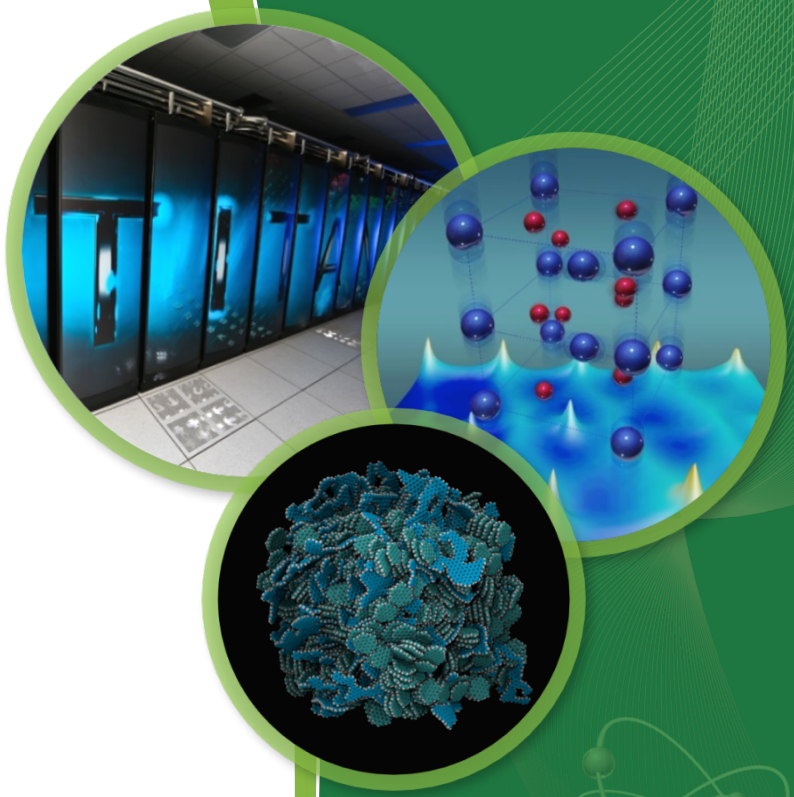
Diving into Petascale Production File Systems through Large Scale Profiling and Analysis

Feiyi Wang

Co-authors: Hyogi Sim, Cameron Harr, Sarp Oral

Oak Ridge National Laboratory

Lawrence Livermore National Laboratory



ORNL is managed by UT-Battelle
for the US Department of Energy

OAK RIDGE National Laboratory

Problem & Design Goals

- Motivation
  - Multiple larg- scale production file systems
  - I/O workload studies at the backend storage
  - jobstats or darshan about I/O on per-job basis
- Goals
  - “File characteristics and usage patterns” on a grand scale
  - Scalable and fast
  - Lightweight and infrastructure-less
  - Portable

A Quick Start

```
$ brew install pkg-config libffi openmpi python
$ pip2 install virtualenv
$ virtualenv pcircle
$ source ~/pcircle/bin/activate
$ (pcircle) $ pip2 install git+https://github.com/olcf/pcircle@dev
```

To run it:

```
$ fprof ~/                    or
$ mpirun –np 8 fprof ~/
```


OAK RIDGE National Laboratory

Design Overview

- Parallelization Engine (PE)
  - Distribute the workload across a cluster of machines to scan the file system
- PE has two key components:
  - work stealing
  - distributed termination

OAK RIDGE National Laboratory

Work Stealing Pattern

1. Each worker maintain a local and independent work queue.
2. If this work queue is empty, it sends work request to other neighbor processes
3. Each worker process will respond such work request from other peers, by split and distribute work items from its work queue.

Each worker processes are peers, no master or slaves.

How do we know or who decide when to quit?

OAK RIDGE National Laboratory

Distributed Termination

- Dijkstra-Scolten algorithm
  - All nodes are arranged into a ring
  - Each node maintain a state of black and white color, they also pass a token of color black & white
  - termination condition: a white n0 receives a white token

The beauty of the solution:

- peer to peer, fully distributed
- self-balanced

OAK RIDGE National Laboratory

Error Recovery

- file system healthy != files are healthy
- “Not stat-able”
  - error code (catch)
  - not return (timer)
  - evil case – “un-interruptable sleep state” (reboot)
  - https://jira.hpdd.intel.com/browse/LU-8696
- This is a case I miss master/slave: better chance of recovery

OAK RIDGE National Laboratory

Other Practical Considerations

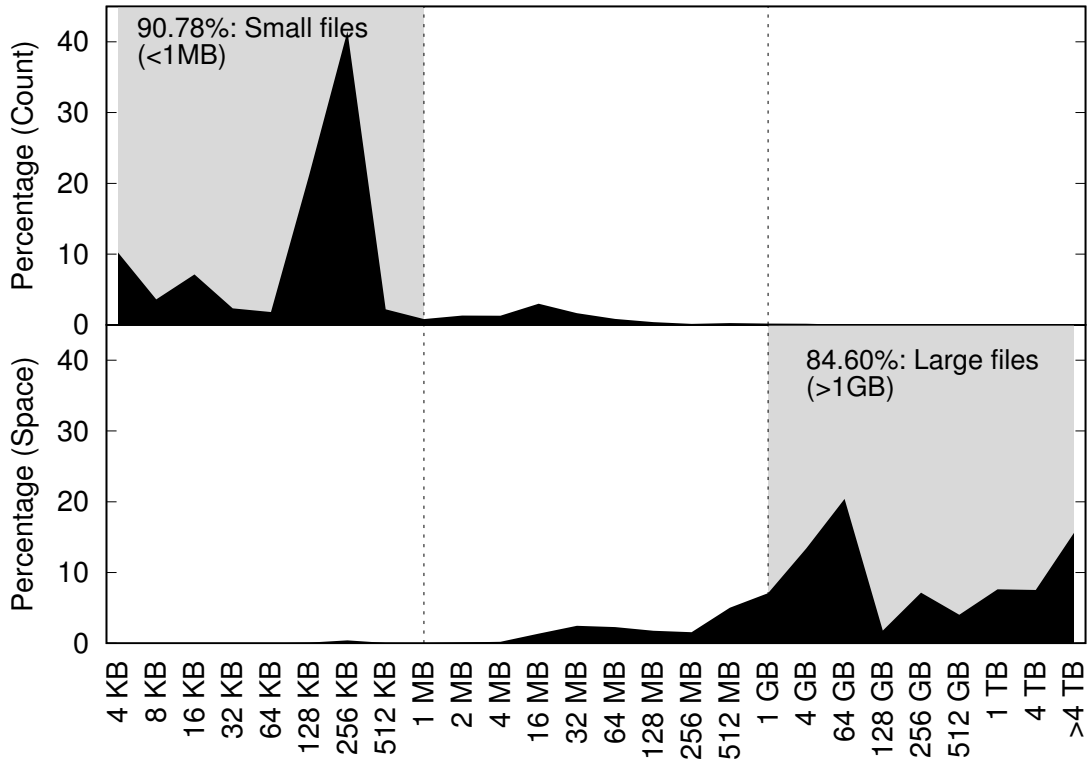
- Work queue size
  - work queue is dynamic, its size depends on file and directory layout
  - extreme case:
    - side-by-side directories, with large # of file
    - 100 directories, each with 10 million files
  - Double ended queue, prioritize file handling
- Sparse file
  - (1) st_block and st_size (2) FILEMAP (3) SEEK_HOLE with lseek()
- LRU cache size
  - Single client scanning billion files
  - lru_size (# client side locks in a LRU queue, default: unlimited)

OAK RIDGE National Laboratory

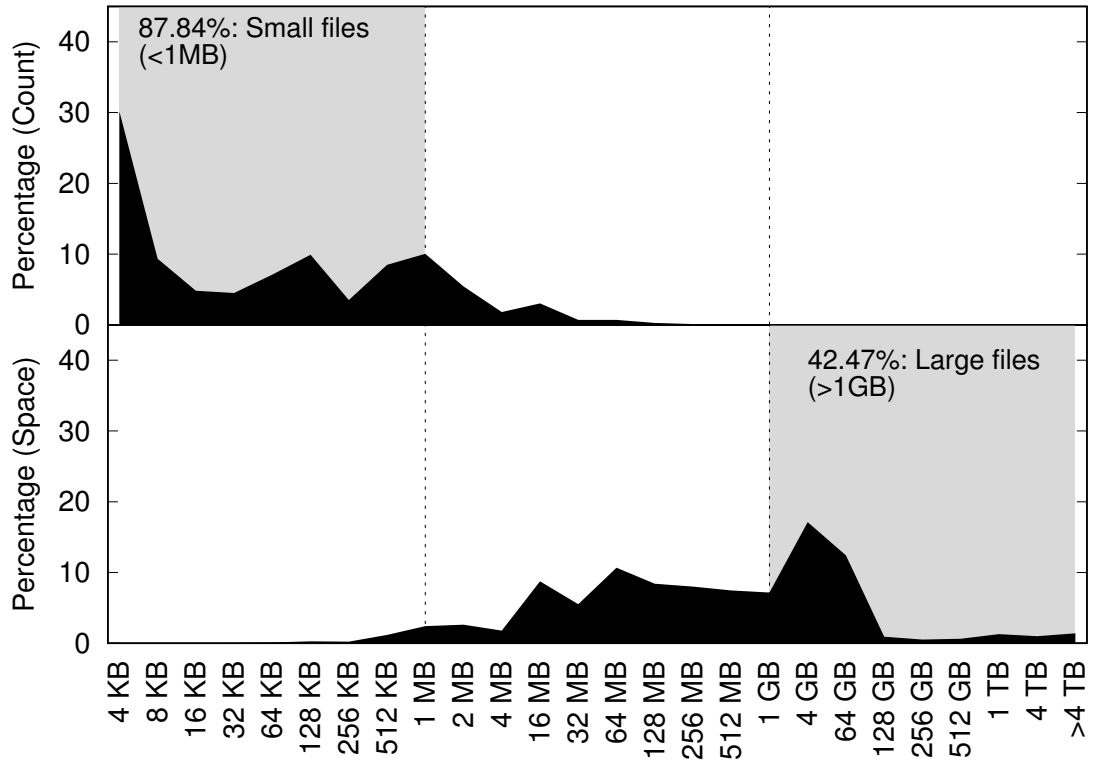
File System Snapshot and Characterizations

	OLCF (atlas1 & atlas2)	LC (lscratche)
File system	Lustre	Lustre
back-end local file system	ldiskfs	ZFS
Capacity (usable)	32 PB	5.7 PB
File count	0.92 billion	1.3 billion
Directory count	115 million	45 million
Hard link count	4,390,426	309,219
Symbolic link count	7,951,784	10,430,723
Sparse file count	3,240,848	N/A
Max # of files in a directory	6,006,529	26,646,573
Average file size	27.67 MB	2.83 MB
Largest file (top N)	32 TiB	12.77 TiB

File size distribution



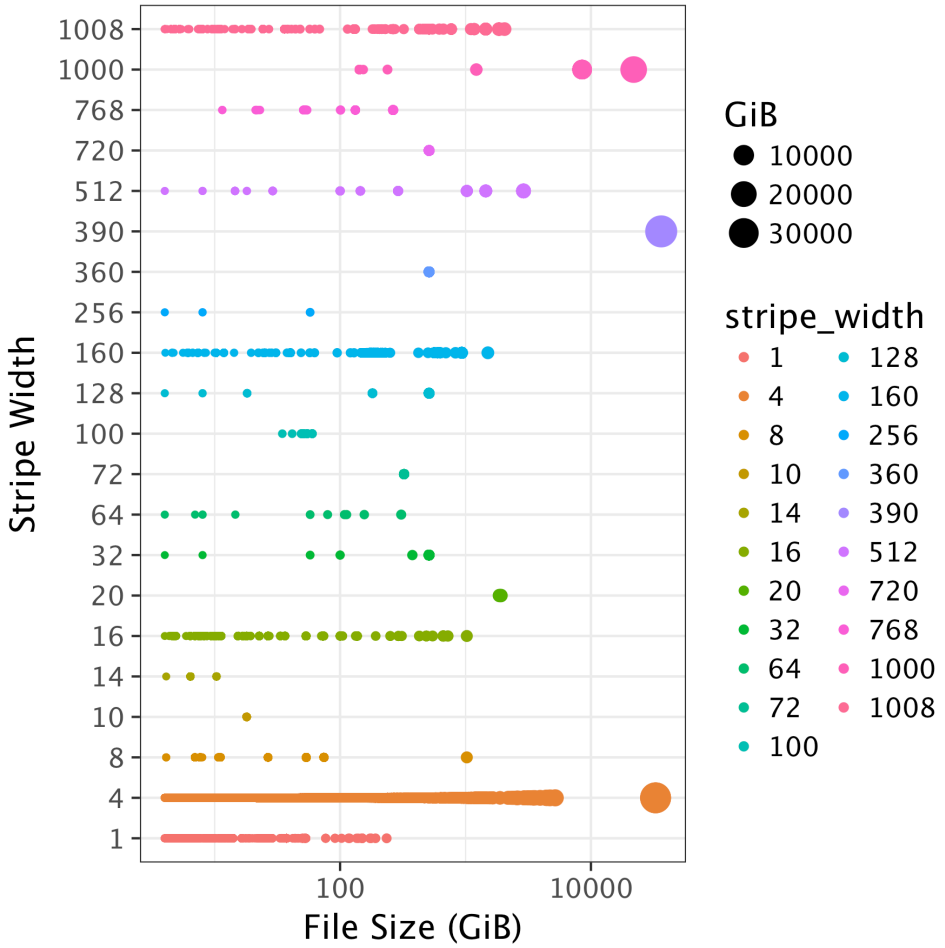
(a) OLCF



(b) LC

OAK RIDGE National Laboratory

Striping Pattern



Stripe count is a perennial cause of performance issues

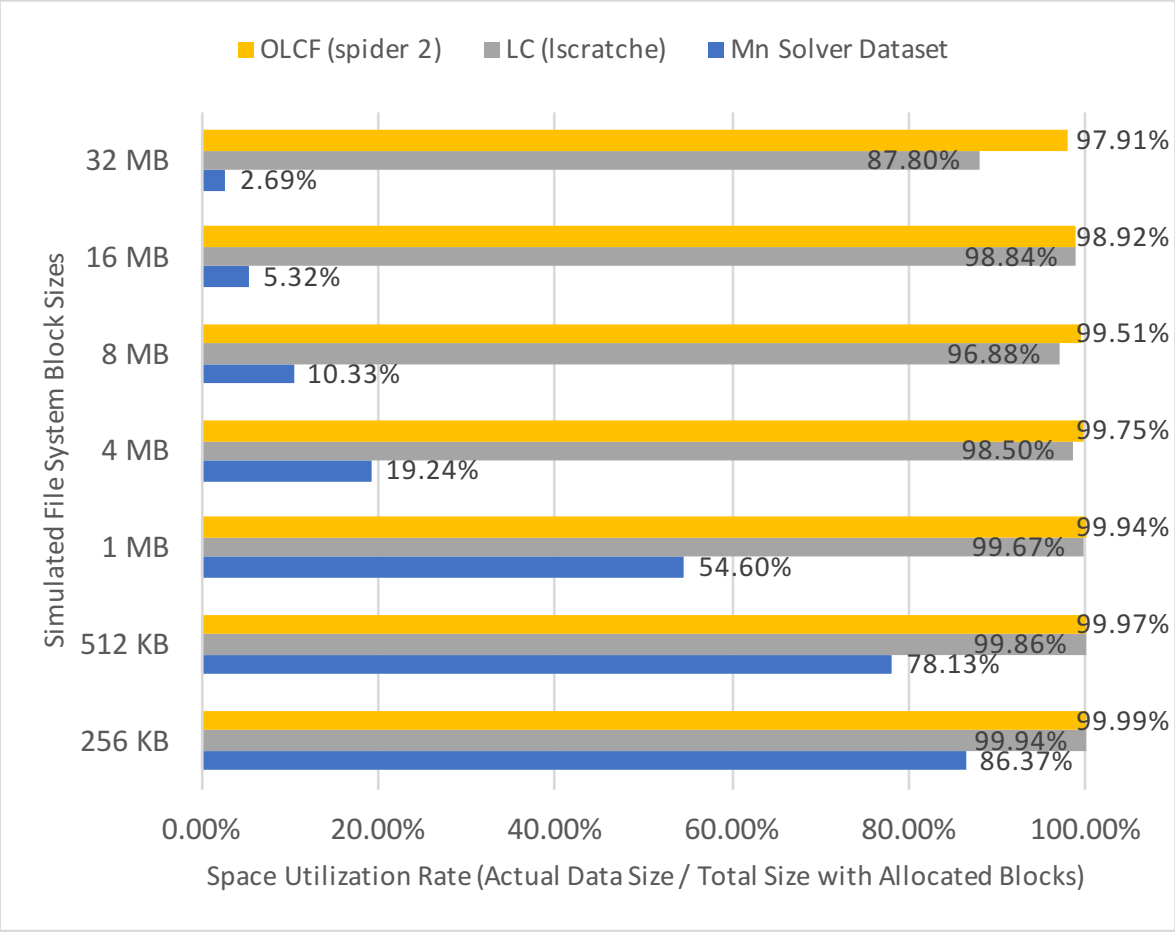
- Ad hoc but justified settings: 4 (OLCF), 1 (LC)
- 513,740 data points collected at OLCF (>=4GB)
- 21 distinct stripe count is in use
- 96.83% files stay with default setting
- 2,262 changed from 4 to 1, suspect file-per-process
- No correlations found between file size and stripe count
  - e.g. 32 TB uses default striping of 4.

OAK RIDGE National Laboratory

Space Utilization Projection

- Why do we want to project?
  - OLCF: Spider 2 (Lustre) -> Spider 3 (GPFS)
  - LC: Sierra is also GPFS-based
  - Build proposal suggested large block size (16MB, 32MB) for better performance
  - Trade-off: increased block size vs. potentially wasted space
- How do we project?
  - Mn Solver Dataset
    - Output from Moment-Closure Solver, close to 100,000, average file size: 11KB
  - OLCF Spider 2
  - LC (lscratche)

Comparison



- Mn Solver Dataset:
  - 86.37% efficiency with 256KB block size; 2.69% with 32MB block size
- OLCF Spider 2:
  - 32MB block for 250PB file system: 2% wasted, or about 5PB
- LC:
  - 32 MB, 13.2% wasted
  - 16 MB is a much better choice

Summary

- We present and demonstrate lightweight, portable and scalable profiling tools
- Three use cases:
  - File system snapshot and characterization
  - Stripe pattern analysis
  - Simulated block analysis and projection

- Available at: http://www.github.com/olcf/pcircle

This research used resources of the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725. The work at LC was performed under the auspices of the DOE by LLNL under Contract DE-AC52-07NA27344.