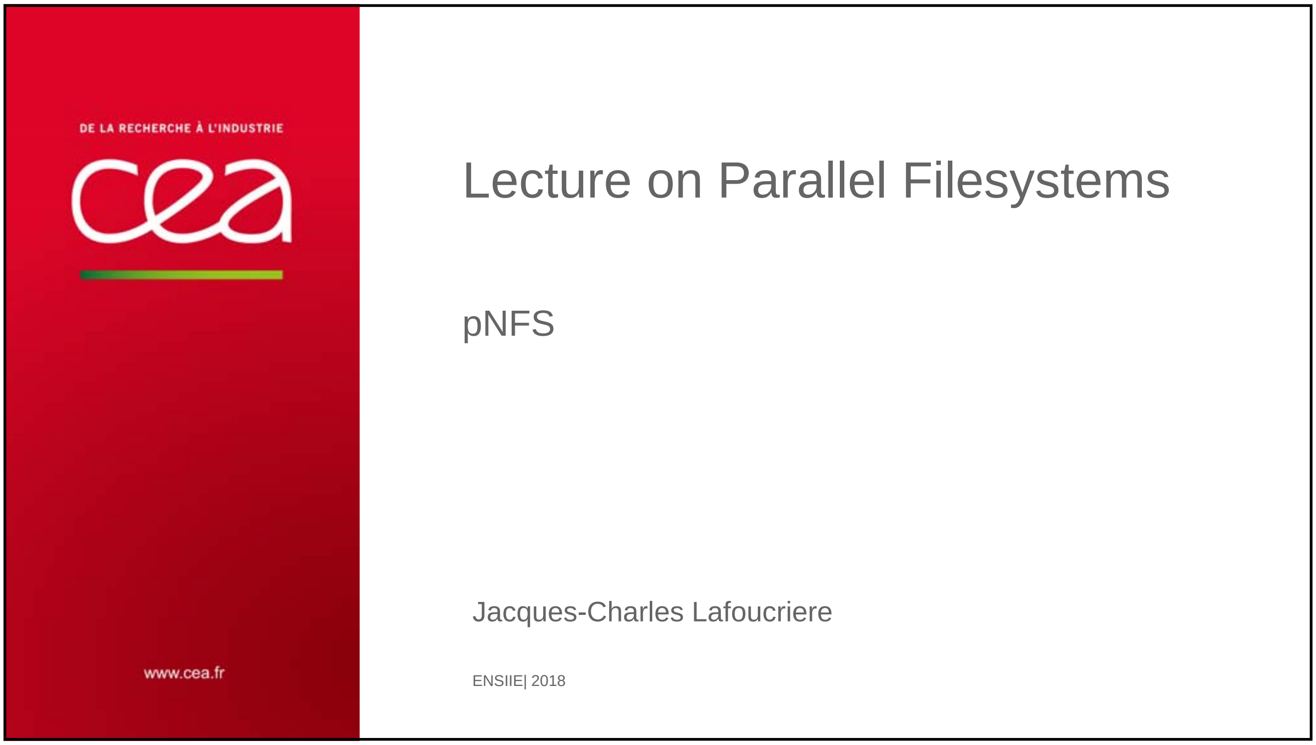


DE LA RECHERCHE À L'INDUSTRIE

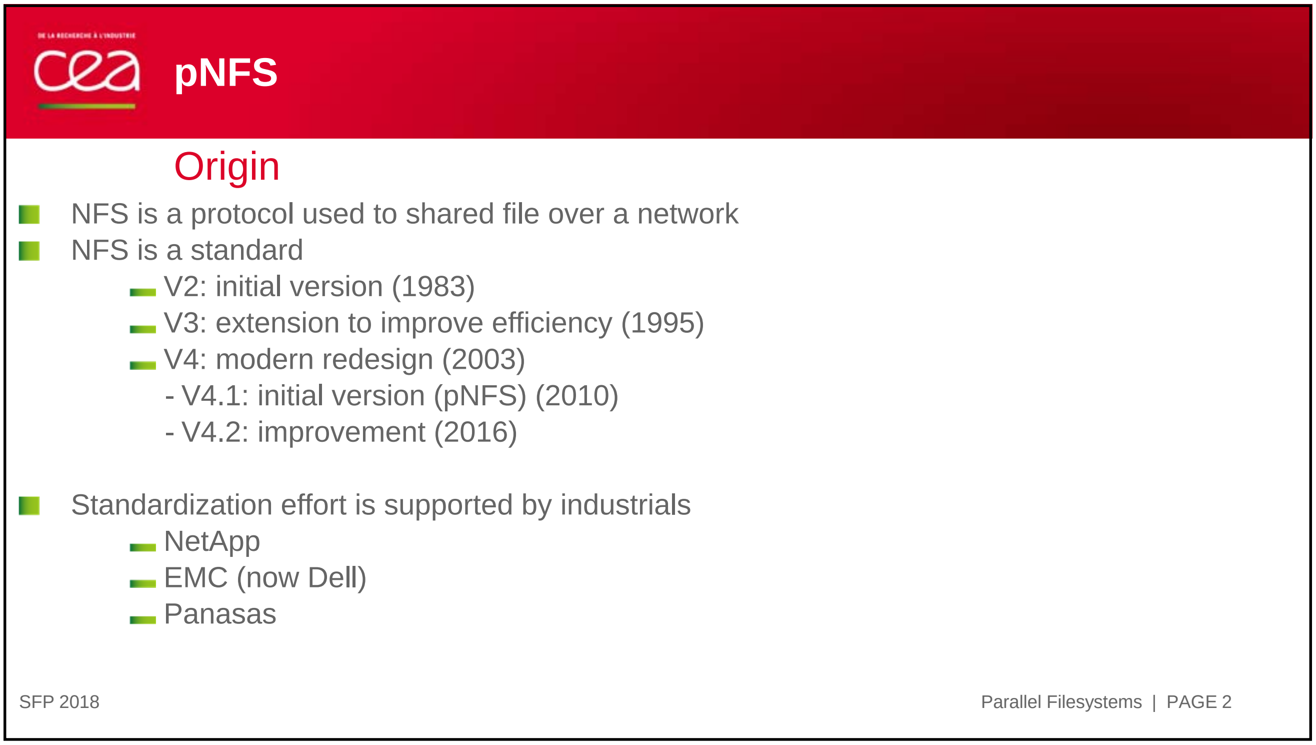
www.cea.fr

Lecture on Parallel Filesystems

pNFS

Jacques-Charles Lafoucriere

ENSIIE| 2018

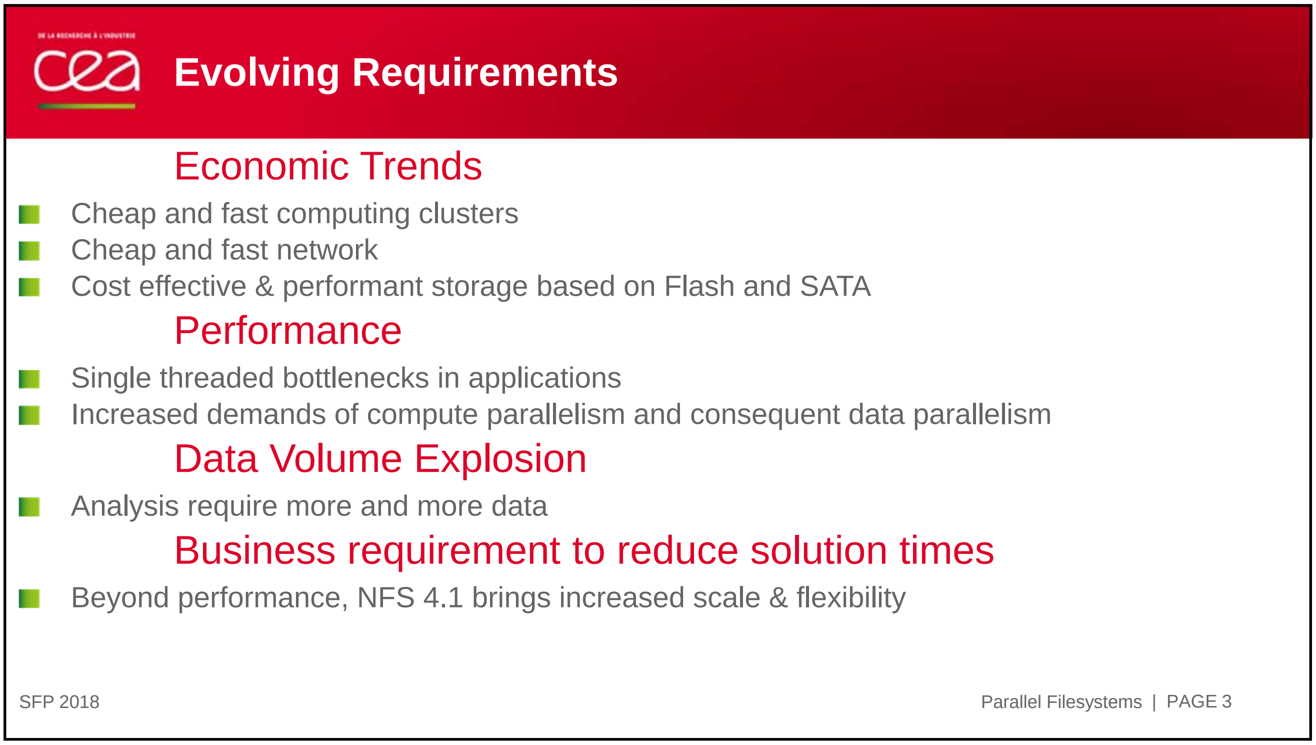


pNFS

Origin

- NFS is a protocol used to shared file over a network
- NFS is a standard
  - V2: initial version (1983)
  - V3: extension to improve efficiency (1995)
  - V4: modern redesign (2003)
    - V4.1: initial version (pNFS) (2010)
    - V4.2: improvement (2016)

- Standardization effort is supported by industrials
  - NetApp
  - EMC (now Dell)
  - Panasas



Evolving Requirements

Economic Trends

- Cheap and fast computing clusters
- Cheap and fast network
- Cost effective & performant storage based on Flash and SATA

Performance

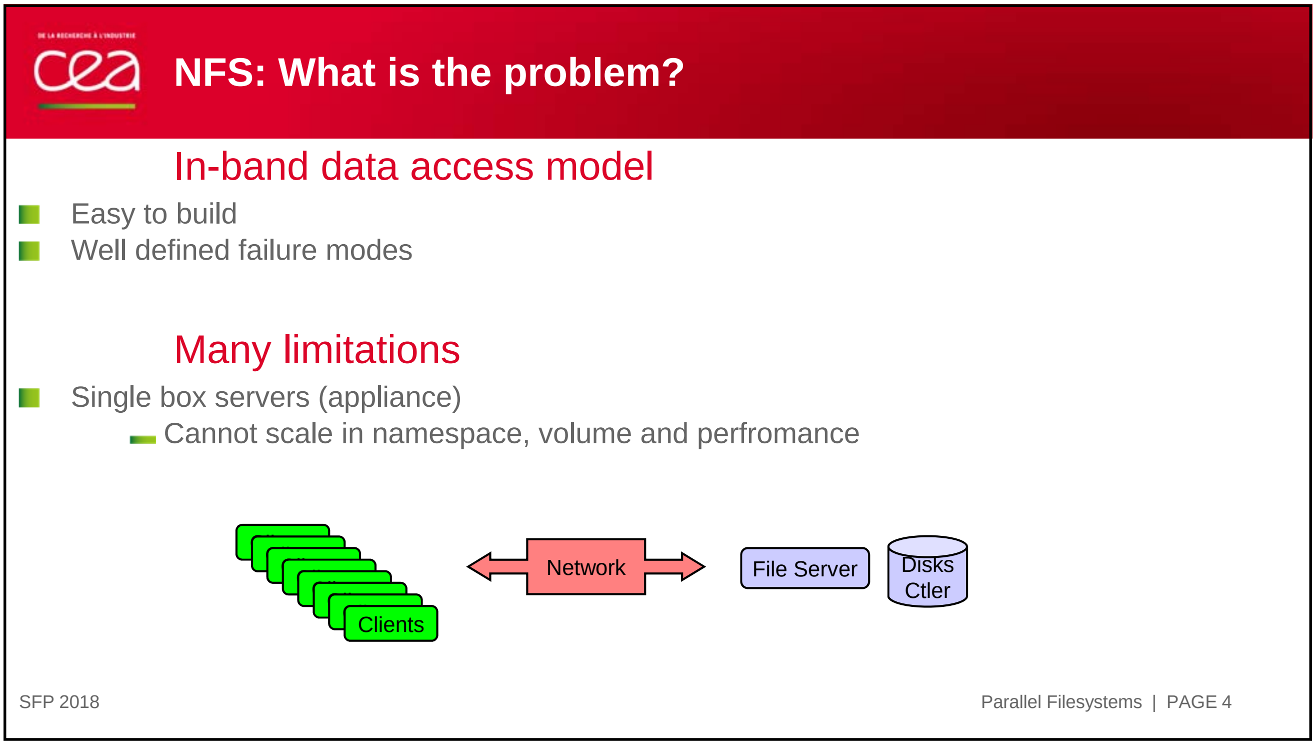
- Single threaded bottlenecks in applications
- Increased demands of compute parallelism and consequent data parallelism

Data Volume Explosion

- Analysis require more and more data

Business requirement to reduce solution times

- Beyond performance, NFS 4.1 brings increased scale & flexibility



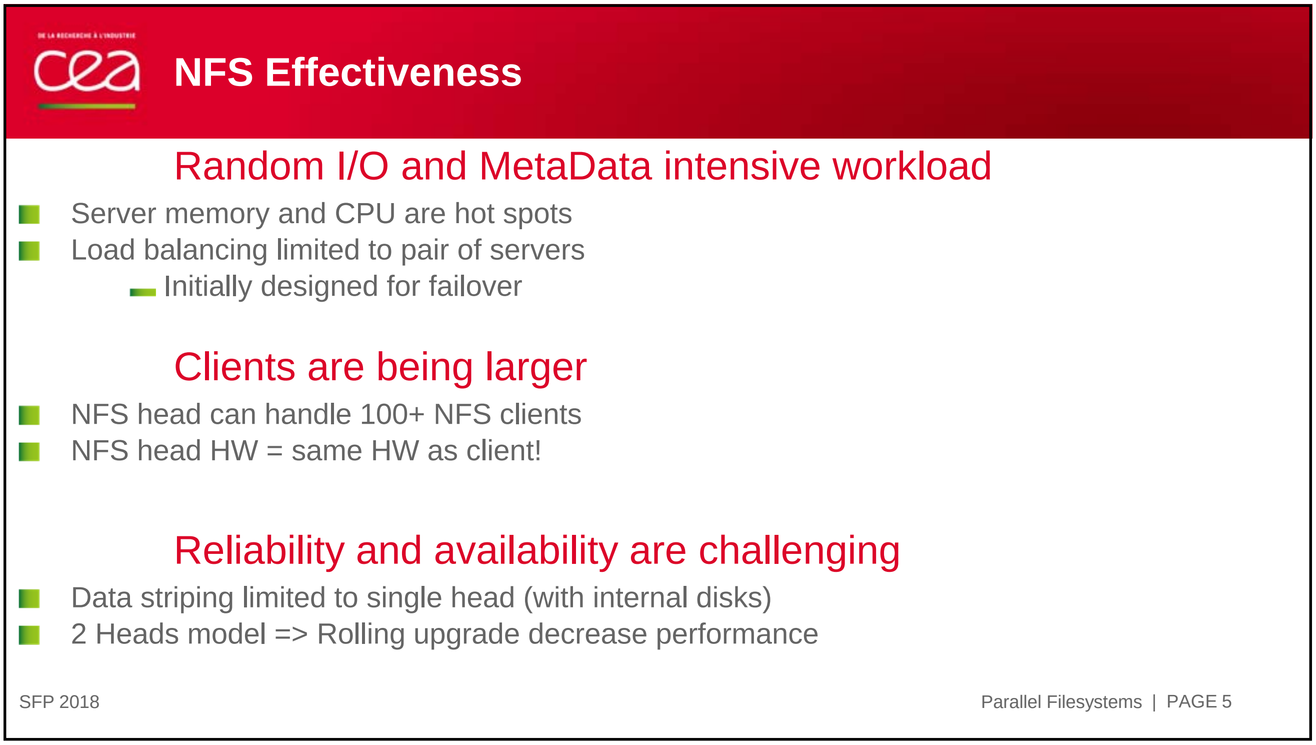
NFS: What is the problem?

In-band data access model

- Easy to build
- Well defined failure modes

Many limitations

- Single box servers (appliance)
  - Cannot scale in namespace, volume and perfromance



NFS Effectiveness

Random I/O and MetaData intensive workload

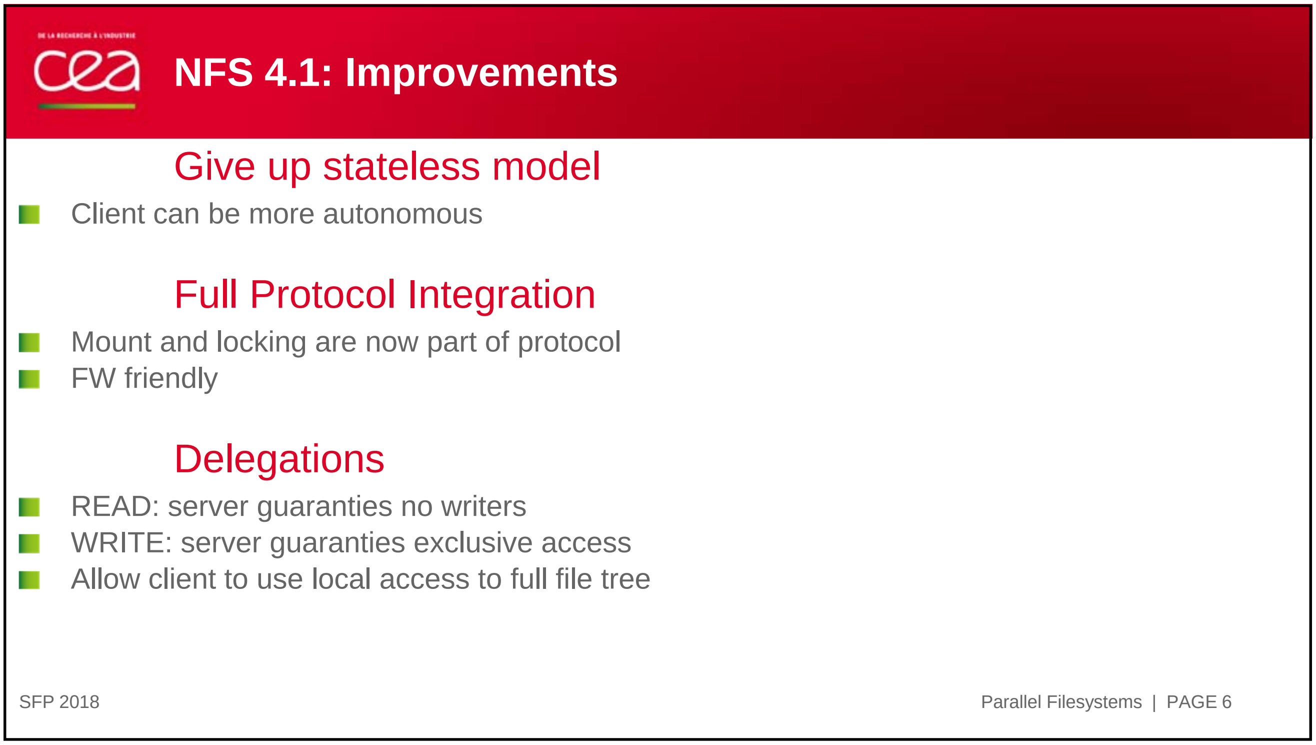
- Server memory and CPU are hot spots
- Load balancing limited to pair of servers
  - Initially designed for failover

Clients are being larger

- NFS head can handle 100+ NFS clients
- NFS head HW = same HW as client!

Reliability and availability are challenging

- Data striping limited to single head (with internal disks)
- 2 Heads model => Rolling upgrade decrease performance



NFS 4.1: Improvements

Give up stateless model

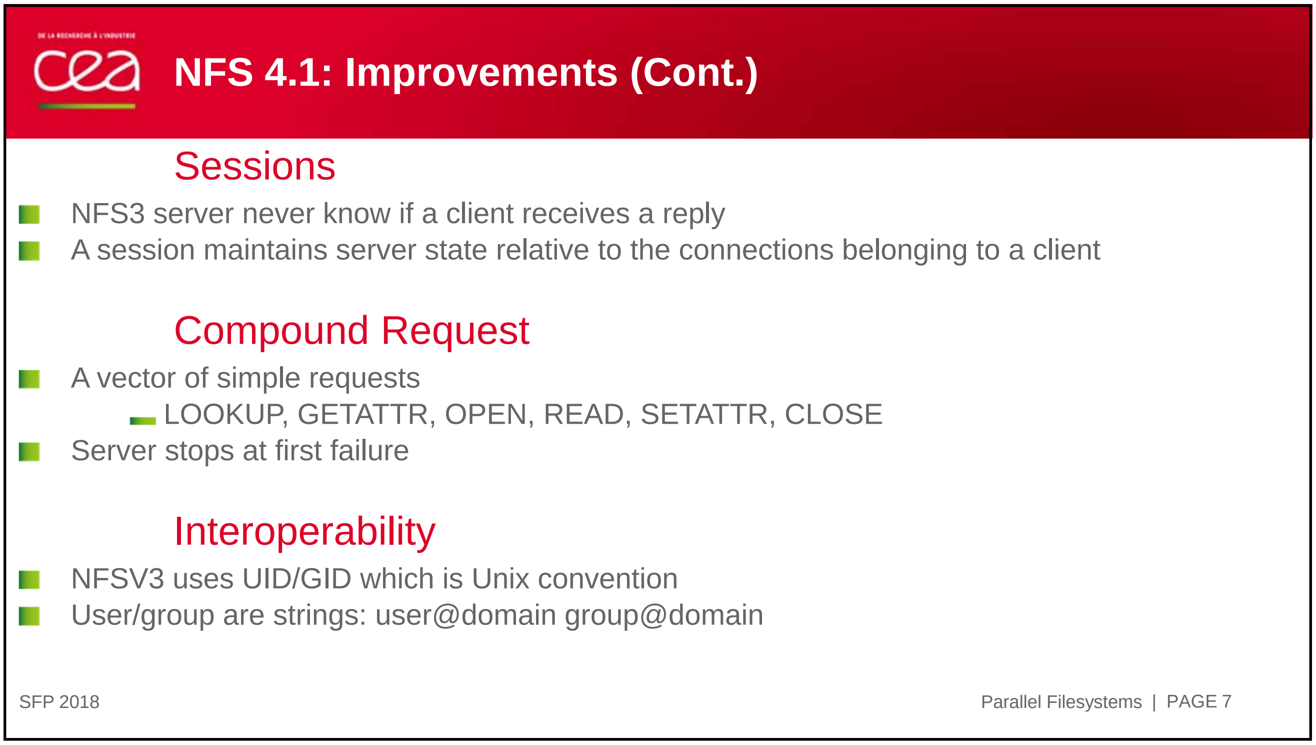
- Client can be more autonomous

Full Protocol Integration

- Mount and locking are now part of protocol
- FW friendly

Delegations

- READ: server guaranties no writers
- WRITE: server guaranties exclusive access
- Allow client to use local access to full file tree



NFS 4.1: Improvements (Cont.)

Sessions

- NFS3 server never know if a client receives a reply
- A session maintains server state relative to the connections belonging to a client

Compound Request

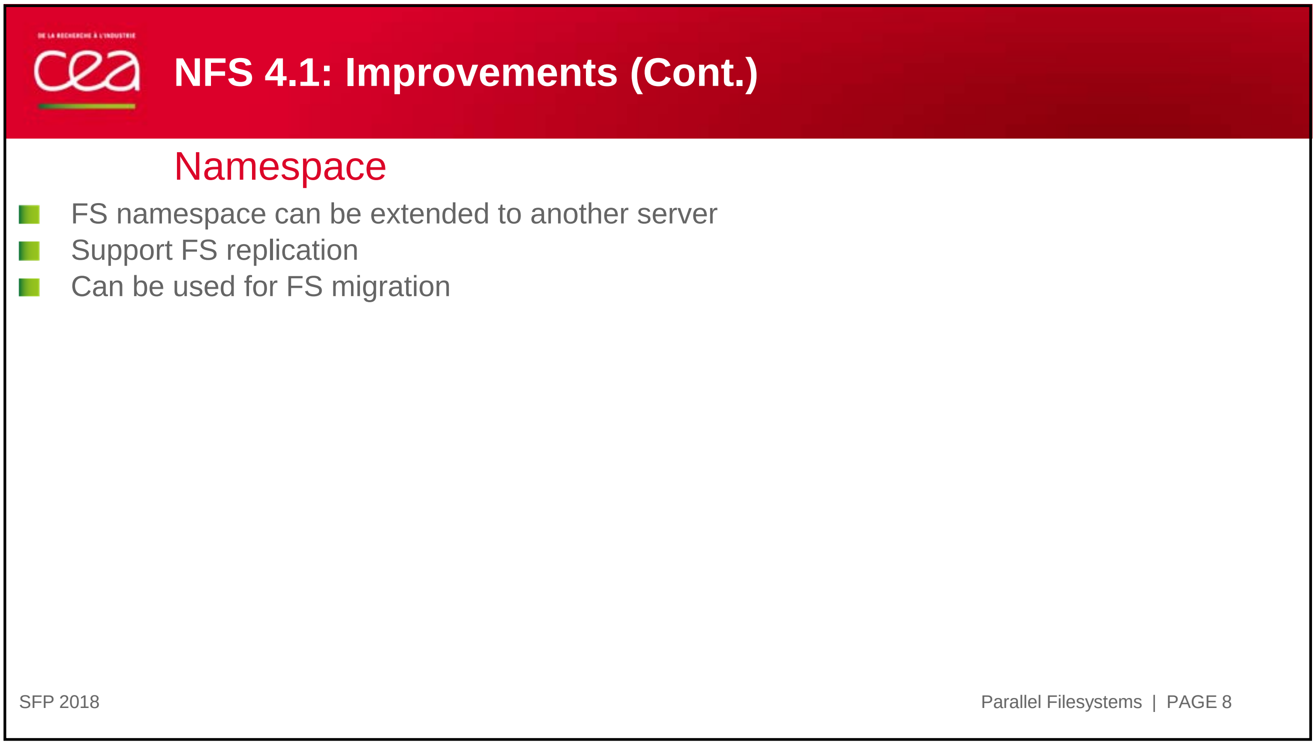
- A vector of simple requests
  - LOOKUP, GETATTR, OPEN, READ, SETATTR, CLOSE
- Server stops at first failure

Interoperability

- NFSV3 uses UID/GID which is Unix convention
- User/group are strings: user@domain group@domain

SFP 2018

Parallel Filesystems | PAGE 7



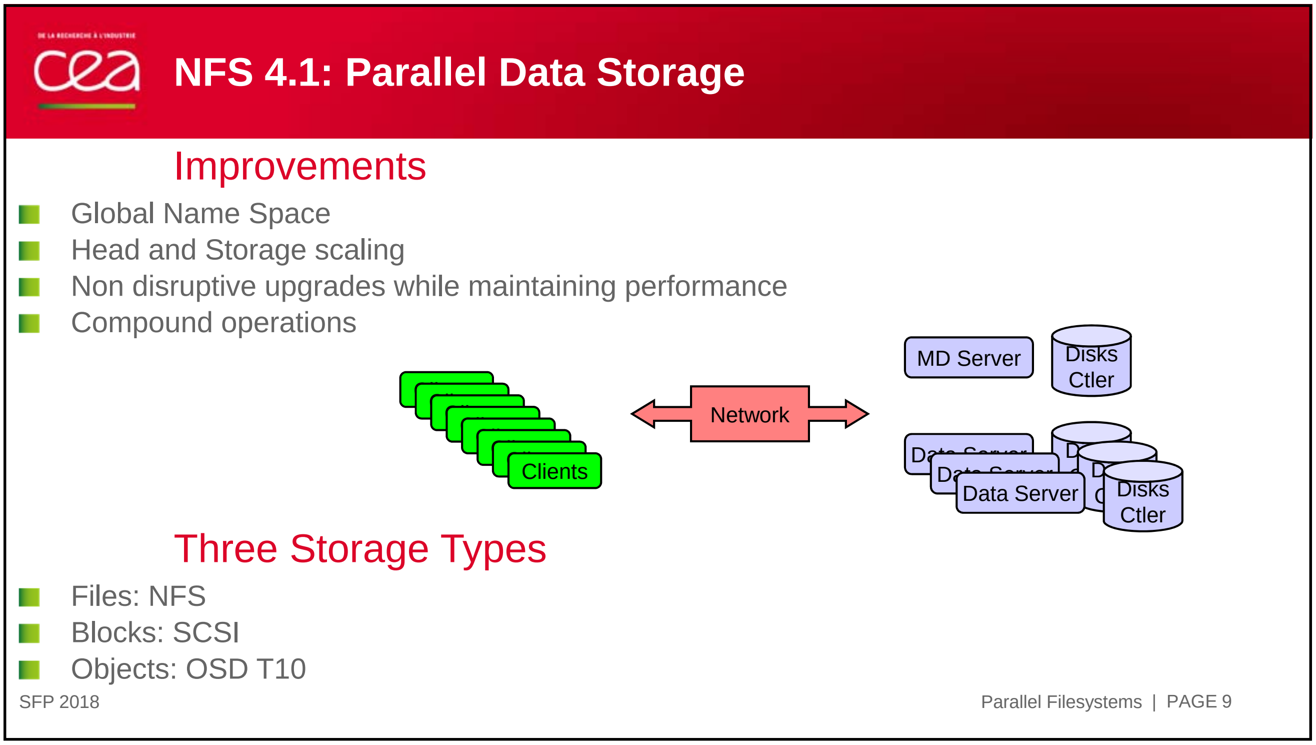
NFS 4.1: Improvements (Cont.)

Namespace

- FS namespace can be extended to another server
- Support FS replication
- Can be used for FS migration

SFP 2018

Parallel Filesystems | PAGE 8



NFS 4.1: Parallel Data Storage

Improvements

- Global Name Space
- Head and Storage scaling
- Non disruptive upgrades while maintaining performance
- Compound operations

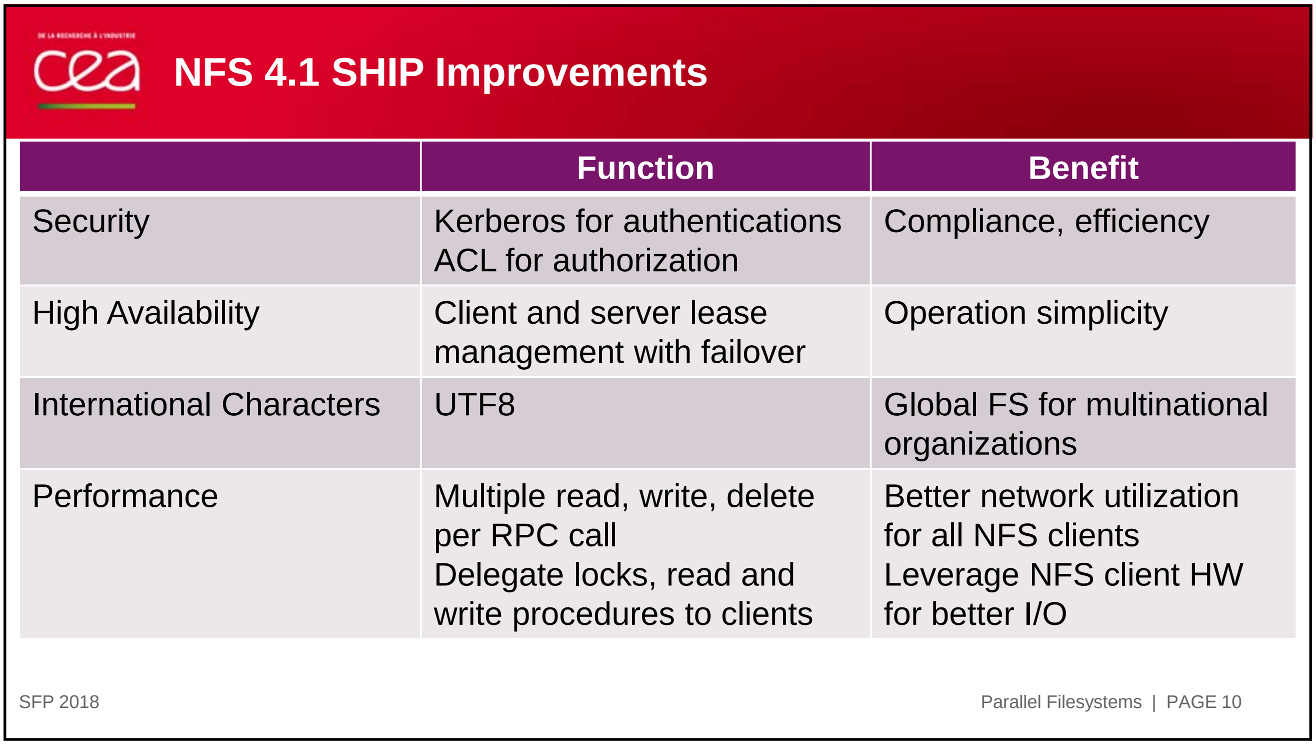
Clients
Network
MD Server
Disks Ctler
Data Server
Data Server
Data Server
Disks Ctler

Three Storage Types

- Files: NFS
- Blocks: SCSI
- Objects: OSD T10

SFP 2018

Parallel Filesystems | PAGE 9

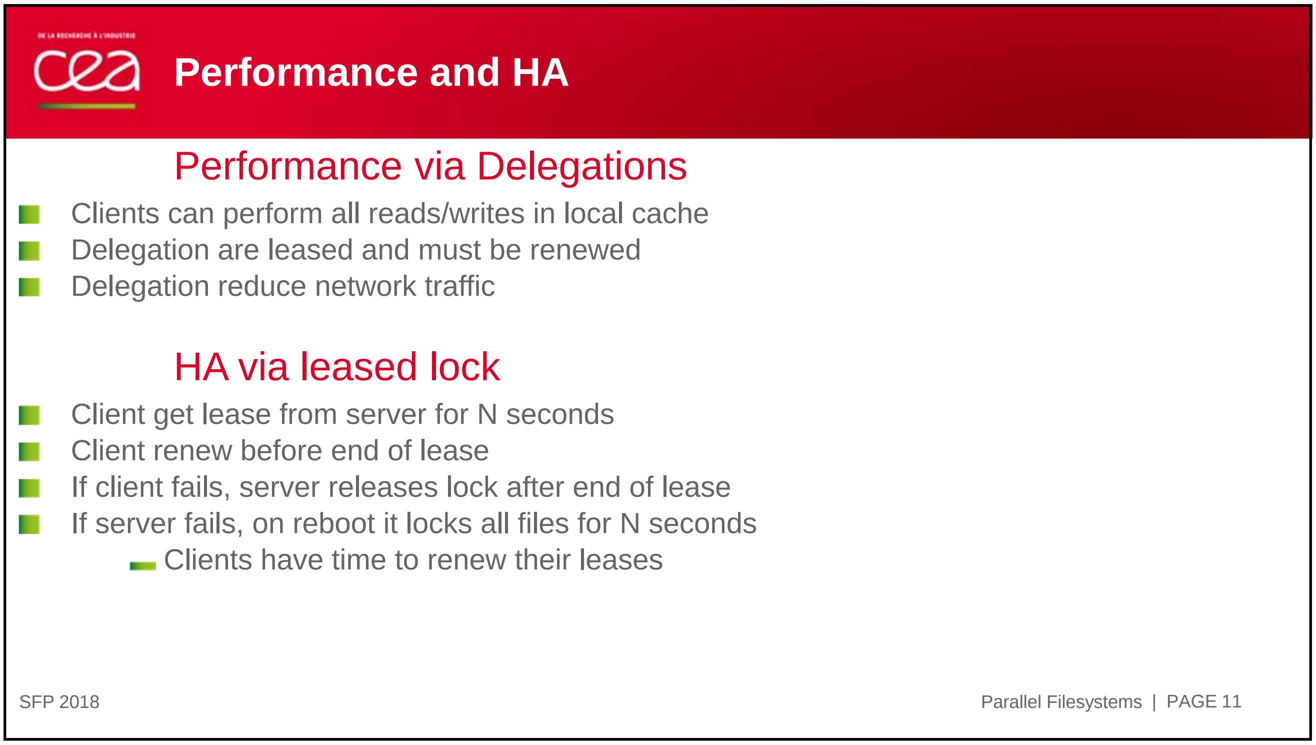


NFS 4.1 SHIP Improvements

	Function	Benefit
Security	Kerberos for authentications ACL for authorization	Compliance, efficiency
High Availability	Client and server lease management with failover	Operation simplicity
International Characters	UTF8	Global FS for multinational organizations
Performance	Multiple read, write, delete per RPC call Delegate locks, read and write procedures to clients	Better network utilization for all NFS clients Leverage NFS client HW for better I/O

SFP 2018

Parallel Filesystems | PAGE 10



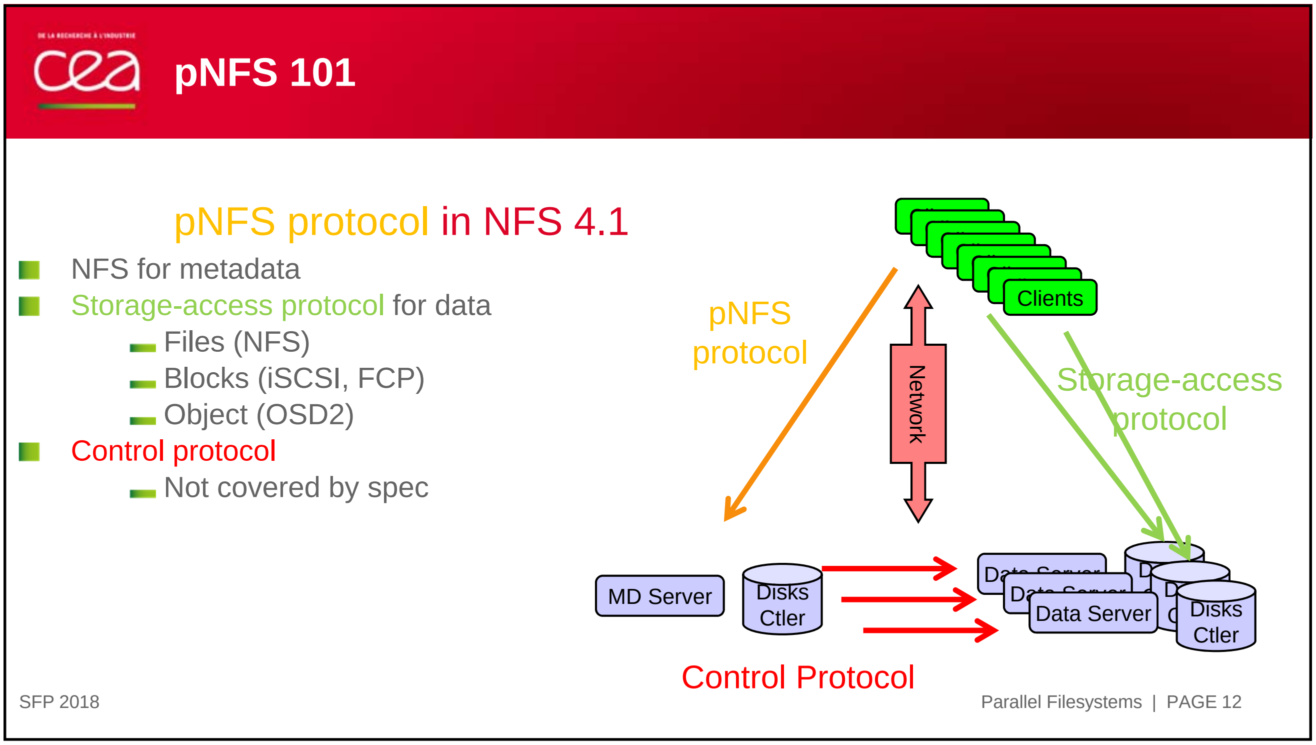
Performance and HA

Performance via Delegations

- Clients can perform all reads/writes in local cache
- Delegation are leased and must be renewed
- Delegation reduce network traffic

HA via leased lock

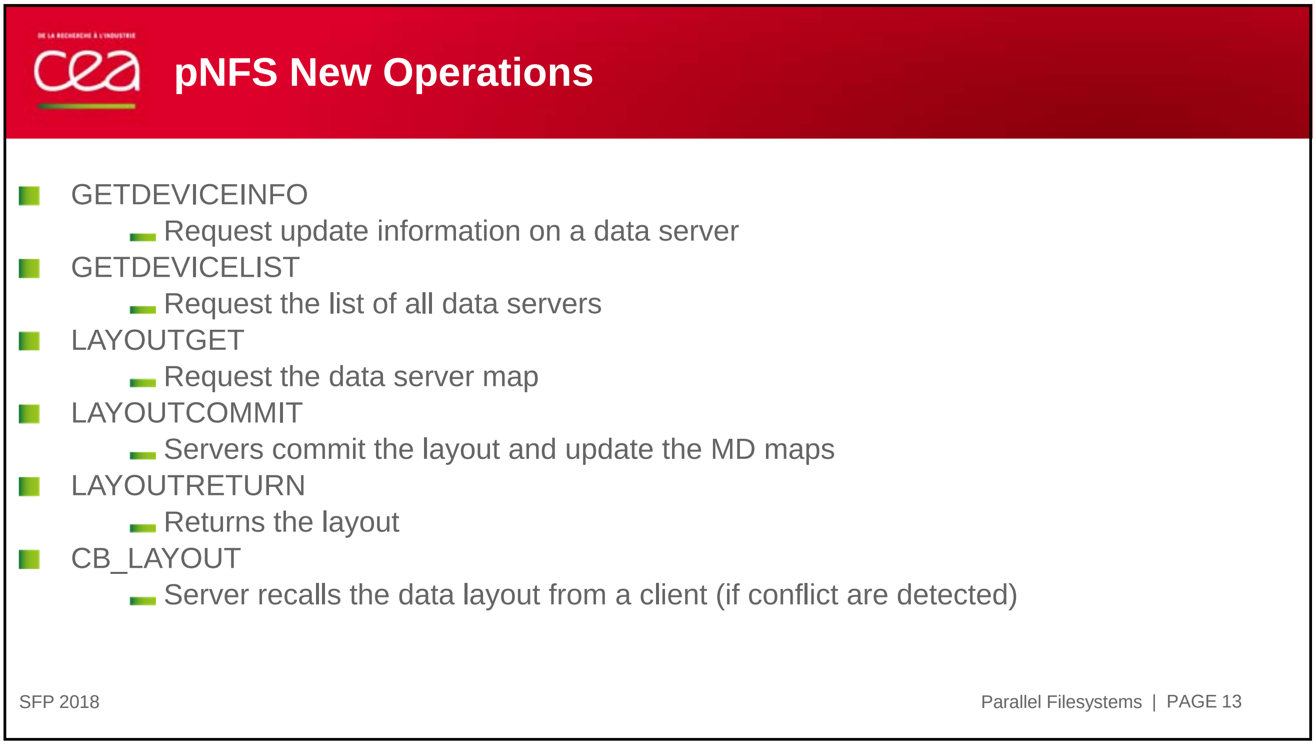
- Client get lease from server for N seconds
- Client renew before end of lease
- If client fails, server releases lock after end of lease
- If server fails, on reboot it locks all files for N seconds
  - Clients have time to renew their leases



pNFS 101

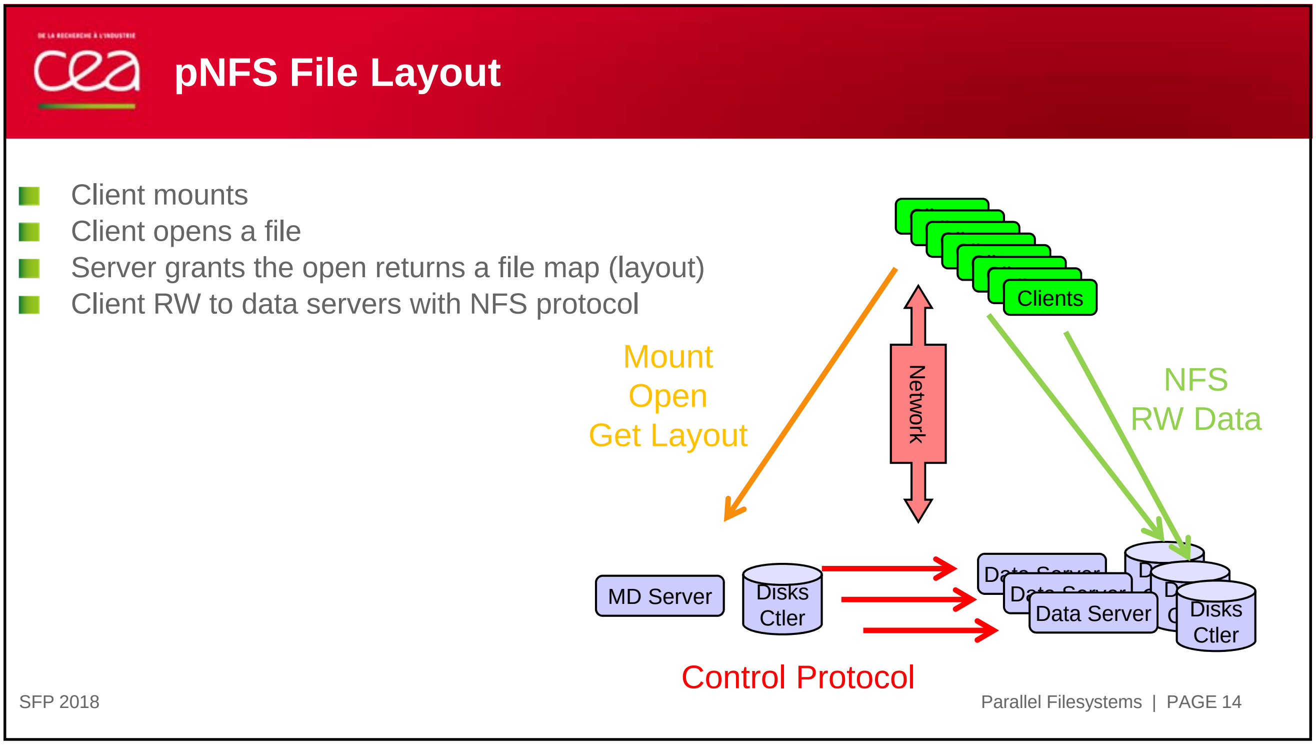
pNFS protocol in NFS 4.1

- NFS for metadata
- Storage-access protocol for data
  - Files (NFS)
  - Blocks (iSCSI, FCP)
  - Object (OSD2)
- Control protocol
  - Not covered by spec



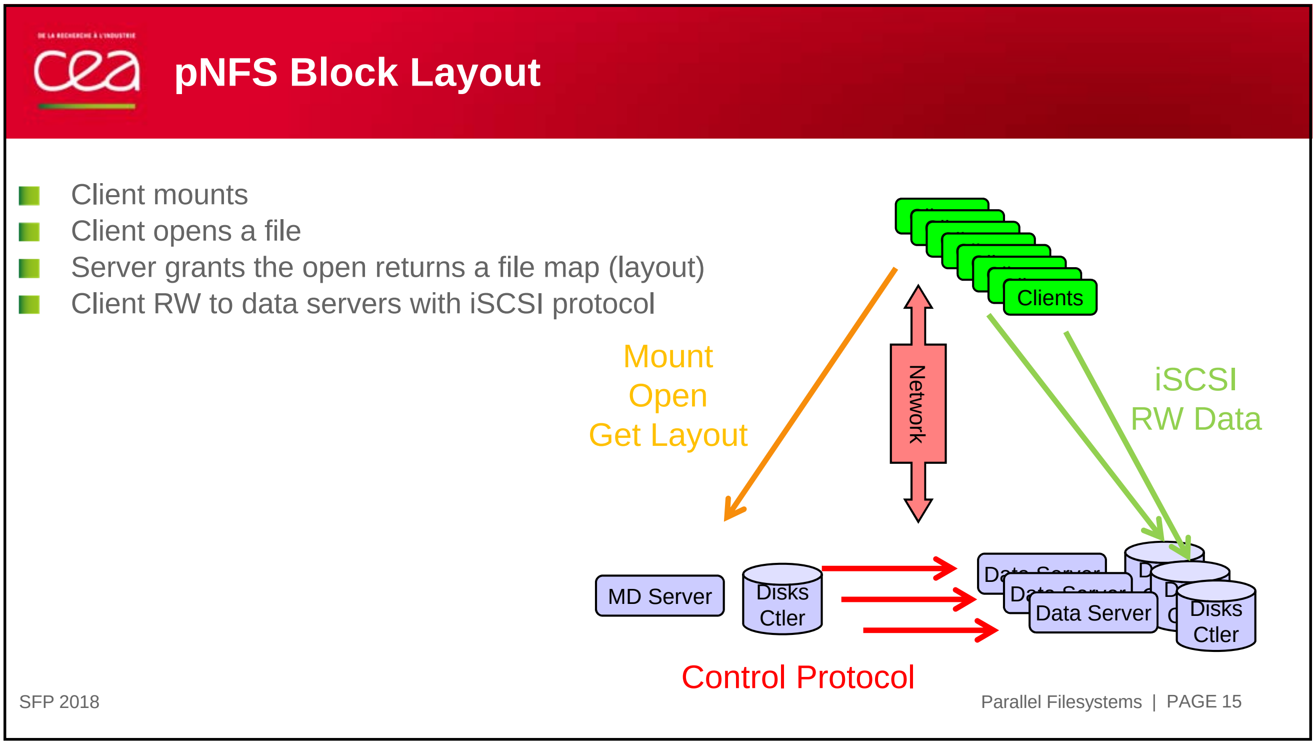
pNFS New Operations

- GETDEVICEINFO
  - Request update information on a data server
- GETDEVICELIST
  - Request the list of all data servers
- LAYOUTGET
  - Request the data server map
- LAYOUTCOMMIT
  - Servers commit the layout and update the MD maps
- LAYOUTRETURN
  - Returns the layout
- CB_LAYOUT
  - Server recalls the data layout from a client (if conflict are detected)



pNFS File Layout

- Client mounts
- Client opens a file
- Server grants the open returns a file map (layout)
- Client RW to data servers with NFS protocol

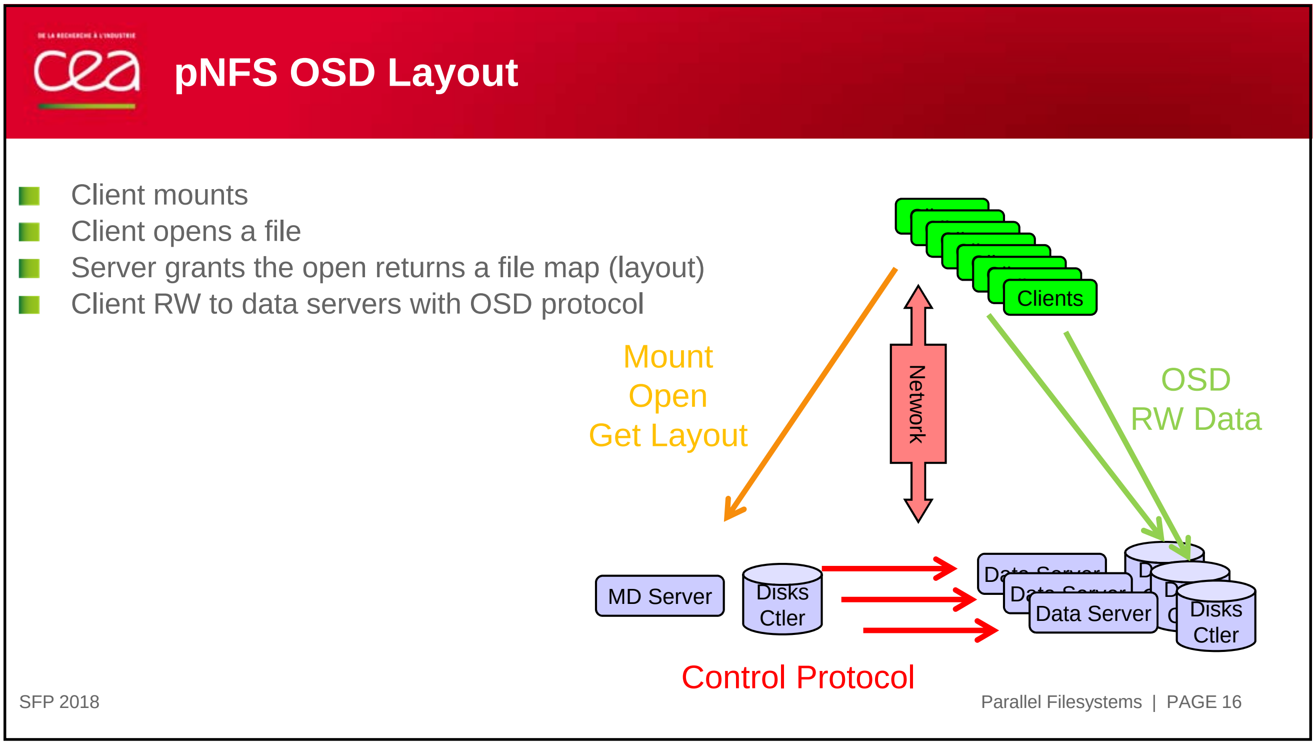


pNFS Block Layout

- Client mounts
- Client opens a file
- Server grants the open returns a file map (layout)
- Client RW to data servers with iSCSI protocol

Clients
Mount
Open
Get Layout
Network
iSCSI
RW Data
MD Server
Disks
Ctler
Data Server
Disks
Ctler
Control Protocol

SFP 2018 Parallel Filesystems | PAGE 15

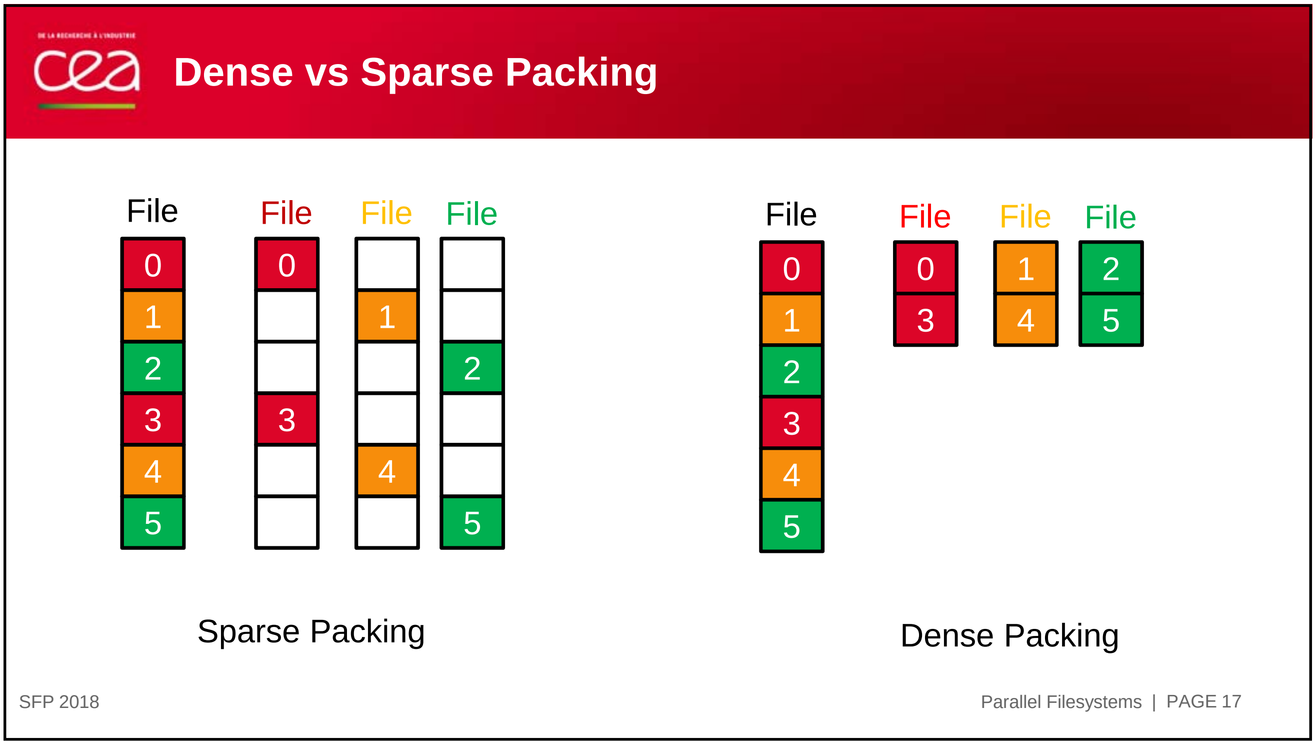


pNFS OSD Layout

- Client mounts
- Client opens a file
- Server grants the open returns a file map (layout)
- Client RW to data servers with OSD protocol

Clients
Mount
Open
Get Layout
Network
OSD
RW Data
MD Server
Disks
Ctler
Data Server
Disks
Ctler
Control Protocol

SFP 2018 Parallel Filesystems | PAGE 16



Dense vs Sparse Packing

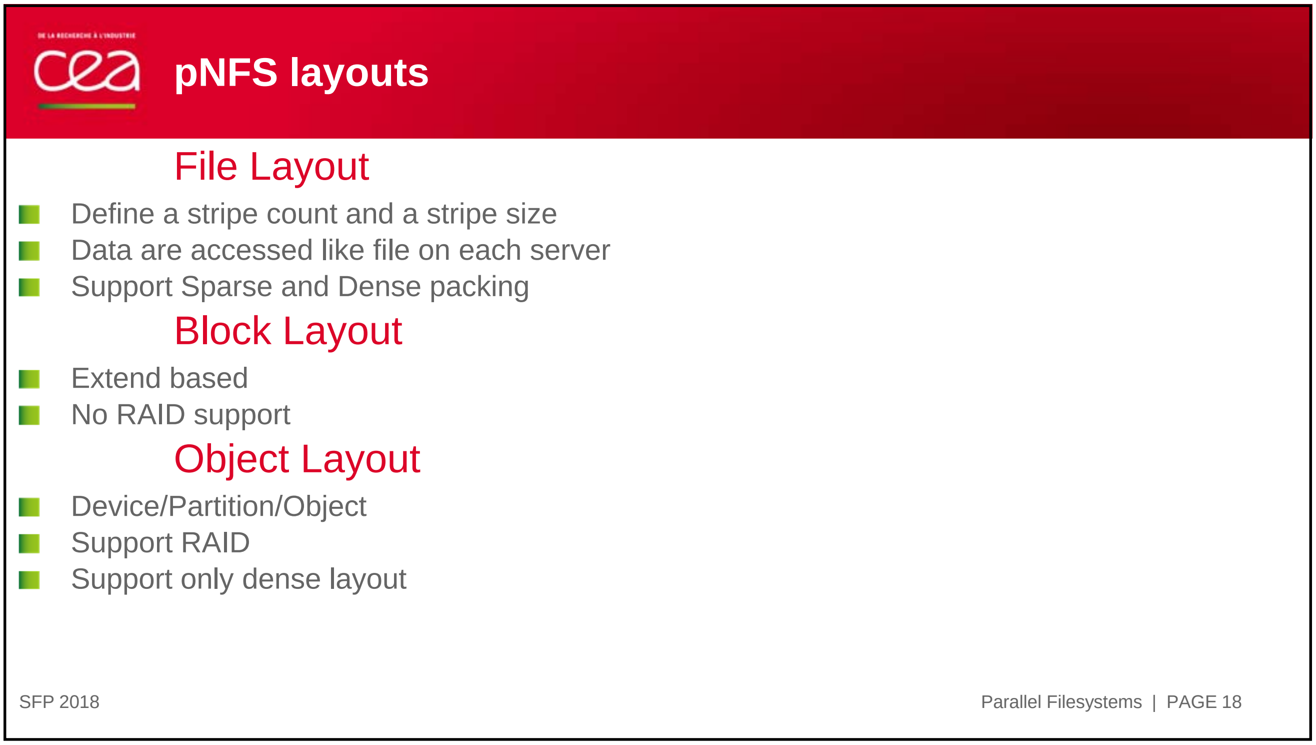
File	File	File	File
0	0		
1		1	
2			2
3	3		
4		4	
5			5

Sparse Packing

File	File	File	File
0	0	1	2
1	3	4	5
2			
3			
4			
5			

Dense Packing

SFP 2018 Parallel Filesystems | PAGE 17



pNFS layouts

File Layout

- Define a stripe count and a stripe size
- Data are accessed like file on each server
- Support Sparse and Dense packing

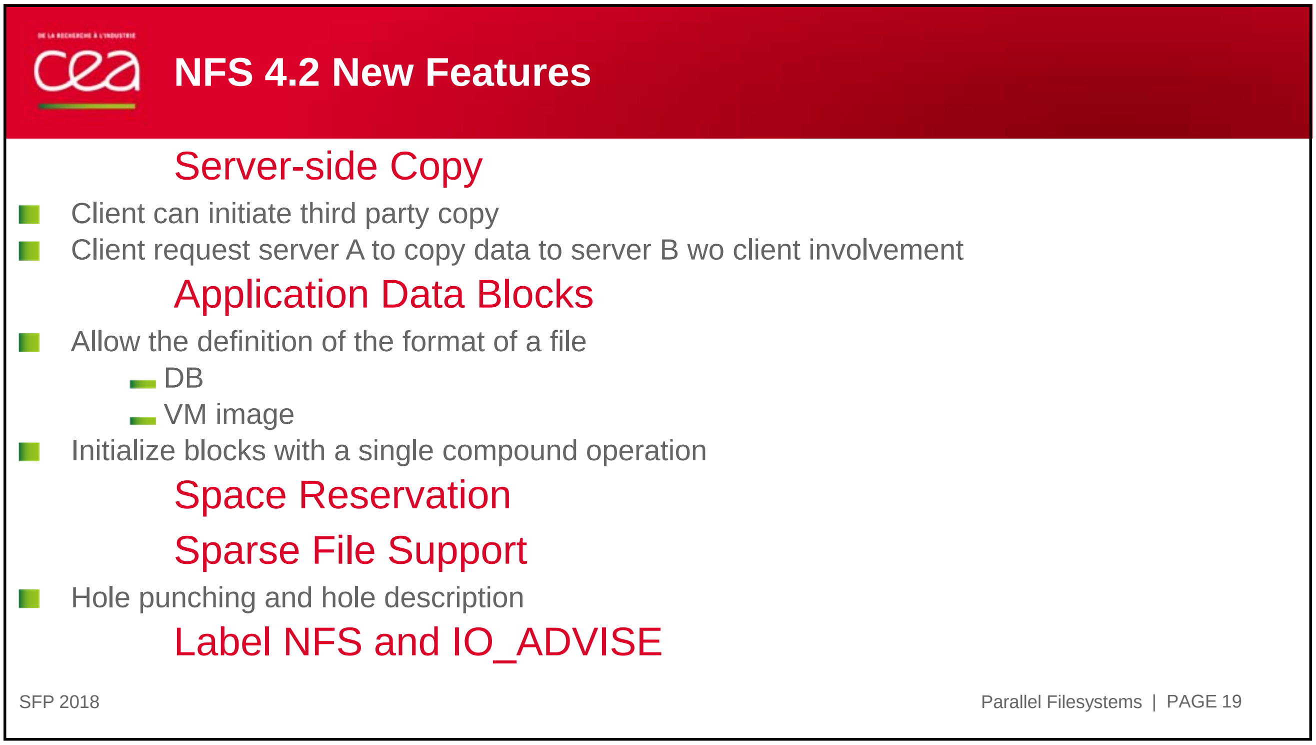
Block Layout

- Extend based
- No RAID support

Object Layout

- Device/Partition/Object
- Support RAID
- Support only dense layout

SFP 2018 Parallel Filesystems | PAGE 18



NFS 4.2 New Features

Server-side Copy

- Client can initiate third party copy
- Client request server A to copy data to server B wo client involvement

Application Data Blocks

- Allow the definition of the format of a file
  - DB
  - VM image
- Initialize blocks with a single compound operation

Space Reservation

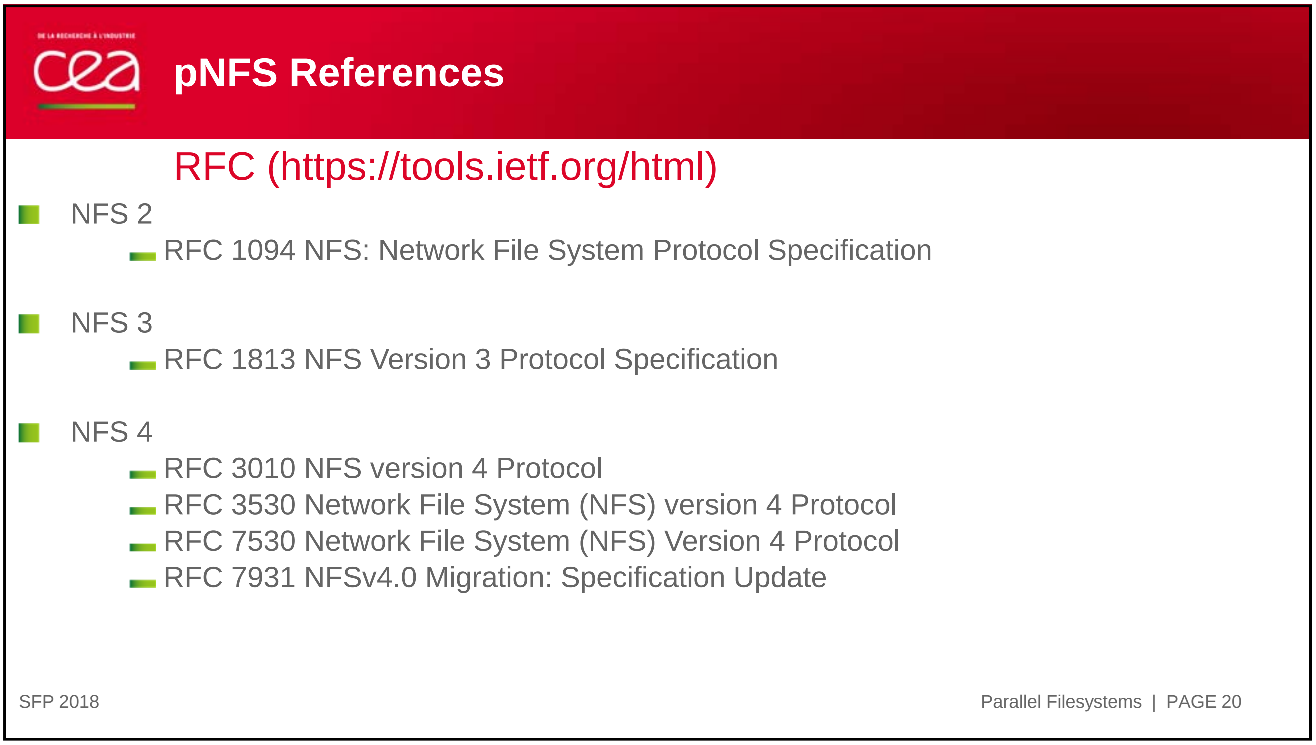
Sparse File Support

- Hole punching and hole description

Label NFS and IO_ADVISE

SFP 2018

Parallel Filesystems | PAGE 19



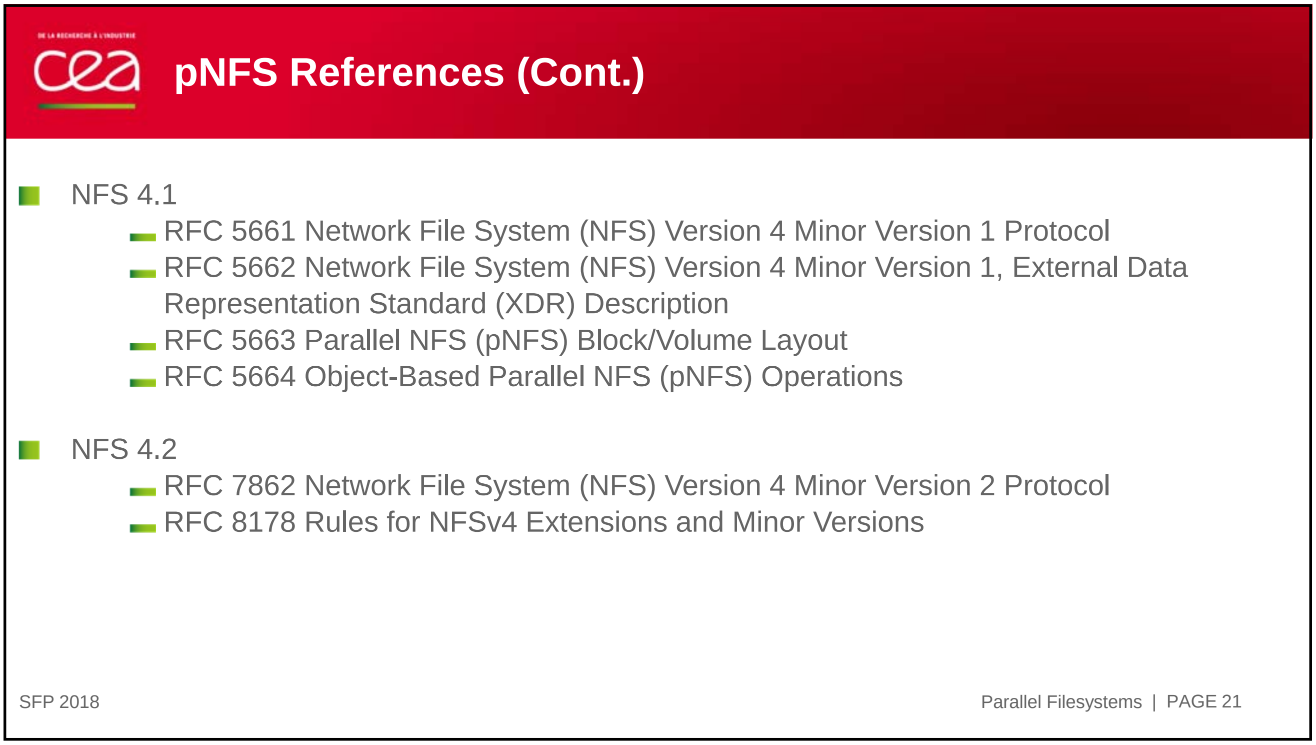
pNFS References

RFC (https://tools.ietf.org/html)

- NFS 2
  - RFC 1094 NFS: Network File System Protocol Specification
- NFS 3
  - RFC 1813 NFS Version 3 Protocol Specification
- NFS 4
  - RFC 3010 NFS version 4 Protocol
  - RFC 3530 Network File System (NFS) version 4 Protocol
  - RFC 7530 Network File System (NFS) Version 4 Protocol
  - RFC 7931 NFSv4.0 Migration: Specification Update

SFP 2018

Parallel Filesystems | PAGE 20

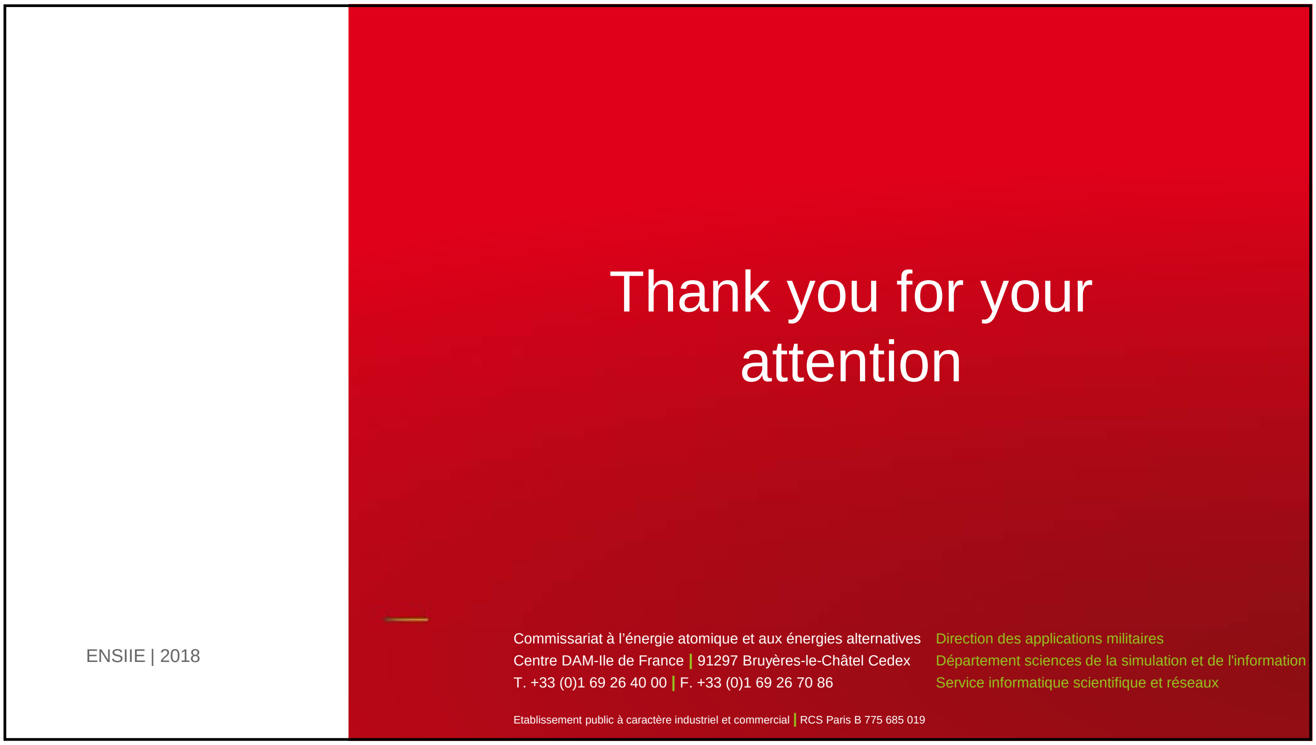


pNFS References (Cont.)

- NFS 4.1
  - RFC 5661 Network File System (NFS) Version 4 Minor Version 1 Protocol
  - RFC 5662 Network File System (NFS) Version 4 Minor Version 1, External Data Representation Standard (XDR) Description
  - RFC 5663 Parallel NFS (pNFS) Block/Volume Layout
  - RFC 5664 Object-Based Parallel NFS (pNFS) Operations
- NFS 4.2
  - RFC 7862 Network File System (NFS) Version 4 Minor Version 2 Protocol
  - RFC 8178 Rules for NFSv4 Extensions and Minor Versions

SFP 2018

Parallel Filesystems | PAGE 21



Thank you for your attention

ENSIIE | 2018

Commissariat à l'énergie atomique et aux énergies alternatives
Centre DAM-Ile de France | 91297 Bruyères-le-Châtel Cedex
T. +33 (0)1 69 26 40 00 | F. +33 (0)1 69 26 70 86

Direction des applications militaires
Département sciences de la simulation et de l'information
Service informatique scientifique et réseaux

Etablissement public à caractère industriel et commercial | RCS Paris B 775 685 019