I/O-500 Status

Julian M. Kunkel[1], Jay Lofstead[2], John Bent[3], George S. Markomanolis[4]

1. Deutsches Klimarechenzentrum GmbH (DKRZ)
2. Sandia National Laboratory
3. Seagate Government Solutions
4. KAUST Supercomputing Core Lab

2017-06-25

The Virtual Institute for I/O

Goals of the Virtual Institute for I/O

- Provide a platform for I/O researchers and enthusiasts for exchanging information
  - Offers info pages for each group
  - Lists information about existing I/O benchmarks together with samples
- Foster training and international collaboration in the field of high-performance I/O
  - Support the community to establish conventions and standards
- Track and encourage the deployment of large storage systems by hosting information about high-performance storage systems

https://www.vi4io.org

Goals of the IO-500 Benchmarking Effort

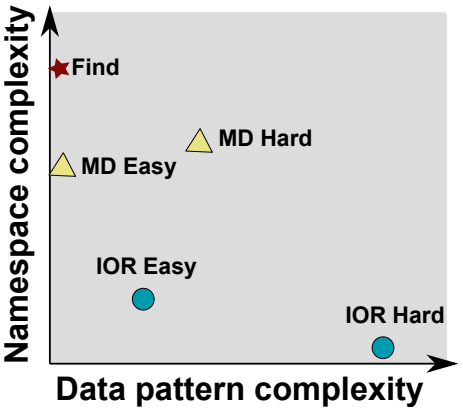
- Foster paradigm shift from compute centric perspective to I/O
- Bound performance expectations for realistic workloads
- Track storage system characteristics behavior over the years
  - Foster understanding of storage performance development
  - Support to identify potent architectures for certain workloads
- Document and share best practices
  - Tuning of the system is encouraged
  - Submitters must submit detailed run parameters
- Support procurements, administrators and users

IO-500 Requirements

Requirements of the benchmarking

- **Representative**: for optimized or naive workloads
  - Describe the natural requirements for users
  - IO-easy: upper bound for optimized IO-heavy workloads
  - IO-hard: expected performance for non-optimized applications
  - MD-easy, MD-hard: likewise but cover small-objects/metadata
- **Inclusive**: cover various storage technology and non-POSIX APIs
  - At best: useful for HPC and Big Data workloads
- **Trustworthy**: representative results and prevent cheating
- **Cheap**: easy to run and short benchmarking time (in the order of minutes)

Covered Access Patterns



- IOR-easy: optimal (large sequential) performance on POSIX files
- IOR-hard: small random performance on a shared POSIX file
- MD-easy: mdtest, per rank directory, with empty files
- MD-hard: more complex metadata operations on 3900 byte files
- find: query and filter files based on name and creation time
- Executing different patterns currently not covered (another dimension)

Benchmarking Phases

1. Create
   1. IOR-easy write
   2. IOR-hard write
   3. MD-easy create
   4. MD-hard create
2. Access
   5. IOR-easy read (ranks shifted!)
   6. MD-easy stat
   7. IOR-hard read (ranks shifted!)
   8. MD-hard stat
   9. find files
3. Cleanup
   10. MD-easy remove
   11. MD-hard remove

Fixed parameters used:

```
ior_easy="-C -Q 1 -g -G 27 -k -vv -e -F $ior_easy_params \\
-o $workdir/ior_easy/ior_file_easy"
# -W (validation) NOT for testing runtime

ior_hard="-C -Q 1 -g -G 27 -k -vv -e -t 47000 -b 47000 \\
-s $ior_hard_writes_per_proc -o ${workdir}/ior_hard/IOR_file"
# -W (validation) NOT for testing runtime

md_easy="-v -u -L -F -u -n $mdtest_easy_files_per_proc \\
-d ${workdir}/mdt_easy"

md_hard="-t -F -w 3900 -e 3900 -n $mdtest_hard_files_per_proc \\
-d ${workdir}/mdt_hard"
```

Benchmarking Rules

Parameterization

- Users may change / provide additional parameters to tune the easy cases
- Runtime for write/create phase at least 5 minutes
  - An autotuning script to determine parameters is provided
- All tests must use the same number of ranks/nodes
- No other changes to IO-500 script allowed
- Proposed usage:
  - Create a batch script that sets tunable parameters
  - It may precreate directory options to optimize (e.g., striping settings)
  - Source the io-500 script to have benchmarks run

Benchmarking Rules

Submission

- One submission per configuration allowed !
- Provide the starter (batch) script
- Submit all data describing the supercomputer, storage and run via VI4IO
- Results are immediately visible on the high-performance storage list
  - HPSL: community-managed list tracking many (theoretic) characteristics
  - `http://www.vi4io.org/hpsl/`
- The IO-500 steering committee will periodically curate the results
  - Release them on `https://io500.org`

Resulting Metrics

- For initial ranking: compute geometric mean for IOR and MD
  - Initial ranking: geom_mean(IOR) · geom_mean(MD)
  - Alternative: convert IOR TP to IOOPs, harmonic mean...
- Support flexible ranking on the list, e.g., memory capacity / MD-easy
  See demo: `https://www.vi4io.org/hpsl/sorting`

Tuning for improving the Geom-Mean value

Description	Input (11 values)	**Geom**	Arithmetic	Harmonic
Balanced system	... 10 10 **10**	10	10	10
One slow bench	... 10 10 **1**	8.1	9.2	5.5
Tuning worst 2x	... 10 10 **2**	8.6	9.3	7.3
Tuning good 2x	... 10 **20 1**	8.6	10.1	5.6
Tuning good 100x	... 10 **100 1**	10	17.4	5.8

Geom mean honors tuning equally, insensitive to “outliers”

Next Steps

IO500

- Running IO500 on more systems (currently: DKRZ + KAUST)
- Scripts are available here: `https://github.com/VI4IO/io-500-dev`
- Checking alternative for MD-hard, the md-real-io benchmark
- Enable extension "modules" for IO-500, e.g., concurrent workloads, I/O kernels

VI4IO aims to support dev. of community benchmarks, e.g., via roadmaps

High-Performance Storage List

- Towards data center list with benchmarking results
- Support subcomponents for compute, e.g., 50 nodes with special equipment
- Support multiple benchmarking reports for cluster/storage

You are welcome to participate in the VI4IO and the IO-500 efforts, `https://io500.org`

History of the IO-500

- Dec. 2015: The High-Performance Storage list has been created
  - Contains a simple approach to store sustained performance
- June 2016: Talks from Lofstead, Kunkel about benchmarking during ISC BoF
- Nov. 2016: joint BoF from Kunkel, Bent, Lofstead during SC
- Nov. 2016: Creation of a mailing list for subsequent discussion
  - There have been discussions about the approach, benchmarks
- June 2017: BoF during ISC presenting approach, benchmarks, rules
- Mid 2017: Selected runs on Top500 sites
- Nov. 2017: Show results during SC in a BoF

Collected Information on the HPSL for Performance

Peak Performance

- Theoretical value based on hardware limits
  - e.g. network (server) throughput, SATA limits
- Best performance of one server x number of servers.
- Describe in the text how the peak is computed

Sustained Performance

- Actually observed performance with an application or benchmark
- You can use any benchmark and measurement protocol
- Just make sure you are not measuring cache effects
- Describe in the text how the value has been measured