Clustering Workloads For Storage Optimisation

Vishwanath Seshagiri† Abigail Juilan† and Avani Wildani†

†Department of Computer Science, Emory College, Emory University

Summary

- Companies today have huge storage systems that support the needs of millions of businesses and individuals across a wide range of unique workloads
- If there were an accurate way of clustering workloads to optimise the configuration of underlying storage systems, there would be both economic and performance benefits in storage systems receiving these workloads
- Current approaches to characterizing workloads apply labels based on the source of the trace, but this is not representative of the features of the workload as workloads from multiple sources can have similar characteristics.
- We extracted relevant features from the traces, and performed Agglomerative Clustering on those to cluster the workloads to achieve better cumulative hit rate.
- In MSR Traces, we achieved a 20% better hit rate when compared to clustering using Source labels. Also, one of the clusters generated reduced the number of storage devices from 3 to 2.

Background / Motivation

- The Workloads from multiple sources, can be similar or complementary to each other, and this property of workloads needs to be maneuvered to come up with better storage systems.
- Prior Works such as GraphLens[3] have focused on extracting insightful information and other works[2] which have made a case for more rigorous workload characterisation.
- We build our work on these works, to use insights and generate more appropriate features for clustering the work
- Storage Clustering methods such as RAID are very coarse, and are designed independent of the characteristics of the workload. Hence, it is only natural that we move towards Storage Clusters which incorporate workload characteristics for utilising the storage device better.

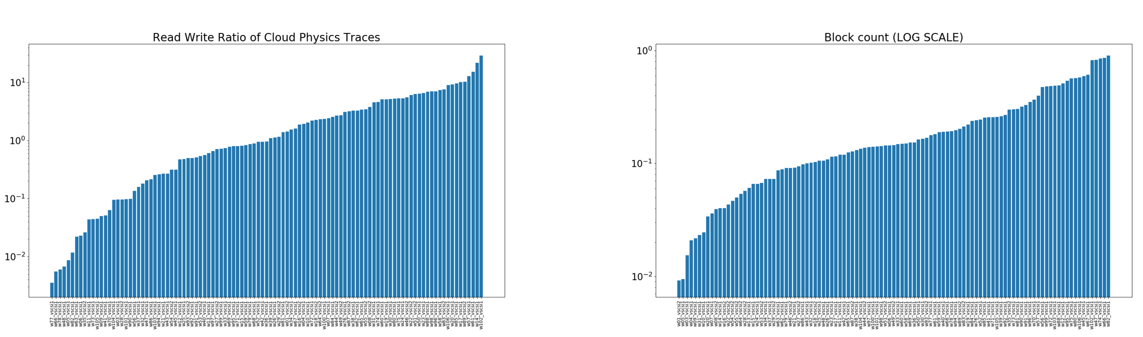

Fig. 1: The unique block count ratio, and read write ratio of Cloudphysics traces.

- Figure 1 shows ratio of unique block counts, and ratio of reads and writes for each trace file in Cloudphysics dataset.
- The traces are similar in terms of the read write ratio, and the number of unique blocks they access.

Data Analysis

- We analysed Block Level Traces from Production Servers at Microsoft Research and Cloudphysics[1] for performing our experiments.
- When checking for the Interarrival times, Some servers have dense distributions, meaning that single server alone is almost constantly in use, and other
- For other servers, there were large gaps in inter-arrival times, meaning the server would not be in use except for certain bursts of activity
- This suggests that potentially clustering these workloads with large gaps could be a better use of storage space (Figure 2)

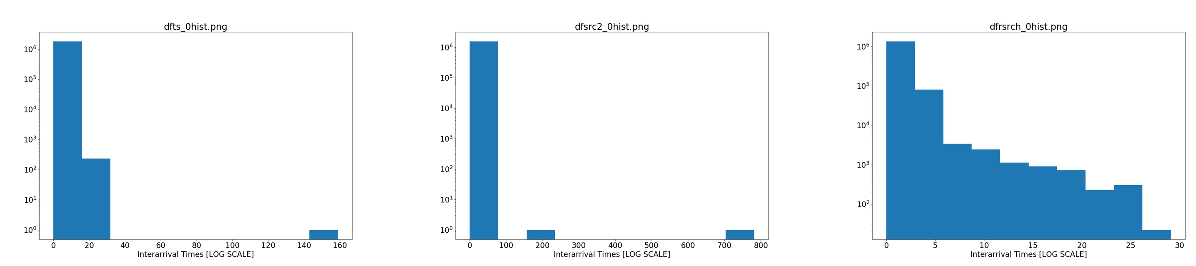

Fig. 2: Interarrival times of 3 servers in MSR Dataset(LOGSCALE)

- Cloudphysics traces also have servers which can be clustered together to optimise the use of storage space in each cache layer (Figure 3)

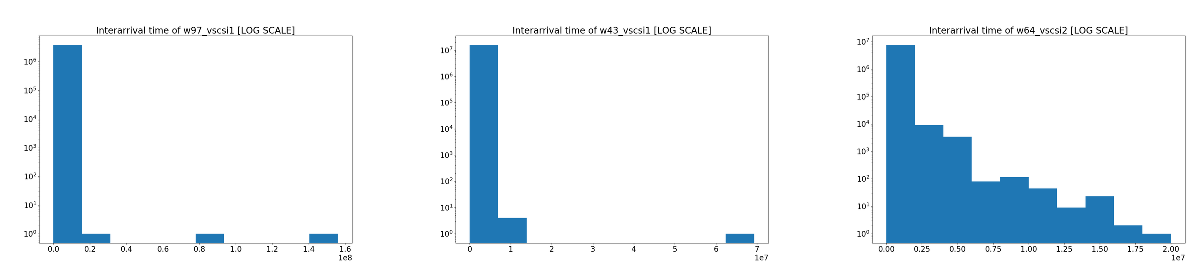

Fig. 3: Interarrival times of 3 workloads in Cloudphysics Trace(LOGSCALE)

- Another important factor of the load is the size of each request
- Looking at the same servers as we did for interarrival times, we see a similar pattern of gaps and peaks (Figure 4)
- This suggests that load size differences could be promising in the clustering of workloads
- This information can be leveraged when applying cache pre-fetching techniques as we will have a rough idea of the size of blocks to be retrieved for each request

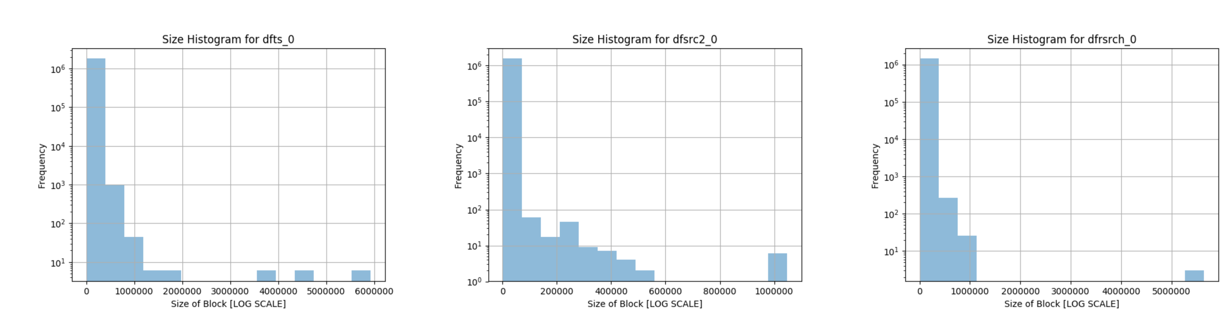

Fig. 4: Size of blocks requested (LOGSCALE)

- These analyses show that multiple workloads can be clustered together onto a single cluster to optimise the Cache Hit Rate

Results

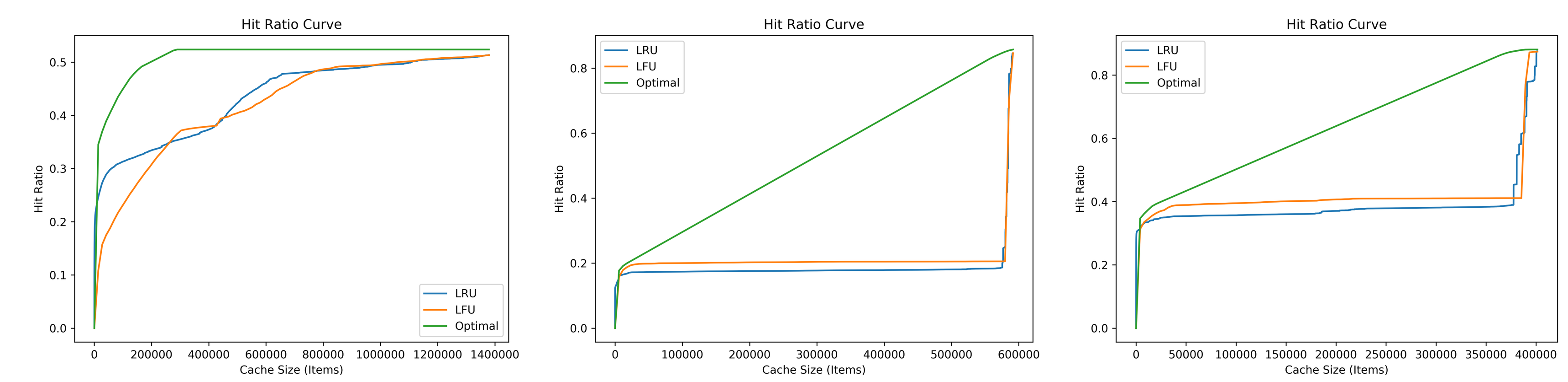

Fig. 5: Cache Hit Rate of Requests in Research, Source Control 2, and Terminal Server respectively.

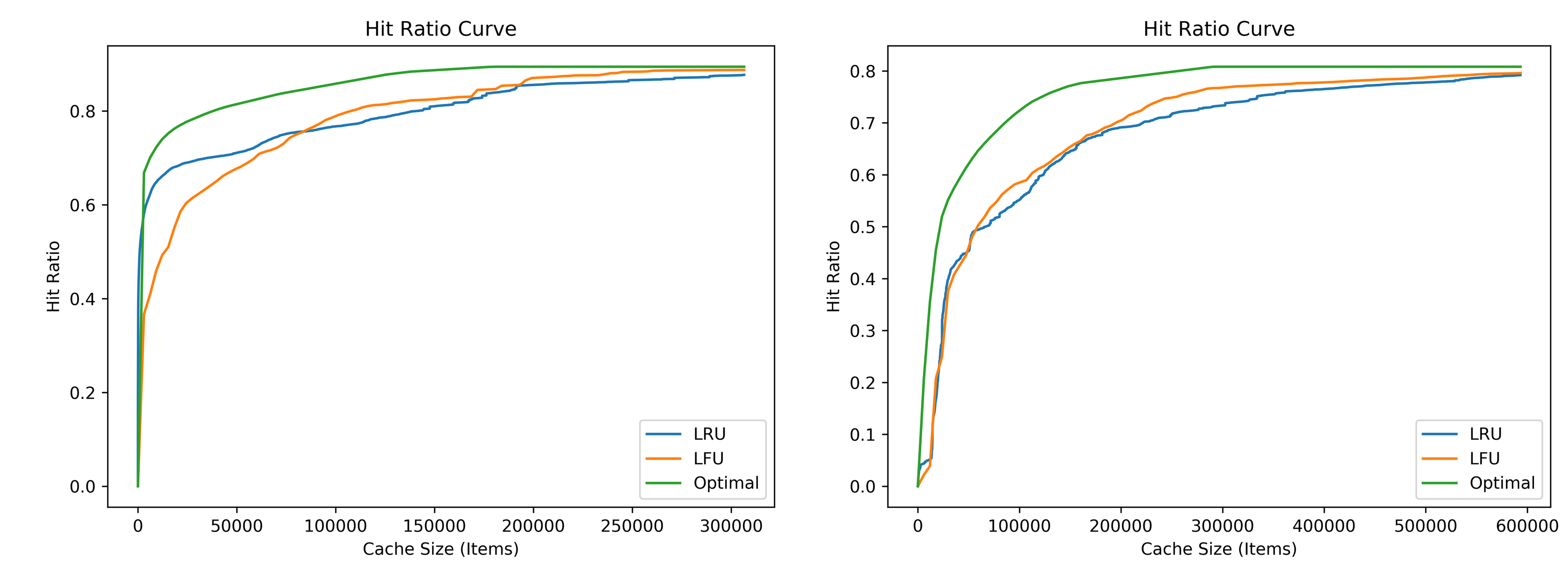

Fig. 6: Cache Hit Rate of Requests in Research, Source Control 2, and Terminal Server respectively.

- We used the Hit Rate Curves to analyze the effectiveness of Cache Provisioning
- Baseline Results (Figure 5): The HRCs for the individual sources isn't the most optimal, with Hit Rates reaching 50% only when using really large cache sizes
- Improved Results (Figure 6): The HRCs for clustered workloads show more effective cache utilization since the access of disks is now by more workloads with similar requirements

Future Work

This study paves way for multiple tangents, including but not limited to the following:

- A feature analysis study of different characteristics of a workload which can be effective in efficiently determining their nature.
- An extension to this study by coming up with better clustering frameworks to produce better results.
- Study of how a workload adapts to multiple replacement policies to determine which replacement policies to employ.
- Analysing the workloads will also help in cache pre-fetching as it can help us estimate the blocks
- Come up with a new clustering framework for identifying similar workloads and the size of cache required to cache them

[1] Carl A. Waldspurger et al. "Efficient MRC Construction with SHARDS". In: *13th USENIX Conference on File and Storage Technologies (FAST 15)*. Santa Clara, CA: USENIX Association, Feb. 2015, pp. 95–110. ISBN: 978-1-931971-201.

[2] A. Wildani and I. F. Adams. "A Case for Rigorous Workload Classification". In: *2015 IEEE 23rd International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*. Oct. 2015, pp. 146–149. DOI: 10.1109/MASCOTS.2015.32.

[3] Y. Zhou et al. "GraphLens: Mining Enterprise Storage Workloads Using Graph Analytics". In: *2014 IEEE International Congress on Big Data*. June 2014, pp. 1–8. DOI: 10.1109/BigData.Congress.2014.11.