Parallel Storage Systems for Large-Scale Machines

Doctoral Showcase

Christos FILIPPIDIS (*cfjs@outlook.com*)
Department of Informatics and Telecommunications,
National and Kapodistrian University of Athens

We acknowledge the support of Special Account for Research Grants of the National and Kapodistrian University of Athens.

Research Challenges

- Large-scale scientific computations tend to stretch the limits of computational power.
- Parallel computing is generally recognized as the only viable solution to high performance computing problems.
- I/O has become a bottleneck in application performance as processor speed skyrockets, leaving storage hardware and software struggling to keep up.

Factors affecting I/O performance

The most important factors affecting I/O performance are:

1. The number of parallel processes participating in the transfers,
2. The size of the individual transfers,
3. The I/O access patterns,
4. The storage architecture being used.

I/O performance limitations

1. Globally shared file systems, using current storage architectures, have several I/O performance limitations when used with large-scale systems, because:

   - Bandwidth does not scale economically to large-scale systems,
   - I/O traffic on the high speed network and on each storage server can be affected by other unrelated jobs.

2. Lack of coordination in the overall data flow (remote-local access)

Doctoral Contribution

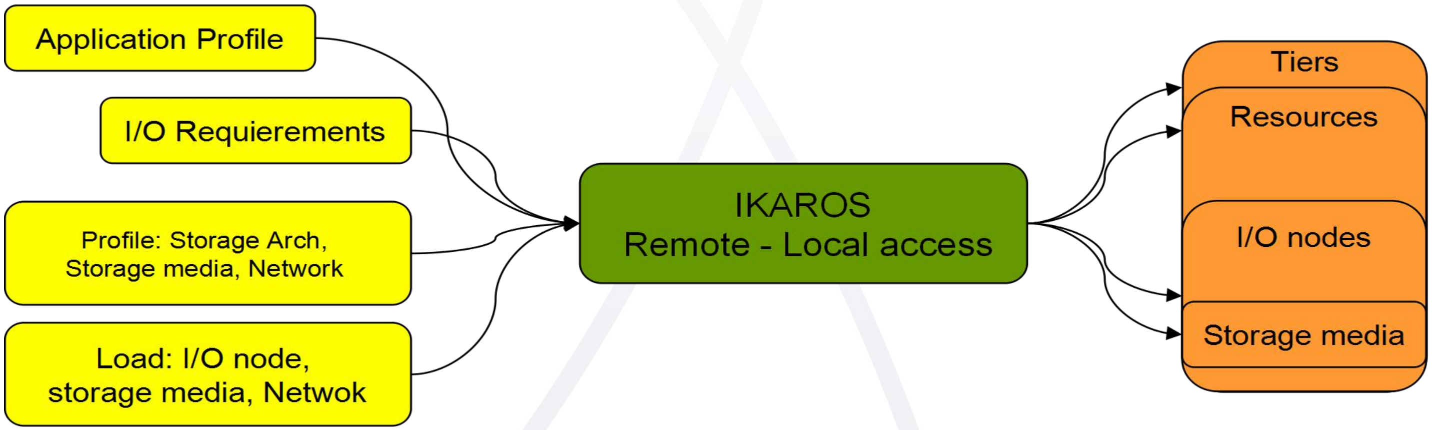
This study:

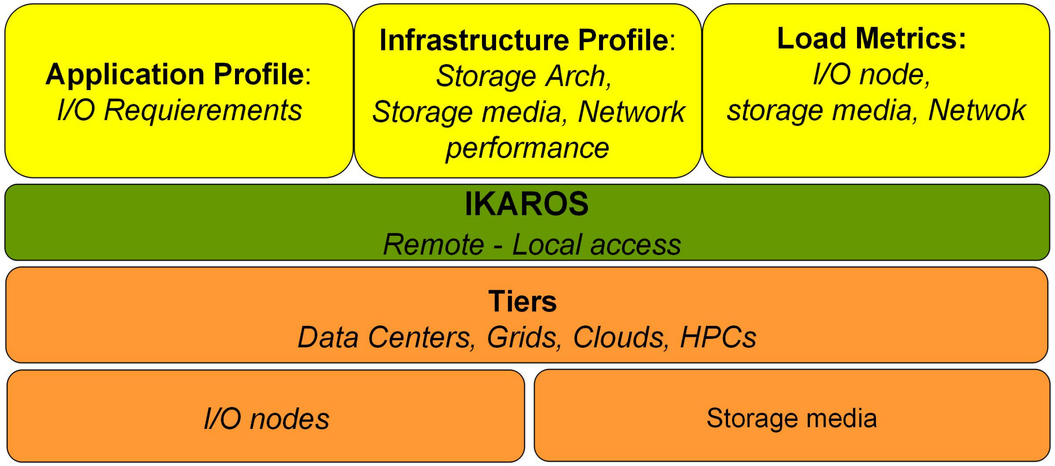
1. Proposes a dynamically coordinated I/O architecture based on input parameters (topology/profile of the infrastructure, the load metrics),

2. Creates, on the fly, dedicated or semi-dedicated clusters of HDDs per job.

3. Provides coordinated parallel data transfers on the overall data flow.

IKAROS Framework





<u>IKAROS</u>:

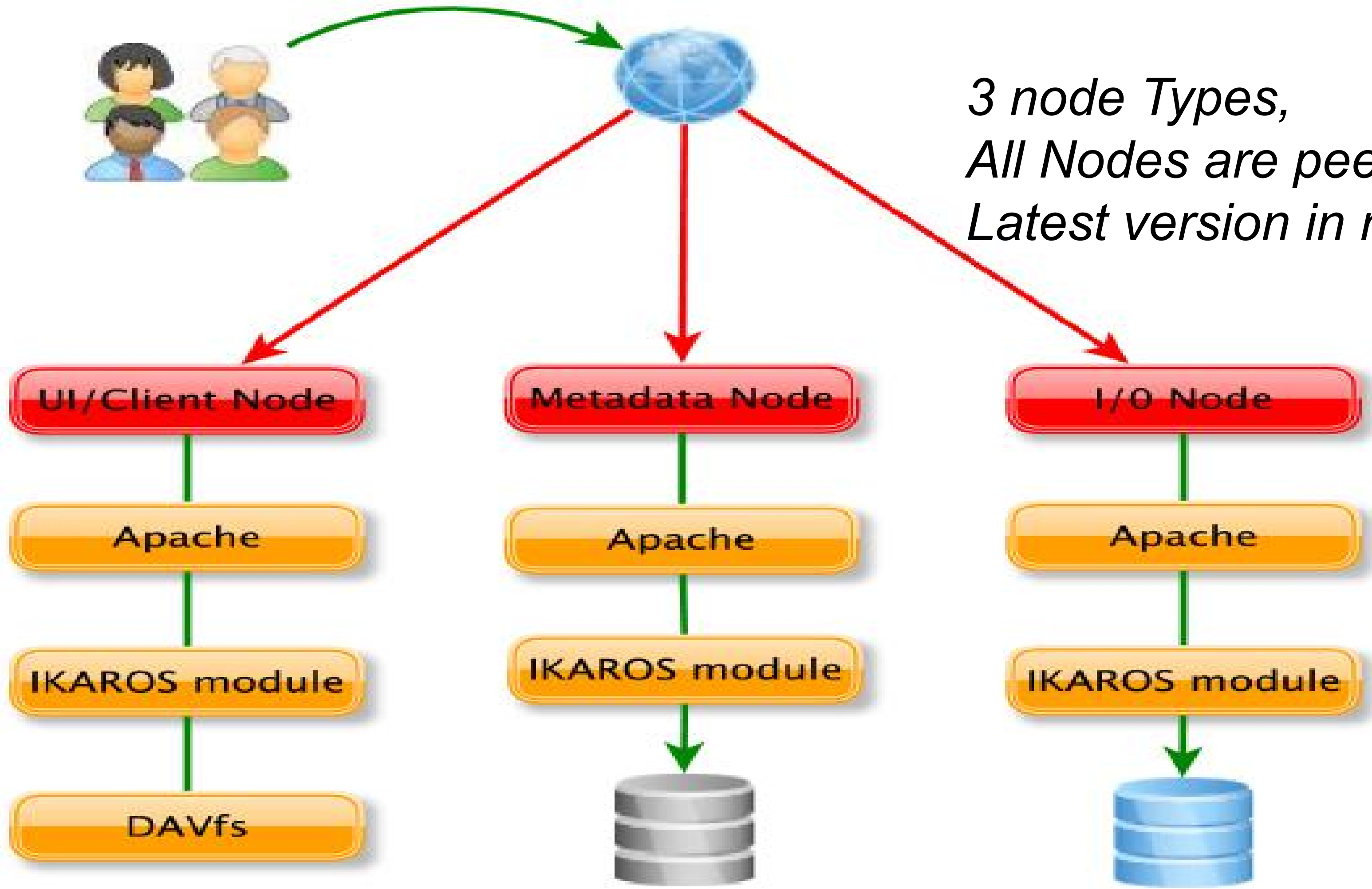
- Write-optimized system.
- Manages storage Resources (I/O nodes, network, storage media) in all Tiers, based on input parameters.

** each Tier is made up of several computing Centers and provides a specific set of services*

IKAROS Features

	HDFS	PVFS2	GPFS	IKAROS
Deployment model	Co-locates compute and storage on the same node	Separate compute and storage nodes	Separate compute and storage nodes	The user/app can choose both models, on the fly
Data layout	Exposes mapping of chunks to data-nodes to Hadoop applications	Maintains stripe layout information as extended attributes but not exposed to applications	not exposed to applications	Decides on demand the chunk mapping, based on input parameters. Exposes mapping of chunks to applications and users
Compatibility	Custom API and semantics for specific users	UNIX	UNIX	UNIX, WINDWOS, MAC
WAN capabilities	Can be exported through webdav	Can be exported through pNFS	Can be exported through pNFS	Build-in remote access capabilities. Supports parallel channels WAN data transfers, stripping servers, third party data transfers.

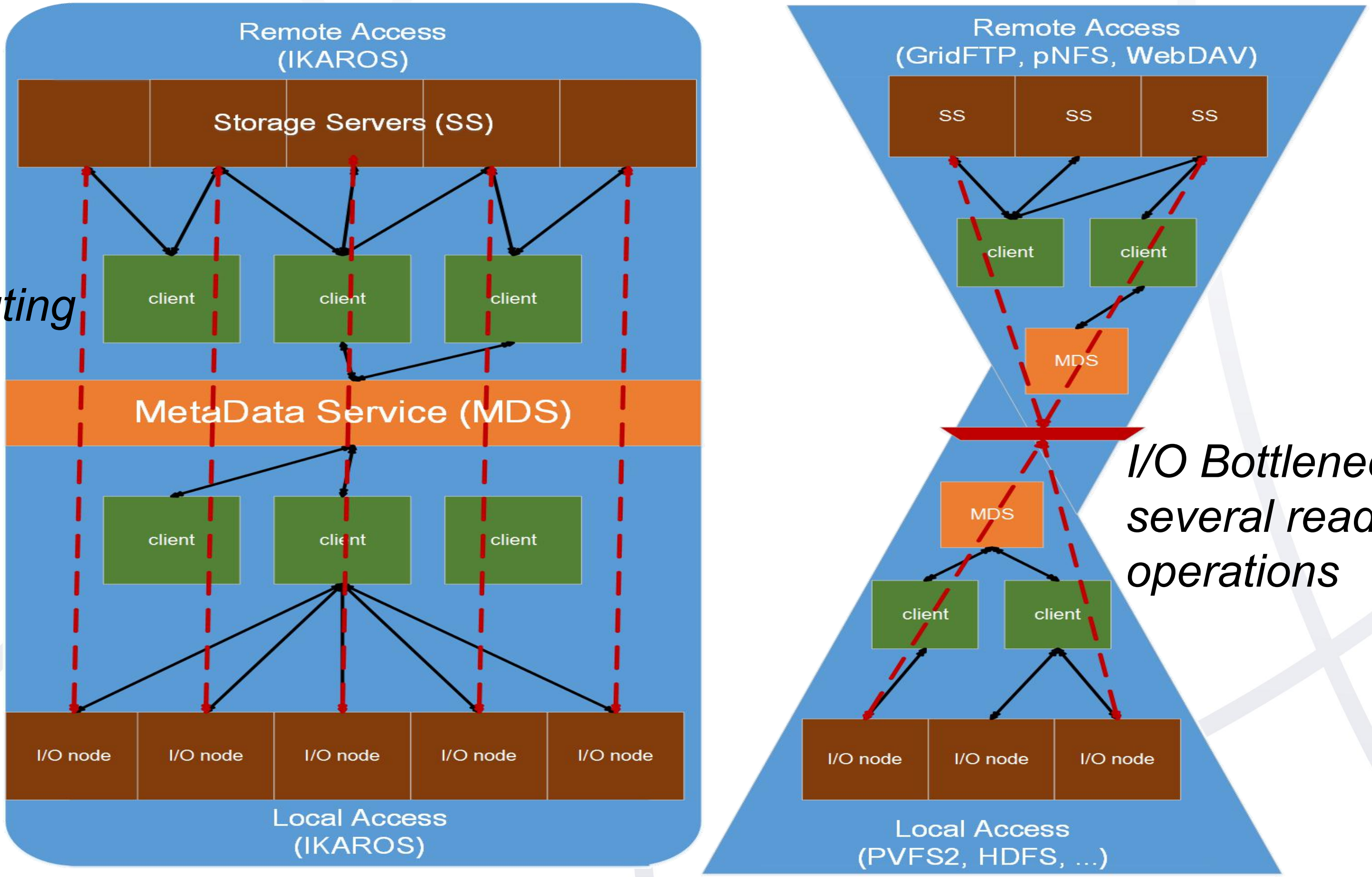
IKAROS Architecture



3 node Types,
All Nodes are peers.
Latest version in nodeJS

Remote-Local Access Overview

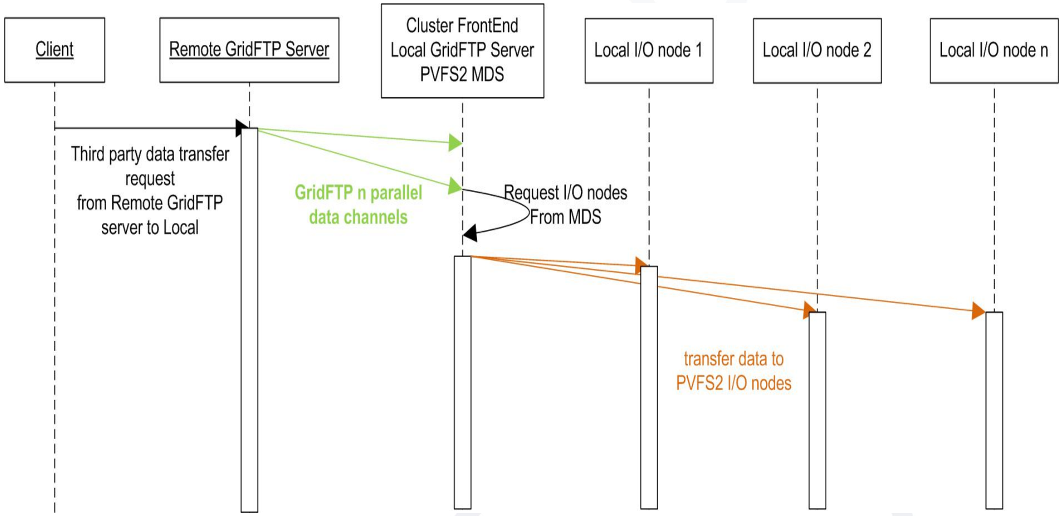
Reverse read,
Reverse HTTP,
mainly data routing



I/O Bottleneck,
several read-write
operations

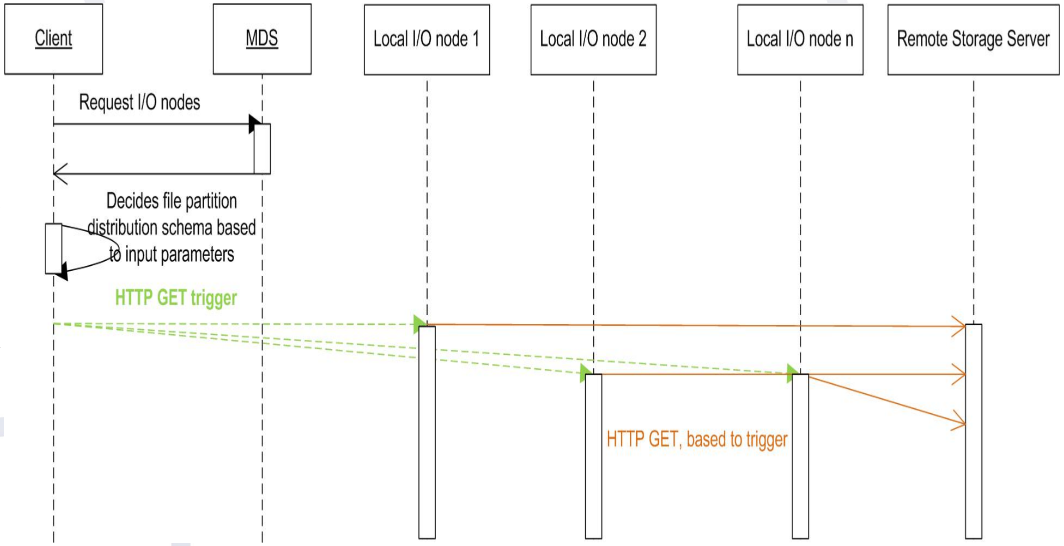
IKAROS: Direct access to each I/O node regardless of the Tier

IKAROS vs PVFS2+GridFTP



PVFS2+GridFTP:

- We must manually synchronize the stripe size and the stripe mapping between them.
- We have to initiate many independent transfers, incurring much overhead to set up and release connections.



IKAROS:

- By using IKAROS we apply only coordinated parallel data transfers
- Minimize disk and network contention

HPC Environment

- We compare IKAROS with GPFS in an **HPC** environment (N clients).
- We create, on the fly, dedicated or semi-dedicated clusters of HDDs per job.
- **Goal**: Isolate I/O functions of a process from other unrelated jobs.

Testbed (Cytera Machine)

	#	Specs
Compute nodes	96	12 Intel Xeon CPU cores, 48 GBs of RAM and 15K rpm local HDD
Storage Nodes	4	360 TBs raw disk space in 18 Raid 6 arrays each with 10 7200 rpms HDDs
GPFS-Meta data System	4, hosted at the storage nodes	Raid 10 arrays (one associated at each server)
Network Connectivity	-	QDR (40Gbit/s) infiniband

Testbed Features, Profiling
(Determine Input parameter)

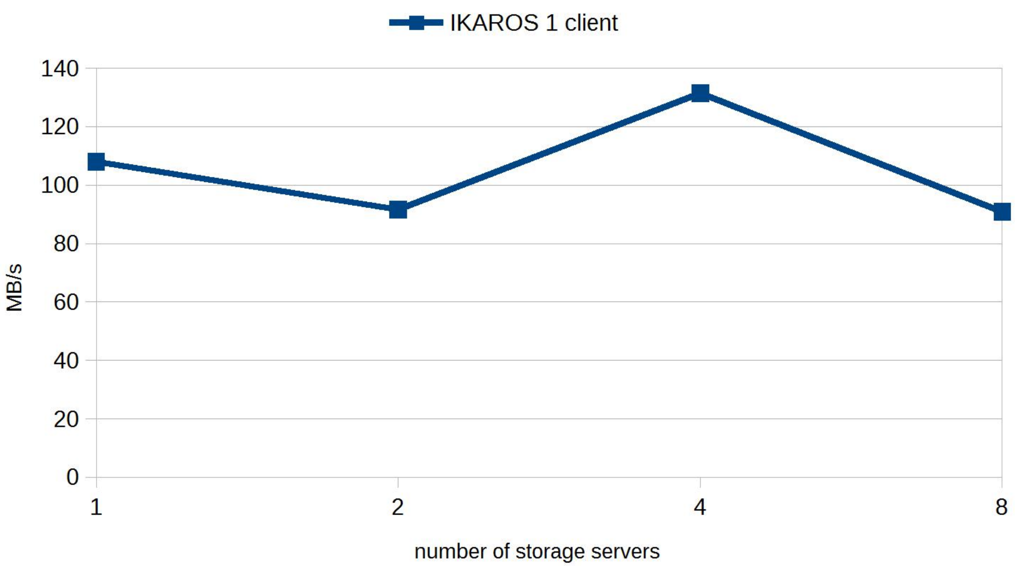
Network: QDR (40Gbit/s) infiniband

Storage media: Compute & Storage nodes (~140MB/s)

Optimal file chunk distribution

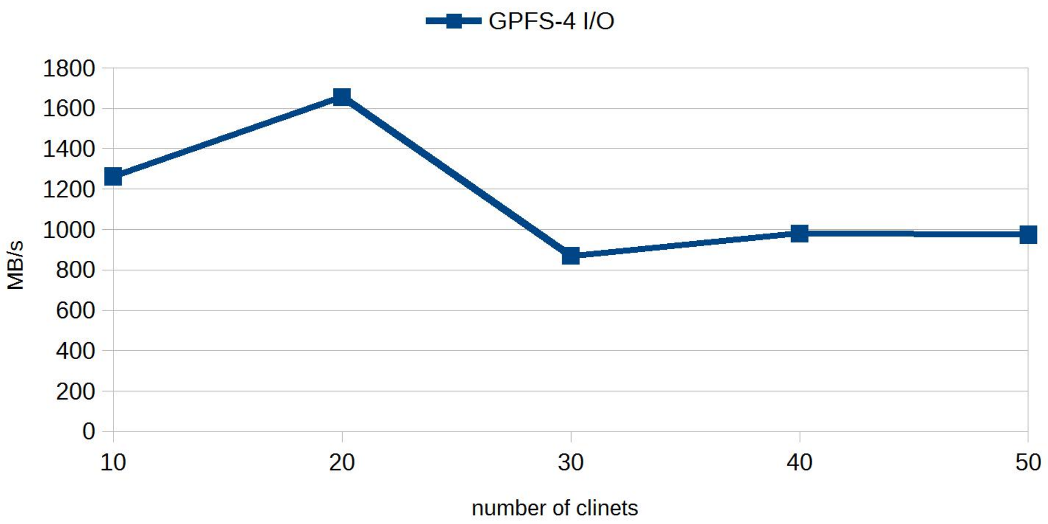
***Input Parameter*:**

(client/HDD ratio = 1/4).



Result: Due to storage media queuing mechanisms.

GPFS performance @ Cytera



<u>GPFS@ Cytera</u>
Clients/Storage-Server: 5/1

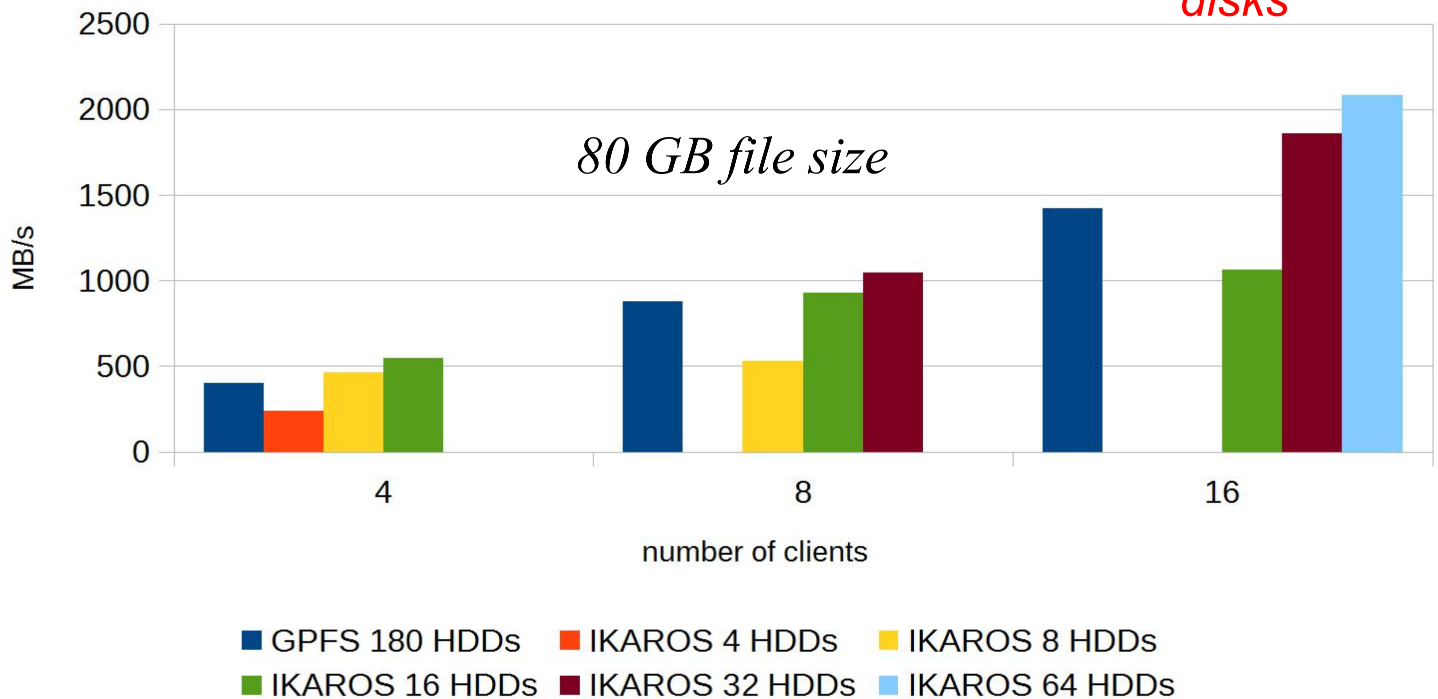
<u>GPFS@LLNL (2000)</u> :
Clients/Storage-Server: 4/1
(38 servers, 152 clients)

- 80 GB file size (do not fit to memory).
- 180 HDD-RAID6.
- 4 Storage Servers.
- Max I/O performance (Write):~1600 MB/s .
- The available storage resources (I/O and Network) are underutilized

Measurements @ Cytera

Create, on the fly, dedicated or semi-dedicated clusters of HDDs per job (input parameter: client/HDD ratio= 1/4).

improve performance by 33% with the 1/3 of the available hard disks



We are be able to fully utilize the available storage resources (I/O and Network).

IKAROS- KM3NeT.org

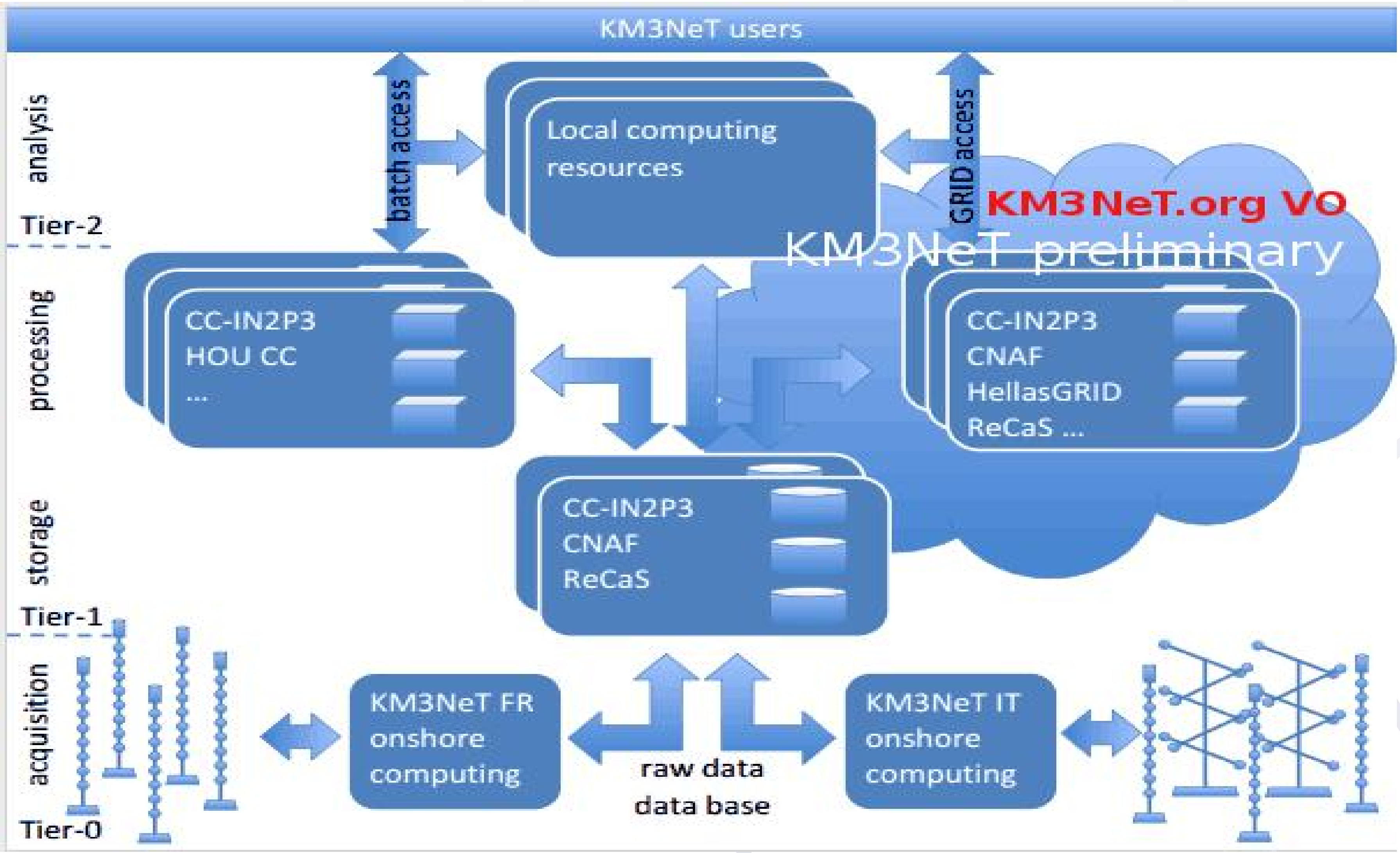
- IKAROS is part of the KM3NeT Computing Model.

- KM3NeT is a future European deep-sea research infrastructure hosting a new generation neutrino detectors.

- It is an ESFRI infrastructure and a CERN recognized experiment.

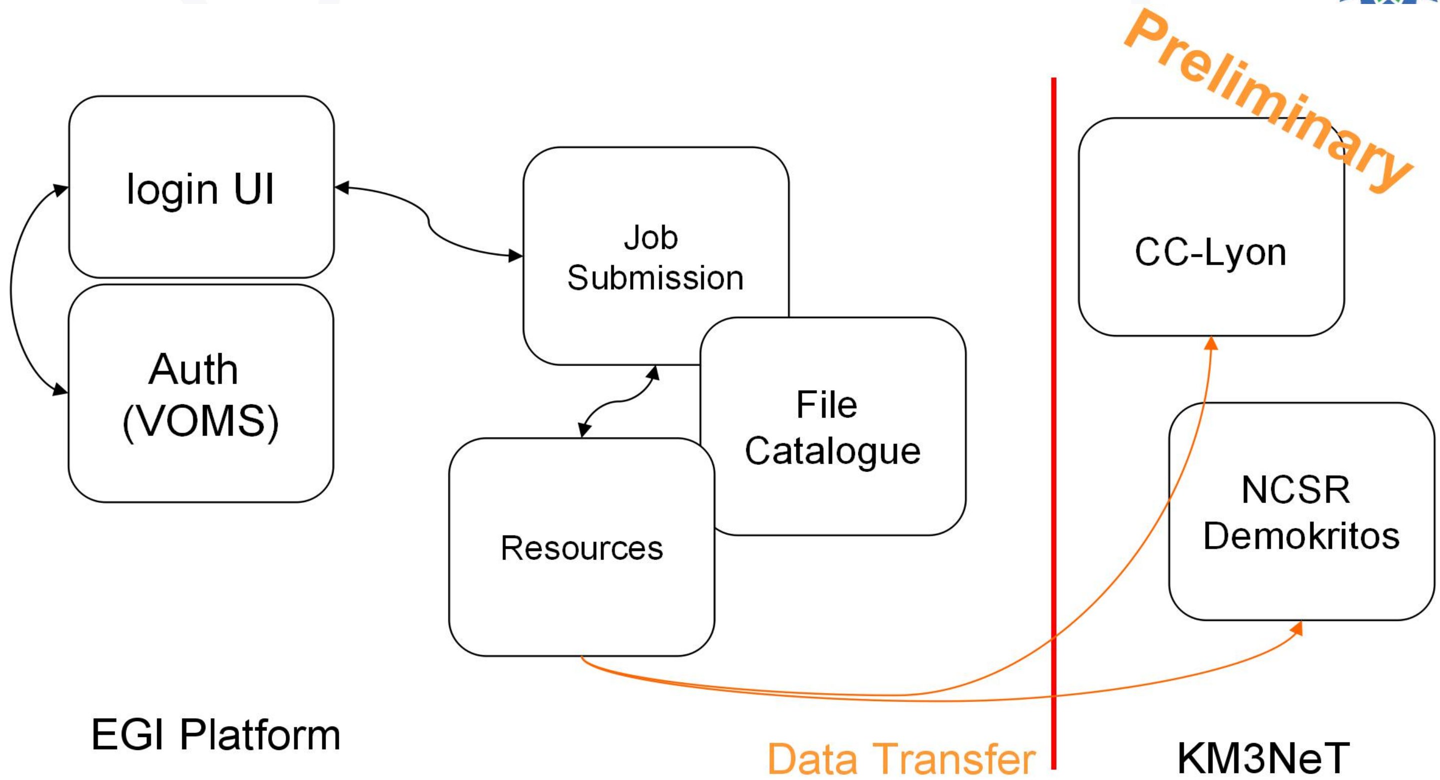
- The collaboration includes about 45 institutes or universities from 13 different countries.

KM3NeT Computing Model Overview



European Grid Infrastructure-KM3NeT Workflow

iRODS,
IKAROS,
DPM, DCACHE



- **Default procedure:** The data output is transferred from the Grid Worker Node to a local Grid storage element (SE),from the local SE to CC-Lyon, and then at the UI, by using the GridFTP and SSH protocols (several read-write operations).
- **IKAROS:** Sends the output directly to the destination (laptop, local computer cluster, CC-Lyon) (one read-write operation)

Conclusions

This study:

- Proposes a dynamically coordinated I/O architecture, based on input parameters.
- Creates, on the fly, dedicated or semi-dedicated clusters of HDDs per job.
- Provides coordinated parallel data transfers on the overall data flow.
- Minimizes disk and network contention.
- Improves I/O performance by 33% with the 1/3 of the available hard disks.