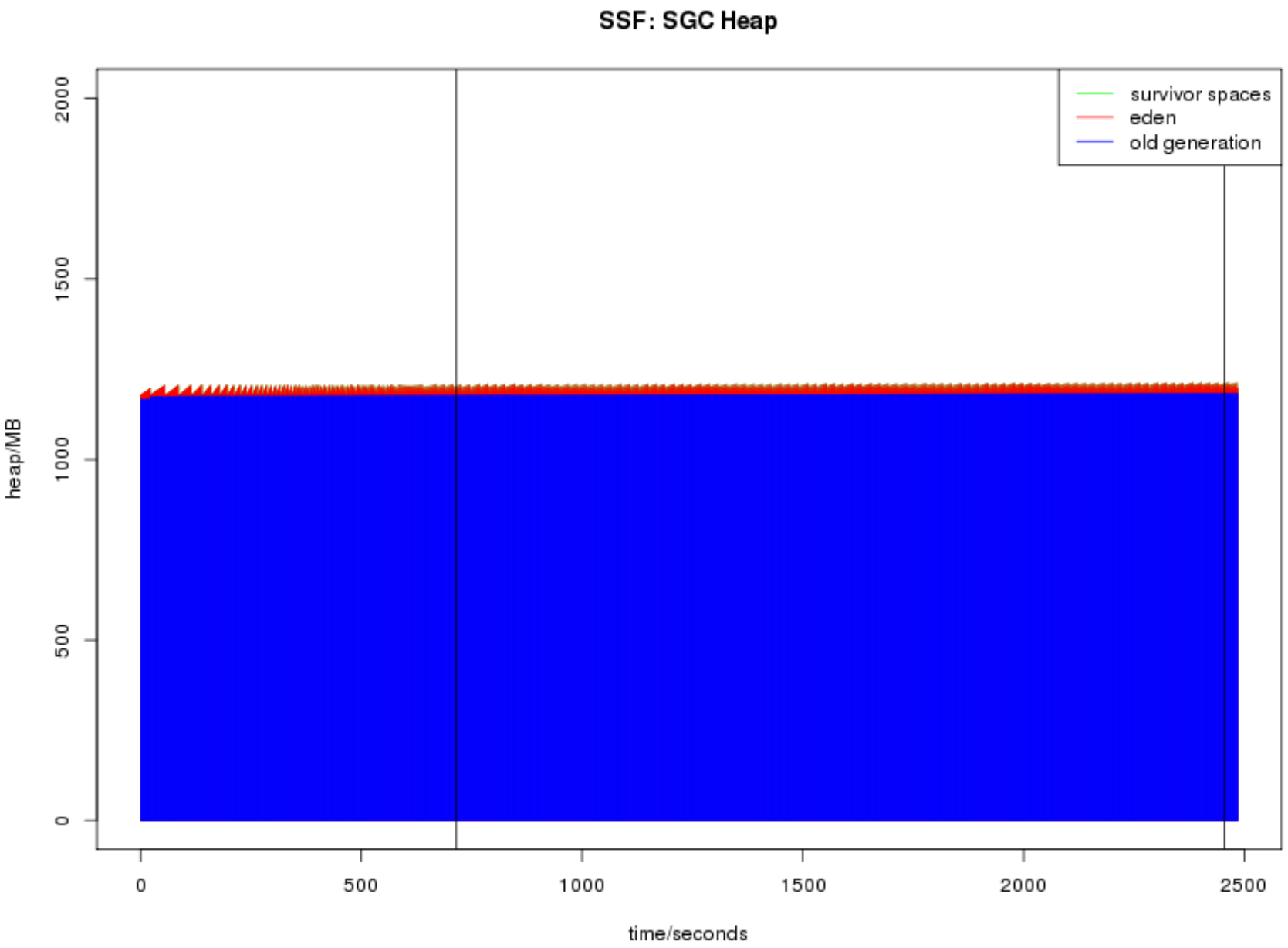
5.2.2 CPU usage



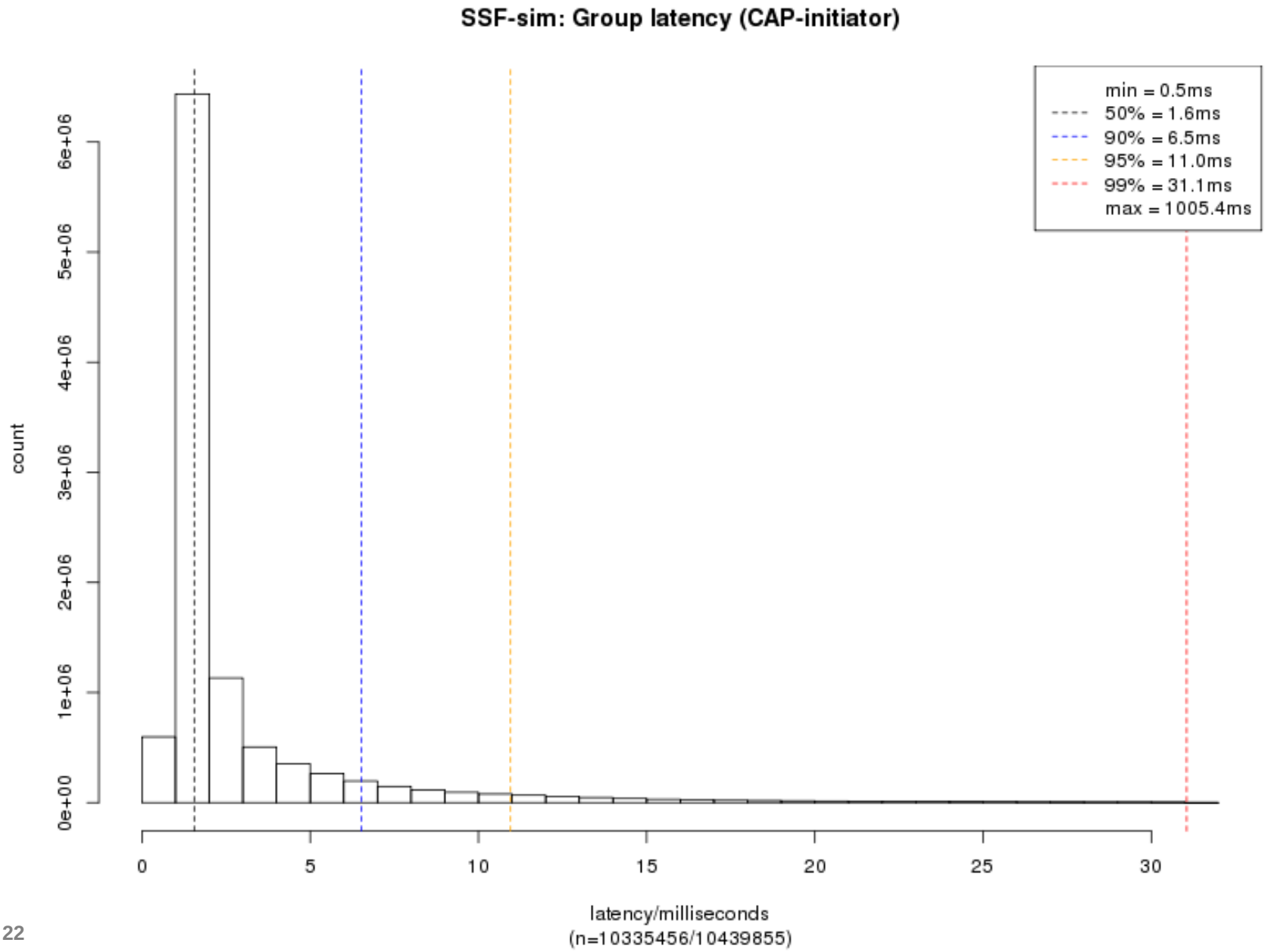

SSF: SGC CPU
user
system
mean (252.3%)
CPU/percent
0
50
100
150
200
250
300
0
500
1000
1500
2000
2500
time/seconds

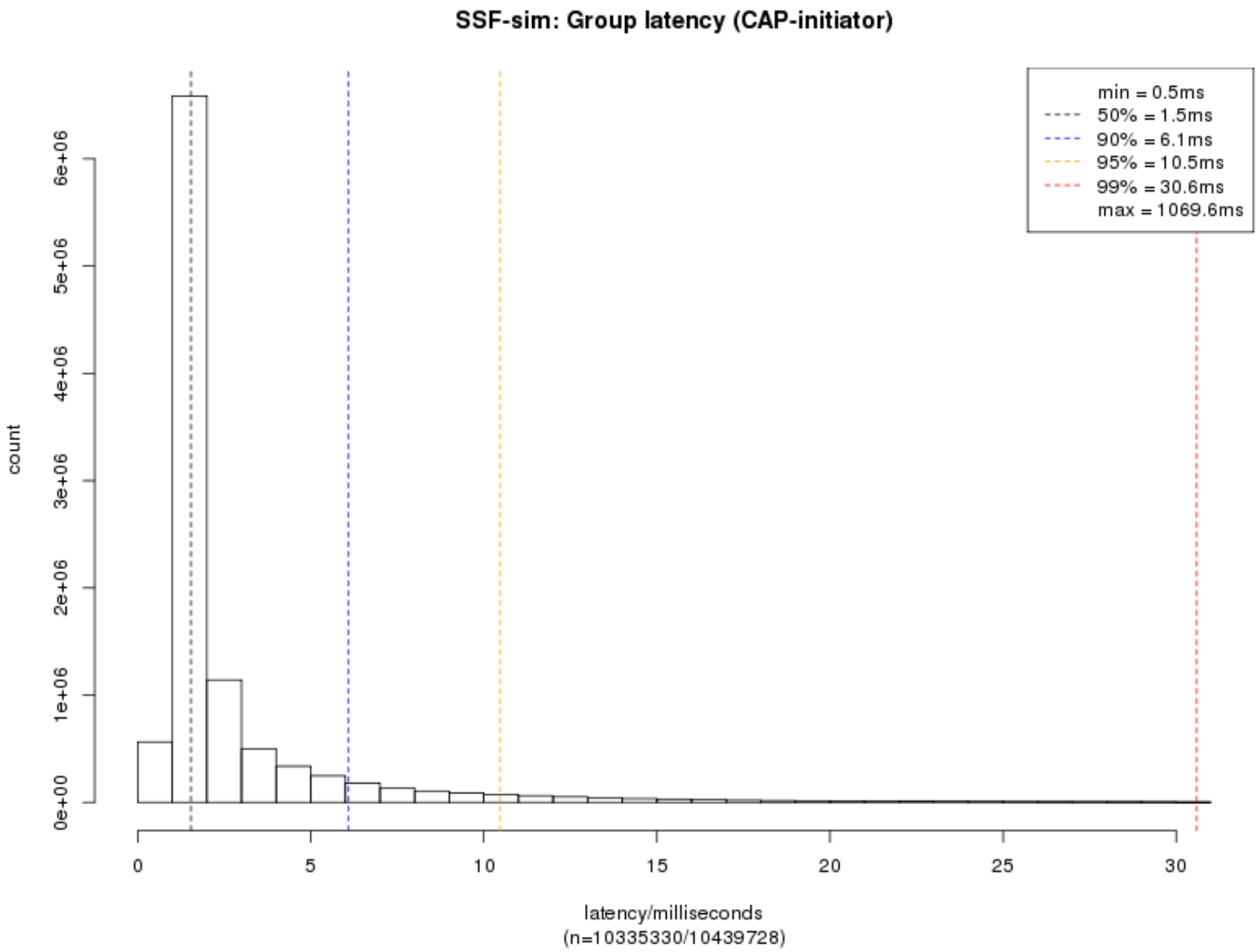
5.2.3 Heap usage




SSF: SGC Heap
survivor spaces
eden
old generation
heap/MB
0
500
1000
1500
2000
time/seconds
0
500
1000
1500
2000
2500

5.2.4 Scenario latencies




SSF-sim: Group latency (CAP-initiator)
min = 0.5ms
50% = 1.5ms
90% = 6.1ms
95% = 10.5ms
99% = 30.6ms
max = 1069.6ms
count
0e+00
1e+06
2e+06
3e+06
4e+06
5e+06
6e+06
0
5
10
15
20
25
30
latency/milliseconds
(n=10335330/10439728)

6 MAP SRI for SM scenario

This page summarises the results for benchmarks executed with the **MAP SRI for SM scenario** . Detailed metrics on page 26 follow the summary tables.

The benchmarks are run in two configurations:

1. OCSS7 as the Initiator
2. OCSS7 as the Responder

The MAP SRI for SM scenario description on page 8 has more information on how these configurations are defined.

Identical call rates was used for both Initiator and Responder benchmarks

6.1 Benchmarks

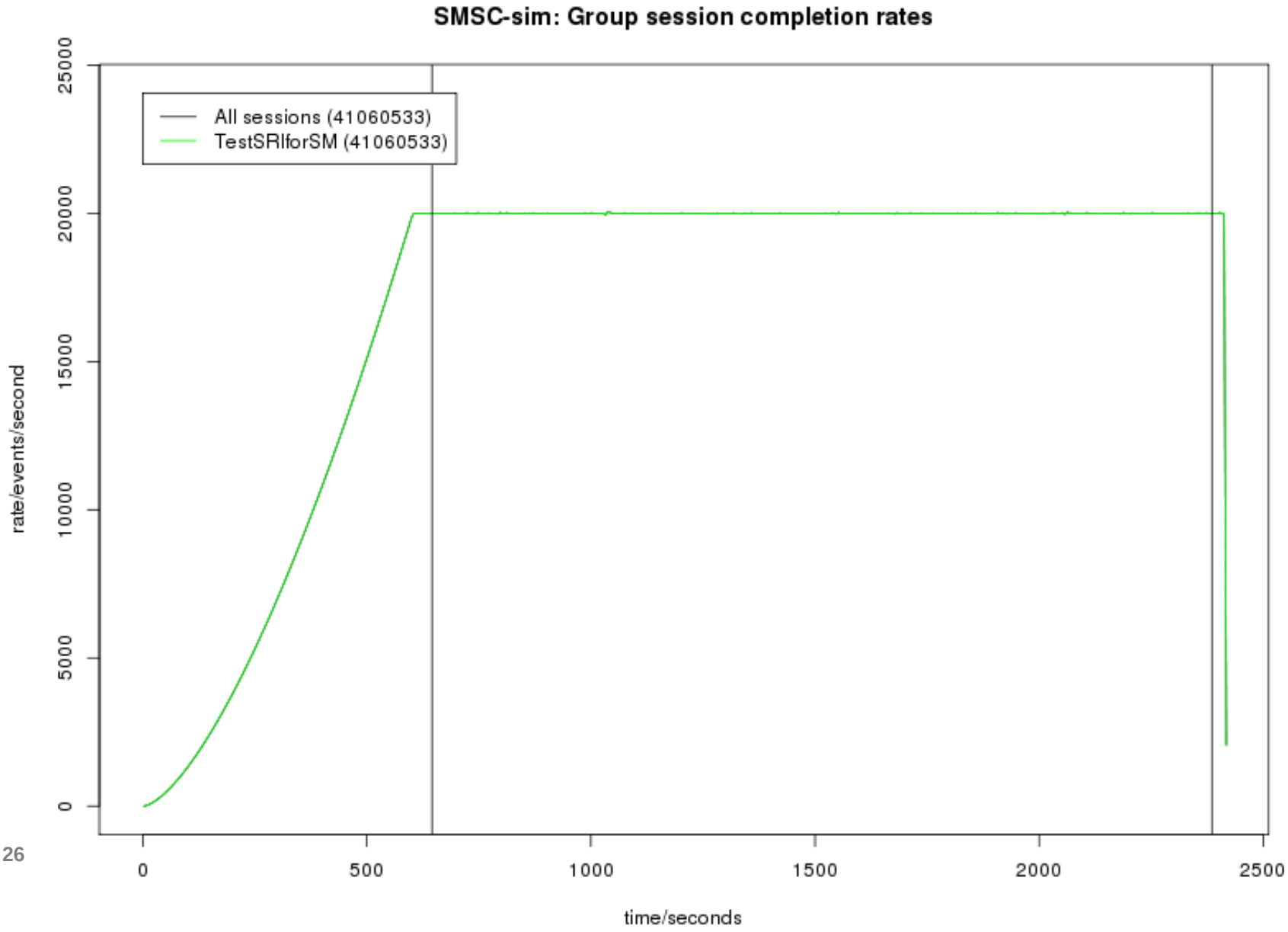
<table>
<tr><td>Call rate on page 26</td><td>20000 calls per second (40000 messages per second)</td></tr>
<tr><td>CPU usage on page 28</td><td>
<table>
<tr><th>Configuration</th><th>Across 12 cores</th></tr>
<tr><td>OCSS7 as Initiator</td><td>355.7%</td></tr>
<tr><td>OCSS7 as Responder</td><td>376.0%</td></tr>
</table>
Maximum theoretical CPU usage is 1200%
</td></tr>
<tr><td>Heap usage on page 30</td><td>
<table>
<tr><th>Configuration</th><th>Average heap</th></tr>
<tr><td>OCSS7 as Initiator</td><td>750MB</td></tr>
</table>
</td></tr>
</table>

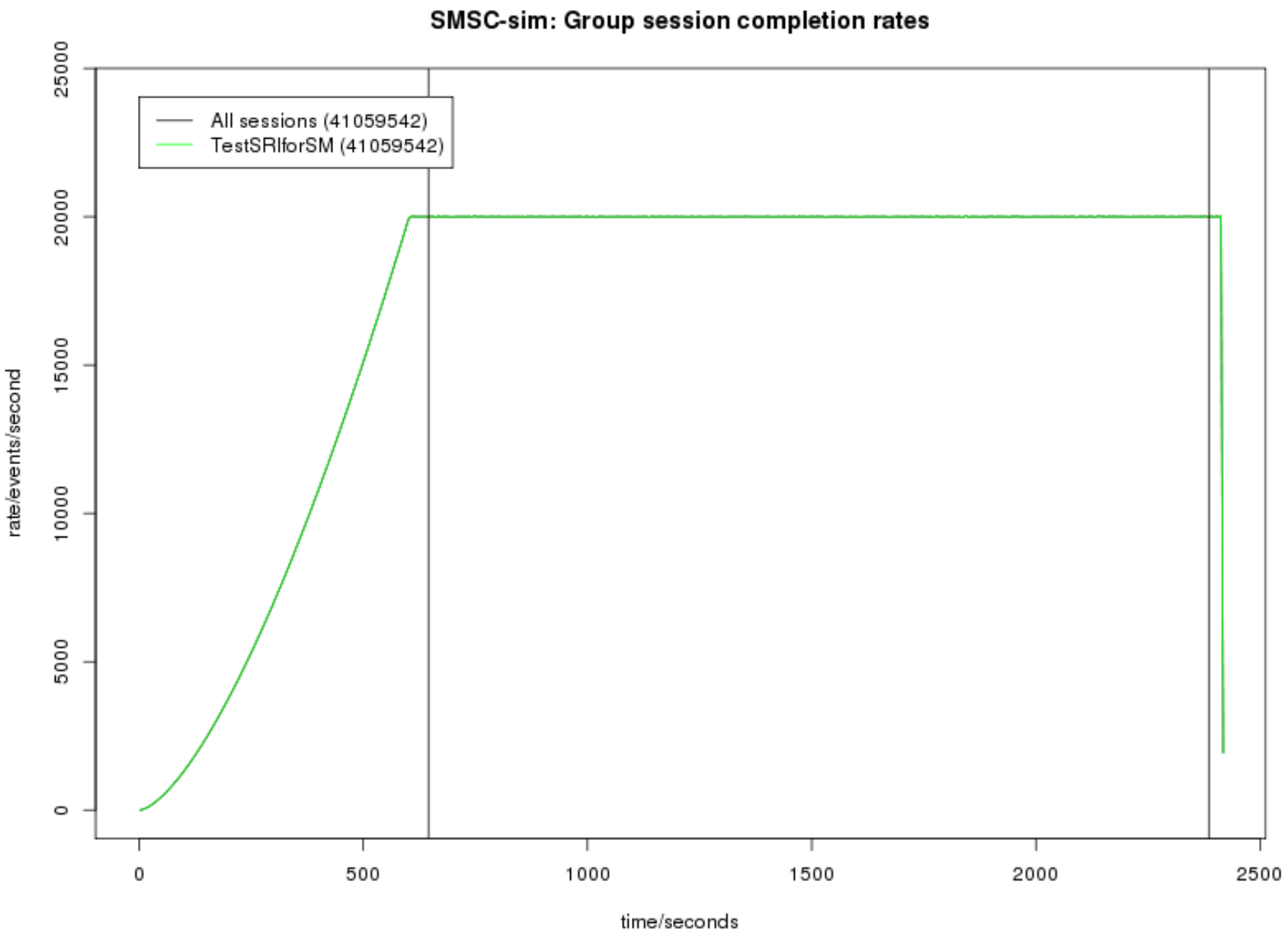
	OCSS7 as Responder: 750MB
Scenario Latencies on page 32	

Configuration	50th	90th	95th	99th
OCSS7 as Initiator	1.6ms	2.2ms	2.7ms	6.9ms
OCSS7 as Responder	1.7ms	2.2ms	2.8ms	7.1ms

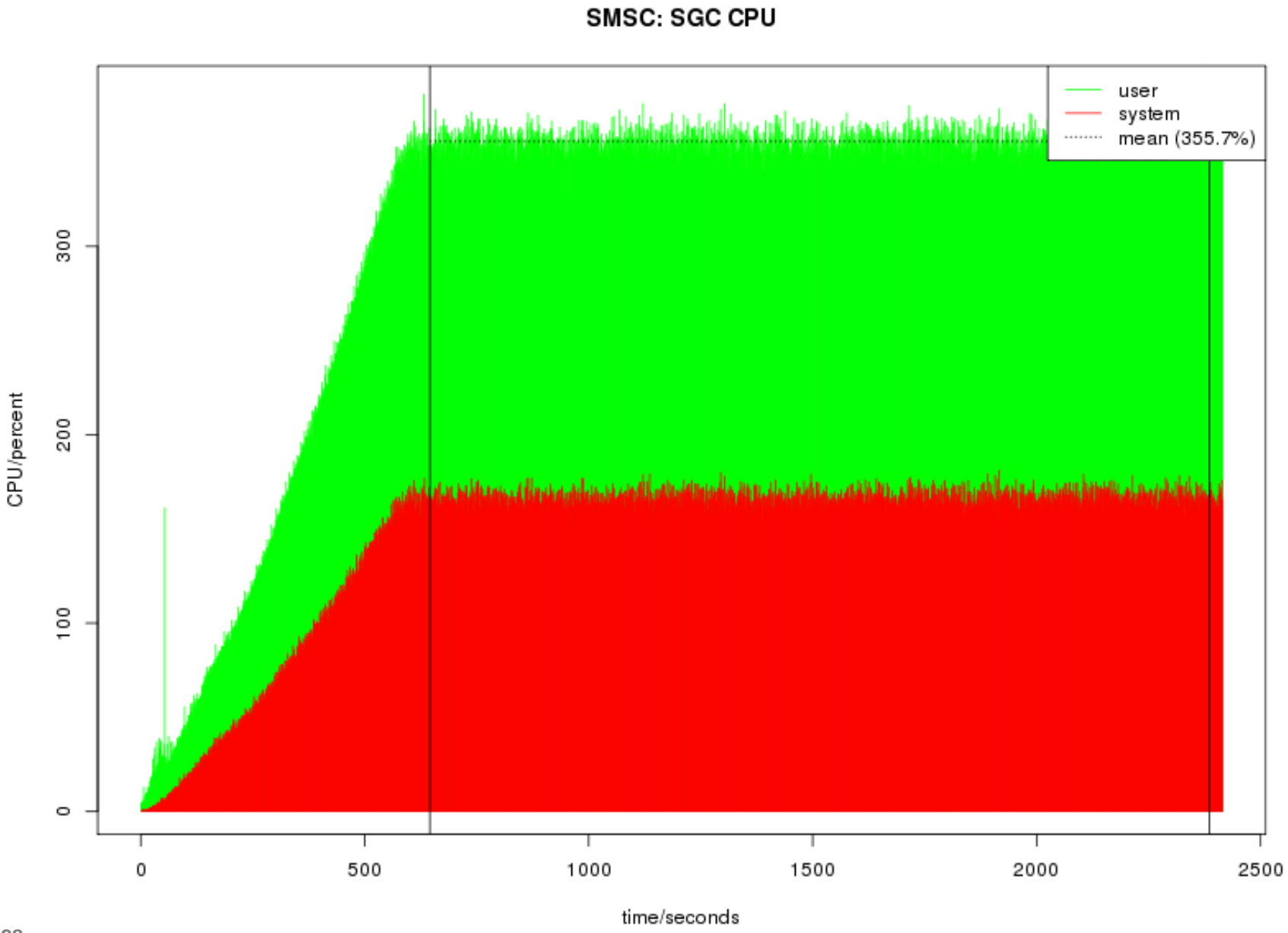
6.2 Detailed metrics

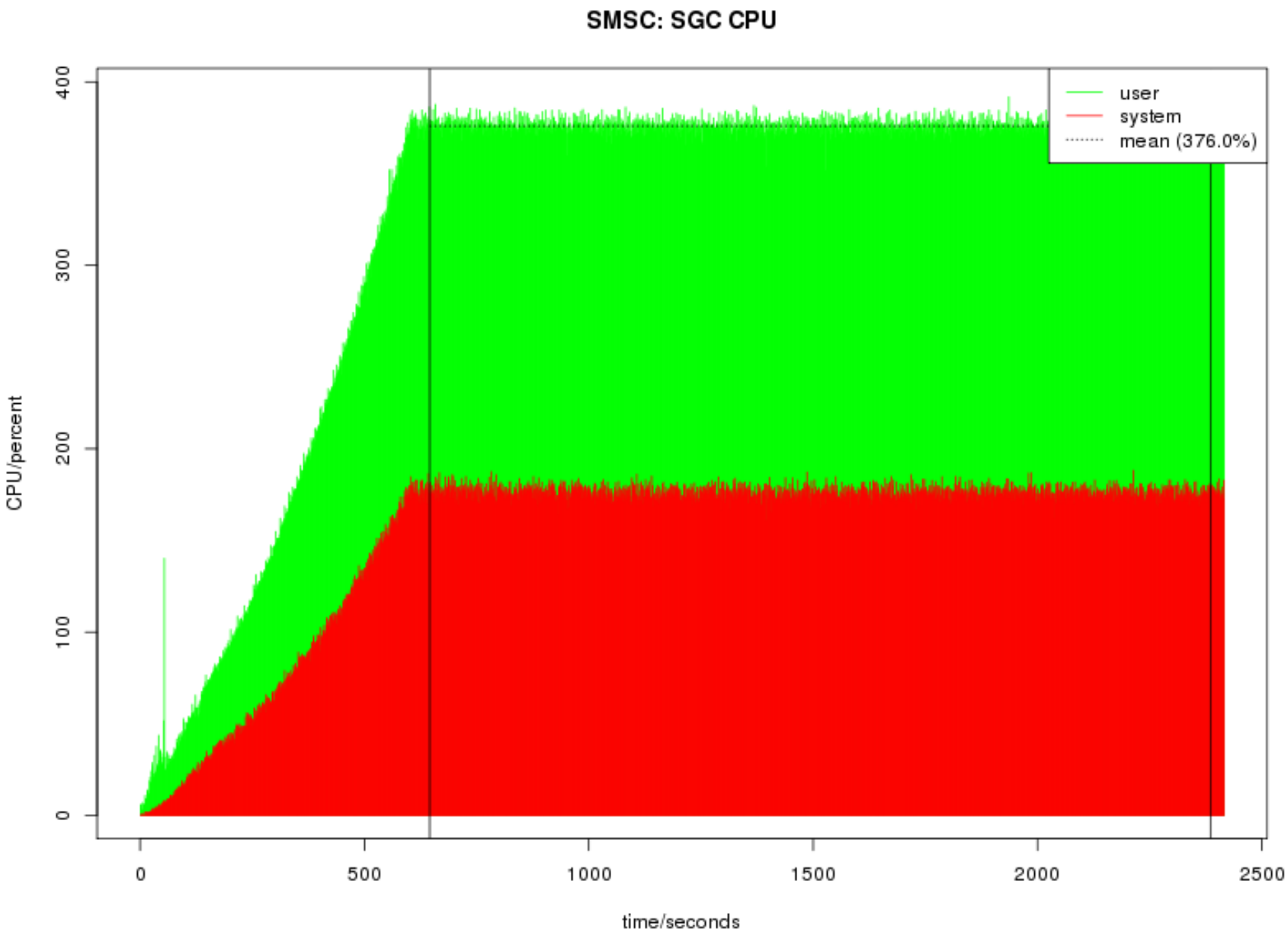
6.2.1 Call Rate



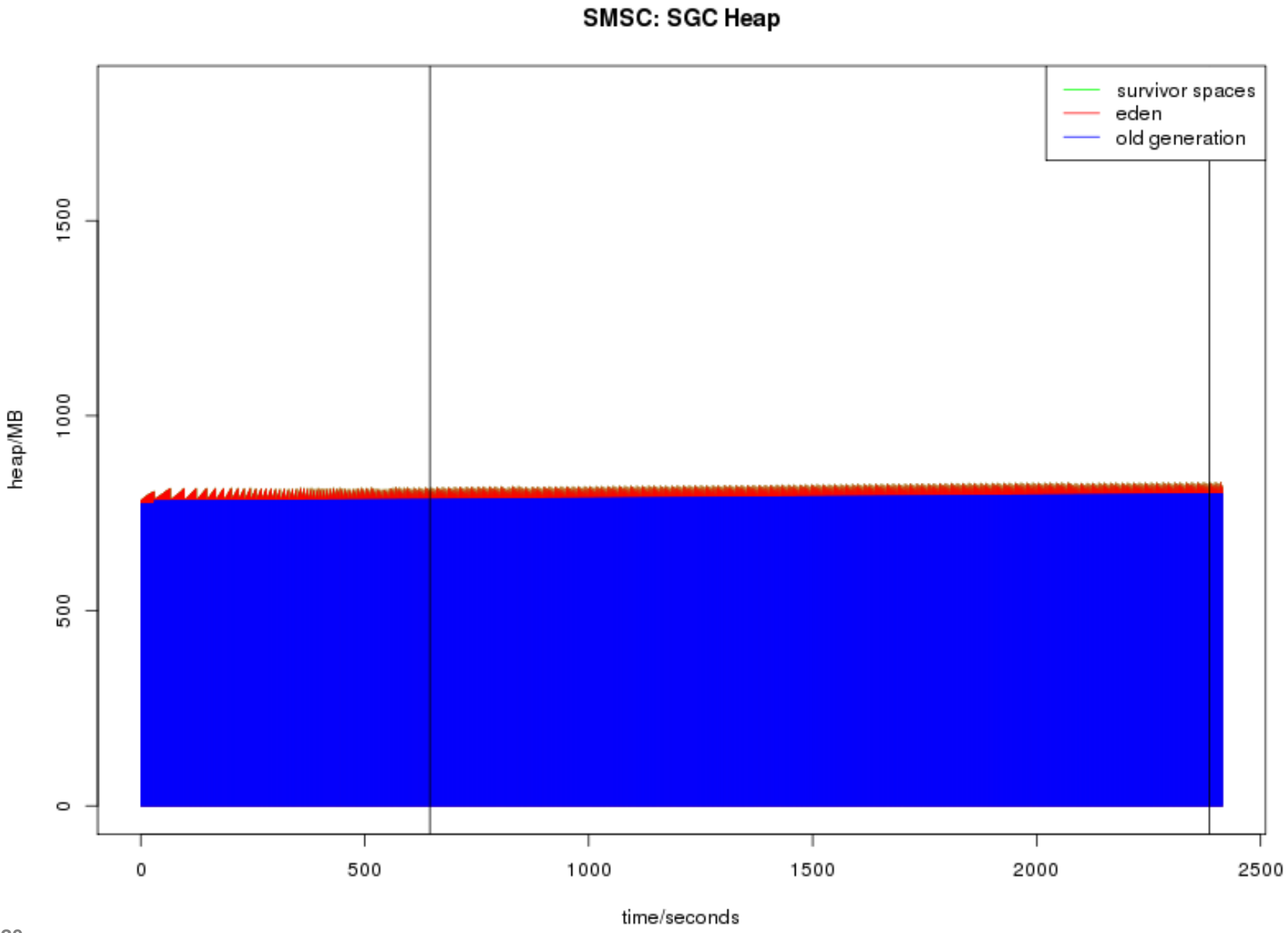

SMSC-sim: Group session completion rates
All sessions (41059542)
TestSRIforSM (41059542)
rate/events/second
0
5000
10000
15000
20000
25000
0
500
1000
1500
2000
2500
time/seconds

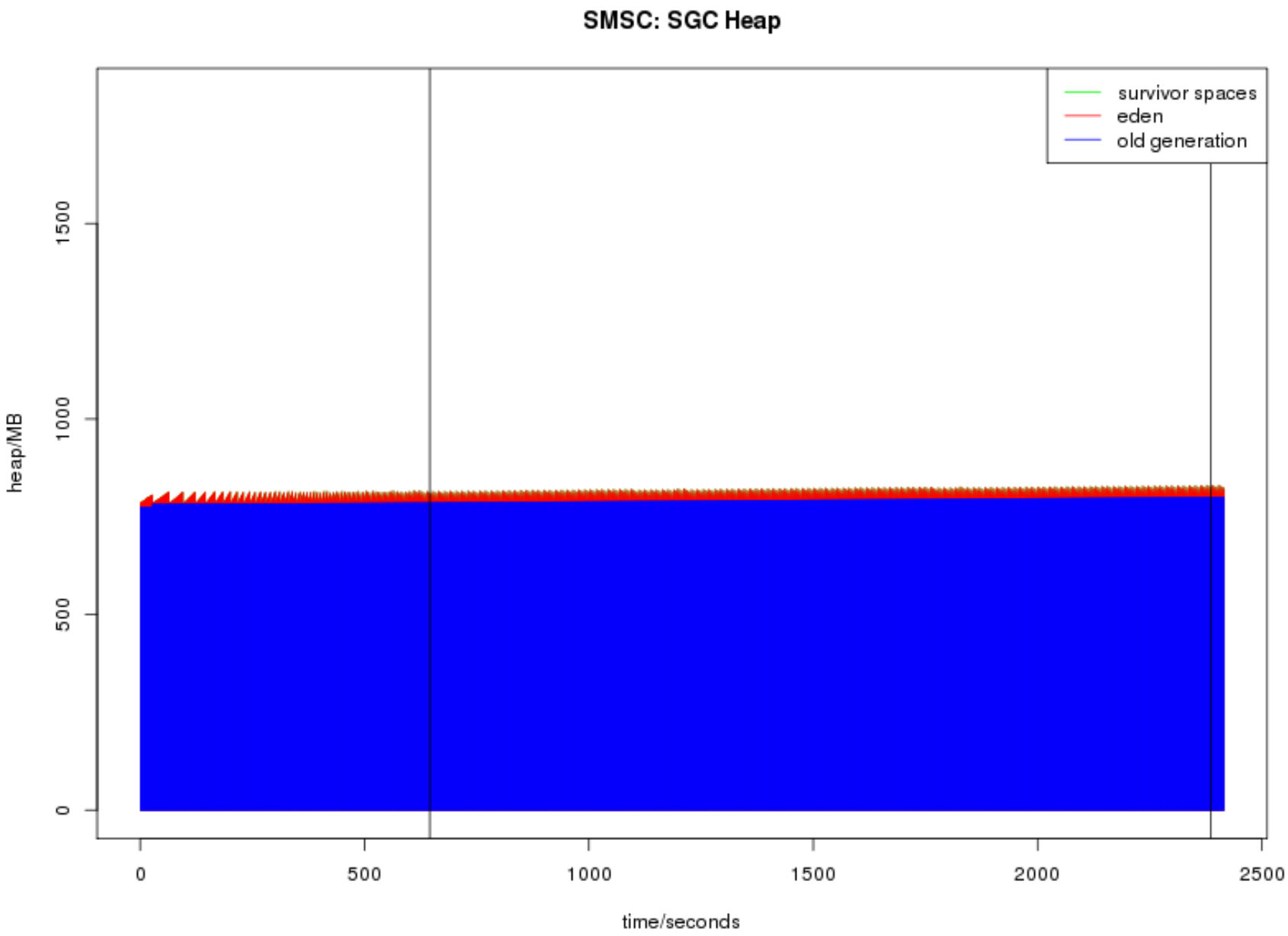
6.2.2 CPU usage



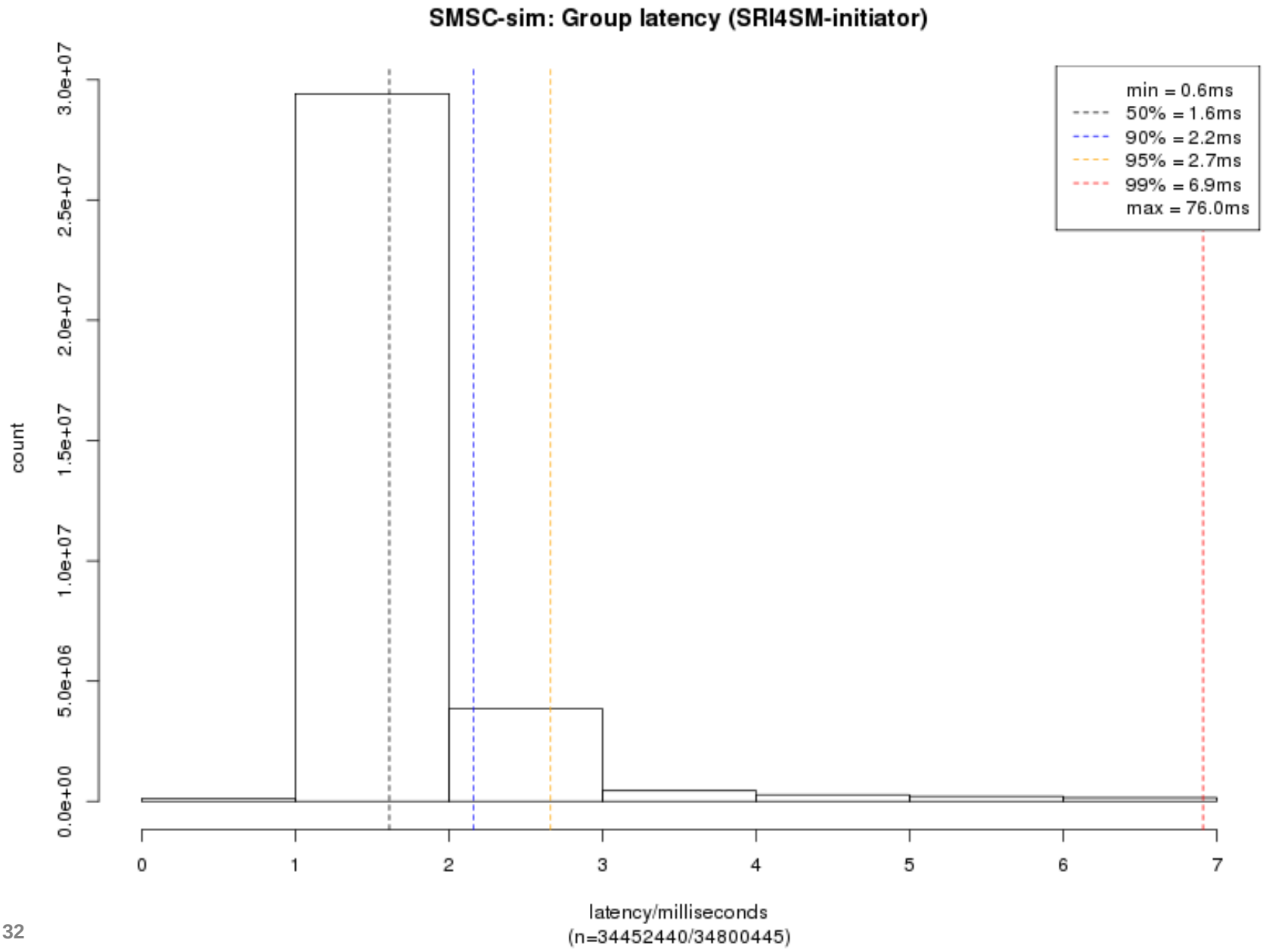

SMSC: SGC CPU
user
system
mean (376.0%)
CPU/percent
0
100
200
300
400
0
500
1000
1500
2000
2500
time/seconds

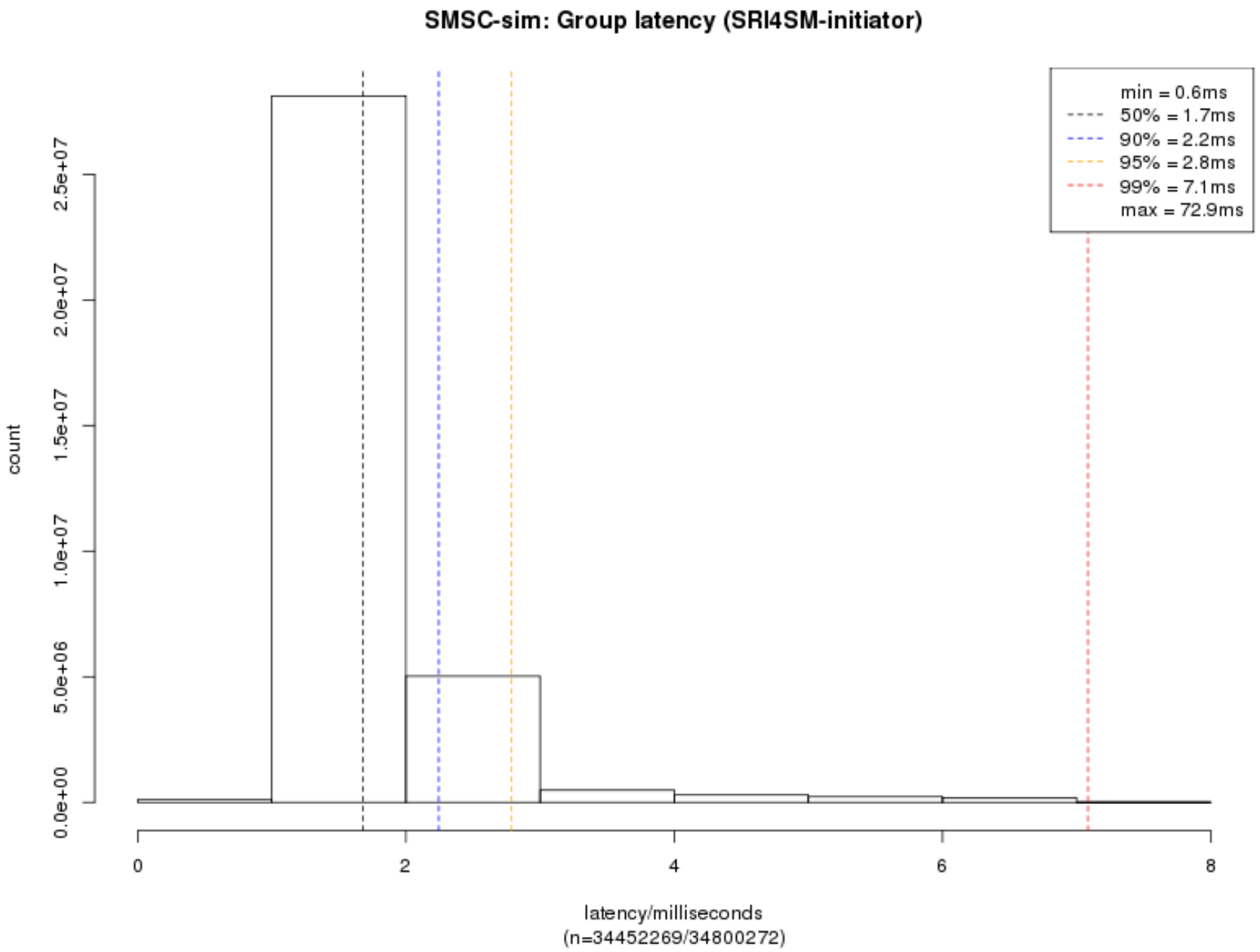
6.2.3 Heap usage




SMSC: SGC Heap
survivor spaces
eden
old generation
heap/MB
0
500
1000
1500
0
500
1000
1500
2000
2500
time/seconds

6.2.4 Scenario latencies




SMSC-sim: Group latency (SRI4SM-initiator)
min = 0.6ms
50% = 1.7ms
90% = 2.2ms
95% = 2.8ms
99% = 7.1ms
max = 72.9ms
count
0.0e+00
5.0e+06
1.0e+07
1.5e+07
2.0e+07
2.5e+07
0
2
4
6
8
latency/milliseconds
(n=34452269/34800272)