Scaling Up With OpenSIPS

By
Paresh Lukka
Chief Architect and VP of Technology
Crexendo Business Solution

Agenda

Discuss how we built a high performance, scalable and Feature Rich Communication Platform using OpenSIPS

OpenSIPS Performance

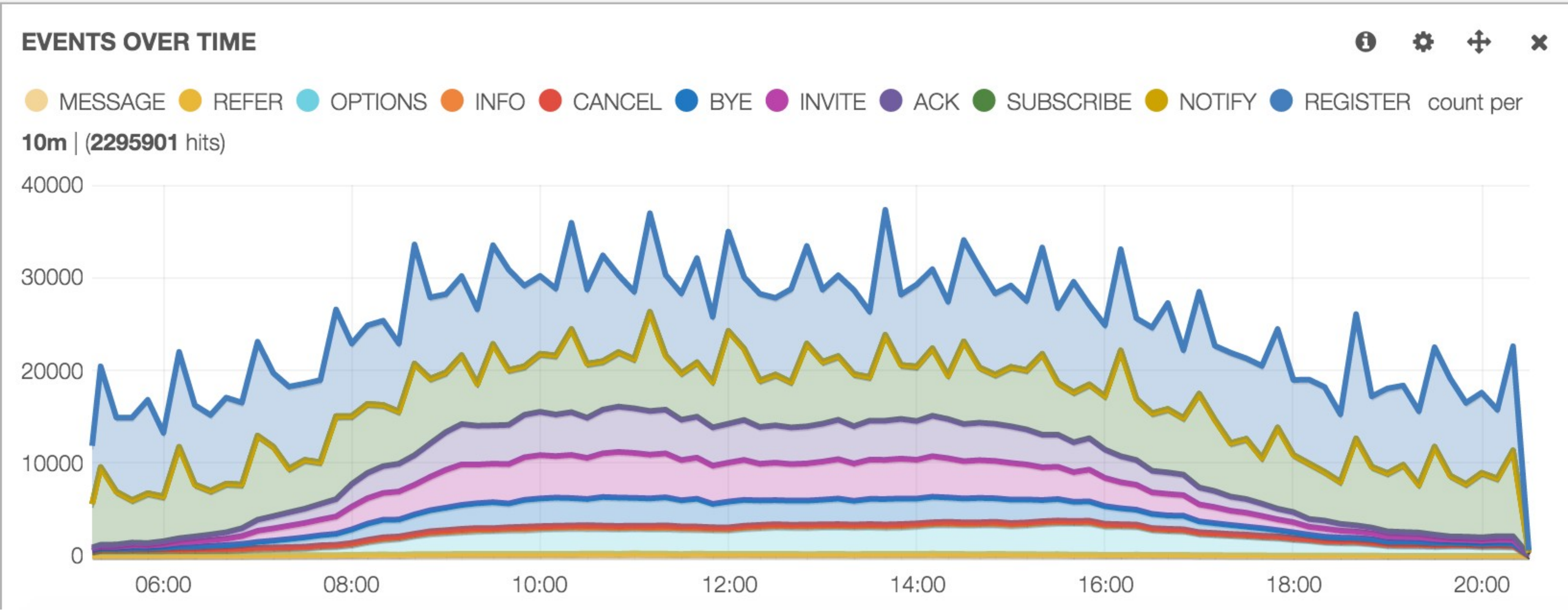
Each OpenSIPS Node:

- Quad Core/8GB Memory
- Simultaneous Requests Handled by one node :
  - 5,000 REGISTER + 3,200 NOTIFY + 1,600 ACK + 1,200 INVITE + 600 BYE + 600 CANCEL + 300 INFO + 100 REFER = 12,600 requests/minute

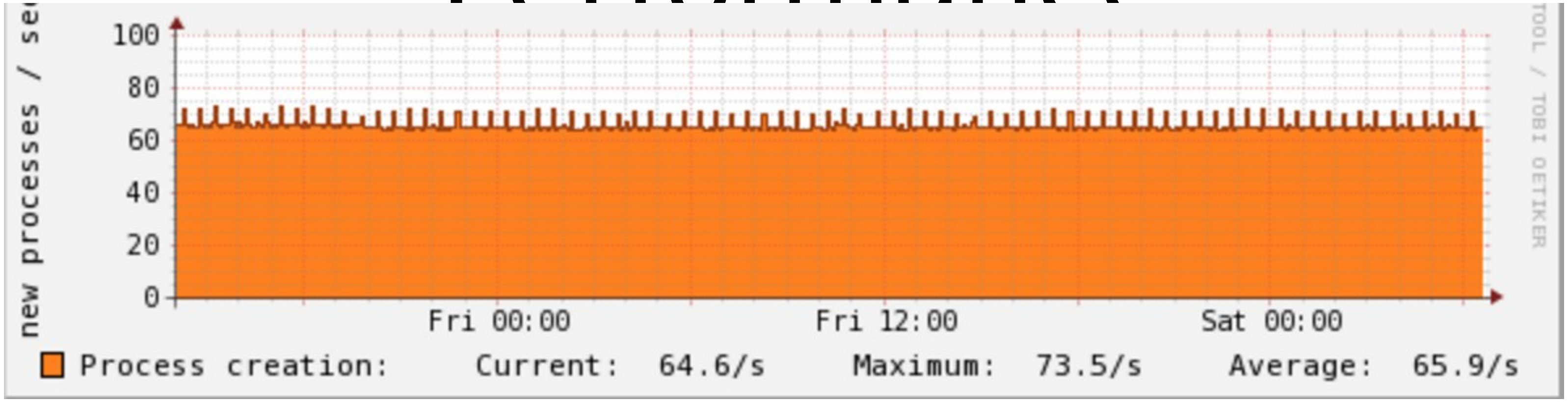
OpenSIPS Performance

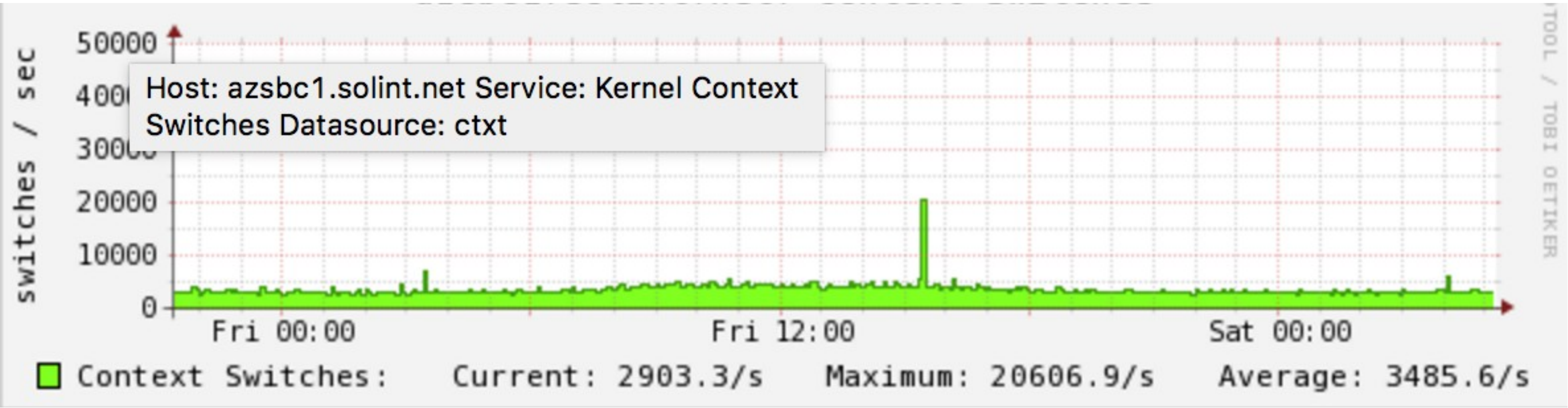
- Process Creation: Average 65/sec
- Kernel Context Switching: Average 3,400/sec
- Average Memory Used: 3G
- Average CPU Used: < 3% at peak load

OpenSIPS - Daily Traffic Pattern

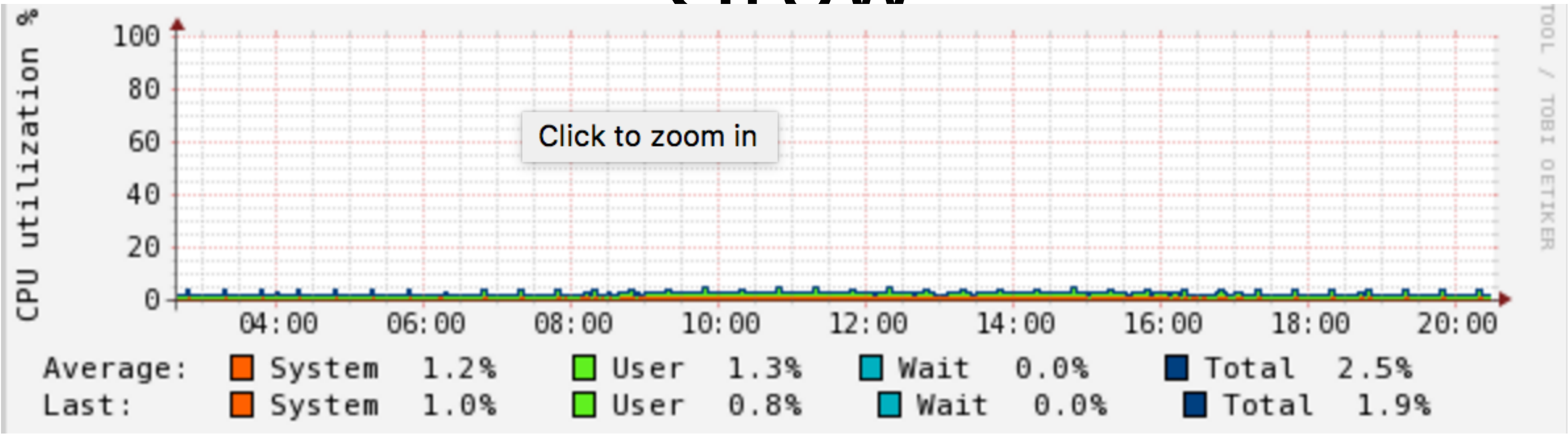


OpenSIPS – Steady Performance

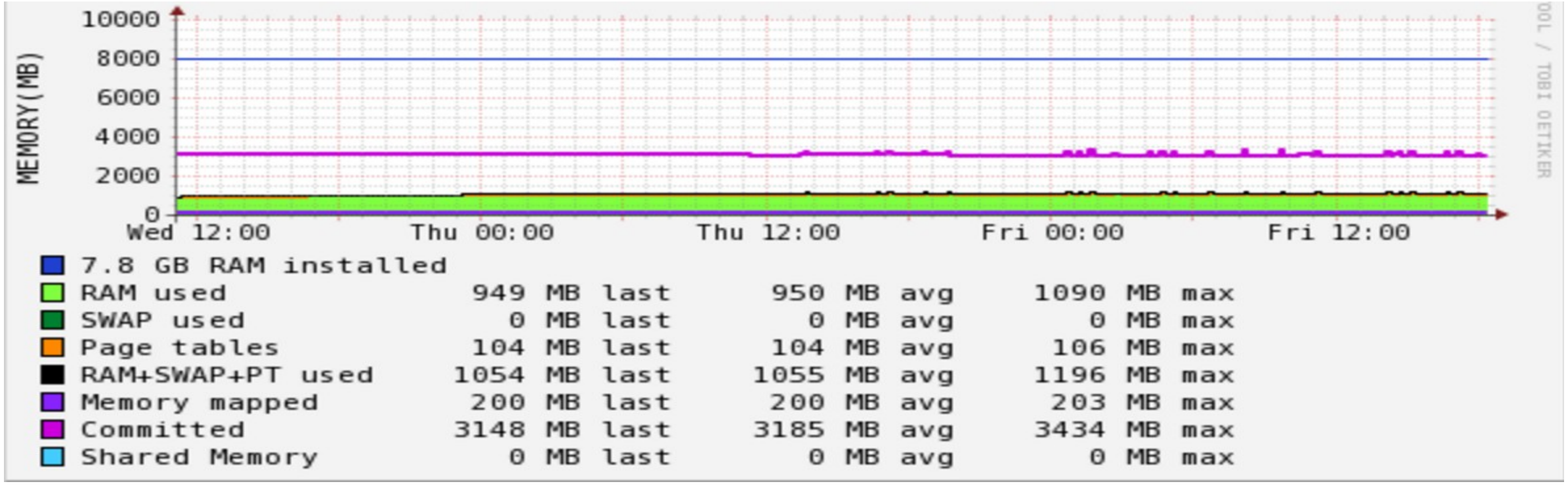



Host: azsbc1.solint.net Service: Kernel Context Switches Datasource: ctxt

OpenSIPS – A Lot of Room to Grow



```
Average:  🟧 System  1.2%   🟩 User  1.3%   🟦 Wait  0.0%   ■ Total  2.5%
Last:     🟧 System  1.0%   🟩 User  0.8%   🟦 Wait  0.0%   ■ Total  1.9%
```

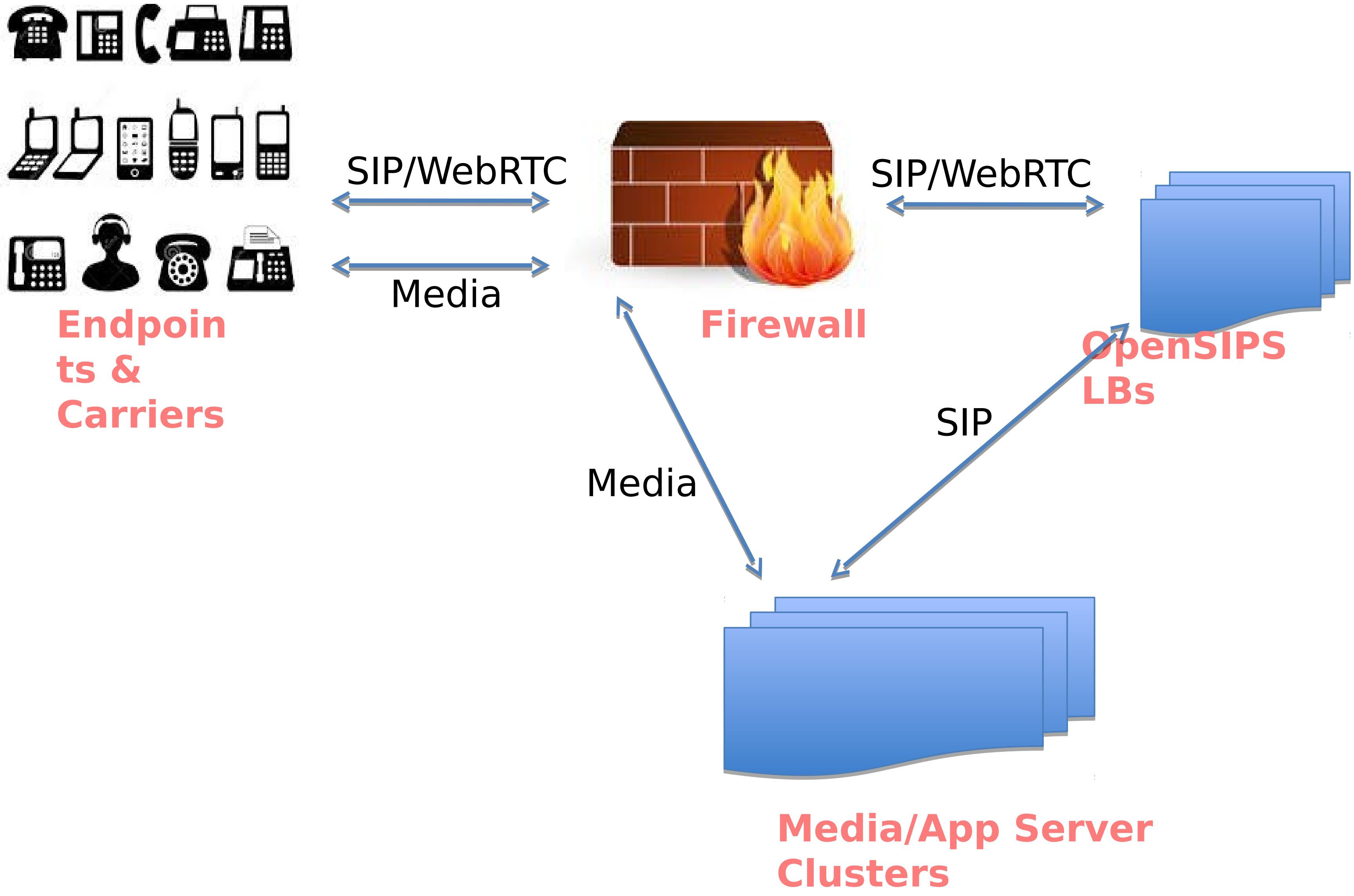


```
🟦 7.8 GB RAM installed
🟩 RAM used            949 MB last     950 MB avg    1090 MB max
🟩 SWAP used             0 MB last       0 MB avg       0 MB max
🟧 Page tables         104 MB last     104 MB avg     106 MB max
■ RAM+SWAP+PT used   1054 MB last    1055 MB avg    1196 MB max
🟪 Memory mapped       200 MB last     200 MB avg     203 MB max
🟪 Committed          3148 MB last    3185 MB avg    3434 MB max
🟦 Shared Memory         0 MB last       0 MB avg       0 MB max
```

The Minimalistic Approach

- Signaling only – No RTP Proxy
- Registrar
- Topology Hiding
- NAT Helper, NAT Pinging
- TLS Termination
- HA/Failover + Loadbalancing
- Dialog Caching for sticky sessions
- Pike
- Router for Different Application Clusters

Architecture



Endpoin
ts &
Carriers

SIP/WebRTC

Media

Firewall

SIP/WebRTC

OpenSIPS
LBs

SIP

Media

Media/App Server
Clusters

Advantages

- OpenSIPS never a point of failure – Distributed Endpoints, Load Balanced Media/App Servers
- Take advantage of full feature set offered by media/app servers and not involve OpenSIPS with complicated app logic/media handling
- Use OpenSIPS as terminator of various protocol/transport and unify everything behind it – Media/App servers don’t need to implement various protocols (like WebRTC) themselves
- Tremendous horizontal scalability – no need for unnecessary POPs
  - Currently one OpenSIPS HA pair can handle 25 media/app servers behind it and a lot more room to grow – not CPU/memory bound at all
- Very easy maintenance/upgrade cycles. OpenSIPS dispatcher mechanism helps drying the traffic to media/app servers. Essentially zero downtime.

Limitations

- I/O
- Log Rotation
- Ping to TLS/TCP endpoint – CPU spikes
- Configurability of Pike per domain absent