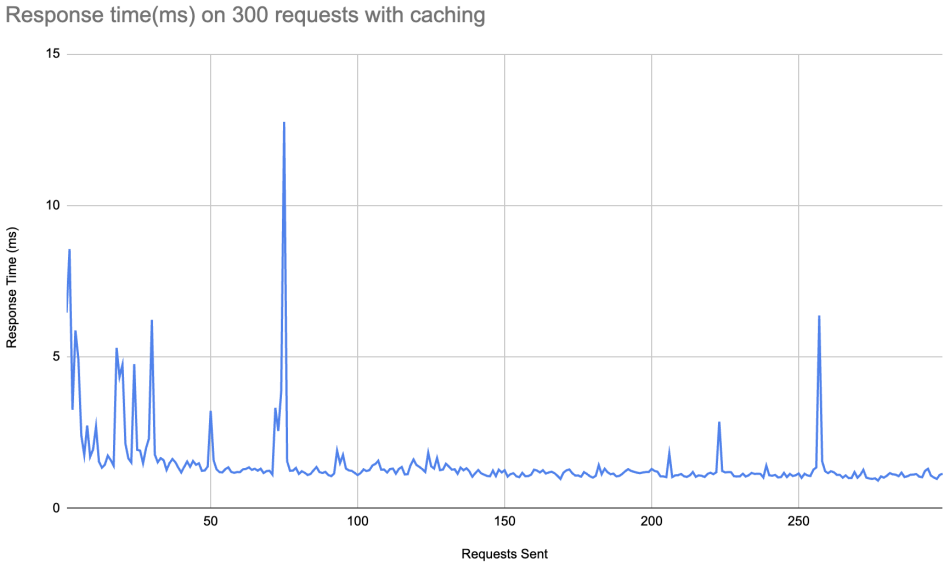
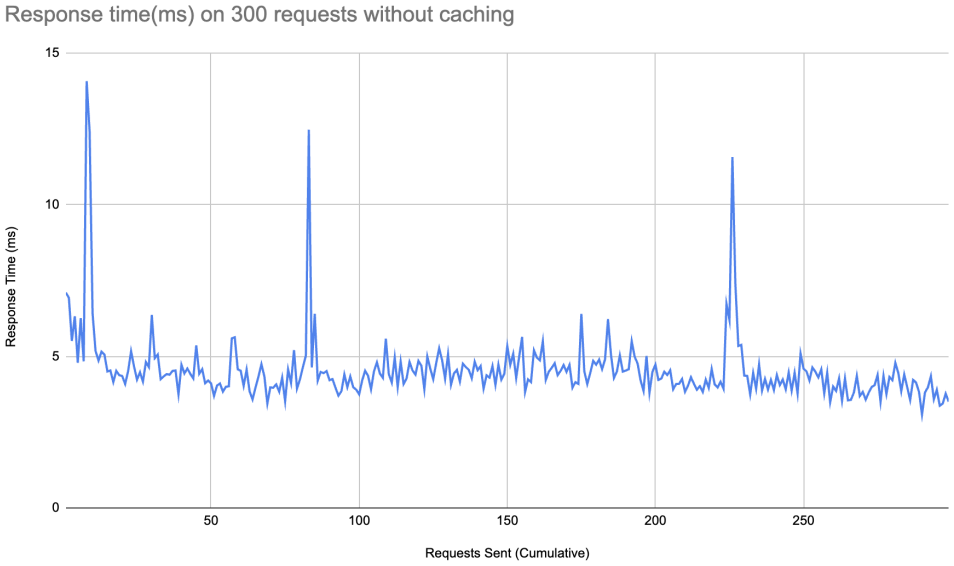
Experimental Evaluation and Performance Metrics

After the replication and caching are added to the server, we need to measure the performance before and after the caching because it has the most effect on performance (replication affects reliability).

So I wrote a script to perform a 300 request on the server as GET requests for the books available in the catalog.
For the 300 requests, the average response time without caching was around 7 milliseconds for each request.
And with caching for 2 milliseconds, which is a clear performance boost, we graphed the response times along the operation, and the results were as follows:





Clearly, we can see that response time with caching goes down after some time because the cache caches most of the items in it.

However, we can notice some spikes in the caching, which is because there was a cache miss, which led to increase in the response time for the frontend node, and we can see that there is a little bit of extra response time when the node has a cache miss, which is because of the overhead of the caching.

And to be specific, I performed a scenario of operations while logging the output of the operations and their response time.

URL	METHOD	Quantity	Response Time(ms)
/book/1	GET	2120	1.4
/book/2	GET	302	2.34
/book/3	GET	302	2.7464
/book/1	GET	2120	2.1474
/purchase/1	PUT	-	13.4742
/book/1	GET	2119	8.314
/book/1	GET	2119	2.712
/purchase/1	PUT	-	13.7
/purchase/1	PUT	-	14.542
/book/1	GET	2117	9.124
/book/1	GET	2117	2.341
/purchase/2	PUT	-	14.507
/book/2	GET	301	7.241
/book/2	GET	301	2.16214

We notice after each purchase request, the cache is invalidated, which results in a cache miss in the following request. We see that the latency for each cache

miss can vary from 7 ms to 9 ms relative to when a cache hit occurs and is bigger than the average response time for the requests without caching by around 1 to 2 ms, which is the overhead for the caching in the frontend node.

Regarding buy queries, the average response time is 15.4775 ms, which won't be affected by caching because it can only be done on GET requests, and caching/replication can make the response time worse for buy queries because we need to hold strong consistency in caching and in our database across our replicas, so we need to notify replicas when a change reaches a replica, and we also need to invalidate the cache in the frontend, which will add some overhead to the operation.