The NEXTGenIO project has received funding from the European Union's Horizon 2020 Research and Innovation programme under Grant Agreement no. 671951

Data Scheduling in NEXTGenIO

Toni Cortes, Ramon Nou*, Alberto Miranda

Bordeaux, Oct. 2018

Team overview

- Two main topics
  - HPC Storage
    - Toni Cortes, Ramon Nou and Alberto Miranda
    - Data-aware scheduling
    - Ephemeral filesystems (per job / per application)
  - Data Sharing - dataClay team
    - Toni Cortes, Anna Queralt and Jonathan Marti
    - Object Store with execution capabilities.

EU Projects

- **NEXTGenIO** (M36 of M48)
    - NVDIMM based hardware
    - Working on:
        - data scheduling
        - *echofs* (ephemeral local filesystem)
        - *gekkofs* (ephemeral distributed filesystem collaboration with André Brinkmann)
- **BigStorage** (MCITN) and **Expertise** (MCITN)
- **Class** (Internet Of Things)
- **Elastic** (Fog Computing) and **mF2C** (Fog Computing)

SLURM (Workflow Support)

- Workflows enhancements (1-to-1, do **we need N-to-N**)?
  - New options for srun/sbatch/salloc (--workflow-start, --workflow-end, workflow-parent=JOB_ID);
  - New options for sstat tools (--workflows);
  - New internal data structures;
  - New priority plugin;
  - New scheduling plugin/algorithm;
  - Parent/child relationships between jobs in the same workflow;
  - Priority of workflow jobs is updated during Scheduling runs;
  - Child workflow jobs are held and will not run before their parent has completed running.

SLURM (Same node Scheduling)

- Same node scheduling
  - Can be requested by the user (using a new option);
  - Can be initiated by the Scheduler;
  - New options for srun/sbatch/salloc (--request-same-node);
  - Updated Job and Node scheduling algorithm.
- Logic:
  - Increasing number of nodes: Reuse the existing ones and add more nodes;
  - Same number of nodes: Reuse the same nodes;
  - Decreasing number of nodes: Use some (the best) of the existing nodes.

SLURM (Storage based Scheduling)

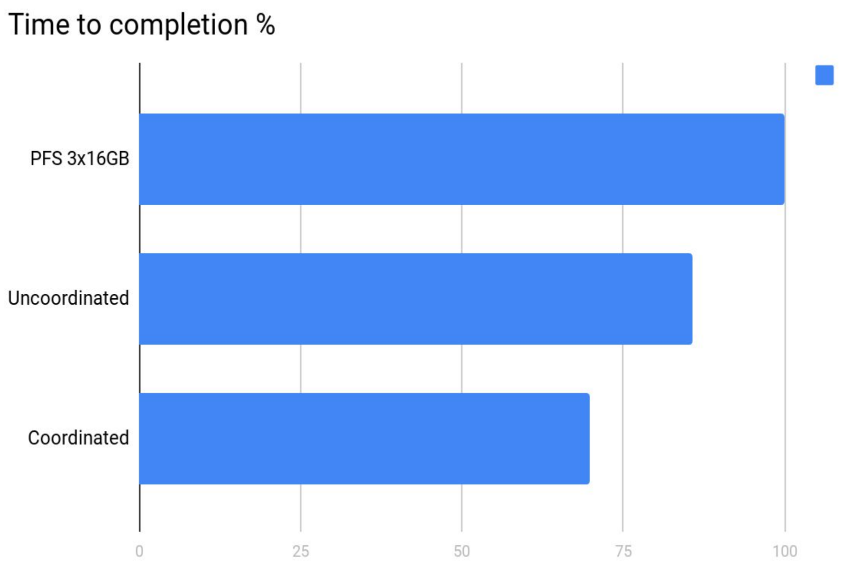
- Storage based scheduling
  - Users can specify storage requirements of their job and the Scheduler will only use nodes that satisfy their specs;
  - New options for srun/sbatch/salloc (--storage-type (IN/OUT), --storage-size=,--storage-location=);
  - Updated Job and Nodes data structures;
  - Updated Job and Node scheduling algorithm.

On-going work

- data scheduling (norns)
  - Coordinate data-movements that may have an impact on the system
    - Node -> PFS || PFS -> Node || Node -> Node
  - Using SLURM and user library as interface
  - Using PFS / system information
    - How many I/O is happening in the system?
    - Will the performance decrease if we add more I/O?
  - Able to register different **backends** and issue **async** transfer requests.
    - i.e. [tmp0://POSIX, 2GiB] [pmem0://, PMDK, 500 GiB]
    - "User" can provide "stream" and "stream iterators"
  - 1 control daemon per node

On-going work

- DS Coordination effect
  - Reading - writing 3x16GB files
    - Directly (at once)
    - With stage-in - stage-out (at once)
    - Coordinated in-out
  - Eventually will look for PFS status and decide if we can issue a transfer or not.



On-going work

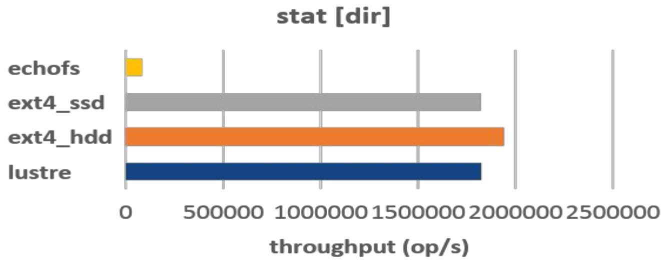
- echofs [1.0 released]
  - Ephemeral file system per app using local storage
  - Local - FUSE based (99% POSIX compatibility)
  - Reduce POSIX features not used to increase performance
  - **Stage-in / Stage-out support (also with norns)**
  - Working on NVDIMM
  - Metadata in memory => Removes PFS Overhead
  - **Issues**: FUSE Overhead, but still some scenarios are promising (and this is an app oriented filesystem).

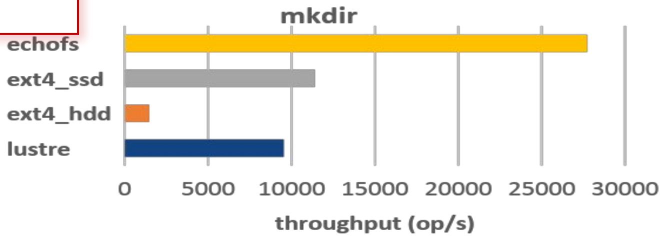
echofs [metadata performance]

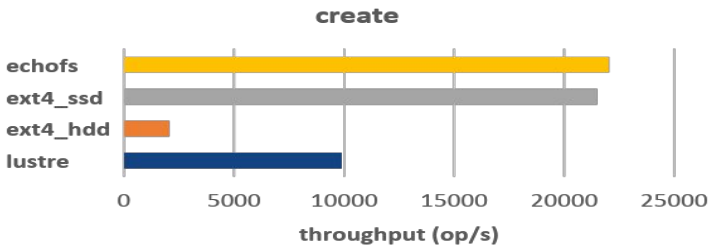
stats are slow

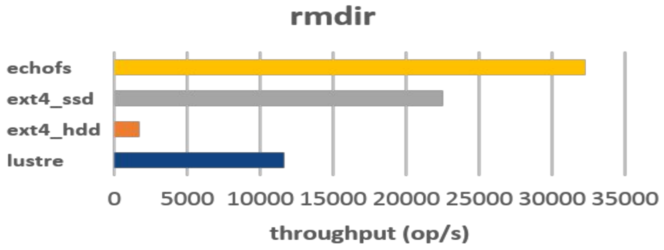
(FUSE's fault)

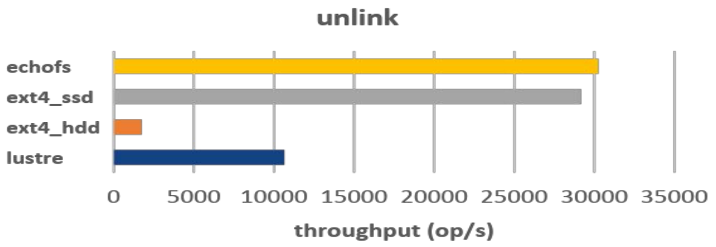
other metadata operations are quite good







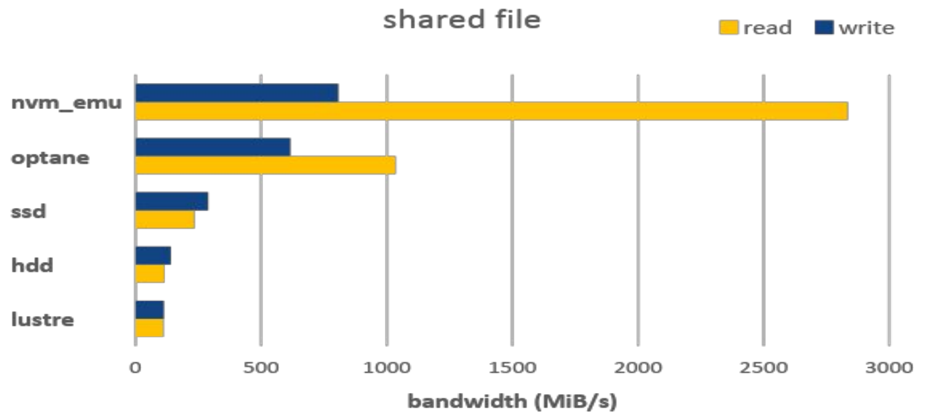




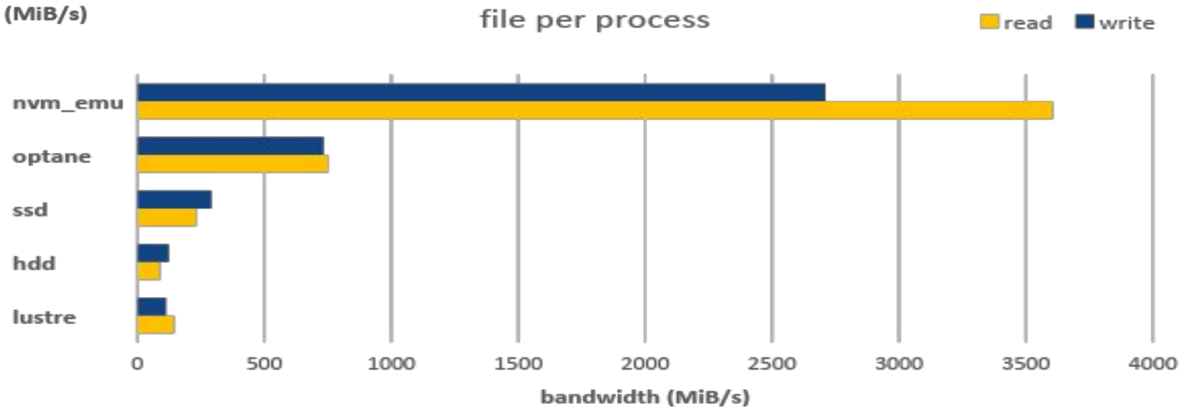
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

nextgenio

echofs [data performance]



IOR microbenchmark
1GiB data (single/multiple files)
128KiB transfer size
4 concurrent processes
10 iterations



despite FUSE's limitations, data performance is significantly better than accessing Lustre

On-going work

- gekkofs [JGU - André Brinkmann’s team]
    - Ephemeral **distributed** file system per app using local storage
    - LDPRELOAD based (low POSIX compatibility)
        - Using pmem syscall_intercept library
    - Adding system calls to increase compatibility
    - Adding NVDIMM backend
    - Metadata and data performance increases by the number of nodes provided.
    - Issues: **LDPRELOAD**, apps, forks, links becomes hard to solve.

Ideas...etc

- FUSE optimization
  - Create the next FUSE-like framework usable on faster devices
- Applications
  - Lots of solutions, but small number of applications
    - Trying OpenFOAM, moving to CASTEP.
- Pattern recognition in I/O
  - Recognition and prediction solved.
  - Need to find places to tune I/O. Working with INRIA Grenoble (JLESC)
- Open to ideas
  - Our team has experience in different layers and topics

BSC Barcelona Supercomputing Center
Centro Nacional de Supercomputación

EXCELENCIA SEVERO OCHOA

THANK YOU!

ramon.nou@bsc.es

toni.cortes@bsc.es

www.bsc.es