

## Assignment 3

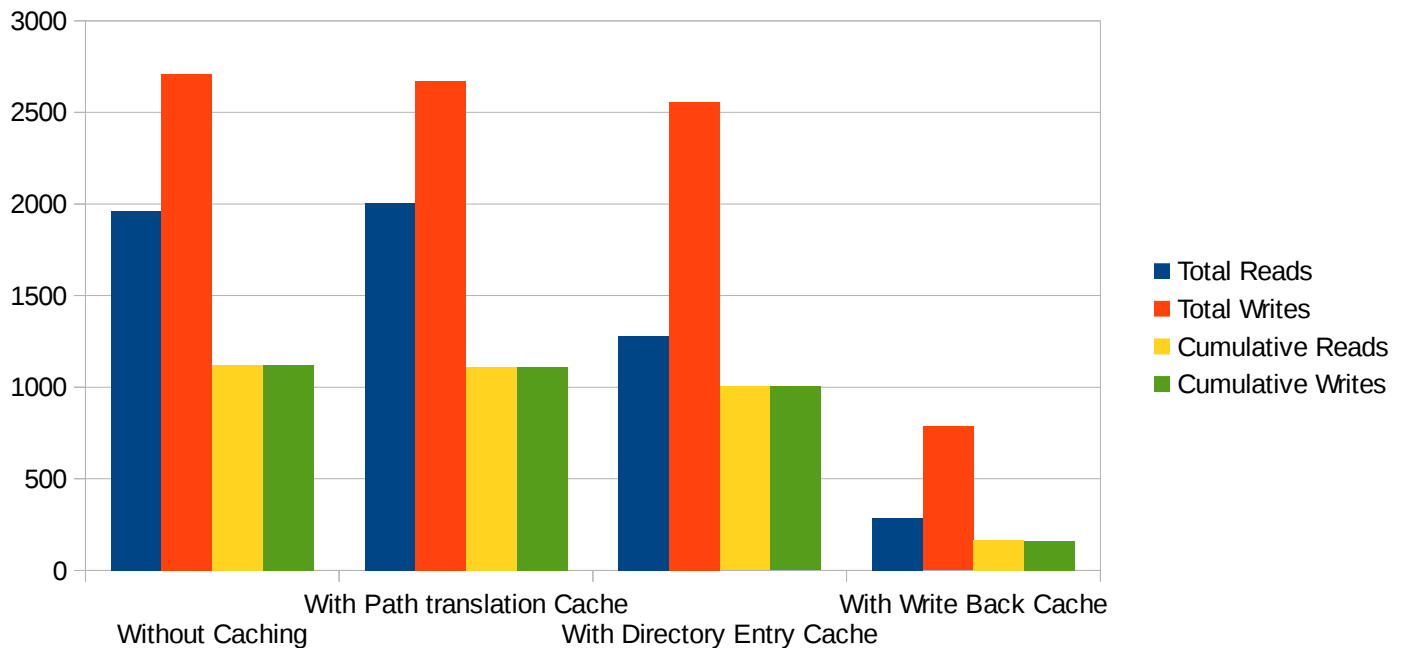
### Experiment:

Experiment on the developed file system was conducted and comparisons were drawn in the following categories:

1. No caching
2. With path translation caching
3. With directory entry caching and 2
4. With write back cache and 2, 3

### Observation:

	Total Reads	Total Writes	Cumulative Reads	Cumulative Writes
Without Caching	1963	2712	1120	1119
With Path translation Cache	2004	2674	1109	1109
With Directory Entry Cache	1278	2554	1004	1004
With Write Back Cache	286	786	163	162



### **Explanation:**

Caching improves the total number of reads and writes that are made from the file system to disk. In this exercise we were asked to implement three different types of caching mechanisms, path translation caching, directory entry caching and write back caching. We observe that the number of cumulative reads and writes obtained by including only the path translation caching did not decrease by a large amount when compared to the ones obtained from without using any cache.

The reason behind this observation is the fact that there were not a lot of files used in testing which had a size greater than 6 blocks or greater than 6144 bytes. Such files (which average sizes greater than 6144 bytes) need extra read and write operations for accessing their consecutive blocks from the different levels of indirection in the inodes.

With path translation caching the overhead of reading the indirect blocks reduces since a mapping of the blocks and the corresponding block numbers is stored in memory. Since, in our test-cases the number of files having sizes greater than 6 blocks in most cases we do not see a significant increase in the count of operations on the system.

With the introduction of the directory entry cache, we significantly increase the performance in terms of reduced number of disk operations. The directory entry cache sheds away an extra overhead of looking up a file name in the directory by saving a mapping of the block numbers and the files associated in the cache. So, in the chart we see that the number of reads decrease by orders of magnitude compared to not using the cache.

Finally, when we introduce write back cache (including both clean and dirty block LRU pools), we see the increase in performance for the file system increases by a lot more. The number of total reads and total writes reduce by a large factor, not only when compared to the scenario of not using a cache, but also to scenarios when we have only a path translation cache, or both path translation and directory caching. With write back, the entire page is saved to disk, and the whole operation of updating pages and reading them can be done through cache. The only time writing takes place to disk is when the cache is full (the dirty write back cache) and the pages are evicted from the cache.

Thus, caching improves the performance of the file system.