Load test of IP PBX Asterisk installed on mid-size server

E.Anvaer, V.Dudchenko,

SoftBCom, Ltd. (www.softbcom.ru)

21.07.2014

Direct load test of IP PBX Asterisk on Intel Xeon E5506 Quad-Core CPU shows that it can handle up to 1600 concurrent calls.

IP PBX Asterisk is an outstanding sample of open source, free software, having proven reliability and effectiveness, widely used everywhere over the world, repeating Linux' success.

At the same time implementing Asterisk for wider purposes than simply PBX solution, say as call-centers kernel, in integrated B2C systems, in help-desks, service- desks and other business-critical structures creates some doubt concerning its scalability.

Common perception that Asterisk could be successfully used only in small installations, somewhat up to 300 subscribers, is one of the main restraining factors of wider implementing such an efficient system.

But we couldn't find any published evidence of such limitations concerning modern Asterisk releases and present day hardware devices. Probably the reasons of such a situation were some difficulties of real load emulation.

The load generation problem was resolved by implementing Loway Wombat Dialer - a system developed for telecasting. Very important is that it's natively integrated with Asterisk and can produce automated calls just out of the box. Designed for automating calls in our case it automated load generation, establishing as many connections as necessary for testing purposes.

Earlier in our tests of Asterisk installed on VM in Hetzner cloud service we have found out that even such a small installation can stand 70 concurrent call load (with solid margin): http://www.slideshare.net/vdudchenko/asterisk-cloud-installation-load-testing-eng .

In the new test the load emulating approach was very similar to that we used for cloud installation. The main difference is that the load source server and the tested server both were installed in local 100Mb network as we were not having necessary 100Mb internet connection in our disposal.

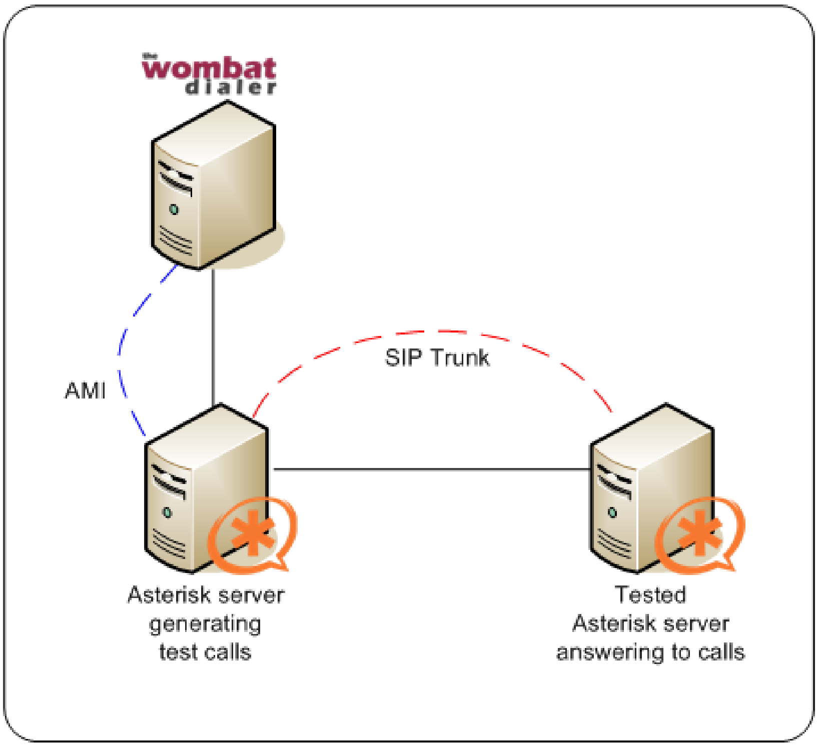
the wombat dialer
AMI
SIP Trunk
Asterisk server generating test calls
Tested Asterisk server answering to calls

Fig.1. Load testing scheme.

The server generating test calls (#1) is installed on a virtual machine (VMware). The server answering the test calls (#2, tested server) is a HW installation with CPU Intel Xeon E5506 Quad-Core, 16GB, RAM, 250 GB HDD, and installed Ubuntu (3.2.0-63-generic #95-Ubuntu). The Asterisk version is 11.8.1 (embedded in FreePBX 2.11.0.35).

The Loway Wombat Dialer is connected to Asterisk generating test calls. The mentioned Asterisk servers are connected via SIP trunk.

The load handled by tested server includes media delivered by RTP.

The load generation was realized in the following way: the calls from the server #1 were directed to the definite extension of the tested server, which then established connections and played music during 8 minutes – the interval covering the whole test duration. After connecting to this number the Asterisk #1 connected tested side to its own extension, from which music played for a short time, and then 900-second waiting interval was executed.

The tested server recorded all the calls (by the MixMonitor command).

This scenario gave the following results:

Number of concurrent calls	Peak CPU load*	Peak RAM load by Asterisk	Hearing quality
10	4%	Less than 3%	Excellent
20	10%	Less than 3%	Excellent
40	17%	Less than 3%	Excellent
50	25%	Less than 3%	Excellent
70	30%	Less than 3%	Excellent
90	36%	Less than 3%	Excellent
100	40%	Less than 3%	Excellent
200	65%	Less than 3%	Excellent
300	100%	Less than 3%	Excellent
409	123%	Less than 3%	Excellent
426	132%	Less than 3%	Excellent
432	126%	Less than 3%	Excellent
526	147%	Less than 3%	Excellent
636	170%	Less than 3%	Excellent
736	194%	Less than 3%	Excellent
835	220%	Less than 3%	Excellent
937	243%	Less than 3%	Excellent
1036	268%	Less than 3%	Excellent
1137	288%	Less than 3%	Excellent
1183	311%	Less than 3%	Excellent
1283	339%	Less than 3%	Excellent
1383	345%	Less than 3%	Excellent
1482	349%	Less than 3%	Excellent
1504	350%	Less than 3%	Excellent
1527	351%	Less than 3%	Excellent
1621	398%	Less than 3%	Excellent
1626	380%	Less than 3%	Excellent
1649	400%	Less than 3%	No connection

Fig.2. The CPU load value to concurrent calls number dependency

Annotation*: Maximum usage of 4-core CPU corresponds to 400% load (as to Ubuntu), 100% per core.

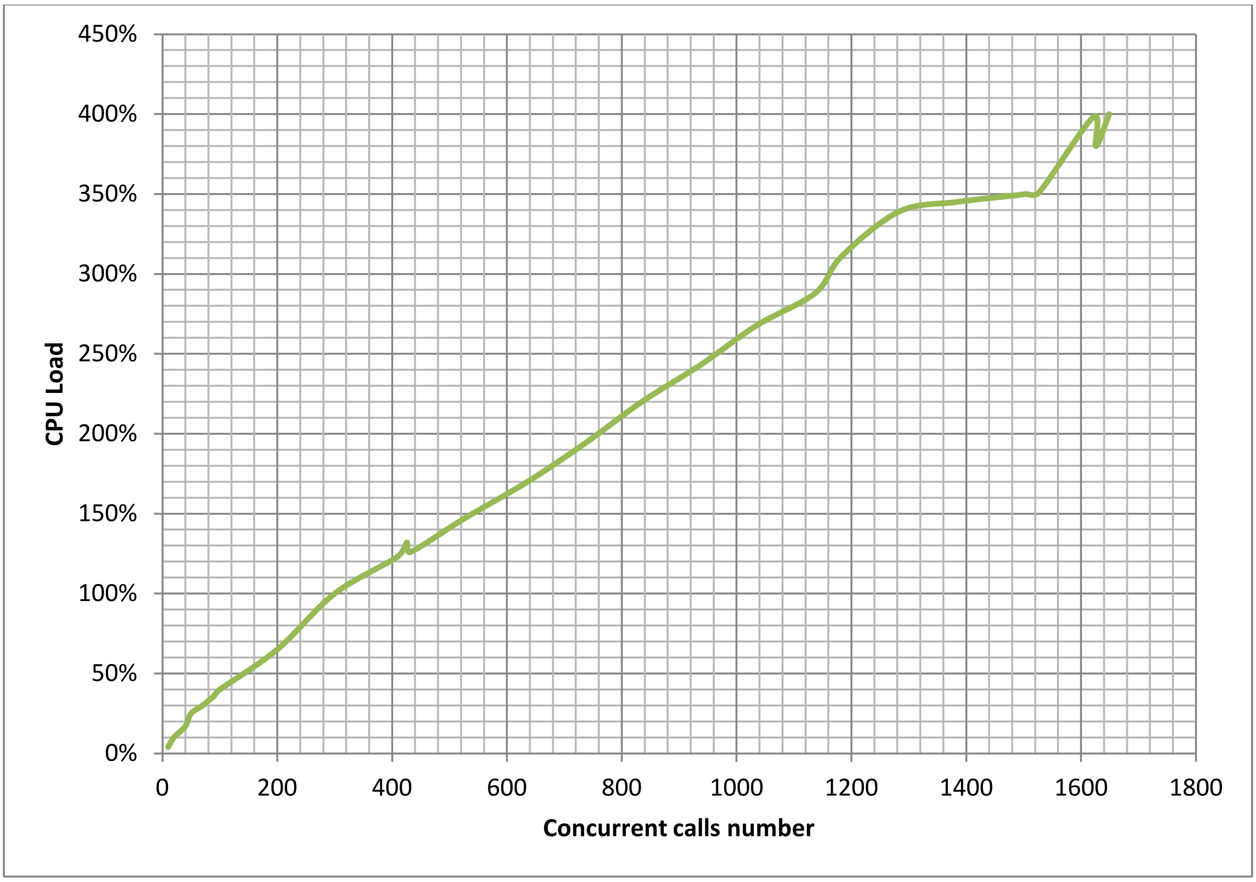

Fig. 3. CPU load value to concurrent calls number graph

The network monitoring wasn't installed, but due to our estimation 100Mb network was to stand more than 1500 concurrent calls. Taking into account the fact that the maximum CPU load in the test was achieved with excellent hearing kept up to 1626 concurrent calls, the network bandwidth didn't impose any limitation in this case.

The hearing quality was estimated subjectively, during an extra call executed in manual mode over the measured load level. "Excellent" means that no noticeable disturbance was experienced.

In the Asterisk cloud installation mentioned above we have been comparing CPU engagement for two load generation modes: for real manual calls and Wombat Dialer generated calls. For up to 20 concurrent calls the noted difference in CPU load for these two modes was of 1% - 2% range, which shows that applied testing methodology in general gives reasonable results.

The analysis and conclusions

The results presented above demonstrate that Asterisk running on Intel Xeon E5506 Quad-Core CPU can serve approximately up to 1600 concurrent calls.

The near- linear character of the recorded dependency (with 8 – 10% deviations) is noticeable. The deviations have been probably caused by accidental factors like OS service processes influence and moments of CPU load values capturing. The capturing moments have been chosen manually trying to catch maximum values on some measurement interval (app. 10 seconds).

But systematic errors are also possible. They could be created by the following factors:

1. The tested Asterisk has not been connected to real phone devices: all the calls were directed to one single extension, and there were no connections established to separate devices in separate points of network.
2. The media has been transferred mainly in one direction – from the tested server to the server #1, while it was continuous, not discrete. One could suppose that as a whole the load created in the test wasn't lower than those we have in real talks, but its character in some way differs.

It is not so simple to estimate the relevance level of the results above to real load, while we believe that applying reasonable margin we can implement the described Asterisk installation for up to 1000 concurrent calls.

The next important conclusion is that Asterisk can fully use the computing power of all available CPU cores, i.e. it has an effective mechanism of load balancing for multiple cores. So the real power of Asterisk installation could be easily increased if necessary by using more computing power (higher CPU frequency, more cores, and multi-CPU systems as well).

At the same time RAM consumption is very low and almost doesn't depend on load (some dependency exists, as it was show in Asterisk cloud installation load testing, but in general necessary RAM volume is much lower than what you get in real systems).

We didn't try to simulate all possible factors which could affect the load capability in real systems, e.g. transcoding, queues monitoring or applications integration facilities influence, etc.

But the described methodology could be used for sizing and load testing for any specific cases, with including specific factors of interest.

We would also recommend to implement CPU load monitoring for all the cases when planned load could reach 50% level of estimated maximum with generating alerts of overcoming some delimited threshold. This is the way to avoid any surprises caused by Asterisk overload.

The Loway Wombat Dialer was kindly presented for the test by Swiss company Loway. Designed for telecasting, it fits 100% to the test demands. Its easy use and power made this test possible.