Folding a Cluster containing a Distributed File-System

15th JLESC Workshop

Quentin GUILLOTEAU Olivier RICHARD Raphaël BLEUSE
Eric RUTTEN

Univ. Grenoble Alpes, INRIA, CNRS, LIG

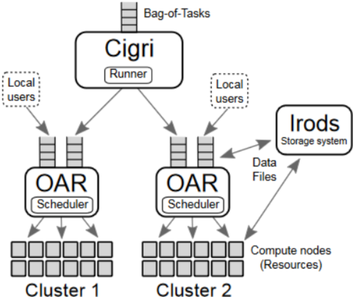
`firstname.lastname@inria.fr`

2023-03-22

HPC — Control Theory — Reproducibility with Nix(OS)

Looking for post-doc in 2024

Context - CiGri



CiGri: Grid middleware [3]

- Goal: use idle grid res.
- Interacts w/ several RJMS (≃ 100 nodes)
- Lowest prio ⇝ can be killed
- Periodic sub to RJMSs

- CiGri jobs can impact DFS perfs ⇝ disturb premium users ☹
- **Need for Regulation**: Sub size w.r.t. Load ⇝ Control Theory [5, 4]
- Need to deploy CiGri at large scale for **representative** experiments

$\Longrightarrow$ **Full scale experiments are too costly**

Context - Problem

How to reduce the number of nodes to deploy while keeping a representative environment?

Potential solutions

↪ Simulation:

- + needs only one node/core
- + fast
- + can represent any cluster
- - **modeled system/env**

↪ Reduce number of nodes:

- + **real system/environment**
- + less resources deployed
- - not representative...
- - real time

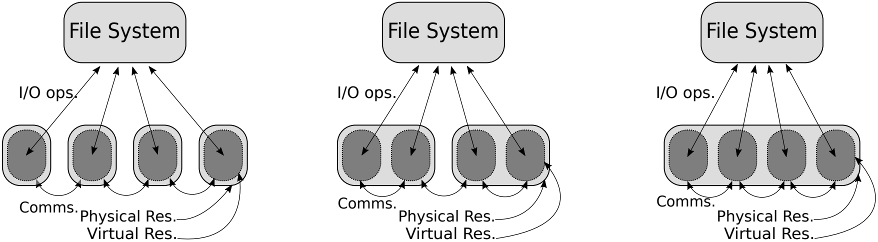
Software & Hardware stacks are **too complex to model**

$\Longrightarrow$ We need an intermediate solution

What we *really* want: less resources deployed & real system/environment

Folding

The idea: Deploy more "virtual" resources on one physical resource



(a) Folding w/ factor 1.

(b) Folding w/ factor 2.

(c) Folding w/ factor 4.

+ less resources deployed ☺

+ represents full scale system

+ real system/environment ☺

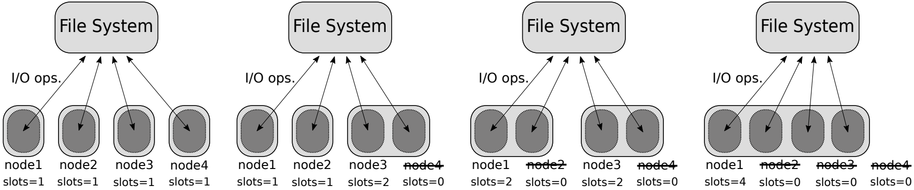
- new job model ⇝ `sleep + dd`

↪ **But, does folding introduce noise to the experiment?**

Protocol

Objective: **Evaluate performances of an I/O app w.r.t. folding**

- Benchmark: IOR [7] (MPI I/O benchmark)
- Platform: Grid'5000 [1] (`gros` cluster[1]), *NixOS Compose* [6]
- Several File-systems: NFS [8] (DFS) & OrangeFS [2] (PFS)



(a) Initial deploy (full scale)

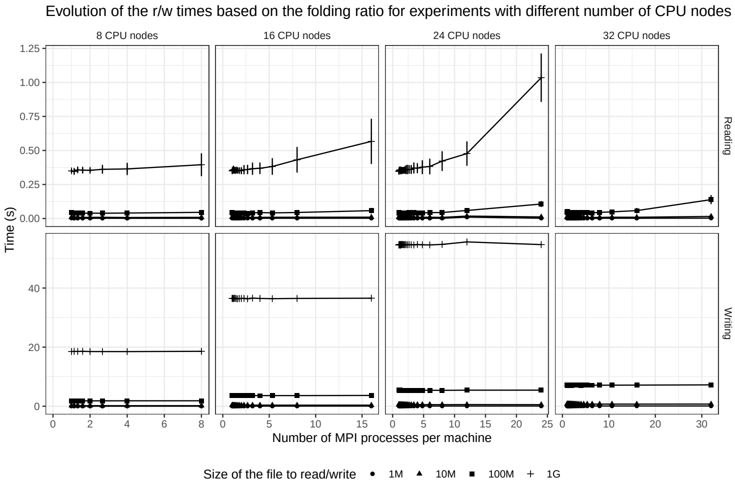
(b) Remove `node4`, add `slot` to `node3`.

(c) Remove `node2`, add `slot` to `node1`.

(d) Remove `node3`, fully folded

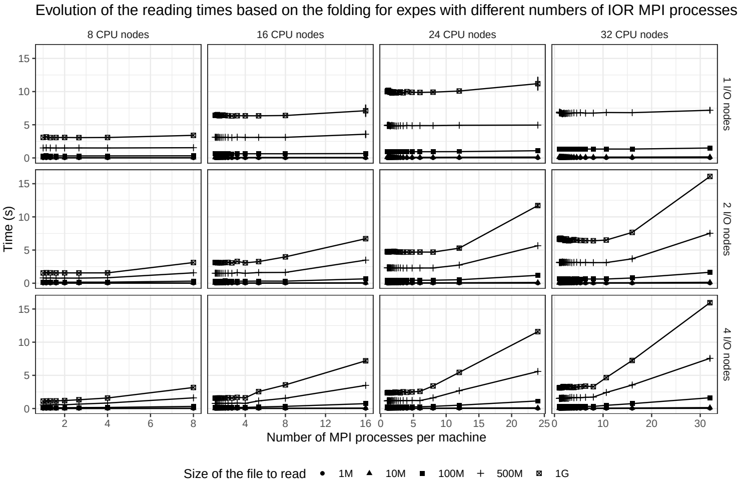
[1] Intel Xeon Gold 5220 CPU w/ 18 cores, 96 GiB of memory, a 2 x 25 Gbps (SR-IOV) network and a 480 GB SSD SATA Micron MTFDDAK480TDN disk

Preliminary Results - NFS



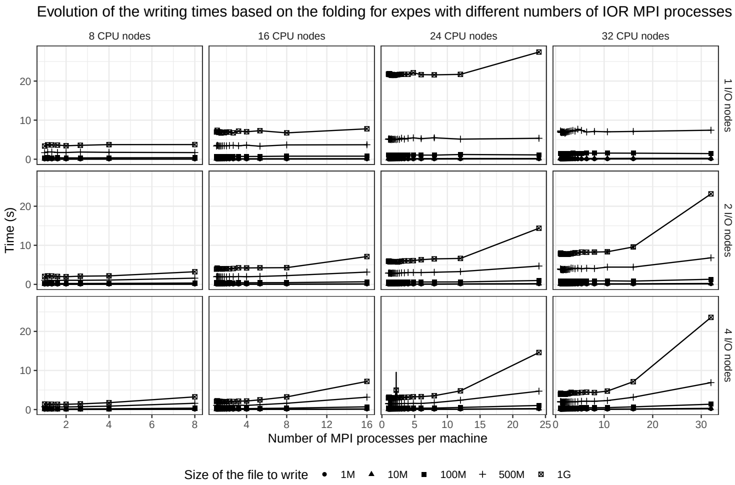
- ↪ Write operations (bottom) **do no seem affected** by folding
- ↪ Read operations (top) are affected by folding ⇝ **quadratic model**

Preliminary Results - OrangeFS - Reading



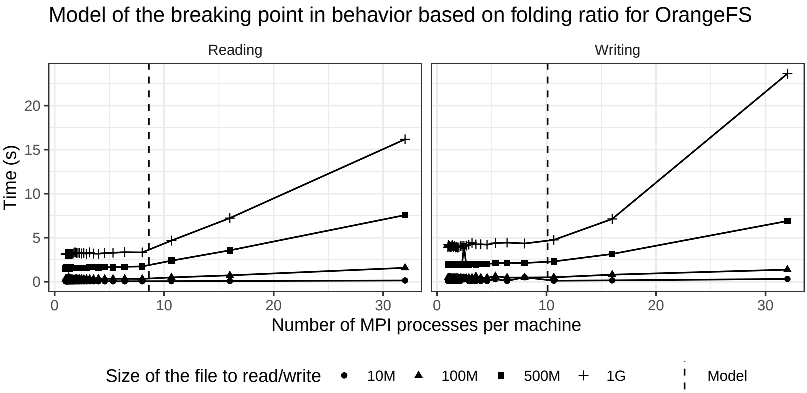
- ↪ **Breaking point in behavior** ⇝ model
- ↪ Performances of fully folded do not depend on number of I/O nodes

Preliminary Results - OrangeFS - Writing



- ↪ Affected by folding (≠ NFS)
- ↪ Also **Breaking point in behavior** ⇝ model

Preliminary Results - OrangeFS - Breaking point Model



Rule of thumb

$$f_{break} \simeq 1 + 0.3 \times nb_{cpu} - 0.5 \times nb_{io}$$

Conclusion & Perspectives

Objective

Investigate the impact of folding on an app. doing I/O on a DFS

Results

- **Folding appropriate until a breaking point**
- Model for overhead in reading time for NFS
- Rule of thumb for the breaking point in behavior for OrangeFS

Perspectives

- Evaluate more popular PFS (Lustre, BeeGFS, etc.)
- Evaluate more Networks (InfiniBand, OmniPath, etc.)

↪ Working document: `https://hal.science/hal-04038000`

References I

[1] D. Balouek et al., "Adding virtualization capabilities to the Grid'5000 testbed," in *Cloud computing and services science*, Vol. 367, edited by I. I. Ivanov et al., Communications in Computer and Information Science (Springer International Publishing, 2013), pp. 3–20.

[2] M. M. D. Bonnie et al., "Orangefs: advancing pvfs," in Usenix conference on file and storage technologies (fast) (2011).

[3] Y. Georgiou et al., "Evaluations of the lightweight grid cigri upon the grid5000 platform," in Third ieee international conference on e-science and grid computing (e-science 2007) (IEEE, 2007), pp. 279–286.

[4] Q. Guilloteau et al., "Controlling the Injection of Best-Effort Tasks to Harvest Idle Computing Grid Resources," in ICSTCC 2021 - 25th International Conference on System Theory, Control and Computing (Oct. 2021), pp. 1–6.

References II

[5] Q. Guilloteau et al., "Model-free control for resource harvesting in computing grids," in Conference on Control Technology and Applications, CCTA 2022 (Aug. 2022).

[6] Q. Guilloteau et al., "Painless Transposition of Reproducible Distributed Environments with NixOS Compose," in CLUSTER 2022 - IEEE International Conference on Cluster Computing, Vol. CLUSTER 2022 - IEEE International Conference on Cluster Computing (Sept. 2022), pp. 1–12.

[7] *Ior benchmark*, `https://github.com/hpc/ior`, Accessed: 2023-01-19.

[8] B. Pawlowski et al., "The nfs version 4 protocol," in In proceedings of the 2nd international system administration and networking conference (sane 2000 (Citeseer, 2000).