DE LA RECHERCHE À L'INDUSTRIE

Implement pNFS support for Lustre via stateless IO agents

Philippe DENIEL (philippe.deniel@cea.fr)

www.cea.fr

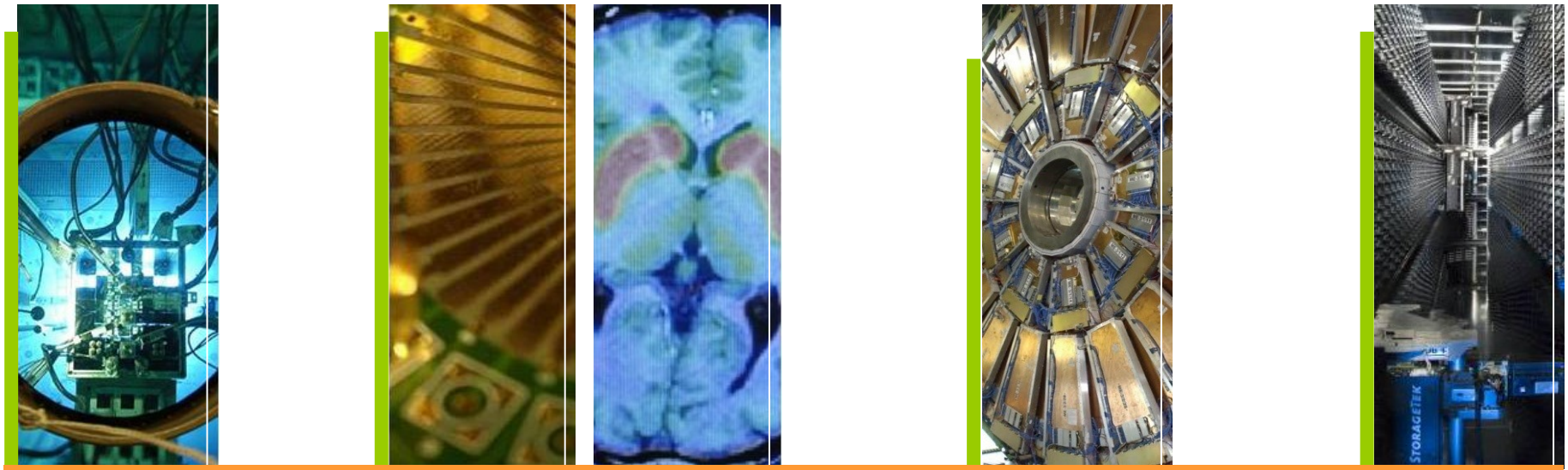
CEA in a few words

Low Carbon Energy

Technologies for Information and health

Very Large Research Infrastructures

Defense and deterrence



Fundamental Research

Teaching and dissemination of knowledge

Valuation and technological dissemination

10 CENTERS IN FRANCE

Materials Sciences, Software Technologies,
High Performance Computing, Biomedical
Ile-de-France

Fontenay-aux-Roses

Saclay

Bruyères-le-Châtel

Valduc

Le Ripault

Materials
Centre, Bourgogne

Micro-Nanotechnologies
Nanobiotechnologies
New Technologies
Rhône-Alpes

Grenoble

Nuclear: nuclear fuel life cycle
and waste management
Vallée du Rhône

Cesta

Gramat

Marcoule

Cadarache

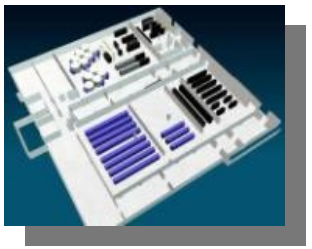
Lasers and plasmas
Aquitaine

Nuclear: Fusion, fission
Provence Alpes Côte d'Azur

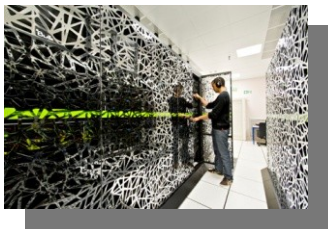
Vulnerability Assessment
Detonics
Midi-Pyrénées

CEA : « From research to industry »

- TERA = defense



- CCRT = CEA + industry partners
  + France Génomique



SAFRAN Snecma
SAFRAN Techspace Aero
SAFRAN Turbomeca
EDF
AREVA
INERIS
ASTRIUM
Valeo
L'ORÉAL
THALES

FRANCE GÉNOMIQUE

- CURIE @ TGCC
  France/Europe HPC

- TER@TEC Campus: hosts industrials, software company, labs (Intel, Bull, DISTENE, ESI, SILKAN…)
  Contribution to a French and European industrial ecosystem

The Big Picture : Lustre and pNFS

About pNFS

- Well... I think that every attendee knows what I am talking about
- So I'll be fast on this point :-)
- pNFS is quite interesting
  - An industrial standard, natively supported in the kernel
- NFS-Ganesha started at CEA (ten years ago) and have pNFS support (for OSD2 layout and Files layout)

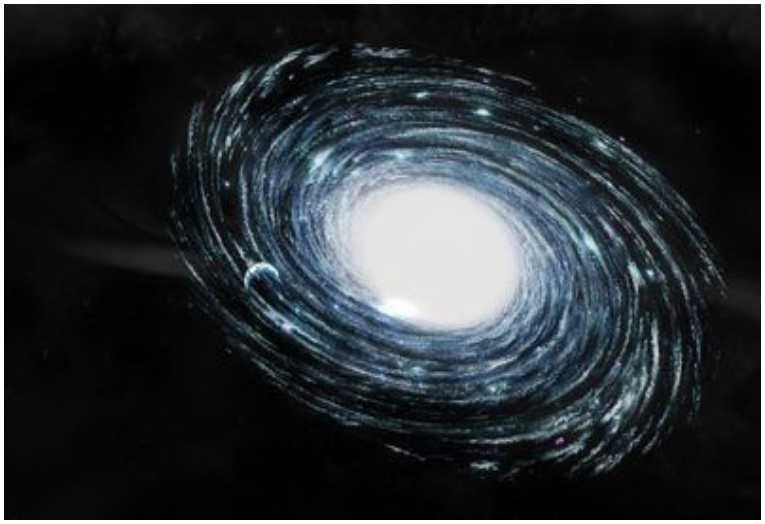
Lustre, a parallel filesystem with focus on HPC

- Machines involved in High Performance produce data massively
  - Real life experice : up to 1.5PB produced in a single weekend
- Currently, two products leads this market
  - GPFS from IBM
  - Lustre, an OpenSource Product, now managed by Intel
- CEA/DAM has been part of the Lustre community for a long time
  - We are active Lustre developers
  - We have large Lustre configurations in production (up to 10PB)

For CEA, using pNFS and Lustre together is natural

Lustre, the galactic filesystem

Scalable clustered filesystem



- Powers the world's most powerful supercomputers

- Tens of thousand of clients
- Hundreds of petabytes of storage
- TeraBytes per second of I/O throughput

- Fully software solution
- Kernel-land (Linux)
- Distributed under the terms of GNU GPLv2

- Actively developed (~100 contributors per major release)
- Drives an entire ecosystem (robinhood policy engine, hadoop adapter...)

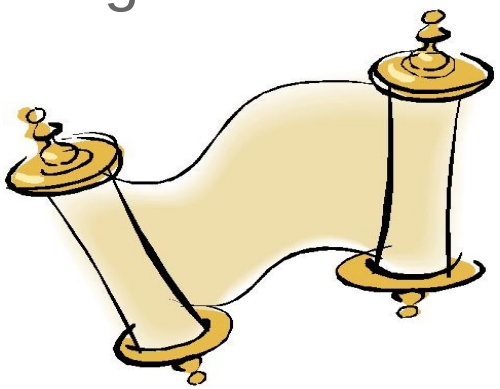
lustre®

Lustre: Project history

- Started 1999, P. Braam at Carnegie Mellon University
- Founded Cluster Filesystem (CFS) company in 2001

- Acquired by Sun Microsystems in 2007
- Acquired by Oracle in 2010, which dropped it less than a year later
- Creation of whamcloud
- Acquired by Intel in 2012
- Xyratex Ltd. bought the IP in Feb. 2013 and gave it back to the community

- The core developers mostly remained the same
- The community organized itself to cope with these changes (OpenSFS, EOFS)



Using pNFS with Lustre : not a new idea

Other people tried to marry Lustre and pNFS

- Design, Implementation, and Evaluation of Transparent pNFS on Lustre (2009)
  - Weikuan Yu, Jeffrey S. Vetter (From ORNL), Oleg Droki (Sun Microsystems)
  - Published in *International Symposium on Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE*
- Parallel NFS (pNFS) Lustre Layout Operations , IETF draft (last version in May 2014)
  - Presented (by Sorin Faibish) at connectathon in 2013
  - Autored by S. Faibish, D. Cote and P. Tao

The basic idea : make it possible to use OSS as pNFS DS

- pNFS as Metadata servers and Data Servers
- Lustre uses MDS (Metadata Servers) and OSS (Object Storage Servers)
- Two approaches
  - Use NFSv4 Files Layout : share pages in-between OSS and pNFS DS
  - Develop a shim layer to allow clients to talk to OSS directly

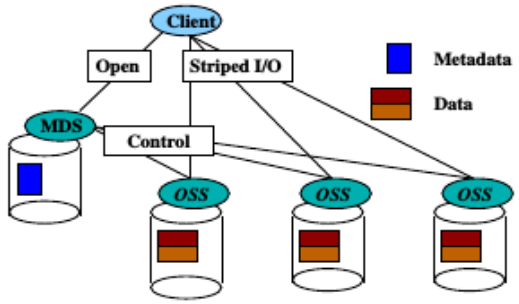


Figure 1. Lustre Architecture

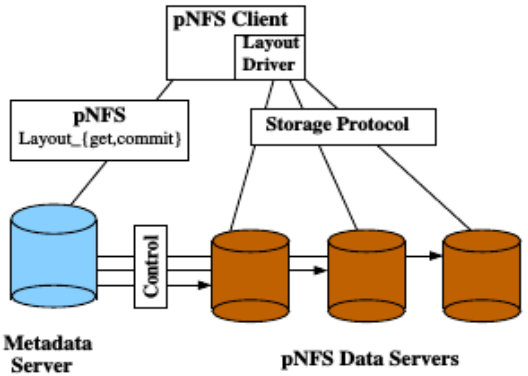


Figure 2. pNFS Architecture

Why those low level approaches are difficult

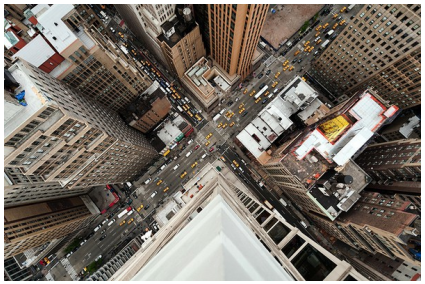
Lustre is greedy

- Lustre consumes lots of ressources
  - Not only disks, memory and CPU are stressed too
- Lustre does not like to lend his toys to the other kids
  - Things quickly go wrong as someone else requires memory on a OSS
  - Lustre produces a strong memory pressure
- Result in production situation may be unstable

Lustre relies on non standardized stuff

- Communicating with Lustre OSS means speaking Lustre Native language
- Unlike other parallel filesystems, Lustre uses opensourced but non-standard models
  - The Object Storage follows a specific protocol
  - The Network layer is RDMA based, but acquainted with very low software level in a non-stantard way
- If a pNFS client can do that, it has 95% of what you need to turn it into a Lustre Client
  - Why use pNFS and pay an protocol overhead that will slow performances ?

Lustre IO Providers



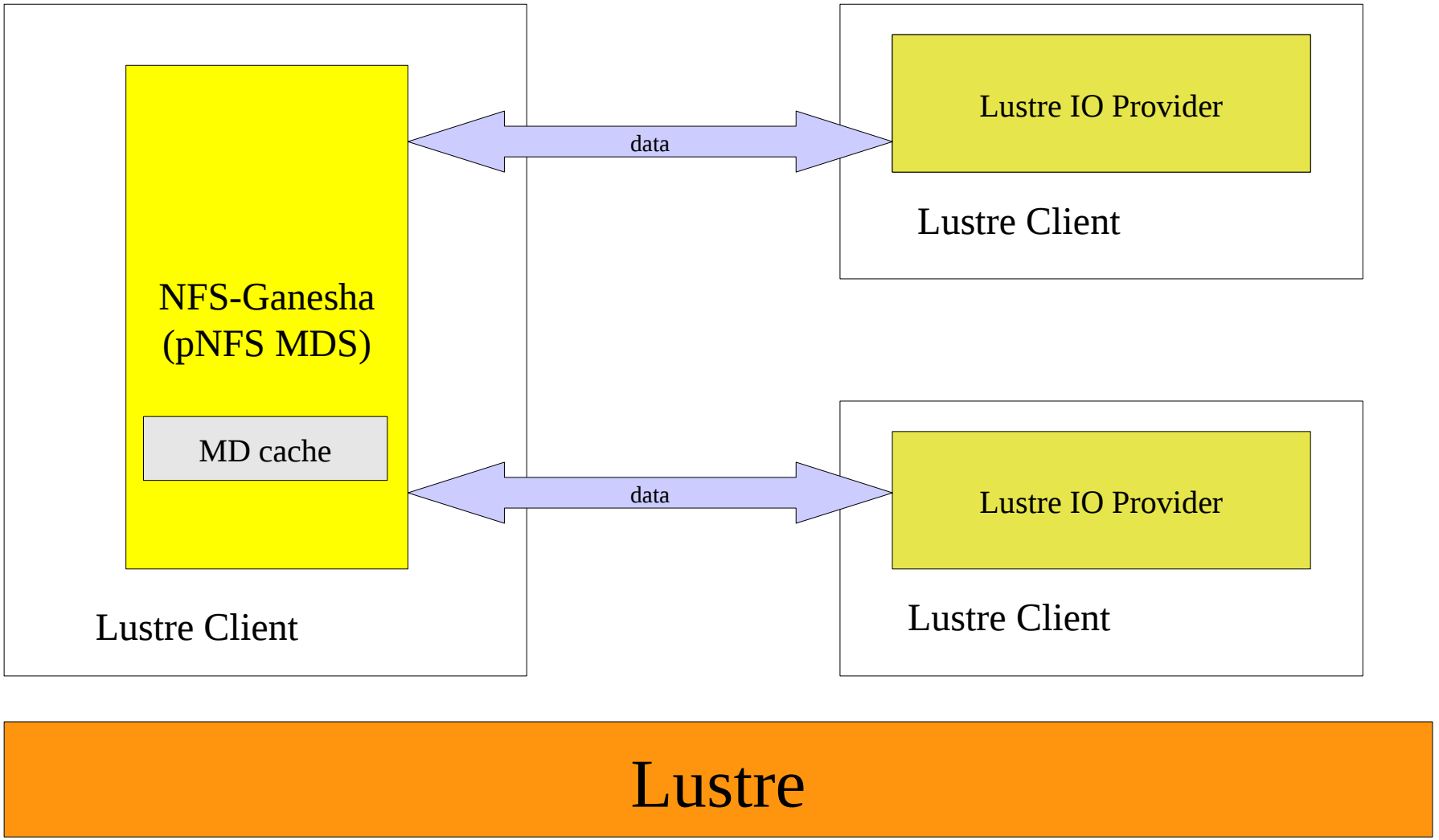
Attack the Lustre Stack from the top

- Avoid what's hard to do:
  - Dealing directly with Lustre consistency : let Lustre do the job
  - Avoid memory congestion : avoid memory overhead to MDS and OSS
  - Lustre protocols are tricky and non-standard : try avoid using them

Introducing Lustre IO Providers

- Still a Work In Progress
- A very simple daemon, accessed by a very simple, RDMA based, protocol
- NFS-Ganesha will be used a the pNFS/MDS for this pNFS access, but LIOP will be a completely separated project, available as Open Source (under the terms of the LGPLv3)
- LIOP are stateless and lightweight, their memory fingerprint is as small as possible
- By relying on Lustre Clients
  - LIOP stay consistent with Lustre
  - LIOP doesn't have to care about all the complicated Lustre machinery
  - Accesses are made via POSIX and Lustre FID-based interface

Lustre IO Provider : a quick view



Lustre IO Providers : the protocol

A really simple protocol

- RDMA based
  - No marshalling to allow zero-copy and memory sharing
  - But TCP/IP will be usable with lesser performances and bigger fingerprints
  - Protocol definition will use the same buffer oriented philosophy as 9p
- A protocol with 3 functions (FID is Lustre File ID)
  - WRITE: write (FID, Offset, length) = DATA
  - READ: DATA = read(FID,Offset,length)
  - STATUS: gets information on the daemon itself
- Daemon may keep open file descriptor in a fd-cache (for LIOPv2)

What the protocol does and what it does not

- LIOP do basic READ/WRITE operations
- No consideration about striping : every LIOP see the whole files as Lustre clients do
- No consideration about locking : let Lustre do the job
- LIOP will have embedded security (LIOPv2)
  - Kerberized authentication but no encryption (krb5i, krb5p)
  - Use “labelled request” following what Labelled NFS does

Marry LIOP and pNFS

LIOP+pNFS= a new pNFS support for Lustre

- What is to be done : client side
  - The kernel needs to talk LIOP Protocol
    - A kernel module is under development
    - We use the 9p over RDMA module as a template
- What is to be done : server side
  - LIOP server is designed, coding started in early 2015
  - NFS-Ganesha will be the natural pNFS/LIOP MDS
- But we need a pNFS layout

Flexible Files Layout seems to be a nice layout for LIOP

- Designed by Primary Data
- Client kernel modules will be part of kernel 3.20 (please correct me if I am wrong)
- The flexible approach behind this layout fits LIOP simplicity
- I need to talk with Tom about this, but I am quite optimistic
  - Flexible Files Layout allow many kinds of DS, including NFSv3 based
  - I define what LIOP will be, I can change it to fit Flexible File Layout requirements

Agenda

Lots of work to do

- Client side
  - Code LIOP kernel client with RDMA and TCP/IP transport layers
  - Marry LIOP and Flexible File Layout
- Server Side
  - Code LIOP server (started)
  - Make NFS-Ganesha ready for LIOP and Flexible File Layout : not started, but shouldn't be the hardest part of the job
- I hope to have alpha version of this by Q4 2015

When 9p turns into 10p

- As shown last year, 9p is a key point in Future IO architectire for HPC at CEA
- 9p does not have parallel features
- We'll add function to 9p to make it capable to do what pNFS does

9p+pNFS = 10p (to be continued...)

Questions ?