uc3m | Universidad Carlos III de Madrid

ARCOS

Hercules: scalable and network portable in-memory ad-hoc file system for data-centric and high-performance applications

Javier Garcia-Blas and Jesus Carretero

University Carlos III of Madrid, Spain

fjblas@inf.uc3m.es

ADMIRE
malleable data solutions for HPC

ADAPTIVE MULTI-TIER INTELLIGENT
DATA MANAGER FOR EXASCALE

EuroHPC
Joint Undertaking

GOBIERNO DE ESPAÑA
MINISTERIO DE CIENCIA E INNOVACIÓN

ADMIRE Users Day - Barcelona

Motivation

- I/O-intensive HPC-based applications have been primarily based on distributed object-based file systems.
  - **Separate data** from **metadata** management.
  - Enable each client to **communicate in parallel** with multiple storage servers.
- Exascale I/O raises the throughput and storage capacity requirements by several orders of magnitude.
- Current challenges:
  - Systems already developed for data analytics are not directly applicable to HPC due to the **fine-granularity** involved in scientific applications.
  - Semantic gap between the application requests and the way they are managed by the storage back-end at the block level.

Hercules

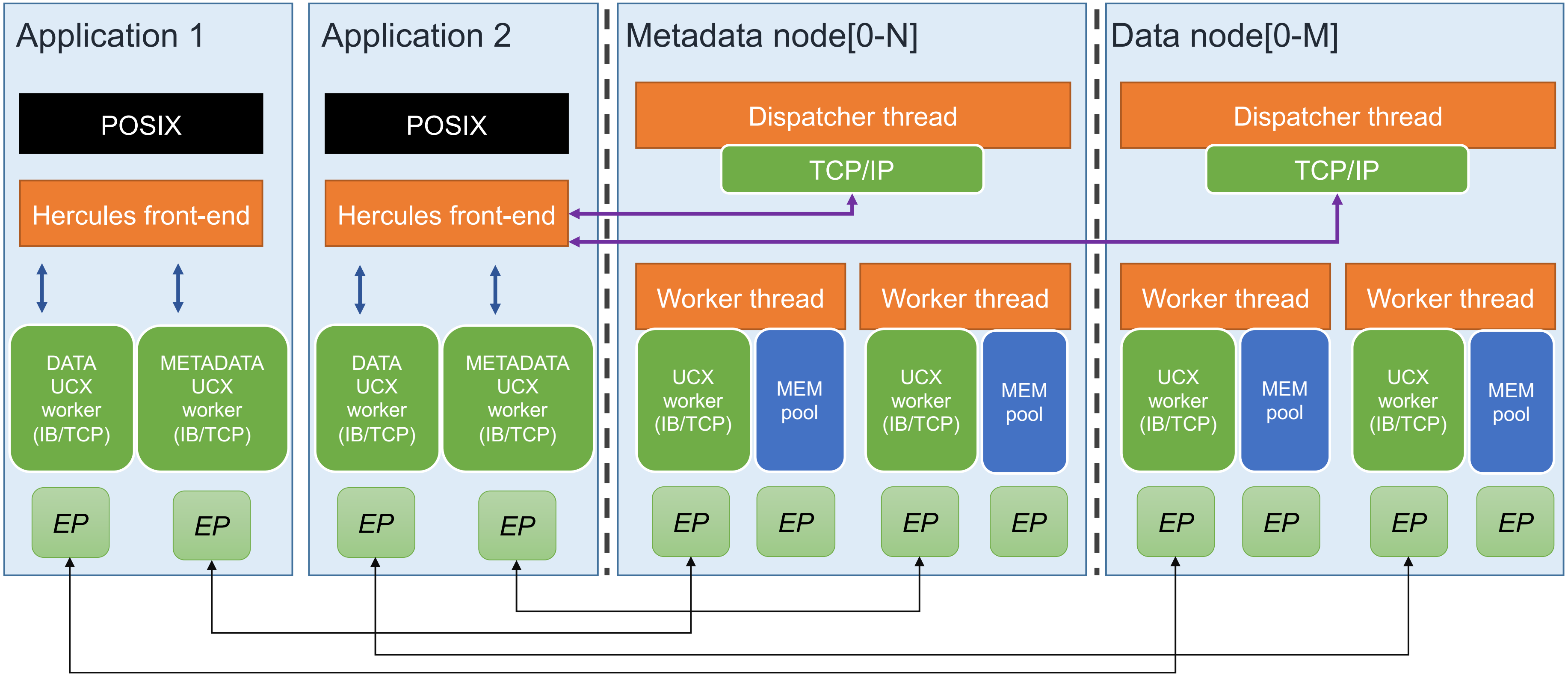
- Ad-hoc/in-memory storage solution for volatile data.
- Distributed key-value store.
- Provides a flexible API.
- Makes use of main memory as the storage device.
- Provides multiple data distribution policies.
- Exposes a POSIX/non-POSIX interface.
- Open source project.
- Fully implemented POSIX support (passed full IO500 benchmark).

https://gitlab.arcos.inf.uc3m.es/admire/hercules

Unified Communication X (UCX)

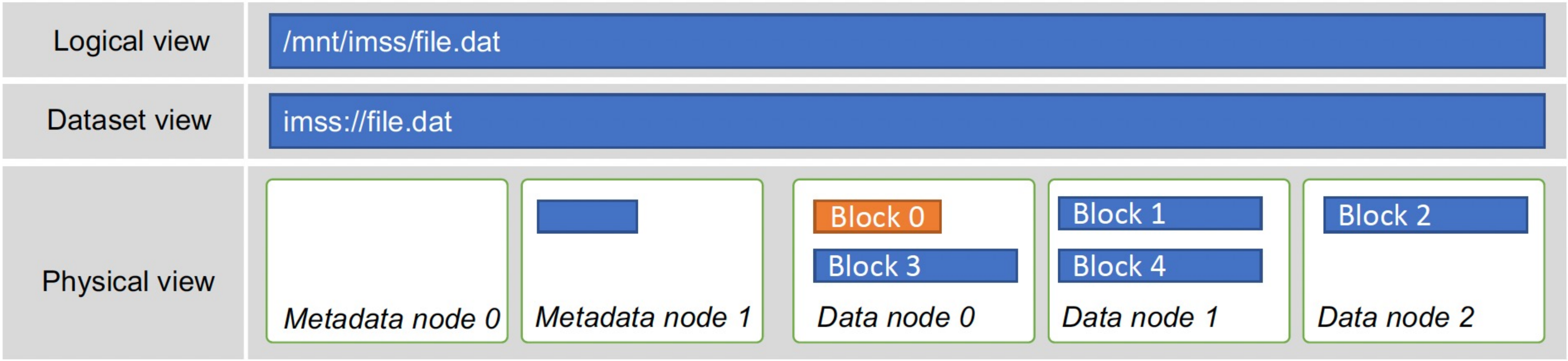
- Generic abstraction of the network layer
- Supported devices: Infiniband, Omni-path, TCP, shared memory
- Benefits of using UCX inside Hercules:
  - Multiple network interfaces/protocols available (TCP/IP, Omnipath, Infiniband supported).
  - Zero-copy message transfers of large data packages (>= 1 Mbytes).
  - Eliminated internal copies from application to network layer.
  - Asynchronous communication between peers.
  - RDMA QoS isolation.
  - End-point/two-sided-based communication.

Hercules Architecture



Internal data layout

- Data paths are translated from logical to dataset shape (logical view).
- Files are divided into multiple blocks under multiple data nodes (physical view).
- Distribution policy determines the physical location of blocks, mapping the physical data/metadata nodes (mapping).
- Inter-node metadata information is stored at the first metadata node mapping(0).



Logical view

/mnt/imss/file.dat

Dataset view

imss://file.dat

Physical view

Metadata node 0

Metadata node 1

Block 0

Block 3

Data node 0

Block 1

Block 4

Data node 1

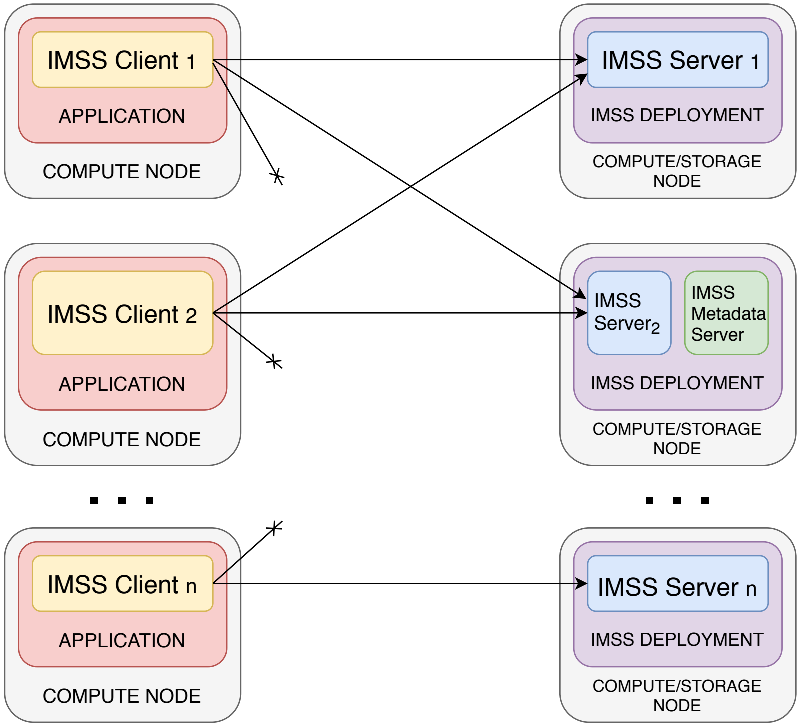
Block 2

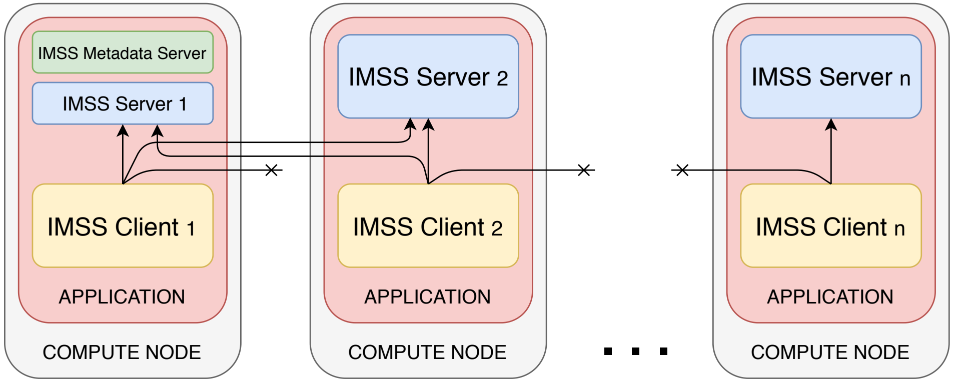
Data node 2

Deployment strategies

application-dettached

application-attached





Hercules Features

- Non-blocking/tag-based communication (MPI style)
- Low-level communication schema (in contrast to Margo RPC)
- Client-side
  - Data and metadata UCX’s workers enables **communication overlap**.
  - Malleability
    - Client nodes store a list of current available workers.
    - This list can be adapted during runtime.
  - QoS
    - Interfaces and protocols can be enabled/disabled to adapt **network requirements**.
    - Communication can be upgraded/downgraded (Infiniband to TCP).
  - Communication parameters configured by using environment variables.
- Server-side
  - One single listener per worker thread.
  - Stores a pool of active end-points (two-sided communication).

Data distribution policies

- **ROUND ROBIN**: data blocks are distributed among the Hercules servers.
- **BUCKETS**: each dataset is divided into the same number of chunks as number of servers. Each chunk is composed by a consecutive number of data blocks, equally distributed. Then, each chunk is assigned to a unique server.
- **HASHED**: a hash operation is applied over each data block key to discover the mapped server.
- **CRC16bits** & **CRC64bits**: similar to HASHED policy, but a sixteen/sixty four bits CRC operation is applied over the data block key.
- **LOCAL**: each data block is handled by the Hercules server running in the same node that the client.

File system malleability

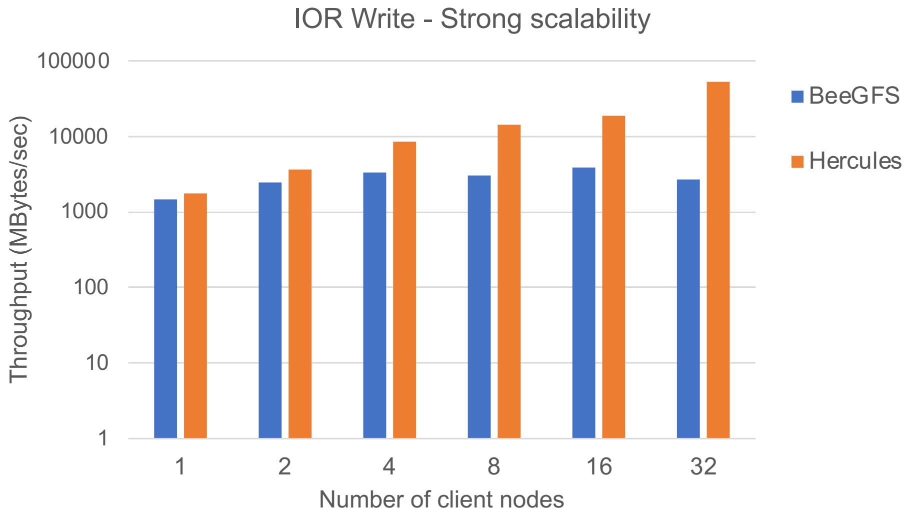
- Malleability operations can be started by two alternatives sources:
  - external controller or
  - internal heuristic.
- Internal heuristic determines whether a malleability operation should be carried out.
- Users define recommended I/O throughput (RIO) for the I/O system.
- Hercules tracks the current throughput provided by the I/O system to the application (AIO).
- Throughput distance is currently computed using a time series obtained by write/read in-place monitoring:
  - Consecutive operations,
  - Datasets accesses.

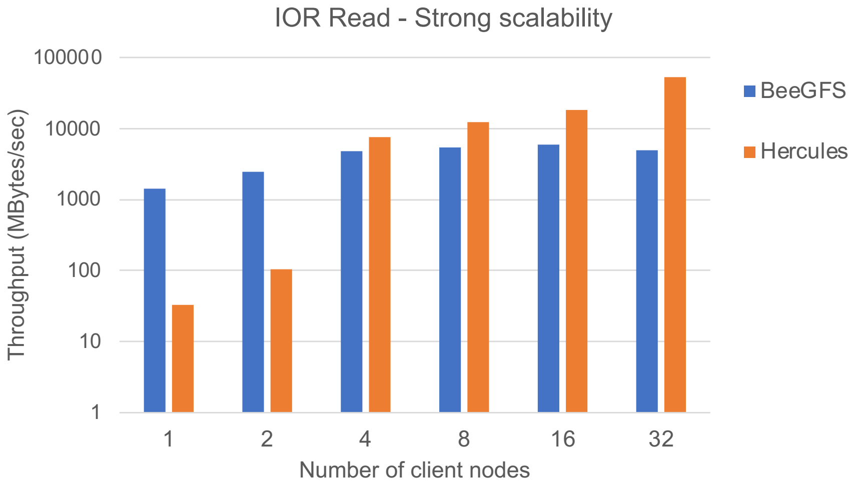
Access to the storage infrastructure

- API library
- FUSE
- LD_PRELOAD by overriding symbols
  - Facilitates to integrated with existing applications.
  - Works on booth attached and detached deployment strategies.
  - Passed IO500 benchmark succesfully.

Evaluation (Scalability)

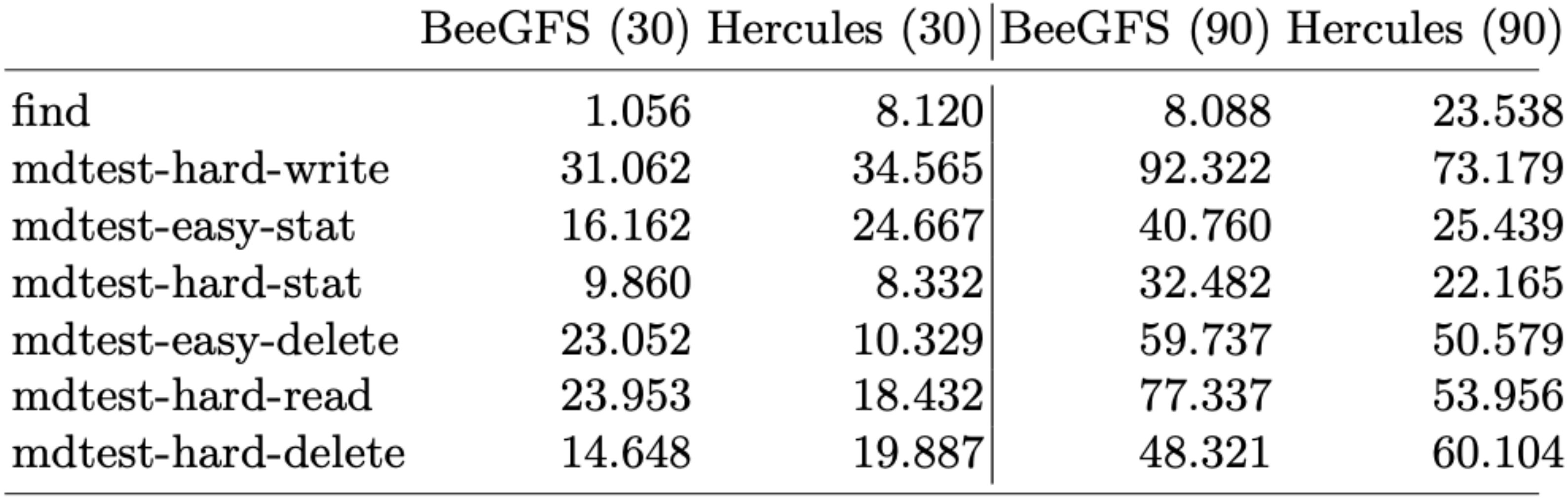
- University of Torino cluster.
- 64 Broadwell compute nodes. Intel Onmi-path running at 100 Gbps
- UCX 1.15. OpenMPI 4.1
- IOR. Strong scalability, single shared file accesses. 512 Kbytes block size.





Evaluation (Metadata)

- IO500 benchmark.



	BeeGFS (30)	Hercules (30)	BeeGFS (90)	Hercules (90)
find	1.056	8.120	8.088	23.538
mdtest-hard-write	31.062	34.565	92.322	73.179
mdtest-easy-stat	16.162	24.667	40.760	25.439
mdtest-hard-stat	9.860	8.332	32.482	22.165
mdtest-easy-delete	23.052	10.329	59.737	50.579
mdtest-hard-read	23.953	18.432	77.337	53.956
mdtest-hard-delete	14.648	19.887	48.321	60.104

Hands-on

- Multiple ways to deploy Hercules:
  - User level space

```
hercules start -s 0 -m /hercules/metadata -d /hercules/data -f
/hercules/conf/hercules.conf
```

  - Slurm

```
hercules start –f /hercules/conf/hercules.conf
```

  - Docker containers

Hands-on (Docker containers)

- Download images from DockerHub

```
docker pull arcosuc3m/hercules_server

docker pull arcosuc3m/hercules_client
```

- Running both data and metadata servers in the same container:

```
docker run –name hercules_server --network="host"

arcosuc3m/hercules_server
```

Share dynamic ports

Hands-on (Docker containers)

- Running some client commands

```
docker run -it --network="host"  arcosuc3m/hercules_client ls -l
/mnt/hercules
```

Mount point

```
docker run -it --network="host"  arcosuc3m/hercules_client ior -k -w -o
/mnt/hercules/test
```

Run IOR

```
docker run -it --network="host"  arcosuc3m/hercules_client ls -l
/mnt/hercules
```

```
docker run -it --network="host"  arcosuc3m/hercules_client ior -k -r -o
/mnt/hercules/test
```

Future work

- Malleability:
  - Current efforts by modifying existing pools for controlling data location.
- Monitoring
  - Performance metrics already gathered (i.e., memory bandwidth, network bandwidth).
- QoS
  - Degrade performance in presence of application computing peaks.
  - Memory usage.

uc3m | Universidad Carlos III de Madrid

Hercules: scalable and network portable in-memory ad-hoc file system for data-centric and high-performance applications

Javier Garcia-Blas and Jesus Carretero

University Carlos III of Madrid, Spain

fjblas@inf.uc3m.es

ADMIRE Users Day - Barcelona