

Seeing the trees for the forest: Describing HPC Filesystem Trees with the Grand Unified File-Index (GUFU)

JENNA M. KLINE, The Ohio State University, USA

JASON LEE (ADVISOR), Los Alamos National Laboratory, New Mexico

RUSTY DAVIS (ADVISOR), Los Alamos National Laboratory, New Mexico

CCS Concepts: • Information systems → Directory structures.

Additional Key Words and Phrases: filesystem metadata management, high performance computing

ACM Reference Format:

Jenna M. Kline, Jason Lee (advisor), and Rusty Davis (advisor). 2023. Seeing the trees for the forest: Describing HPC Filesystem Trees with the Grand Unified File-Index (GUFU). In . ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

High performance computing (HPC) filesystems are extremely large, complex, and difficult to manage with existing tools. Los Alamos National Laboratory (LANL) maintains numerous such filesystems for different purposes, including home directories, project directories, scratch spaces, and archives. Using standard tools to analyze active filesystems is prohibitive due to time constraints, and analysis with non-standard tools designed for HPC filesystems requires subject expertise. Furthermore, previous studies of filesystem characteristics largely predate the modern HPC filesystems of the last decade [1–3]. This project presents a methodology to characterize and analyze modern filesystems as a first step in describing how HPC filesystems change over time. We use the Grand Unified File-Index (GUFU) [4] to quickly collect metadata for six HPC filesystems of varying types maintained by LANL, representing 2.8 PB of data, containing 36 million directories, and 600 million files. We present eight metrics developed by studying these six filesystems to help administrators understand key characteristics and provide a preliminary solution for how these metrics can be used by administrators.

2 METHODOLOGY

2.1 HPC Filesystems Analyzed

Six anonymized LANL HPC filesystems, listed in Table 1, were converted into GUFU trees and queried to gather metadata. Obtaining the metadata for each filesystem took an average of ten minutes. The traces *ttscratch* and *anony* were obtained from the same source filesystem, collected in April 2019 and April 2020 respectively. The remainder of the filesystem traces were collected once between 2019 and 2020. *scr3* and *scr4* are both scratch systems and are used to store intermediate job data. *yellprojs* is a project folder, and *yellusers* is a user directory. *ttscratch*, *anony*, *scr3*, and *scr4* are Lustre filesystems and *yellprojs* and *yellusers* are both NFS filesystems.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2023 Association for Computing Machinery.

Manuscript submitted to ACM

Table 1. LANL HPC Filesystems

Filesystem	Total Size	Directories Count	Files Count	Depth
anony	411 TB	7.4 M	157.6 M	37
scr3	975 TB	2.2 M	59.2 M	59
scr4	1 PB	5.1 M	118.9 M	37
ttscratch	423 TB	5.5 M	117.5 M	37
yellprojs	28 TB	14.1 M	133.3 M	112
yellusers	2 TB	1.6 M	12.9 M	48
Totals	2.8 PB	36 M	600 M	-

Table 2. Metrics to Describe HPC Filesystems

Metric	Short Name	Definition
Directory Avg File Count	$Density_d$	Average number of files in a directory by level
Subtree Avg File Count	$Density_s$	Average number of files in a subtree by level
Directory Avg Size by Level	$Weight_d$	Size (bytes) of a directories averaged by level
Subtree Avg Size by Level	$Weight_s$	Size (bytes) of subtrees averaged by level
Directory Width	$Width$	Total number of directories at each level
Tree Depth	$Depth$	Maximum level of a tree
Directory Sparsity	$Sparsity$	Count of empty directories and files in a tree
User popularity	$Popularity$	Count of unique user ids

2.2 Metrics to Describe HPC Filesystems

Using several parameters available from GUF1, we analyzed each filesystem from the perspective of directories and subtrees. Directory analysis includes only the files contained in the directory itself, but not in any of its child directories. Subtree analysis includes all data underneath the directory: the files immediately in the directory as well as the contents of all subdirectories. These parameters included the directory branching factor (number of immediate child directories), level, directory and file age, and directory and subtree size and file count. To distinguish file count from size, we refer to these measures as density and weight respectively, for both directories and subtrees. After a thorough analysis, we identified eight metrics that uniquely identify the shape of each filesystem, listed in Table 2. To visualize the unique shapes, we normalized the metrics for each filesystem and plotted the values on radar charts as shown in Figure 1.

3 RESULTS

One surprising result from our analysis was that the branching factor was not a meaningful distinguishing measure for HPC filesystems. We had hypothesized that the branching factor would contribute to the shape of the tree, however the branching factor quickly approaches zero after level 2 for these filesystems. The top twenty levels are most unique to each filesystem, as shown in Figure 2 and Figure 3, regardless of the overall depth of the filesystem. As shown in Figure 1, the general shape of *ttscratch* and *anony* remained consistent over time, although the levels in this filesystem widened as it aged. The radar charts also demonstrate that scratch directories, such as *scr3* and *scr4*, tend to be larger and denser than other filesystems. The user directory, *yellusers*, is characterized by a relatively high number of unique user ids, aka popularity, which is to be expected. *yellprojs*, by contrast, is characterized by its depth.

We created an add-on to GUF1 to allow system administrators, called Scorecard, to quickly generate a high-level view of a filesystem directory. This command line tool returns the analysis for depth, density, weight, width, sparsity, and

105 popularity for a filesystem by directory and subtree. For example, a subtree that is relatively wide with many small
 106 children utilizes a disproportionate amount of metadata storage, which may affect the performance of the I/O operations.
 107 We hope that this tool will be of use to HPC filesystem administrators.
 108

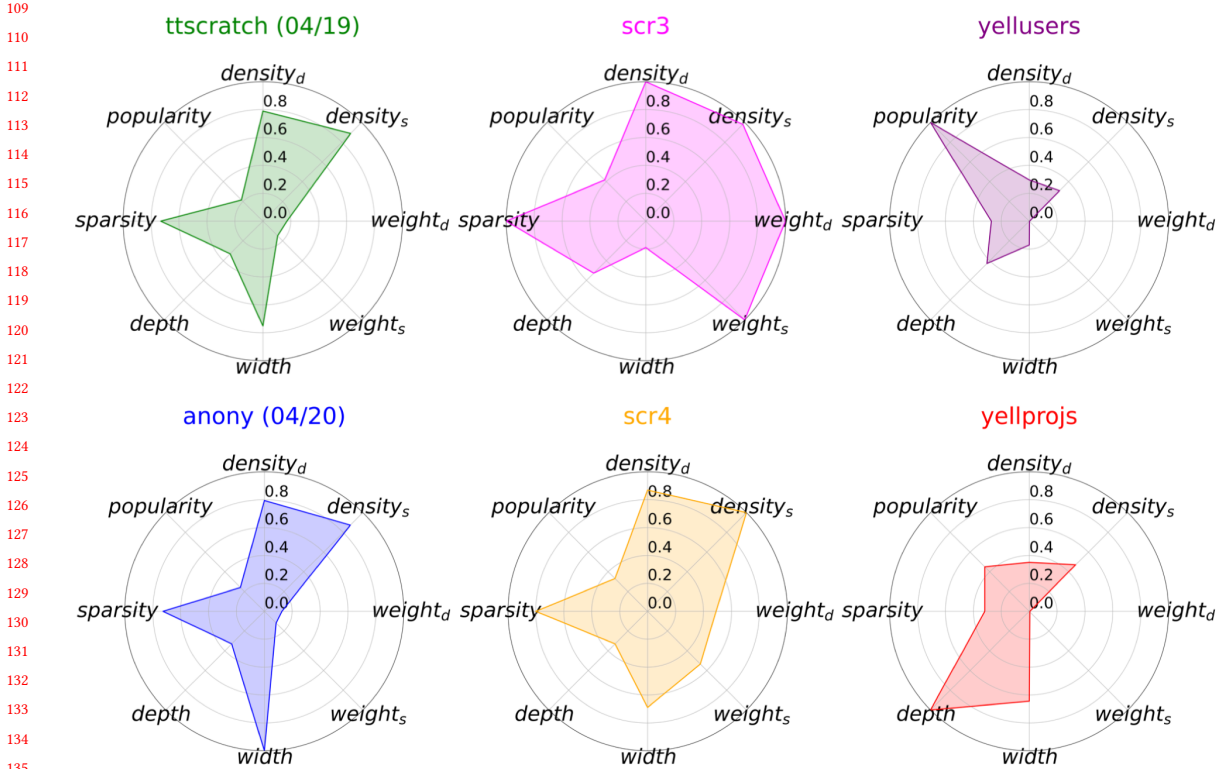


Fig. 1. HPC File System Trees Radar Charts

4 FUTURE WORK

141 Future work can utilize the Scorecard tool to analyze additional HPC filesystems. In addition, GUF can be used to build
 142 more complex metrics to describe filesystem trees, such as permission distribution. Our proposed methodology can be
 143 used to analyze past and future changes as HPC filesystems grow in size and complexity. The data from Scorecard can
 144 be integrated with machine learning techniques to categorize HPC filesystems and predict requirements for future
 145 systems.
 146
 147

5 ACKNOWLEDGEMENTS

150 This manuscript has been approved for unlimited release and has been assigned LA-UR-23-28957.
 151
 152
 153
 154
 155
 156

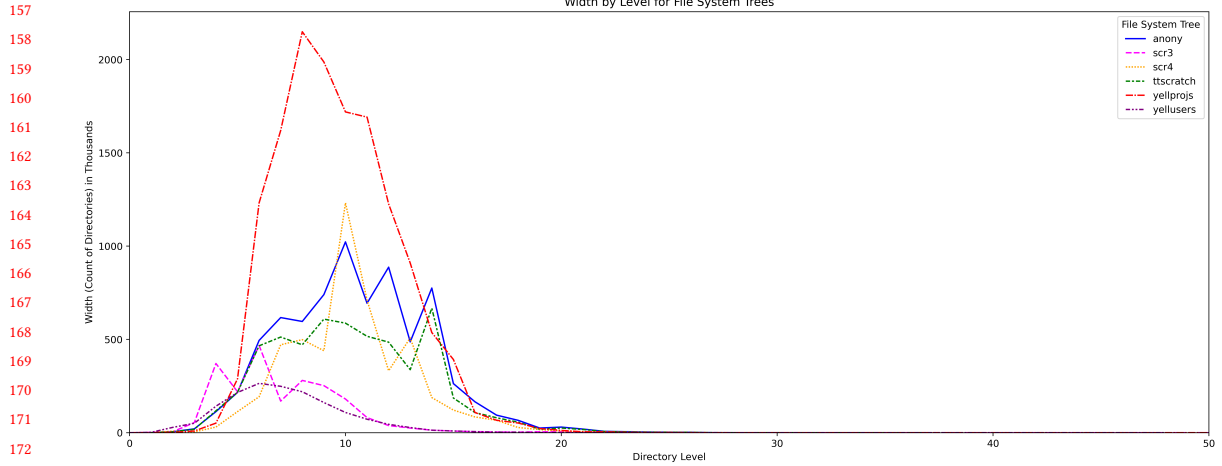


Fig. 2. Average Directory Width by Level

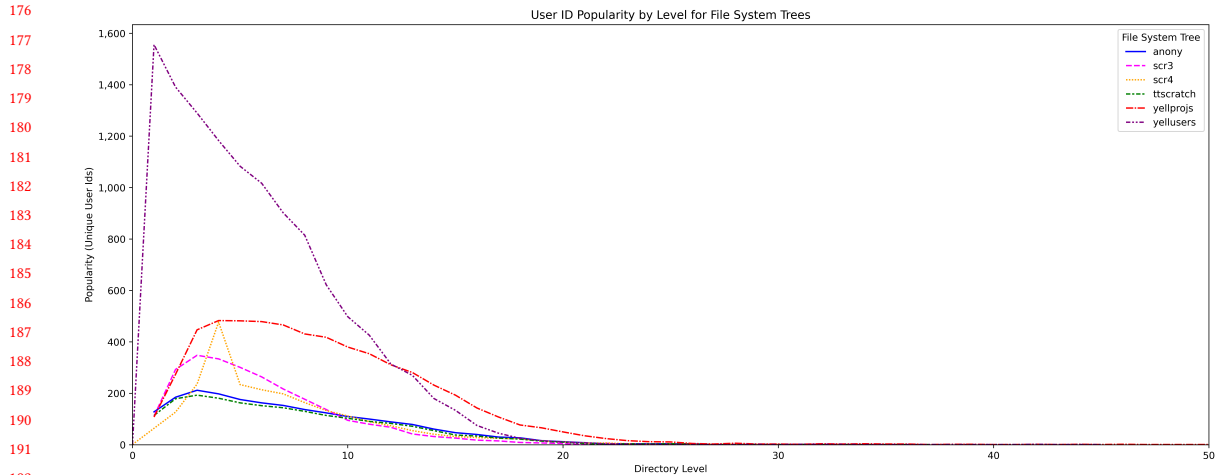


Fig. 3. Average Directory User ID Count by Level

REFERENCES

[1] Nitin Agrawal, William J. Bolosky, John R. Douceur, and Jacob R. Lorch. 2007. A five-year study of file-system metadata. *ACM Transactions on Storage* 3, 3 (Oct 2007), 9–es. <https://doi.org/10.1145/1288783.1288788>

[2] Shobhit Dayal. 2008. Characterizing HEC storage systems at rest. *Parallel Data Lab, CMU* (2008).

[3] Andrew W Leung, Shankar Pasupathy, Garth Goodson, and Ethan L Miller. 2008. Measurement and Analysis of {Large-Scale} Network File System Workloads. In *2008 USENIX Annual Technical Conference (USENIX ATC 08)*.

[4] Dominic Manno, Jason Lee, Prajwal Challa, Qing Zheng, David Bonnie, Gary Grider, and Bradley Settlemyer. 2022. GUFF: Fast, Secure File System Metadata Search for Both Privileged and Unprivileged Users. In *SC22: International Conference for High Performance Computing, Networking, Storage and Analysis*. 1–14. <https://doi.org/10.1109/SC41404.2022.00062>