HOW TO

REDUCE ASTERISK SYSTEM LOADS BY 70%

SANGOMA

This document expresses views and commentaries on telecom systems and architectures, and does not necessarily reflect precise product capabilities at time of publishing. This is due to the complexity of telecommunications systems design and the many interactive variables affecting system performance. Statements in this or any other Sangoma document about product performance are guidelines only. Actual performance results may vary based on specific application and system design. Please consult Sangoma to determine how this document can apply to your system design.

09/2009

Why is it important to be able to reduce system loads on an Asterisk® system?

The system load related to PSTN connectivity is a constant drain on system resources, much like a tax on your processor that you have to pay before you can do anything useful.

Any Asterisk system will always be more reliable under lower system load. But when trying to configure really large systems, that support 400 simultaneous calls or more, excessive system loads can force the use of additional servers. Likewise, minimally embedded PBX systems – intended to support only a few analog ports – can be employed to handle a T1/E1 and more, simply by reducing system load.

So how do you do it?

Zaptel/dahdi with Adjustable Chunk Size

By default, zaptel/dahdi runs with an 8 byte chunk size. This equates to an interrupt every 1ms from TDM hardware. The default chunk size of 8 bytes exists because of the constraints of running the software-based echo cancellation that was originally designed into the zaptel drivers. However, since Sangoma introduced hardware-based echo cancellation to the Asterisk community in 2005, this constraint has been eliminated except for the smaller analog installations that still use the zaptel/dahdi echo cancellation routines.

Thus, the developers of zaptel/dahdi have now introduced a configurable chunk size that can be taken advantage of by any card that incorporates hardware echo cancellation.

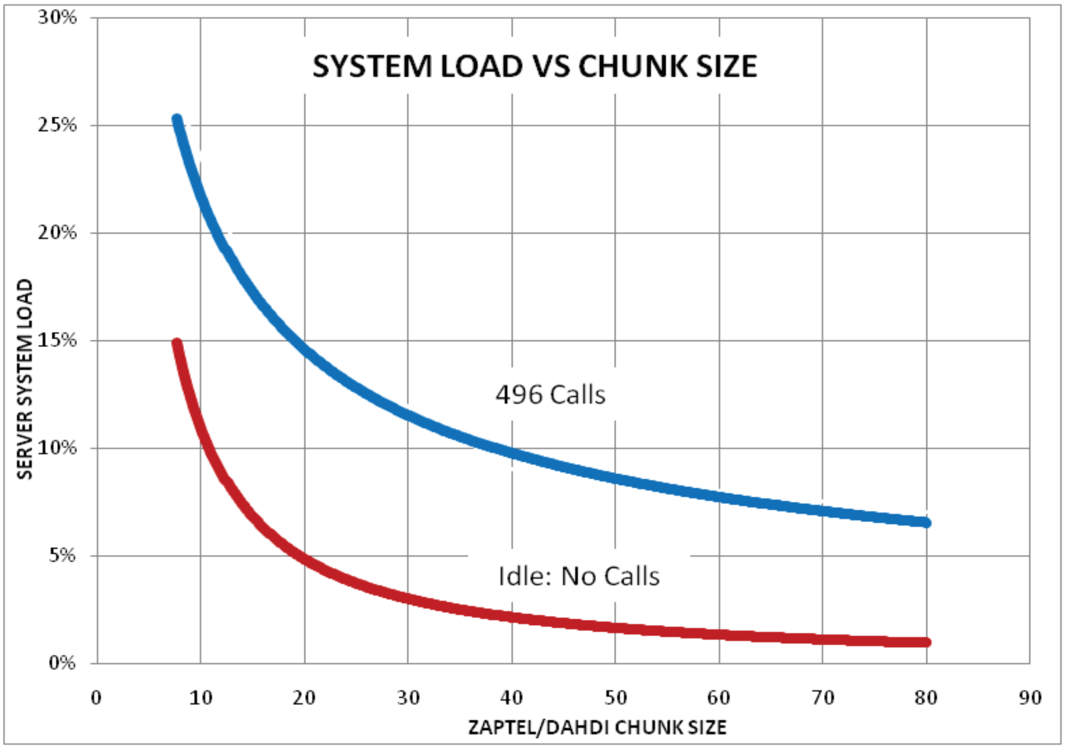
All non-analog Sangoma hardware is able to automatically detect the zaptel/dahdi chunk size and adjust accordingly. You can configure zaptel/dahdi for 8 bytes (1 ms - default), 16 bytes (2 ms), 40 bytes (5 ms) or 80 bytes (10 ms) chunk size.

What does that do for you?

It reduces the interrupt and context switching loads for the driver to a small fraction of its 1 ms value. Both the loads due to interrupt handling and context switches (changes from user space to kernel space) are reduced. The interrupts have to be handled whether or not calls are up, while the context switching load depends on the number of active calls.

What do you get?

The results below are for a machine with an Intel® Core™2 Quad CPU Q9550 @ 2.83GHz with 4 Gbytes of RAM. Two A108D (8-port digital card with hardware echo cancellation) cards provided a 496 call capacity (31 channels per span x 16 E1 spans). A small user program emulating the Asterisk zaptel/dahdi interface was used to exercise the driver. It avoided the variations of load in user space that would occur with Asterisk itself, and allowed the use of the channels, normally used for the D channel, to be used for "calls" adding it to the steady system load. Performance under real Asterisk is the same, except with greater variability.



The reductions are quite dramatic. At 496 calls, the system load was reduced from 26% to 7%, as the chunk size goes from 8 bytes (1 ms) to 80 bytes (10 ms). That's a decrease of over 70%. At idle, the reduction went from 15% down to a negligible 1%.

Nearly all the benefit is achieved by using a chunk size of 40 bytes, which is the maximum that gives reliable timing for meetme and music on hold. However, if you really want to go to the maximum, the cards all have an alternate timer that can be used for meetme synchronization.

What are the limitations?

1. This does not work for ANALOG and T1/E1 E&M wink. This is because zaptel/dahdi currently hard-codes the chunk size to 8 in some of the Analog/RBS timing routines.
2. You have to use a card with hardware echo cancellation. The software echo cancellation routines in zaptel/dahdi, or the OSLEC will not handle the larger chunk sizes well.
3. This does not apply for Sangoma's SMG/boost solutions that use chan_woomera. For alternate ways of achieving high performance with SMG/boost, please contact sales@sangoma.com.

How do you do it?

To configure zaptel/dahdi for higher chunk size:

```
./Setup install --zap-chunk=<CHUNK SIZE>
```

Where `CHUNK SIZE` is 8, 16 , 40 or 80, setup will patch zaptel/dahdi source for selected chunk size and will remove wct4xxx (Digium) driver from the zaptel/dahdi Makefile. The reason for the wct4xxx removal is that the wct4xxx driver does not support any chunk size other than 8 bytes (1 ms).

Once setup recompiles zaptel/dahdi and wanpipe drivers, there are no other configuration changes needed. Just start wanpipe and zaptel/dahdi and Asterisk:

```
wanrouter start
ztcfg –vvv
asterisk -c -r
```

To confirm that you are running with new zaptel/dahdi chunk size, run:

```
ifconfig w1g1
```

And confirm that the MTU = configured chunk size.
Please contact Sangoma for technical support at **techdesk@sangoma.com** or **+1 905 474 1990**.

The future

The zaptl/dahdi design is an amazing piece of work, but was never designed for large systems, or systems not as large, yet still operate on marginal software. There is – believe it or not – still considerable scope for further significant reductions in system load by changing the system architecture. Watch for additional information from Sangoma for lowering system loads using SMG/woomera on Asterisk, FreeSwitch, Yate and others.

Asterisk is a registered trademark of Digium Corporation.

Appendix
Machine Configuration

```
Machine Info:

cat /proc/cpu

[root@tesla ~]# cat /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 23
model name      : Intel(R) Core(TM)2 Quad CPU    Q9550  @ 2.83GHz
stepping        : 10
cpu MHz         : 2842.866
cache size      : 6144 KB
physical id     : 0
siblings        : 4
core id         : 0
cpu cores       : 4
fdiv_bug        : no
hlt_bug         : no
f00f_bug        : no
coma_bug        : no
fpu             : yes
fpu_exception   : yes
cpuid level     : 13
wp              : yes
flags         : fpu vme de pse tsc msr pae mce cx8 apic mtrr pge mca cmov pat pse36 clflush
dts acpi mmx fxsr sse sse2 ss ht tm pbe nx lm constant_tsc pni monitor ds_cpl vmx smx est
tm2 cx16 xtpr lahf_lm
bogomips        : 5688.50

processor       : 1
vendor_id       : GenuineIntel
cpu family      : 6
model           : 23
model name      : Intel(R) Core(TM)2 Quad CPU    Q9550  @ 2.83GHz
stepping        : 10
cpu MHz         : 2842.866
cache size      : 6144 KB
physical id     : 0
siblings        : 4
core id         : 1
cpu cores       : 4
fdiv_bug        : no
hlt_bug         : no
f00f_bug        : no
coma_bug        : no
fpu             : yes
fpu_exception   : yes
cpuid level     : 13
wp              : yes
flags         : fpu vme de pse tsc msr pae mce cx8 apic mtrr pge mca cmov pat pse36 clflush
```

```
dts acpi mmx fxsr sse sse2 ss ht tm pbe nx lm constant_tsc pni monitor ds_cpl vmx smx est
tm2 cx16 xtpr lahf_lm
bogomips        : 5685.57

processor       : 2
vendor_id       : GenuineIntel
cpu family      : 6
model           : 23
model name      : Intel(R) Core(TM)2 Quad CPU    Q9550  @ 2.83GHz
stepping        : 10
cpu MHz         : 2842.866
cache size      : 6144 KB
physical id     : 0
siblings        : 4
core id         : 2
cpu cores       : 4
fdiv_bug        : no
hlt_bug         : no
f00f_bug        : no
coma_bug        : no
fpu             : yes
fpu_exception   : yes
cpuid level     : 13
wp              : yes
flags         : fpu vme de pse tsc msr pae mce cx8 apic mtrr pge mca cmov pat pse36 clflush
dts acpi mmx fxsr sse sse2 ss ht tm pbe nx lm constant_tsc pni monitor ds_cpl vmx smx est
tm2 cx16 xtpr lahf_lm
bogomips        : 5685.59

processor       : 3
vendor_id       : GenuineIntel
cpu family      : 6
model           : 23
model name      : Intel(R) Core(TM)2 Quad CPU    Q9550  @ 2.83GHz
stepping        : 10
cpu MHz         : 2842.866
cache size      : 6144 KB
physical id     : 0
siblings        : 4
core id         : 3
cpu cores       : 4
fdiv_bug        : no
hlt_bug         : no
f00f_bug        : no
coma_bug        : no
fpu             : yes
fpu_exception   : yes
cpuid level     : 13
wp              : yes
flags         : fpu vme de pse tsc msr pae mce cx8 apic mtrr pge mca cmov pat pse36 clflush
dts acpi mmx fxsr sse sse2 ss ht tm pbe nx lm constant_tsc pni monitor ds_cpl vmx smx est
tm2 cx16 xtpr lahf_lm
bogomips        : 5685.60

cat /proc/memory
[root@tesla ~]# cat /proc/meminfo
MemTotal:      3369408 kB
```

```
MemFree:        880972 kB
Buffers:        593672 kB
Cached:        1377928 kB
SwapCached:          0 kB
Active:         910204 kB
Inactive:      1285396 kB
HighTotal:     2489856 kB
HighFree:       868080 kB
LowTotal:       879552 kB
LowFree:         12892 kB
SwapTotal:      923728 kB
SwapFree:       923728 kB
Dirty:              80 kB
Writeback:           0 kB
AnonPages:      223984 kB
Mapped:          67616 kB
Slab:           266820 kB
PageTables:       5708 kB
NFS_Unstable:        0 kB
Bounce:              0 kB
CommitLimit:   2608432 kB
Committed_AS:   410352 kB
VmallocTotal:   114680 kB
VmallocUsed:     45960 kB
VmallocChunk:    64500 kB
HugePages_Total:     0
HugePages_Free:      0
```

Typical vmstat outputs
Chunk size= 8 bytes, no load

```
procs -----------memory---------- ---swap-- -----io---- --system-- -----cpu------
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
0  0      0 798684 591196 1459432    0    0     0     0 3027  514  0 14 86  0  0
```

Chunk size= 8 bytes, 496 channels

```
procs -----------memory---------- ---swap-- -----io---- --system-- -----cpu------
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
4  0      0 625020 589796 1459612    0    0     0     0 3071 65015 12 26 62  0  0
```

Chunk size= 80 bytes, no load

```
procs -----------memory---------- ---swap-- -----io---- --system-- -----cpu------
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
0  0      0 882400 593672 1377936    0    0     0     0 1226  446  0  1 99  0  0
```

Chunk size= 80 bytes, 496 channels

```
procs -----------memory---------- ---swap-- -----io---- --system-- -----cpu------
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
6  0      0 706272 591500 1377936    0    0     0     0 1209 46195 12  6 81  0  0
```

About Sangoma

Sangoma is the premium provider of voice and data connectivity components for software-based communication applications. Sangoma's data cards, voice cards, gateways and connectivity software are used in leading PBX, IVR, Contact Center and data communication applications worldwide. The product line represents a comprehensive toolset for deploying cost-effective, powerful, and flexible software communication applications. Founded in 1984, Sangoma Technologies Corporation is publicly traded on the TSX Venture Exchange (TSX VENTURE:STC). For more information, please visit: www.sangoma.com