DATA70141

ASSIGNMENT 2 (GROUP): AMAZONE – INDIVIDUAL REPORT

11356880

Personal Reflection

1. Role in the Amazone Project

My Role as a group member in the Amazone project was to work with the team and design a schema which would be suitable for the website's need to operate efficiently. As an individual, I was responsible for designing and constructing the "products" collection where we store all of the products' information that Amazone sells, and one additional query.

The design of this collection was not as rigid as most of the other collections. Most of the products, depending on what category they fall into and what type of product it is, was the driving factor of what fields are going to be defined for it. As the brief of the assignment describes the attributes that we have to specify for the products according to their type and category, that was followed. About the average rating for the products, it was embedded inside the "products" collection. First reason was for faster reading speeds as embedding is faster for reading. Secondly, only other place it could be embedded was customers collection which is not the ideal choice as that information is not important to customer, but more important for the database for the recommendation process. Going on to the recommended products from this, these were embedded within the 'customers' collection to make them appear instantly as the customers log in.

About the query I designed which displayed products' performance in current orders and their inventory levels for that day, I designed that query in a function so that it can be called easily and repeatedly. The purpose was to provide a way for the caller of the query to get some summary insights about what products they should be looking at to stock more or less of, depending on their performance.

2. Learning Experience and Challenges

One thing I particularly learned and started getting better at while doing the Amazone project is PyMongo. Not being from a coding background, I started learning python as my first coding language. To be able to do the backend database-related work in a language which I am learning, and also learn new things like aggregation pipelines in it is a great learning experience. Error handling is also a part I learnt through this project which is very crucial when making aggregation pipelines, so that we know exactly what's not working if something goes wrong.

JSON functions is something I found challenging. As I am not particularly familiar with JavaScript, it was hard to switch from Python to JavaScript, not knowing the syntax or the variables at hand. Even then, I challenged myself to make the average rating query into a function, where I got great support from my group also.

Company Expansion to EU countries

This section focuses on company expanding its operations to some EU countries and set up an additional data centre in Europe for this purpose.

A) Replication Algorithm

As Amazone is an ecommerce website, ensuring consistency and integrity across data centres, reducing the risk of downtime are some of the crucial things we have to consider for an overall great experience on the website, whilst retaining a smooth operation of the database.
I believe going with Multi-Leader Replication, using a Synchronous Approach in this scenario is optimal for the following reasons:

1. Consistency across Geographical Distributions
   As the company expands to more EU countries, synchronous replication makes sure users across different regions go through a consistent and up to par experience. Using multiple leader nodes when there are multiple data centres reduces latency in sending requests to the leader as compared to having a single leader.

2. Strong Consistency for Critical Transactions
   Synchronous approach ensures write operations are acknowledged after being executed on all leader nodes, which is important for an ecommerce business where write operations are crucial such as ordering products.

3. Reliable Order Fulfilment
   This strategy is particularly great for order processing, where reliable and consistent data is very important for accurate order achievement. It ensures that each leader has access to a synchronized information list, minimizing risk of error.

4. Data Integrity
   As every write operation requires acknowledgment from all leader nodes, the system as a whole accomplishes high level of trust. This is important for an ecommerce platform as the company has to ensure accurate product information such as pricing and availability, are relayed accurately.

While having strong points of trust, consistency and data integrity, the replication strategy is always a balancing act, based on ACID principle and CAP theorem, there are some trade-offs with this approach, which are as follows:

1. Latency Impact
   The synchronous approach does come with its drawback of higher latency. Delayed Responsiveness during sale periods like Black Friday and Christmas can introduce challenges as write operations must wait for acknowledgments from all leader nodes before being considered complete. I believe this is a necessary trade-off for strong consistency.

2. Scalability Challenges
   With a synchronous approach, high scalability can be challenging during high write periods. Due to the coordination required through all leader nodes, synchronous writes could become a bottleneck.

B) Partition and Allocation Strategy

To align with my Replication Strategy, I have chosen a Geo-Partitioning Strategy and Allocation, i.e., copies of partitions on multiple nodes for fault tolerance. This will benefit Amazone for the following reasons:

1. Localized Access and Reduced Latency
   With a Geo-Partitioning Strategy, data is stored locally within each partition. This approach significantly reduces latency for read operations by minimizing the physical distance between users and the data they request, i.e., proximity to data. This could potentially reduce latency for writes depending on how the distribution of writes across partitions is handled.

2. Consistent Data Across Leaders
   This partitioning strategy, when combined with the replication algorithm specified above, guarantees changes are applied on a consistent manner across all regions, which ensures a cohesive view of data.

3. Legal Obligations
   Geo-Partitioning helps the company meet the regulatory compliance of specific regions by keeping data within specific geographical boundaries. This aligns well with the replication strategy while adhering to regional data protection regulations.

4. Scalability and Fault Isolation
   Geo-Partitioning enables independent operation of each partition. This promotes Horizontal Scalability as increased workload can be dealt with adding new leader nodes which can help the company handle growing traffic and data volume.
   In respect to isolation of a fault, if a leader node goes down because of failure or maintenance needs, the impact is localized. This means other nodes continue to work independently, ensuring fault isolation while minimizing disruptions to the overall system.

The trade-offs for this kind of partitioning are the following:

1. Initial Set Complexity and Cost Implications
   The initial set-up for this partition is complex and so can be difficult to achieve on a large scale. This could be remedied by thorough planning and testing.
   The maintenance cost and initial setup cost are something to be taken into consideration for which comprehensive analysis is vital to assess financial implications.

2. Potential Data Hotspots
   With this partitioning, nodes can experience uneven distribution of user activity as a shard may cover more urban areas and experience higher data traffic. This may lead to data hotspots in specific partitions. To keep this in check and mitigate any potential issues, monitoring usage patterns is essential.

C) Re-Design of the database for operations expansion

As Amazone is based off of UK and expanding to EU countries, we have to take into consideration the strict legal guidelines of the European Union. This being our main factor, design of our database has to be adapted/modified for the following reasons:

1. GDPR Compliance
   As the customers collection holds sensitive information like addresses and payment details, handling this data require secure procedures. Secure storage procedures and processing of sensitive information, should be approached transparently. The goal is to align these practices with GDPR requirements.

2. Multilingual Support
   As the company expands to more EU countries where English may not be the prevalent language, adapting for multi-language support could be a huge advantage to penetrate the local areas and expand their customer base.

3. Network Architecture
   As data centres will be communicating on a global scale, optimisation of data transfer protocols should be taken into consideration. Choosing network protocols that are optimized for global communication. Protocols, like HTTP/3, provide improved compression, which results in efficient communication and data transfer, and low latency, which are essential for secure security measures.

4. Scalability
   As Amazone will most likely see an increased traffic with this expansion, assessing the scalability of the database must be taken into consideration. Indexing meaningful fields like “avg_rating” in “products” collection, separating “past orders” from “customers” collection, “reviews” and “recommended products” can be separated to their own collection because of increased complexity in their algorithms are some of the things that can optimize performance as the system scales.

This concludes my individual report on the Amazone Project.