Benchmarking Working Group (BWG)

Sarp Oral, ORNL
Rick Roloff, Cray

April17, 2013

OpenSFS BWG LUG'13 Update

- Third face-to-face meeting
  - LUG12, SC'12, LUG13
  - SC and LUG are 6 months apart; having one meeting at each gives us semiannual course correction capability
- Bi-weekly concalls on Fridays @ 11:30 AM Eastern
  - Dial in: +1-877-709-0823 Passcode: 4840841
  - Next meeting will be on May 3rd, 2013
- Email list: openbenchmark@lists.opensfs.org
- To join
  - http://www.opensfs.org/get-involved/benchmarking-working-group/
  - http://lists.opensfs.org/listinfo.cgi/openbenchmark-opensfs.org

OpenSFS BWG LUG'13 Update

- Growing
  - ➢25 intuitions/companies actively participating as of LUG13

ORNL
LBNL
FNAL
Exxon Mobil
Intel
DDN
Terascala
NetApp
Cray
Xryatex
InkTank
Instrumental

NREL
Routing Dynamics
Indiana Uni.
Informatik Uni., Germany
ARSC
Dresden Uni., Germany
HPC Results
Illinois Uni.
SDSC
NICS/UTK
EMC
Stanford Uni
Fujitsu

What have we done so far?

- Had a face-to-face meeting at SC'12
  - ➢ Reestablished our goals
    - Finalized the benchmarking spreadsheet
    - Discussed the I/O workload characterization survey and the results
    - Discussed what we have done since LUG'12
    - Discussed what we are going to do until LUG'13

What have we done so far?

- At SC'12 our accomplishments up-to-date were found as
  - ➢ Released our I/O workload characterization survey to the public
    - Had five responses
      - ➢ ARSC, OLCF, NICS, SDSC, Fujitsu
  - ➢ Started our benchmark characterization effort
- Since SC'12, we have finalized both of these two efforts

What have we done so far?

- At SC'12 our future goals were stated as
  - Provide a mechanism to obtain a hero performance number from a parallel file system.
  - Provide a mechanism to obtain workload based performance numbers from a parallel file system
  - Provide methods or tools to monitor a parallel file system
  - Provide methods or tools to assess and evaluate the metadata performance

What have we done so far?

- Five task groups were formed to follow up these goals
  - Block I/O hero run best practices effort
  - I/O workload characterization effort
  - Application I/O kernel extraction effort
  - Methods or tools to monitor a parallel file system effort
  - Metadata performance evaluation effort
- We have already started making progress on the tasks, at LUG each task group leader will provide an update

Block I/O hero run best practices

Ben Evans, Terascala

April 17, 2013

Members:

- Ben Evans: Terascala, Task Lead
- Mark Nelson: Inktank
- Ilene Carpenter: NREL
- Rick Roloff: Cray
- Nathan Rutman: Xyratex
- Liam Forbes: University of Alaska

Areas of Focus

- Definining tuning limitations
  - “As used in production” is our current working philosophy
- Defining tests
  - Read/write streaming
  - Read/write random
  - Single file, file per process
- Formula of results from tests become “hero number”

Tuning Limitations

- Hero number will cover all filesystems
- Specifying things that should not be done may be too filesystem-specific
- “production tuning” is the shortest path to what we’re looking for: as little ‘cheating’ as possible

Defining Tests

- Streaming, Random, FPP, Single file, …
- Metadata?
- Performing the tests:
  - Ramp up the number of clients and threads until peak throughput is achieved
  - Measure the sustained throughput on the FS servers

Calculating the Hero Number

- Combine the results from all the tests in such a way as to represent a metric for the filesystem
- Something like (streaming*streaming/random ?)
- Unknowns
  - How to add FPP/Single file
  - How to balance metadata results

I/O Workload Characterization

Pietro Cicotti – SDSC

April 17, 2013

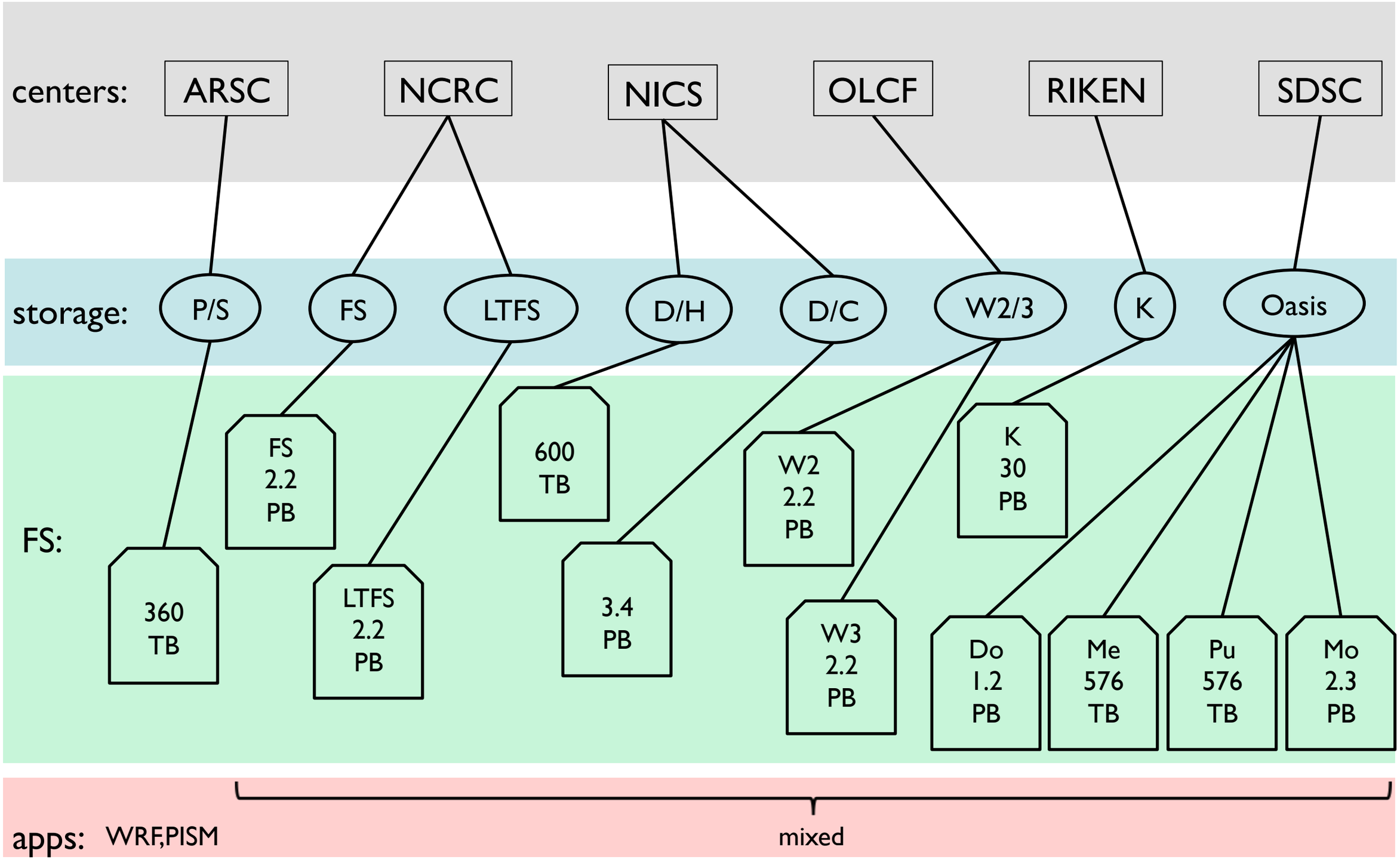
Members

- Leader: Pietro Cicotti - SDSC
- Members:
  - Ilene Carpenter - NREL
  - Rick Mohr - UTK
  - Mike Booth – HPC Results
  - Ben Evans – Terascala

Workload Characterization Effort

- Goals
  - Understand and characterize common workloads
  - Identify and create a set of representative synthetic workloads
- Synergies
  - Kernel extraction/creation
  - Monitoring

Survey Responses



Some stats…

	OLCF (widow2)	OLCF (widow3)	NCRC FS	NCRC LTFS	NICS (Kraken)	NICS (Medusa)	SDSC	ARSC	RIKEN
# users	2000	2000	500-600	500-600	1650	1650	100+	345	NA
server Version	1.8.8	1.8.8	1.8.8	1.8.8	1.8.4	1.8.6	1.8.7	2.1.2	NA
Client version	1.8.8-1.8.9	1.8.8-1.8.9	1.8.8-1.8.9	1.8.8-1.8.9	1.8.4	1.8.6, 1.8.8	1.8.7	>=1.8.6	NA
# clients	19042	19042	3908	40	9440	400	1638	500	88000
Interconnects (server-client)	DDR IB, Cray Gemini	DDR IB, Cray Gemini	QDR IB, Cray Gemini	QDR IB, Cray Gemini	Cray SeaStar	QDR IB	10 GigE, IB, Myrinet	IB, Ethernet	IB, Tofu
size (raw)	2.2 PB	2.2 PB	900 TB	3.1 PB	3.36 PB	600 TB	4608 TB	360 TB	10 -30PB
# files	107M	117M	65M	38M	256M	18.2M	141M	7.1M	NA

2.x

IB

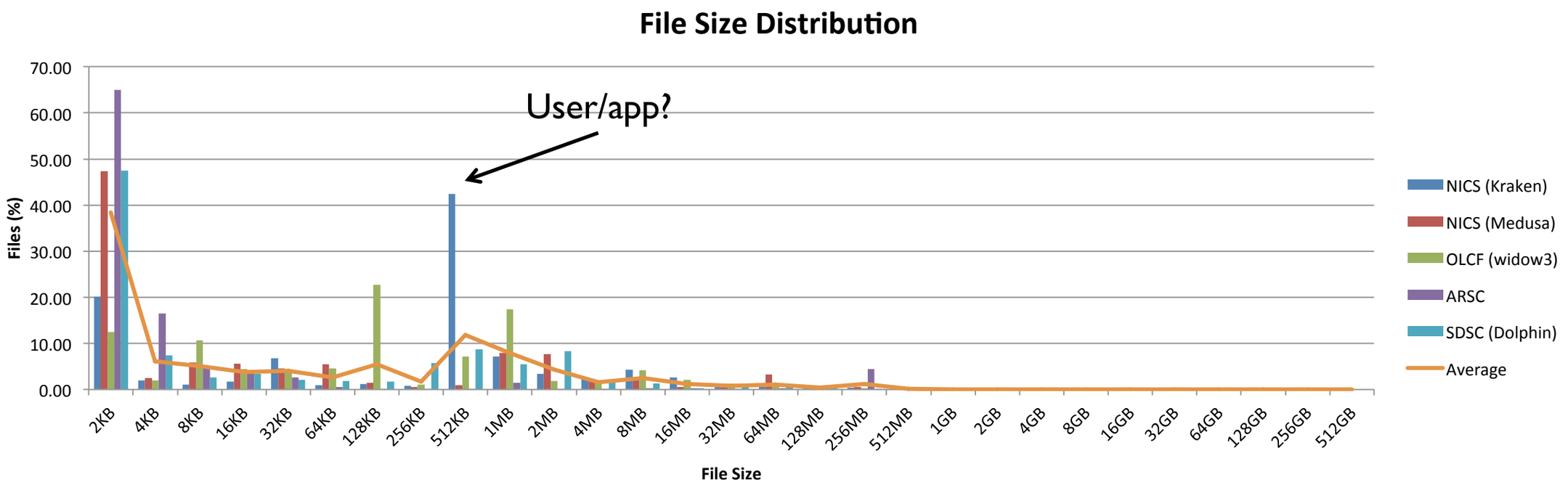
Cray

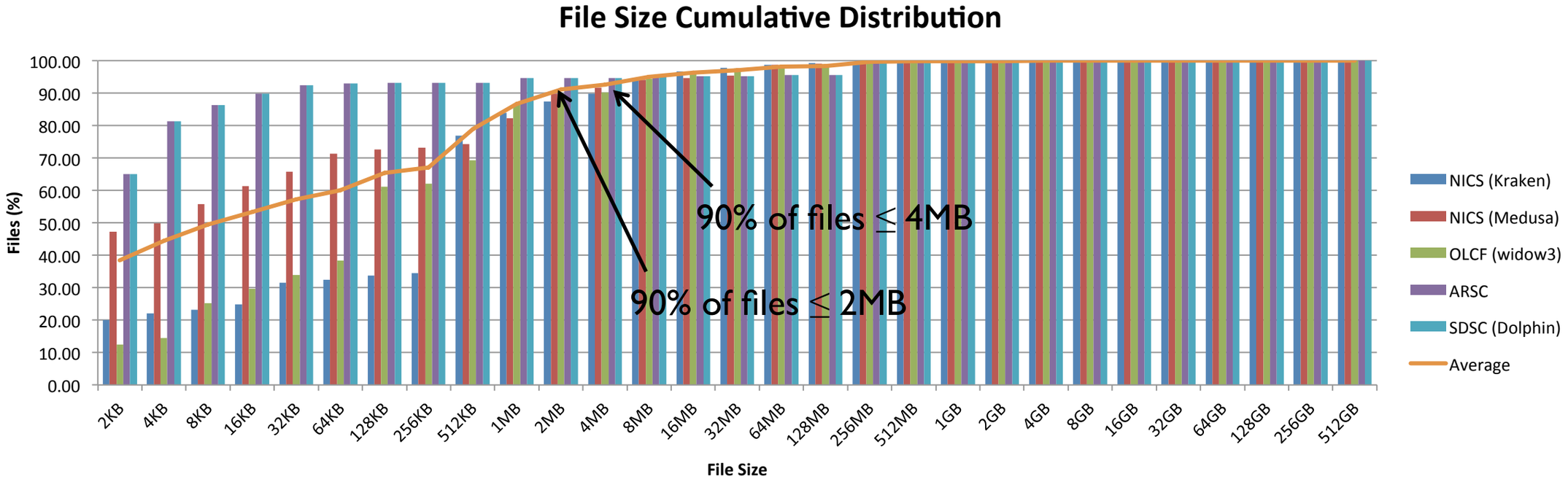
ethernet

Myrinet

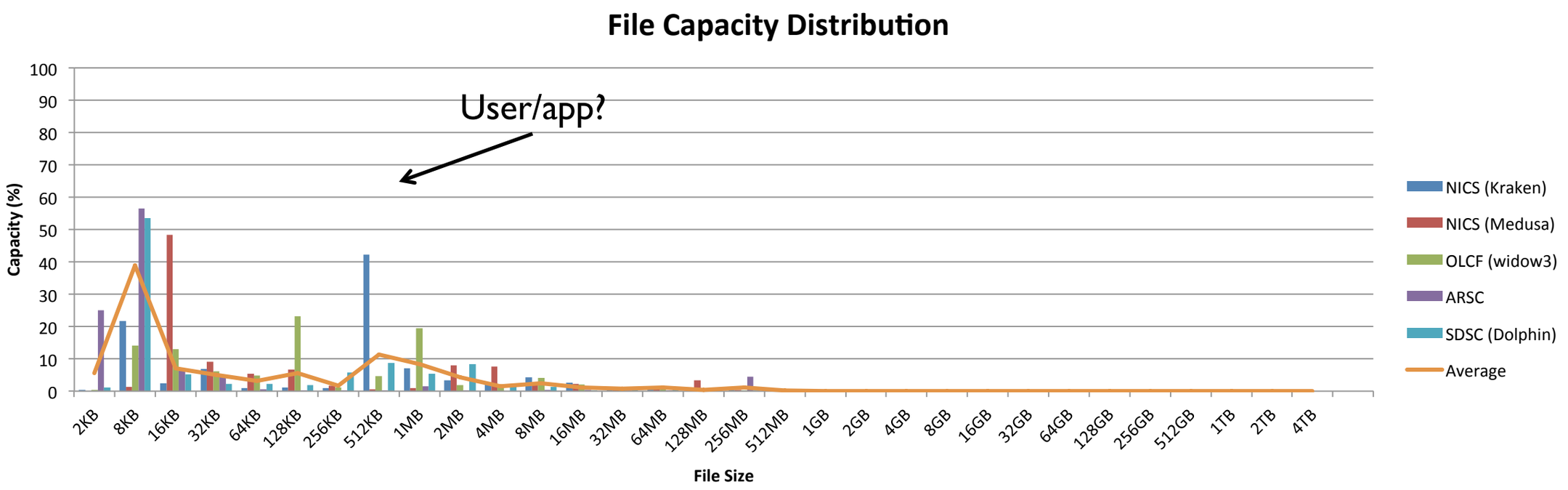
Tofu (Fujitsu)

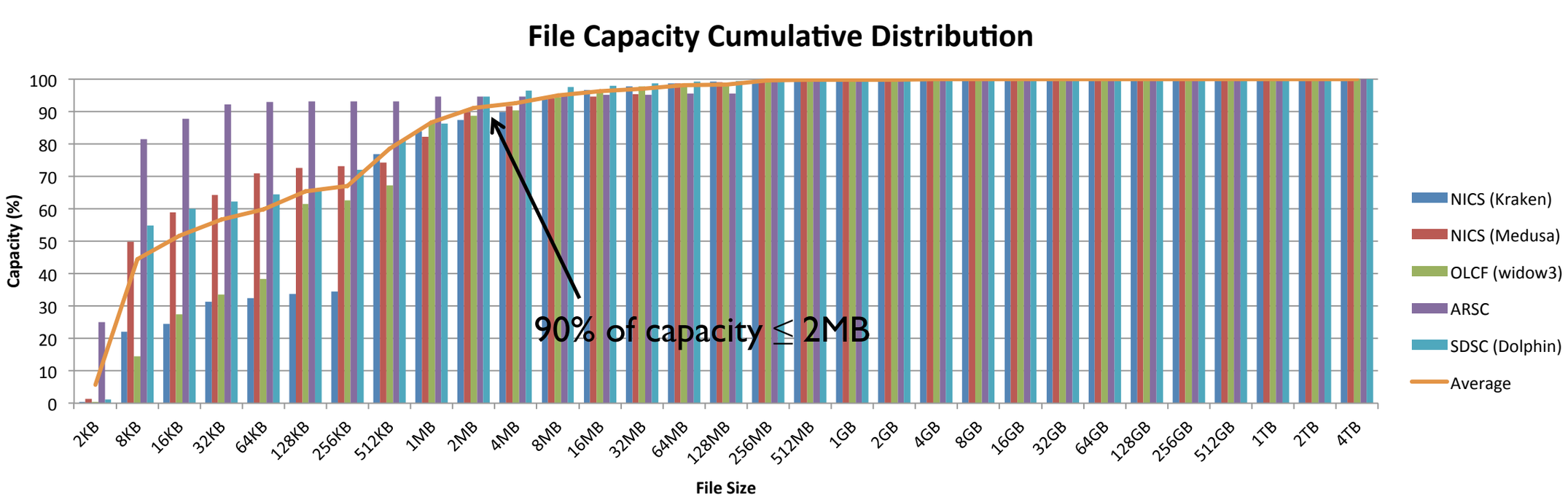
File Size Distribution



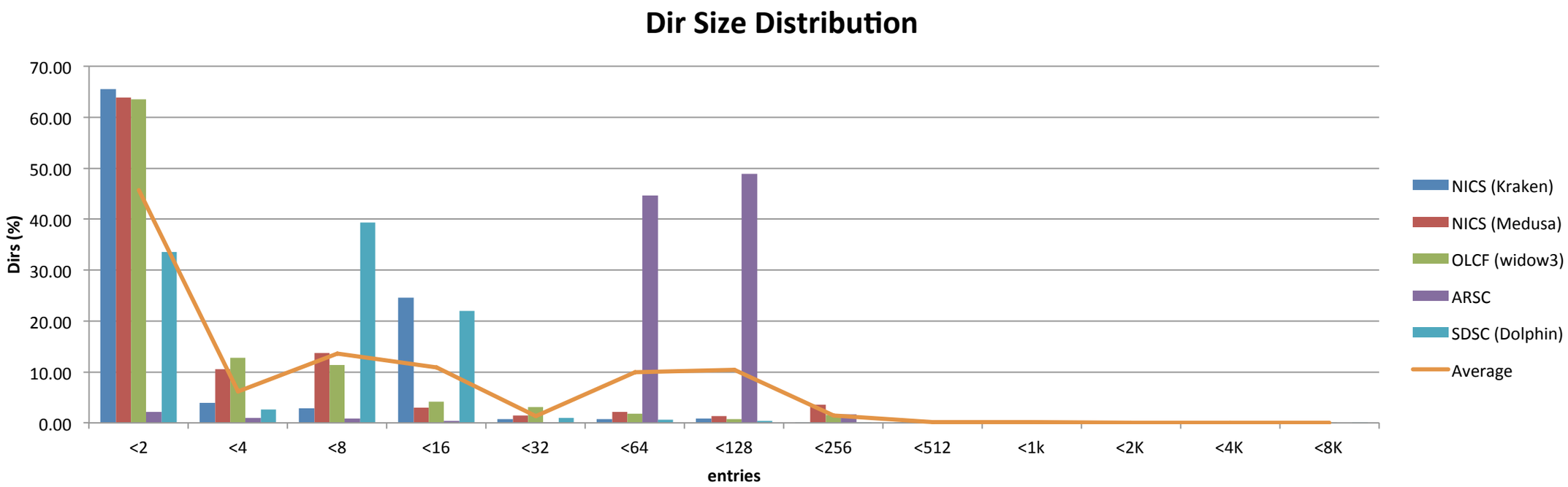


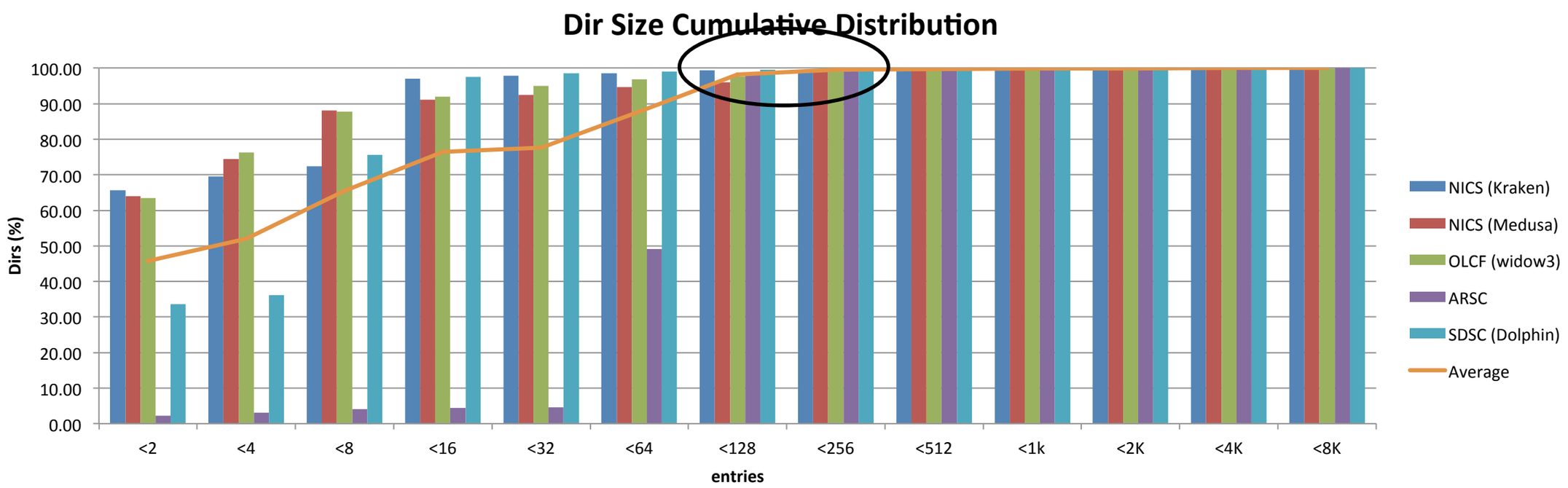
File Capacity Distribution



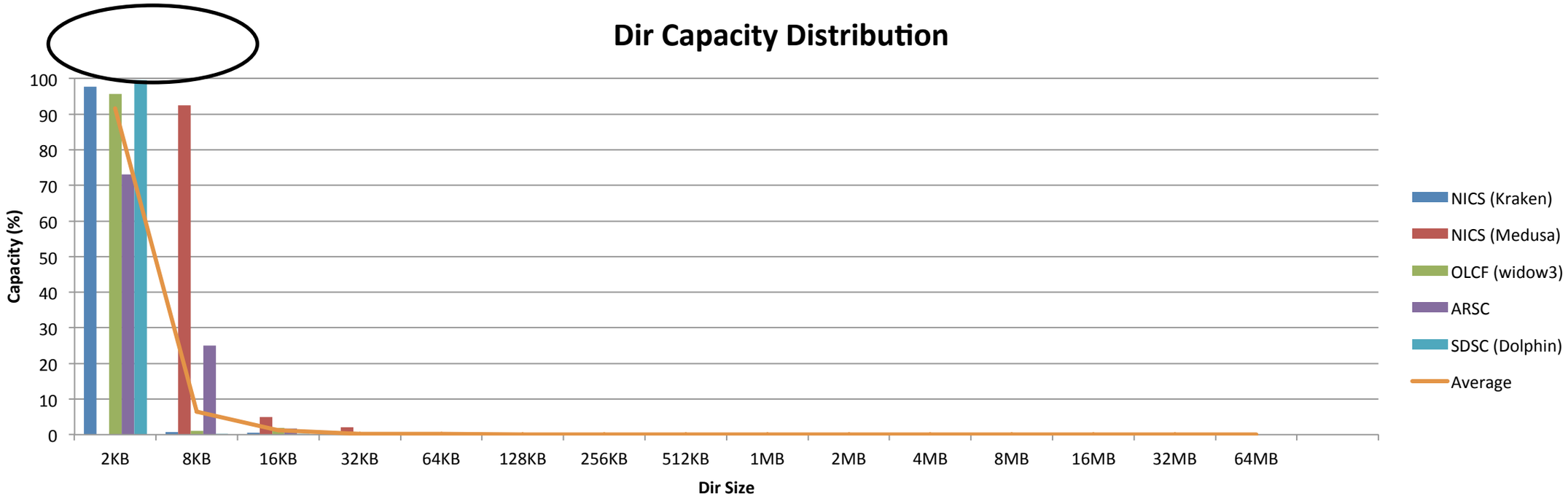


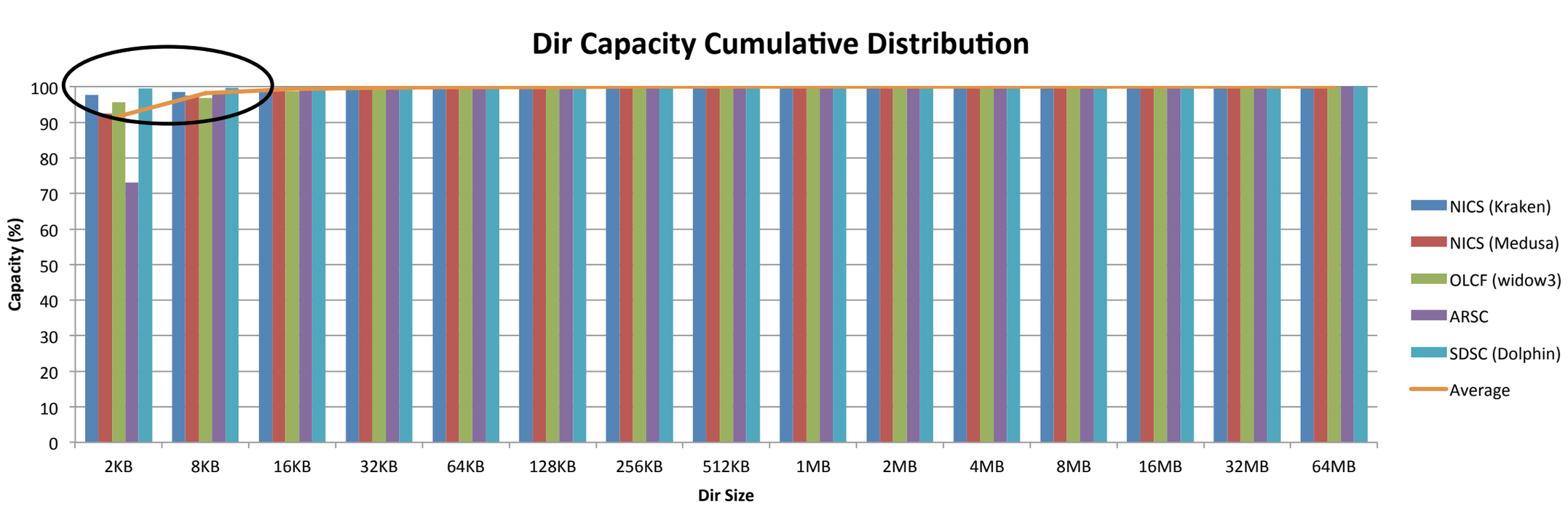
Dir Size Distribution



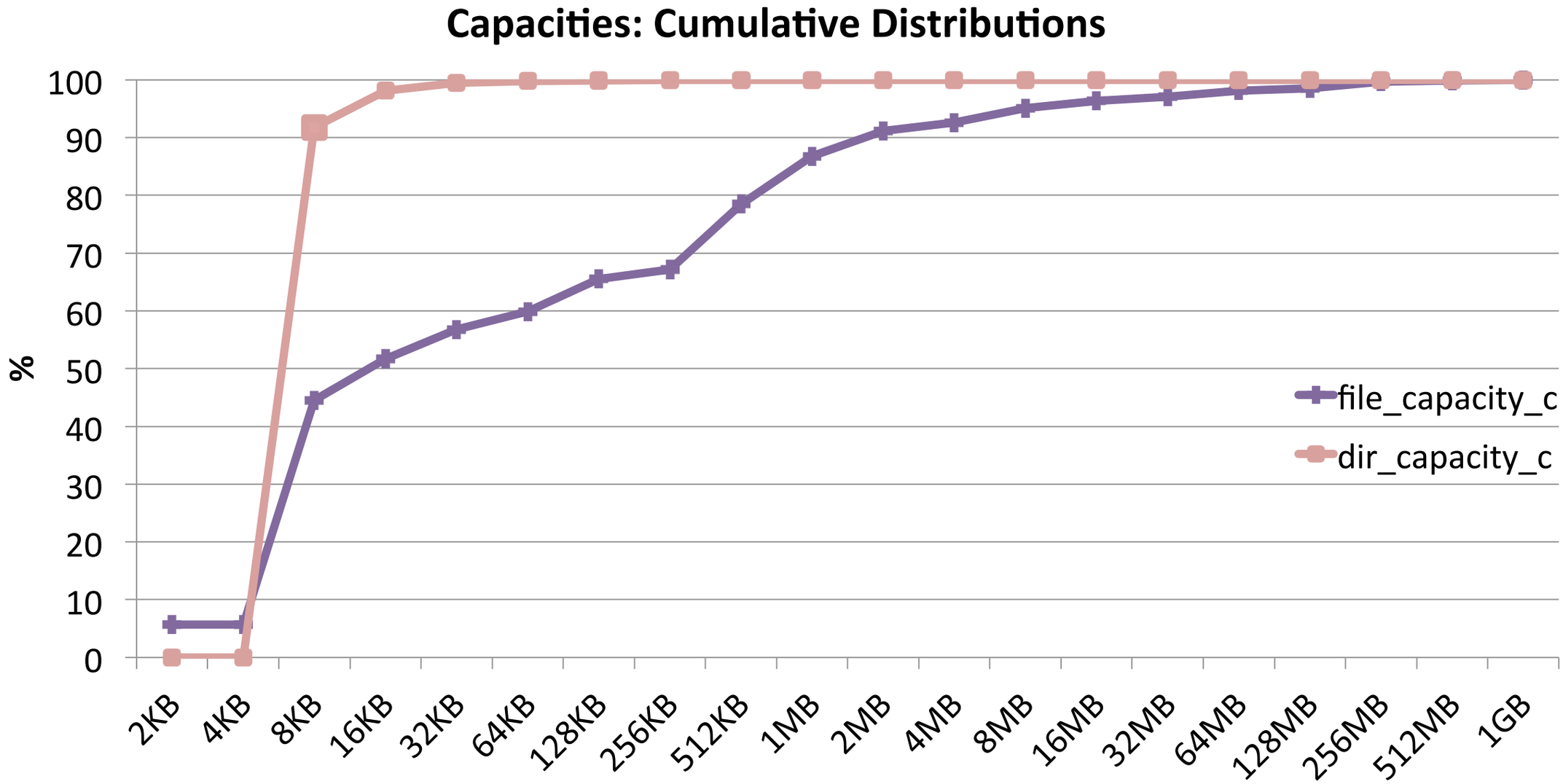


Dir Capacity Distribution





Dir & File Capacity



Next?

- Complete surveys analysis
  - Timestamps
  - Summarize our analysis in a report
- Focused experiments
  - Engage one or more centers
  - Monitoring (see monitoring effort)
- Propose a way to reproduce workloads
  - Use/combine existing benchmarks
  - Create our own tools (see kernel extraction effort)

Application I/O kernel creation effort

Ilene Carpenter (Pietro Cicotti), NREL

April 17, 2013

Members

- Leader: Ilene Carpenter - NREL
- Members:
  - Jeff Layton - Dell
  - Pietro Cicotti - SDSC
  - Bobbie Lind - Intel

Application IO Kernel Creation Effort Charter

- Develop application kernels to complement those that already exist, to allow evaluation of File System performance and scalability for specific application workloads.
  - Extraction
  - Creation of kernel that mimics something we can't extract
- Address the high end HPC as well as small and medium installations benchmark needs
- Tools applicable to Lustre and other file systems
- All tools will be open source

Existing application I/O kernels

- Flash I/O (HDF5)
- MadBench2 (cosmology)
- Chombo I/O (AMR, HDF5)
- QIO (QCD)
- GCRM (climate) – parallel netcdf

Proposed Roadmap

- Evaluate the workloads that can benchmarked
- Develop a process to create workloads that are representative of commercial or sensitive application for which source code may be unavailable.
  - strace
  - other methods
- Develop workloads representative of HTC
- Build scripts to allow ease of use of the recommended tools
- Write documentation for using tools
- Collect statistics from users of application I/O benchmarks

Application IO kernel group asks from OpenSFS

- Share any open source synthetic benchmarks code that represents end-user application IO patterns
- Share the workloads that create pain points to Lustre FS
- Share cases of poor performance workloads and applications

Tools for Lustre File System Monitoring

Andrew Uselton, NERSC

April 17, 2013

Members

- Andrew Uselton, NERSC, Task Lead
- Ben Evans, Terascala
- Liam Forbes, University of Alaska
- Jeff Layton, Dell
- Mark Nelson, Inktank

Overview

- Use cases
- Data sources
- Collection tools
- Presentation tools

Use Cases:

- Real time view
- Workload analysis
- Incident investigation
- Anomaly detection

Answering the question:

- What is the weather like right now?
- What is the climate like on this system?
- Why is performance so poor?
- What is this odd phenomenon?

Data Sources

- Linux /proc
- RAID controller API
- Benchmark tests
- ?

Collection Tools

- The Lustre Monitoring Tool (LMT) and Cerebro - Andrew
- collectl and ganglia - Ben
- collectd and graphite -
- blktrace - Mark
- Ceph -
- perf -
- sysprof –
- lltop and xltop – Richard Henwood

The Lustre Monitoring Tool (LMT)

- Read and write bytes per second on each OST
- CPU utilization on each OSS
- Metadata operations per second on the MDS
- CPU utilization on the MDS
- Bytes moved per second on each Inet router
- https://github.com/chaos/lmt/wiki

collectl

- CPU, Memory, IO, TCP, Infiniband and more
- Per-process and slab memory monitoring
- Runs as a daemon or via the command-line
- Supports sub-second time intervals
- Supports multiple front-ends and interfaces
- File system agnostic
- http://collectl.sourceforge.net/

Presentation Tools

- LMT
  - ‘ltop’
  - ‘lwatch’
  - Ad hoc scripts to query MySQL
- Cacti
- ?

Metadata Performance Evaluation

Sorin Faibish, EMC

April 17, 2013

Members

- Leader: Sorin Faibish - EMC
- Members:
  - Branislav Radovanovic - NetApp
  - Richard Roloff - Cray
  - Cheng Shao, Wang Yibin - Xyratex
  - Keith Mannthey, Bobbie Lind – Intel
  - Gregory Farnum - Inktank

Metadata Performance Evaluation Effort Charter

- Build/select tools that will allow evaluation of File System Metadata performance and scalability
- The tools will help detect pockets of Metadata low performance in cases when users complain of extreme slowness of MD operations
- Benchmark tools will support: POSIX, MPI, and Transactional operations (for CEPH and DAOS)
- Address the very high end HPC as well as small and medium installations benchmark needs
- Tools applicable to Lustre and: CEPH, GPFS…

MPEE Proposed Tools

- The current proposed list of benchmarks:
  - mdtest – widely used in HPC
  - fstest - used by pvfs/OrangeFS community
  - Postmark and MPI version - old NetApp benchmark
  - Netmist and MPI version – used by SPECsfs
  - Synthetic tools – used by LANL, ORNL
  - MDS-Survey - Intel's metadata workload simulator.
  - Any known open source metadata tools used in HPC
  - Add new Lustre statistics specific to MD operations.

MPEE Usecases

- **mdtest**: test file MD operations on MDS: open, create, lookups, readdir; used in academia and as a comparison tool of FS MD.
- **fstest**: small I/O's and small files as well as lookups, targeting both MDS and OSS operations and MD HA for multiple MDS's.
- **Postmark**: old NetApp benchmark – I built an MPI version; it is used to measure MD operations and file size scalability and files per directory scalability.
- **Netmist**: used to model any workload from statistics including all MD operations and file operations. Can model Workload objects for I/O performance mixes and combination of I/O and MD. Suitable for initial evaluation of storage as well as for performance troubleshooting.

MPEE Proposed Roadmap

- Collect benchmark tools candidates from OpenSFS
- Evaluate all the tools and the workloads that can benchmarked
- Recommend a small set of MD benchmark tools to cover the majority of MD workloads
- Collect stats from users of MD benchmarks
- Build scripts to allow ease of use of the recommended tools
- Write documentation for troubleshooting MD performance problems using the toolset
- Create a special website for MD tools

MPEE Asks from OpenSFS

- Share any open source synthetic benchmarks code
- Share a list of MD benchmark tools they currently use to allow select the most suitable and used candidates
- Share MD operations tested to allow build Netmist workload objects
- Share the MD workloads that create pain points to Lustre FS
- Share cases of poor MD performance workloads and applications

What do the next?

- Suggestions?

Questions?