



aasish-kumar.sharma@gwdg.de

Aasish Kumar Sharma

IO500 Benchmark Suite: Understanding and Running The Tests

High-Performance Computing System Administration (HPCSA)

Course Structure

Session Breakdown (90 minutes total)

- Part 1: Theory (20-25 min) Benchmarking concepts, components, and metrics
- Part 2: Theory (30-35 min) IO500 Benchmark concepts, components, and metrics
- Part 3: Hands-on (35-40 min) Practical exercises on cluster
- Part 4: Discussion (10-15 min) Q&A and troubleshooting

Learning Objectives

- By the end of this session, you will be able to:
 - ▶ Understand IO500 test components and scoring
 - ▶ Install and configure IO500 on HPC systems
 - ▶ Run benchmarks and interpret results
 - ▶ Diagnose common storage performance issues

Outline

1 Theory Part 2: Introduction to IO500

2 Test Components

3 Performance Metrics

4 Installation & Configuration

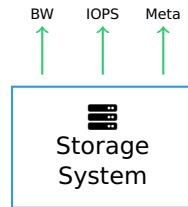
5 Running & Analyzing

6 Hands-On Exercises

7 Conclusion

What is IO500?

- **Industry standard** storage benchmark suite
- Measures **real-world** storage performance
- Used by supercomputing centers worldwide
- Provides **comparable** results across systems
- Tests **both** bandwidth and metadata



Real-World Analogy

- Traditional benchmarks are like testing a car's top speed on an empty highway.
- IO500 tests how it performs in real traffic, with turns, hills, and stops.

Why IO500 for HPC?

Research Computing Challenges

- Scientific workflows mix large data transfers with many small files
- Checkpointing requires consistent write performance
- Data analysis involves random access patterns
- Shared systems need fair resource allocation

Traditional Benchmarks:

- Single workload
- Peak performance
- Limited applicability

IO500 Approach:

- Mixed workloads
- Sustained performance
- Real-world relevance

Outline

1 Theory Part 2: Introduction to IO500

2 Test Components

3 Performance Metrics

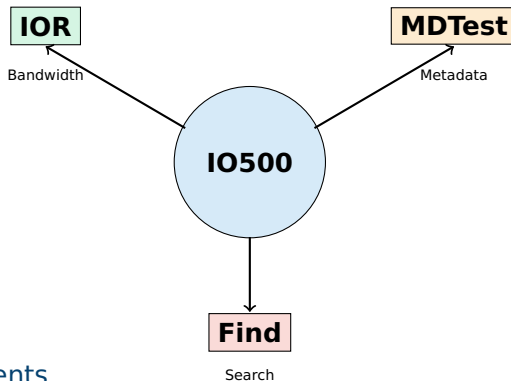
4 Installation & Configuration

5 Running & Analyzing

6 Hands-On Exercises

7 Conclusion

IO500 Test Suite



Three Main Components

IOR - Measures data throughput (GiB/s)

MDTest - Measures metadata operations (kIOPS)

Find - Measures file discovery performance (kIOPS)

IOR: Interleaved Or Random Bandwidth Tests

Test	Access	Block	Workload
Easy Write	File-per-process	2 MiB	Checkpoint
Easy Read	File-per-process	2 MiB	Analysis
Hard Write	Shared file	47 KB	Logging
Hard Read	Shared file	47 KB	Database

Easy vs Hard

- **Easy:** Each process works independently (parallelizes well)
- **Hard:** Processes share files (tests coordination overhead)

MDTest: Metadata Operations Test

Operations Tested:

- File creation
- File stat (metadata query)
- File reading
- File deletion

Directory Structures:

- **Easy:** Flat (single level)
- **Hard:** Nested (tree)

Why This Matters for Research

- Version control systems (Git)
- Simulation output (millions of small files)
- Backup and archival operations
- Build systems and compilation

Outline

- 1 Theory Part 2: Introduction to IO500
- 2 Test Components
- 3 Performance Metrics**
- 4 Installation & Configuration
- 5 Running & Analyzing
- 6 Hands-On Exercises
- 7 Conclusion

IO500 Scoring

Geometric Mean Formula

- Uses geometric mean to prevent good result from hiding
- One poor performance elsewhere could average the good one.

Bandwidth Score:

$$BW = \sqrt[4]{\prod IOR_{tests}}$$

Total Score:

$$IO500 = \sqrt{BW \times IOPS}$$

IOPS Score:

$$IOPS = \sqrt[n]{\prod MD_{tests}}$$

This ensures balanced performance across all workload types

Interpreting Results

Score Range	Classification	Typical System
> 10	High Performance	Top supercomputers
1–10	Good Performance	Research clusters
0.1–1	Entry Level	Department systems
< 0.1	Performance Issues	Needs attention

Performance Ratios to Watch

- **Easy/Hard Ratio** - Reveals coordination overhead
- **Write/Read Ratio** - Shows caching effectiveness
- **BW/IOPS Balance** - Indicates system optimization focus

Outline

1 Theory Part 2: Introduction to IO500

2 Test Components

3 Performance Metrics

4 Installation & Configuration

5 Running & Analyzing

6 Hands-On Exercises

7 Conclusion

Quick Installation

Prerequisites

```
1 # Load MPI module
2 module load openmpi
3
4
5 # Verify MPI installation
6 mpicc --version
7
```

Build IO500

```
1 git clone https://github.com/IO500/io500.git
2
3 cd io500
4
5 git checkout io500-sc23
6 ./prepare.sh
7
```

Basic Configuration

Minimal Configuration (config-test.ini)

```
1 [global]
2 datadir = ./datafiles
3 resultdir = ./results
4
5 [ior-easy]
6 transferSize = 2m
7 blockSize = 1g
8
9 [ior-hard]
10 transferSize = 47008
11 blockSize = 47008
12
13 [mdtest-easy]
14 files-per-proc = 10000
15
16 [mdtest-hard]
17 files-per-proc = 10000
18 tree-depth = 2
19
```

Note

- For learning,
 - ▶ Use smaller file counts and
 - ▶ Shorter runtimes

Outline

1 Theory Part 2: Introduction to IO500

2 Test Components

3 Performance Metrics

4 Installation & Configuration

5 Running & Analyzing

6 Hands-On Exercises

7 Conclusion

Running IO500

Single Node Test

```
1 # Run with 2 processes
2 mpirun -np 2 ./io500 config-test.ini
3
4
5 # Monitor progress
6 tail -f results/*/result.txt
7
```

Cluster Test (SLURM)

```
1 #!/bin/bash
2 #SBATCH --nodes=2
3 #SBATCH --ntasks-per-node=4
4 #SBATCH --time=01:00:00
5
6 module load mpi
7 mpirun ./io500 config-cluster.ini
```

Reading Results

Example Output

```
1 [RESULT] ior-easy-write    0.142 GiB/s : time 383s
2 [RESULT] mdtest-easy-write 4.953 kIOPS : time 301s
3 [RESULT] ior-hard-write   0.105 GiB/s : time 323s
4 [RESULT] mdtest-hard-write 2.811 kIOPS : time 301s
5 [RESULT] find              40.483 kIOPS : time 58s
6 [SCORE ] Bandwidth 0.112 GiB/s : IOPS 7.505 kiops
7 [SCORE ] TOTAL 0.915 [INVALID]
8
```

Red Flags in This Output

- **INVALID** tag indicates data corruption
- Very low bandwidth (< 0.15 GiB/s)
- Good metadata performance (bright spot)

Outline

1 Theory Part 2: Introduction to IO500

2 Test Components

3 Performance Metrics

4 Installation & Configuration

5 Running & Analyzing

6 Hands-On Exercises

7 Conclusion

Exercise Session Overview

Setup (5 minutes)

- Connect to cluster
- Verify MPI environment
- Clone IO500 repository

Four Exercises (30 minutes)

- 1 (10 min)** Installation and basic configuration
- 2 (10 min)** Running your first benchmark
- 3 (5 min)** Analyzing results
- 4 (5 min)** Performance comparison

Wrap-up (5 minutes)

- Discussion of results
- Q&A

Exercise 1: Installation (10 min)

Objective

- Install IO500 on the cluster and verify the build

Tasks

- 1 SSH to cluster: `ssh username@cluster.uni-goettingen.de`
- 2 Load MPI module: `module load openmpi`
- 3 Clone repository: `git clone https://github.com/IO500/io500.git`
- 4 Checkout version: `cd io500 && git checkout io500-sc23`
- 5 Build: `./prepare.sh`
- 6 Verify: `ls -l io500`

Expected Output

- You should see compiled binaries: `io500, ior, mdtest, pfind`

Exercise 2: First Benchmark (10 min)

Objective

- Run a minimal IO500 test and observe execution

Tasks

```
1 # 1. Create test configuration
2 cp config-minimal.ini config-exercise.ini
3
4 # 2. Edit config (reduce file counts for speed)
5 vim config-exercise.ini
6 # Change files-per-proc to 1000
7
8 # 3. Run benchmark
9 mpirun -np 2 ./io500 config-exercise.ini
10
11 # 4. Monitor in another terminal
12 tail -f results/*/result.txt
13
```

Questions to Consider

- Which test takes longest? Why?
- What resources are being used?

Exercise 3: Result Analysis (5 min)

Objective

- Interpret your benchmark results and identify patterns

Tasks

```
1 # 1. View summary
2 cat results/*/result_summary.txt
3
4 # 2. Calculate Easy/Hard ratios
5 # Easy_Write / Hard_Write = ?
6 # Easy_Read / Hard_Read = ?
7
8 # 3. Identify issues
9 # Are any results marked INVALID?
10 # Which tests performed best/worst?
11
```

Group Discussion

- Compare your results with neighbors.
 - ▶ What differences do you see?
 - ▶ Why might they differ?

Exercise 4: Performance Comparison (5 min)

Objective

- Test with different configurations and compare

Tasks

```
1 # 1. Run with different process counts
2 mpirun -np 4 ./io500 config-exercise.
   ↪ ini
3
4 # 2. Compare results
5 diff results/run1/result_summary.txt \
6     results/run2/result_summary.txt
7
8 # 3. Analyze scaling
9 # Does 2x processes = 2x performance?
10
```

Expected Observation

- Performance rarely scales linearly.
 - ▶ Why? (overhead, contention, limits)

Exercise Troubleshooting

Common Issues

- **Build fails** - Check MPI module: `module list`, try `module load mpi`
- **Permission denied** - Ensure write access: `chmod 755 datafiles`
- **Slow execution** - Normal for complete runs; reduce file counts for testing
- **INVALID results** - May indicate storage issues; check `df -h`

Get Help

- Check log files in `results/*/*.log`
- Consult with respective instructors
- Review documentation at <https://io500.org>

Outline

1 Theory Part 2: Introduction to IO500

2 Test Components

3 Performance Metrics

4 Installation & Configuration

5 Running & Analyzing

6 Hands-On Exercises

7 Conclusion

Key Takeaways

Understanding IO500

- Comprehensive benchmark suite for storage systems
- Tests both bandwidth (IOR) and metadata (MDTest)
- Uses geometric mean for balanced scoring
- Reflects real-world application behavior

Practical Skills Gained

- Installation and configuration
- Running benchmarks on HPC systems
- Interpreting performance results
- Identifying storage bottlenecks

Resources for Further Learning

Official Resources

- **Website:** <https://io500.org>
- **GitHub:** <https://github.com/IO500/io500>
- **Documentation:** <https://io500.org/documentation>
- **Results Database:** <https://io500.org/list>

Next Steps

- Run IO500 on your research filesystem
- Compare with other systems in IO500 list
- Experiment with different configurations
- Use results to guide storage optimizations

Questions & Discussion

Learn and Share!

Questions & Discussion

Feedback - Contact:

Aasish Kumar Sharma

`aasish-kumar.sharma@gwdg.de`

Materials:

<https://hps.vi4io.org/teaching/ressources/start>

Reference: Complete Config File

```
1 [global]
2 datadir = ./datafiles
3 resultdir = ./results
4
5 [debug]
6 stonewall-time = 300
7
8 [ior-easy]
9 transferSize = 2m
10 blockSize = 9920000m
11 API = POSIX
12
13 [ior-hard]
14 transferSize = 47008
15 blockSize = 47008
16 API = POSIX
```

```
1 [mdtest-easy]
2 files-per-proc = 1000000
3 API = POSIX
4
5 [mdtest-hard]
6 files-per-proc = 1000000
7 tree-depth = 3
8 branch-factor = 1
9 API = POSIX
10
11
12 # Find test
13 #=====
14
15 [find]
16 external-script = ./pfind
```

Reference: Monitoring Script

```
1 #!/bin/bash
2 # monitor-io500.sh - Real-time monitoring
3
4 while true; do
5     clear
6     echo "=== IO500 Monitor $(date) ==="
7     echo
8     echo "Disk Usage:"
9     df -h ./datafiles | tail -1
10    echo
11    echo "I/O Statistics:"
12    iostat -x 1 1 | grep -A5 "Device"
13    echo
14    echo "Active Processes:"
15    ps aux | grep -E '(io500|ior|mdtest)' | grep -v grep
16    sleep 10
17 done
```

Reference: Performance Analysis Checklist

When Analyzing Results

- 1 Check for INVALID markers (data integrity issues)
- 2 Calculate Easy/Hard ratios (coordination overhead)
- 3 Compare Write/Read performance (caching effects)
- 4 Examine BW vs IOPS balance (system optimization)
- 5 Look at individual test durations (bottlenecks)
- 6 Compare with similar systems in IO500 list

Red Flags

- Easy/Hard ratio > 3 : Poor shared access handling
- Write/Read ratio > 2 : Cache or read path issues
- Any INVALID results: Storage reliability concerns
- Extremely low scores: Configuration or hardware problems