Fgure 1.

The first optimization that was made was to use read-write locks rather than global mutices. The effect of this changed was analysed by comparing the average time per request on the server side when using the two types of locks. It was observed that the read-write locks generally had a lower average time per request than the global lock, as can be seen in figure 1. This is because the GET operation now uses its own lock, and need not contend with

POST and DELETE operations, so in general threads spend less time waiting for locks. Another interesting observation is that the average time per request increases when the number of threads increase, even though it was expected that a higher number of threads would increase concurrency and reduce these times. One reason for this may be the overhead of thread management. With increasing numbers of threads there will be more and more context switching between threads. Threads would find themselves waiting more frequently on the scheduler’s queue, so the time between a request’s receipt and its completion would increase accordingly. Additionally, with increasing numbers of threads and parallelism, there would be more frequent modifications to the key-value store, resulting in an increased cache coherency overhead.

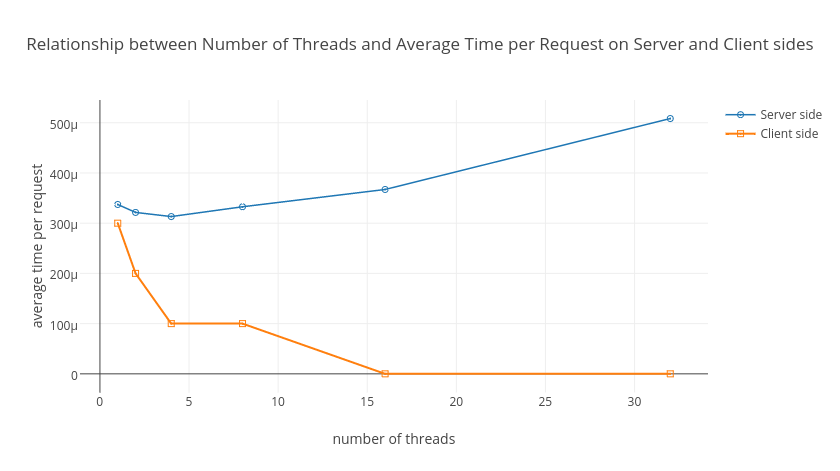
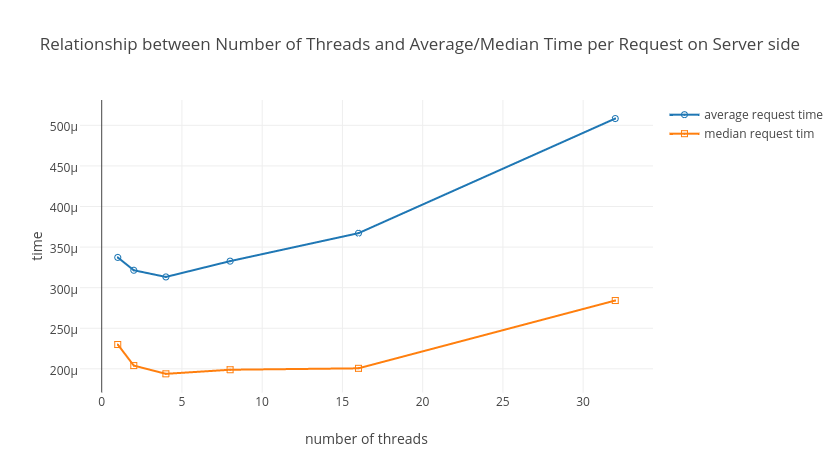
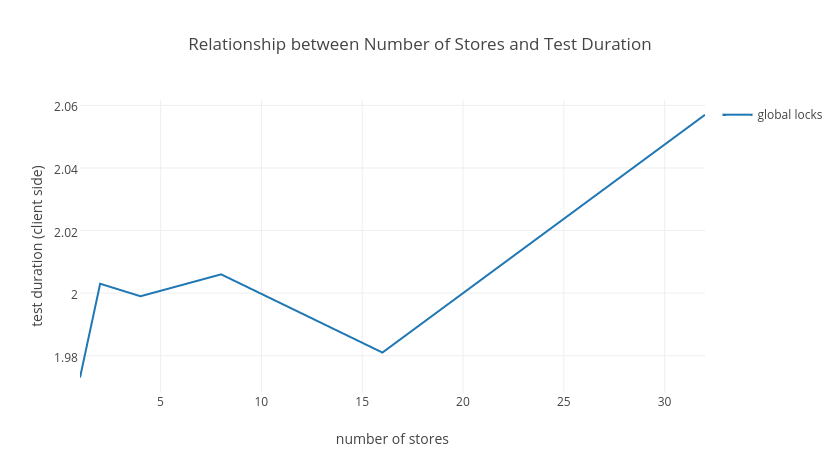
Figure 2.

Figure 2. above shows the relationship between the number of threads and the average time per request as measured on the server and client sides. It was expected that the client side would report higher average times per request due to latency in receiving responses over a network, however it was the opposite in reality. The reason for this as follows: is that the client side only measures the time between its first request being sent and its last response being received. The server side however measures the time taken for each request to complete. Since threads are executing in parallel, time will be “double-counted” when summing to compute the average. This can be seen in Table 1. The *total request time* on the server side is always greater than the *test duration* reported by the client side. The average time per request on the client side is not in reality 0 for 16+ threads, this is a result of truncation of decimal places.

 Figure 3.

The next observation was that the average request time on the server side was always larger than the medium request time, but they followed the same trend as the number of threads increased. As mentioned before, the reason for this is the “double-counting” that happens during the calculation of the average. The median on the other hand is the amount of time that a certain request took to complete, the request being the one that took longer than half of all requests.

Figure 4.

An attempted optimization was to divide the key space amongst number of stores. It was expected that there would be less contention between threads so that the test duration measured by the client side would decrease. However, there was no defined trend of decrease, as shown in Figure 4. There was rather a significant increase in test duration between 16 and 32 stores. The reason for this may be that the increased number of stores meant they were more scattered around memory, so that there were more cache misses.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **Number of Threads** | | | | | |
|  |  |  | **1** | **2** | **4** | **8** | **16** | **32** |
| **with global lock** | server side | total request time | 10.804900 | 10.780800 | 10.364200 | 10.439500 | 12.698200 | 20.994600 |
| Minimum request time | 0.000075 | 0.000088 | 0.000055 | 0.000050 | 0.000067 | 0.000069 |
| Maximum request time | 0.002563 | 0.002547 | 0.012324 | 0.003280 | 0.005704 | 0.009133 |
| Median request time | 0.000227 | 0.000222 | 0.000191 | 0.000188 | 0.000199 | 0.000348 |
| Average time per request | 0.000330 | 0.000329 | 0.000316 | 0.000319 | 0.000388 | 0.000641 |
| client side | test duration | 10.849000 | 6.569000 | 3.430000 | 1.901000 | 1.289000 | 0.973000 |
| time per request | 0.000300 | 0.000200 | 0.000100 | 0.000100 | 0.000000 | 0.000000 |
| requests per second | 3020.3 | 4988.3 | 9553.8 | 17234.1 | 25424.4 | 33676.3 |
| **with read-write locks** | server side | total request time | 11.050400 | 10.532600 | 10.262700 | 10.900700 | 12.029700 | 16.658200 |
| Minimum request time | 0.000063 | 0.000071 | 0.000057 | 0.000058 | 0.000053 | 0.000072 |
| Maximum request time | 0.002536 | 0.003194 | 0.002931 | 0.003986 | 0.004997 | 0.006909 |
| Median request time | 0.000230 | 0.000204 | 0.000194 | 0.000199 | 0.000201 | 0.000284 |
| Average time per request | 0.000337 | 0.000321 | 0.000313 | 0.000333 | 0.000367 | 0.000508 |
| client side | test duration | 11.092000 | 6.574000 | 3.456000 | 1.977000 | 1.198000 | 0.922000 |
| time per request | 0.000300 | 0.000200 | 0.000100 | 0.000100 | 0.000000 | 0.000000 |
| requests per second | 2954.3 | 4984.5 | 9481.6 | 16574.8 | 27359.9 | 35551.3 |

Table 1: Statistics with varying number of threads

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **number of stores** | | | | | |
|  |  |  | **1** | **2** | **4** | **8** | **16** | **32** |
| **global lock** | Server side | total request time | 13.708800 | 13.542200 | 13.864600 | 13.864800 | 14.374700 | 13.361200 |
| Minimum request time | 0.000098 | 0.000778 | 0.000787 | 0.000089 | 0.000098 | 0.000085 |
| Maximum request time | 0.383058 | 0.419038 | 0.431574 | 0.394284 | 0.417977 | 0.368699 |
| Median request time | 0.000174 | 0.000174 | 0.000176 | 0.000177 | 0.000180 | 0.000175 |
| Average time per request | 0.000406 | 0.000413 | 0.000423 | 0.000423 | 0.000439 | 0.000408 |
| Client side | test duration | 1.973000 | 2.003000 | 1.999000 | 2.006000 | 1.981000 | 2.057000 |
| **read-write locks** | Server side | total request time | 15.076200 | 14.234300 | 16.642300 | 13.962400 | 14.474700 | 15.196900 |
| Minimum request time | 0.000092 | 0.000096 | 0.000118 | 0.000081 | 0.000095 | 0.000084 |
| Maximum request time | 0.485557 | 0.481147 | 0.495171 | 0.404236 | 0.381940 | 0.533097 |
| Median request time | 0.000185 | 0.000179 | 0.000192 | 0.000177 | 0.000183 | 0.000180 |
| Average time per request | 0.000460 | 0.000434 | 0.000492 | 0.000426 | 0.000442 | 0.000464 |
| Client side | test duration | 1.860000 | 2.024000 | 2.022000 | 2.061000 | 2.006000 | 2.104000 |

Table 2: Statistics with varying number of stores. All time is in seconds.