European Union
European Social Fund

OPERATIONAL PROGRAMME
EDUCATION AND LIFELONG LEARNING
investing in knowledge society
MINISTRY OF EDUCATION & RELIGIOUS AFFAIRS
MANAGING AUTHORITY
Co-financed by Greece and the European Union

NSRF
2007-2013
programme for development
EUROPEAN SOCIAL FUND

Using IKAROS as a data transfer and management utility within the KM3NeT computing model

Christos FILIPPIDIS (a,b), Yiannis COTRONIS (b), Christos MARKOU (a)
a:NCSR Demokritos, b: University of Athens

Large-scale scientific computations

●Large-scale scientific computations tend to stretch the limits of computational power.

●I/O has become a bottleneck in application performance.

●The most important factors affecting 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.

I/O access patterns

1.Compulsory, consist of I/Os that must be made to read a program's initial state from the disk and write the final state back to disk when the program has finished.

2.Checkpoint/restart, are used to save the state of a computation in case of a hardware or software error which would require the simulation to be restarted.

3.Regular snapshots of the computation's progress.

4.Out-of-core read/writes for problems which do not fit to memory.

5.Continuous output of data for visualization and other post-processing.

Globally shared file systems

●Globally shared file systems have several performance limitations when used with large-scale systems, because:

1.Bandwidth does not scale economically to large-scale systems.

2.I/O traffic on the high speed network can impact on and be influenced by other unrelated jobs.

3.I/O traffic on the storage server can impact on and be influenced by other unrelated jobs.

IKAROS Framework

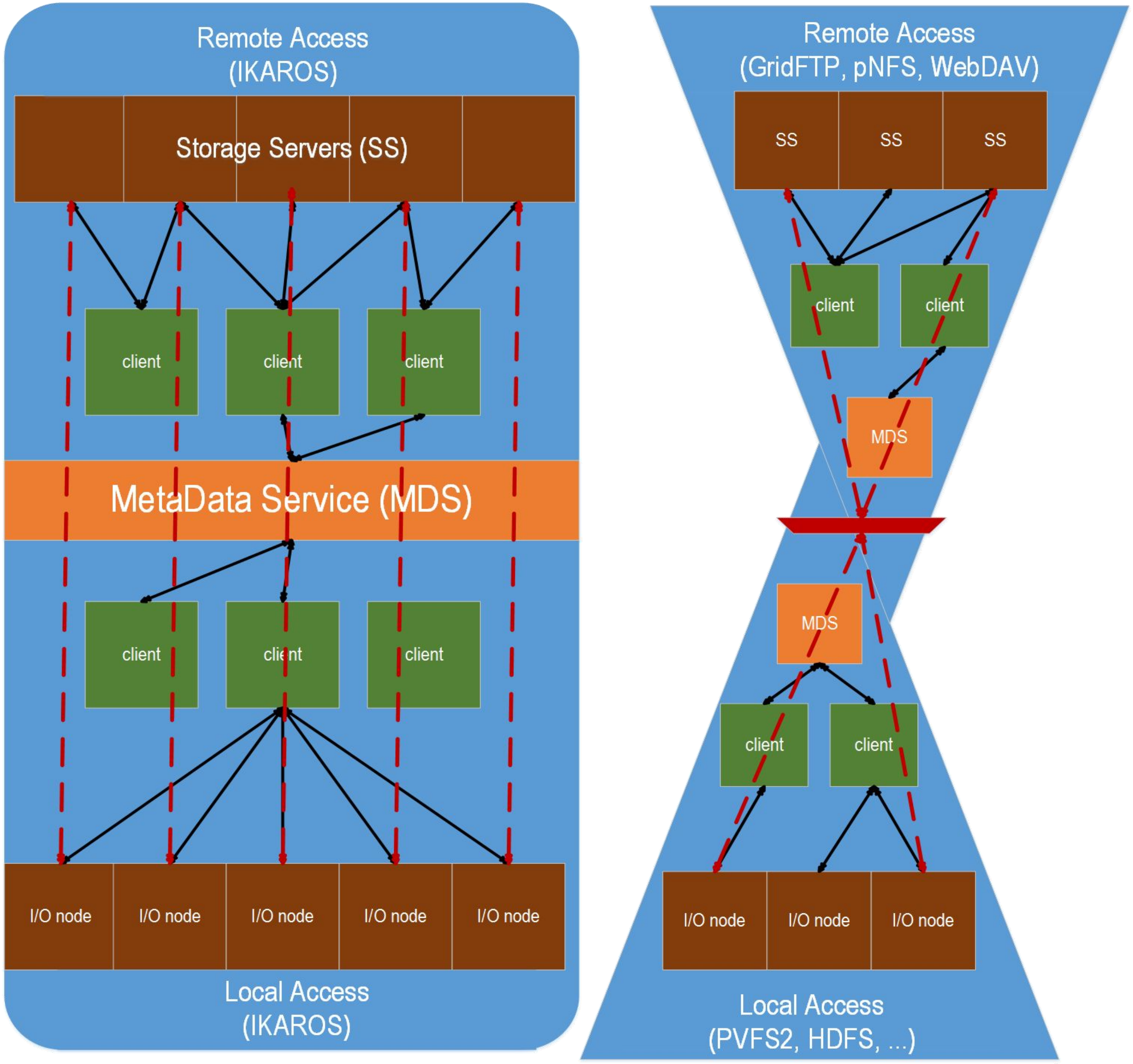
●Enables us to create ad-hoc nearby storage formations in order to isolate the resources being used and increase performance.

● Can use a huge number of I/O nodes in order to increase the available bandwidth (I/O and network).

●Unifies remote and local access in the overall data flow, by permitting direct access to each I/O node.

●This approach enables us to connect, at the users level, the several different computing facilities used (Grids, Clouds, HPCs, Data Centers, Local computing Clusters and personal storage devices), on-demand, based on the needs.

IKAROS Design (1/2)



IKAROS Design (2/2)

- Allows data in a file to be striped across multiple disk volumes on multiple heterogeneous nodes.

- Provides the utility for the storage system to access and transfer a data file in parts and in parallel mode, without a specific order, according to a client request.

- Defines three types of nodes: The User Interface(UI)/Client node, the Meta-data node and the I/O node.

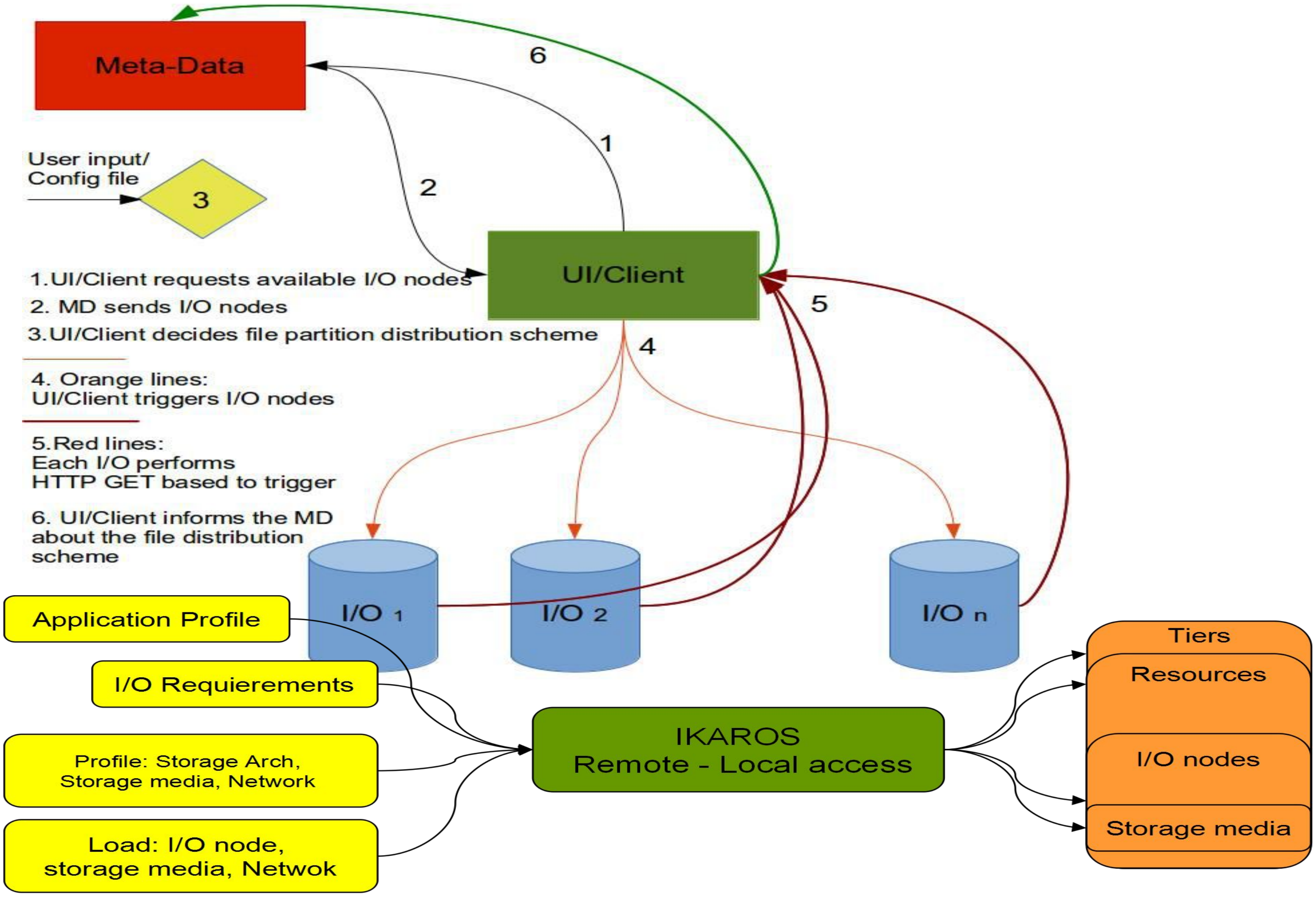
- Node types are peers with the ability to act in any mode.

- First version was developed as an Apache Dynamic Shared Object (DSO), the latest version is written in nodeJS.

HDFS, PVFS2, GPFS, 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	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 Basic Usage Scenario



1.UI/Client requests available I/O nodes

2. MD sends I/O nodes

3.UI/Client decides file partition distribution scheme

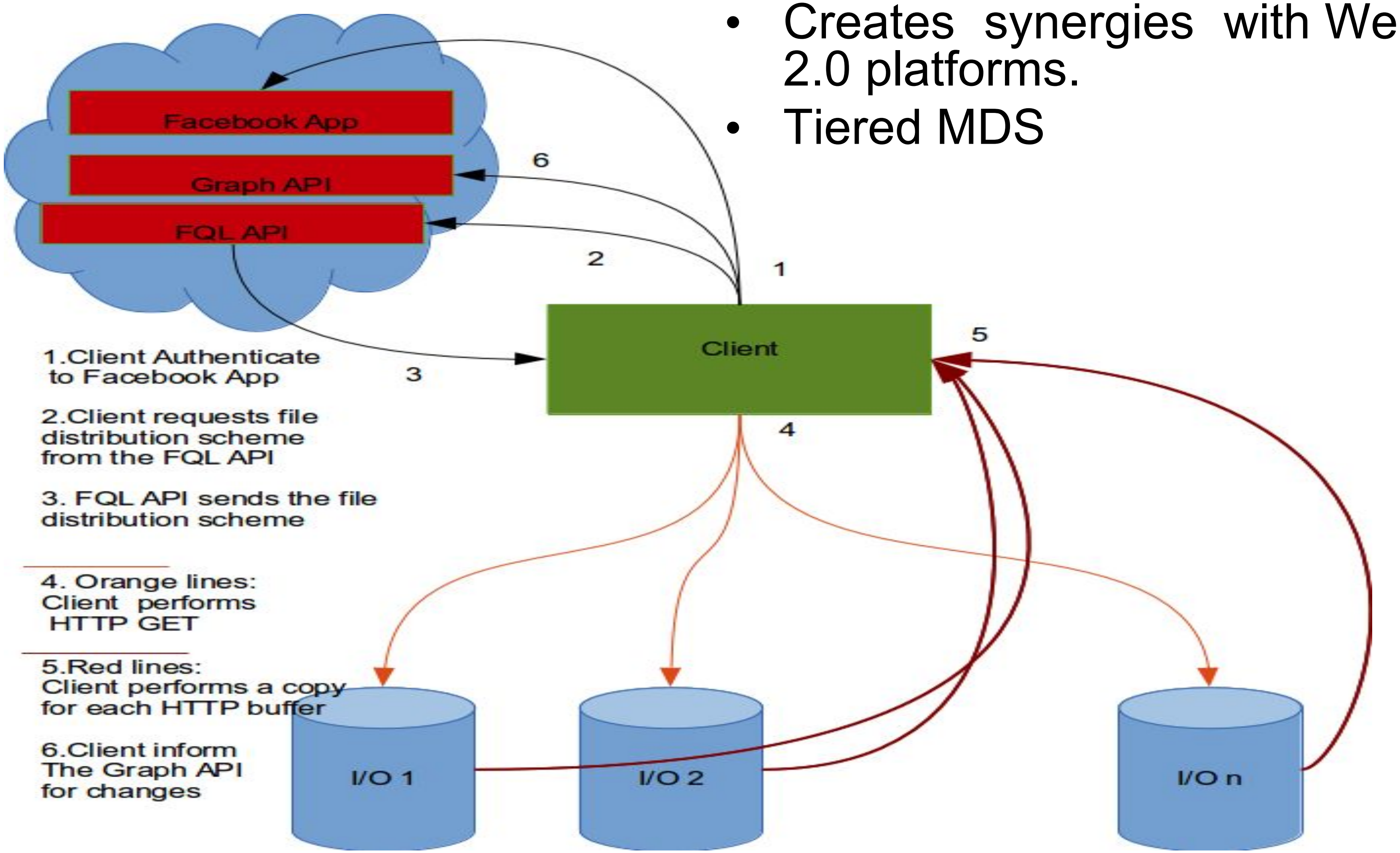
4. Orange lines:
UI/Client triggers I/O nodes

5.Red lines:
Each I/O performs
HTTP GET based to trigger

6. UI/Client informs the MD
about the file distribution
scheme

IKAROS hybrid model (1/2)

- Is using well known standards such as the HTTP and the JSON
- Creates synergies with Web 2.0 platforms.
- Tiered MDS



1.Client Authenticate to Facebook App

2.Client requests file distribution scheme from the FQL API

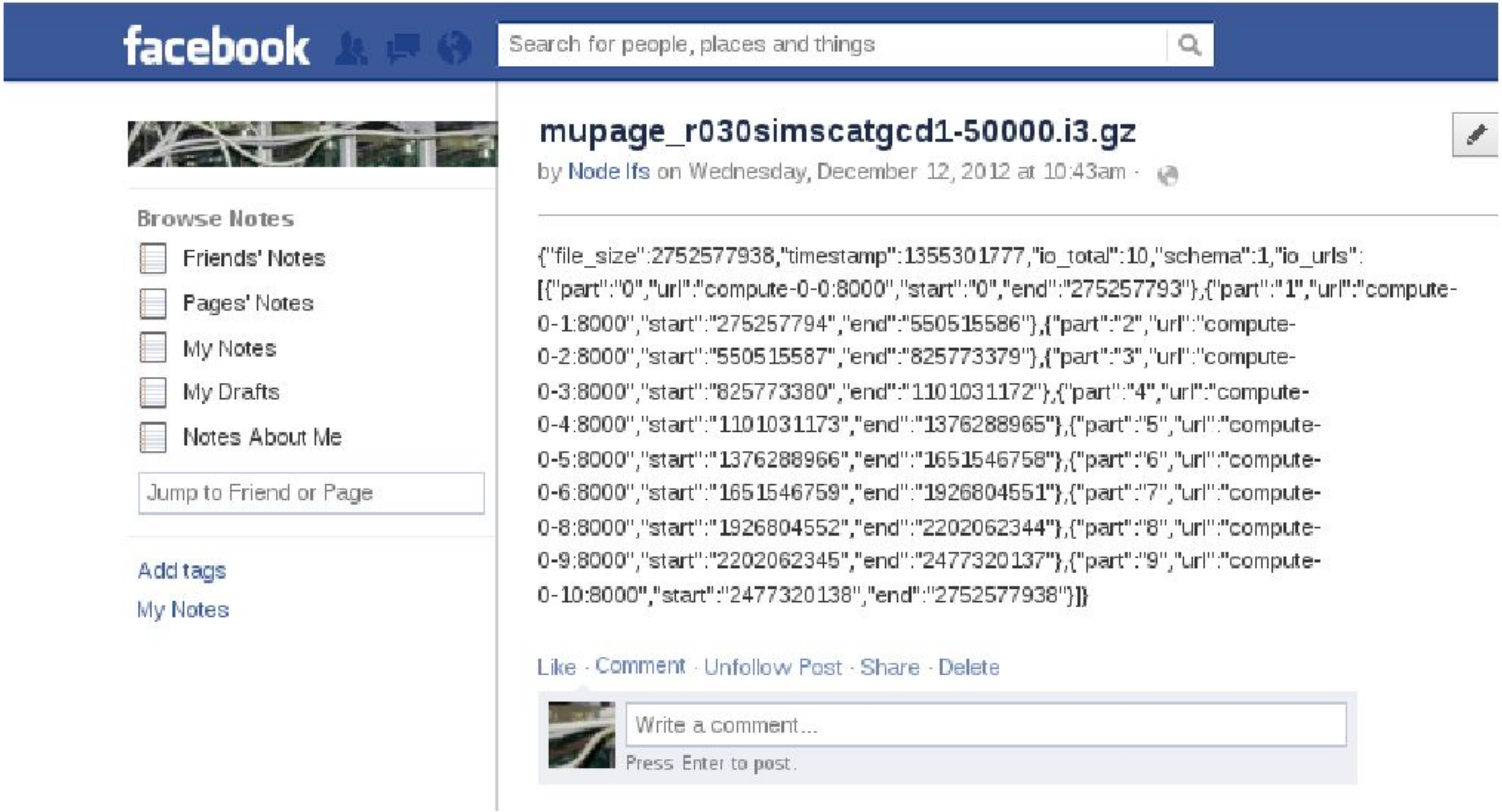
3. FQL API sends the file distribution scheme

4. Orange lines: Client performs HTTP GET

5.Red lines: Client performs a copy for each HTTP buffer

6.Client inform The Graph API for changes

IKAROS hybrid model (2/2)



mupage_r030simscatgcd1-50000.i3.gz

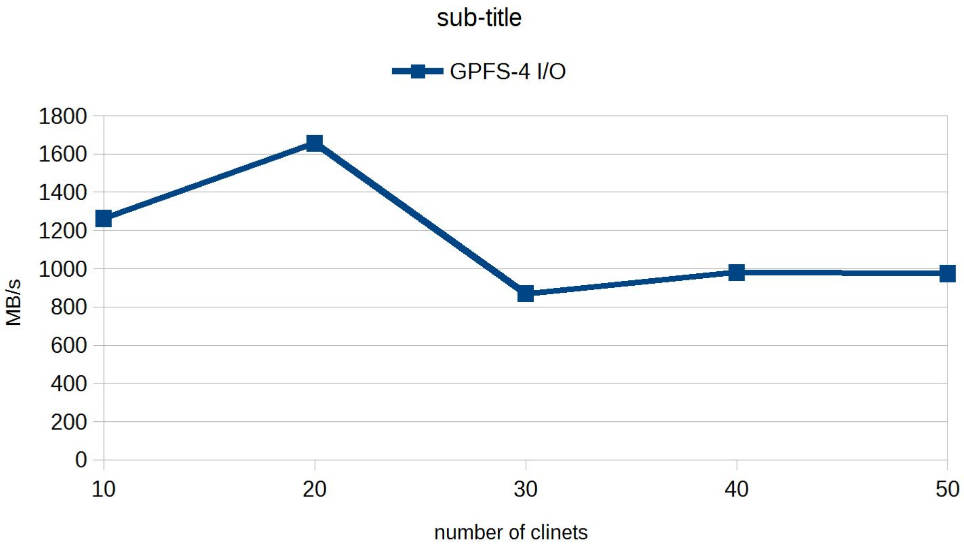
by Node Ifs on Wednesday, December 12, 2012 at 10:43am ·

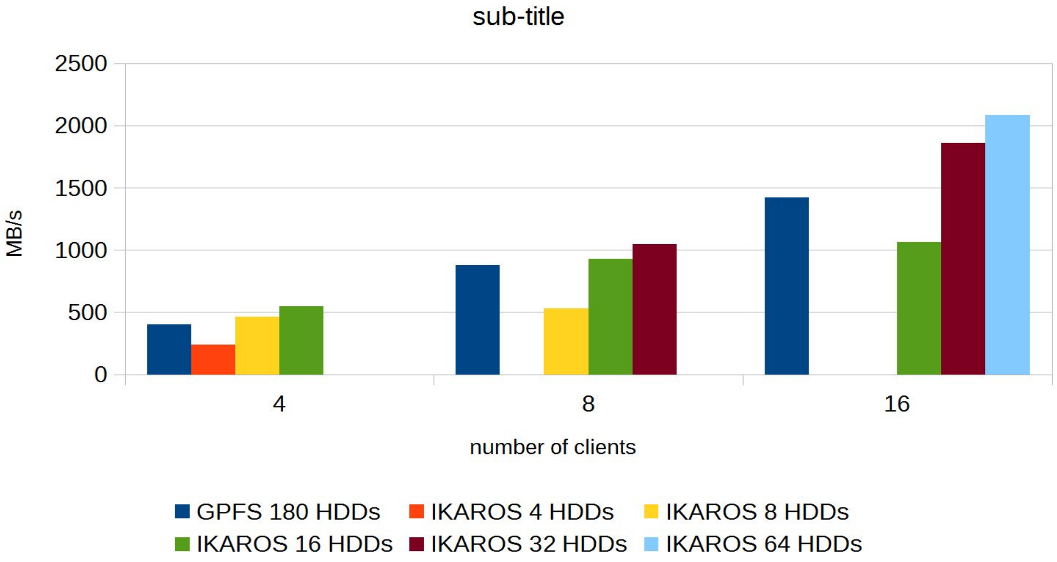
{"file_size":2752577938,"timestamp":1355301777,"io_total":10,"schema":1,"io_urls":
[{"part":"0","url":"compute-0-0:8000","start":"0","end":"275257793"},{"part":"1","url":"compute-
0-1:8000","start":"275257794","end":"550515586"},{"part":"2","url":"compute-
0-2:8000","start":"550515587","end":"825773379"},{"part":"3","url":"compute-
0-3:8000","start":"825773380","end":"1101031172"},{"part":"4","url":"compute-
0-4:8000","start":"1101031173","end":"1376288965"},{"part":"5","url":"compute-
0-5:8000","start":"1376288966","end":"1651546758"},{"part":"6","url":"compute-
0-6:8000","start":"1651546759","end":"1926804551"},{"part":"7","url":"compute-
0-8:8000","start":"1926804552","end":"2202062344"},{"part":"8","url":"compute-
0-9:8000","start":"2202062345","end":"2477320137"},{"part":"9","url":"compute-
0-10:8000","start":"2477320138","end":"2752577938"}]}

Experiments-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

Comparing IKAROS with GPFS (Cytera-Machine)





IKAROS-KM3NeT

Corsika Grid job test submission:

- 200 jobs submitted (5000 events). Output ~ 14GB send directly at Demokritos & Lyon (single step: WN-> Lyon).

- 50 jobs submitted (50000 events). Output ~74GB send directly at Demokritos & Lyon (single step: WN-> Lyon).

- Default procedure: grid WN -> Storage Element (SE) -> User Interface (UI) -> pc/cluster

Conclusions

- IKAROS helps us address several limitations which current file systems facing with large scale infrastructures.

- IKAROS approach enables us to create more user-driven computing facilities with application users and owners playing a decisive role in governance and focusing on placing computer science and the harvesting of ‘big data’ at the center of scientific discovery.