DE LA RECHERCHE À L'INDUSTRIE

www.cea.fr

IO Proxies via Ganesha/9P

Philippe DENIEL (philippe.deniel@cea.fr)

IO Challenges for 2020

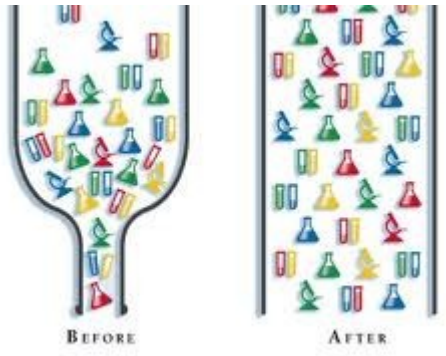
Manycore processors imply a reduced ratio of memory per core

The operating system will have less memory buffers for its own needs

- Less room in the OS for the file systems
- Even the TCP/IP network stack may become too expensive and be replaced by lower level but faster paradigm (like RDMA)

Kill the bottleneck!!

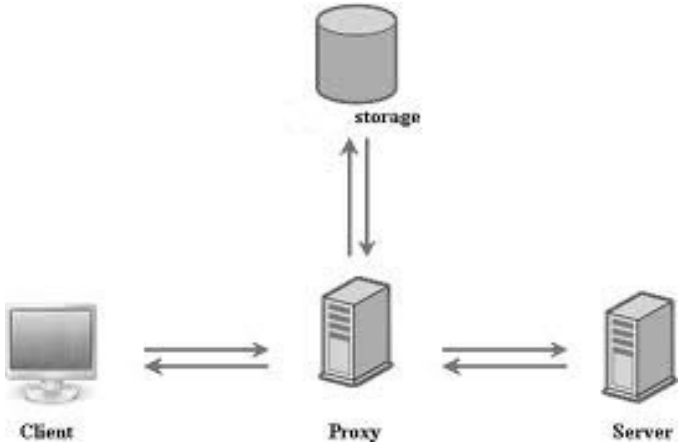
- Need for mechanisms to manage larger data without generating bottlenecks
- The former approach used in SMP is not valid anymore and would lead to an explosion of the number of clients



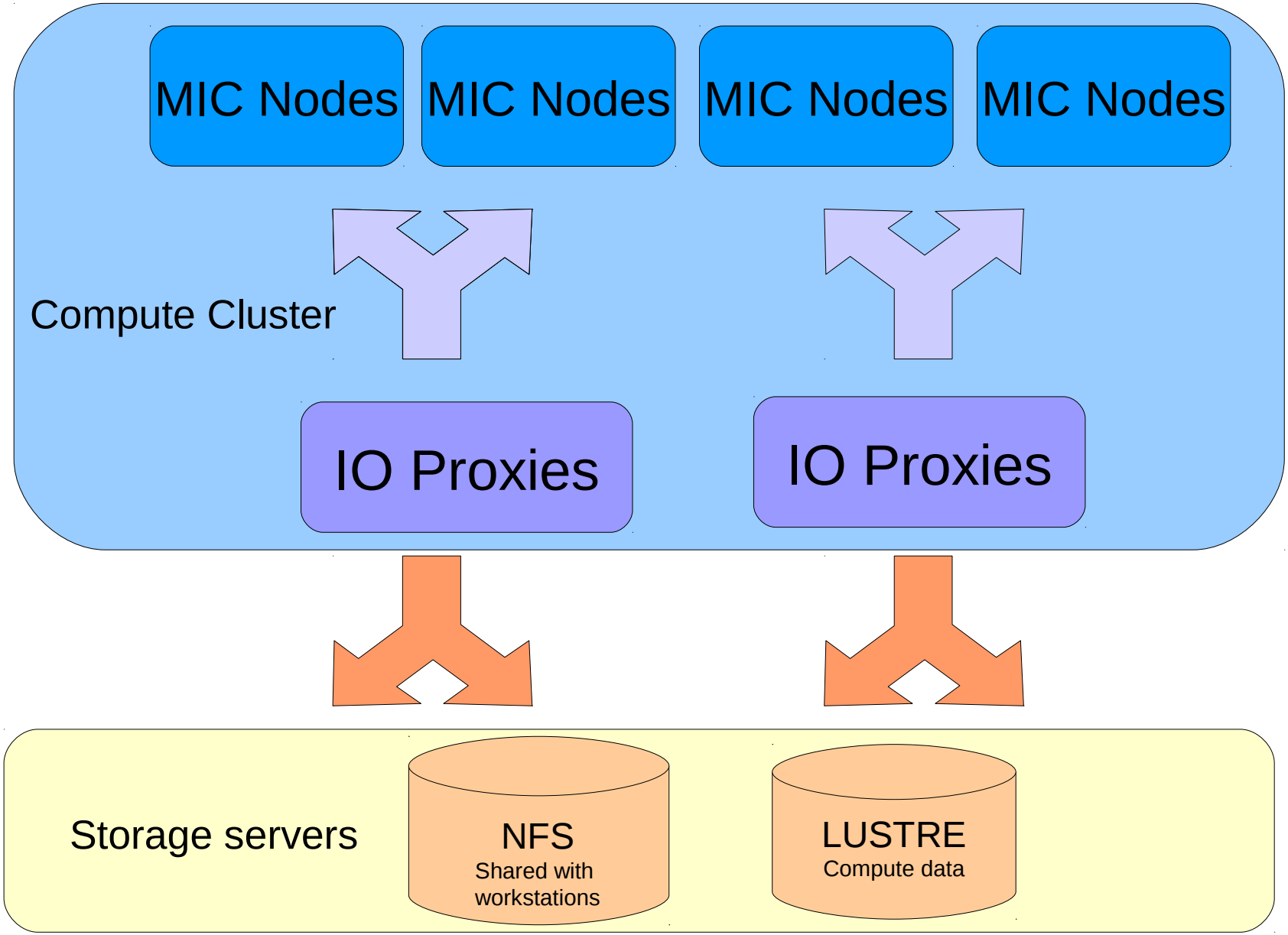
Using IO Proxy

Sets of node running a simulation job be served by a dedicated agent called IO Proxy

- The proxy is the only “data view provider” to the job
- The proxies are the only actual clients of storage resources (kill bottlenecks)
- The proxy is the natural place for optimization based on hints provided by upper layers (IO Libraries and simulation code)
  - Impact on data cache policies (keep only what is tagged as essential)
  - Impact on metadata cache policies (do not flush what will be used soon)



IO proxies inside future architecture



IO Proxies

IO Proxies are internal to the future compute machine

- Single path for computes nodes to access data
  - Lustre Filesystems
  - NFS remote servers

IO Proxies as “fuse”

- A single “evil” command can easily collapse a storage system
- A “rogue study” will only mess its own proxy
  - Use of internal metrics will help identifying toxic behaviors
  - In such a case, the proxy would slow pause, pause or even stop to protect the back-end
- A major failure on the IO Proxy will crash it, preventing the trouble to contaminate the whole machine

What we need for building an IO Proxy

Three key questions make the core of the design

- Data are moving on a network from IO Proxies to compute nodes
  - What kind of transport protocol for the best efficiency ?

1

  - Must have **lightweight** implementation on MIC nodes
  - Compute codes should not be limited by IOs: network must be **fast**

- From the MIC node's point of view, an IO Proxy is a file server
  - What kind of file server's protocol will we use ?

2

  - Should have **lightweight** client implementation due to MIC's constraints
  - Should provide full **POSIX semantics**

  - What kind of file server ?

3

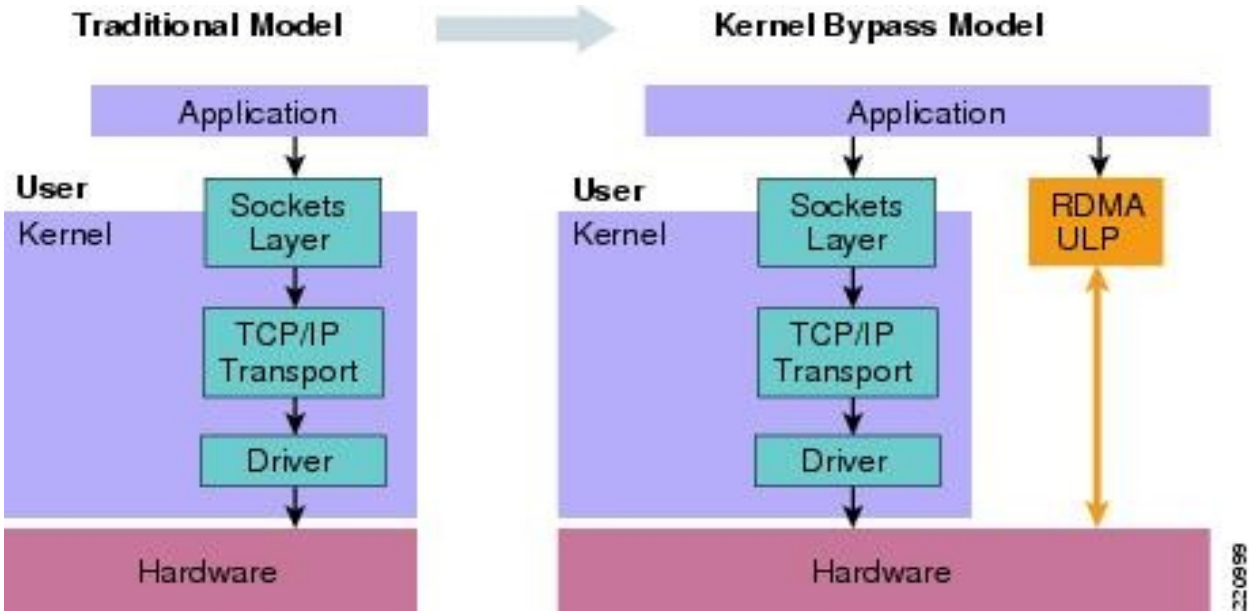
  - Must be able to **access LUSTRE and NFS** (at least)
  - Must support the chosen network and file server protocols (see question 1 & 2)

Efficient networking: the RDMA transport layer

RDMA: Remote Direct Memory Access

A machine allows another to write directly in a few windows in its own memory

- Simpler implementation compared to TCP/IP
- A *de facto* standard via Infiniband, iWARP and RoCE technologies
- LAN dedicated but fast network model
- Bypass several OSI layers to optimize performances



9P is a good choice for IO Proxy

9P fits IO Proxy's requirements

- 9P is a very lightweight protocol
- 9P is fast to interpret
  - Use little-endianess making XDR-like marshaling unnecessary
- 9P is buffer oriented, which fits well RDMA transport
- 9P make zero-copy based implementation easy and requires less memory
- 9P has all you need to implement full POSIX semantics
- 9P is quite complete (including lock support and xattr support)
- 9p.2000L client side is implemented inside the kernel (as v9fs) and is a living piece of code

File Server: Using Ganesha as an IO Proxy

What makes Ganesha fitting this need ?

- Ganesha's framework was opened enough to add 9P support to it
  - Integrating 9P took a few months
- Ganesha layered architecture allowed to add RDMA as new transport feature
  - Integration of RDMA as a transport layer
  - Using Mooshika library
- It embeds all of the mechanism to serve as a IO proxy
- Has both LUSTRE and NFS back-ends

Mooshika : RDMA transport layer

CEA is developing Mooshika, a user space library designed to provide easy RDMA integration in user space program

Mooshika was designed to be integrated to Ganesha

- Mooshika provides RDMA support for 9P/RDMA implementation
- Mooshika provides RPC/RDMA support for NFS/RDMA implementation

Mooshika is released as open source software

- Even outside Ganesha, an easy-to-use API designed for file server is a useful piece of software
- Mooshika will be a standalone project

IO is one of Jupiter's moons