Bazar Lab2

Caching:

In the front-end server, caching was implemented using a Python dictionary for key-value pairs and a list for Least Recently Used (LRU) queue functionality. **Book lookups** were cached using the book ID as the key and the corresponding JSON object as the value. An invalidation endpoint was introduced to remove specific entries from the cache, triggered by catalog server updates. **Search results** caching was more complex, using the search string as the key and storing both the JSON response and a set of topics. Invalidation requests for specific topics or a complete cache clear were handled accordingly.

Replication:

The first modification made to the catalog server was the private addresses in the servers, as it now needs to know the addresses of other catalog servers. Consistency issues in read and write operations were addressed using sequence numbers. For read operations, servers maintained an in-memory list of consistent books. If a book was not found locally, a request hopped between servers until a consistent version was found. Write operations involved checking sequence numbers with other servers, resolving conflicts, and updating the local copy. Replication updates were then sent to other servers.

Order Servers:

Order servers communicated with catalog servers and handled write conflicts by fetching the book again and retrying with reduced quantity. If a catalog server was unreachable, a Gateway Timeout response was returned.

Front-end Server:

The front-end server handled replication by using environment variables to support multiple catalog and order servers. It attempted connections in a round-robin fashion and dealt with timeouts by trying alternative servers.

Dockerization:

Dockerfiles were provided for each server to create Docker images. A Windows batch file facilitated the setup of a user-defined Docker network, image creation, and container running.

Metrics:

Bazar Lab1 vs Bazar Lab2:

Performance metrics were compared for Bazar Lab1 and BazarLab2. Bazar Lab2 demonstrated significantly improved read speeds but showed a slight decrease in write speeds due to additional consistency operations.

Look-ups with Valid Ids:

In this test, 200 look-up requests were directed to each front-end server. All IDs provided are guaranteed to exist in the catalog server(s). Bazar Lab2 initiated with an empty cache. While the 200 IDs are randomly generated, they remain consistent across both versions.

The average response times for these look-up requests are outlined below:

Bazar Lab1	Bazar Lab2
10.71 ms	4.53 ms

The cache demonstrated a notable improvement of 58% in handling numerous valid read operations.

Look-ups, some invalid Ids:

The cache demonstrated a reduced improvement of 35% in handling read operations with some invalid IDs, where non-existent items were consistently requested from catalog servers. Despite the decrease in improvement, having a cache still proved more efficient compared to not having one.

Bazar Lab1	Bazar Lab2
12.05 ms	75.12 ms

Look-ups, some invalidations(write):

In terms of write operations, Bazar Lab2 experienced a performance hit of approximately 14.50% compared to Bazar Lab1. This impact is attributed to the additional time incurred by messages exchanged between catalog servers for replication consistency and between catalog servers and the front-end server for cache consistency.

Bazar Lab1	Bazar Lab2
38.84 ms	44.43 ms

conclusion:

Bazar Lab2 demonstrates superior performance in terms of read speeds compared to Bazar Lab1. However, it lags behind in write speeds, primarily due to the additional time associated with messages exchanged for replication and cache consistency operations between catalog servers and the front-end server. The trade-off between improved read speeds and slightly reduced write speeds highlights the system's design considerations and the impact of introducing caching and replication mechanisms.

Outage:

Bazar Lab2's performance during outages was tested with defined timeout values for communication between servers. The system showed expected performance hits when specific servers were down, highlighting the importance of fine-tuning timeout values.

Server	**Connection Establishment Timeout**	**Full Timeout**
Front End	200 ms	2000 ms
Order	150 ms	1500 ms
Catalog	100 ms	1000 ms

Tests were performed using 50 identical random read and buy requests. Results were as follows:

	No outage	**Order outage**	**Catalog outage**
Read	5.75 ms	No effect	112.08 ms
Write	45.88 ms	450.10 ms	288.42 ms

Conclusion:

Bazar Lab2 outperformed Bazar Lab1 in read speeds but experienced a slight decrease in write speeds. The system demonstrated resilience to outages, with performance hits that could be optimized through further tuning.

The work of the two students: Amr Salman &Ahmed Amira