Primljen: 30.09.2015. Pregledni rad
Prihvaćen: 3.11.2015. UDK 621.395:004.738.5

A performance comparison of three SIP softswitches: Asterisk, FreeSWITCH, and Yate

Usporedba performansi tri "softswitcha": Asterisk, FreeSWITCH i Yate

[1] Igor Tomičić, [2] Matija Turk, [3] Mišo Lovrenčić
[1] Fakultet organizacije i informatike, Pavlinska 2, Varaždin
[2, 3] Novi-Net d.o.o., Franje Punčeca 2, Čakovec
e-mail: [1] igor.tomicic@foi.hr, [2] matija.turk@gmail.com, [3] mlovrenci@gmail.com

Abstract: *In the telecommunications network or the public Internet, softswitches are the software implementation of central devices that connect calls between phone lines, usually executing on a general-purpose computer system. "Softswitch" is short for "software switch", which implies the use of general purpose servers and VoIP technology, instead of purpose-built electronic hardware. This paper presents (1) an overview of some of the main characteristics of three popular softswitches: Asterisk, FreeSWITCH, and Yate; (2) measures their performances on a designed scenario with identical initial parameters, and (3) presents the results of conducted performance tests. The methodology is comprised of two test scenarios; Test 1 implies generating 800 active calls on a freshly booted system, and sustaining them for 20 minutes. Monitored parameters include CPU utilization and Linux 5 minute system load. Test 2 consists of sustained 5 calls per second, and monitored parameter is the number of active calls; the purpose is to obtain the maximum active calls sustained. By analyzing test outcomes of the performed simulations, FreeSWITCH showed highest performance results in both scenarios.*

Keywords: *softswitch, asterisk, yate, freeswitch, performance test*

Sažetak: *U kontekstu telekomunikacijskih mreža i javnog interneta, "softswitch" je naziv za programsku implementaciju centralnih uređaja koji povezuju pozive između pojedinih telefonskih linija, te koji se uobičajeno izvršavaju na računalnim sustavima općenite namjene. "Softswitch" je skraćeni naziv za pojam programskog preklopnika (eng "software switch"),* *što implicira upotrebu računalnih poslužitelja općenite namjene i VoIP tehnologije umjesto namjenske elektronske opreme. Unutar ovoga rada izložen je (1) pregled glavnih karakteristika tri popularna "softswitcha": Asterisk, FreeSWITCH i Yate, (2) mjerene su njihove performanse nad kreiranim scenarijem uz identične početne parametre, te su (3) prezentirani rezultati provedenih testova. Metodologija rada se sastoji od specifikacije dva testna scenarija, definiranja metrike performansi, te detaljiziranja početnih parametara simulacije. Test 1 uključuje generiranje 800 aktivnih poziva nad sustavom, te zadržavanje poziva 20 minuta. Mjereni parametri uključuju iskorištenje centralne procesorske jedinice, te 5-minutno opterećenje Linux sustava. Test 2 sastoji se od zadržavanih 5 poziva po sekundi, a mjereni parametar je broj aktivnih poziva; cilj testa jest utvrđivanje maksimalnog broja zadržanih poziva. Analizom izlaznih vrijednosti izvršenih simulacija, FreeSWITCH je pokazao znatno veće performanse u oba scenarija.*

Ključne riječi: *programski preklopnik, asterisk, yate, freeswitch, test performansi*

1. Introduction

Asterisk is an open source framework which implements a telephone private branch exchange (PBX), with the possibility to enable ordinary personal computer to become a communications server (Get Started: What is Asterisk?). More specifically, it is an open source hybrid time division multiplexing (TDM) and packet voice private branch exchange (PBX), and interactive voice response (IVR) platform, with automatic call distributor (ACD) functionality (Spencer et al., 2003.).

FreeSWITCH™ is "a scalable open source cross-platform telephony platform designed to route and interconnect popular communication protocols using audio, video, text or any other form of media." (The World's First Cross-Platform, n.d.). FreeSWITCH facilitates a number of telephony applications through its modules. These applications include, but are not limited to: interactive voice response, XML-remote procedure call (RPC) control of live calls, conferencing, speech synthesis, speech recognition, voice over IP protocols such as SIP (Session Initiation Protocol), SCCP (Skinny Client Control Protocol), H.323, T.38, XMPP (Extensible Messaging and Presence Protocol), etc. (Modules, 2014).

Yate is an abbreviation for „Yet Another Telephony Engine", which represents a free and open source communications software with support for video, voice and instant messaging

(What is Yate, 2013). It is extensible, written in C++ with a modular design, and currently focused on Voice over Internet Protocol (VoIP) and Public Switched Telephone Network (PSTN).

2. Asterisk

As stated in (Van Meggelen et al., 2007, pp. XV), “Asterisk combines more than 100 years of telephony knowledge into a robust suite of tightly integrated telecommunications applications“. Applications such as conferencing, call queuing, music on hold, voicemail, call parking, are all built into the software. Asterisk also supports station-to station calls, line trunking, sending and receiving caller ID, call routing based on the caller ID, call waiting, call return, call forwarding, call transferring, advanced call distribution (routing decisions based on the attributes of the received call), call detail records (keeping complete call detail record, CDR, in a file, or database), call recording, etc. The complete list of features is maintained and available at official Asterisk website (Features).

Asterisk users can implement new features and functionalities through Asterisk's own extensions languages, by implementing Asterisk Gateway Interface (AGI) programs, or by adding custom loadable modules written in C.

Several standard voice over IP protocols are supported by Asterisk, and these include the Session Initiation Protocol (SIP), the Media Gateway Control Protocol (MGCP), and H.323. A native Asterisk protocol named IAX (Inter-Asterisk eXchange) provides trunking of calls among Asterisk PBX-es.

In addition to supporting voice over IP protocols, Asterisk also supports ISDN (Integrated Services Digital Network) and SS7 (Signaling System 7), traditional circuit-switching protocols, but the implementation itself requires additional hardware interface cards.

3. FreeSWITCH

FreeSWITCH uses freely available software libraries in order to eliminate complexity. These libraries perform FreeSWITCH functions, and include: SQLite, Perl Compatible Regular Expressions (PCRE), an open-source SIP user agent library „Sofia-SIP“, Apache Portable Runtime, Speex DSP library „libspeex“, an open-source implementation of the Secure Real-time Transport Protocol „libSRTP“. Although initially written in C programming language, FreeSWITCH may be launched from languages such as C, C++, Python, Perl, Lua, Java, JavaScript. It has a modular and extensible architecture with only a few essential functionality in its core. Supported operating systems include Windows, Linux, Mac OS X, BSD, Solaris, on both 32 and 64-bit platforms.

The official FreeSWITCH website lists possible uses for their system ("Specifications", 2014):

- Rating & Routing Server
- Transcoding B2BUA
- IVR & Announcement Server
- Conference Server
- Voicemail Server
- SBC (Session Border Controller)
- Basic Topology Hiding Session Border Controller
- DAHDI, Khomp, PIKA, Rhino, Sangoma and Xorcom Hardware Support
- Fax server
- PBX

The same source lists some FreeSWITCH performance metrics:

- Tested under load for over 100 hours
- 10,000,000+ calls
- At rates exceeding 50 CPS

4. Yate

Yate's core software is written in C++, but it supports scripting in languages such as Python, Perl, PHP, and Unix shell.

According to (Yate architecture, 2013), message-passing system is the most important aspect of Yate, where modules are passing messages between them. This feature should allow for a larger flexibility in contrast to using plain functions. This is due to the facts that (1) messages in Yate can have an arbitrary number of parameters, and (2) messages can be sent to more than one module by changing the priority.

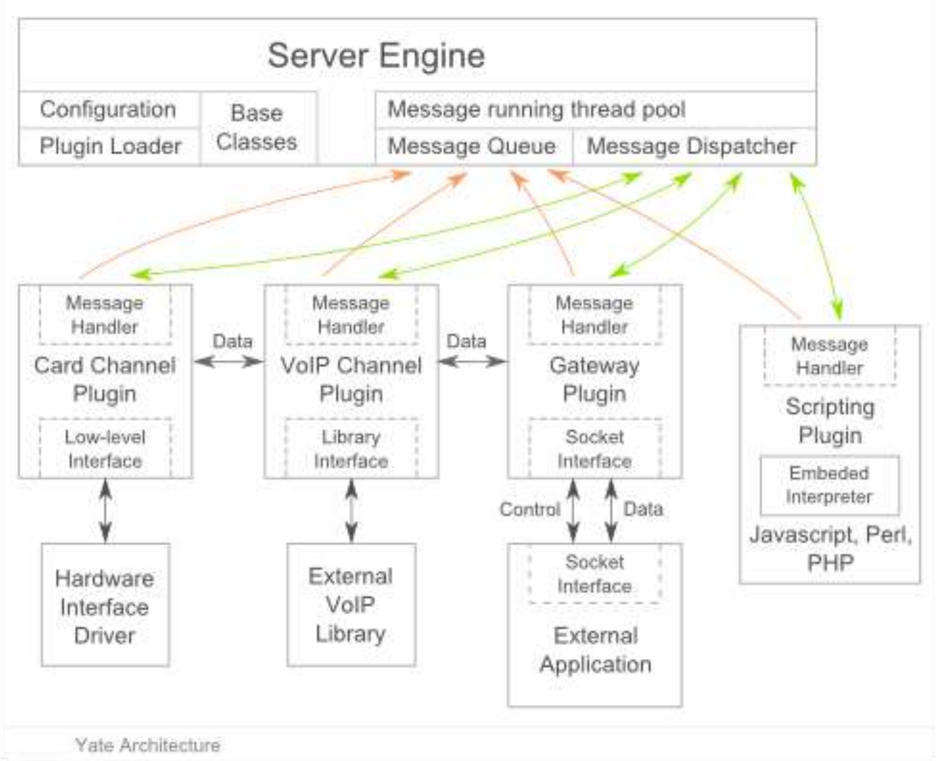
The four main components of Yate are the following (Yate architecture, 2013):

1. *Core*. This includes generic classes like String, Thread, Socket, Mutex.

2. *Message Engine*. Message related classes (specifically, Message, Engine and Plugin classes).
3. *Telephony Engine*. Telephony related classes (like Driver and Channel).
4. *Yate Modules*.

Yate's architecture is presented in Figure 1.

Figure 1. Yate Architecture



Source: "Yate architecture", 2013

5. Comparison of the Observed Softswitches

The initial, static comparison of the observed softswitches is presented via Table 1 and Table 2, and are focused on the main characteristics of softswitches such as protocols in use, supported operating systems, programming languages they were written in, supported audio and video codecs, etc.

Table 1. Main characteristics of observed softswitches

Software	Protocols	Supported operating systems	Written in
Asterisk	H.323 SIP IAX ISDN MGCP SS7	Linux BSD Mac OS X Solaris	C
FreeSWITCH	H.323 SIP IAX ISDN JINGLE SCCP STUN SIMPLE XMPP MRCP RSS Skype	Windows (native) Linux/BSD Solaris Mac OS X	C
Yate	H.323 SIP IAX ISDN JINGLE XMPP MGCP SS7 over IP Cisco SLT SCTP SCCP TCAP MAP CAMEL	BSD Linux Windows Mac OS	C++

Source: author's compilation

Table 2. Comparison of softswitch servers

Media server	**Asterisk**	**FreeSwitch**	**Yate**
www	Asterisk.org	Freeswitch.org	Yate.null.ro
License	GPL	MPL	GPL
Last stable rel.	1.8.2.3	1.0.6	3.3.0
Operating system	Linux, Mac OS X, *BSD, Solaris, Windows	Linux, Mac OS X, *BSD, Solaris, Windows	Linux, Mac OS, FreeBSD, Windows
Wrritten in	C	C/C++	C++
Architecture	B2BUA	B2BUA	—
Modular		Yes	
NAT traversal	No	STUN	No
Authentication, authorization against database	MySQL, PostgreSQL, LDAP, Radius	MySQL, PostgreSQL, LDAP, Radius	MySQL, PostgreSQL, Radius
VoIP signalling protocols	SIP, H,323, SCCP, MGCP, IAX, GoogleTalk	SIP, H.323, IAX, SCCP	SIP, H.323, MGCP, IAX, Jingle
Telephony signalling protocols	ISDN/SS7, FXS/FXO	ISDN/SS7	ISDN/SS7, FXS/FXO, Sigtran
Messaging protocol	XMPP	SIMPLE, XMPP	XMPP/Jabber
Call encryption	SRTP	SRTP	No
Transport protocols	UDP, TCP, SCTP, TLS	UDP, TCP, SCTP, TLS	UDP, TCP, SCTP
IPv4/IPv6	Yes/Yes	Yes/Yes	Yes/—
Web GUI	Yes	Yes	—
SIMPLE	No	Yes	No
SIP gateway	Yes	Yes	Yes
Audio codecs	ADPCM, PCMU, PCMA, G.722, G.722.1, G.722.1 Annex C, G.723.1, G.726, G.729a, GSM, iLBC, Linear, LPC-10, Speex	CELT, G.722.1, G.722.1C, G.722, PCMU, PCMA, GSM, G.726, AAL2 and RFC 3551, G.723.1, G.729AB, AMR, iLBC, Speex, LPC-10, DVI4, SILK	GSM, speex, iLBC, AMR-NB
Video codecs	No	Theora, H.261, H.263, H.263+, H.263++, H.264, MP4	No
Transcoding		Yes	
IVR and Announc.		Yes	
Voice mail	Yes	Yes	—
Audio conference		Yes	
Call recording	Yes	Yes	—
IP/PBX features		Yes	
CDR		Yes	
Fax	T.30, T.38	T.30, T.38	—
Text to speech	Yes	Yes	—
SIP API		No	
Programming languages	With CGI any language, Adhersion	C/C++, Python, Perl, Lua, Java, JavaScript, Erlang, Ruby	Python

Source: Segec and Kovacikova, 2011

6. Lab environment

Softswitch lab environment consists of two machines: client for generating calls and server to handle the load. Machines are initialized in the same subnet, in order to exclude router overhead. Softswitches are installed on the same server machine, but only one of them remains active while performing test. They are installed with default configuration and are executing simple dial plan with *answer*, *playback* and *hung-up*. It is important to note that there are some configuration tweaks to achieve maximum server performance and to utilize most of it. Logging and unnecessary modules are disabled.

Client machine is a bare minimum Debian Jessie operating systems with the SIP performance tester application installed. This application is a free Open Source test tool / traffic generator for the SIP protocol, named “SIPp” (Gayraud and Jacques, 2014), and it includes a few basic user agent scenarios, with the possibility to establish and release multiple calls with the INVITE and BYE methods.

Server hardware specification:

- Xeon Quad core @ 3GHz
- 10 GB DDR2
- 72GB 10K SAS

Client hardware specification:

- Pentium Dual core @ 3GHz
- 4 GB DDR3
- 60GB SSD

6.1 Test Scenarios

There are two test scenarios employed for the purpose of performance measuring. Command parameters that are used in the test scenarios are listed hereafter:

- "**-d**" used to specify the pause delay, in milliseconds
- "**-s**" service field, as passed in the **-s service_name**
- "**-r**" used to specify the call rate in number of calls per seconds

- "**-l**" set the maximum number of simultaneous calls. Once this limit is reached, traffic is decreased until the number of open calls goes down

Monitoring is done via SNMP concentrator software on a separate machine. SNMP information is read from operating system SNMP service and separate SNMP modules in every softswitch tested. Tests will be run in two scenarios:

1. 800 active calls sustained.
2. Maximum active calls sustained.

6.1.1 Test 1 Methodology

Test comprises of achieving 800 active calls on a freshly booted system, and sustaining them for the total of 20 minutes. To minimize the impact of the call initiation (which greatly affects system load), CPS is set to 10 calls per second. After each test session, server is restarted.

SIPP command issued at the client is presented in Listing 1.

```
sipp -sn uac -rtp_echo -d 2h -s 44444 10.0.101.41 -l 800 -nd -r 10
```

Listing 1. SIPP command for the first test

Monitored parameters include CPU utilization (average CPU utilization across all cores), and Linux 5 minute system load.

All three systems successfully finished the tests without dropping any calls.

6.1.2 Test 2 methodology

Test 2 consists of five sustained calls per second, traffic generated at client with SIPP. System max call limit is defined as maximum opened calls at which softswitch is able to respond to SIP messages in timely manner defined by SIP timer values in RFC 3261 (“RFC 3261”, 2002).

Monitored parameter is the number of active calls. It is expected to observe significant superiority of FreeSWITCH and Yate over Asterisk, argued by the basic Asterisk architecture and design principles.

SIPP command issued at the client is presented in Listing 2.

```
sipp -sn uac -rtp_echo -d 2h -s 44444 10.0.101.41 -l 5000 -nd -r 5
```

Listing 2. SIPP command for the second test

7. Test results and comparisons

The results of conducted Test 1 are presented in Table 3. It is evident that FreeSWITCH was least resource intensive under 800 call load. With a difference of 9% percent, CPU utilization for Yate was also lower than Asterisk's, with a curious result obtained on average system load metric. As all tests were completed without encountering any difficulties, one should conclude that presented Yate system load should not signify a problem, in case no other mission critical services are run on the same server – which in the production environment should never be the case.

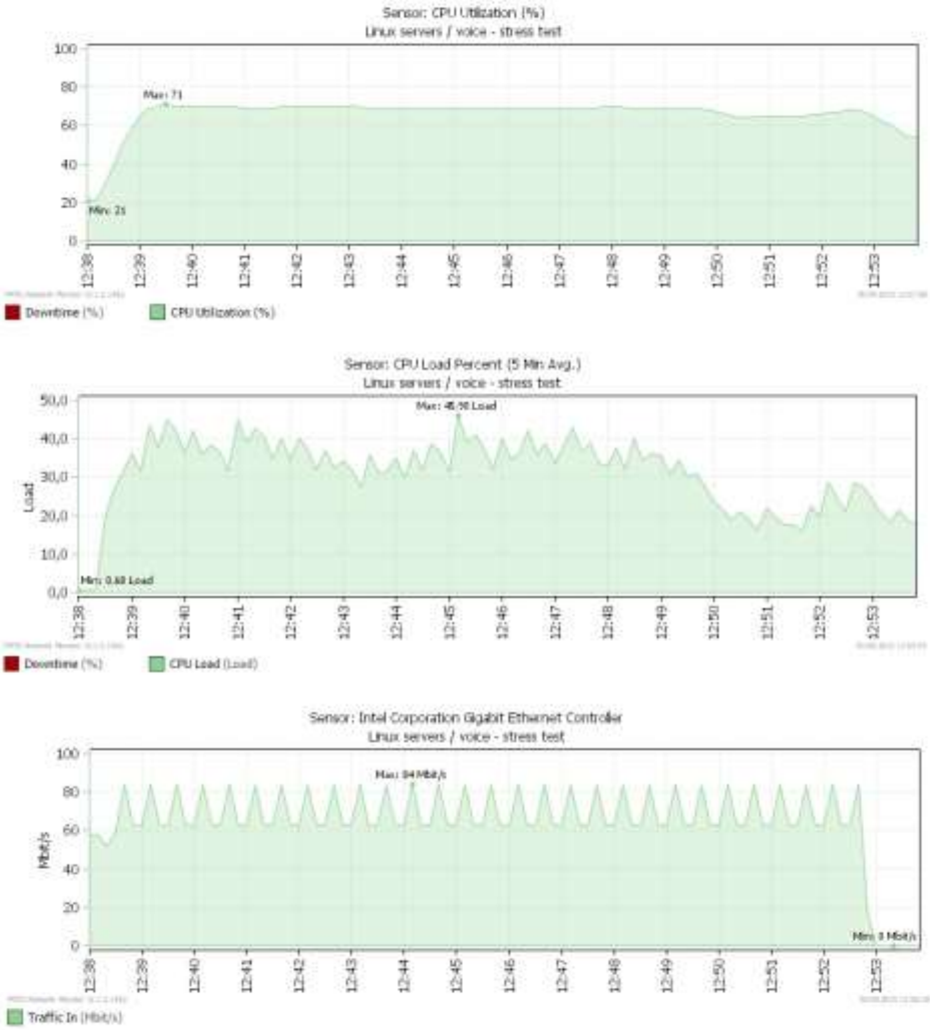
Table 3. Test 1 results

System	Average CPU utilization	Average system load
Asterisk	75%	9
FreeSWITCH	60%	3
Yate	66%	25

Source: test results from this paper

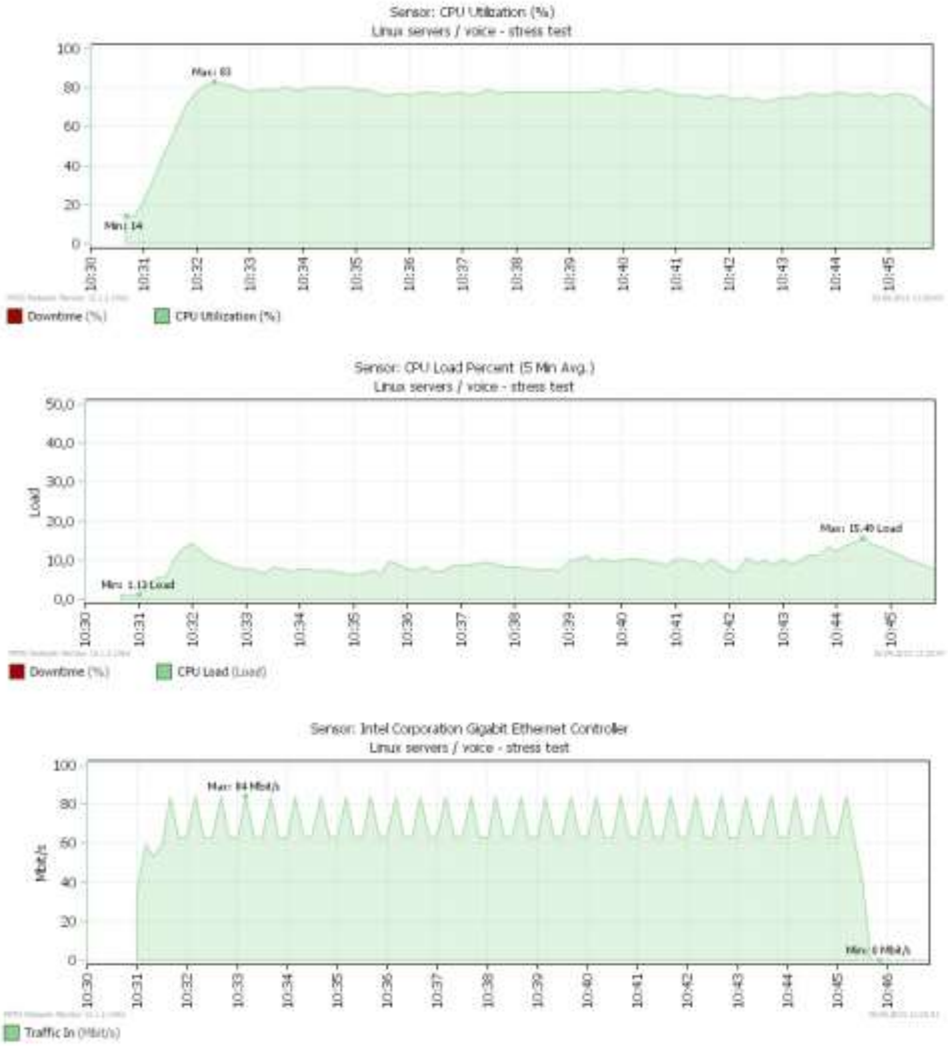
The following three figures depict performance details of conducted Test 1 for all three softswitches.

Figure 2. Yate performance, Test 1



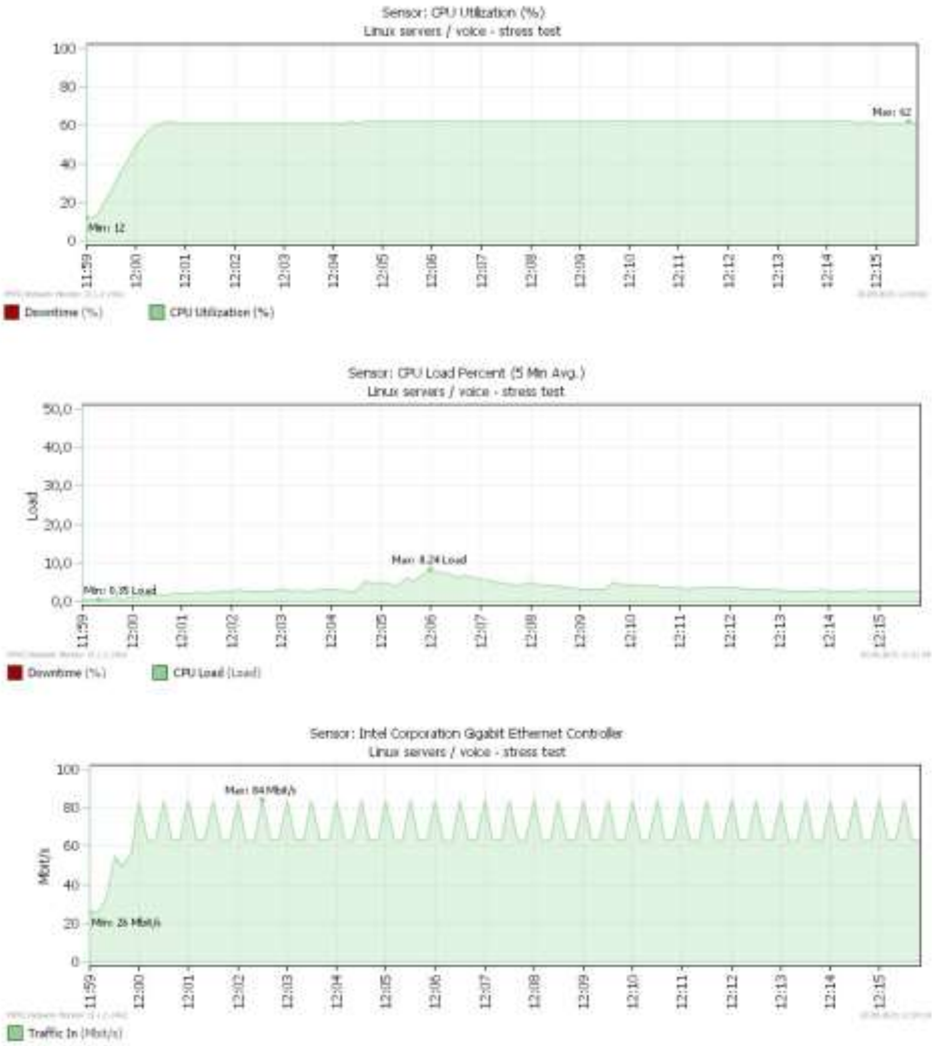
Source: test results from this paper

Figure 3. Asterisk performance, Test 1



Source: test results from this paper

Figure 4. FreeSWITCH performance, Test 1



Source: test results from this paper

Results of the Test 2 are presented via Table 4. FreeSWITCH seems to have a clear advantage in maximum concurrent calls scenario. Yate softswitch was second best in this context, but with significant straggle behind FreeSWITCH. This result could be argued with the fact that Yate developers wrote their own SIP stack (as did Asterisk developers), while FreeSWITCH uses well known, fast and stable Sofia-SIP stack – an open source project started in 2005 by Nokia Research Center (Sofia SIP Stack, 2015).

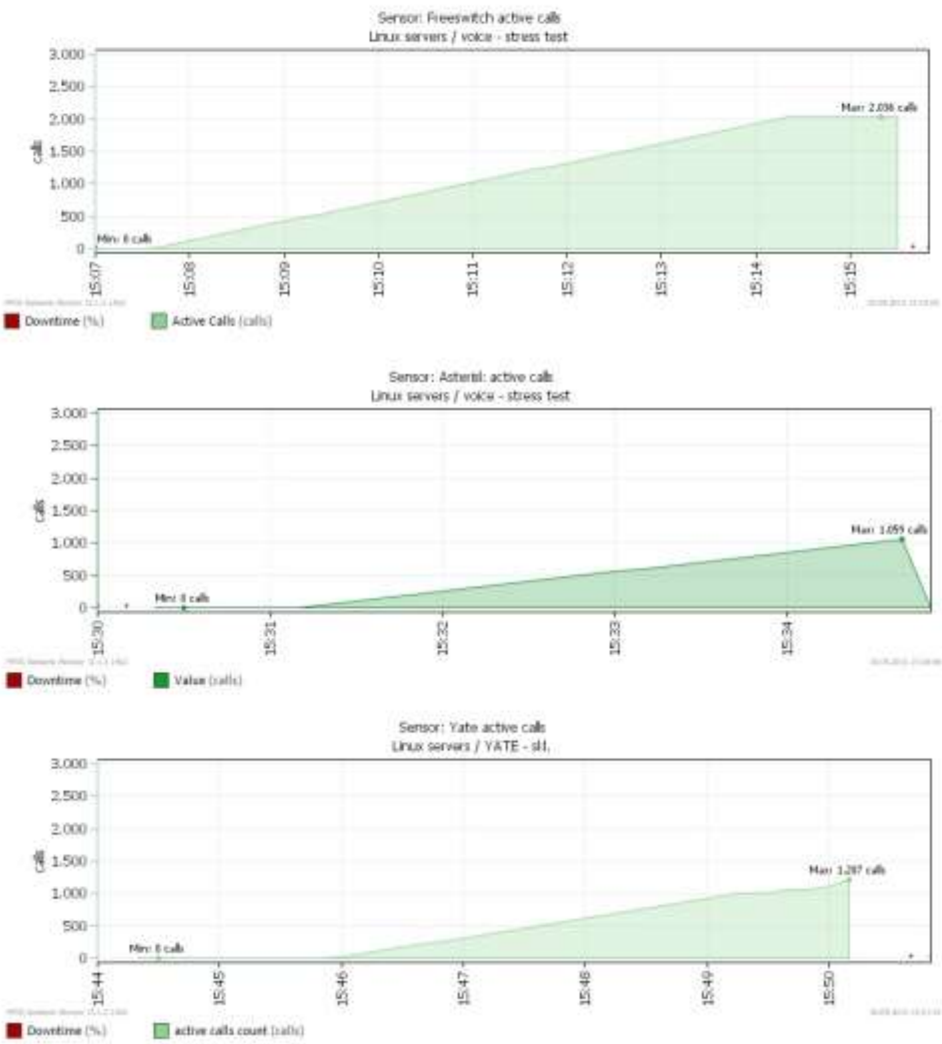
Table 4. Test 2 results

System	**Max concurrent calls**
Asterisk	1059
Freeswitch	2036
Yate	1207

Source: test results from this paper

Detailed results of the Test 2 scenario are presented in Figure 5 for all three softswitches.

Figure 5. Results of the Test 2 scenario



Source: test results from this paper

8. Conclusion

The main purpose of this paper was to present a performance comparison of three most popular open-source softswitch solutions, based on the amount of calls they can handle, and the amount of system resources needed for reasonably high softswitch load (800 concurrent calls). In practice, whenever a system is designed for such a high load, it is always made redundant and distributed, so the load is balanced across multiple softswitches. Nevertheless, it is important to choose a solution that is more stable, and that can handle more calls per server, as the difference is multiplied by the number of servers implemented.

By observing the test results established in this dedicated test scenario, the optimal system for high VoIP load should utilize FreeSWITCH. Asterisk being the lowest scoring in all the test scenarios can be argued with its purpose as a software – written mainly to replace proprietary PBX systems, it can be used as a softswitch up to a certain load and at the expense of inefficient use of hardware resources. FreeSWITCH and Yate were designed for enterprise (telecom) softswitch environment, which is consistent with the test results.

References

1. Features. http://www.asterisk.org/get-started/features (24.09.2015.).
2. Gayraud, R.; Jacques, O. Welcome to SIPp. http://sipp.sourceforge.net/ (23.09.2015.).
3. Get Started: What is Asterisk?. http://www.asterisk.org/get-started (24.08.2015.).
4. Modules. https://freeswitch.org/confluence/display/FREESWITCH/Modules (25.09.2015.).
5. RFC 3261, SIP: Session Initiation Protocol. http://www.rfc-base.org/txt/rfc-3261.txt (15.09.2015.).
6. Segec, P.; Kovacikova, T. (2011) “A Survey of Open Source Products for Building a SIP Communication Platform”. *Advances in Multimedia*, vol. 2011(5), 1-21.
7. Sofia SIP Stack. https://freeswitch.org/confluence/display/FREESWITCH/Sofia+SIP+Stack (28.09.2015.).
8. Specifications. https://freeswitch.org/confluence/display/FREESWITCH/Specifications (28.09.2015.).
9. Spencer, M.; Allison, M.; Rhodes, C. (2003). The asterisk handbook. 2. izd. Digium Inc.

10. The World's First Cross-Platform Scalable Free Multi-Protocol Soft Switch. https://freeswitch.org/ (25.09.2015.).
11. Van Meggelen, J.; Madsen, L.; Smith, J. (2007). Asterisk: the future of telephony. 2. izd. Sebastopol, O'Reilly.
12. What is Yate. http://docs.yate.ro/wiki/What_is_Yate (27.09.2015.).
13. Yate architecture. http://docs.yate.ro/wiki/Yate_architecture (29.09.2015.).