The Evolution of Samba Active Directory

Experiences in implementing and optimizing Active Directory features in Samba

Garming Sam – Catalyst IT

catalyst
open source technologists

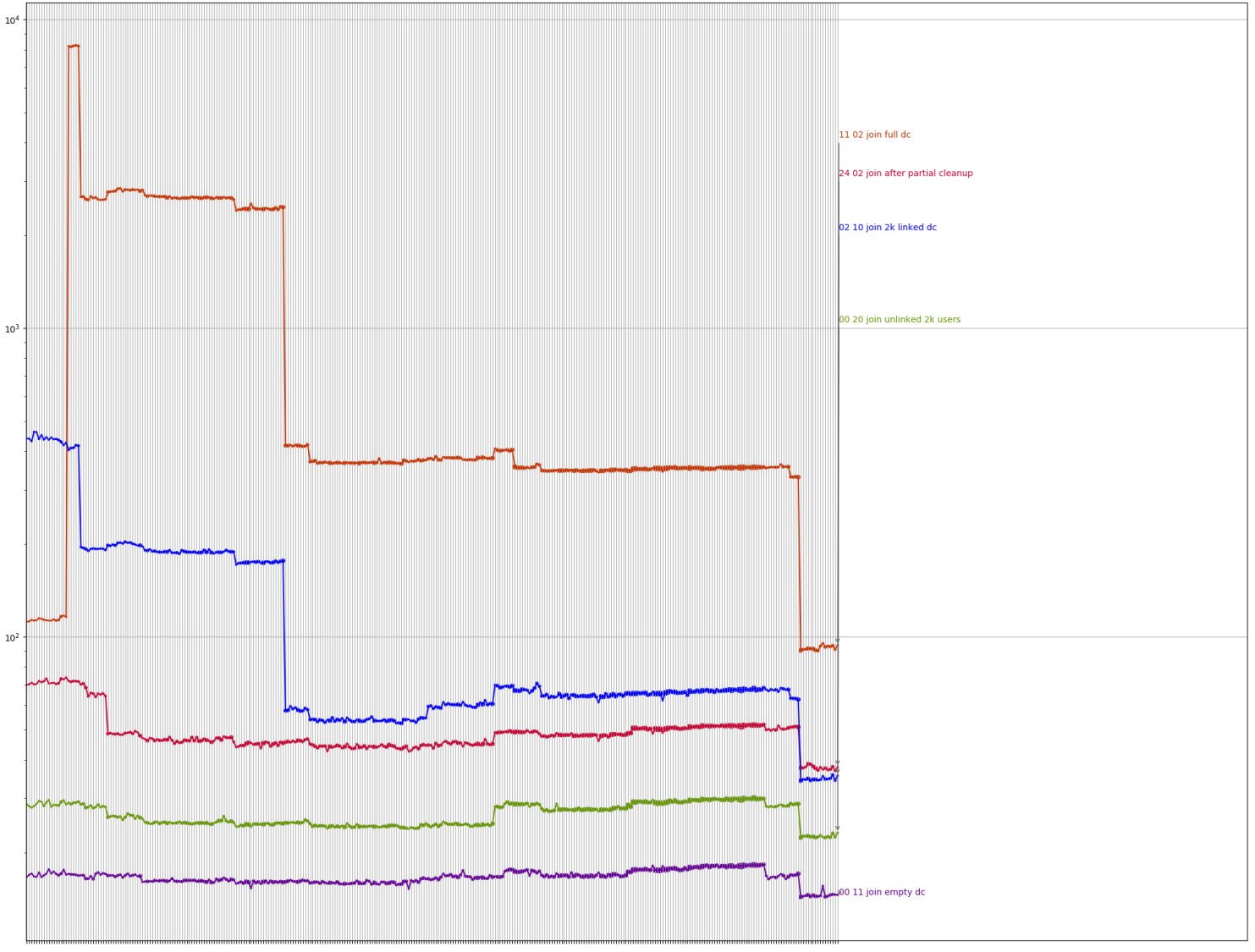
What has been done in the last year?

Samba 4.9

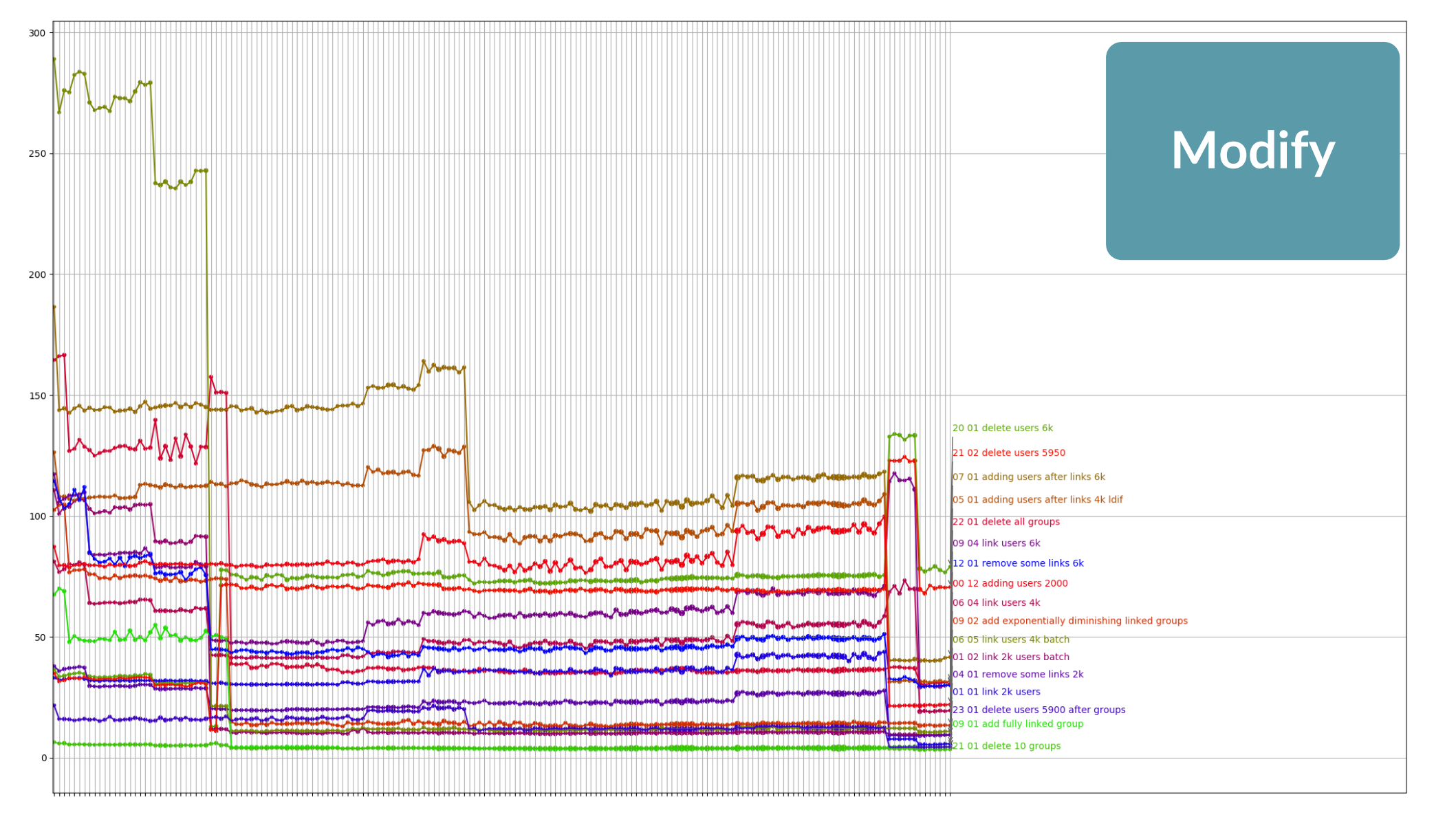
- Password and membership change auditing
- LMDB back-end*
- Fine grained password policies
- Domain backup, restore and rename tools
- Better DRS partner visualization
- Automatic DNS site coverage*
- DNS scavenging support*
- Improved trust support and more...

Samba 4.10

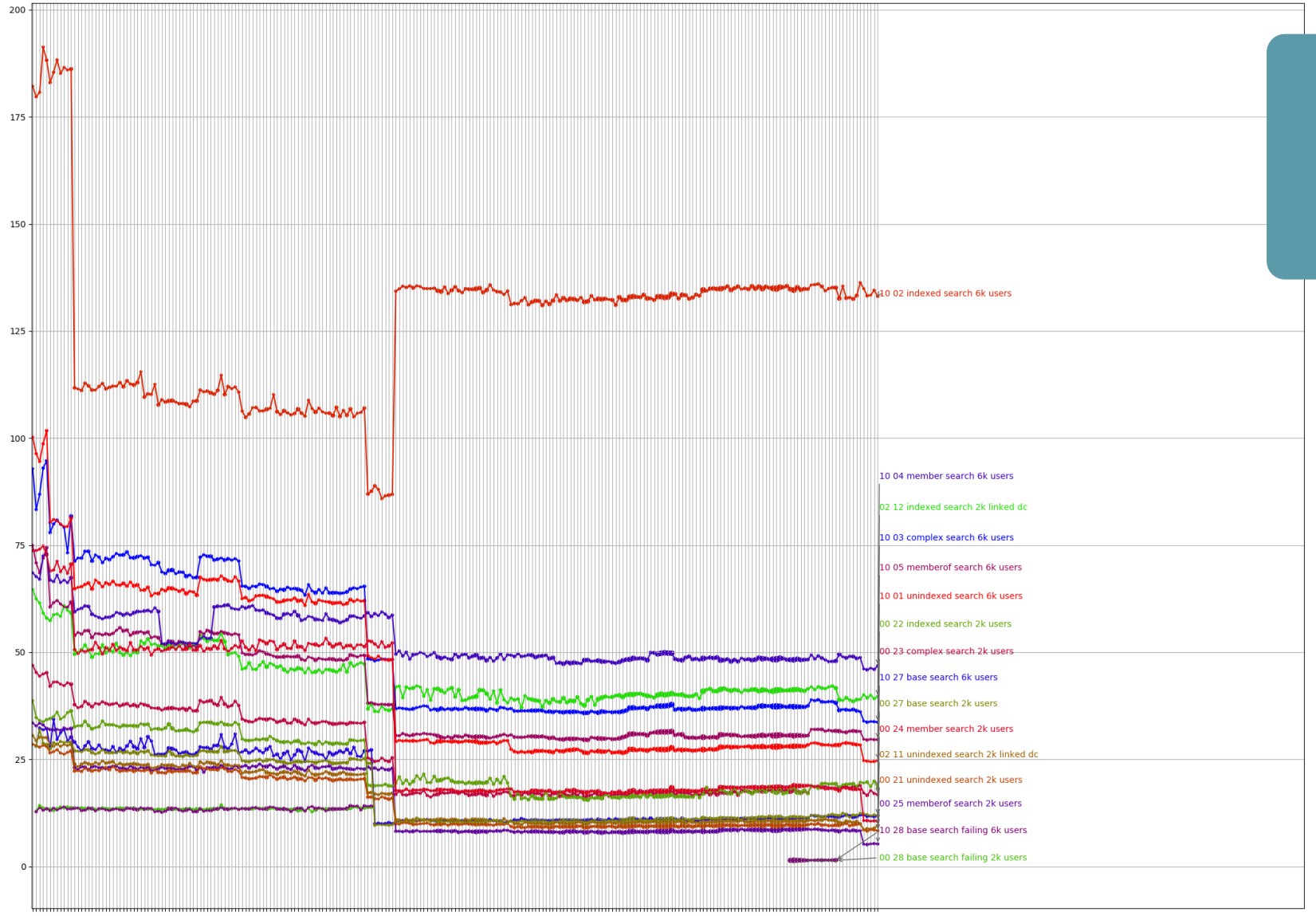
- GPO import and export*
- KDC* and NETLOGON prefork (default in 4.11)
- (Prefork) improvements for restarting services automatically
- Changes to LDAP paged results to save memory*
- Offline domain backup
- Python 3 support
- Audit logging with MS event IDs and more...


10^4
10^3
10^2
11 02 join full dc
24 02 join after partial cleanup
02 10 join 2k linked dc
00 20 join unlinked 2k users
00 11 join empty dc

Join


Modify
300
250
200
150
100
50
0
20 01 delete users 6k
21 02 delete users 5950
07 01 adding users after links 6k
05 01 adding users after links 4k ldif
22 01 delete all groups
09 04 link users 6k
12 01 remove some links 6k
00 12 adding users 2000
06 04 link users 4k
09 02 add exponentially diminishing linked groups
06 05 link users 4k batch
01 02 link 2k users batch
04 01 remove some links 2k
01 01 link 2k users
23 01 delete users 5900 after groups
09 01 add fully linked group
21 01 delete 10 groups

Search


200
175
150
125
100
75
50
25
0
10 02 indexed search 6k users
10 04 member search 6k users
02 12 indexed search 2k linked dc
10 03 complex search 6k users
10 05 memberof search 6k users
10 01 unindexed search 6k users
00 22 indexed search 2k users
00 23 complex search 2k users
10 27 base search 6k users
00 27 base search 2k users
00 24 member search 2k users
02 11 unindexed search 2k linked dc
00 21 unindexed search 2k users
00 25 memberof search 2k users
10 28 base search failing 6k users
00 28 base search failing 2k users

Performance, performance, performance

Replication improvements, linked attribute performance, rename performance, large scale improvements, ... as well as other things like schema updates

Traffic replay runner

Recording real network traffic and playing it back at different speeds

Naive traffic runner results (2 vCPU, 8GB RAM)

v4.6 – 113 operations / second

v4.7 – 94 operations / second (changes to LDAP multi-process)

v4.8 – 154 operations / second (only in new prefork process mode)

v4.9 – 157 operations / second (only in prefork mode)

v4.10 – Same as 4.8 and 4.9

Git master (prefork is default) – about 165 operations / second

Traffic sample is largely DNS, name resolution, LDAP bind, NETLOGON (typical)

Basic steps for replaying traffic

Network trace

Run wireshark and get a pcap output

Traffic summary

Anonymize the traffic and pick out important details to replay

Traffic model (optional)

Create a statistical model for generating proportionally similar traffic

Basic steps for replaying traffic

Play traffic

Run either the summary or the model file

Analyze the results

Successes or failures, median, mean, max, 95th

Basic steps for replaying traffic

Play traffic

Run either the summary or the model file

Analyze the results

Successes or failures, median, mean, max, 95th

That's it!

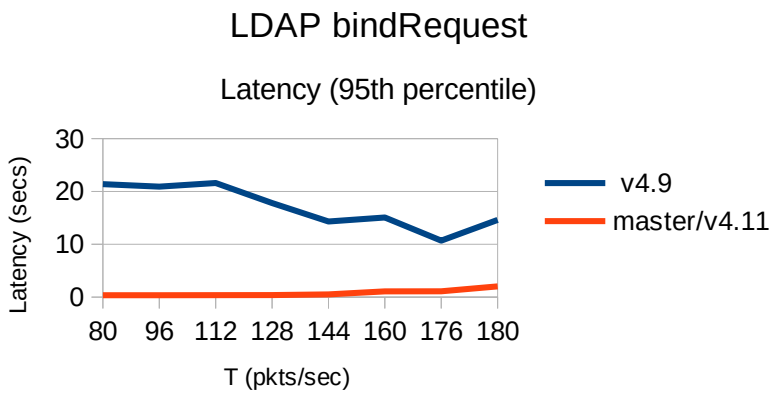
We're fast, 100,000 users, no problems!

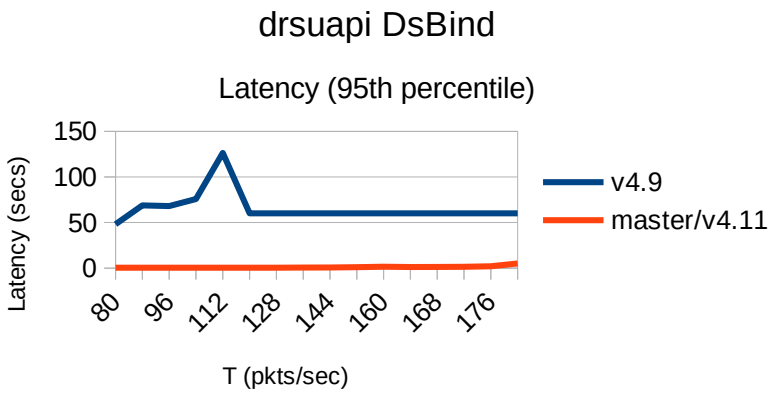
The real difference between Samba 4.9 and the upcoming 4.11

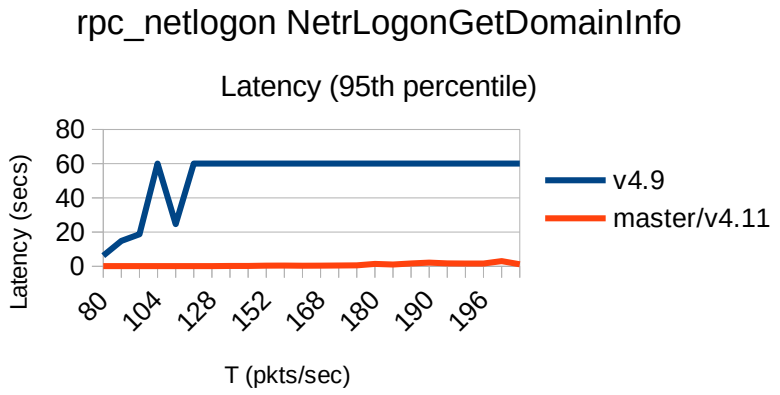
In particular environments, performance can degrade quickly based on database size:

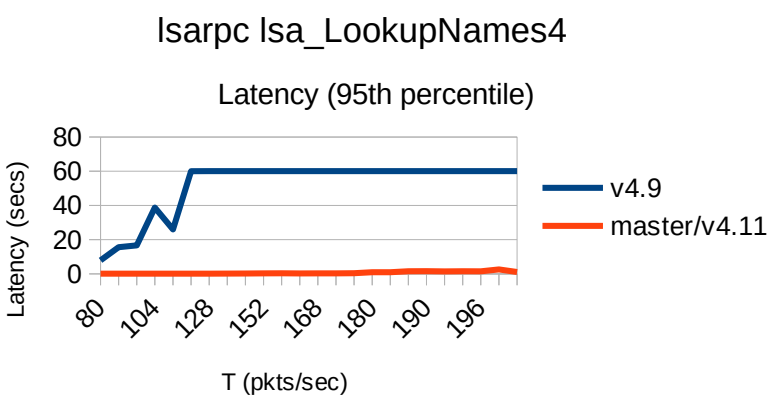
Build	Users	Machines	T (pkts/sec)	Ops/sec	Latency
Master/v4.11	4,400	4,400	240	~165	4.78s
v4.9	4,200	4,200	228	~157	3.24s
Master/v4.11	50,000	62,500	196	~135	4.58s
v4.9*	50,000	62,500	? < 80	? < 43	51.90s

The real difference between Samba 4.9 and the upcoming 4.11

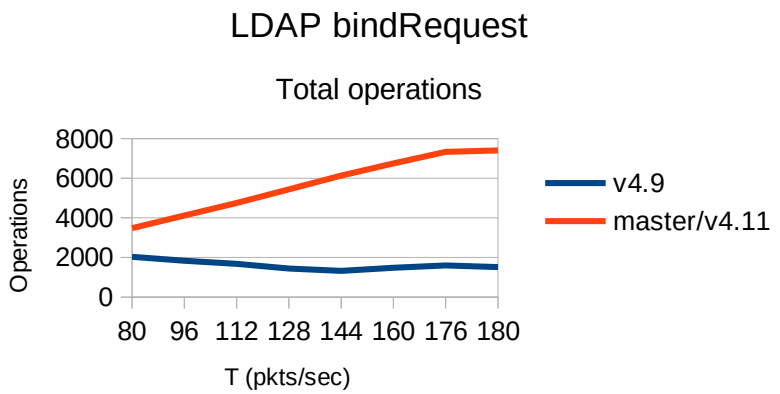


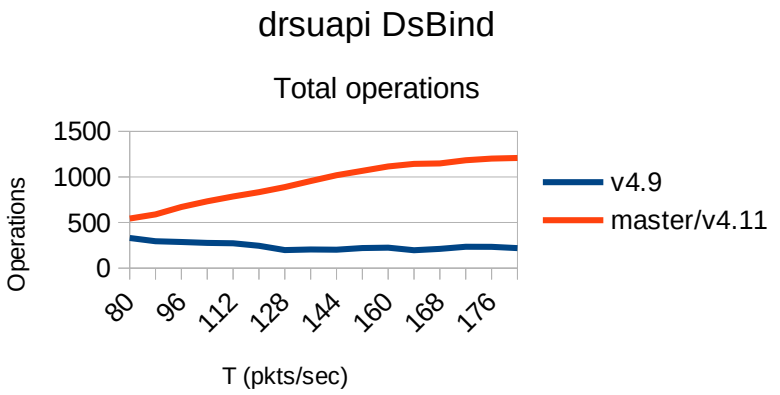


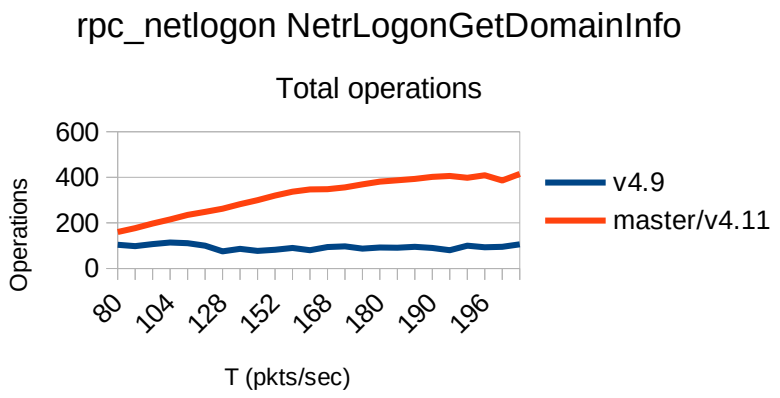


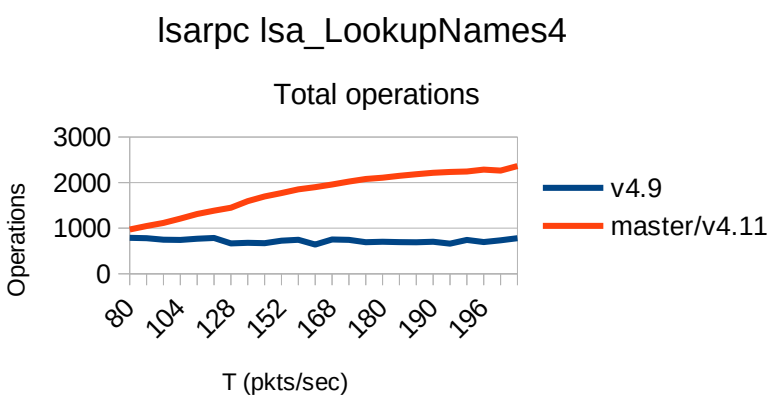


The real difference between Samba 4.9 and the upcoming 4.11









How did we do it?

1. Created a new backup regime
2. Automated set-up of large domains
3. Periodic and targeted traffic replay

Domain backup

A new method of backing up an AD Domain in Samba 4.9 + 4.10

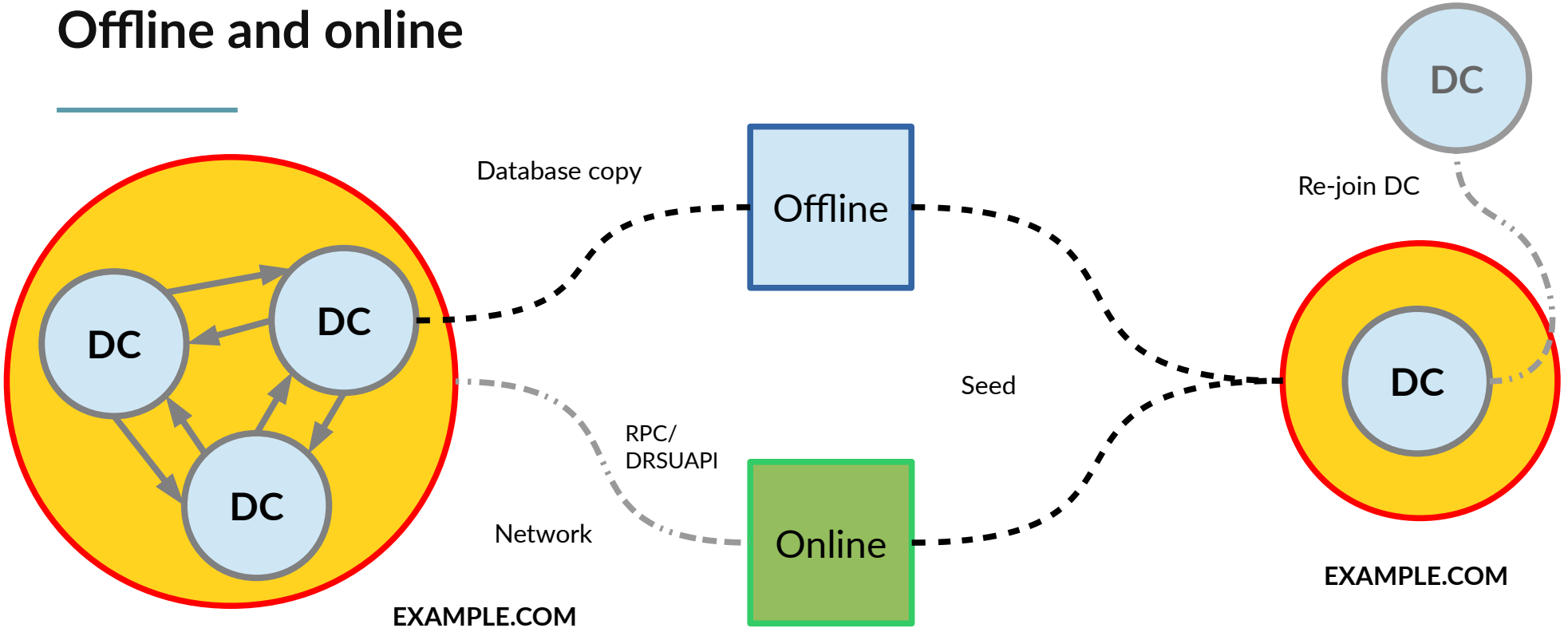
Work done for a separate client

Why?

- Existing samba_backup script had a number of problems
  - With a running DC it wasn't certain to produce a valid copy
  - It was safer than a standard copy, but didn't respect lock ordering
  - Might have caused deadlocks, corrupt or inconsistent (secrets) data

- Single source of truth of the domain data (multi-master replication)
  - Forcing a pristine backup to override corrupt data elsewhere is non-trivial
  - Restoring into competing data, might look replicated due to old versioning
  - Avoid some database inconsistencies by creating a replication (online) backup

Offline and online



`samba-tool domain backup [online|offline]` → Tar file → `samba-tool domain backup restore`

https://wiki.samba.org/index.php/Back_up_and_Restoring_a_Samba_AD_DC

Benefits

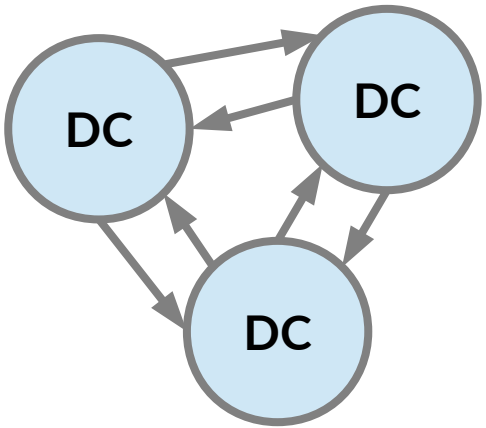
- Reproducible testing is easier, upgrade testing is easier
- Testing under different conditions is much easier
- For those who re-deploy in a certain way, it’s the (almost) ideal tool

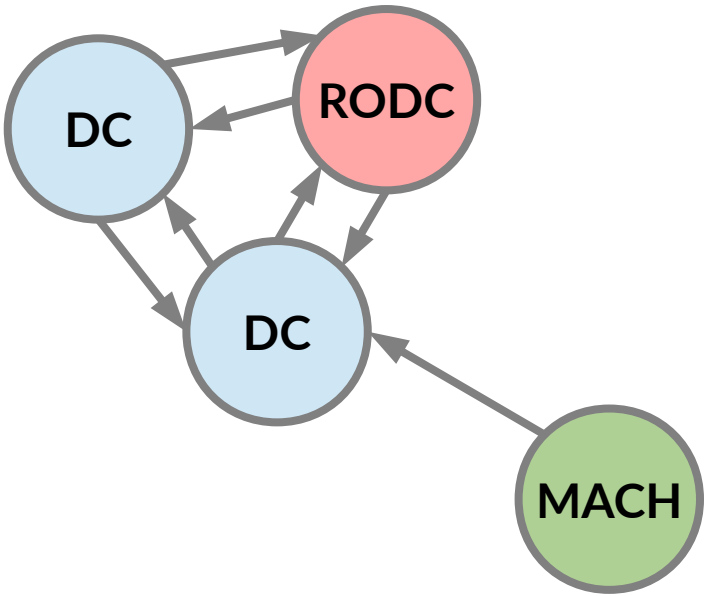
- Still has issues from a user-perspective but very useful as a developer tool
- Validating changes with large domains now becomes possible

Automation

Actually running the traffic runner for real (making it reproducible and periodic)

Automation





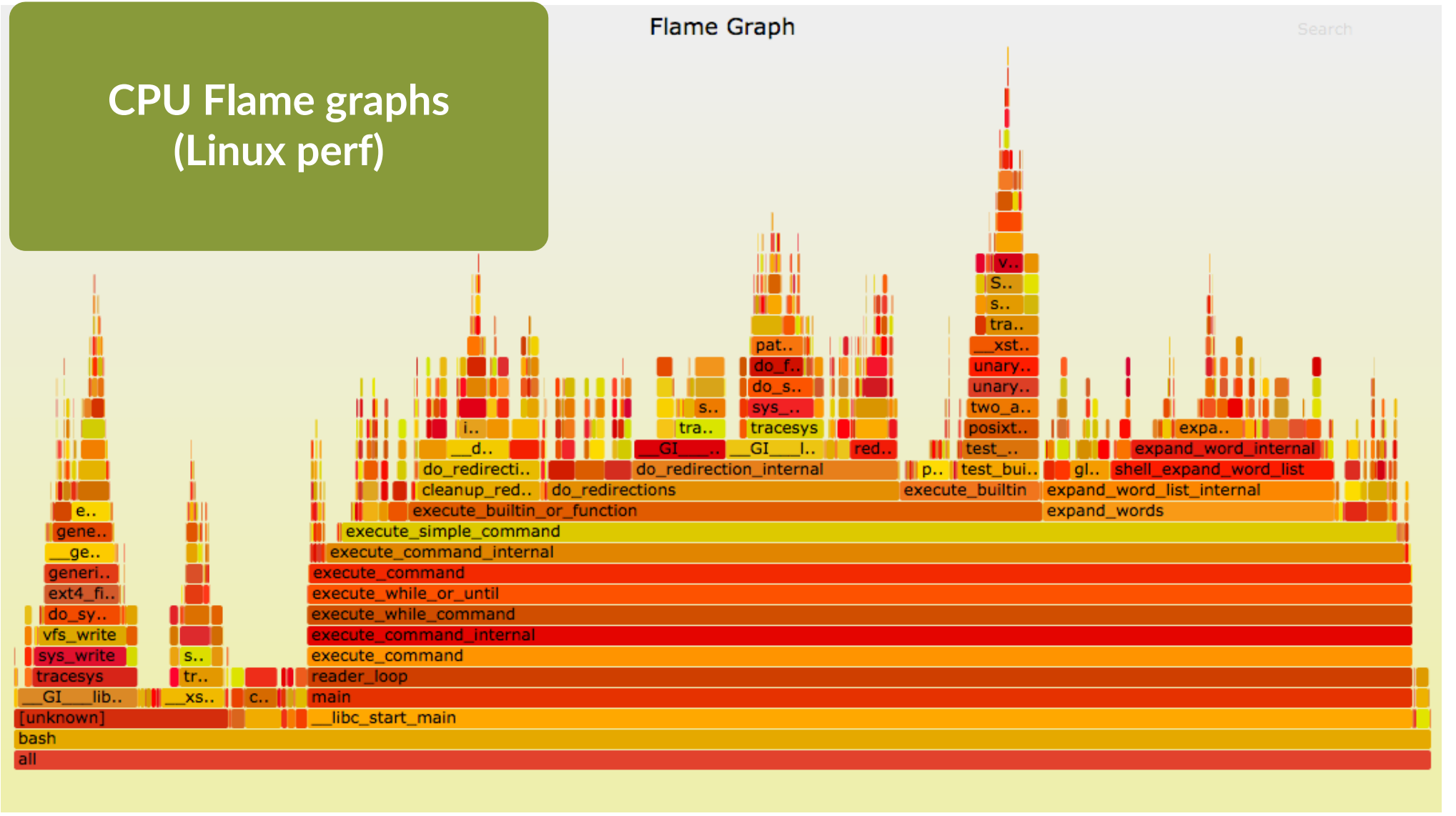
Seed AD domain from a backup

Replicating... forever

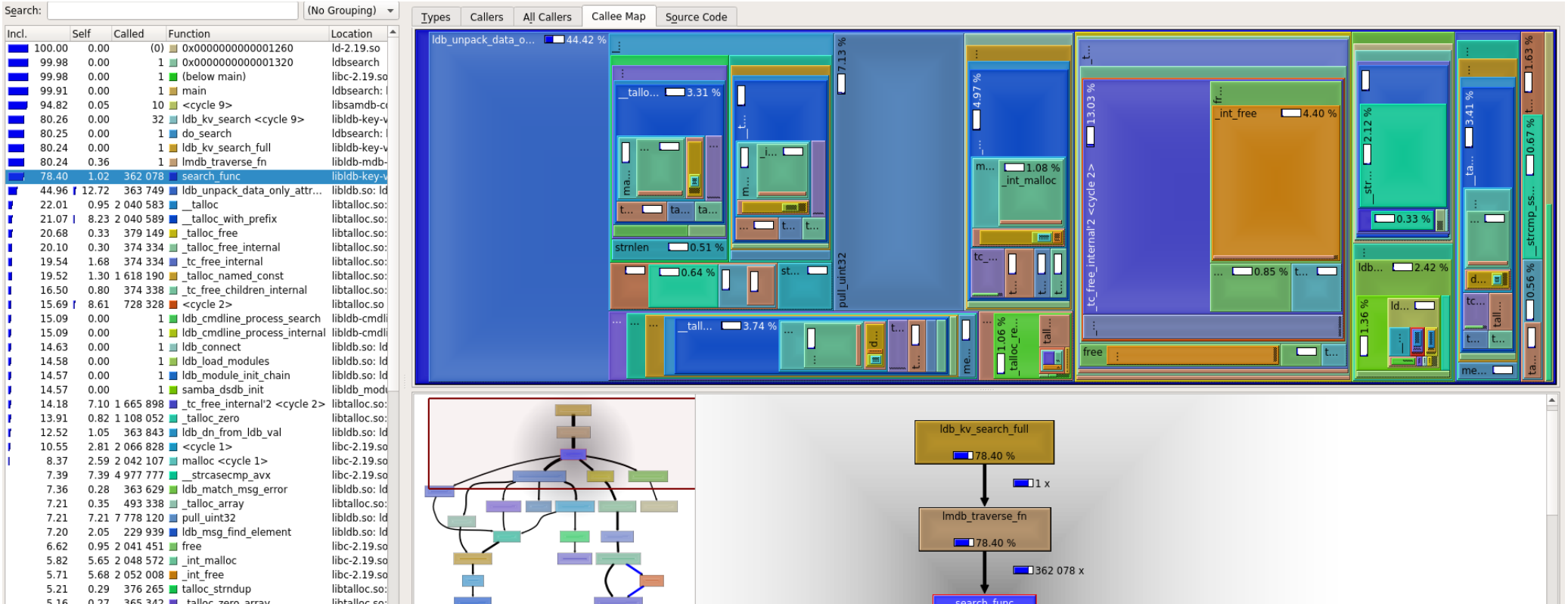
After joining a new domain controller to a restored domain, ongoing replication would never end.

Why doesn’t it only take as long as the join (30 minutes)?

CPU Flame graphs (Linux perf)



Callgrind


Search:
(No Grouping)
Incl. Self Called Function Location
100.00 0.00 (0) 0x0000000000001260 ld-2.19.so
99.98 0.00 1 0x0000000000001320 ldbsearch
99.98 0.00 1 (below main) libc-2.19.so
99.91 0.00 1 main ldbsearch:
94.82 0.05 10 <cycle 9> libsamdb-c
80.26 0.00 32 ldb_kv_search <cycle 9> libldb-key-v
80.25 0.00 1 do_search ldbsearch:
80.24 0.00 1 ldb_kv_search_full libldb-key-v
80.24 0.36 1 lmdb_traverse_fn libldb-mdb-
78.40 1.02 362 078 search_func libldb-key-v
44.96 12.72 363 749 ldb_unpack_data_only_attr... libldb.so: ld
22.01 0.95 2 040 583 __talloc libtalloc.so:
21.07 8.23 2 040 589 __talloc_with_prefix libtalloc.so:
20.68 0.33 379 149 _talloc_free libtalloc.so:
20.10 0.30 374 334 _talloc_free_internal libtalloc.so:
19.54 1.68 374 334 _tc_free_internal libtalloc.so:
19.52 1.30 1 618 190 _talloc_named_const libtalloc.so:
16.50 0.80 374 338 _tc_free_children_internal libtalloc.so:
15.69 8.61 728 328 <cycle 2> libtalloc.so
15.09 0.00 1 ldb_cmdline_process_search libldb-cmdli
15.09 0.00 1 ldb_cmdline_process_internal libldb-cmdli
14.63 0.00 1 ldb_connect libldb.so: ld
14.58 0.00 1 ldb_load_modules libldb.so: ld
14.57 0.00 1 ldb_module_init_chain libldb.so: ld
14.57 0.00 1 samba_dsdb_init libldb_mod
14.18 7.10 1 665 898 _tc_free_internal'2 <cycle 2> libtalloc.so:
13.91 0.82 1 108 052 _talloc_zero libtalloc.so:
12.52 1.05 363 843 ldb_dn_from_ldb_val libldb.so: ld
10.55 2.81 2 066 828 <cycle 1> libc-2.19.so
8.37 2.59 2 042 107 malloc <cycle 1> libc-2.19.so
7.39 7.39 4 977 777 __strcasecmp_avx libc-2.19.so
7.36 0.28 363 629 ldb_match_msg_error libldb.so: ld
7.21 0.35 493 338 _talloc_array libtalloc.so:
7.21 7.21 7 778 120 pull_uint32 libldb.so: ld
7.20 2.05 229 939 ldb_msg_find_element libldb.so: ld
6.62 0.95 2 041 451 free libc-2.19.so
5.82 5.65 2 048 572 _int_malloc libc-2.19.so
5.71 5.68 2 052 008 _int_free libc-2.19.so
5.21 0.29 376 265 talloc_strndup libtalloc.so:
Types
Callers
All Callers
Callee Map
Source Code
ldb_unpack_data_o... 44.42 %
_tallo... 3.31 %
strnlen 0.51 %
0.64 %
pull_uint32
7.13 %
4.97 %
m... 1.08 %
_int_malloc
tc_...
_tc_free_internal'2 <cycle 2> 13.03 %
_int_free 4.40 %
0.85 %
free
2.12 %
_str... 0.33 %
ldb... 2.42 %
1.36 %
3.41 %
_ta...
1.63 %
_strcmp_ss...
0.67 %
0.56 %
_tall... 3.74 %
1.06 %
_talloc_re...
ldb_kv_search_full
78.40 %
1 x
lmdb_traverse_fn
78.40 %
362 078 x

Print debugging

top (htop/iotop)

trial and error

basic arithmetic

gdb (attach to pid)

perf top

luck

Lessons

- It turns out there was a bug in the backup code, but it found real performance issues that we then fixed

- Accidentally doing the wrong thing means running out of memory quickly with a large database.
    - Piecemeal growth ≠ dealing with everything at once

- Learning how LMDB behaves - completely different to our previous database backend (copy-on-write)

Database changes motivated by our traffic statistics analysis

The effort required to improve figures to the degree shown in the upcoming 4.11 release

Changes to our database and core database services

- Implemented indexing of >= and <= queries
- Searching across 300,000 records went from ~4 sec to ~0.2 sec
- Replication more responsive
  - Example testing: ~4 sec to ~1 sec for no data, ~6 sec to ~1 sec for 200 new records

- Paged results – Reducing required memory usage from GB to KB or MB
- LDAP efficiency improvements – Impossible operations can now be done (search)
- Subtree renames – Large renames from days to less than an hour

Changes to the database record storage format

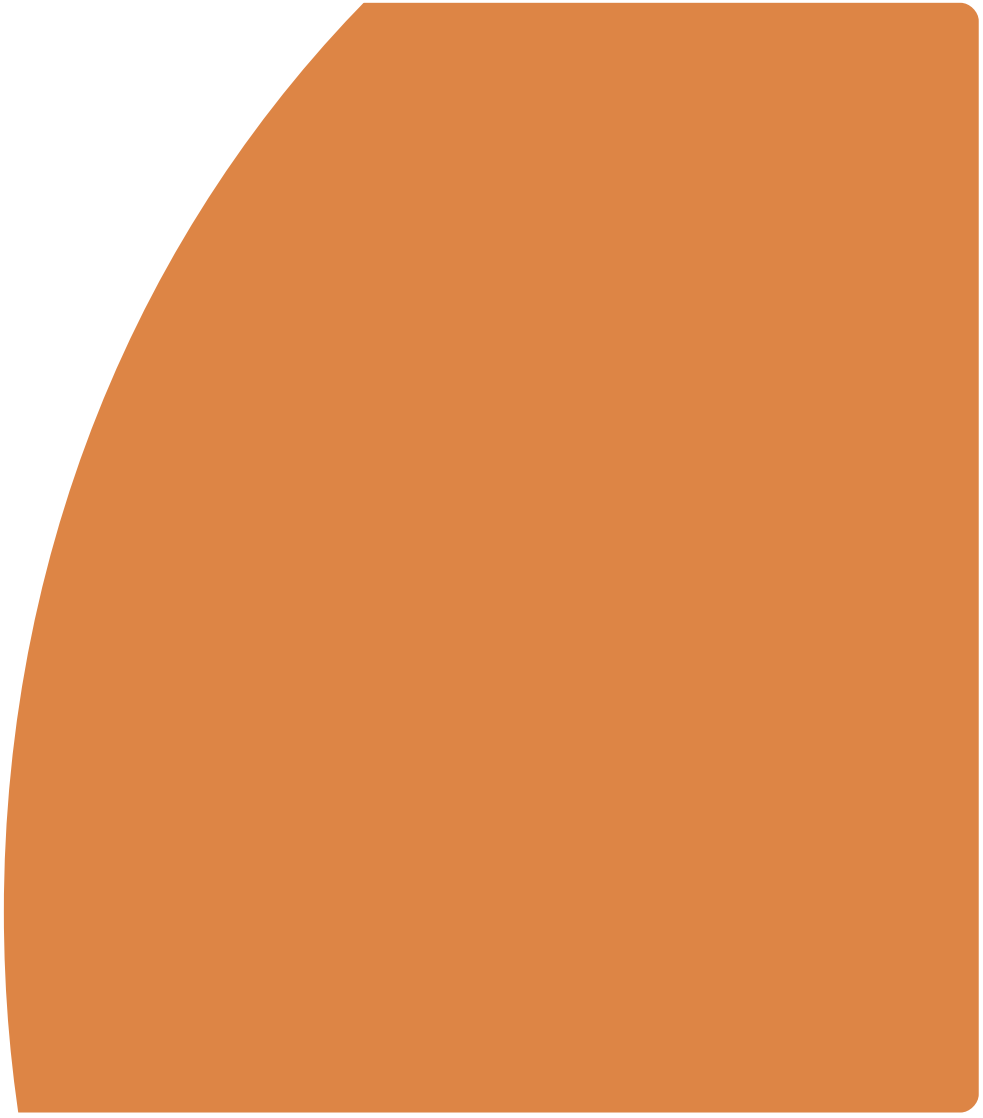
CN=test_user,CN=Users
2 elements
name | 1 value | test_user
member | 5 values
group_1, group_2
group_3
group_4, group_5

CN=test_user,CN=Users
2 elements
name | 1 | member | 5
test_user
group_1, group_2
group_3
group_4, group_5

Only read data when necessary, offset data is stored at the beginning, massive size reduction.
Applies to more than just storage of group membership but also big blobs of data (e.g. photos).

2012 R2 Schema Support

Pre-requisite for 2012 R2 functional level and newer AD (security) features



2012 R2 schema support

- First half (replication support) is already slated for 4.11
- Second half (default schema) is currently waiting final sign-off and tweaks

- Integration with newer Windows servers no longer causes ongoing replication problems and 2012 R2 servers can join Samba without issue now

GPO import/export

A new way of copying over a SYSVOL that functions across domains

Exports to XML with XML entities

MS-GPNRPT

MS-GPFR

fdeploy1.ini

MS-GPWL

MS-GPOL

MS-GPSCR

User/Documents & Settings

.xml

MS-GPREG

registry.pol

MG-GPFAS

MS-GPOD

MS-GPAC

Machine/Microsoft/Windows NT/SecEdit

MS-GPSI

.aas

MS-GPDPC

audit.csv

GptTmpl.inf

MS-GPPREF

MS-GPSB

MS-GPIPSEC

MS-GPREF

Using GPO Import/Export

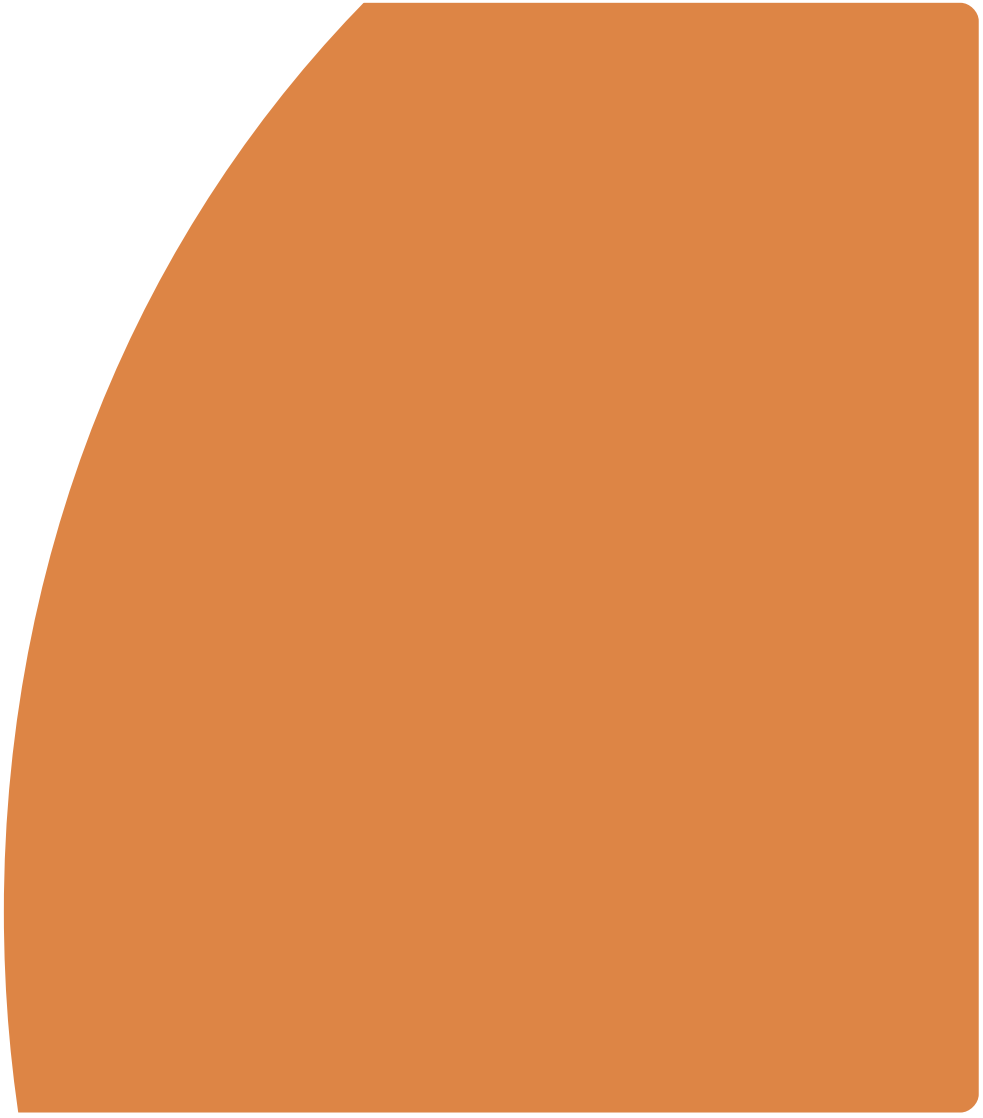
```
samba-tool gpo backup
samba-tool gpo restore

samba-tool gpo backup --generalize --entities=$OUT_PATH
samba-tool gpo restore --entities=$IN_PATH
```

```
<!ENTITY SAMBA____USER_ID_____7b7bc2512ee1fedcd76bdc68926d4f7b__ "Guest">
```

https://wiki.samba.org/index.php/GPO_Backup_and_Restore

Re-indexing

Example of an operation where our tooling failed and discovered improvements to join time of 2-4x



Re-indexing timings (mm:ss.ss)

```
100,000 users approx 230,000 records.
  Hash size         re-index time
      1,000              14:42.06
     10,000               1:59.56
    100,000                 39.92
    200,000                 37.48
    300,000                 43.16

50,000 users approx 110,000 records.
  Hash size         re-index time
      1,000               3:46:93
     10,000                 37:29
    100,000                 18.95
```

20x improvement

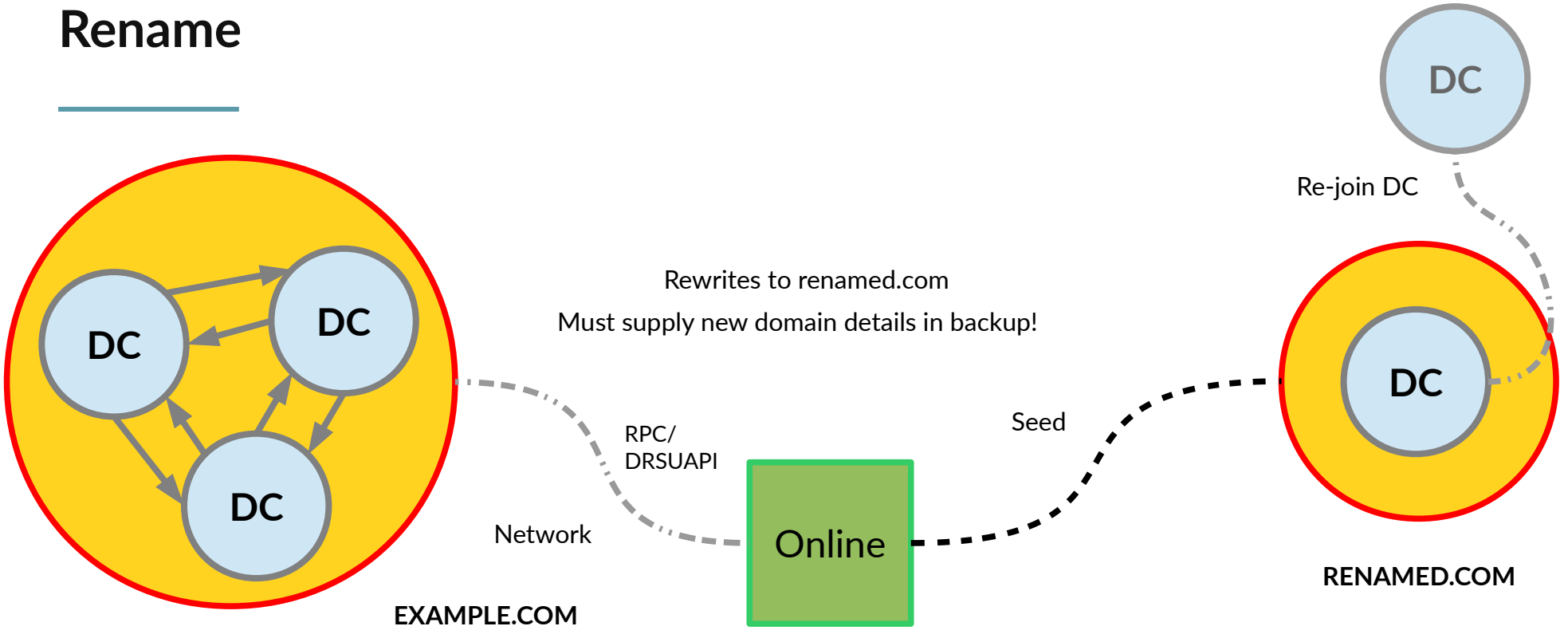
Basically a one line change

Domain rename

Create testing environments and lab domains (without passwords and secrets)

Good for load testing an isolated network

Rename



`samba-tool domain backup rename` → Tar file → `samba-tool domain backup restore`

https://wiki.samba.org/index.php/Create_a_samba_lab-domain

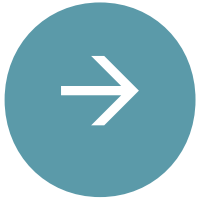
Final takeaways

1) We've been investing significant amounts of effort into more realistic testing
2) Normals tools and rules do not apply (to understand high load and concurrency)
3) Identified new areas for improvement (increase responsiveness, robustness and avoid issues arising in the future)

4) Important pieces (backup) that helped deliver the scale and breadth of performance improvements for 4.11 were funded independently

Thanks

...



garming@catalyst.net.nz

garming@samba.org

linkedin.com/in/garming-sam

catalyst