

# Quantifying Temporal and Spatial Localities Due to File System Caching

Cory Fox, Dragan Lojpur, and An-I Andy Wang  
*Florida State University, {fox, lojpur, awang}@cs.fsu.edu*

Modern storage consists of many optimizations along the data path, from the front-end file system cache and disk layout management to the back-end disk controller cache and on-disk caching. Various components generally exploit the temporal and spatial localities of a workload to improve performance. However, how localities of a workload are transformed by each optimization is not well understood, resulting in designs that are more reflective of the understanding of the front-end workload than that of the locality characteristics immediately before the component.

This research studies how file system caching transforms temporal and spatial localities. This understanding helps us quantify how before-cache workloads (e.g., references sent from an application to the OS) properly stress after-cache components (e.g., disks), how synthesized before-cache workloads have different after-cache effects than original workloads from which they are derived, and what opportunities remain for further after-cache optimizations.

Quantifying localities is challenging for three reasons: (1) File system caching induces system traffic due to prefetching, buffered writes, page replacements, metadata accesses, and system events, which are sensitive to time and memory resources. Therefore, accesses before cache do not have one-to-one mapping to accesses after cache. (2) The semantics of locality depends on the granularity of analysis. Accesses before cache are in files, directories, and links. Accesses after cache are in blocks. Locality computed based on the distance between file references tends to be poorer than that based on blocks, since blocks tend to be referenced sequentially within files. (3) Locality metrics need to be comparable across workloads and environments. A workload with a “90% spatial locality” on a 50-GB drive should show meaningful behaviors when being applied to a 100-GB drive.

Existing research evaluates caching, temporal locality, and spatial locality independently, with certain limitations. The cache-hit-rate metric [3] is used to compare caching policies without showing how workloads are transformed. Stack and block distances [1] can measure how localities are transformed by caches, but these metrics are highly sensitive to system settings, making results difficult to compare across workloads and environments. Some studies evaluate the effects of caching algorithms and cache sizes on the reference stream, but often exclude system traffic from analyses [2].

We propose two metrics—temporal and spatial affinities:

$$\text{temporal\_affinity} = \frac{1}{\log_{10}(10 + \text{reference\_dist})}$$
$$\text{spatial\_affinity} = \frac{1}{\log_{10}(10 + \text{block\_dist})}$$

The reference distance is the number of unique blocks referenced before accessing the same block. The block distance is the difference between block locations on disk. Affinity metrics have three advantages: (1) The range of affinities is between 0% (poorest locality) and 100% (highest), and is intuitive for comparisons. (2) Since the distance measures are denominators with logarithmic transformations, affinities are less sensitive to the maximum distance across generations of hardware. (3) The use of disk blocks, the largest common granularity, allows localities to be compared throughout the storage data path.

For our preliminary numbers, we synthesized workloads with combinations of extreme temporal and spatial localities. Affinity metrics show intuitive behaviors and reveal intrinsic locality characteristics captured by the directory structure. Through replaying two web server and two software development workloads, affinity metrics showed consistent trends and meaningful comparisons across workloads and environments. Notably, we observed how temporal and spatial locality trends are reversed across the cache when under the read-mostly workloads. Moreover, affinity values reflect performance trends, and can reveal the available opportunities for improvement. Further, we used affinity metrics to illustrate how generated workloads with the same aggregate statistics of those of the original web traces can fail to preserve the locality characteristics. For one, the after-cache temporal locality of a synthetic web trace declines asymptotically (as most compulsory misses occur at the beginning of the trace), while that of the original trace shows a steady level of temporal locality throughout.

This research is still a work-in-progress. We are studying how each storage component transforms locality characteristics. We are also using affinities to study the fidelity of various benchmarks. Finally, we plan to use affinities as a guide to design better storage data path components and benchmarks.

## References

- [1] Cherkasova L, Ciardo G, Characterizing Temporal Locality and its Impact on Web Server Performance, *Proc. of ICCCN*, 2000.
- [2] Vanichpun S, Makowski AM, The Output of a Cache under the Independent Reference Model – Where Did the Locality of Reference Go?, *Proc. of the 2004 ACM SIGMETRICS*, 2004.
- [3] Williams S, Abrams M, Standbridge CR, Abdulla G, Fox EA, Removal Policies in Network Caches for World Wide Web documents, *Proc. of the ACM SIGCOMM*, 1996.