Chasing the Signal: Statistically Separating Multi-Tenant I/O Workloads

Si Chen
Department of Computer Science
Emory University
Atlanta, GA, 30322
si.chen2@emory.edu

Avani Wildani *
Department of Computer Science
Emory University
Atlanta, GA, 30322
avani@mathcs.emory.edu

Abstract

Identifying the characteristics of a storage workload is critical for resource provisioning for metrics including performance, reliability, and utilization. Although multi-tenant systems are increasingly commonplace, characterization of multiple workloads within a single system trace is difficult because workloads are highly dynamic and typically not labeled. We show that, by converting a block I/O workload to a signal and applying blind source separation, we are able to successfully separate many application workloads.

1 Introduction

Tuning a storage system is a difficult, delicate balance of reliability, availability, security, and performance spread over the workloads that the system serves. For characterization, we define a *workload* as a functionally distinct usage of the system.

As storage systems increase in size, multi-tenancy, or multiple workloads sharing the same space, has become commonplace. As a result of normal OS processes such as batching or CPU scheduling, I/Os from different workloads tend to appear interleaved when viewed in a trace. In a recent study, 60% of traces recorded were strided and composed of interleaved workloads [8].

An interleaved storage workload could arise from something as simple as a small organization running both a database and a web-server simultaneously on the same hardware. Other common instances of this *functional multi-tenancy* include co-locating a set of VMs or balancing several users using a service on the same hardware such as for streaming services. Part of provisioning a storage system is understanding how to balance the unique needs of multiple clients with individual service level agreements (SLAs). SLAs are hard to consistently meet when it is difficult to isolate a client's activity. Moreover, a single client may have multiple, functionally distinct workloads that are not rigorously defined [1]. Though projects such as Crystal [3] discuss how to implement multi-tenant solutions in the presence of multiple workloads, to the best of our knowledge no one has studied separating workloads by functional qualities.

Block I/O traces are commonly available measures of workloads because they are relatively simple to instrument and do not incur the storage and privacy concerns of traces from higher in the stack. If a trace is translated into a signal by binning across the spatial dimension, the problem of separating workloads becomes analogous to separating any set of signals that share a noisy channel. We thus posit that they are amenable to blind source separation techniques such as independent component analysis (ICA) that traditionally are used for signal separation.

*Corresponding Author

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

2 Methods

A multi-user, multi-application workload is similar to a noisy party: there are many different "conversations" happening in the room, but we are recording them with a single microphone and we'd like to know who said what. Methods originating in signal processing have shown considerable promise in systems data analysis. Separating out signals by visually examining trace logs is very difficult, as I/O contention and system activity results in the individual workloads being mixed even more thoroughly than our cocktail-party example.

Blind source separation (BSS) is, very generally, trying to answer the following question: if the components of an observed p-variate vector x are linear combinations of the components of a latent unobserved p-variate source vector y, and one can write $x = \Omega y$ where Ω is an unknown full-rank mixing matrix, can we derive Ω based on a set of observations $x_1, \ldots, x_n$ [7].

We begin with the hypothesis that functional workload signals are non-Gaussian, which our preliminary results (Figure 1) support. Blind source separation techniques like independent components analysis (ICA) isolate 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. This is accomplished through measuring signal kurtosis and using non-Gaussianity as a proxy for independence [4]. ICA has been used successfully to separate other dependent parallel time series such as financial data [4] and EEG recordings [10], providing sound evidence that ICA could be used to separate storage workloads as a precursor to identification. ICA takes the number of source signals as a parameter; while in this work we provide a source number as input, techniques exist to estimate this number from an arbitrary mixed workload trace [2].

2.1 Creating the source signal

Without a standard definition of workloads, we tentatively use PID as a workload identifier. In this way, we can transfer one single trace (with the metrics of PID and corresponding logical block address) into several sub channels of workload signals. The mixing matrix can be thought of as simulating I/O contention and scheduling issues between the workloads. Since the ICA mixing process is that the N observations are a linear combination of the N sources. Using the vector-matrix notation, the ICA mixing model can be written as $x = As$, where x is the observation with N vectors, s is the source with N vectors, and A is the $N * N$ mixing matrix. The goal of ICA is to find matrix W (the inverse of A) that will reconstruct the source s', which satisfies $s' = Wx$.

While solving the ICA problem, we should keep in mind of two kinds of ambiguities[4] . First, ambiguity comes from the magnitude and scale. Since both s and A are unknown, scalar k multiplied in s can be counteracted by dividing the corresponding a_i by the same k. This gives us an uncertain solution. Second, the order of the independent components is not fixed because if a permutation matrix P is multiplied to s, since $x = AP^{-1}Ps$, the mixing matrix became $W = AP^{-1}$. Under either circumstance, directly calculating the Mean Square Error (MSE) by comparing the generated s' and source s is not appropriate. However, the first problem can be simply overcome using unit variance $E\left\{s_i^2\right\} = 1$. While this still leaves the sign ambiguity, we set the absolute value to the outcomes. The Second ambiguity can also be eliminated by trying every possible signal order during calculation. With the above processing, the MSE could be used to test the performance of ICA algorithms.

2.2 BSS Techniques

The FastICA algorithm is a generative model that separates out the non-Gaussian independent components using an iterative Gram-Schmidt-like decorrelation to find the weight vectors of the unmixing matrix that will return the component signals. Second-order source separation was created by the signal processing community to address situations with significant correlation between samples, which is more representative of workloads contending for the same hardware resources. AMUSE [9] (Algorithm for Multiple Unknown Signals Extraction) is a standard second-order technique that uses covariances and autocovariances with different lags $\tau = 1, 2, \ldots$ of an observed multivariate time series. Second-Order Blind Identification (SOBI) relies on second-order statistics to explode the time-correlation structure assumption of the signals. Finally, Joint Approximation Diagonalization of Eigenmatrices (JADE) uses target matrices as the fourth order cumulants of whitened signals.

Other techniques, in particular Dependent Component Analysis (DCA), were considered but were not appropriate for the size of our data. While DCA allows dependence within the components group, it still requires mutual independence between the groups. In our case, this flexibility benefits little because and some ICA algorithms could achieve local even global consistency.

3 Results

We experimented with block I/O traces collected at Florida International University [5]. Each I/O is represented by a tuple of `<Time,PID,LBA,size in 512 Bytes blocks,Write or Read>`. For the sake of demonstration, we use PID as a ground truth indicator of workload label. To further simplify the problem, we removed accesses that did not correspond to one of the top 10 PIDs. All source traces were converted to signals, centered, whitened, and mixed using a mixing matrix of full rank. To validate, we calculate MSE between the recovered signals and the true source signals.

3.1 Signal Quality

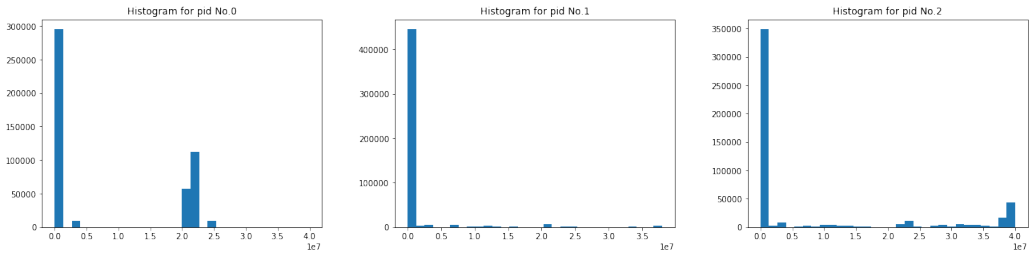

Figure 1: These sample histograms of PID incidence over time show that these signals are non-Gaussian. Kurtosis values for the first 3 PIDs are -1.67, 24.6, and .303, respectively.

Figure 1 shows the non-Gaussianity of our data. All other PIDs we tested had similar distribution.

Table 1: Sample PID Correlations

`PID 1`	`PID 2`	Correlation
1	2	-0.4679
1	3	-0.1701
1	4	-0.1813
1	5	-0.1204
3	5	-0.0189

We calculate correlation (Table 1) to measure the independence of our PID-derived signals.

3.2 Signal Separation

Table 2: Mean Squared Error for BSS

Source PIDs	**FastICA**	PCA	Amuse	Jade	Sobi
1,2	0.6975	0.7464	0.4445	0.1345	0.4302
1,3	0.347	0.5618	0.1339	0.016	0.178
1,4	0.3446	0.5647	0.1146	0.0115	0.0232
0,2	0.8218	0.618	0.9403	0.3183	0.3359
1,2,3	0.4579	0.7203	0.2352	0.0969	0.2857
0,1,5	0.5779	0.8699	0.3018	0.0077	0.0142
1,3,5	0.2289	0.531	0.1999	0.01	0.0192
1,2,3,4,5	0.2708	0.9787	0.5867	0.0634	0.1974

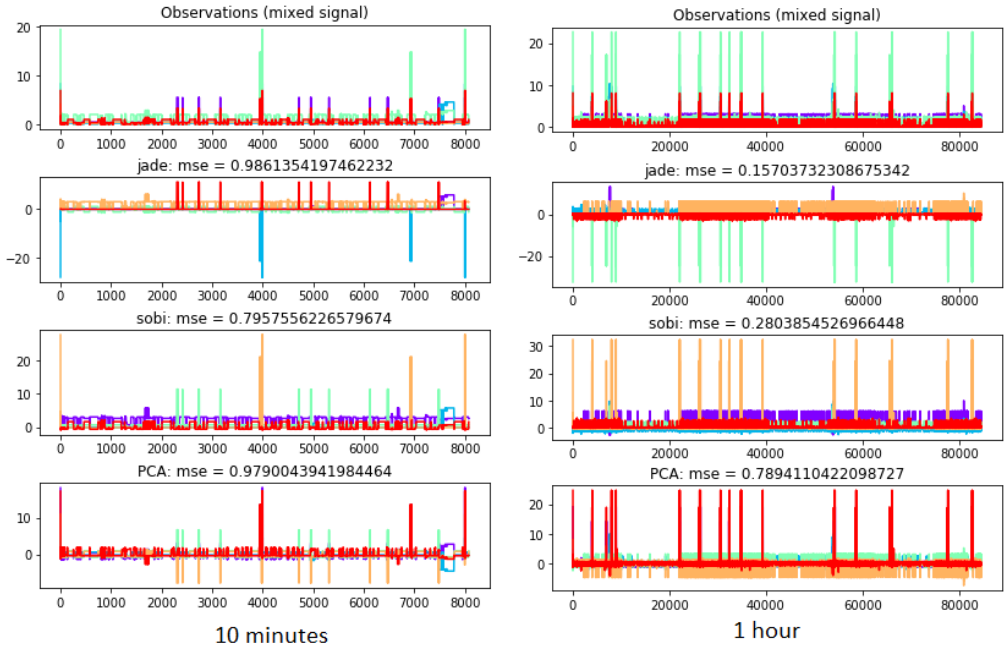

Figure 2: Five application traces [6] were separated with BSS techniques and PCA. Both visual examination and the MSE that the component signals returned by BSS techniques are closely resemble ground truth (note: sign is arbitrary). The PCA signals are much less accurate. Also, while MSE was lowest for a long trace(the y axis represents frequency, and the x axis is time), even the first ten minutes of I/Os led to low MSE for BSS.

Table 2 shows the MSE for separating different mixtures of source PIDs. Jade, based on higher-order statistic, outperforms all other BSS techniques. To put the BSS techniques we tried in context, we also calculated the MSE for signals returned from principal components analysis (PCA). We see, as expected, the MSE is considerably higher for PCA because PCA only has the assumption of orthogonality between signals, while ICA does not.

All of the MSE values in Table 2 are calculated using a week of trace data. Figure 2 shows that even ten minutes of source data is sufficient to de-interleave trace data in the workloads we examined.

4 Discussion

A major issue with using ICA for this type of analysis is the inherent assumption that the instances (traces) that we use to separate workloads are iid. Since workloads are by the nature of the problem description running on the same storage system, there will be dependence in a trace as a result of resource contention in the underlying architecture. Additionally, while PID is not a perfect indicator of workload(because of variety in long time slot), we are specifically looking for functional groups of signals: it is fine if one "signal" corresponds to multiple users or server types if they behave similarly.

Block I/O traces are generally possible to collect, but for HPC systems and other applications where dynamic trace collection is infeasible, it may be possible to separate interleaved workloads by clustering features in metadata snapshots and then analyzing with a visual classification model.

In conclusion, we have demonstrated that, surprisingly, mixes of I/O workloads are separable using BSS techniques. Even though they run on the same hardware, PID signals are independent and leveraging the excess kurtosis shows very low error values. While our sample data here is limited, it is collected from a heterogeneous group of users and applications, indicating that it will be noisier than datasets from industry. Our next step is to test statistically separation on more datasets without filtering noisy PID signals. Once we could retrieve the split trace by the split signals, we can implement duplication and grouping data on disk to avoid unnecessary disk activity, thus improving SLAs.

References

[1] Maureen Chesire, Alec Wolman, Geoffrey M Voelker, and Henry M Levy. Measurement and analysis of a streaming media workload. In *USITS*, volume 1, pages 1–1, 2001.

[2] Pando Georgiev, Fabian Theis, and Andrzej Cichocki. Sparse component analysis and blind source separation of underdetermined mixtures. *IEEE Transactions on Neural Networks*, 16(4):992–996, 2005.

[3] Raúl Gracia-Tinedo, Josep Sampé, Edgar Zamora, Marc Sánchez-Artigas, Pedro García-López, Yosef Moatti, and Eran Rom. Crystal: Software-defined storage for multi-tenant object stores. In *Proceedings of the 15th Usenix Conference on File and Storage Technologies*, pages 243–256. USENIX Association, 2017.

[4] Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. *Independent component analysis*, volume 46. John Wiley & Sons, 2004.

[5] R. Koller and R. Rangaswami. I/o deduplication: utilizing content similarity to improve i/o performance. *ACM Transactions on Storage (TOS)*, 6(3):1–26, 2010.

[6] Ricardo Koller and Raju Rangaswami. I/o deduplication: Utilizing content similarity to improve i/o performance. *ACM TOS*, 6(3):13, 2010.

[7] Jari Miettinen, Klaus Nordhausen, Hannu Oja, and Sara Taskinen. Statistical properties of a blind source separation estimator for stationary time series. *Statistics & Probability Letters*, 82(11):1865–1873, 2012.

[8] Bumjoon Seo, Sooyong Kang, Jongmoo Choi, Jaehyuk Cha, Youjip Won, and Sungroh Yoon. Io workload characterization revisited: A data-mining approach. *IEEE Transactions on Computers*, 63(12):3026–3038, 2014.

[9] Lang Tong, VC Soon, YF Huang, and RALR Liu. Amuse: a new blind identification algorithm. In *Circuits and Systems, 1990., IEEE International Symposium on*, pages 1784–1787. IEEE, 1990.

[10] Ricardo Vigário, Jaakko Sarela, V Jousmiki, Matti Hamalainen, and Erkki Oja. Independent component approach to the analysis of eeg and meg recordings. *IEEE transactions on biomedical engineering*, 47(5):589–593, 2000.