Flock of birds

Scaling Route Servers Easily

Antonio M. Moreiras – IX.br

CGI.br is the Brazilian Internet Stering Committee

Multistakeholder Committe - Internet Governance in Brazil

The CGI.br is comprised of members from the government, the corporate sector, the third sector and the academic community, and as such constitutes a unique Internet governance model for the effective participation of society in decisions involving network implementation, management and use. Based on the principles of multilateralism, transparency and democracyl

Brasil Internet Exchange

ix.br

Brazilian Network Information Center
- civil non-profit corporation
- executive arm of CGI.br

registro.br
- ccTLD '.br'
- Brazilian NIR

cert.br
- security incident response
- CSIRTs fostering and coordenation

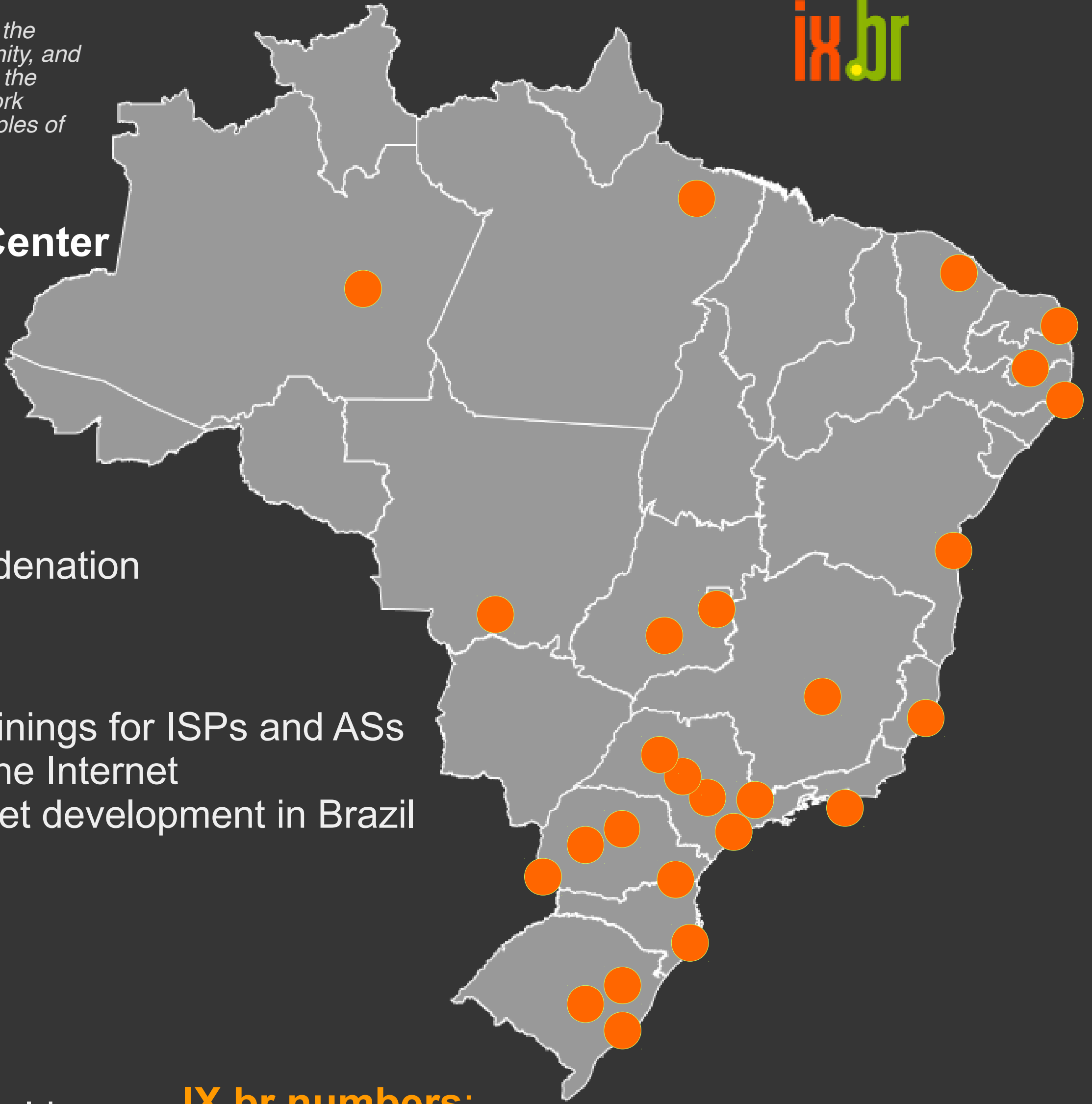
cetic.br
- ICT indicators

ceptro.br
- IPv6 and best practices trainings for ISPs and ASs
- quality measurements on the Internet
- projects to foster the Internet development in Brazil

ix.br
- Internet Exchanges

ceweb.br
- Web related projects

W3C Brasil
- Brazilian office of W3C (World Wide Web Consortium)



IX.br numbers:

> **26** independent Internet Exchanges
> **1300+** ASs participants, and
2 Tbps of peak traffic at all IXs aggregated
> **940+** ASs, **30** PoPs (PIXs), and
1.5 Tbps at IX.br São Paulo, SP

IX.br

- 26 Internet Exchanges
- IX.br São Paulo is the biggest:
  - Around 1000 Autonomous Systems
  - Most of them are in the multilateral peering agreement
  - 4 route servers
    - Participants are required to have BGP sessions with all 4, for redundancy

Route server problems

- Quagga can’t deal with more than 1000 BGP sessions (due to the way sockets are implemented with select)
  - We had to separate IPv4 and IPv6 in different processes
  - Performance problems:
    - Quagga showed to be sensible to BGP session oscillations
    - Quagga can’t use more than 1 core (it’s one single process)…

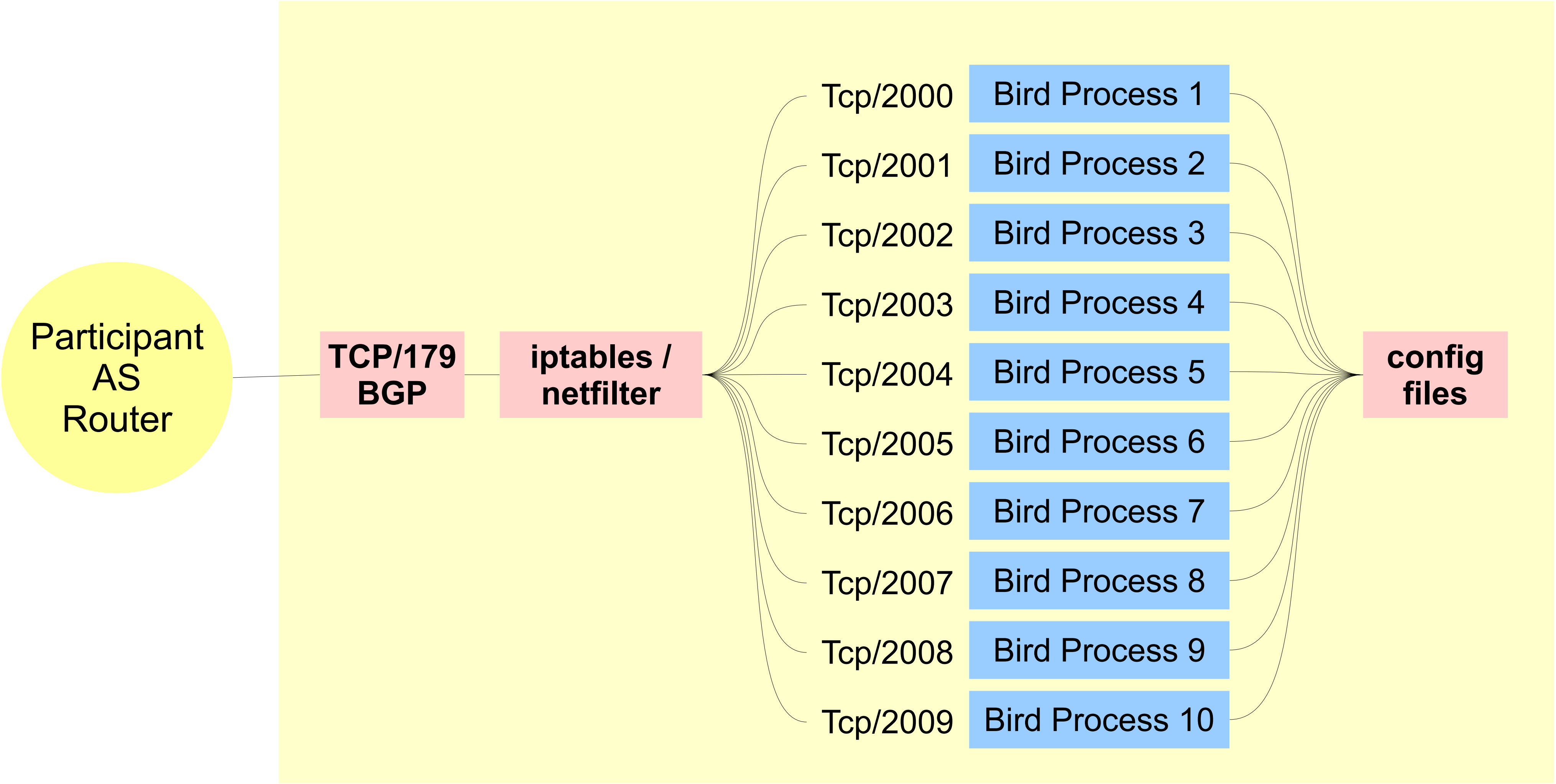
Bird?

- Bird stable version 1.4.5 over Linux was not able to scale above 1,000 peers due to SELECT function on code for sockets allocation
- Laboratory tests with Bird version 1.5 over Linux showed to solve this issue, but the code seems to be not mature enough for production
- It’s still one single process

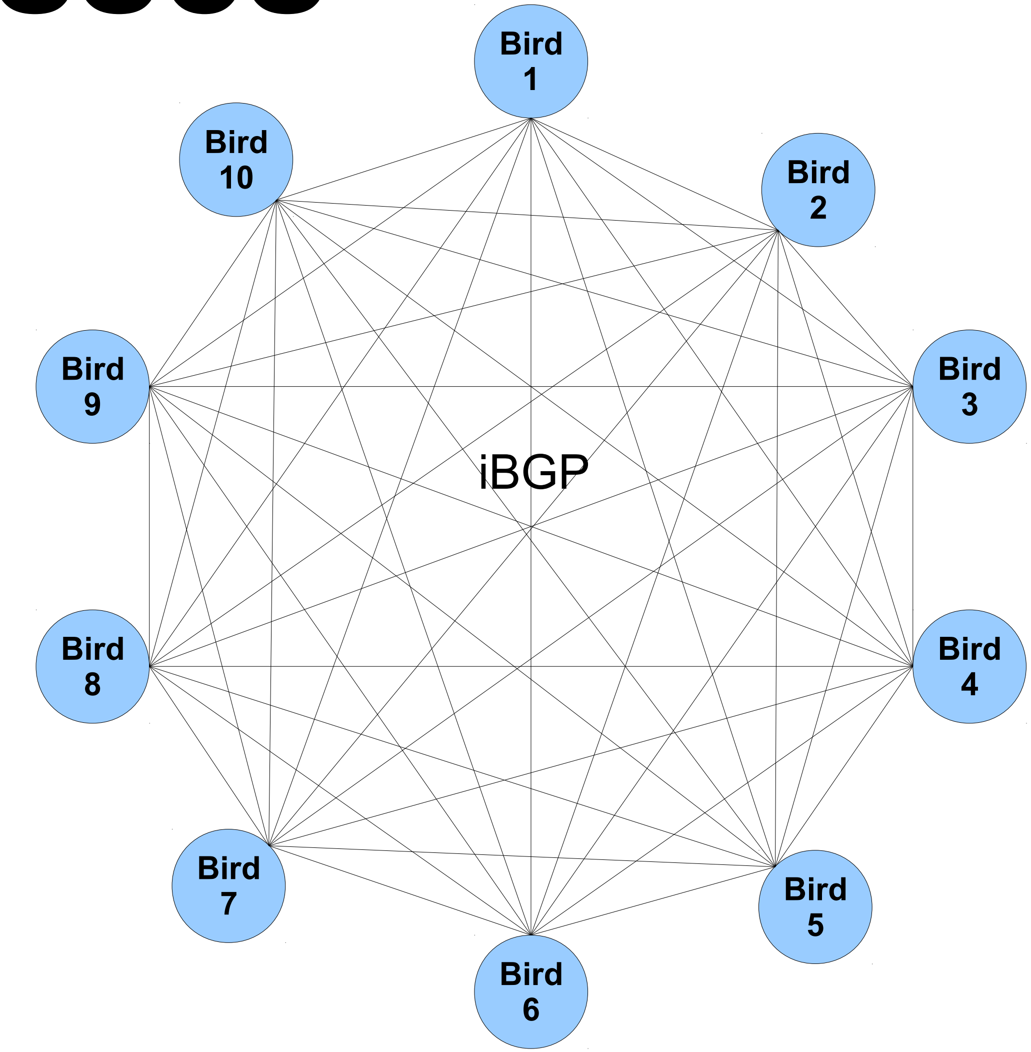
Adopted solution

- Multiple BIRD processes, instead of a single one, sharing the load
  - Each process in a different port (and IP)
  - Each process with a different BGP Router ID (and not the same as the public IP)
  - Each process share the same configuration files (for the client sessions)
  - Full mesh between the BIRD processes
  - Passive mode
  - Linux netfilter does the ‘magic’ of load sharing

“Multi BIRD”


Participant AS Router
TCP/179 BGP
iptables / netfilter
Tcp/2000
Tcp/2001
Tcp/2002
Tcp/2003
Tcp/2004
Tcp/2005
Tcp/2006
Tcp/2007
Tcp/2008
Tcp/2009
Bird Process 1
Bird Process 2
Bird Process 3
Bird Process 4
Bird Process 5
Bird Process 6
Bird Process 7
Bird Process 8
Bird Process 9
Bird Process 10
config files

Full mesh between processes



Config excerpts

```
log syslog all;
router id 187.16.217.255;
listen bgp port 2002;
define myas = 26162;
define MyLoIP = 127.0.0.12;
protocol device { }
protocol kernel { import none; }

include "/etc/bird/templates/peers*.conf";
include "/etc/bird/templates/rspeers*.conf";
include "/etc/bird/functions/*.conf";

#iBGP (loopback interface)
protocol bgp ibgp_p2000 from RSPEERS { neighbor 127.0.0.10 port 2000 as myas; source address MyLoIP; }
protocol bgp ibgp_p2001 from RSPEERS { neighbor 127.0.0.11 port 2001 as myas; source address MyLoIP; }
#this peer
#protocol bgp ibgp_p2002 from RSPEERS { neighbor 127.0.0.12 port 2002 as myas; source address MyLoIP; }
protocol bgp ibgp_p2003 from RSPEERS { neighbor 127.0.0.13 port 2003 as myas; source address MyLoIP; }
protocol bgp ibgp_p2004 from RSPEERS { neighbor 127.0.0.14 port 2004 as myas; source address MyLoIP; }
protocol bgp ibgp_p2005 from RSPEERS { neighbor 127.0.0.15 port 2005 as myas; source address MyLoIP; }
protocol bgp ibgp_p2006 from RSPEERS { neighbor 127.0.0.16 port 2006 as myas; source address MyLoIP; }
protocol bgp ibgp_p2007 from RSPEERS { neighbor 127.0.0.17 port 2007 as myas; source address MyLoIP; }
protocol bgp ibgp_p2008 from RSPEERS { neighbor 127.0.0.18 port 2008 as myas; source address MyLoIP; }
protocol bgp ibgp_p2009 from RSPEERS { neighbor 127.0.0.19 port 2009 as myas; source address MyLoIP; }

#peers (clients)
include "/etc/bird/peers/*.conf";
```

Config excerpts

```
# as22548.conf - last change: 2016-11-01 02:15:02

# asn,description,mark,filters
# 22548,V4_AS22548,22548,28571 61580

# ipv4,asn,description,maximum_prefix,password,passive,shutdown
# 187.16.217.2,22548,V4_AS22548,100,,True,False

filter bgp_in_as22548
{
    if (DenyATMv4BlockPrefix()) then reject;
    bgp_in(22548);
    bgp_community.add((26162,22548));
    accept;
}

filter bgp_out_as22548
{
    # filter as28571 - USP - mark 28571
    if (26162,28571) ~ bgp_community then reject;
    # filter as61580 - OpenCDN.nic.br - mark 61580
    if (26162,61580) ~ bgp_community then reject;
    accept;
}

protocol bgp as22548_187_16_217_2 from PEERS {
    description "as22548 ATM IPv4 - V4_AS22548";
    neighbor 187.16.217.2 as 22548;
    passive on;
    import limit 100 action restart;
    import filter bgp_in_as22548;
    export filter bgp_out_as22548;
}
```

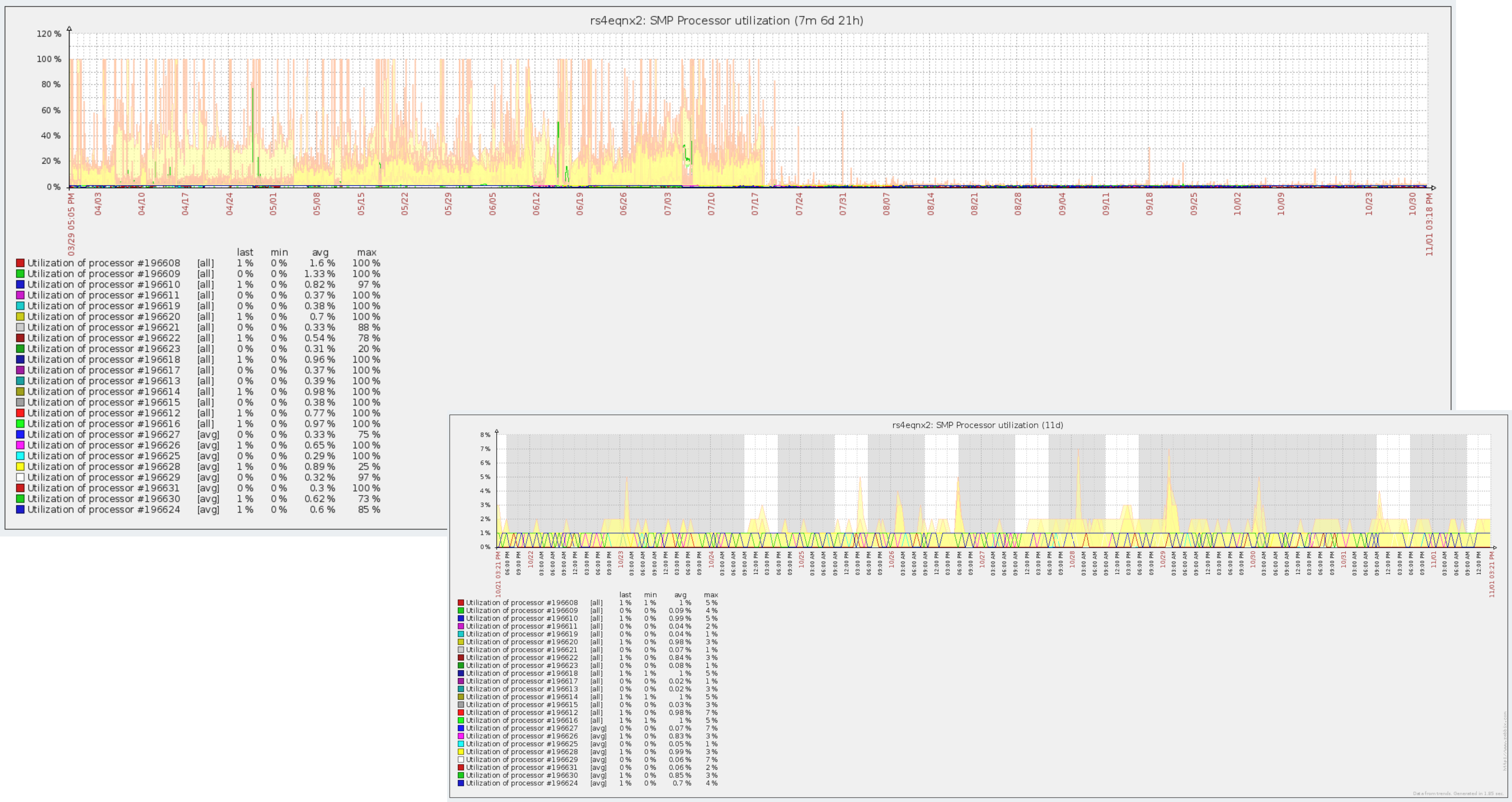
Config excerpts

```
# port redirecting - load sharing
-A PREROUTING -p tcp -i em2.2012 --dport 179 -m state --state NEW -m
statistic --mode nth --every 10 --packet 0 -j DNAT -to-destination
187.16.216.254:2000
-A PREROUTING -p tcp -i em2.2012 --dport 179 -m state --state NEW -m
statistic --mode nth --every 9 --packet 0 -j DNAT -to-destination
187.16.216.254:2001
-A PREROUTING -p tcp -i em2.2012 --dport 179 -m state --state NEW -m
statistic --mode nth --every 8 --packet 0 -j DNAT -to-destination
187.16.216.254:2002
-A PREROUTING -p tcp -i em2.2012 --dport 179 -m state --state NEW -m
statistic --mode nth --every 7 --packet 0 -j DNAT -to-destination
187.16.216.254:2003
-A PREROUTING -p tcp -i em2.2012 --dport 179 -m state --state NEW -m
statistic --mode nth --every 6 --packet 0 -j DNAT -to-destination
187.16.216.254:2004
-A PREROUTING -p tcp -i em2.2012 --dport 179 -m state --state NEW -m
statistic --mode nth --every 5 --packet 0 -j DNAT -to-destination
187.16.216.254:2005
-A PREROUTING -p tcp -i em2.2012 --dport 179 -m state --state NEW -m
statistic --mode nth --every 4 --packet 0 -j DNAT -to-destination
187.16.216.254:2006
-A PREROUTING -p tcp -i em2.2012 --dport 179 -m state --state NEW -m
statistic --mode nth --every 3 --packet 0 -j DNAT -to-destination
187.16.216.254:2007
-A PREROUTING -p tcp -i em2.2012 --dport 179 -m state --state NEW -m
statistic --mode nth --every 2 --packet 0 -j DNAT -to-destination
187.16.216.254:2008
-A PREROUTING -p tcp -i em2.2012 --dport 179 -m state --state NEW -m
statistic --mode nth --every 1 --packet 0 -j DNAT -to-destination
187.16.216.254:2009
```

Results

- It worked very well!
- Smaller memory footprint than quagga (~ 4Gbytes)
- Better distribution of the load between the multiple cores/processors
- Smaller load, better performance

Results



rs4eqnx2: SMP Processor utilization (7m 6d 21h)

			last	min	avg	max
🟥	Utilization of processor #196608	[all]	1 %	0 %	1.6 %	100 %
🟩	Utilization of processor #196609	[all]	0 %	0 %	1.33 %	100 %
🟦	Utilization of processor #196610	[all]	1 %	0 %	0.82 %	97 %
🟪	Utilization of processor #196611	[all]	0 %	0 %	0.37 %	100 %
🟦	Utilization of processor #196619	[all]	0 %	0 %	0.38 %	100 %
🟨	Utilization of processor #196620	[all]	1 %	0 %	0.7 %	100 %
⬜	Utilization of processor #196621	[all]	0 %	0 %	0.33 %	88 %
🟥	Utilization of processor #196622	[all]	1 %	0 %	0.54 %	78 %
🟩	Utilization of processor #196623	[all]	0 %	0 %	0.31 %	20 %
🟦	Utilization of processor #196618	[all]	1 %	0 %	0.96 %	100 %
🟪	Utilization of processor #196617	[all]	0 %	0 %	0.37 %	100 %
🟦	Utilization of processor #196613	[all]	0 %	0 %	0.39 %	100 %
🟨	Utilization of processor #196614	[all]	1 %	0 %	0.98 %	100 %
⬜	Utilization of processor #196615	[all]	0 %	0 %	0.38 %	100 %
🟥	Utilization of processor #196612	[all]	1 %	0 %	0.77 %	100 %
🟩	Utilization of processor #196616	[all]	1 %	0 %	0.97 %	100 %
🟦	Utilization of processor #196627	[avg]	0 %	0 %	0.33 %	75 %
🟪	Utilization of processor #196626	[avg]	1 %	0 %	0.65 %	100 %
🟦	Utilization of processor #196625	[avg]	0 %	0 %	0.29 %	100 %
🟨	Utilization of processor #196628	[avg]	1 %	0 %	0.89 %	25 %
⬜	Utilization of processor #196629	[avg]	0 %	0 %	0.32 %	97 %
🟥	Utilization of processor #196631	[avg]	0 %	0 %	0.3 %	100 %
🟩	Utilization of processor #196630	[avg]	1 %	0 %	0.62 %	73 %
🟦	Utilization of processor #196624	[avg]	1 %	0 %	0.6 %	85 %

rs4eqnx2: SMP Processor utilization (11d)

			last	min	avg	max
🟥	Utilization of processor #196608	[all]	1 %	1 %	1 %	5 %
🟩	Utilization of processor #196609	[all]	0 %	0 %	0.09 %	4 %
🟦	Utilization of processor #196610	[all]	1 %	0 %	0.99 %	5 %
🟪	Utilization of processor #196611	[all]	0 %	0 %	0.04 %	2 %
🟦	Utilization of processor #196619	[all]	0 %	0 %	0.04 %	1 %
🟨	Utilization of processor #196620	[all]	1 %	0 %	0.98 %	3 %
⬜	Utilization of processor #196621	[all]	0 %	0 %	0.07 %	1 %
🟥	Utilization of processor #196622	[all]	1 %	0 %	0.84 %	3 %
🟩	Utilization of processor #196623	[all]	0 %	0 %	0.08 %	1 %
🟦	Utilization of processor #196618	[all]	1 %	1 %	1 %	5 %
🟪	Utilization of processor #196617	[all]	0 %	0 %	0.02 %	1 %
🟦	Utilization of processor #196613	[all]	0 %	0 %	0.02 %	3 %
🟨	Utilization of processor #196614	[all]	1 %	1 %	1 %	5 %
⬜	Utilization of processor #196615	[all]	0 %	0 %	0.03 %	3 %
🟥	Utilization of processor #196612	[all]	1 %	0 %	0.98 %	7 %
🟩	Utilization of processor #196616	[all]	1 %	1 %	1 %	5 %
🟦	Utilization of processor #196627	[avg]	0 %	0 %	0.07 %	7 %
🟪	Utilization of processor #196626	[avg]	1 %	0 %	0.83 %	3 %
🟦	Utilization of processor #196625	[avg]	0 %	0 %	0.05 %	1 %
🟨	Utilization of processor #196628	[avg]	1 %	0 %	0.99 %	3 %
⬜	Utilization of processor #196629	[avg]	0 %	0 %	0.06 %	7 %
🟥	Utilization of processor #196631	[avg]	0 %	0 %	0.06 %	2 %
🟩	Utilization of processor #196630	[avg]	1 %	0 %	0.85 %	3 %
🟦	Utilization of processor #196624	[avg]	1 %	0 %	0.7 %	4 %

Issues and workarounds

- Troubleshooting: in which process is each client?
  - We wrote some scripts to manage the multiple birds as a single router
- MD5 works only with active mode
  - We choosed one single bird process to configure all clients with MD5 in active mode
- Some (very few) clients have problems with passive mode in RSs
  - We configured them in the same process that we used for MD5 issue

Next steps with our RSs

- Substitute the IPv6 RS, that is still quagga, for the same solution with multiple bird processes
- Implement communities for filters between clients
- Implement mitigation of path hiding
- 2 route servers instead of 4, with external load balancers distributing the load between redundant servers
- Substitute Cisco for another solution
  - Multiple quagga?
  - GoBGP?

Obrigado! Dzięki!
Thanks!

www.ix.br

moreiras@nic.br

@moreiras