SANGOMA TECHNOLOGIES

The State of Asterisk

Matthew Fredrickson

Asterisk Project Lead

February 3rd 2019

Nice new company logo, right?

Asterisk - Last Year in Review

Asterisk 15.x - 6 bug-fix releases - current is 15.7.1

Asterisk 14.x - 8 security releases - current (and final!) is 14.7.7

Asterisk 13.x - 6 bug-fix releases - current is 13.24.1

3300 merged code reviews (across all branches)


drum roll
please...

Asterisk 16.0.0 was released!

1000 commits (close to 3 per day)

72 individual contributors

(actually we're up to 16.1.1)

Standard Release vs LTS

Standard Releases:

1 year bug fixes

1 additional year of security fixes

Examples: 12.x, 14.x, 15.x

LTS:

4 years of bug fixes

1 additional year of security fixes

Examples: 11.x, 13.x

A little bit of history

Historically, every other release (odd numbers) are LTS releases

Asterisk 15 broke that trend (whoops)

4 years since last LTS

Asterisk 13 (LTS) has become a really solid branch to work from.

What about 16?

Multistream extensions in the core of Asterisk have been remarkably stable

Asterisk 16 is an LTS!

4 years of bug fixes (2022)

1 additional year of security fixes (2023)

Asterisk 16 - What’s New?

- WebRTC Video Quality Improvements
- WebRTC API Additions
- Improving Performance in chan_pjsip

Asterisk 16 - Improving Video Resilience

Why?

- Video is a lot more sensitive to packet loss than audio
- The loss of a single packet can have significant impact on the quality of the video stream
- In Asterisk 15 we took the sledgehammer approach, and just requested a new full video frame every time we detected packet loss
- Browsers have incorporated a number of better technologies to help combat packet loss scenarios, particularly with regards to video

Asterisk 16 - Improving Video Resilience

NACK:

- RTCP-FB message

- Exactly what is sounds like - a proactive negative acknowledgement RTCP packet from the receiver of an RTP stream.

- NACK can be for one or more previous RTP sequence numbers that may be missing.

Asterisk 16 - Improving Video Resilience

Receiver Estimated Maximum Bitrate (REMB):

- Receiver tries to estimate its maximum received bitrate.
- Receiver tries to send this calculation to the remote video sender so that it can reduce or increase its video encoding bitrate accordingly
- Asterisk gathers these reports and tries to intelligently interpret them and inform senders to help them adjust their video encoding bandwidth.
- RTCP-FB message
- Support is negotiated in SDP

Asterisk 16 - Enhanced Messaging

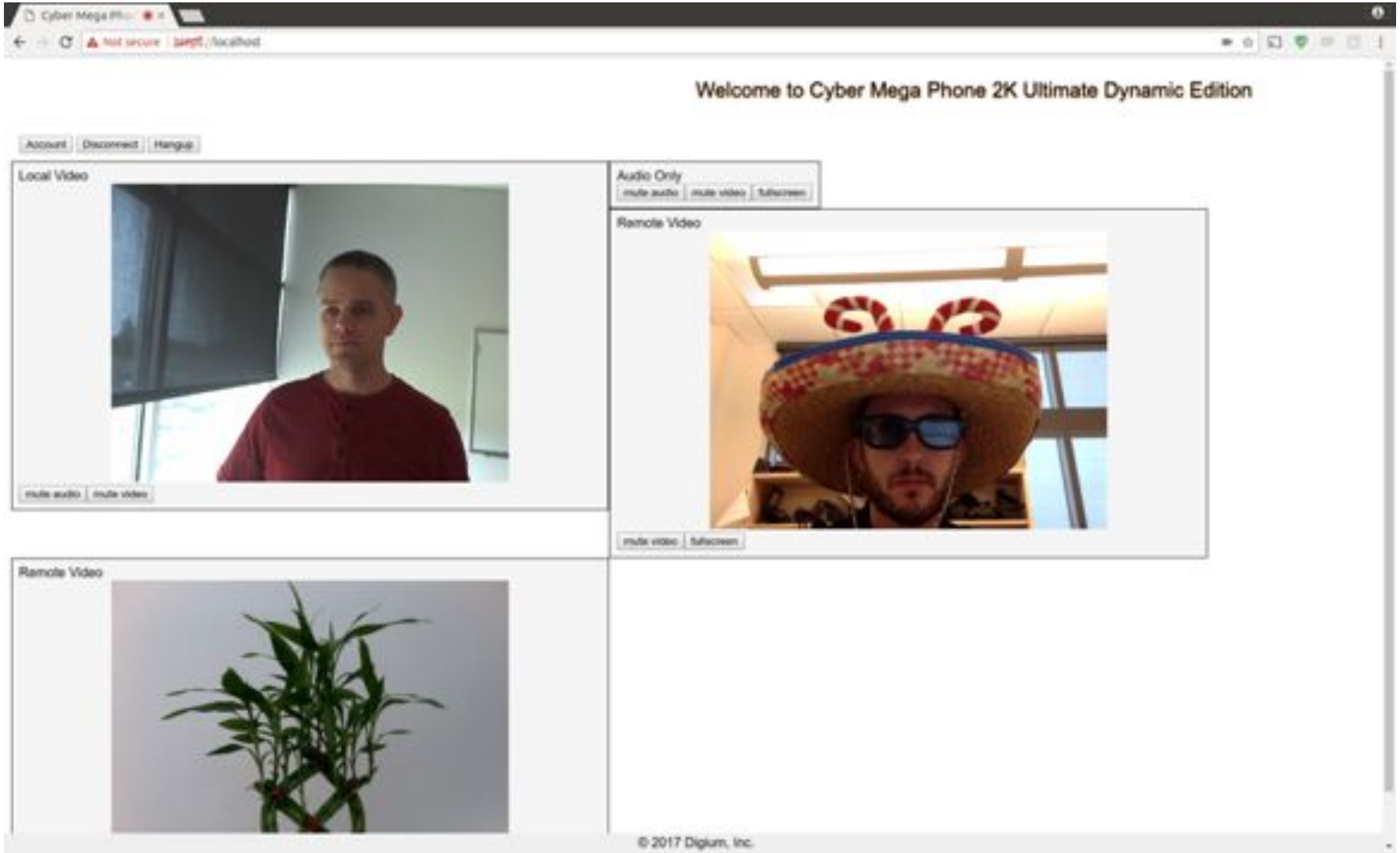
Background:

- Updated internal APIs in Asterisk to support multiple audio/video streams per call

- Built prototype web application of an SFU using browser based SIP stack

- Added ARI (Asterisk REST Interface) support for new SFU style bridge mixing

Asterisk 16 - Enhanced Messaging



Asterisk 16 - Enhanced Messaging

Problem:

- Realized that it was challenging (from the browser) to correlate incoming streams on a call to participant metadata

- TL&DR: No good way to figure out the CallerID or other call related metadata programmatically of a particular video stream coming to the browser

Asterisk 16 - Enhanced Messaging

- ConfBridge now has the ability to send conference level participant info events to participant in the conference.

- Allows browsers to figure out which CallerID information (and other information) to associate with the video boxes of the people in the conference.

- Contributes to a much richer and more complete client experience

Asterisk 16 - Enhanced Messaging Support

- Confbridge now supports participant to participant message passing
- Includes support for MIME type and other features
- Primary endpoint protocol right is in-dialog SIP MESSAGE
- Allows participants to be able to “chat” with each other
- Allows participants to be able to send any other programmatic data to enrich the conference experience

SANGOMA

PJSIP Performance Improvements

- Good talk at Astricon last year about performance differences between chan_sip and chan_pjsip

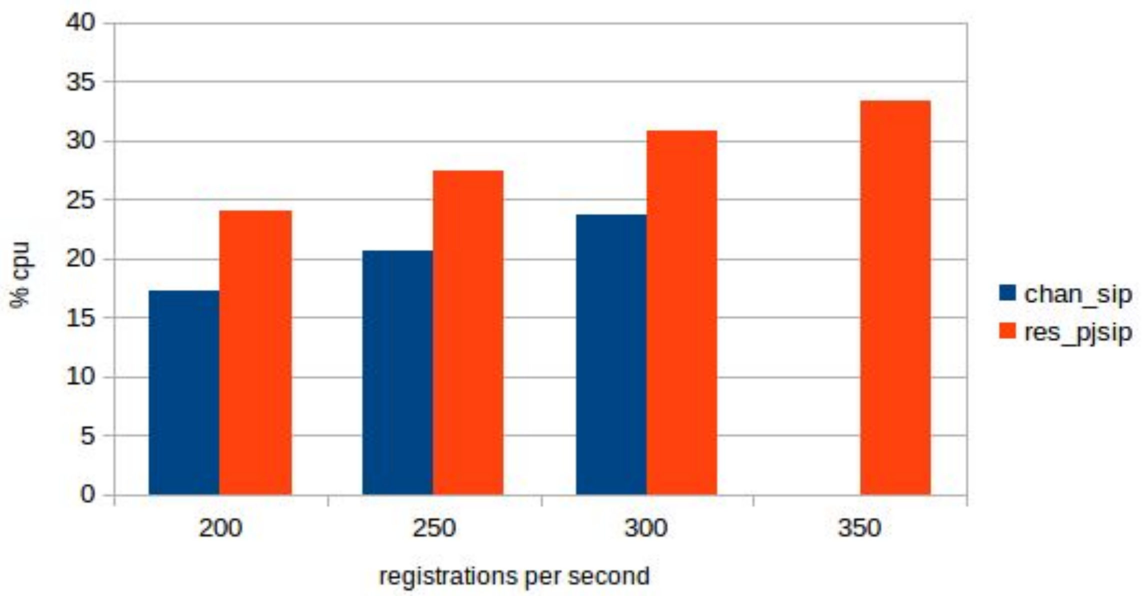
- In many cases, chan_pjsip performed better

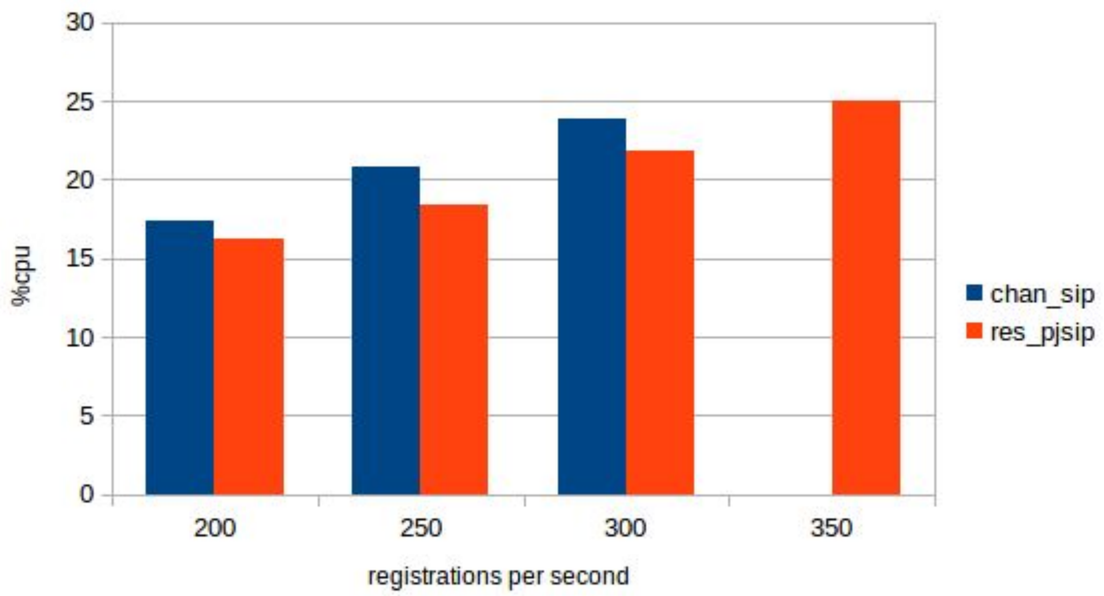
- There were a few cases where chan_sip had better performance that we wanted to address.

- Not significant difference - in the testing they did, it was on the order of one to two simultaneous calls more in some specific use cases.

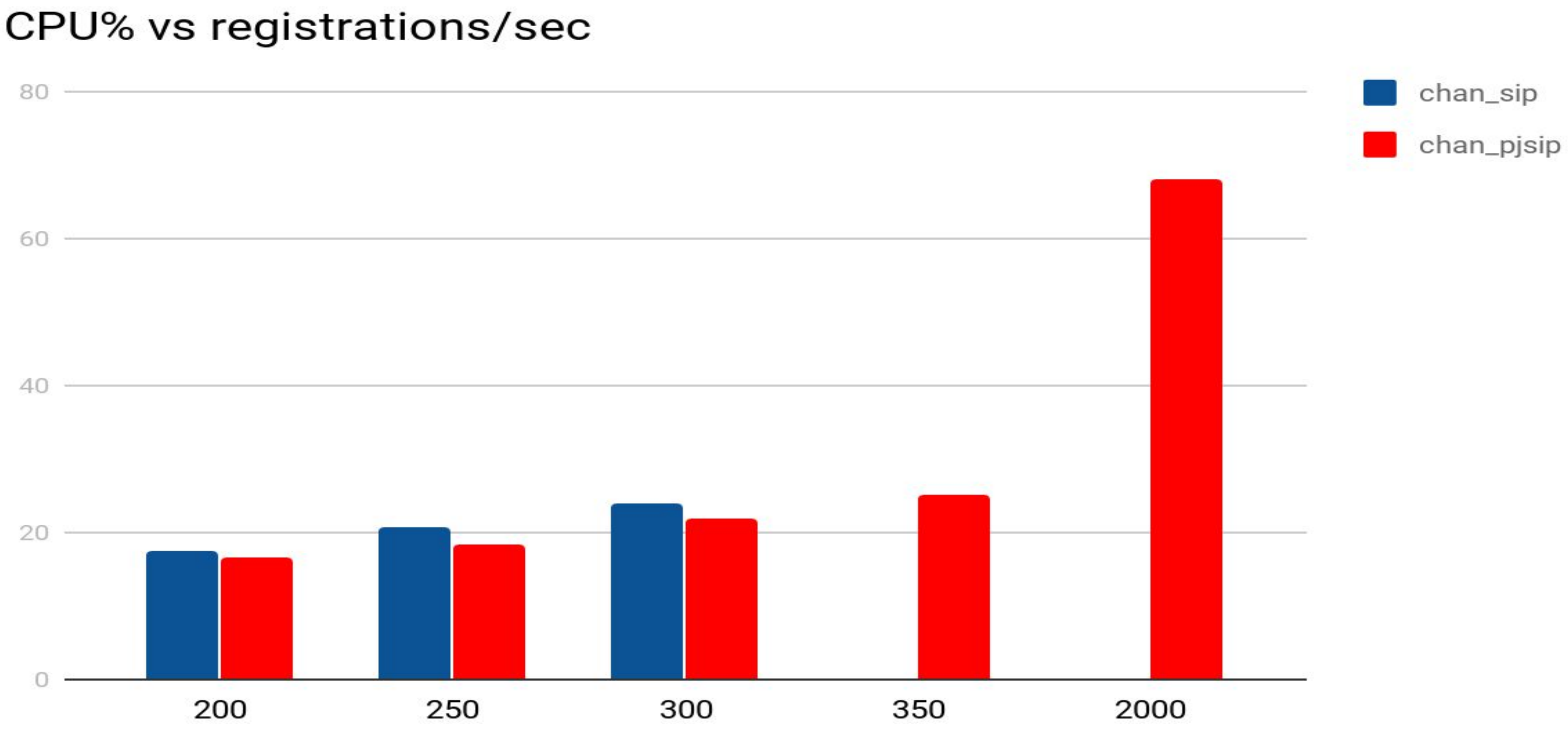
PJSIP Performance Improvements

Inbound SIP registration performance pre-16 vs 16.0.0:



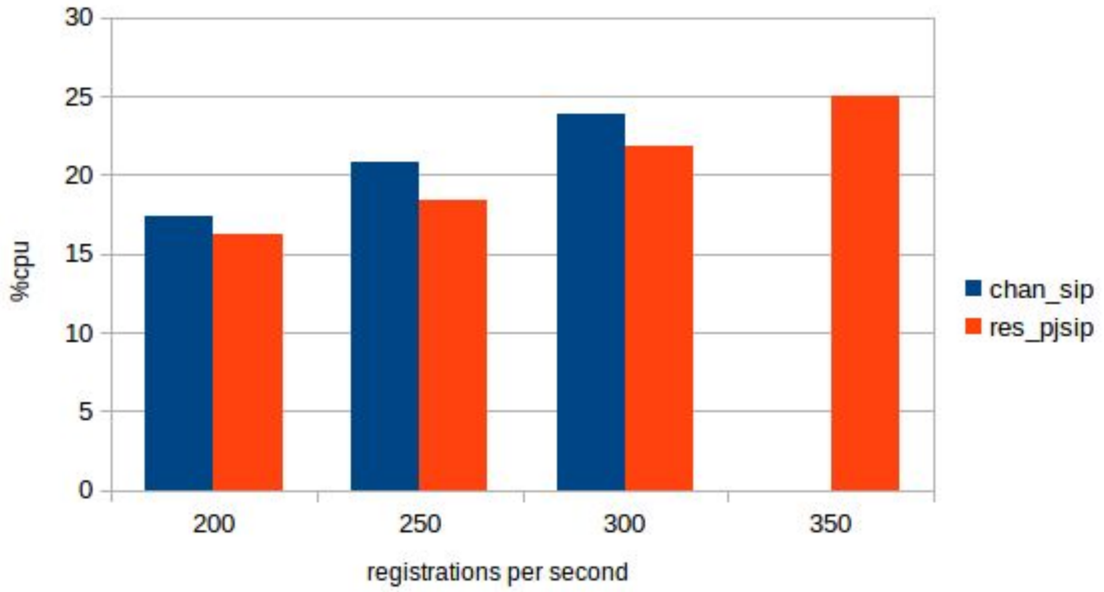


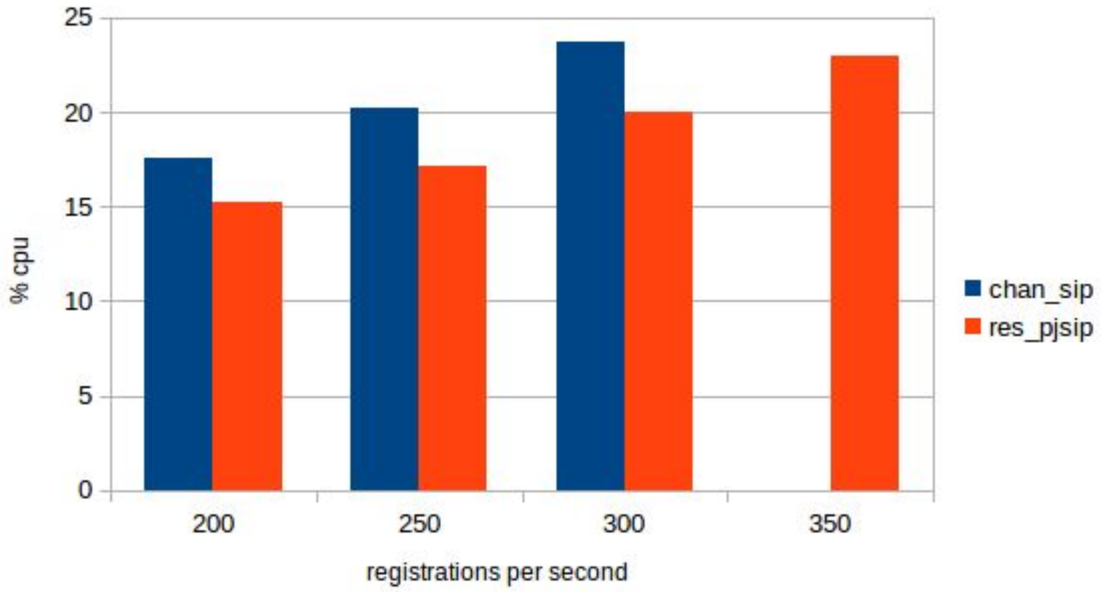
PJSIP Performance Improvements (there’s more)



PJSIP Performance Improvements (there's more)

Inbound SIP registration performance 16.0.0 vs 16.1.0:





Preview of what’s next (Post 16.0.0/17.x)

More performance improvements (Stasis, PJSIP)

ARI event filter support

ARI without the dialplan

Reminder

- Asterisk 14 went EOL at the end of September. No more security fixes or bug fix fixes. It's time to move forward.

- Asterisk 15 has recently graduated to security fix only mode - You have less than a year to move forward to 16 if you're relying on 15.x features.

- Keep track of what's happening in newer (non-LTS) major releases of Asterisk - if you don't, you potentially can experience big surprises when you move forward.

THANK YOU!

Follow me @creslin287 on Twitter

SANGOMA