Universidad Carlos III de Madrid
www.uc3m.es

ARCOS

The International Workshop On DISCS Data-Intensive Scalable Computing Systems

Experimental Evaluation of a Flexible I/O Architecture for Accelerating Workflow Engines in Cloud Environments

Francisco Rodrigo Duro, Javier García Blas
Florin Isaila, **Jesús Carretero**
Jesus.carretero@uc3m.es

Data-Intensive Scalable Computing Systems (DISCS-2015)
SC15 - November 15, 2015 – Austin, TX

Overview

- Introduction
- Hercules
- Evaluation
- Conclusions
- Future work

Introduction

- Classical scientific computing approach
  - CPU-intensive applications: massive simulations, complex mathematics, etc.
  - Parallel file systems for checkpointing
  - Specific platforms: supercomputers, clusters…
- However, due to massive data generation
  - Measurement instruments, social networks, multimedia, simulation results, etc.
- There is a shift towards data-intensive computing

Data-intensive computing

- New applications
  - Visualization, data analysis, predictions…
- New paradigms
  - Many task, big data, MapReduce, workflows
- Intermediary temporary data
  - *Workflow phases*
- I/O-bounded applications
- Heterogeneous platforms and cloud
  - Persistent shared storage
    - i.e. Amazon S3

New architectures need

- High-performance I/O systems
  - Scalable
  - Flexible
  - Elastic
- In-memory caching systems
  - Evolution in network technologies
  - High performance network + main memory
  - Integration of persistent memory
- To minimize data movement
  - Exposing and exploiting data locality

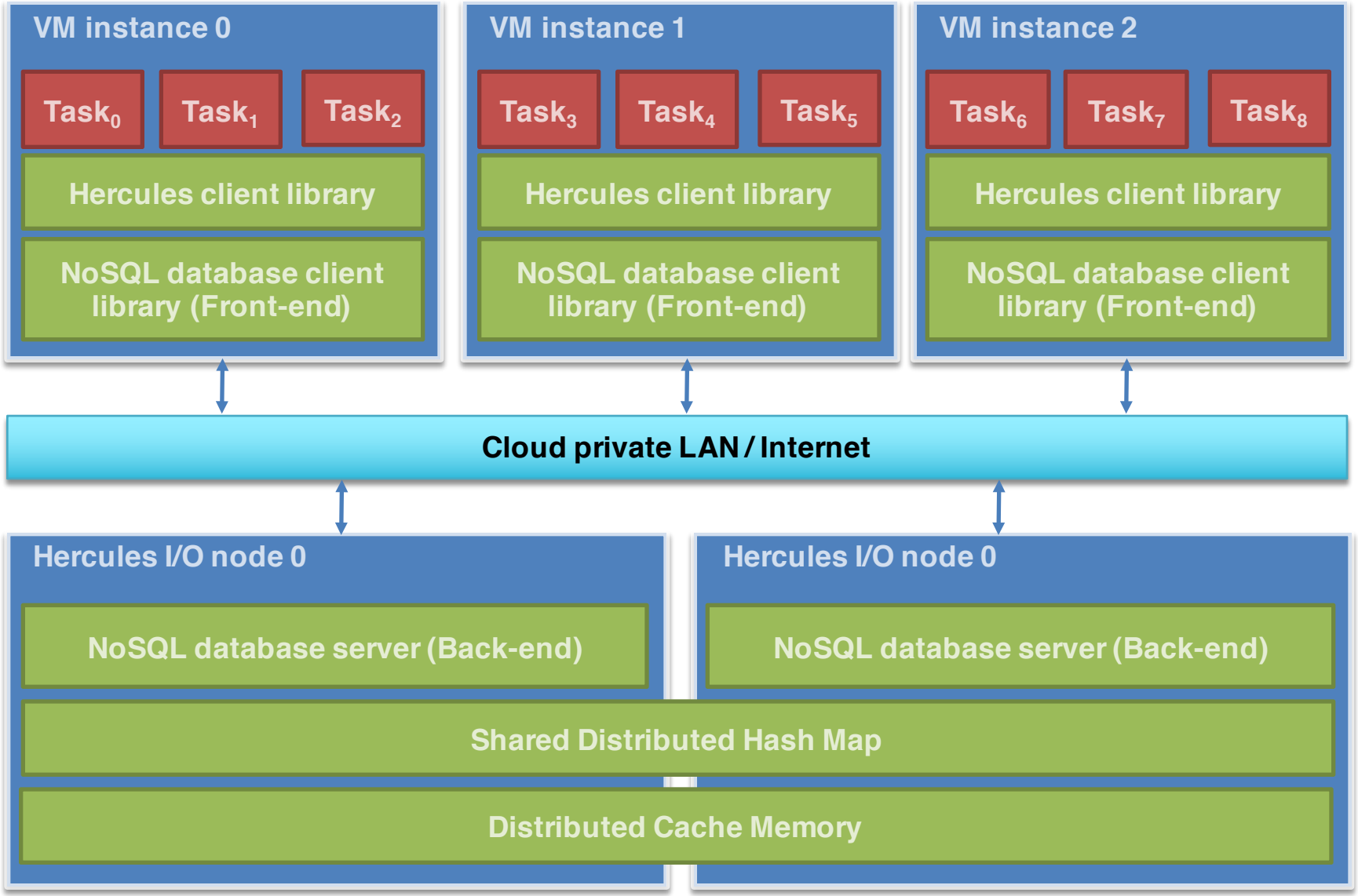
Hercules

- Distributed in-memory storage system
  - Configurable backend: Memcached, Redis
- Adaptable to clusters and cloud environments
  - Scalability
  - Flexibility
  - Easy deployment
  - High performance (memory + network)
- Architecture
  - User-level library + I/O server
  - Layered design

Motivation

- Avoid performance limits in storage services offered by the providers
- I/O performance isolation from peak loads produced by other concurrent applications
- Adapt the I/O infrastructure for the requirements of each specific application and/or experiment
- Minimize storage costs and temporary data movement
- Reduce effort for application portability
  - POSIX-like interface to applications

Hercules architecture



Hercules advantages (I)

- Scalability
  - Fully distributed data and metadata
  - Client-side calculated data placement
  - Fully stateless servers
- Easy deployment
  - No special permissions needed
    - User-level library and servers
  - Only needed information: servers IPs
  - Straightforward integration of persistent memory

Hercules advantages (II)

- Flexibility
  - POSIX-like interface
  - TCP-connections: public and private Ips
  - Different configurations for different experiments
    - As many I/O nodes as necessary
- Performance
  - Inter-node parallelism
    - More I/O nodes implies better bandwidth available
  - Intra-node parallelism
    - Memcached/Redis multithread implementation
    - Concurrent requests serving

Distributed NoSQL Systems

- Memcached
  - Distributed memory
  - User-level unified memory space
  - Fully in-memory (pure LRU)
    - No replication
    - No persistence
- Redis
  - Persistence
    - Snapshots or operations log
  - Replication
  - Memory reduction techniques
  - “Improved” Memcached
  - ¿Performance?

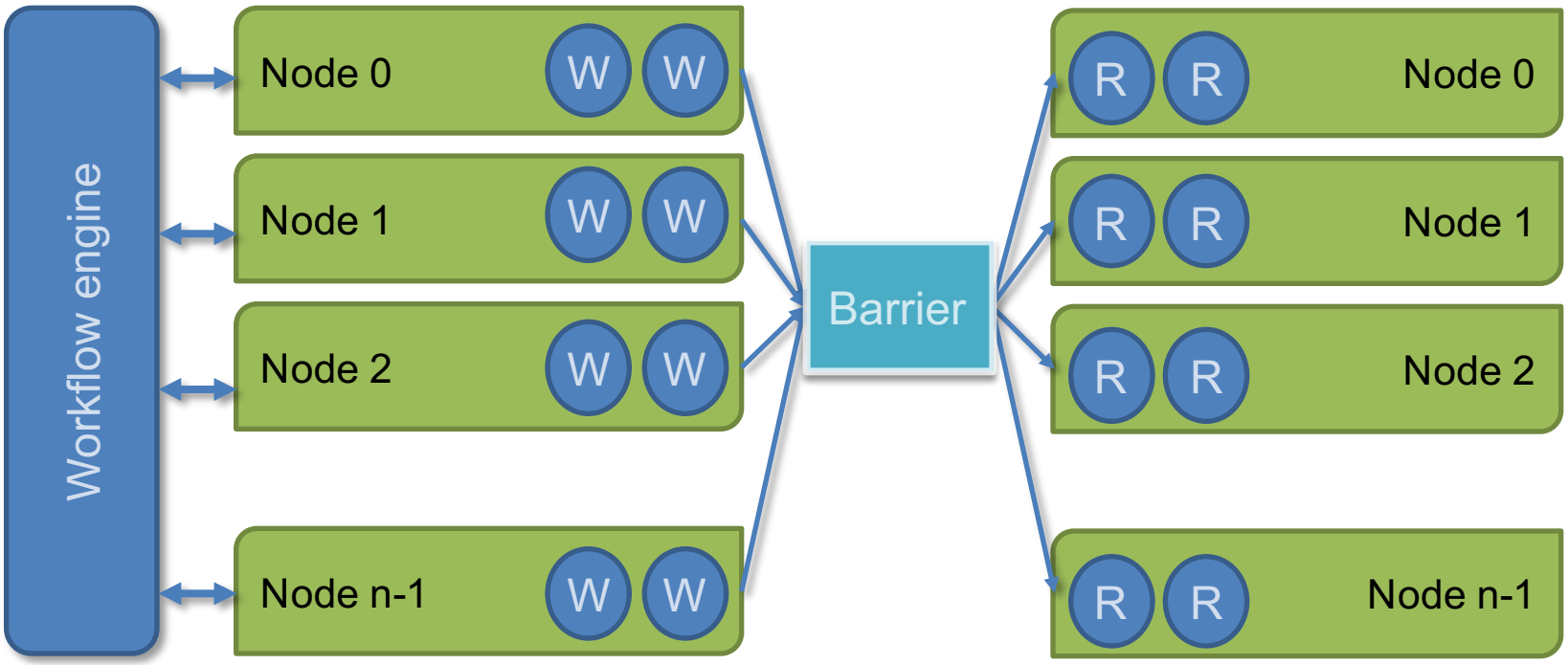
Evaluation environment

- Amazon EC2 deployment
- Compared with Amazon S3
  - Access through S3FS
  - FUSE based
- Up to 64 client nodes
  - m3.xlarge (4 cores / 15 GBytes DDR3)
- Up to 32 I/O server nodes
  - m3.2xlarge (8 core / 32 GBytes DDR3)
- Gigabit Ethernet

File copy micro-benchmark

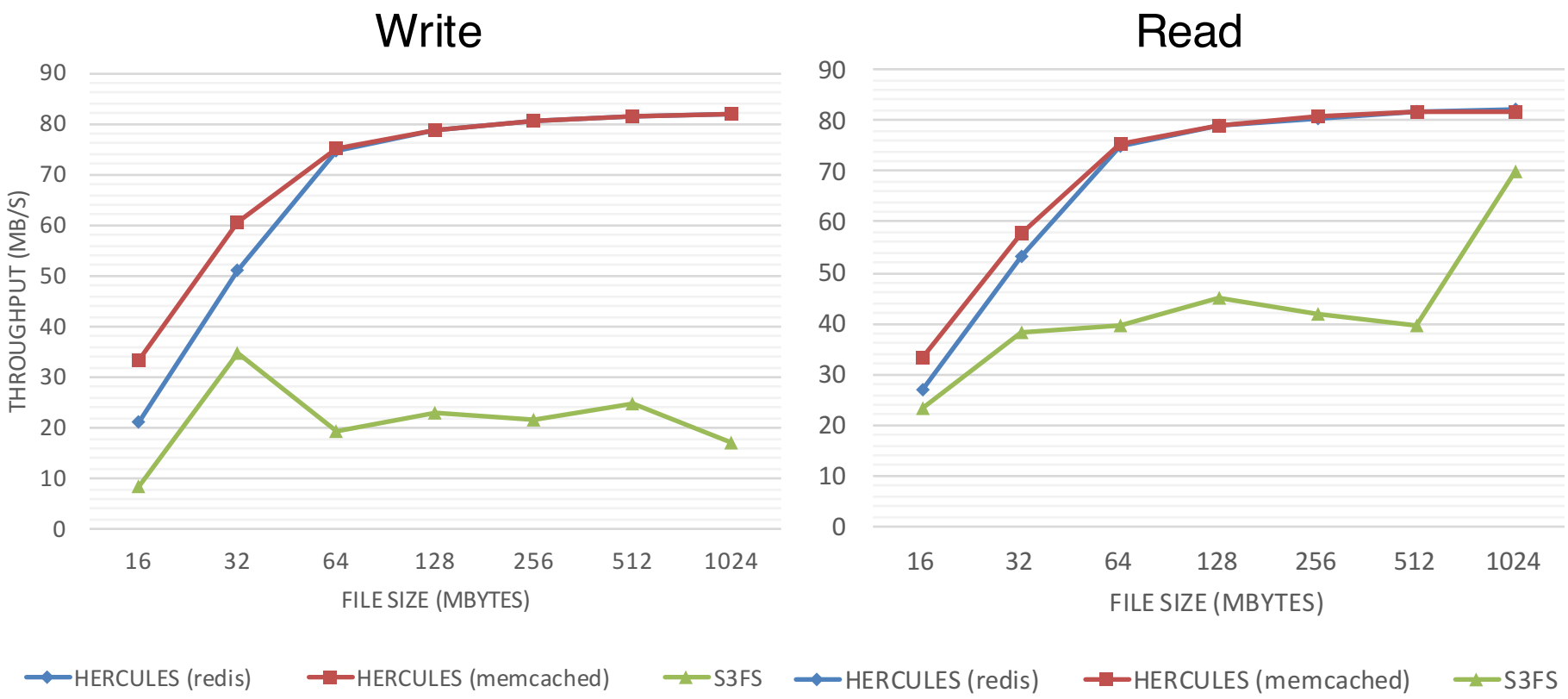
- Emulate typical workflow engine behavior
  - High load of I/O tasks
- *n* concurrent tasks write/read files
- Fully configurable
  - Number of worker nodes
  - Number of workers per node (ppn)
  - Number of tasks executed per worker
  - File size
  - *Chunk* size

File copy micro-benchmark workflow



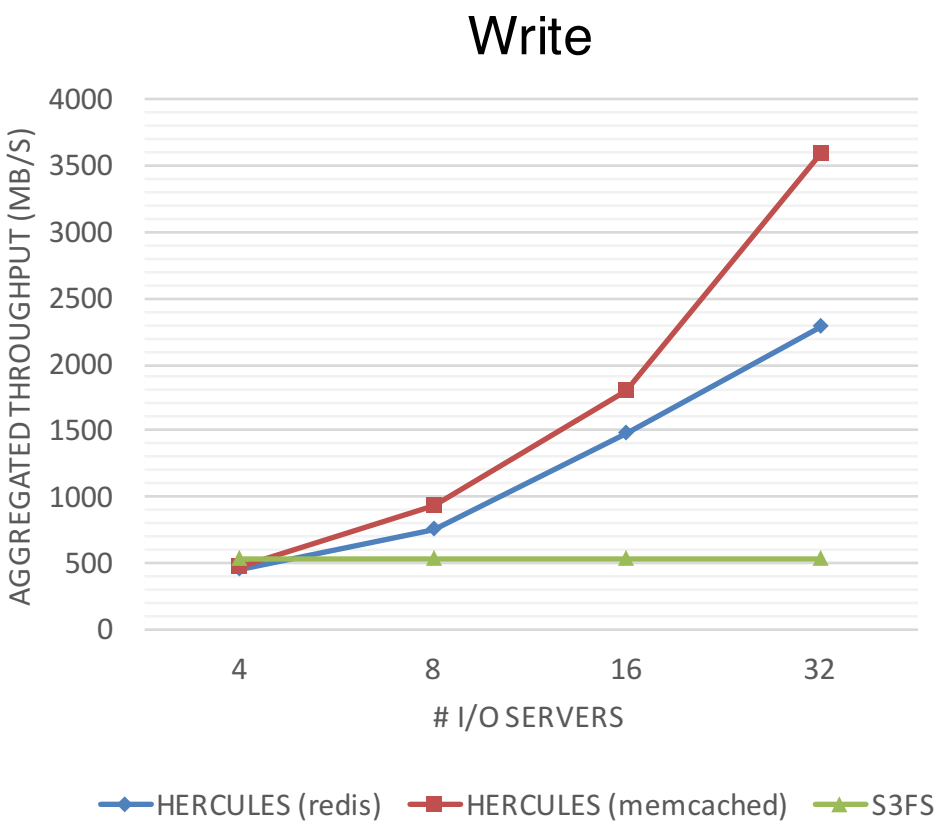
Scalability with File size

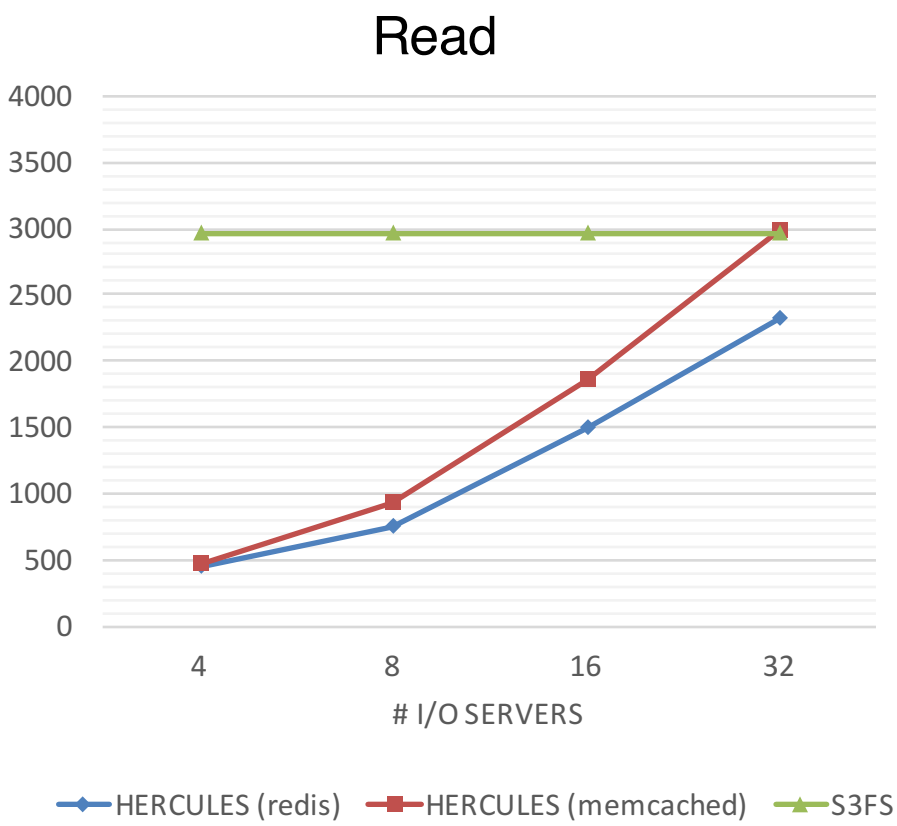
- 1 client and 1 I/O server
- File size: 16 MB – 1024 MB



Scalability with I/O server nodes

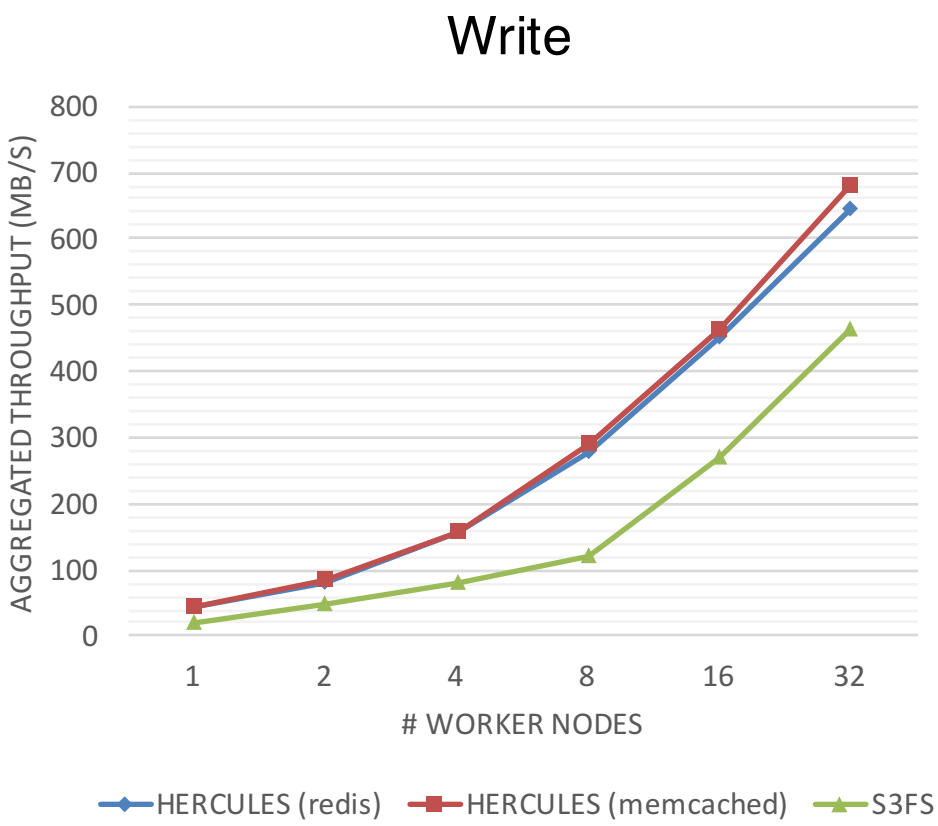
- I/O node servers scaled while maintaining the workload
  - 64 clients x 2 GB per file
  - 4 – 32 I/O servers

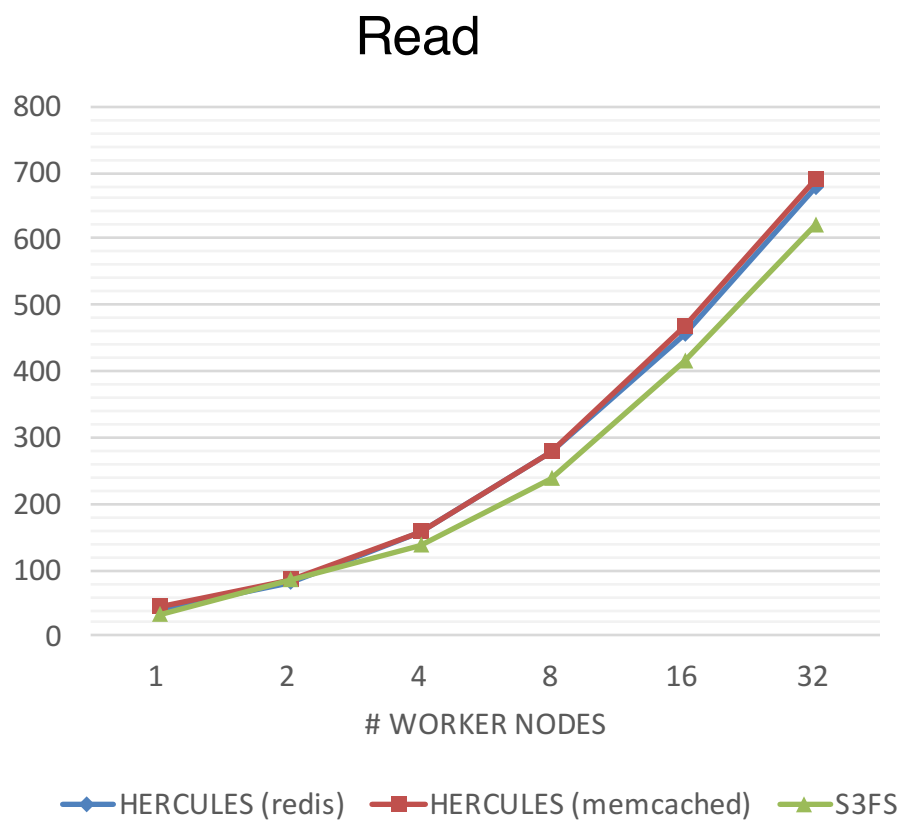




Scalability with small files

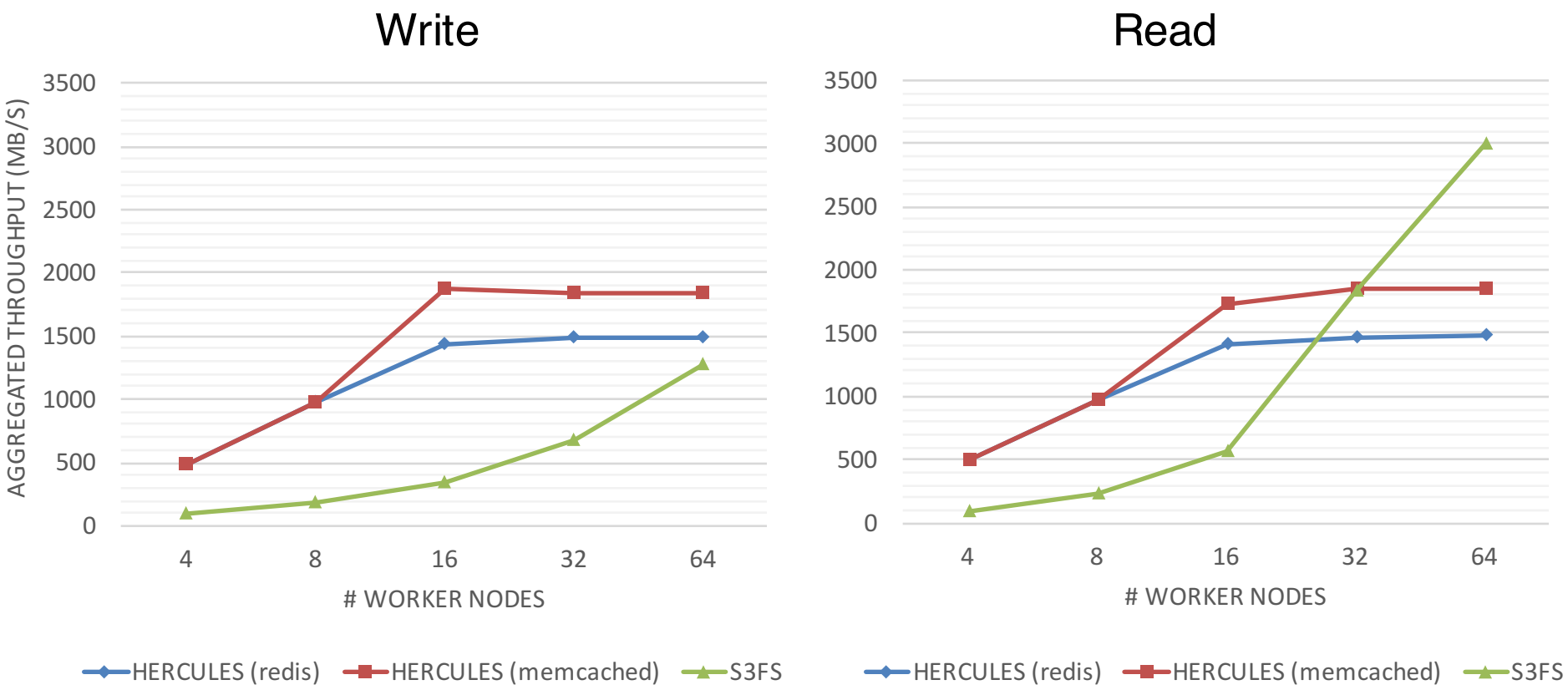
- Concurrent requests for large files
  - 32 I/O nodes
  - Up to 32 clients x 8 ppn x 16 MB





Performance with big data sets

- Concurrent requests for large files
  - 16 I/O nodes
  - Up to 64 clients x 1 ppn x 2048 MB



Conclusions

- Hercules proposal is feasible as shown in the evaluation
- Scalability
  - Good in every evaluated configuration
- High performance
  - Fully utilization of the available bandwidth
  - Greatly improves S3FS write performance
  - Competitive in read operations
- Flexible
  - Can choose persistency or not as needed
    - Pay for performance, not for persistency
- Adaptability to other platforms
    - Mobile Cloud Computing
    - HPC platforms
      - Swift/T (Argonne National Laboratory)
- Easy deployment in all environments

Future work

- High scalability evaluation
  - Investigate Hercules scalability limits
  - Investigate S3FS performance limits
- Compare with other solutions
  - Amazon Redshift
  - Amazon ElastiCache
- Evaluate more access patterns and/or real applications
  - And more cloud platforms (Azure WIP)
- Cost/performance analysis

Universidad Carlos III de Madrid
www.uc3m.es

ARCOS

Experimental Evaluation of a Flexible I/O Architecture for Accelerating Workflow Engines in Cloud Environments

Francisco Rodrigo Duro, Javier García Blas
Florin Isaila, **Jesús Carretero**

Jesus.carretero@uc3m.es

The International Workshop On DISCS Data-Intensive Scalable Computing Systems

Data-Intensive Scalable Computing Systems (DISCS-2015)
SC15 - November 15, 2015 – Austin, TX