



Advancing Digital Storage Innovation



# Lustre Samba improvements

Alexey Lyashkov <[Alexey\\_Lyashkov@xyratex.com](mailto:Alexey_Lyashkov@xyratex.com)>

# Agenda

- Introduction
- Typical samba usage
- Samba scalability issues
  - Data access bottleneck
  - Metadata scalability issue

# Introduction

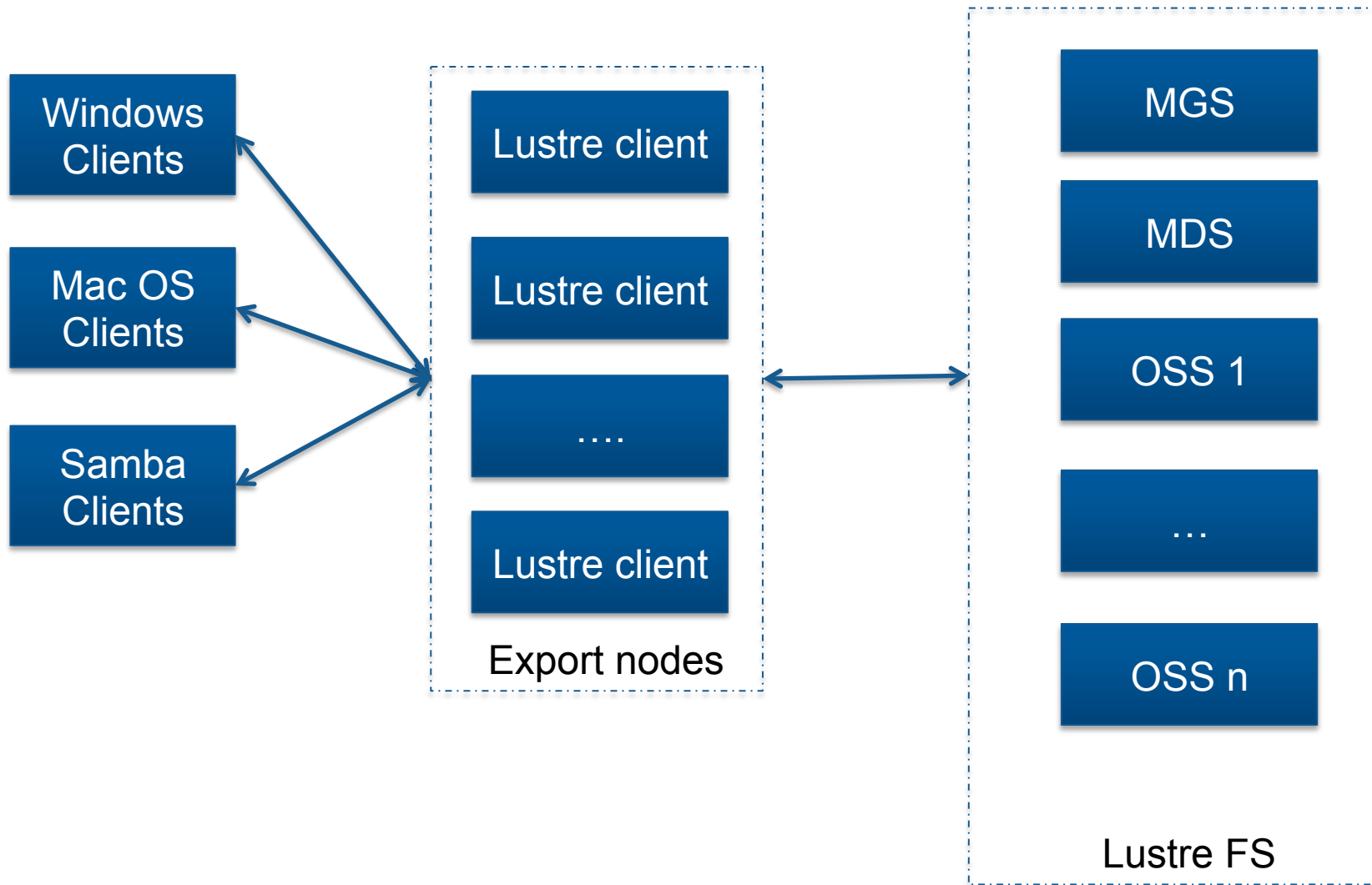
Samba is it Windows SMB protocol implementation for a POSIX compatible operation systems.

Samba clients exist on Windows and MacOS OS'es, so that is on of ways to access to the lustre FS in mixed networks where windows or mac os clients have.

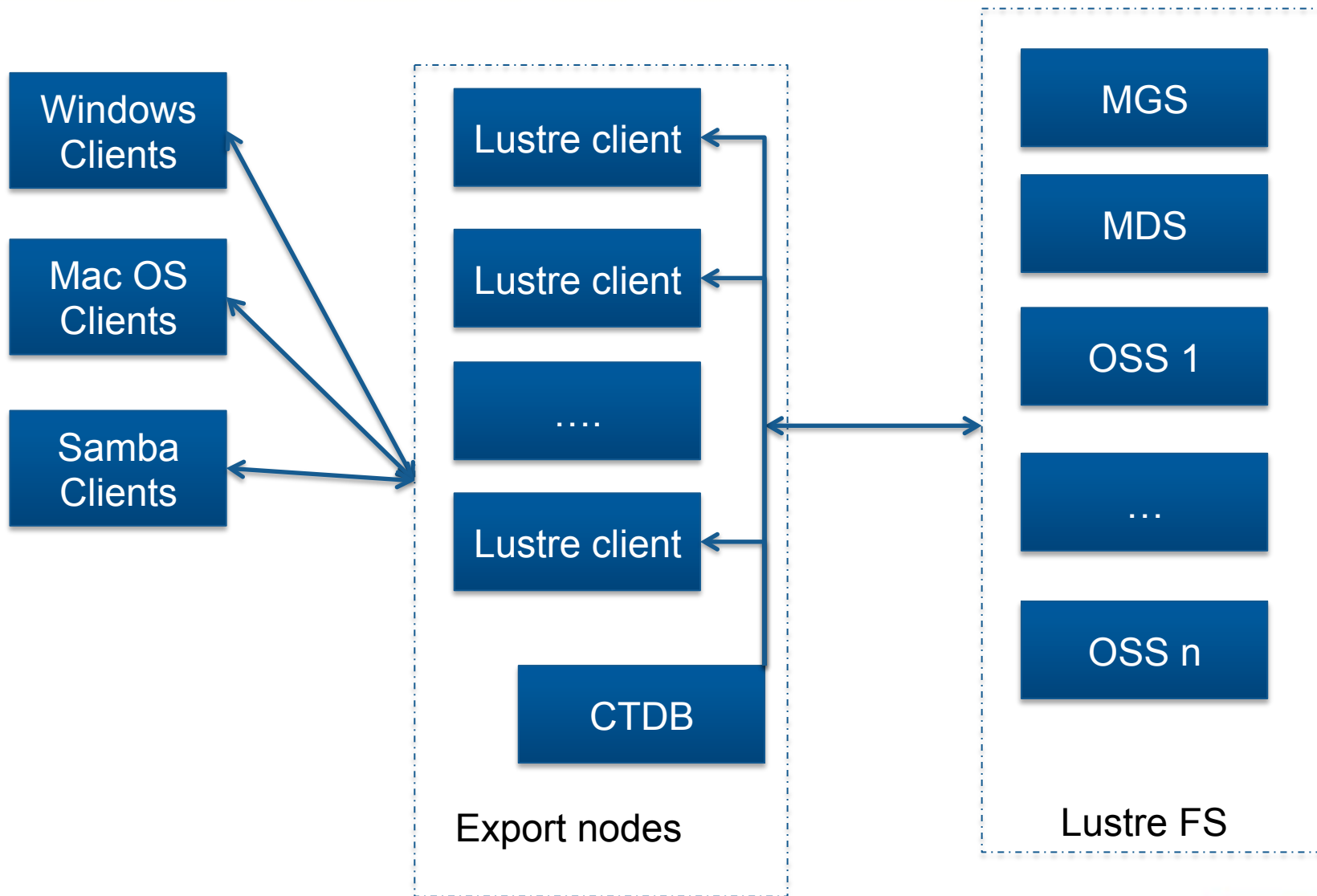
Samba have own access control functions and own node synchronization primitives.

That report describes a some bottlenecks in using Samba to the export Lustre FS and ways to solve it.

# Typical SAMBA usage scheme



# Clustered Samba – way to solve data bottlenecks

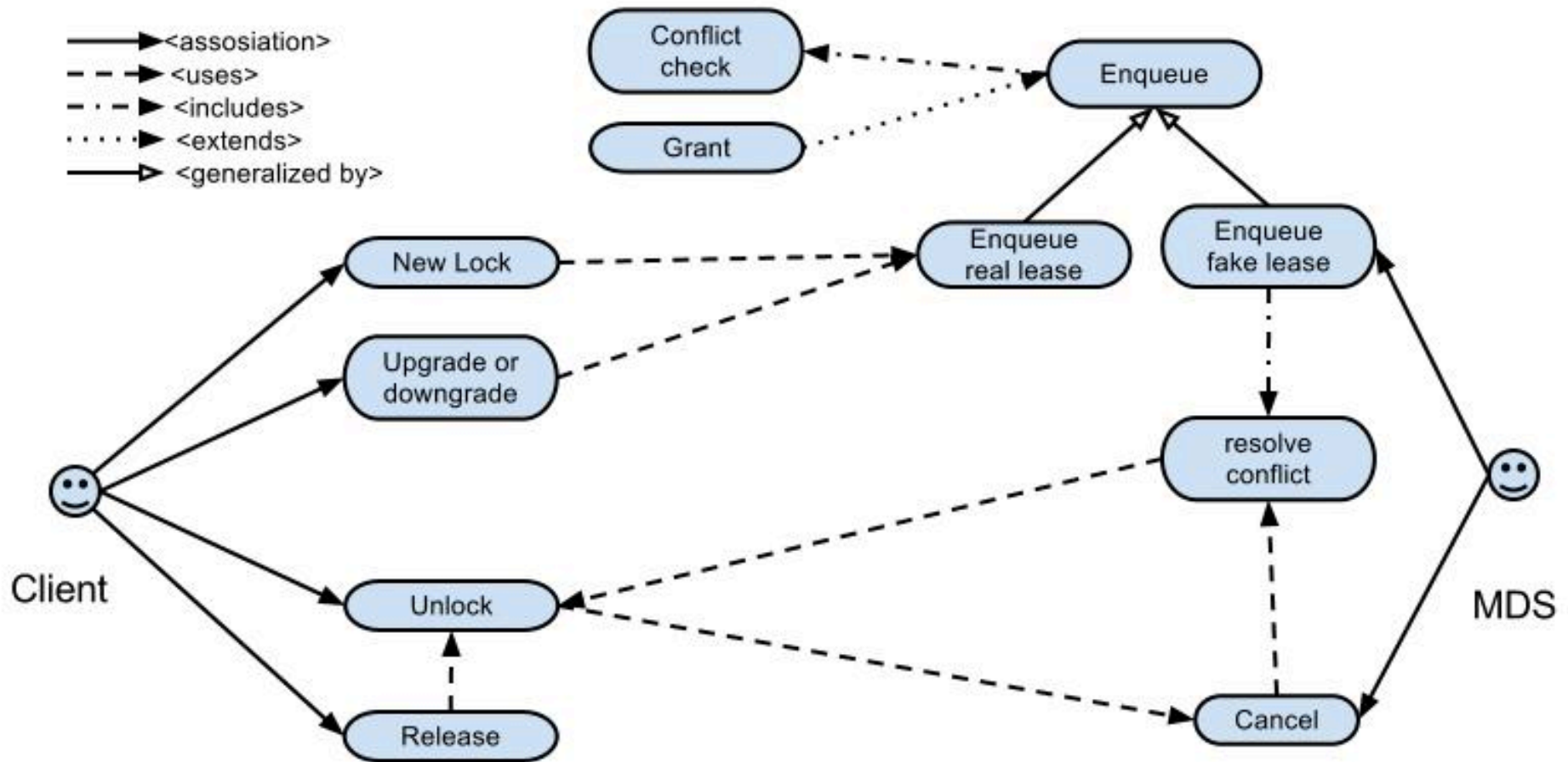


# Lease Locking

Main idea of lease lock – create a some lock incompatible with open/truncate operations to provide an ability to flush a data to persistent storage.

Lease locks introduced in early 2.6 releases for a NFS v4 support but now uses in both – Samba and NFS v4 to make writeback cache work correctly.

# Lease locking proposal



## Eviction and recovery

- Client may don't know about eviction for long time. So never release own lease lock. May be solved with RENEW\_LEASE command, similar to NFS v4
- Client without release locks have open files, but after recovery client don't reopen a files, so new lease locks can't conflicts with these files. That is produce inconsistent between clients with that type sincronization.



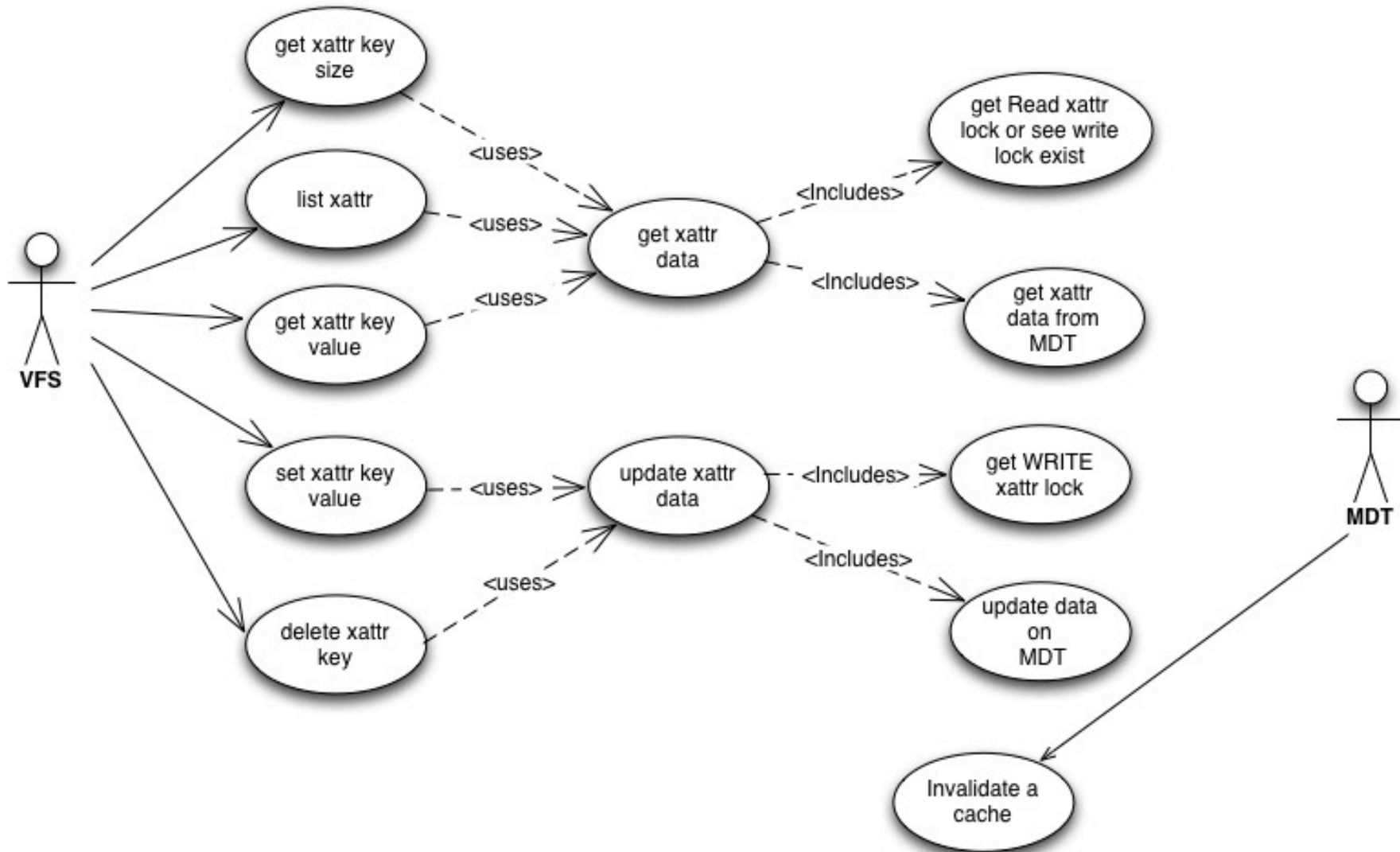
## XATTR caching – speedup optimizations

Samba uses an extended attributes data to store own security attributes and check it's on each md object access.

That is reason to have a directory listing at samba share to be 3-4x slow in comparing to the listing on lustre mount.

SELinux and VFS capability data is an additional reasons to have a XATTR caching on client side.

# XATTR caching



## XATTR caching highlights

- Use a write lock on client side to minimize a number of cache flushes when modification executed on same node
- R/O cache work same same as other caches used on lustre client
- Will simple – no locks for each xattr key, just protect a whole data in inode.
- Will inode bits lock – so we may combine with other locks in future.

## Conclusions

- Implementing a lease locking will help with clustered Samba or (and) NFS v4/pNFS export nodes.
- Implementing a XATTR cache will have additional usage for a Selinux, VFS capability so provide better integration into lustre source tree.
-

Final

Thanks!

Alexey Lyashkov <Alexey\_Lyashkov@xyratex.com>