
Platform for Control and Delivery of services in Next Generation Networks

DELIVERABLE 3.3

Tests



PROPRIETARY INFORMATION

©CEFRIEL 2008 All Rights Reserved

©PoliTO 2008 All Rights Reserved

This document and the data included in this document is proprietary to CEFRIEL and Torino's polytechnic, and is not to be reproduced, used, or disclosed in whole or in part to anyone without the express written permission of the parts above. The content of this document is provided for informational use only and is subject to change without notice.

Questions about this document or the features it describes should be directed to:

CEFRIEL

RECORD OF CHANGES

Version	Date	Description
0.1	18/01/2011	First draft version
1.0	28/04/2011	Final version

Index

Index	4
1 Introduction	5
2 Scope	6
2.1 What does this document entail?.....	6
2.2 What is being tested?.....	6
2.3 Exclusions.....	6
2.4 PICO and IMS Integration test	7
3 Testbed	8
3.1 Hardware	8
3.2 Tools	8
3.3 Architecture.....	9
4 Test Execution	10
5 Test results	16
5.1 How to improve performances and reduce CPU load.....	16
Index of Figures	17

1 Introduction

This document describes the appropriate strategies, process, workflows and methodologies used to plan, organize, execute and manage testing of PICO platform. The platform is composed by several components and the tests will be split in two parts, the test of the application server which includes the JAVA core application, the database and the reasoner; and the test of the IMS and the communication between the devices which use the IMS architecture to perform a communication or an application sharing.

The documents starts with the scope listing the objectives, the requirements and the tools used, afterwards there will be a section win which the test will be executed and finally will be presented a section with the results and considerations.

2 Scope

2.1 What does this document entail?

The purpose of this document is to test the components of PICO Architecture. PICO architecture is composed by several components such as the JAVA core application, the database, the reasoner, the IMS framework and so on.

Because the goal of PICO is to evaluate IMS architecture and its integration in particular, we'll focus on IMS communication testing.

2.2 What is being tested?

The components that will be tested on PICO are:

- Application JAVA core
- IMS communication backend side
- IMS communication client side
- ContextML file exchange

2.3 Exclusions

The PICO system exposes also a web interface which offers several features. The main features are:

1. The opening of an emergency: The way to open an emergency from a web interface.
2. Emergency Map: Is the map in which all emergencies and all users are shown

For our purpose just the first feature will be tested because during this operation some queries into the database will be performed. The second feature won't be tested because the performances depend from Google's http requests.

2.4 PICO and IMS Integration test

In the field of benchmarking, considerable efforts have been done for defining accurate workload models. The document gives more details about the integration between PICO and its IMS integration, and a performance evaluation is presented in order to show the obtained results when the specification is used for benchmarking an IMS deployment.

The benchmark execution lasts only 24 minutes, and the tested system is only evaluated with loads of 20 and 80 calls per second.

3 Testbed

3.1 Hardware

We arranged a testbed to perform our test which will test the IMS communication from backed point of view. To have a complete test solution, it is necessary to arrange a testbed for the mobile client but it is out of our scope for this deliverable. Tfb which stands for Testbed for backed is composed by 3 hosts:

Host	CPU	RAM	OS	Role
picoSERVER.cefriel.it	1,5 Ghz	1024 MB	Ubuntu 10.1	System Under Test (SUT)
xpresenter.cefriel.it	1,5 Ghz	512 MB	Windows XP	Test System (TS)
pistillosieben.cefriel.it	2,2 Ghz	3 GB	Windows 7	SUT

The picoSERVER is the computer which runs the SIP/IMS server and the database, pistillosieben is the machine where the JAVA core application run and the xpresenter machine is the test system where the test tool are executed.

3.2 Tools

We have used essentially a simple tool called SipP. It is a test tool / traffic generator for the SIP protocol with XML customized scenario (<http://sipp.sourceforge.net/>). We have used this tool using these requirements:

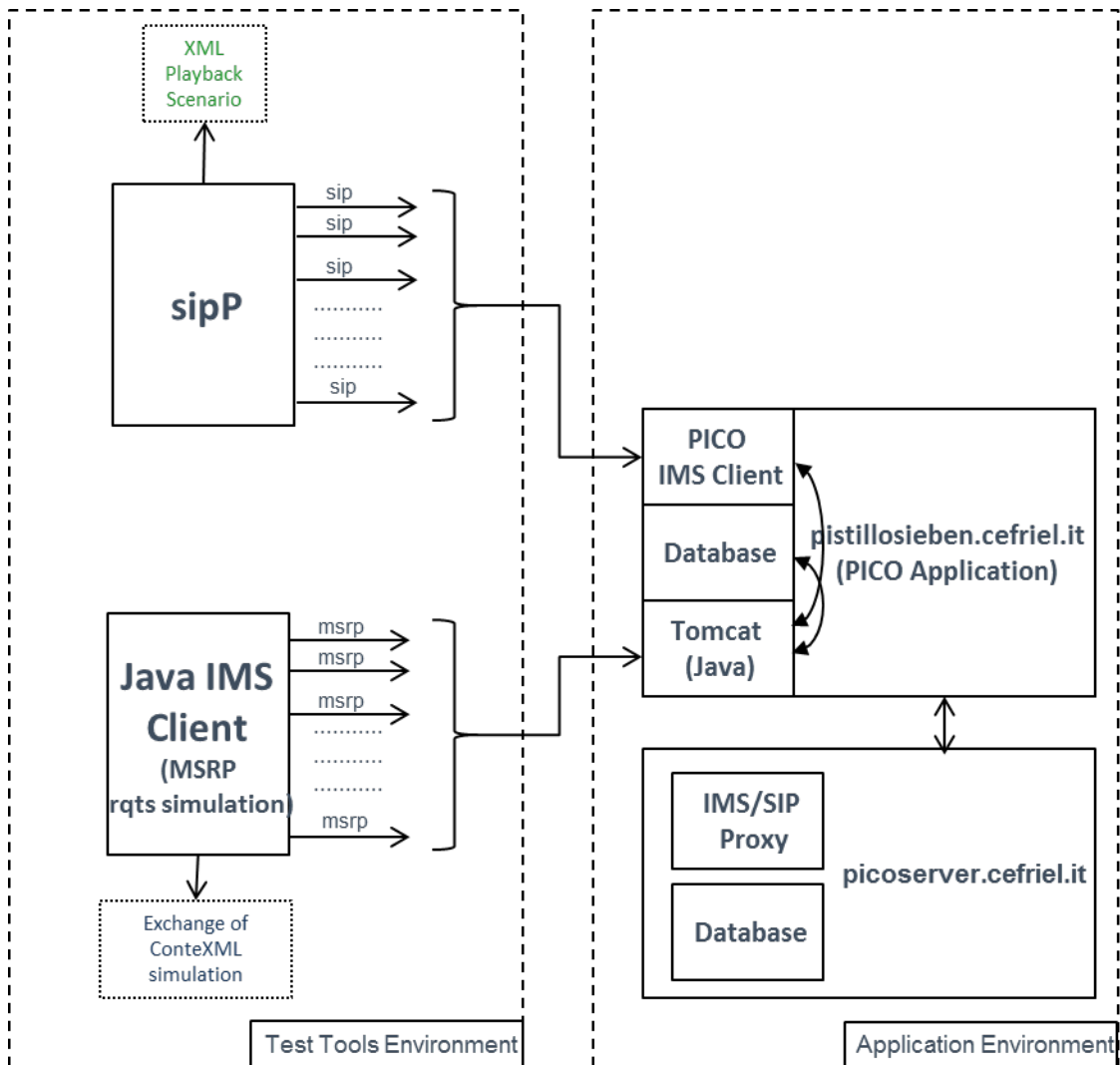
- ✓ DTMF tones sent via 2833 protocol
- ✓ Audio sent via RTP stream using .pcap files (packet captured analyzing network traffic)
- ✓ Call per second (callrate): 10calls/
- ✓ Fixed Time between menus

Another tool which has been used is a small Java IMS Client written only for test purpose. It will simulate the exchanging of ConteXML using the MSRP protocol. It uses doubango as IMS framework.

3.3 Architecture

The box on the left represents the environment which executes the test also called Test System (TS). The machine used is: *xpresenter.cefriel.it*. This machine will execute two tools, *sipP* to simulate sip traffic, and a small Java IMS client to simulate MSRP traffic (ConteXML updates).

The box on the right also known as System Under Test (SUT) is composed by two machine, the first one, *pistillosieben.cefriel.it*, will execute the application core and the embedded IMS client. The second machine will execute the database and the IMS/SIP Proxy.



4 Test Execution

In the first test, the sipP tool generates up to 250 sip calls. Each call takes 1:30 minutes. CPU grows up to the **25%** with peaks till **85%** while accepting incoming calls. This increment is caused by the PICO java code that handles incoming calls. During playback, which is provided by IMS framework using SIP, the CPU is below **10%** with some peaks at **40%** even if 250 calls are managed.

At the end of the playback, while PICO java code manages the call drop, the CPU grows level comparable with the one reached at the beginning of the test.

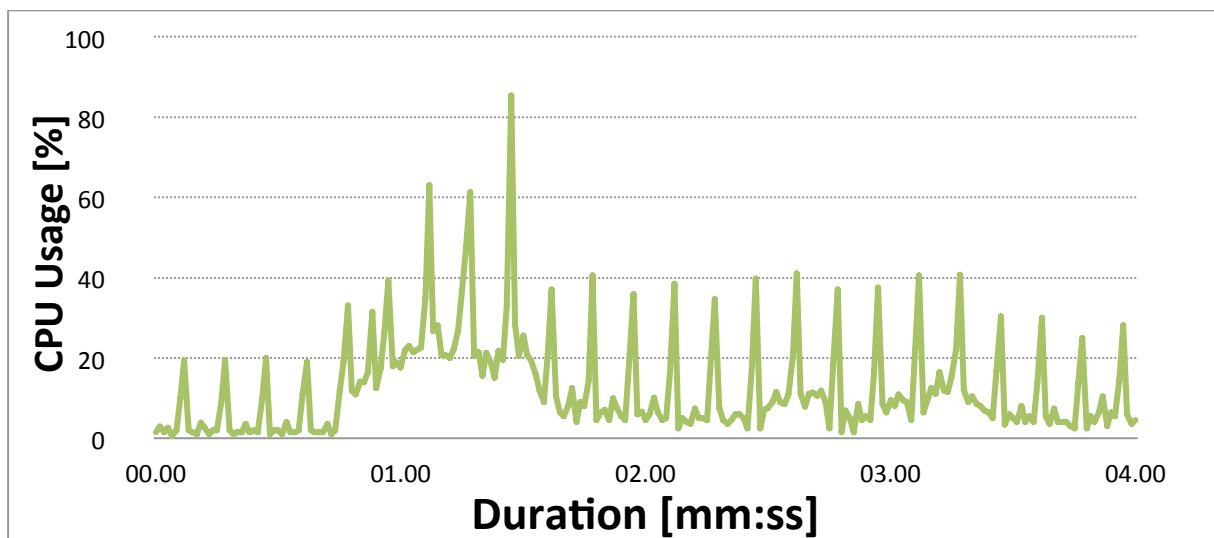


Figure 1: 250 Simultaneous playback SIP calls

The Figure 2 shows the load of the two cores during the test. CPU load is quite balanced on 2 cores available, any of the cores is stressed to higher levels of load and peaks are positioned at the same times in both cores.

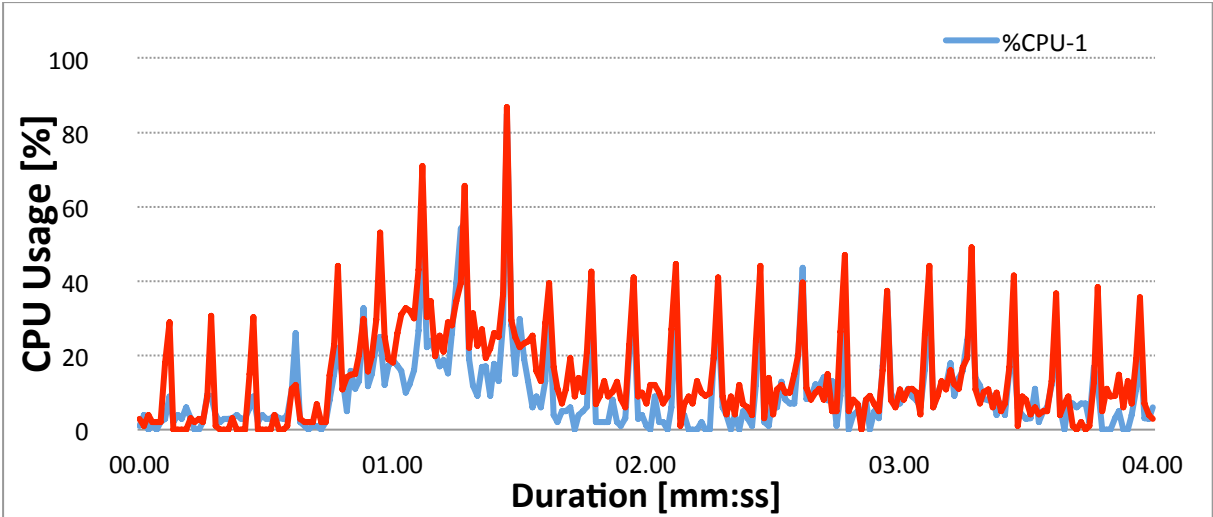


Figure 2: CPU 1 & 2 Load

The second (Figure 3) test adds to the sip calls also the exchange of the ConteXML file using MSRP protocol. In this scenario we have 125 Sip calls and 125 MSRP sessions.

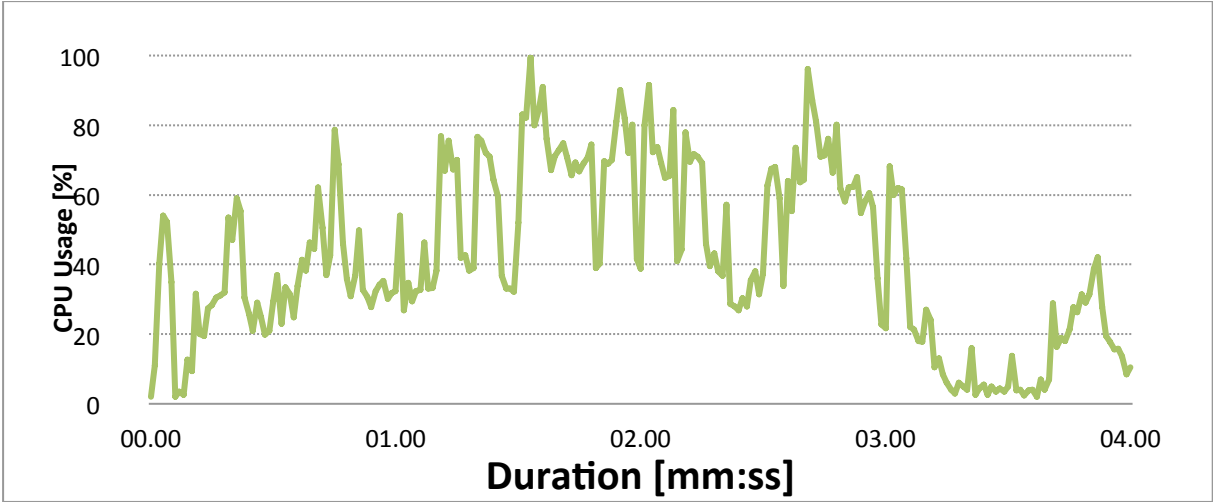


Figure 3: 125 SIP Calls & 125 MSRP Sessions

As shown in previous results, MSRP sessions require more CPU usage than Playback because each file is processed and parsed. While 125 channels are in playback contemporarily the CPU grows up to levels higher than the ones reached with 200 in playback and only 50 MSRP Sessions.

The general increment is due to the 125 MSRP session and that are still not enough to put CPU in I/O waits state to wait for hard disk writing of the ConteXML file.

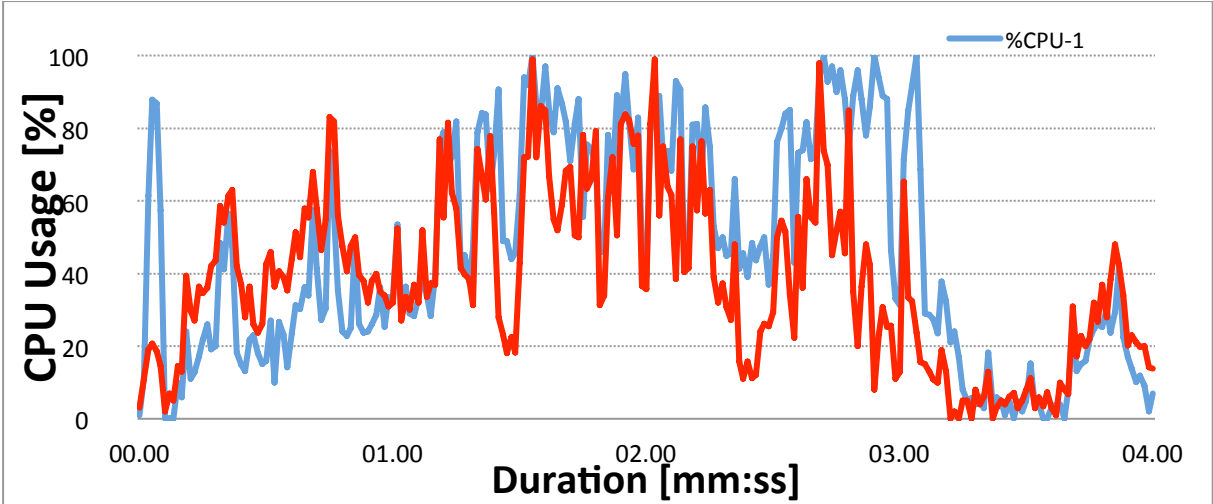


Figure 4: Core 1 & Core 2, 125 sip calls and 125 MSRP sessions

Even if the distribution of the load is different in the 2 cores, the load is still balanced. None of the cores is considerably different from the other.

only SIP

In the Figure 5 is shown the usage of the RAM during the tests. Compared to the only SIP calls test that uses about 70/80% of RAM, while managing MSRP Sessions, the used memory is around 97% but never reaches 100%. MSRP session test has quite same results that SIP test. The increment in RAM usage is primarily caused by Java Application that manages the parsing of the ConteXML files.

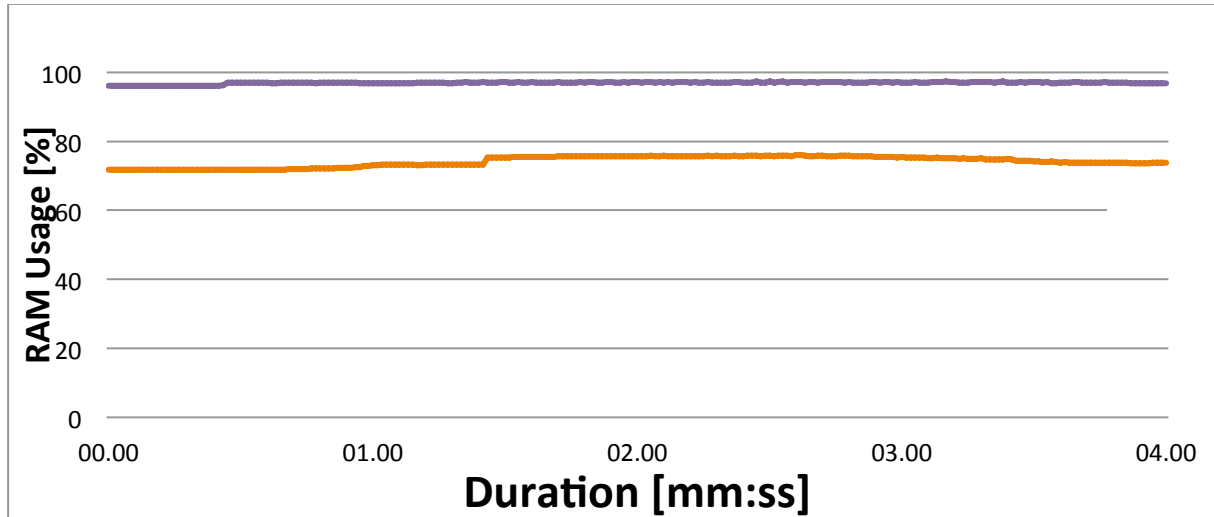


Figure 5: RAM Usage

250 sip calls coded with G.711 at 64kbs each one result in a total bandwidth of about **20000kb/s** as theoretically calculated. Equivalent bandwidth is measured for outgoing and incoming traffic while running playback.

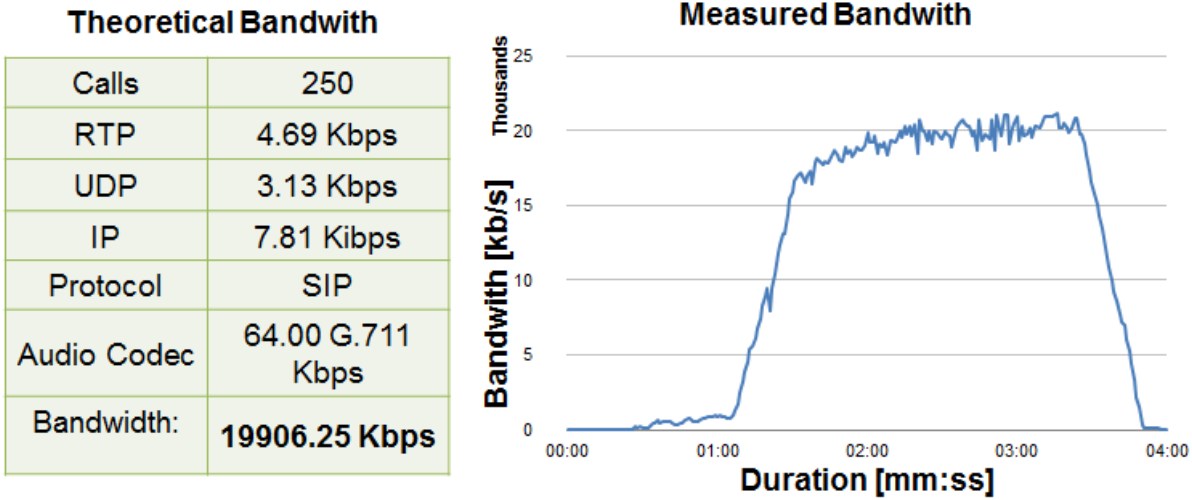


Figure 6: Network Usage

5 Test results

The Figure 7 shows a comparison among SIP calls only, MSRP session only and SIP & MSRP sessions together.

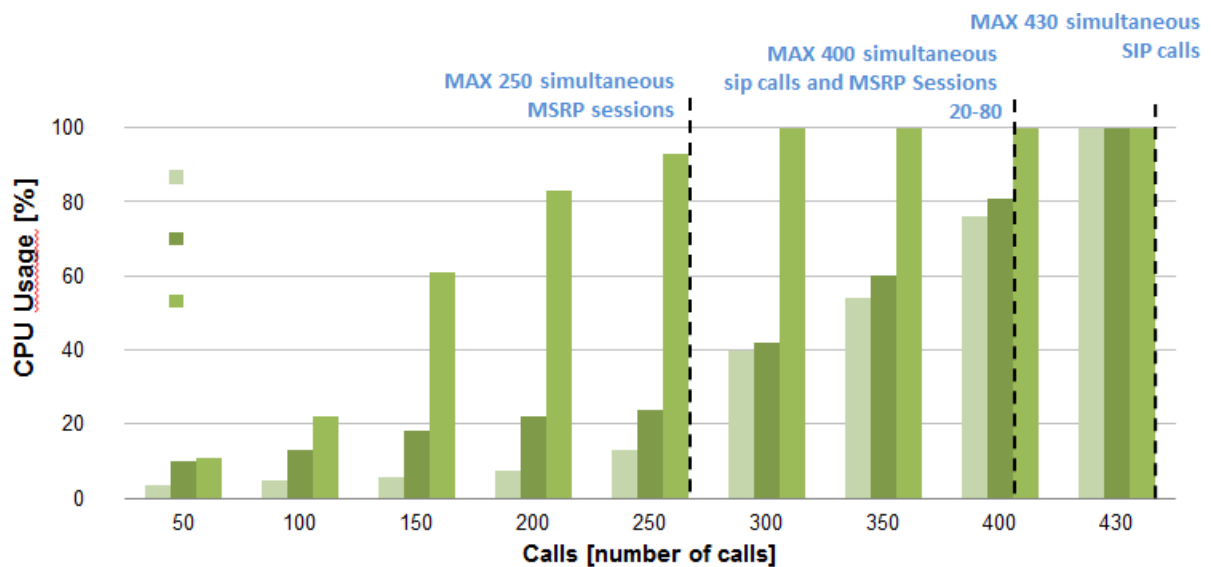


Figure 7: SIP Calls and MSRP comparison

MSRP session test reach 100% CPU load with less calls than sip call test. That because the MSRP session required to write a file on filesystem and parse it. MSRP sessions maximum limit is 250 which requires a CPU load of about 93%.

Test using mixed sources of 20% MSRP sessions and 80% in sip calls is higher, but more similar, to the sip calls only test's result. Maximum limit for mixed configuration is 400 (80 MSRP sessions and 320 in sip call playback)

5.1 How to improve performances and reduce CPU load

- ✓ Use a dedicated hard disk for PICO Application only because it requires to write files on file system
 - ✓ Avoid writing of file using it in memory
 - ✓ RAID Configuration (i.e. RAID 5)
 - ✓ Disk with greater performances (i.e. 10,000 rpm)
 - ✓ Distribute PICO application, Database and IMS framework on different servers
 - ✓ More ram (the ram used 2gb was to low)
-

Index of Figures

Figure 1: 250 Simultaneous playback SIP calls.....	10
Figure 2: CPU 1 & 2 Load	11
Figure 3: 125 SIP Calls & 125 MSRP Sessions	12
Figure 4: Core 1 & Core 2, 125 sip calls and 125 MSRP sessions.....	13
Figure 5: RAM Usage	14
Figure 6: Network Usage	15
Figure 7: SIP Calls and MSRP comparison	16
