

Hiding infrastructure problems from users: load balancers at the RAL Tier-1

Andrew Lahiff
HEP SYSMAN June 2016

What we used to do (& still mostly do)

- DNS aliases

```
-bash-4.1$ nslookup srm-atlas.gridpp.rl.ac.uk  
Server:      130.246.8.13  
Address:     130.246.8.13#53
```

```
Name:        srm-atlas.gridpp.rl.ac.uk  
Address:     130.246.181.163  
Name:        srm-atlas.gridpp.rl.ac.uk  
Address:     130.246.181.164  
Name:        srm-atlas.gridpp.rl.ac.uk  
Address:     130.246.181.171  
Name:        srm-atlas.gridpp.rl.ac.uk  
Address:     130.246.181.162
```

- What's wrong with this?

Problems with DNS aliases

- What if one ATLAS SRM dies overnight?
 - The dead machine is still visible to users via the DNS alias
 - 1/N requests will fail
 - We would need to contact RAL networking to change it
 - We get a pager alarm
 - Someone will look into it, even at 2am Sunday morning
- Upgrades & reboots
 - As machines being updated are still in the DNS alias, maintainance is visible to users
 - 1/N requests will fail
 - what happens if a “quick reboot” takes longer than expected? (fsck, problem with VM, ...)

What if we had a more dynamic DNS?

- CERN seem to do this (I think)
- This is perhaps better, but still has problems
 - What about DNS libraries not respecting DNS TTLs and caching the results of name lookups?
 - Many applications do DNS lookups once and cache the results
 - Issues with IPv6 DNS round-robin
 - round-robin doesn't work – many clients will always pick the “first” host

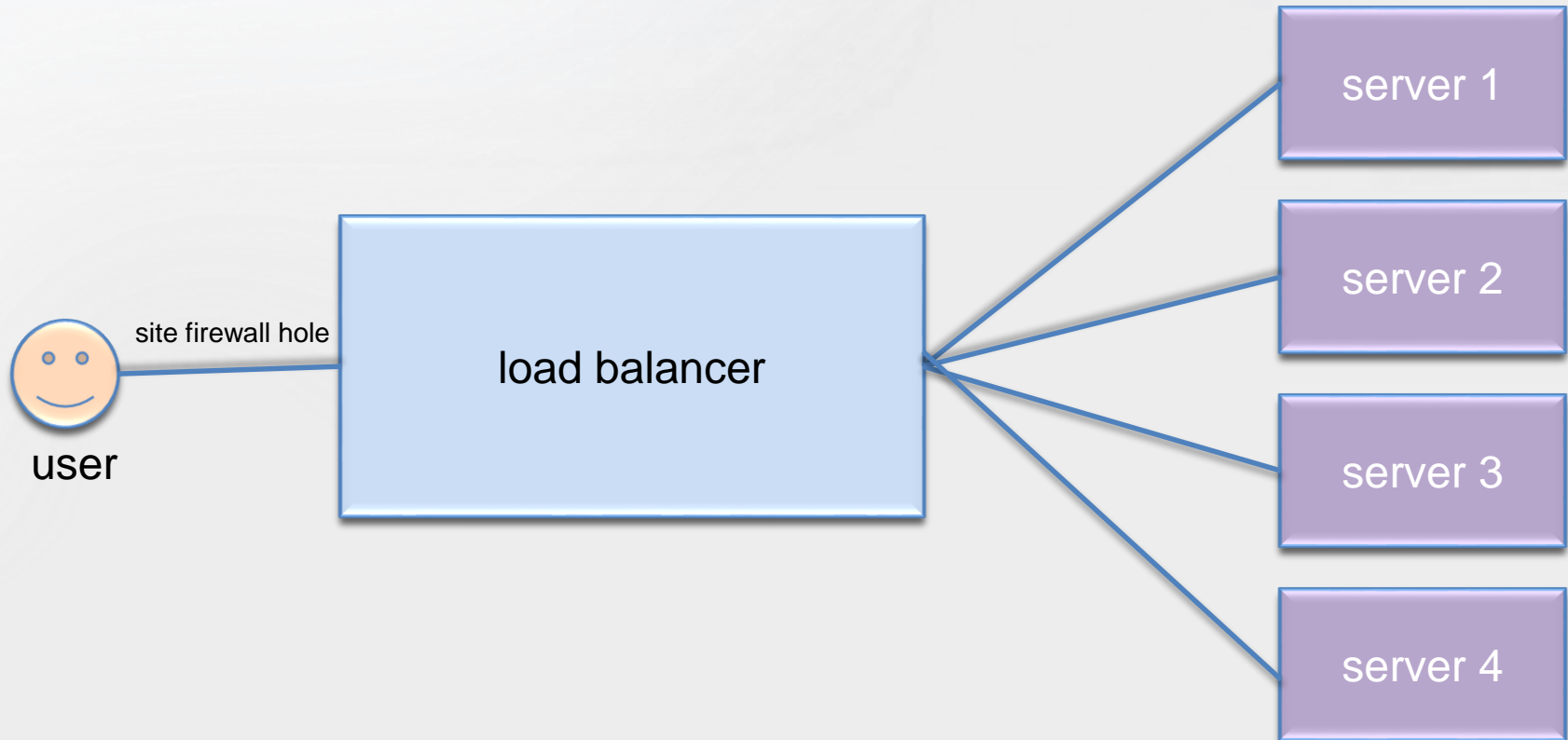
The alternative

- Instead of users connecting directly to servers...



The alternative

- Put a load balancer in between

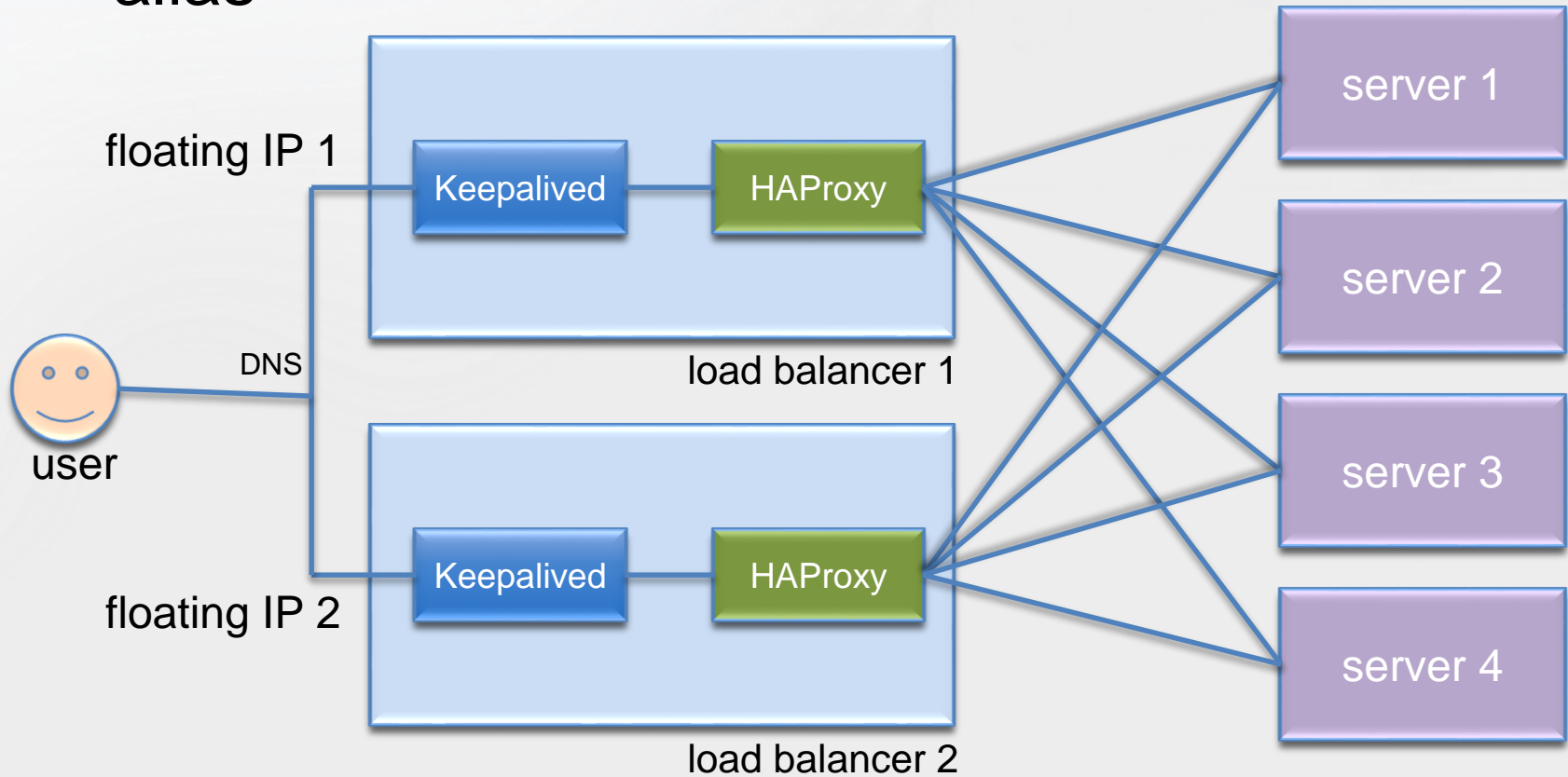


Building blocks

- HAProxy
 - Open source load balancer for TCP & HTTP
- Keepalived
 - Linux Virtual Server (LVS) router
 - Can provide HA floating IP addresses using Virtual Redundancy Routing Protocol (VRRP)

Architecture at RAL

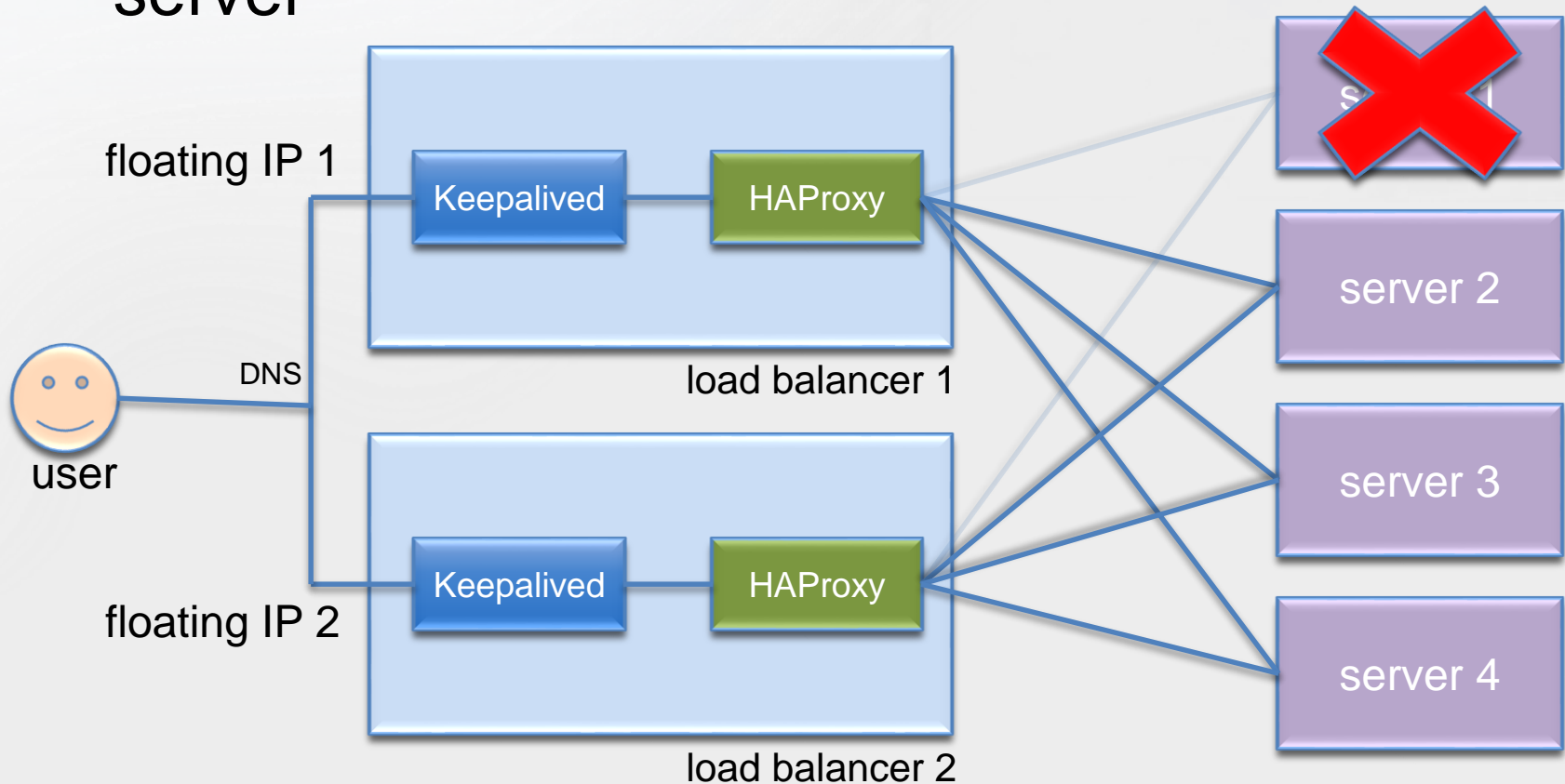
- Each service uses 2 floating IPs with a DNS alias



Each load balancer running on a VM

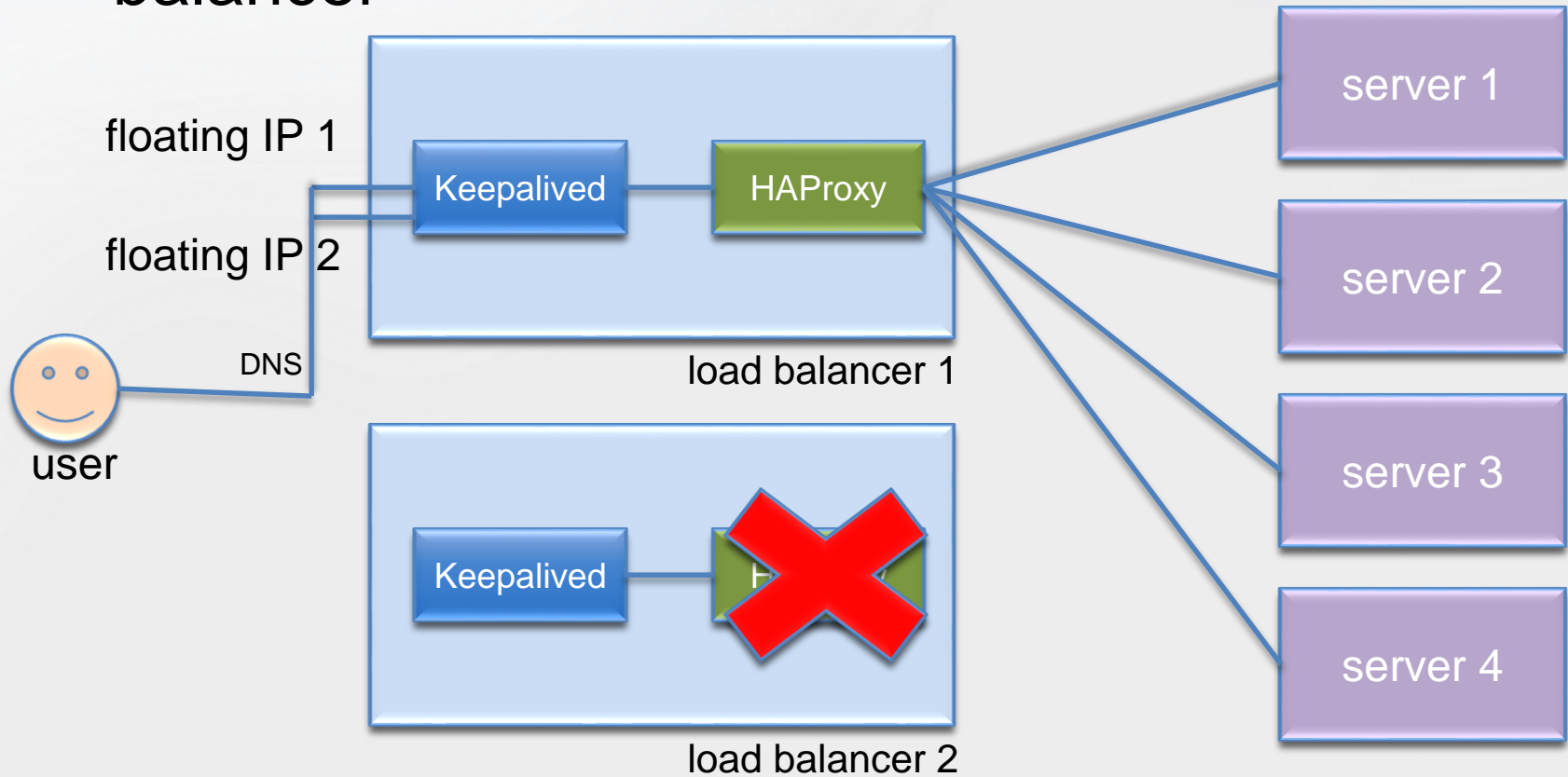
If a backend server dies

- HAProxy stops sending requests to the broken server



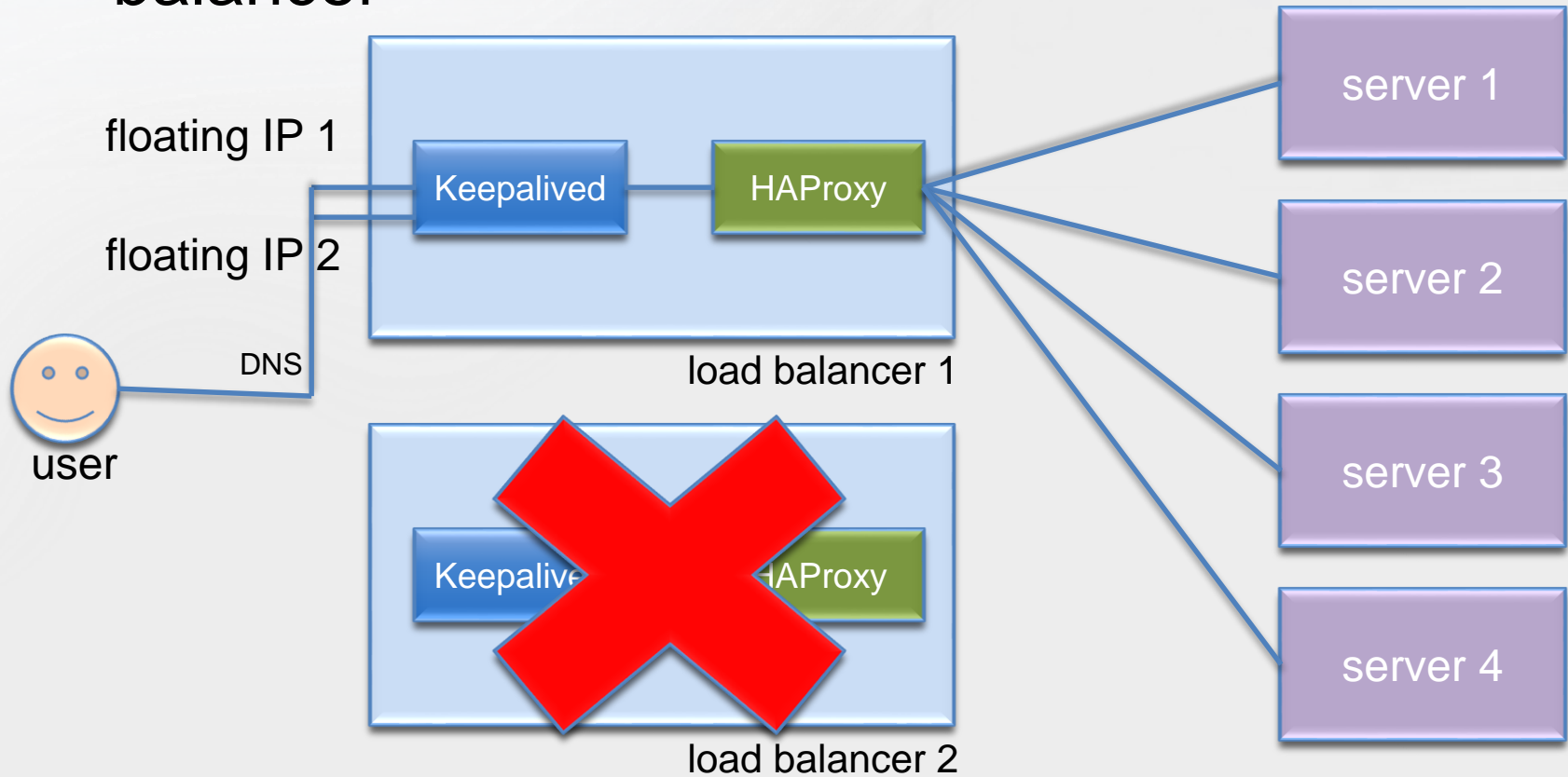
If a HAProxy dies

- The floating IP(s) move to the other load balancer



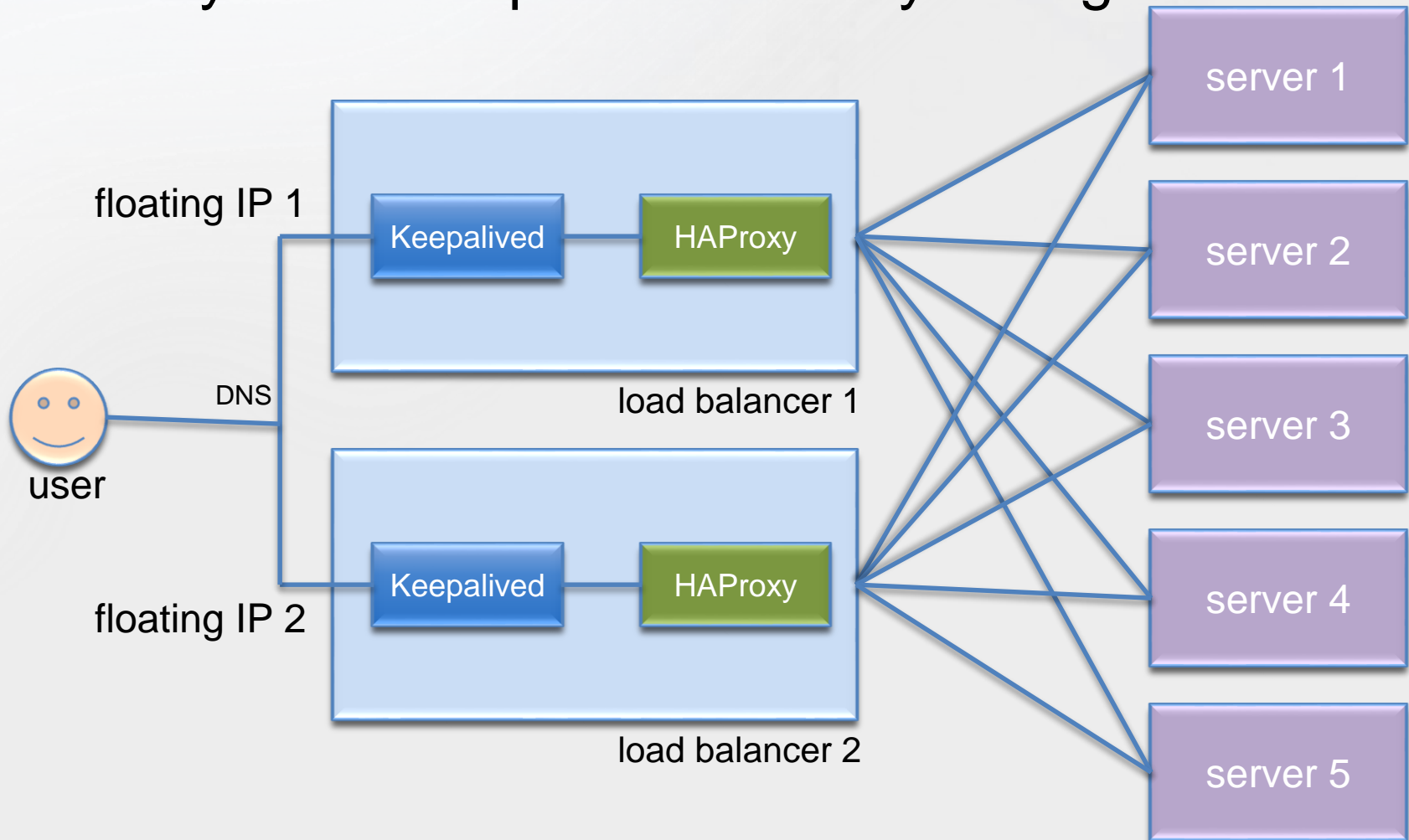
If a load balancer host dies

- The floating IP(s) move to the other load balancer



Add a new backend server

- Only need to update HAProxy config



How to check if backend servers are healthy?

- HAProxy has configurable built-in checks
 - tcp connection attempt
 - tcp, expect response to contain a specified string
 - SSLv3 client hello
 - http response code
 - http, expect response to contain a specified string
 - MySQL, PostgreSQL
 - SMTP
- For FTS3
 - using tcp (SOAP API), SSLv3 (RESTful API, monitoring app)

Load balancing options

- Round-robin
 - each backend server used in turn
 - using for FTS3 (soap), FTS3 (REST)
- Source
 - each client IP always goes to the same backend server
 - used for FTS3 (monitoring app)
 - no more endless “identify yourself with a certificate” requests from your browser!
 - compare <https://fts3.cern.ch:8449/fts3/ftsmon/#/> to <https://lcgfts3.gridpp.rl.ac.uk:8449/fts3/ftsmon/#/>
- Also some others
 - leastconn, first, ...

Monitoring

- HAProxy stats page

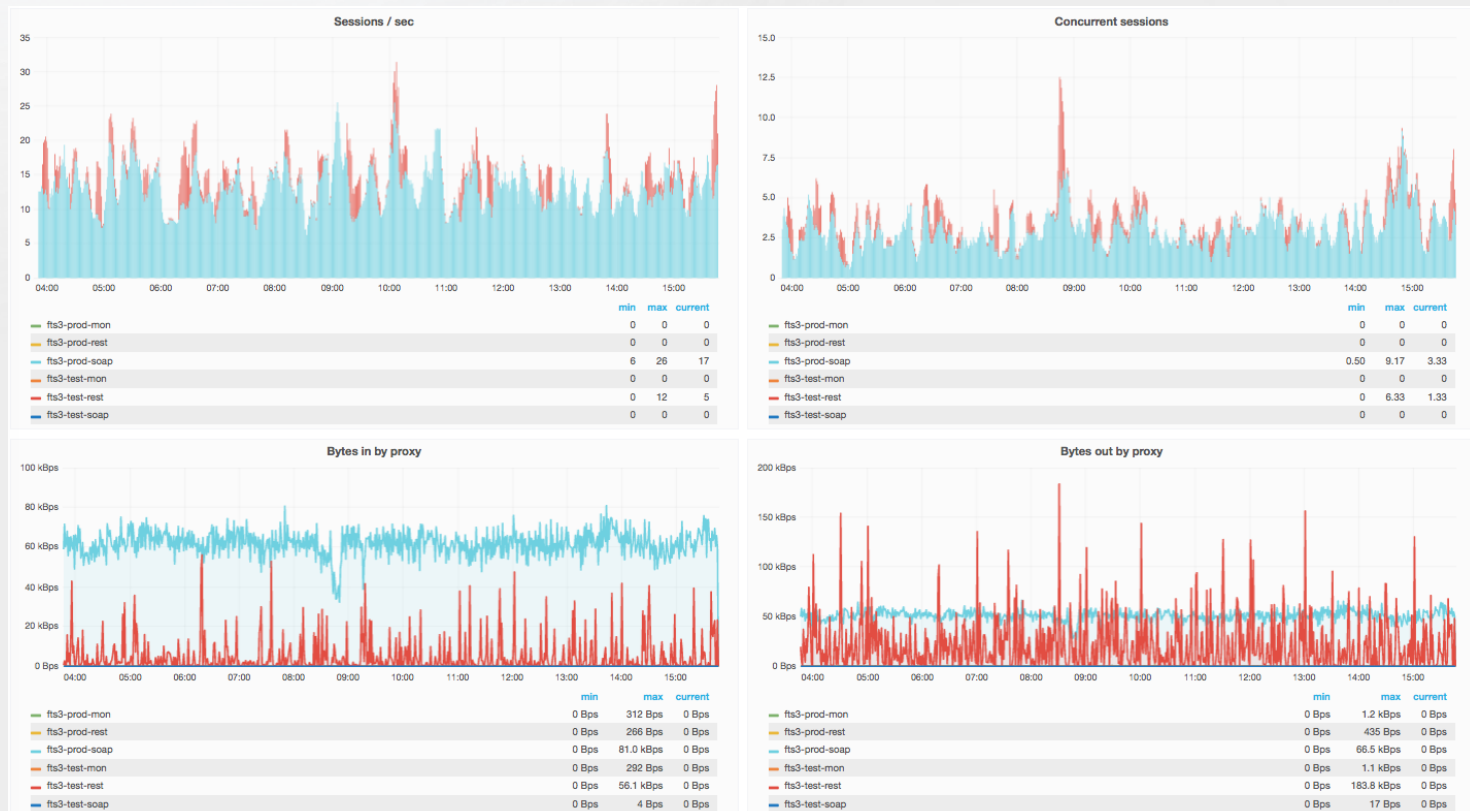
fts3-test-soep																															
	Queue			Session rate			Sessions					Bytes		Denied		Errors			Warnings		Server										
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle	
Frontend				0	54	-	0	24	4 000		214 704			13 699 831 212	14 618 836 194	0	0	1				OPEN									
fts-test01.grndpp.rl.ac.uk	0	0	128	0	7	-	0	2	-	214 704	214 704	3h14m	1 711 943 017	1 823 987 172	0	0	0	0	4	0	0	5d2h UP	L4OK in 0ms	100	Y	-	4	2	58s	-	
fts-test02.grndpp.rl.ac.uk	0	0	128	0	7	-	0	3	-	214 704	214 704	2d22h	1 713 042 996	1 831 465 299	0	0	0	0	3	0	0	5d2h UP	L4OK in 0ms	100	Y	-	4	2	2m8s	-	
fts-test03.grndpp.rl.ac.uk	0	0	128	0	7	-	0	8	-	214 703	214 703	4d14h	1 712 470 004	1 831 642 055	0	0	0	0	3	0	0	5d2h UP	L4OK in 0ms	100	Y	-	8	4	22h12m	-	
fts-test04.grndpp.rl.ac.uk	0	0	128	0	7	-	0	4	-	214 703	214 703	4d14h	1 712 888 457	1 828 976 339	0	0	0	0	2	0	0	5d2h UP	L4OK in 0ms	100	Y	-	4	2	2m23s	-	
fts-test05.grndpp.rl.ac.uk	0	0	128	0	7	-	0	4	-	214 703	214 703	4d14h	1 712 640 951	1 826 992 337	0	0	0	0	0	0	0	5d1h UP	L4OK in 0ms	100	Y	-	4	2	2m23s	-	
fts-test06.grndpp.rl.ac.uk	0	0	128	0	7	-	0	5	-	214 702	214 702	2d22h	1 712 693 760	1 828 489 793	0	0	0	0	4	0	0	5d1h UP	L4OK in 0ms	100	Y	-	22	2	2m18s	-	
fts-test07.grndpp.rl.ac.uk	0	0	128	0	7	-	0	4	-	214 702	214 702	2d22h	1 712 146 934	1 823 951 706	0	0	0	0	5	0	0	5d1h UP	L4OK in 0ms	100	Y	-	18	2	2m53s	-	
fts-test08.grndpp.rl.ac.uk	0	0	128	0	7	-	0	4	-	214 702	214 702	2d22h	1 712 005 094	1 823 331 493	0	0	0	0	5	0	0	5d1h UP	L4OK in 0ms	100	Y	-	25	2	2m13s	-	
Backend	0	0		0	54	-	0	24	400		1 717 625	1 717 623	3h14m	13 699 831 212	14 618 836 194	0	0	0	26	0	0	36d2h UP		800	8	0			0	0s	

fts3-test-rest																															
	Queue			Session rate			Sessions					Bytes		Denied		Errors			Warnings		Server										
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle	
Frontend				0	120	-	0	53	4 000		4 509 284			44 225 547 267	230 127 296 165	0	0	0				OPEN									
fts-test01.grndpp.rl.ac.uk	0	0	128	0	15	-	0	11	-	557 888	557 888	54s	5 546 921 646	28 746 379 266	0	0	0	0	0	0	0	6d12h UP	L6OK in 0ms	100	Y	-	7	2	19s	-	
fts-test02.grndpp.rl.ac.uk	0	0	128	0	15	-	0	9	-	557 888	557 888	54s	5 548 399 400	28 939 883 442	0	0	0	0	0	0	0	5d2h UP	L6OK in 0ms	100	Y	-	4	1	14s	-	
fts-test03.grndpp.rl.ac.uk	0	0	128	0	15	-	0	30	-	544 540	544 540	54s	5 418 237 459	26 045 371 245	0	0	0	0	0	0	0	5d2h UP	L6OK in 3ms	100	Y	-	16	7	22h11m	-	
fts-test04.grndpp.rl.ac.uk	0	0	128	0	15	-	0	15	-	557 891	557 891	39s	5 545 268 086	28 659 267 552	0	0	0	0	0	0	0	5d2h UP	L6OK in 0ms	100	Y	-	4	1	14s	-	
fts-test05.grndpp.rl.ac.uk	0	0	128	0	15	-	0	12	-	557 896	557 896	54s	5 544 094 154	28 764 558 330	0	0	0	0	0	0	0	5d1h UP	L6OK in 1ms	100	Y	-	5	1	49s	-	
fts-test06.grndpp.rl.ac.uk	0	0	128	0	15	-	0	9	-	557 878	557 878	54s	5 541 564 081	28 521 113 012	0	0	0	0	0	0	2	5d2h UP	L6OK in 1ms	100	Y	-	56	4	56s	-	
fts-test07.grndpp.rl.ac.uk	0	0	128	0	15	-	0	8	-	557 858	557 858	54s	5 547 500 927	29 062 775 424	0	0	0	0	1	0	1	5d1h UP	L6OK in 0ms	100	Y	-	54	4	1m22s	-	
fts-test08.grndpp.rl.ac.uk	0	0	128	0	15	-	0	11	-	557 827	557 827	54s	5 533 561 514	29 387 947 894	0	0	0	0	1	0	0	5d1h UP	L6OK in 1ms	100	Y	-	51	5	1m46s	-	
Backend	0	0		0	120	-	0	53	400		4 509 284	4 449 666	39s	44 225 547 267	230 127 296 165	0	0	0	2	0	3	36d2h UP		800	8	0			0	0s	

fts3-test-mon																															
	Queue			Session rate			Sessions					Bytes		Denied		Errors			Warnings		Server										
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle	
Frontend				0	58	-	0	11	4 000		37 787			120 179 405	611 339 296	0	0	0				OPEN									
fts-test01.grndpp.rl.ac.uk	0	0	128	0	17	-	0	7	-	1 351	1 351	1d3h	4 231 058	35 872 300	0	0	0	0	0	0	0	6d12h UP	L6OK in 0ms	100	Y	-	7	2	19s	-	
fts-test02.grndpp.rl.ac.uk	0	0	128	0	40	-	0	6	-	93	93	5d1h	240 044	723 970	0	0	0	0	0	0	0	5d2h UP	L6OK in 0ms	100	Y	-	4	1	14s	-	
fts-test03.grndpp.rl.ac.uk	0	0	128	0	17	-	0	6	-	3 049	3 049	6d1h	10 408 259	31 393 844	0	0	0	0	0	0	0	5d2h UP	L6OK in 3ms	100	Y	-	16	7	22h11m	-	
fts-test04.grndpp.rl.ac.uk	0	0	128	0	58	-	0	6	-	1 036	1 036	5h7m	3 280 059	9 147 149	0	0	0	0	0	0	0	5d2h UP	L6OK in 1ms	100	Y	-	4	1	14s	-	
fts-test05.grndpp.rl.ac.uk	0	0	128	0	23	-	0	6	-	1 117	1 117	1d3h	3 635 966	21 385 607	0	0	0	0	0	0	0	5d1h UP	L6OK in 1ms	100	Y	-	5	1	45s	-	
fts-test06.grndpp.rl.ac.uk	0	0	128	0	24	-	0	11	-	15 278	15 278	21h8m	35 590 545	119 997 333	0	0	0	0	0	0	0	5d2h UP	L6OK in 1ms	100	Y	-	68	4	40s	-	
fts-test07.grndpp.rl.ac.uk	0	0	128	0	16	-	0	6	-	2 435	2 435	1d5h	2 863 482	26 495 863	0	0	0	0	0	0	0	5d1h UP	L6OK in 1ms	100	Y	-	51	4	1m10s	-	
fts-test08.grndpp.rl.ac.uk	0	0	128	0	24	-	0	6	-	13 428	13 428	3m31s	59 929 992	366 323 230	0	0	0	0	0	0	0	5d1h UP	L6OK in 1ms	100	Y	-	59	3	1m22s	-	
Backend	0	0		0	58	-	0	11	400		37 787	37 787	3m31s	120 179 405	611 339 296	0	0	0	0	0	0	36d2h UP		800	8	0			0	0s	

Monitoring

- Use Telegraf to send HAProxy metrics to InfluxDB



Monitoring

- Nagios tests per load balancer
 - check that the number of healthy backend servers for each service is above a minimum threshold

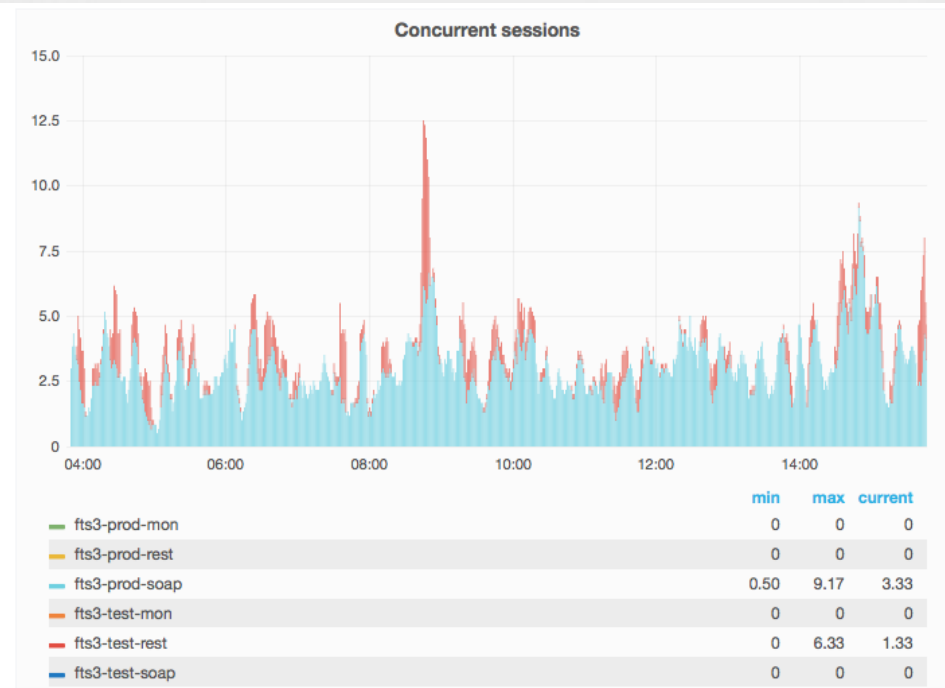
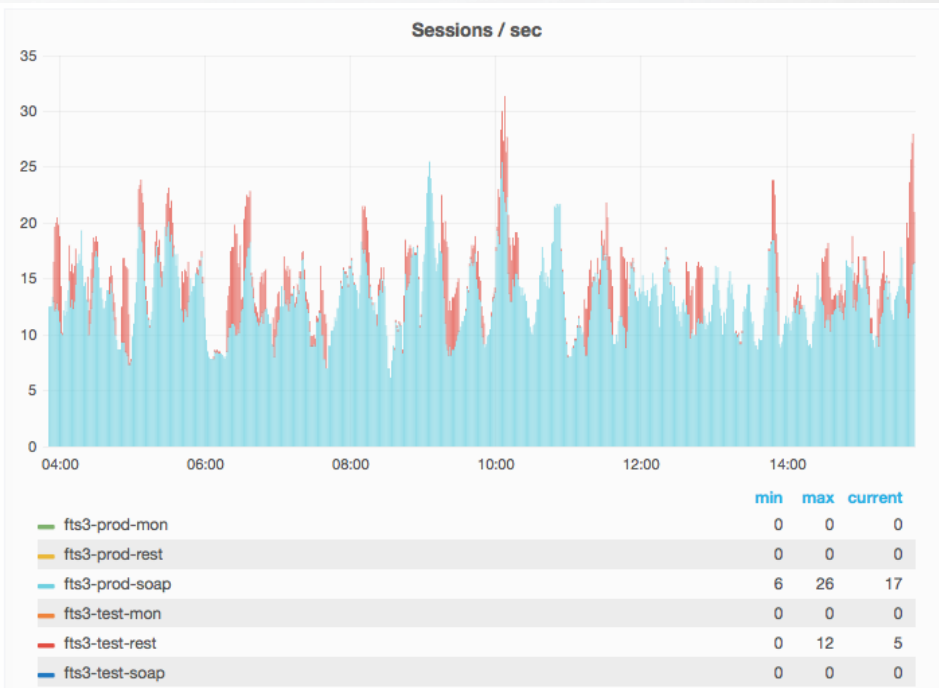
Check proc HAProxy		OK	13:15:13	92d 2h 11m 16s	1/3	PROCS OK: 1 process with command name haproxy	
Check proc monit		OK	13:15:00	92d 2h 9m 53s	1/3	PROCS OK: 1 process with command name monit	
Check procs keepalived		OK	13:15:13	92d 2h 8m 36s	1/3	PROCS OK: 3 processes with command name keepalived	
Check proxy fts3-prod-mon		OK	13:02:17	4d 19h 38m 32s	1/3	Check haproxy OK - checked proxies: fts3-prod-mon	
Check proxy fts3-prod-rest		OK	13:02:17	4d 19h 38m 32s	1/3	Check haproxy OK - checked proxies: fts3-prod-rest	
Check proxy fts3-prod-soap		OK	13:02:17	4d 19h 38m 32s	1/3	Check haproxy OK - checked proxies: fts3-prod-soap	
Check proxy fts3-test-mon		OK	13:02:17	5d 1h 42m 34s	1/3	Check haproxy OK - checked proxies: fts3-test-mon	
Check proxy fts3-test-rest		OK	13:02:17	5d 1h 42m 34s	1/3	Check haproxy OK - checked proxies: fts3-test-rest	
Check proxy fts3-test-soap		OK	13:02:17	5d 1h 42m 34s	1/3	Check haproxy OK - checked proxies: fts3-test-soap	

- Tests for floating IPs

Service ▲▼	Status ▲▼	Last Check ▲▼	Duration ▲▼	Attempt ▲▼	Status Information ▲▼
Check floating IP fts3-prod-ip1	OK	13:16:16	31d 22h 54m 33s	1/5	TCP OK - 0.000 second response time on port 8446
Check floating IP fts3-prod-ip2	OK	13:16:16	18d 23h 3m 18s	1/5	TCP OK - 0.000 second response time on port 8446
Check floating IP fts3-test-ip1	OK	13:16:16	88d 20h 33m 30s	1/5	TCP OK - 0.002 second response time on port 8446
Check floating IP fts3-test-ip2	OK	13:16:16	98d 23h 40m 20s	1/5	TCP OK - 0.000 second response time on port 8446

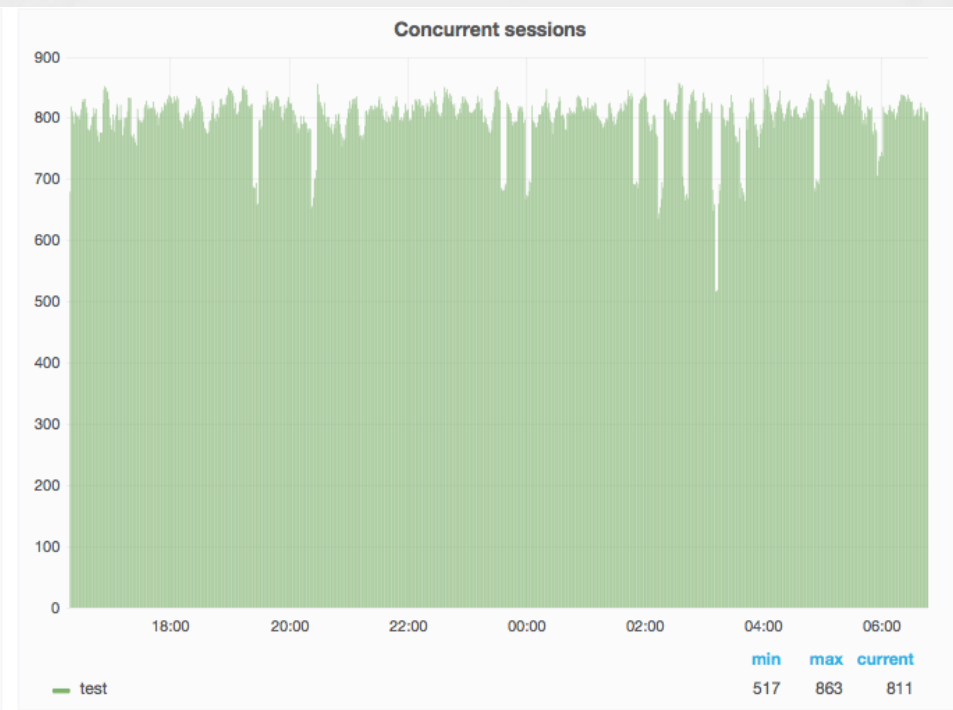
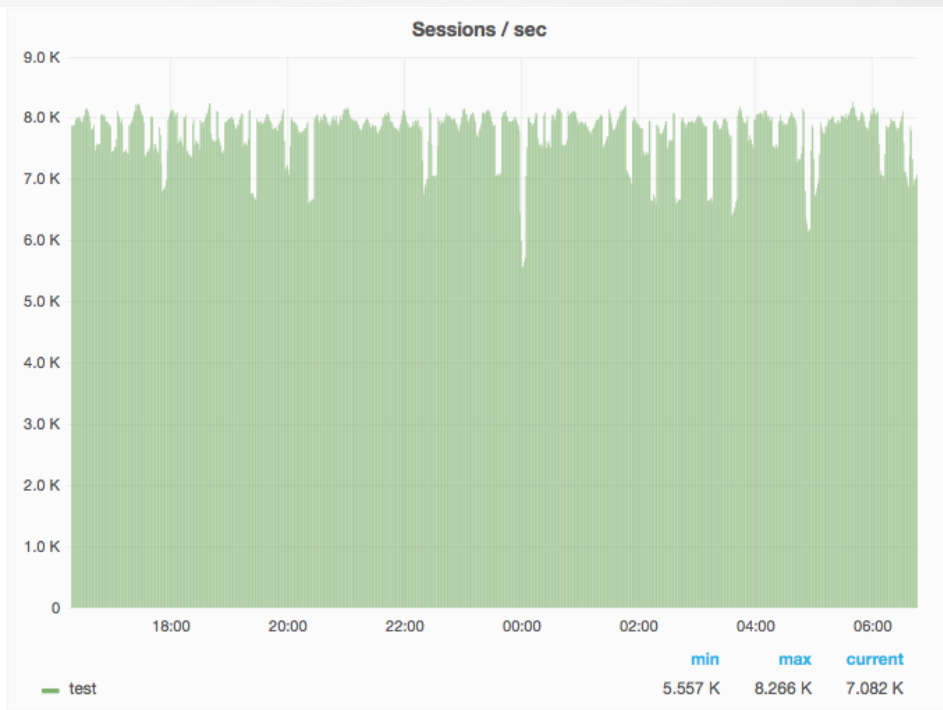
Current load

- With FTS3 only, load is very, very low!
 - Average 13 sessions/sec, 3 concurrent sessions (per lb)



Testing at higher scales

- Random example using Apache Bench, zero tuning
 - 8000 sessions/sec, 800 concurrent sessions



(using backend servers running httpd)

Current status

- Services fully using the load balancers:
 - FTS3 (“test” instance, i.e. ATLAS): since 26th April
 - FTS3 (“prod” instance, i.e. CMS etc): since 31st May
- No problems so far

Future plans

- More advanced HAProxy health checks
 - e.g. host cert expiry, CPU load too high, ...
- More services
 - Other existing services
 - Future services that require true high availability
 - SCD OpenStack (dev instance already using HAProxy, Keepalived)
 - Ceph gateways (e.g. gridFTP control traffic)
- It's also the first step required before moving to a more dynamic infrastructure
 - e.g. container orchestration