A Case for Rigorous Workload Classification

Avani Wildani
avani@salk.edu
The Salk Institute

Ian F. Adams
ian.adams@seagate.com
Seagate Technology

***Abstract*—Traditional workload labels such as "archival" and "HPC" are poorly understood and inconsistently applied. As usage of systems has evolved, the language to describe this usage has stagnated. To better understand how workload type translates into system design requirements, we use a combination of longitudinal analysis and statistical feature extraction to categorize workload traces and study how the properties of classical workload types, such as the "write-once, read-maybe" assumption for archives, have evolved over time.**

Once this step is complete, we intend to move to active classification of workloads to replace these broad, poorly specified categories with quantitative metrics that can be used to improve metrics such as power, availability, and performance by mathematically relating storage algorithms with workload properties.

I. Introduction

Storage systems research is the study of trade-offs between security, reliability, availability, and performance. Many systems and algorithms in storage are designed with particular workload properties in mind that guide this tradeoff; exploiting properties that workload types are assumed to have while relaxing restrictions on features, such as strong metadata consistency, that particular workloads are assumed to not need due to their patterns of access. Currently, workloads are labeled with monikers such as "archival" or "enterprise" with only the loosest guidelines of what these terms translate to in terms of expectations and requirements. While systems have evolved and become increasingly multi-tenant and multi-application, our methods of categorization and labeling have been neglected. As a result, researchers are designing systems optimized for non-existent archetypes of workloads, making the translation of research between labs or from academia into industry even more difficult.

Consider, as an analogy, labels on beer. There are broad labels like "lager" and "stout", but within these is huge variation that relies on the ingredients and methods of production. For example, one stout may have chocolate, while another does not. They taste very different, but are both labeled as "stout". Similarly, in storage, the top level labels are useful for quick categorization, but more precision is needed to make good decisions. As a more concrete example, consider archival storage systems. Many academic "archival" storage systems assume that data is "write once, read-maybe/never" [17], [15], [21], but that assumption breaks down in long-tail workloads such as accesses to Facebook's image store [10] or government databases with frequent data migration [3]. This ambiguity makes it difficult to determine how a system will perform given a particular workload, *e. g.*, many archival systems rely on stereotypical properties. This issue is exacerbated by the shift towards shared storage systems, where many workload types may be interleaved, making any general categorization difficult, if not impossible.

Early results support our assertions that workloads with the same high-level classification can have different properties. We examine workload statistics across putative archival, enterprise, user, and high performance workloads, and demonstrate that the feature space is large, and even within like dimensions there is high variance even within features incident across workloads of the same type. These issues highlight the vital need for a *quantitative taxonomy* of storage workloads. That is, methods of numerically analyzing and qualifying workload labels and characteristics that are *consistent* and *comparable*. Until this is done, we are incapable of designing, or even validating, workload aware storage systems. In our work, we tackle two areas. First, we look at existing workload studies to show that our currently labeling and analysis approaches are inadequate for truly useful labels. Second, we propose a set of quantitative analyses techniques to both identify relevant features of workloads, and ultimately to characterize the interleaved workload components of a modern-scale multi-user, multi-application storage system.

II. Background and Related Work

Even from a qualitative standpoint, workload labels are, at best, vague. Consider the term "archival". Venti considered archiving to be more akin to long-running backups/versioning [15]. Adams *et al.* considered archives to be long-term historical and scientific data [3], while the authors of the Pergamum system considered archival data to have the "write-once, read-maybe" semantics typified by financial compliance data [17]. Similarly, Chen *et al.* showed that there is wide feature disparity in the space of user traces [4]. "HPC" workloads have transitioned from

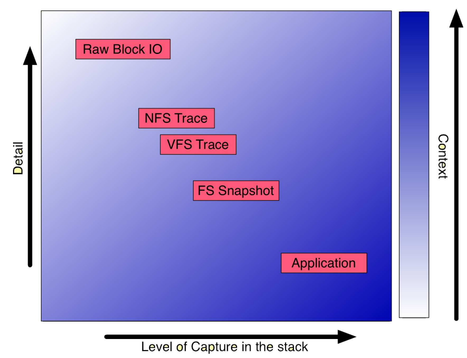


Fig. 1. Qualitative illustration of how traces gathered at different layers give us different insights into the system.

only performance constraints to both performance and power constraints [9]. Finally, Wildani *et al.* showed that workloads in a multi-use system may be interleaved [20].

Adding further complication to this issue is how the capture level of the trace influences the analysis. Traces from lower levels of the system stack often have a great amount of detail, but can be unwieldy and lack *context*, that is the semantic details behind what is driving a particular operation. Traces from higher in the stack (*e. g.*, application logs), are often rich in this context. They can give deep insight into the desired result from a user perspective, but lack details into how the lower level system is actually handling the request and propagating it through the system. Figure II illustrates. While it is important to understand systems from different perspectives, this makes it very hard, if not impossible, to compare and validate two systems that may be identical in workload, but captured data at different layers. Thus, the method and layer of capture is a critical aspect of any taxonomy and quantification.

This issue has been noted by others as well. Chen *et al.* pointed out that the mechanism for trace collection has an outsize impact on the eventual classification of the trace [4]. To take an extreme example, a database workload that is traced after all of the sequential accesses have been removed is essentially noise, whereas the same workload traced at a lower level of the storage stack shows a recognizable access pattern [6]. As such, the encoding of workload statistics must be level-aware; traces should only be compared against traces taken at similar levels unless a quantitative mapping exists between classes at different points in the storage stack.

While feature classification is a significant problem for localized storage, the same classification issues that plague even those simpler use-cases become compounded in shared cloud systems. In the cloud, and even in some single-use systems, many workload types may be interleaved, making any general categorization difficult with the currently available qualitative or single-descriptor vocabulary in systems for describing workloads. If workloads are described solely quantitatively, however, the problem of separating workloads becomes analogous to separating any set of signals that share a noisy channel, and thus becomes amenable to blind source separation techniques such as independent component analysis (ICA). The rigorous classification is necessary because, once the inputs are separated, retracing to understand what the workload was doing is nearly impossible, whereas classifying the workload based on its feature profile is a straightforward side effect of ICA.

Another feature of quantitative workload analysis is adaptability. Once we transition to a more granular, unified model of workload characterization, we will be able to better detect changes in the usage patterns of workloads as they inevitably shift over time. Cherkasova and Gupta [5] characterize the evolution of two enterprise workloads, and even with such a small sample set see significant variation over time in the number of unique clients and popularity of new files. If we replace a single workload label with a set of features, we can track shifts in features individually over time and adapt the system to best fit the workload.

Chen *et al.* [4] performed a feature-based analysis of enterprise storage traces to examine storage traces at multiple layers in the storage stack as well as to perform feature-based trace analysis. They claim that feature selection requires domain knowledge, which we disagree with based on the work of [14], which showed that features derived from one workload were generalizable to new systems that lacked domain-specific data. While they support a multi-variate analysis of workload features, the *k*-means analysis they perform makes several assumptions about the distribution of workload features that do not generalize to other datasets. This is a fundamental limitation of the *k*-means clustering algorithm [1]; we discuss alternative unsupervised learning methods in Section II-A.

A. *Methodologies*

We propose two methods that together will allow us to create a quantitative taxonomy of workloads. First, we are conducting a longitudinal meta-analysis of published workloads to categorize what features persist *between* workload types and track the evolution of statistics such as read/write ratios or user activity. We use historical data and citations to learn what features are most relevant when discussing traces, and are comparing traces from a variety of systems along these axes to formulate quantitative definitions and understand the feature variance for what are considered “archival,” “enterprise,” or “high performance” workloads.

Second, within and across the various *types* of

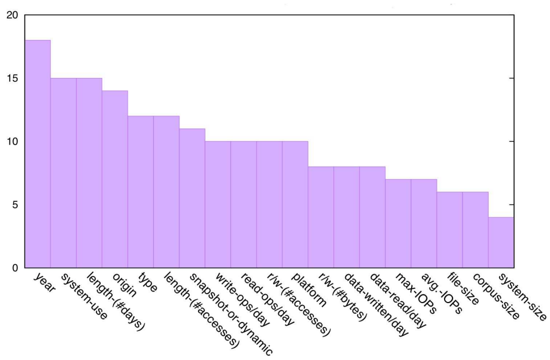


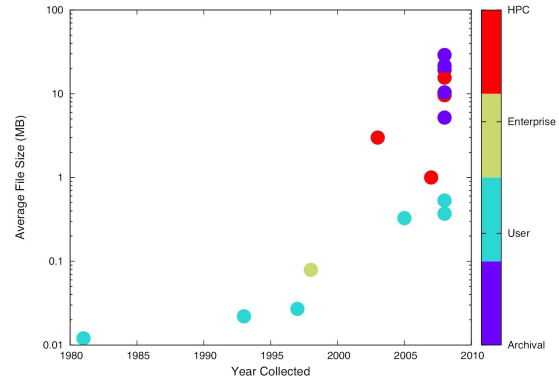
Fig. 2. Histogram of popular features collected from 29 workloads

TABLE I. COMMON WORKLOAD FEATURES

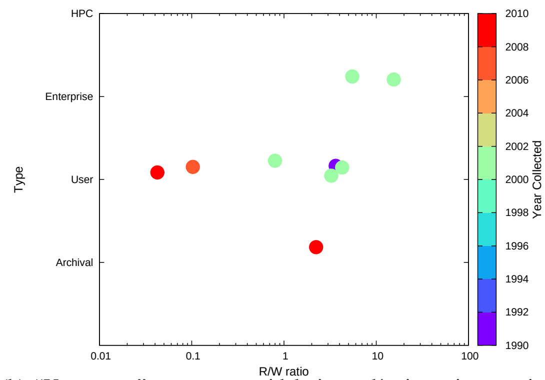
Type	Year	Avg.File Size (MB)	#Reads per day	#Writes per day
User [7]	1981	.012	–	–
User [16]	1991	–	207000	57000
User [7]	1993	.022	–	–
User [7]	1997	.027	–	–
User [16]	2000	–	303000	71000
User [8]	2001	–	350000	438000
User [8]	2001	–	19290000	5930000
User [7]	2005	.327	–	–
User [12]	2007	–	14847	144447
User [7]	2008	.531	–	–
User [7]	2008	.37	–	–
User [11]	2010	–	34593.52	814773.19
Enterprise [16]	2000	–	1270000	231000
Enterprise [16]	2000	–	2320000	150000
Enterprise [7]	2008	19.3	–	–
HPC [18]	2003	3	–	–
HPC [3]	2007	1	–	–
HPC [7]	2008	10.3	–	–
HPC [7]	2008	15.6	–	–
HPC [7]	2008	9.6	–	–
Archive [7]	2008	29.0	–	–
Archive [7]	2008	21.7	–	–
Archive [7]	2008	10.4	–	–
Archive [7]	2008	5.2	–	–
Archive	2010	–	1326137	595661

workloads, we explore the possibilities for common metrics and terminology. While current qualitative labels are convenient for discussion, we extend these types through statistical analysis and machine learning, particularly using blind source separation and feature selection techniques to classify and separate workload streams. A multi-user, multi-application workload is similar to a noisy party: there are many different "conversations" happening in the room, and we would like to separate the speakers out by features such as age, gender, or volume. Independent components analysis isolates individual non-Gaussian signals within a shared data pipeline by selecting a set of candidate signals and then minimizing the mutual information across said signals.

In addition to a machine-learning based feature selection, we can use domain information to better understand what features are most critical for workload classification. Figure 2 is an incidence histogram of the



(a) "Archival" traces from the same year have a wide variation in average file size.



(b) "User space" traces vary widely in read/write ratio over time (jitter added to y-axis)

Fig. 3. Trace features vary both over time and across types.

most prevalent features from 29 traces that we gathered details of from previous workload studies. Table I lists relevant details of these workloads. Based on the feature distribution, we can conclude that reported features that are easy to collect, such as year and origin, are likely to be prominent, followed by basic statistics such as daily reads and writes. However, of the trace analyses we studied, the vast majority of features reported were only reported for one trace in our sample, indicating that there is a need for a common, continuous language for describing workload statistics.

Another technique for feature identification to pursue would be to track what features of a published trace followup studies rely on when citing the work. The concept is similar to web ranking [13], where repeated references to a particular concept indicate that that concept is a good "keyword" or descriptor for a document, which in this case is a proxy for the workload the document describes.

III. ANALYSIS

Figures 3(a) and 3(b) show the discrepancy between the descriptors used for storage workloads and the characteristics of the workloads themselves. The meaning of terms changes both over time and across different instantiations. For example, several "archival" datasets collected in the same year (Figure 3(a)) show vastly different average file sizes. Similarly, designing a workload in user space that relies on the balanced read/write ratio of older workloads is a mistake in modern systems (Figure 3(b)), but the terminology describing the workload remains unchanged.

There are a number of reasons why it is difficult to compare and categorize workloads based on aggregate statistics. Captured workload data are often based on the requirements of the local administrator, and thus it is highly variable between studies and the presence of particular features is not necessarily correlated with feature importance.

While it is unsurprising that the definitions of trace types has changed over time, the variation in "archival" or "user" workload characteristics between traces collected in the same year is a strong indication of the need to create a more portable and permanent descriptive language to classify workloads. Additionally, the diversity of features reported in the trace studies that we examined indicates that trace classification will be both high dimensional and sparse.

IV. Conclusion

In this work, we have argued that the trace characterization currently used by the storage community is flawed and lacking in rigor. Workloads with the same classification vary widely, and there is no consensus on what features are important to report when categorizing workloads. Additionally, workload characteristics differ based on where the tracing is instrumented.

Having a rigorous, extensible, easily communicable workload characterization will provide several benefits to the systems community. The characterization will provide a metric for communicating precise requirements, helping designers architect systems that are well adapted for the expected workload. Additionally, dynamic categorization of workloads will allow storage providers to develop better SLAs that tie performance expectations to storage characteristics. Finally, we propose that a quantitative characterization is the first step towards more realistic workload simulation and modeling. Currently, we are working to automate feature extraction from static and dynamic workload traces. To date, we have shown that static workload traces can be clustered usefully by features [19], and have evidence that these features are identifiable with component analysis. This is critical because organizations are much more comfortable sharing static snapshots than they are full traces, and the more data we can integrate into our analysis the stronger it will be.

References

[1] "K-means clustering is not a free lunch http://varianceexplained.org/r/kmeans-free-lunch/," Jan 2015.

[2] I. F. Adams, M. W. Storer, and E. L. Miller, "Analysis of workload behavior in scientific and historical long-term data repositories," *ACM Transactions on Storage (TOS)*, vol. 8, no. 2, p. 6, 2012.

[3] Y. Chen, K. Srinivasan, G. Goodson, and R. Katz, "Design implications for enterprise storage systems via multi-dimensional trace analysis," in *Proc of SOSP '11*, 2011.

[4] L. Cherkasova and M. Gupta, "Characterizing locality, evolution, and life span of accesses in enterprise media server workloads," in *Proc. of NOSSDAV '02*, 2002.

[5] C. Constantinescu, J. S. Glider, and D. D. Chambliss, "Mixing deduplication and compression on active data sets," in *Proc. DCC '11*, 2011.

[6] S. Dayal, "Characterizing hec storage systems at rest," *Parallel Data Lab, CMU*, 2008.

[7] D. Ellard, J. Ledlie, P. Malkani, and M. Seltzer, "Passive nfs tracing of email and research workloads," in *Proc. of FAST '03*, pp. 15–15, USENIX Association, 2003.

[8] C.-H. Hsu and W.-C. Feng, "A power-aware run-time system for high-performance computing," in *Proc. of SC '05*.

[9] Q. Huang, K. Birman, R. van Renesse, W. Lloyd, S. Kumar, and H. C. Li, "An analysis of facebook photo caching," in *Proc. of SOSP '13*, SOSP '13, ACM.

[10] R. Koller and R. Rangaswami, "I/o deduplication: Utilizing content similarity to improve i/o performance," *ACM TOS*, vol. 6, no. 3, p. 13, 2010.

[11] D. Narayanan, A. Donnelly, and A. Rowstron, "Write off-loading: Practical power management for enterprise storage," *ACM TOS*, vol. 4, no. 3, p. 10, 2008.

[12] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web," tech. rep., Stanford, 1999.

[13] R. Powers, M. Goldszmidt, and I. Cohen, "Short term performance forecasting in enterprise systems," in *Proc. of SIGKDD '05*.

[14] S. Quinlan and S. Dorward, "Venti: a new approach to archival storage," in *Proc. of FAST '02*, 2002.

[15] D. S. Roselli, J. R. Lorch, T. E. Anderson, *et al.*, "A comparison of file system workloads.," in *USENIX '00*, pp. 41–54.

[16] M. W. Storer, K. Greenan, E. L. Miller, and K. Voruganti, "Pergamum: Replacing tape with energy efficient, reliable, disk-based archival storage," in *Proc. of FAST '08*, 2008.

[17] F. Wang, Q. Xin, B. Hong, S. A. Brandt, E. L. Miller, D. D. E. Long, and T. T. Mclarty, "File system workload analysis for large scientific computing applications," in *Proc. of MSST '04*.

[18] A. Wildani, I. F. Adams, and E. L. Miller, "Single-snapshot file system analysis," in *Proc. of MASCOTS 2013*, Aug. 2013.

[19] A. Wildani, E. L. Miller, and O. Rodeh, "Hands: A heuristically arranged non-backup in-line deduplication system," in *Proc of ICDE '13*, 2013.

[20] L. You, K. Pollack, and D. D. Long, "Deep store: An archival storage system architecture," in *Proc. of ICDE '05*, 2005.