
# Kerberos Integration with Lustre

Jon Diprose, BMRC, University of Oxford

LUG-UK 2022

# About us

- Biomedical Research Computing
- Bare metal HPC & OpenStack Cloud
- Departmental facility
- Support a wide range of biomedical research
  - Genomics – large sequential read
  - Imaging – millions of tiny files
  - …and everything in between
- Historically Spectrum Scale on DDN kit
- Collaboration with DDN to show Lustre works for us

# Why Lustre & why Kerberos?

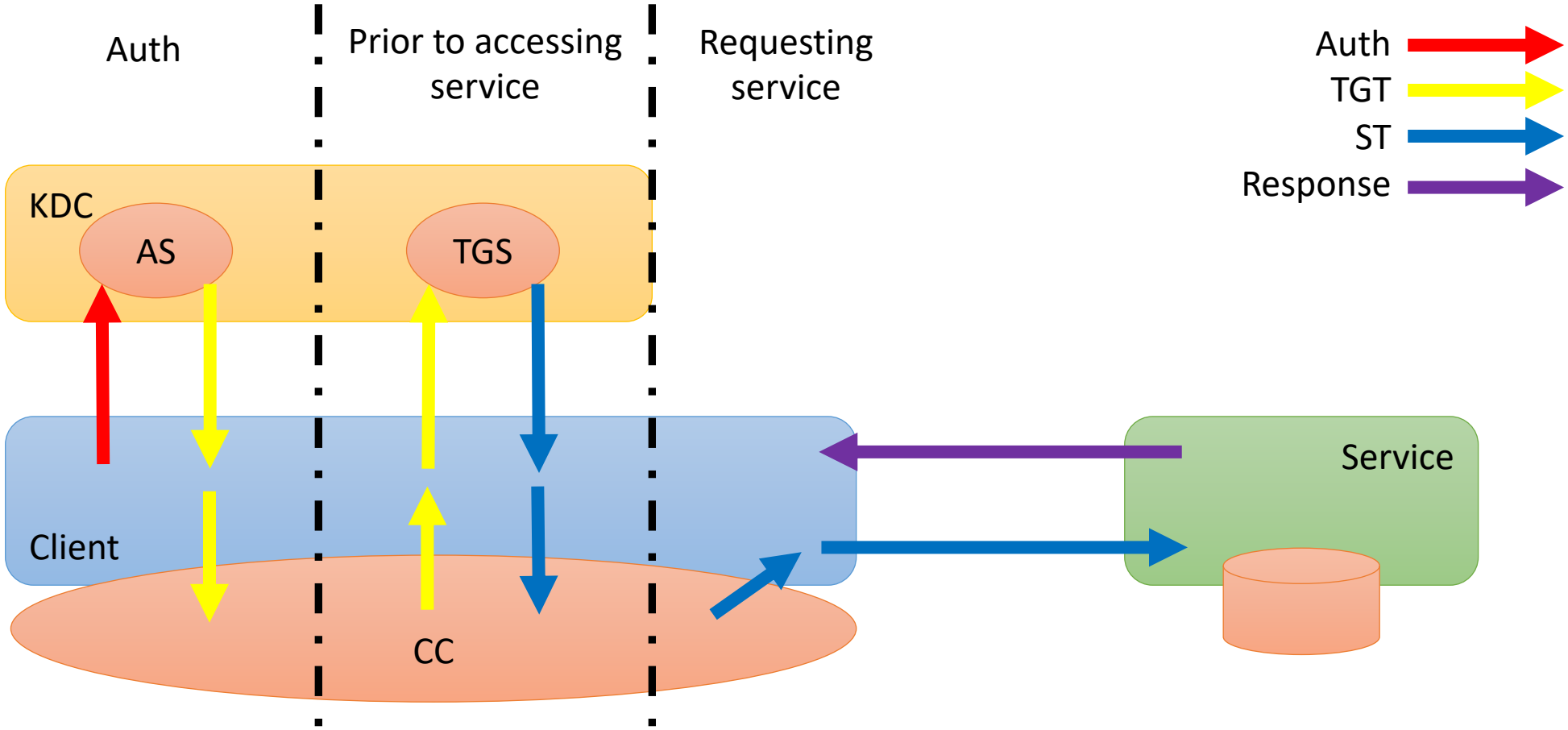
- Multi-tenancy
  - Particularly the Secure Lustre work from Sanger
- Lustre Kerberos support has relevant features
  - Strong auth (krb5n)
  - Message checksumming and integrity (krb5a, krb5i)
  - Encryption in flight (krb5p)
- Kerberos already in use
  - Using FreeIPA for identity management
  - The upstream for RHEL IdM

# Kerberos concepts

- Key Distribution Centre (KDC)
  - Authentication Server (AS)
  - Ticket Grant Server (TGS)
- Ticket
  - Ticket granting ticket (TGT)
  - Service Ticket (ST)
  - Stored in Credential Cache (CC)
- Principal
  - A thing that can have a ticket
  - User, Host, Service
- Keytab
  - Encrypted password store

- User authenticates to AS
  - Receives TGT on success
  - TGT stored in CC
- User presents TGT to TGS and requests access to a service
  - Receives ST on success
  - ST stored in CC
- User presents ST to service
  - Receives service if ST is valid

# Kerberos message flow



# First head- Credential cache

- Where the tickets are persisted
- RHEL6 used FILE, RHEL7 uses KEYRING, RHEL8 uses KCM
- Lustre uses FILE
- /etc/krb5.conf, [libdefaults] section
  - default_ccache_name = KEYRING:persistent:%{uid}
- becomes
  - default_ccache_name = FILE:/tmp/krb5cc_%{uid}

# Second head – Name resolution

- Service principals are of the form service/hostname.domain@REALM
- “On networks for which name resolution to IP address is possible, like TCP or InfiniBand, the names used in the principals must be the ones that resolve to the IP addresses used by the Lustre NIDs.”
- We are using multirail, so the service is available on more than one IP address

# Second head – Name resolution

- Server-side:
  - Server hostname must resolve to one of the client-facing IPs
  - All client-facing IPs must resolve to the hostname of the server
- Client-side:
  - All server-facing IPs must resolve to the hostname of the client
- Mount using IPs and not hostnames
- That DNS config feels wrong…

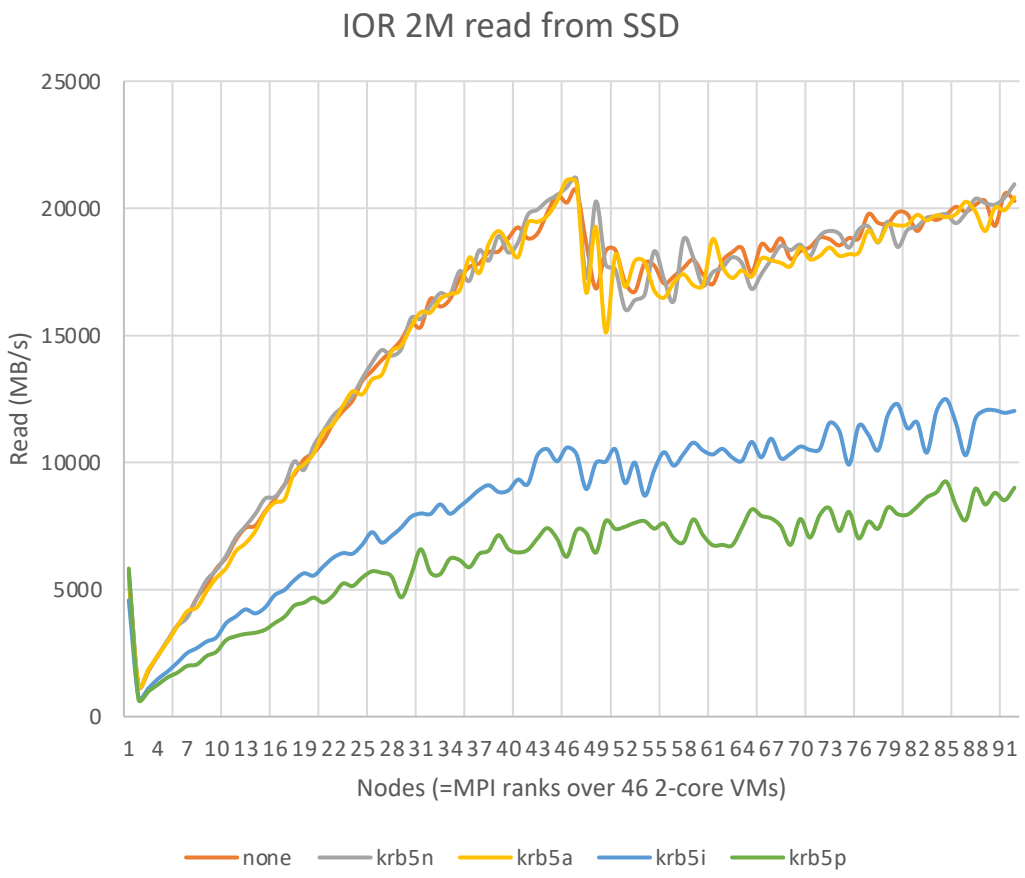
# Third head – Key quota exhaustion

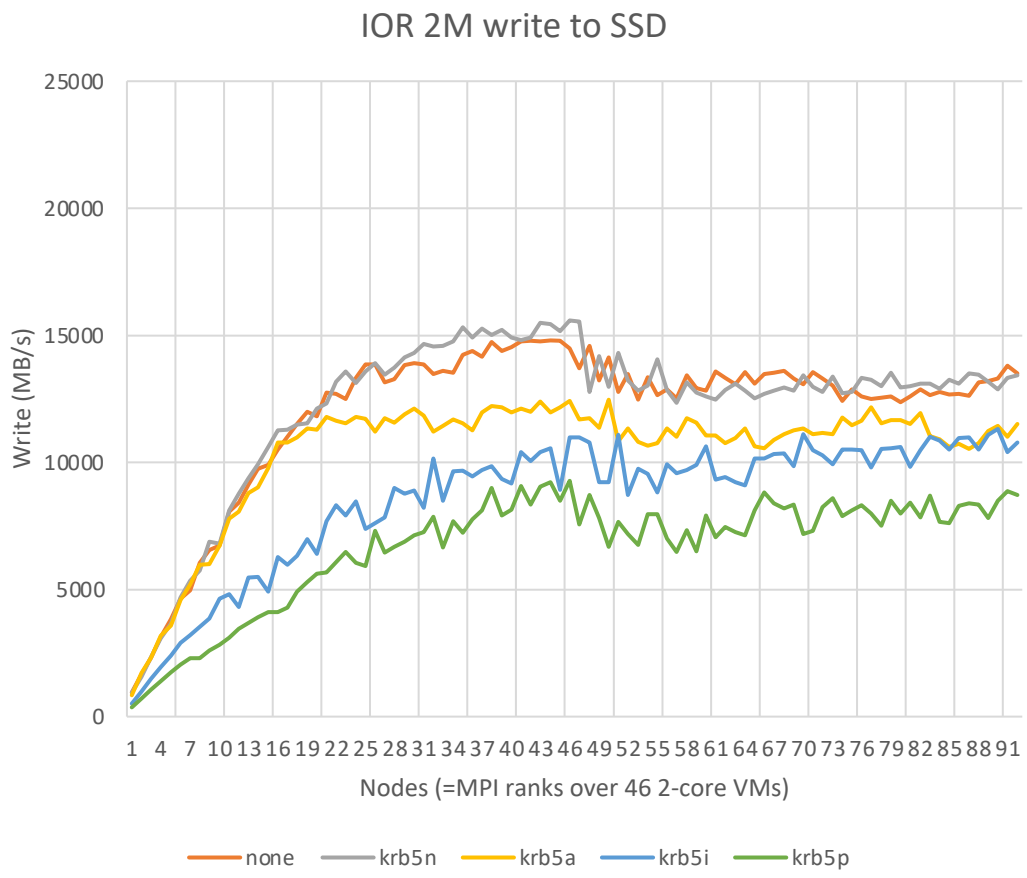
- Kerberos used to set up a Lustre GSS context
  - Similar to kerberized NFS
- Context represented by an lgss key stored in the kernel key store
  - Kernel key store has per-user quota on key count and size
- The lgss keys are attached to a session keyring
  - specific to that ssh session
- Calling “lfs flushctx” destroys the lgss keys in that session
- Logging out destroys the session keyring but not the lgss keys
- Orphaned keys persist and count against the per-user key quotas

# Third head – Key quota exhaustion

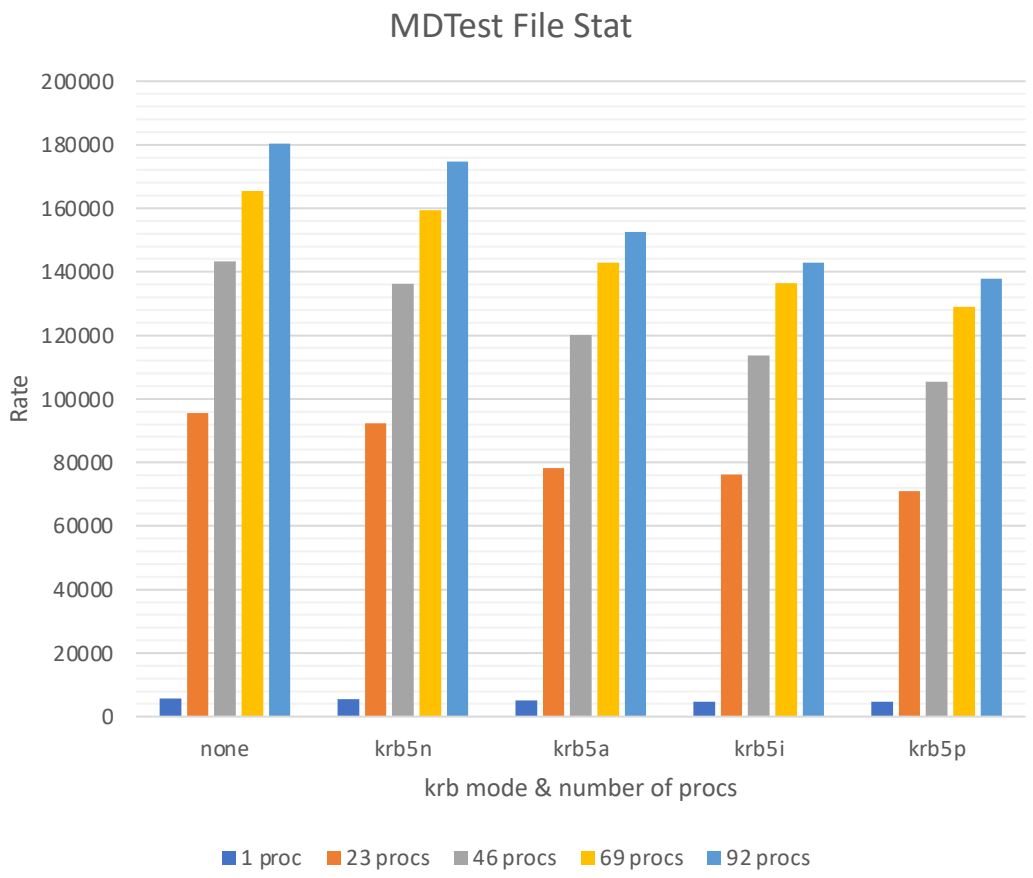
- Increase key quota
  - /etc/sysctl.d/90-es-key-quota.conf
    - kernel.keys.maxkeys=6800
    - kernel.keys.maxbytes=60000
- Session keyring created and destroyed by pam_keyinit.so
- Adapted pam_keyinit to call “lfs flushctx -r” prior to session keyring destruction, as pam_keyinit_lfs.so
  - Hopefully on its way to Whamcloud to be done properly
  - sed -i ‘s/pam_keyinit.so/pam_keyinit_lfs.so/g’ /etc/pam.d/sshd

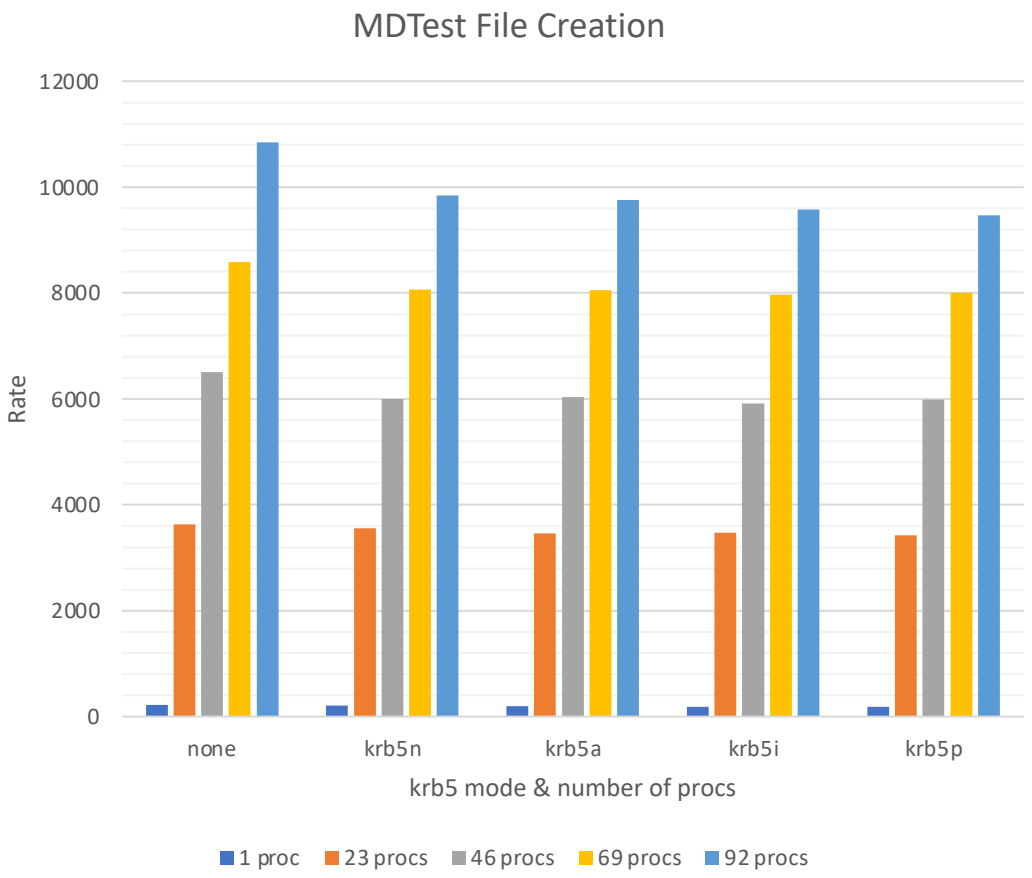
# Numbers - ior


IOR 2M read from SSD
Read (MB/s)
25000
20000
15000
10000
5000
0
1 4 7 10 13 16 19 22 25 28 31 34 37 40 43 46 49 52 55 58 61 64 67 70 73 76 79 82 85 88 91
Nodes (=MPI ranks over 46 2-core VMs)
none
krb5n
krb5a
krb5i
krb5p


IOR 2M write to SSD
Write (MB/s)
25000
20000
15000
10000
5000
0
1 4 7 10 13 16 19 22 25 28 31 34 37 40 43 46 49 52 55 58 61 64 67 70 73 76 79 82 85 88 91
Nodes (=MPI ranks over 46 2-core VMs)
none
krb5n
krb5a
krb5i
krb5p

# Numbers - mdtest



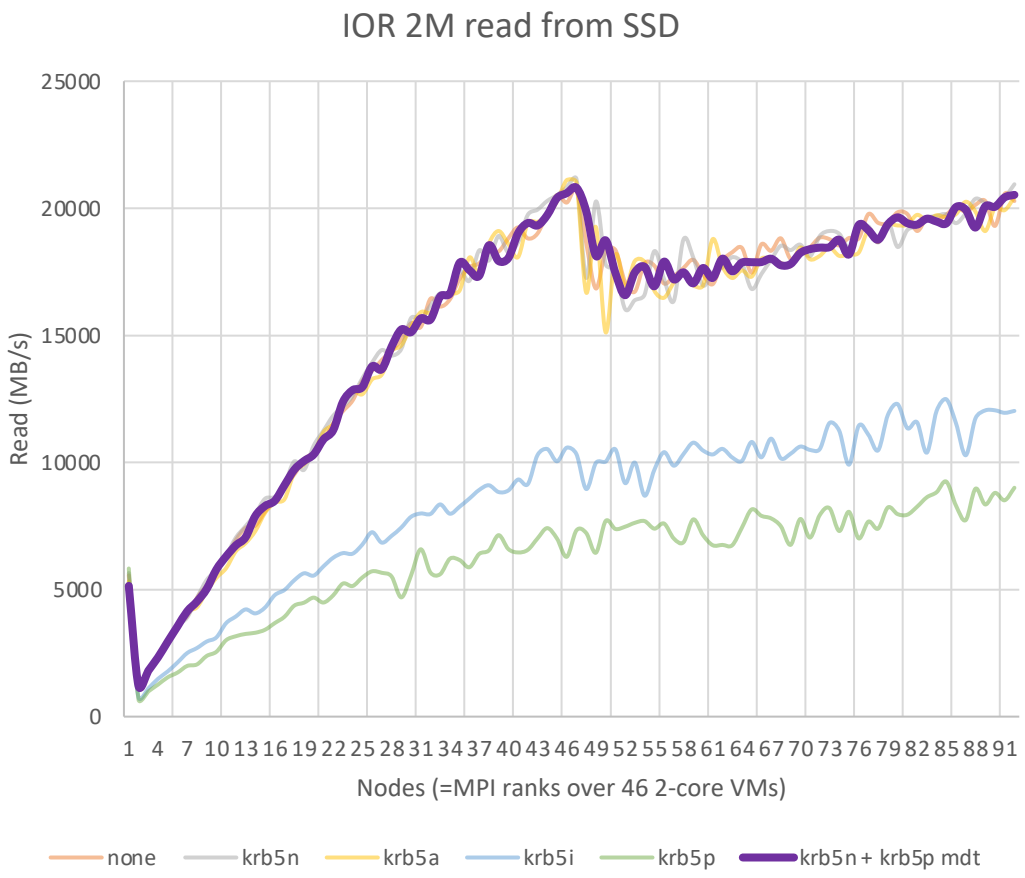
# Client-side encryption + Kerberos

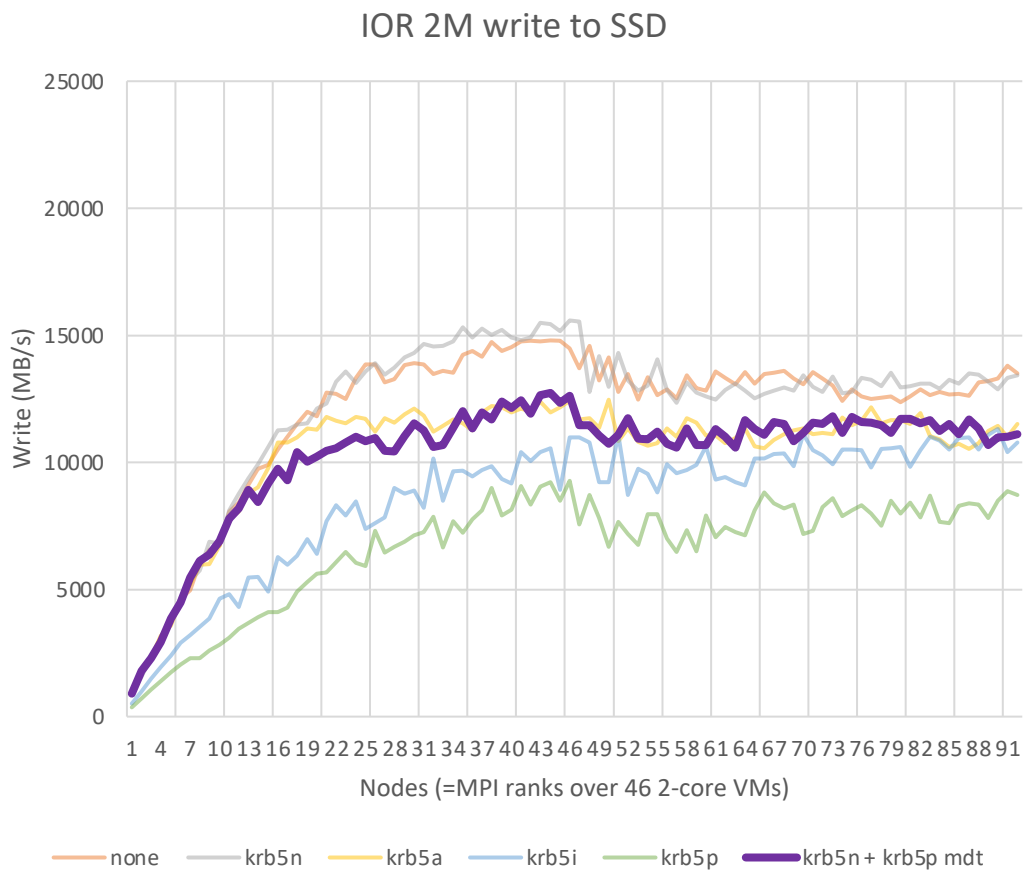
- krb5p not ideal
  - performance sucks
  - doesn’t solve on-disk encryption requirements
  - Lustre Kerberos is on or off – sort of
- Client-side encryption likely to be a better approach
  - https://www.youtube.com/watch?v=jicZ6bEB8lU (Buisson LUG2022)
  - Per-directory config
  - No impact on server
  - Client OS >= RHEL8.1
- 2.14 encrypts file content, 2.15 adds file name encryption
- But still want strong auth, so plus krb5n

# Client-side encryption + Kerberos

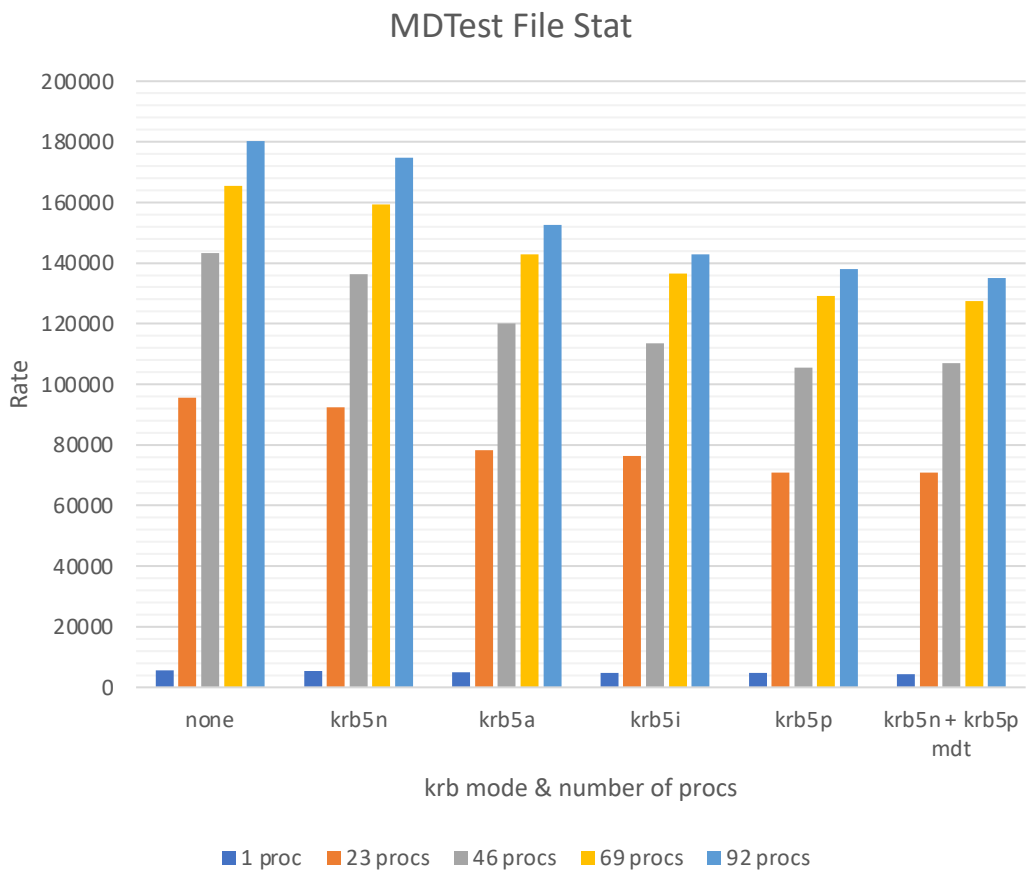
- Other file metadata still plaintext
- Lustre Kerberos support allows some fine-grained configuration
  - <target>.srpc.flavor.<network>[.<direction>]=flavor
    - <target> can be filesystem name, or specific MDT/OST device name
    - <network> is the LNet network name, or ‘default’ for all
    - <direction> can be one of cli2mdt, cli2ost, mdt2mdt, mdt2ost
- Strong auth + encrypted metadata
  - exafs.srpc.flavor.default=krb5n
  - exafs.srpc.flavor.default.cli2mdt=krb5p

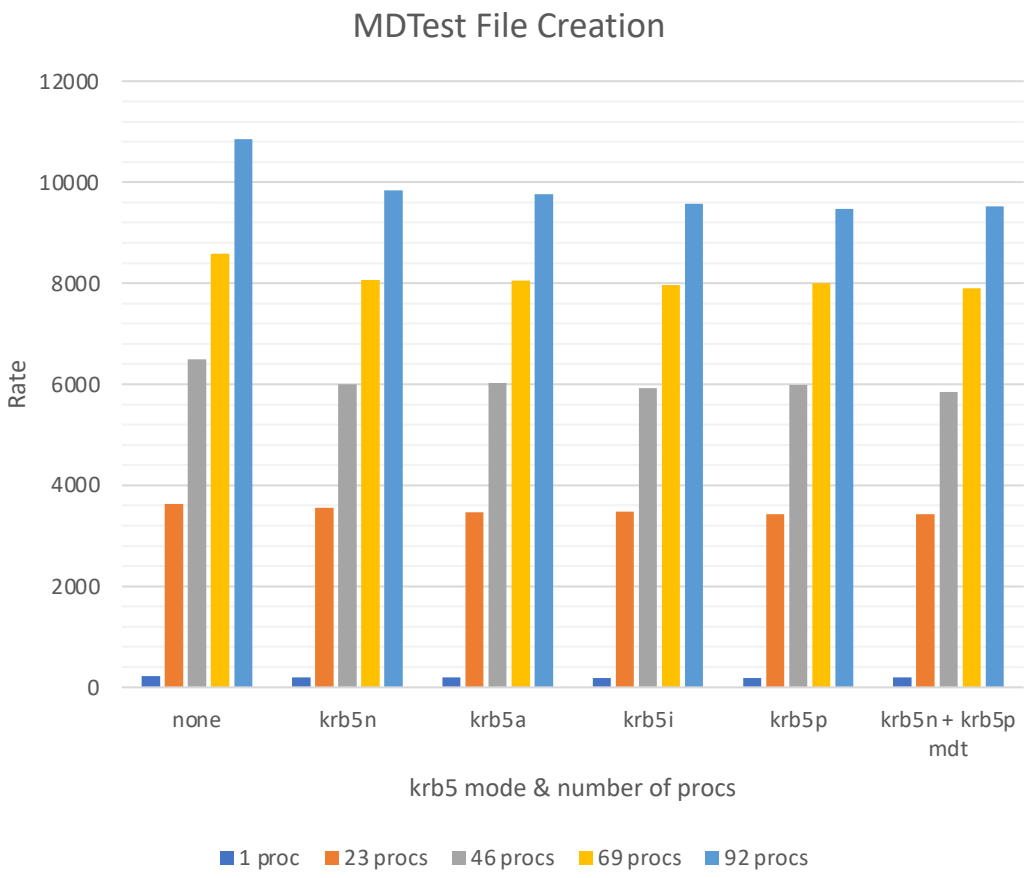
# Numbers - ior


IOR 2M read from SSD
Read (MB/s)
25000
20000
15000
10000
5000
0
1 4 7 10 13 16 19 22 25 28 31 34 37 40 43 46 49 52 55 58 61 64 67 70 73 76 79 82 85 88 91
Nodes (=MPI ranks over 46 2-core VMs)
none
krb5n
krb5a
krb5i
krb5p
krb5n + krb5p mdt


IOR 2M write to SSD
Write (MB/s)
25000
20000
15000
10000
5000
0
1 4 7 10 13 16 19 22 25 28 31 34 37 40 43 46 49 52 55 58 61 64 67 70 73 76 79 82 85 88 91
Nodes (=MPI ranks over 46 2-core VMs)
none
krb5n
krb5a
krb5i
krb5p
krb5n + krb5p mdt

# Numbers - mdtest



# Thanks




- BMRC
  - Rob Esnouf
  - Adam Huffman
  - Colin Freeman
  - Geoffrey Ferrari
  - Charles Roberts
- DDN/WhamCloud
  - Vic Cornell
  - Sébastien Buisson