Distributed Operating Systems

Turning the Bazar into an Amazon: Replication, Caching and Consistency

By : sama Abusair ,Marah Hanini

1.Introduction

The design and architecture of a book purchasing system with three primary services—**Catalog**, **Order**, and **Frontend**—are described in this document. Users can read book information, search for books, and make purchases using the system. Docker is used to build the services, and HTTP API calls are used for inter-service communication.

2.System Overview

The system is made up of three services:

1. **Catalog Service**: Manages the book catalog, provides book details, and decrements the stock when a purchase is made.

2. **Order Service**: Handles the purchasing process, including checking stock availability, processing purchases, and storing order records.

3. **Frontend Service**: Acts as the user interface, allowing users to search for books, view information, and make purchases.

These services are containerized using Docker and communicate with each other through REST APIs. Each service is built and deployed as a separate Docker container.

3. System Components

3.1 Catalog Service

- **Purpose**: Manages the catalog of books, including book information (title, description, quantity, etc.).

- **Endpoints**:

  - `GET /info/:id`: Returns information about a book by its ID.

  - `GET /search/:topic`: Returns a list of books based on the topic search.

  - `POST /purchase/:id`: Decrements the stock of a book after a successful purchase.

- **Data Storage**: Book data is stored in a JSON file (`catalog.json`).

- **Docker**: The service is containerized using Docker and listens on port `5000`.

3.2 Order Service

- **Purpose**: Manages the order process, including checking book availability and storing purchase records.

- **Endpoints**:

  - `POST /purchase/:id`: Handles the purchase of a book. It checks the book's stock, decrements the stock, and stores the order details in a file (`orders.json`).

- **Data Storage**: Purchase records are stored in a JSON file (`orders.json`).

- **Docker**: The service is containerized using Docker and listens on port `5001`.

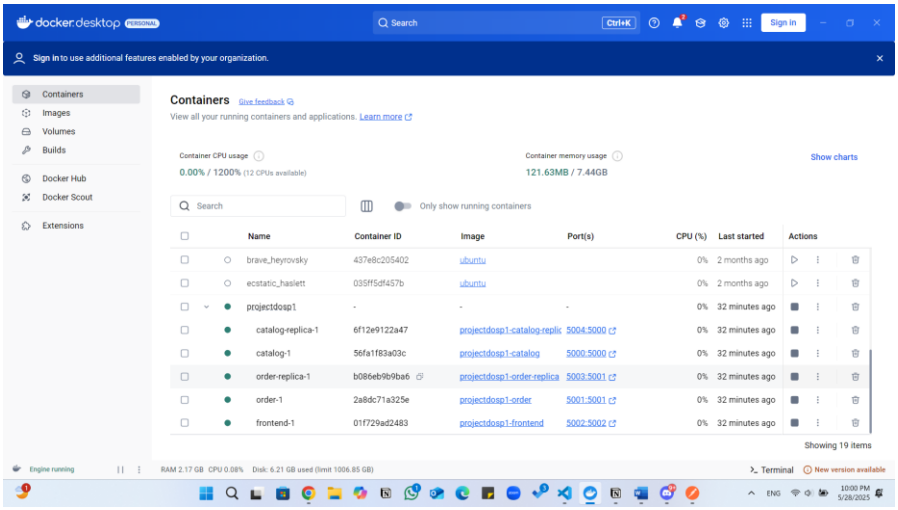
3.3 Frontend Service

The **Frontend Service** acts as the main entry point for users to interact with the system. It handles incoming requests from clients, communicates with both the **Catalog** and **Order** services, manages caching, and provides intelligent routing and logging. This service is responsible for the following key functions:

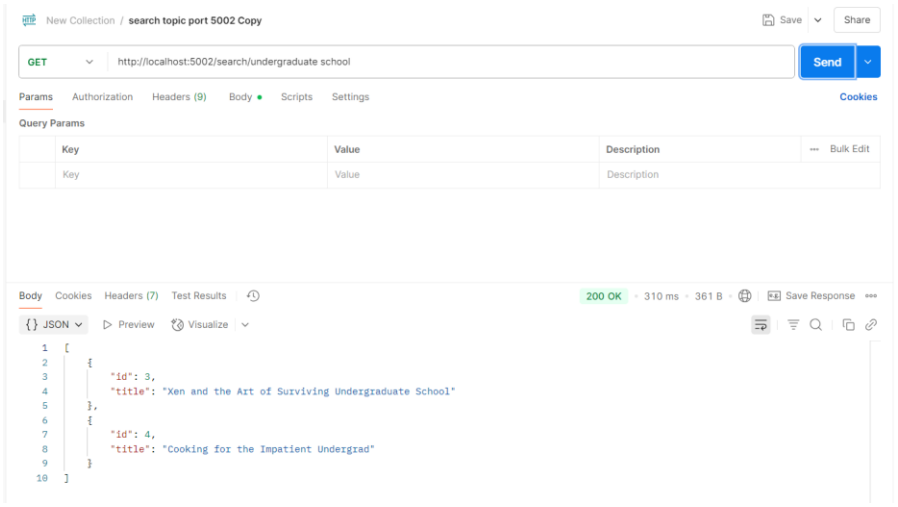
Features and Responsibilities:

1. **Search for Books by Topic**
    - **Endpoint**: GET /search/:topic
    - If the requested topic is found in the cache (searchCache), the data is returned immediately with a low-latency response (CACHE HIT).
    - If the topic is not cached, the request is forwarded to one of the Catalog Service instances using a **round-robin strategy**. The results are then stored in the cache for future use (CACHE MISS).

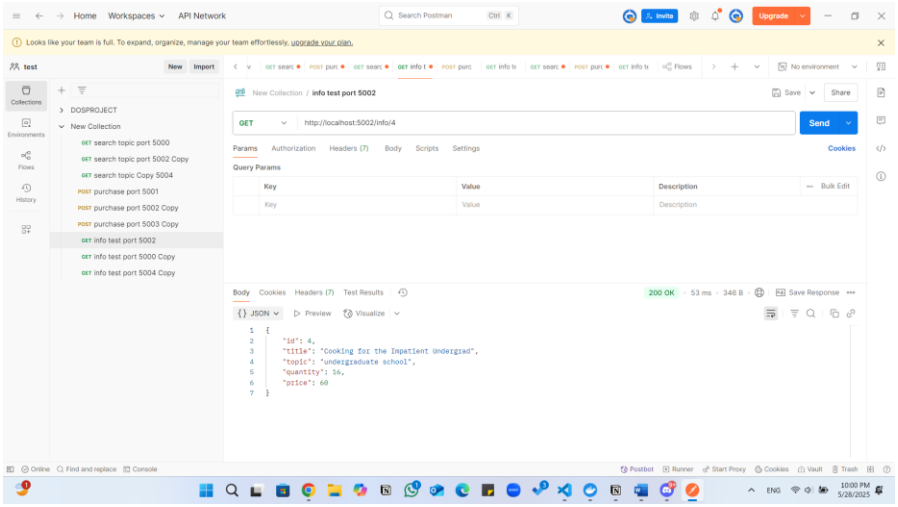
2. **Get Book Information by ID**
    - **Endpoint**: GET /info/:id
    - Checks the infoCache for book data by ID.
    - If the data is cached, it's returned directly (CACHE HIT); otherwise, the Frontend queries one of the Catalog Services and stores the result in the cache (CACHE MISS).
3. **Purchase a Book**
    - **Endpoint**: POST /purchase/:id
    - Sends a purchase request to one of the Order Service instances (also chosen via round-robin).
    - Upon a successful purchase, the related book entry is **invalidated** from the infoCache to ensure consistency.
    - Logs the latency of the purchase process and displays which Order instance handled the request.
4. **Manual Cache Invalidation**
    - **Endpoint**: POST /invalidate/:id
    - Allows manual deletion of a book's cache entry by its ID.
    - Logs whether the cache existed or not, along with the invalidation overhead in milliseconds.
5. **Load Balancing Between Services**
    - Uses **round-robin rotation** to alternate between:
        - Catalog servers: http://catalog:5000 and http://catalog-replica:5000
        - Order servers: http://order:5001 and http://order-replica:5001
6. **Performance Logging**
    - Each operation logs detailed latency information using high-resolution timers.
    - Logs clearly show whether the request was a **CACHE HIT** or **CACHE MISS**, the selected server, and the operation time.



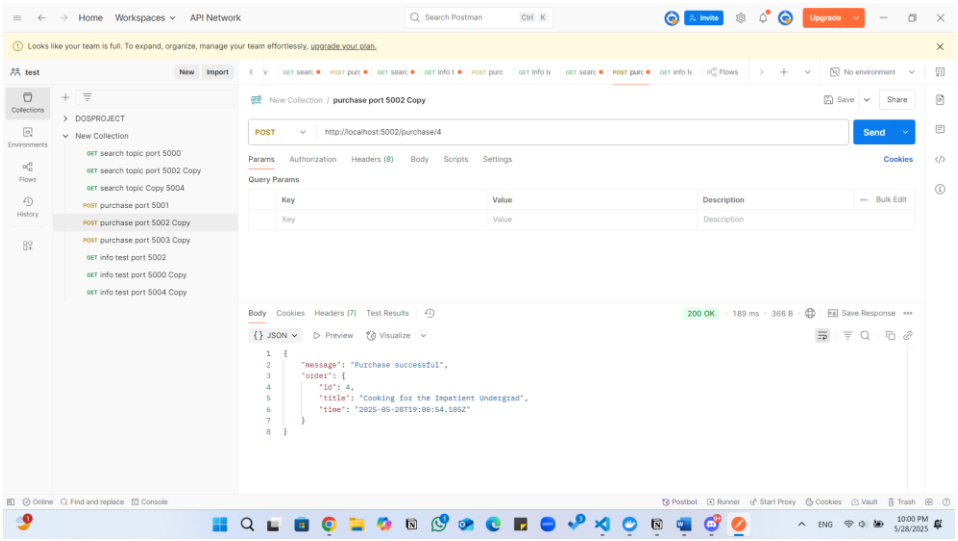
http://localhost:5002/search/undergraduate school



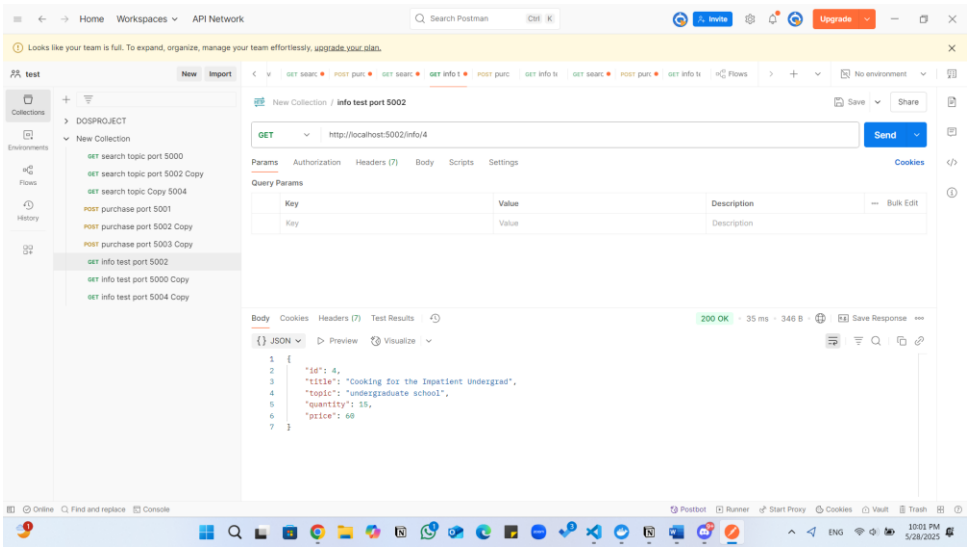
http://localhost:5002/info/4



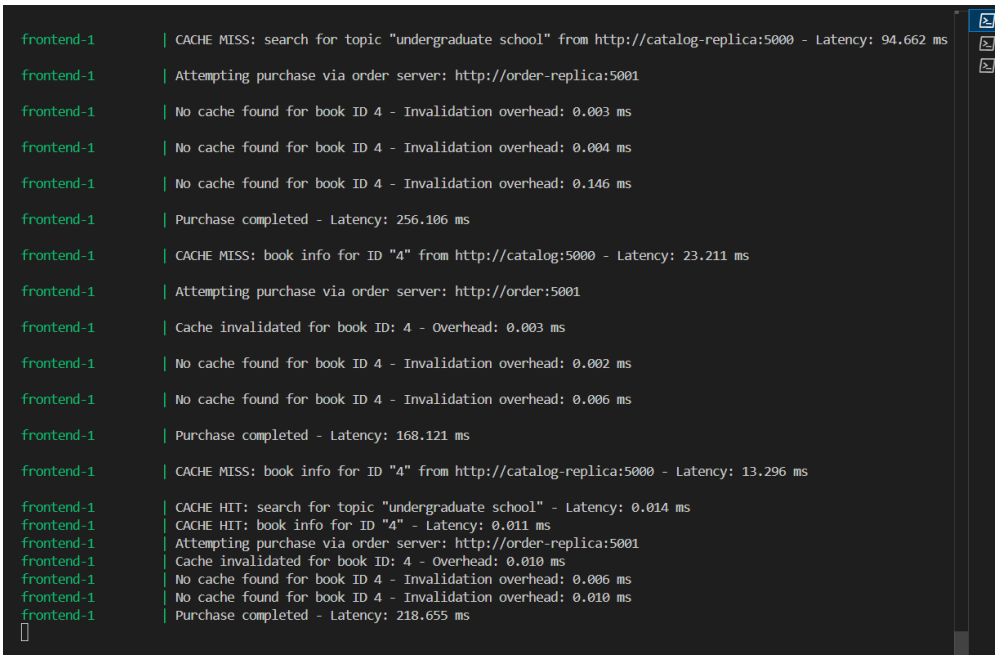
http://localhost:5002/purchase/4



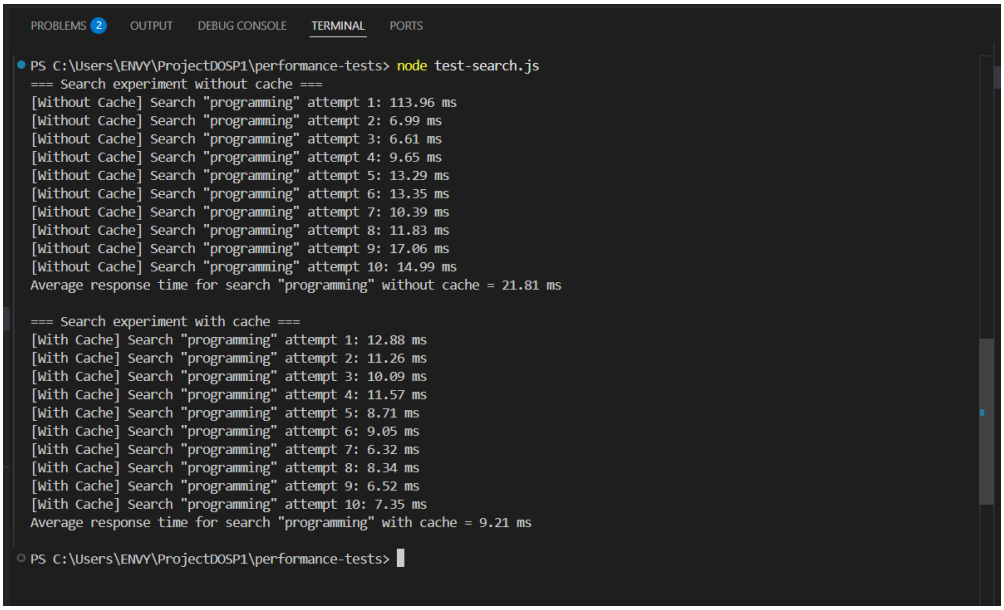
http://localhost:5002/info/4



THE OUTPUT:



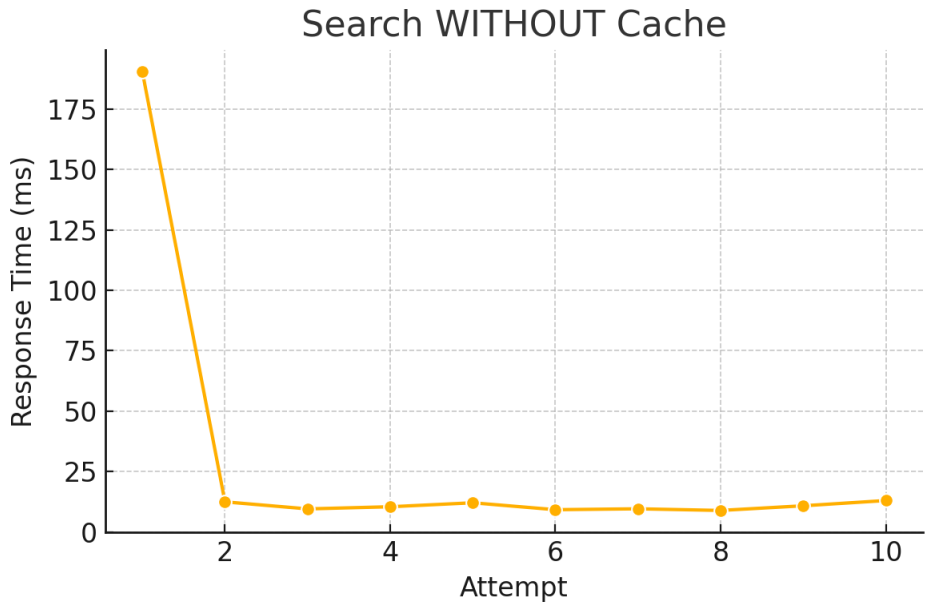
To Test the performance-tests:



Search WITHOUT Cache

Attempt	Response Time (ms)
1	190.560
2	12.430
3	9.510
4	10.400
5	12.040
6	9.180
7	9.550
8	8.870
9	10.800
10	12.970

Average Response Time = 28.631 ms

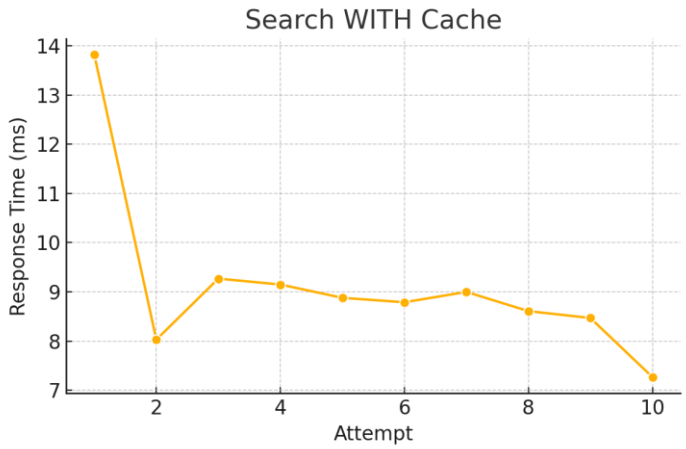


Search WITH Cache

Attempt	Response Time (ms)
1	13.820
2	8.030
3	9.270
4	9.150
5	8.880
6	8.790
7	9.000

8	8.610
9	8.470
10	7.270

Average Response Time = 9.129 ms



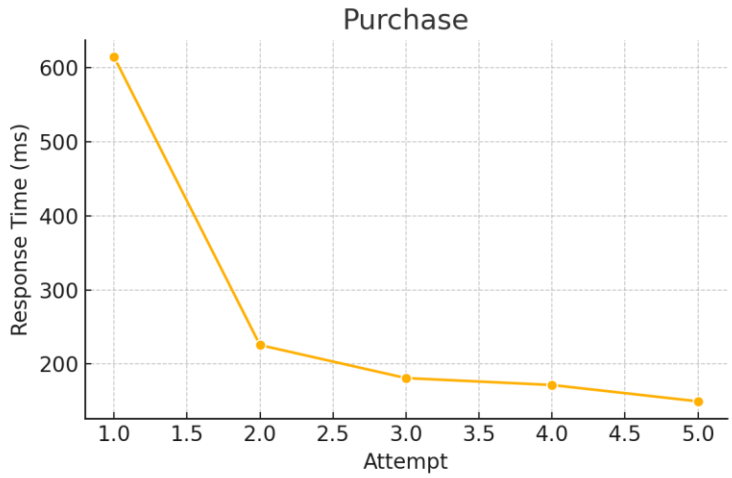

```
Attaching to catalog-1, catalog-replica-1, frontend-1, order-1, order-replica-1
catalog-replica-1  | Catalog Service running on port 5000
catalog-1          | Catalog Service running on port 5000
order-replica-1    | Order Service running on port 5001
order-1            | Order Service running on port 5001
frontend-1         | Frontend Service running on port 5002
frontend-1         | CACHE MISS: search for topic "programming" from http://catalog:5000 - Latency: 38.547 ms
frontend-1         | CACHE HIT: search for topic "programming" - Latency: 0.006 ms
frontend-1         | CACHE HIT: search for topic "programming" - Latency: 0.008 ms
frontend-1         | CACHE HIT: search for topic "programming" - Latency: 0.013 ms
frontend-1         | CACHE HIT: search for topic "programming" - Latency: 0.020 ms
frontend-1         | CACHE HIT: search for topic "programming" - Latency: 0.017 ms
frontend-1         | CACHE HIT: search for topic "programming" - Latency: 0.018 ms
frontend-1         | CACHE HIT: search for topic "programming" - Latency: 0.022 ms
frontend-1         | CACHE HIT: search for topic "programming" - Latency: 0.022 ms
frontend-1         | CACHE HIT: search for topic "programming" - Latency: 0.061 ms
frontend-1         | CACHE HIT: search for topic "programming" - Latency: 0.012 ms
frontend-1         | CACHE HIT: search for topic "programming" - Latency: 0.009 ms
frontend-1         | CACHE HIT: search for topic "programming" - Latency: 0.009 ms
frontend-1         | CACHE HIT: search for topic "programming" - Latency: 0.010 ms
frontend-1         | CACHE HIT: search for topic "programming" - Latency: 0.008 ms
frontend-1         | CACHE HIT: search for topic "programming" - Latency: 0.007 ms
frontend-1         | CACHE HIT: search for topic "programming" - Latency: 0.010 ms
frontend-1         | CACHE HIT: search for topic "programming" - Latency: 0.011 ms
frontend-1         | CACHE HIT: search for topic "programming" - Latency: 0.007 ms
frontend-1         | CACHE HIT: search for topic "programming" - Latency: 0.013 ms
```

Purchase

Attempt	Response Time (ms)
1	614.690
2	225.600
3	181.030

4	171.670
5	149.620

Average Response Time = 268.522 ms



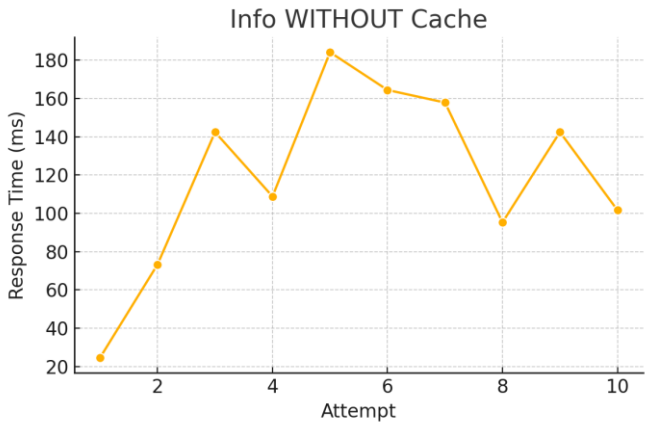

```
PS C:\Users\ENVY\ProjectDOSP1\performance-tests> node test-purchase.js
=== Purchase experiment ===
Purchase book ID=1 attempt 1: 217.14 ms
Purchase book ID=1 attempt 2: 183.27 ms
Purchase book ID=1 attempt 3: 148.45 ms
Purchase book ID=1 attempt 4: 192.01 ms
Purchase book ID=1 attempt 5: 186.02 ms
Average response time for purchase book ID=1 = 185.38 ms

PS C:\Users\ENVY\ProjectDOSP1\performance-tests>
```

```
frontend-1  | Attempting purchase via order server: http://order:5001
frontend-1  | No cache found for book ID 1 - Invalidation overhead: 0.006 ms
frontend-1  | No cache found for book ID 1 - Invalidation overhead: 0.005 ms
frontend-1  | No cache found for book ID 1 - Invalidation overhead: 0.005 ms
frontend-1  | Purchase completed - Latency: 155.100 ms
frontend-1  | Attempting purchase via order server: http://order-replica:5001
frontend-1  | No cache found for book ID 1 - Invalidation overhead: 0.006 ms
frontend-1  | No cache found for book ID 1 - Invalidation overhead: 0.007 ms
frontend-1  | No cache found for book ID 1 - Invalidation overhead: 0.006 ms
frontend-1  | Purchase completed - Latency: 175.293 ms
frontend-1  | Attempting purchase via order server: http://order:5001
frontend-1  | No cache found for book ID 1 - Invalidation overhead: 0.005 ms
frontend-1  | No cache found for book ID 1 - Invalidation overhead: 0.006 ms
frontend-1  | No cache found for book ID 1 - Invalidation overhead: 0.005 ms
frontend-1  | Purchase completed - Latency: 143.150 ms
frontend-1  | Attempting purchase via order server: http://order-replica:5001
frontend-1  | No cache found for book ID 1 - Invalidation overhead: 0.004 ms
frontend-1  | No cache found for book ID 1 - Invalidation overhead: 0.005 ms
frontend-1  | No cache found for book ID 1 - Invalidation overhead: 0.007 ms
frontend-1  | Purchase completed - Latency: 184.062 ms
frontend-1  | Attempting purchase via order server: http://order:5001
frontend-1  | No cache found for book ID 1 - Invalidation overhead: 0.006 ms
frontend-1  | No cache found for book ID 1 - Invalidation overhead: 0.005 ms
frontend-1  | No cache found for book ID 1 - Invalidation overhead: 0.012 ms
frontend-1  | Purchase completed - Latency: 179.645 ms
```

Info WITHOUT Cache

Attempt	Response Time (ms)
1	24.606
2	73.279
3	142.393
4	108.779
5	184.108
6	164.383
7	157.819
8	95.244
9	142.418
10	101.622

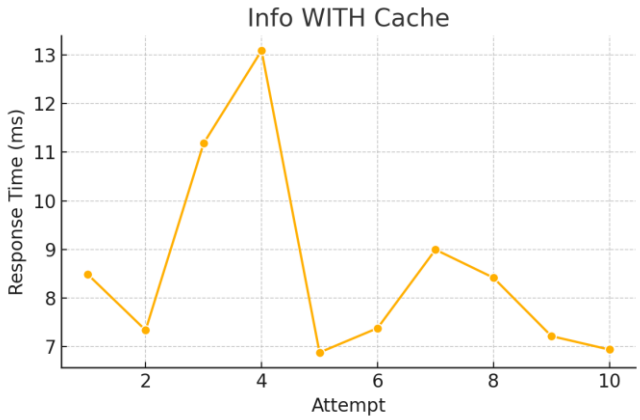
Average Response Time = 119.465 ms



Info WITH Cache

Attempt	**Response Time (ms)**
1	**8.488**
2	**7.338**
3	**11.181**
4	**13.091**
5	**6.879**
6	**7.380**
7	**8.995**
8	**8.418**
9	**7.216**
10	**6.939**

Average Response Time = 8.592 ms




```
PS C:\Users\ENVY\ProjectDOSP1\performance-tests> node test-info.js
=== Testing WITHOUT Cache ===
[Without Cache] Run 1: 14.588 ms
[Without Cache] Run 2: 18.061 ms
[Without Cache] Run 3: 15.612 ms
[Without Cache] Run 4: 12.038 ms
[Without Cache] Run 5: 19.990 ms
[Without Cache] Run 6: 16.422 ms
[Without Cache] Run 7: 13.137 ms
[Without Cache] Run 8: 12.841 ms
[Without Cache] Run 9: 11.089 ms
[Without Cache] Run 10: 13.392 ms

>>> Average latency (without cache): 14.717 ms

=== Testing WITH Cache ===
[With Cache] Run 1: 5.555 ms
[With Cache] Run 2: 5.556 ms
[With Cache] Run 3: 3.892 ms
[With Cache] Run 4: 4.822 ms
[With Cache] Run 5: 4.727 ms
[With Cache] Run 6: 3.821 ms
[With Cache] Run 7: 4.271 ms
[With Cache] Run 8: 4.649 ms
[With Cache] Run 9: 4.523 ms
[With Cache] Run 10: 3.976 ms

>>> Average latency (with cache): 4.579 ms
```

```
frontend-1          | No cache found for book ID 1 - Invalidation overhead: 0.005 ms
frontend-1          | CACHE MISS: book info for ID "1" from http://catalog-replica:5000 - Latency: 9.228 ms
frontend-1          | Cache invalidated for book ID: 1 - Overhead: 0.011 ms
frontend-1          | CACHE MISS: book info for ID "1" from http://catalog:5000 - Latency: 10.708 ms
frontend-1          | Cache invalidated for book ID: 1 - Overhead: 0.007 ms
frontend-1          | CACHE MISS: book info for ID "1" from http://catalog-replica:5000 - Latency: 9.576 ms
frontend-1          | Cache invalidated for book ID: 1 - Overhead: 0.008 ms
frontend-1          | CACHE MISS: book info for ID "1" from http://catalog:5000 - Latency: 8.083 ms
frontend-1          | Cache invalidated for book ID: 1 - Overhead: 0.005 ms
frontend-1          | CACHE MISS: book info for ID "1" from http://catalog-replica:5000 - Latency: 12.304 ms
frontend-1          | Cache invalidated for book ID: 1 - Overhead: 0.007 ms
frontend-1          | CACHE MISS: book info for ID "1" from http://catalog:5000 - Latency: 9.872 ms
frontend-1          | Cache invalidated for book ID: 1 - Overhead: 0.007 ms
frontend-1          | CACHE MISS: book info for ID "1" from http://catalog-replica:5000 - Latency: 8.079 ms
frontend-1          | Cache invalidated for book ID: 1 - Overhead: 0.007 ms
frontend-1          | CACHE MISS: book info for ID "1" from http://catalog:5000 - Latency: 8.032 ms
frontend-1          | Cache invalidated for book ID: 1 - Overhead: 0.007 ms
frontend-1          | CACHE MISS: book info for ID "1" from http://catalog-replica:5000 - Latency: 6.892 ms
frontend-1          | Cache invalidated for book ID: 1 - Overhead: 0.005 ms
frontend-1          | CACHE MISS: book info for ID "1" from http://catalog:5000 - Latency: 8.184 ms
frontend-1          | CACHE HIT: book info for ID "1" - Latency: 0.006 ms
frontend-1          | CACHE HIT: book info for ID "1" - Latency: 0.008 ms
frontend-1          | CACHE HIT: book info for ID "1" - Latency: 0.005 ms
frontend-1          | CACHE HIT: book info for ID "1" - Latency: 0.006 ms
frontend-1          | CACHE HIT: book info for ID "1" - Latency: 0.008 ms
frontend-1          | CACHE HIT: book info for ID "1" - Latency: 0.007 ms
frontend-1          | CACHE HIT: book info for ID "1" - Latency: 0.005 ms
frontend-1          | CACHE HIT: book info for ID "1" - Latency: 0.011 ms
frontend-1          | CACHE HIT: book info for ID "1" - Latency: 0.008 ms
frontend-1          | CACHE HIT: book info for ID "1" - Latency: 0.006 ms
```