IO500

George Markomanolis
IO500 Committee: John Bent, Julian M. Kunkel, Jay Lofstead

2017-11-12

http://www.io500.org

IBM Spectrum Scale User Group, Denver, Colorado, USA

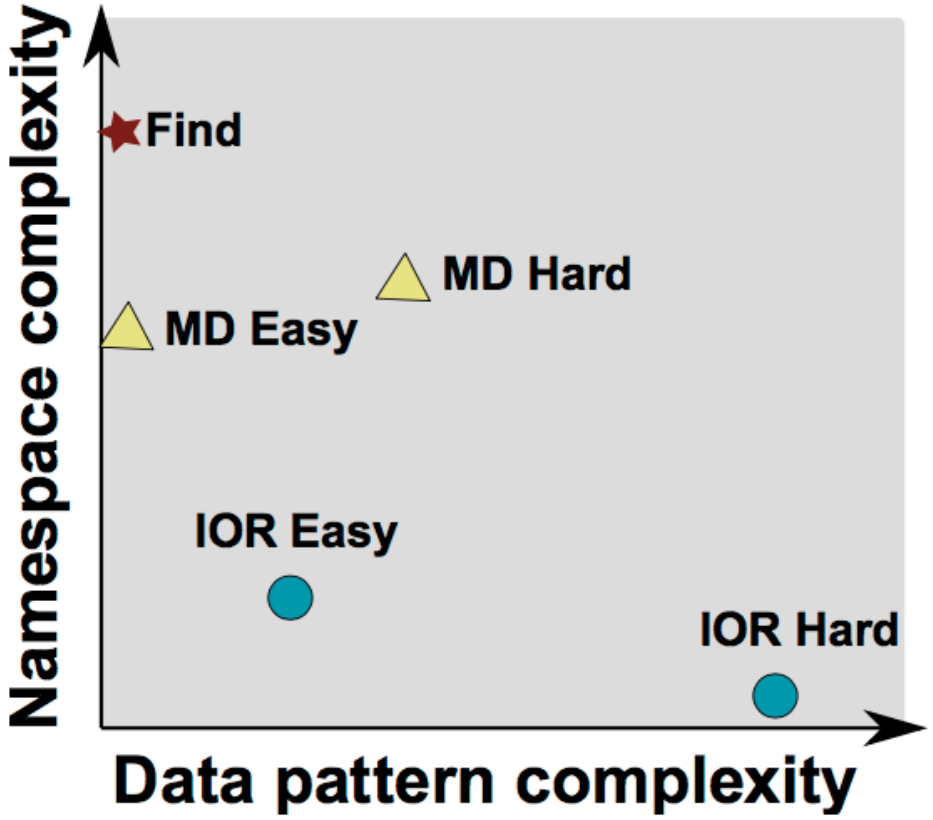
Why?

- The increase of the studied domains, lead to larger data output, thus more stress on filesystem
- Customers buy a storage only by evaluating the max GB/s achieved by IOR, while many real applications can not achieve similar performance
- The I/O efficiency can be downgraded by interference with multiple users
- A real case, commercial application using one node, was consuming more than 15% of the overall metadata capacity
- We need a suite of benchmarks in order to understand what are the real performance expectations
- Tracking storage performance and sharing best practices

How?

- Community driven effort, discussing through mailing list, Slack etc. Everything is in github (https://github.com/VI4IO/io-500-dev.git)
- Patterns: metadata, data, search
  - Easy for optimized patterns
  - Hard for naïve patterns
- Relies on community benchmarks, such as IOR, mdtest (for now)

What is IO-500?



IOR Easy: This is what is used during the procurements, where we measure the most efficient I/O pattern, user can declared the parameters and we save one file per MPI process

IOR Hard: Single-shared file, 47008 byte random access, POSIX

MD Easy: Create rank directories with N empty files

MD Hard: Single shared directory, files of 3901 bytes, POSIX

Find: Find functionality searches for files of 3901 bytes across all the created files. Sven added the mmfind.sh script for Spectrum scale environment (**io-500-dev**/utilities/find/**mmfind.sh)**

Challenges & Approach I

- Representative of applications and user requirements
- Using different workloads for extracting upper and lower performance in the cases of optimized and non-optimized application respectively
- Report meaningful metrics
- Implement a find functionality (we tried 3 different versions)
- Libcircle is used by parallel find and it is not friendly with machines which do not provide the wrapper mpicc, problem is solved with some manual modifications

Challenges & Approach II

- Concurrent runs to be integrated, already initial tests provide interesting results
- 5 minutes limit per experiment to avoid long runs
- Extended IOR/mdtest for phase-out stonewalling options
- Easy to build, less than 70 seconds for the basic version to be installed

How to run IO-500

- git clone https://github.com/VI4IO/io-500-dev
- cd io-500-dev
- ./utilities/prepare.sh
- ./io500.sh (submit this script if you use a scheduler)
- email results to submit@io500.org

Demo installation of IO500

Modify IO-500

- Modify io500.sh accordingly, for example:

```
io500_mpirun="mpirun"
io500_mpiargs="-np 2"
io500_ior_easy_params="-t 2048k -b 2g -F"
io500_mdtest_easy_files_per_proc=25000
```

Modify IO-500 II

- Modify io500.sh accordingly, select which experiments to be executed:

```
io500_run_ior_easy="True"
io500_run_md_easy="True "
…
io500_run_md_hard_delete="True"
```

- For **valid** submission, you need to execute all the tests while the write phases should take at least 5 minutes

Modify IO-500 III

- Modify io500.sh accordingly, uncomment these lines and declare the path to your pfind wrapper:

```
#io500_find_mpi="True"
#io500_find_cmd="$PWD/bin/pfind"
```

Example of a test case

```
[RESULT] BW   phase 1          ior_easy_write          96.133 GB/s : time 187.24 seconds
[RESULT] BW   phase 2          ior_hard_write          11.230 GB/s : time  46.79 seconds
[RESULT] BW   phase 3          ior_easy_read          109.249 GB/s : time 164.76 seconds
[RESULT] BW   phase 4          ior_hard_read            7.871 GB/s : time  66.74 seconds
[RESULT] IOPS phase 1       mdtest_easy_write           49.231 kiops : time  19.61 seconds
[RESULT] IOPS phase 2       mdtest_hard_write           15.444 kiops : time  17.05 seconds
[RESULT] IOPS phase 3             find                  8.120 kiops : time  98.45 seconds
[RESULT] IOPS phase 5       mdtest_easy_stat            5.313 kiops : time 127.18 seconds
[RESULT] IOPS phase 6       mdtest_hard_stat            6.772 kiops : time  30.43 seconds
[RESULT] IOPS phase 7       mdtest_easy_delete          14.873 kiops : time  49.98 seconds
[RESULT] IOPS phase 8       mdtest_hard_read            45.599 kiops : time  10.16 seconds
[RESULT] IOPS phase 9       mdtest_hard_delete          30.776 kiops : time  11.84 seconds
[SCORE] Bandwidth 31.04 GB/s : IOPS 16.1537 kiops : TOTAL 501.4108
```

Experience with IO500 benchmark

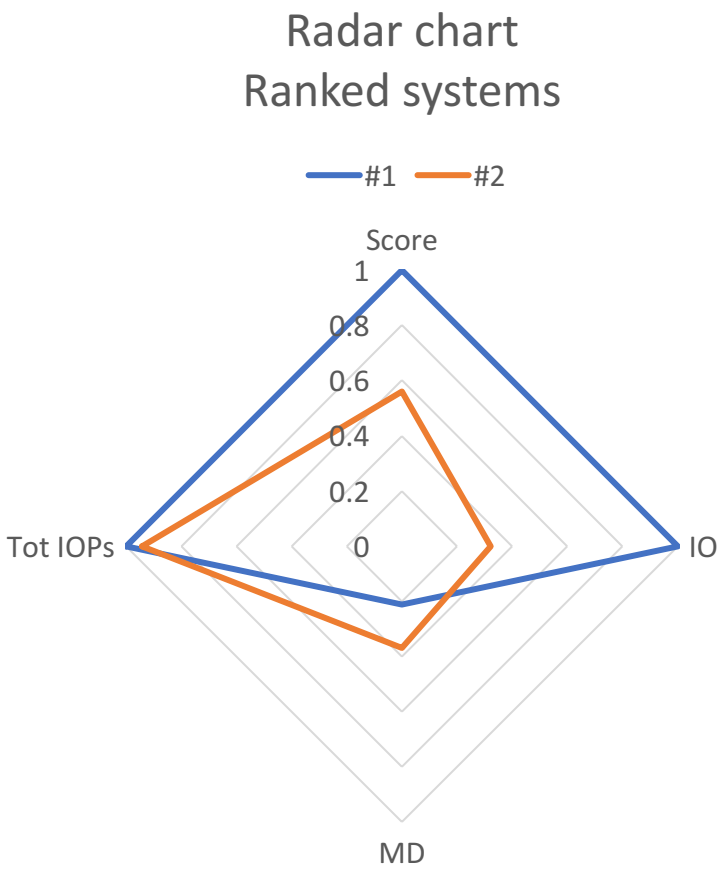
- With not proper tuning, the benchmark will finish either too fast or too slow
- Start tuning with small values and increase them till you find the ones that produce the required outcome
- Be sure that you have enough space for the output data
- Check form the IOR output if it recognizes correctly the number of processes and how many are used per node
- If the benchmark is too slow without reason, check if other users execute intensive I/O applications
- Be sure that you do not harm the system, try to execute the benchmark when the system is not too busy or during maintenance
- For the IOR Hard, you could stripe the corresponding folder

KAUST – Cray DataWarp – IO-500

- 300 compute nodes, 2400 processes, 268 DataWarp nodes

- ior_easy_params="-t 2m -b 192616m”
- ior_hard_writes_per_proc=77872
- mdtest_hard_files_per_proc=1630
- mdtest_easy_files_per_proc=10800

Presenting data in radar chart



The best storage I/O system should be represented in a full diamond graph

NASA - IOPS Galore Encore

IOPS Galore Encore: Upgrading a Supercomputer's Metadata with Non-Volatile Memory

Overview

The most recent wave of enhancements to the NASA Center for Climate Simulation (NCCS) Discover supercomputing cluster included an upgrade of the metadata storage. This has significantly enlarged the metadata capacity of the Discover cluster's storage; increased user and administrator abilities in metadata operations, including input/output operations per second (IOPS); and introduced new technology into the General Parallel File System (GPFS) cluster.

Some GPFS systems are in the first IO500 list which will be presented on Wednesday at IO500 BOF.

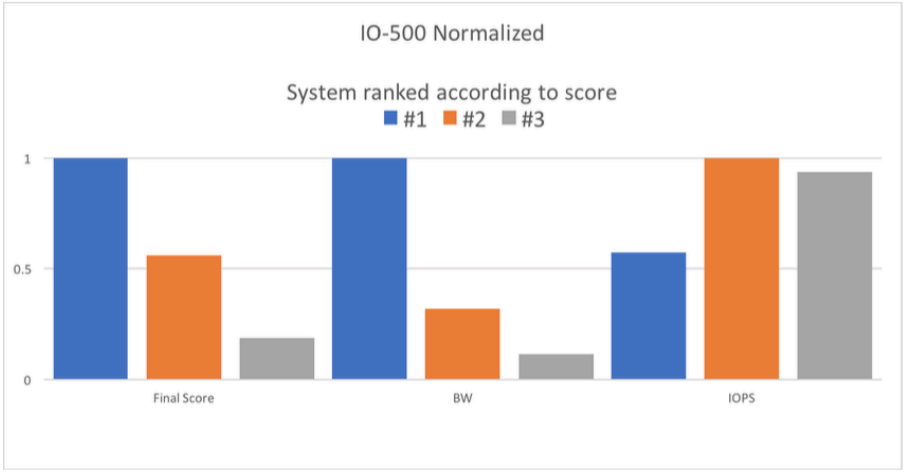
Would you be interested in providing new results?

Conclusions

- Till now the IOR easy is considered the normal approach for procurement, however, this does not correspond to the real application
- We need a better way to understand the procurement of storage and IO500 seems to be in the right direction
- A customer can conclude to decisions based on his application requirements
- We plan some future additions, such as mix workload
- More submissions we have, the better to understand the various filesystems

You are welcome to IO500 BOF!

IO500



Getting Stared with IO500

- git clone https://github.com/VI4IO/io-500-dev
- cd io-500-dev
- ./utilities/prepare.sh
- ./io500.sh
- # Tune and rerun until good
- # email results to submit@io500.org

Contact us

http://www.io500.org
Slack: vi4io.slack.com
Twitter: IO500benchmark

Come see the full IO-500 results at SC17 BOF
Wednesday, 15 November, 17:15, room 201-203

"Results from ThinkParQ BeeGFS, Cray DataWarp, DDN IME, IBM Spectrum Scale, and Lustre!"