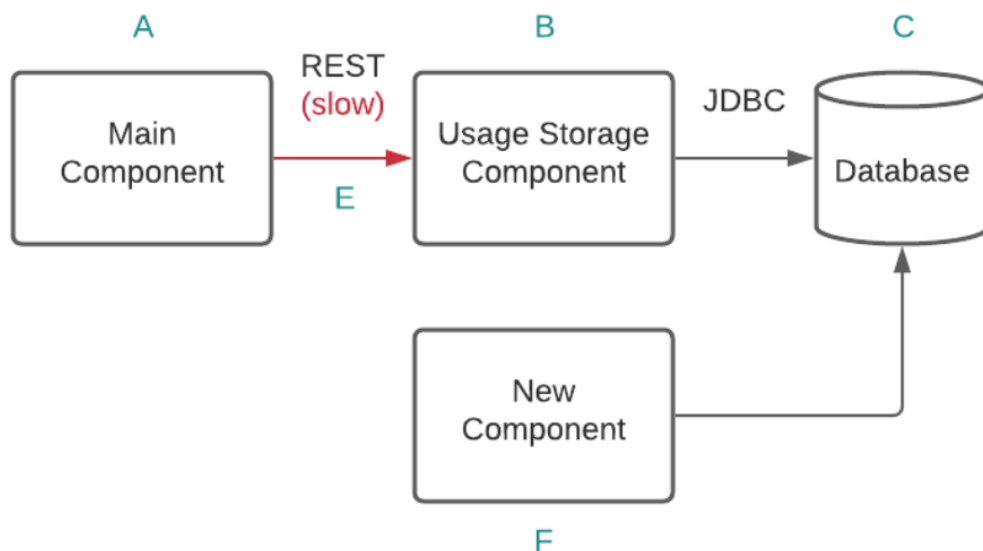


Software Dev Technical Questionnaire and Task

Deployment and Cloud

Existing situation



A system has been successfully deployed for a long time, but in recent weeks there have been some problems. An investigation has indicated that a new component (F) was introduced. The new component (F) runs a batch process each day that is putting pressure on the database (C) for an hour or two. This in turn is probably why REST API calls (E) from the Main Component to the Usage Storage Component are timing out around the same time. The Main Component (A) is not doing anything with the responses from the Usage Storage Component, but just checks to see if it gets a 200 HTTP response. Briefly outline any possible architectural remedies you can imagine being helpful in this situation.

Suggested solution 1

Assumptions

My answers will be based on AWS

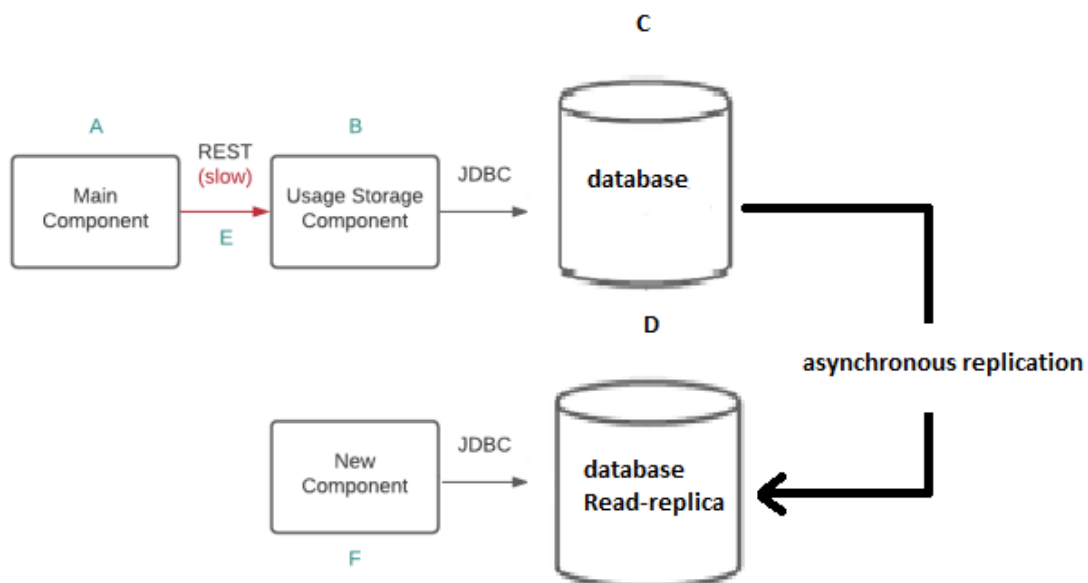
The existing architecture is solely hosting on AWS, not hybrid

The batch process mentioned are mainly focus on READ operation, like generating reports

The database is RDBMS

Description

Create a read-replica which is dedicated for the batch process (component F). Read replicator support for the same region or different region (involve extra cost). The workload pressure from the batch process will shift to the read-replica from the database instance.



Potential drawbacks

Read-replica supports read only. If the batch process involves CREATE/UPDATE, the request would redirect back to the database instance.

Reference

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_ReadRepl.html

Suggested solution 2

Assumptions

My answers will be based on AWS

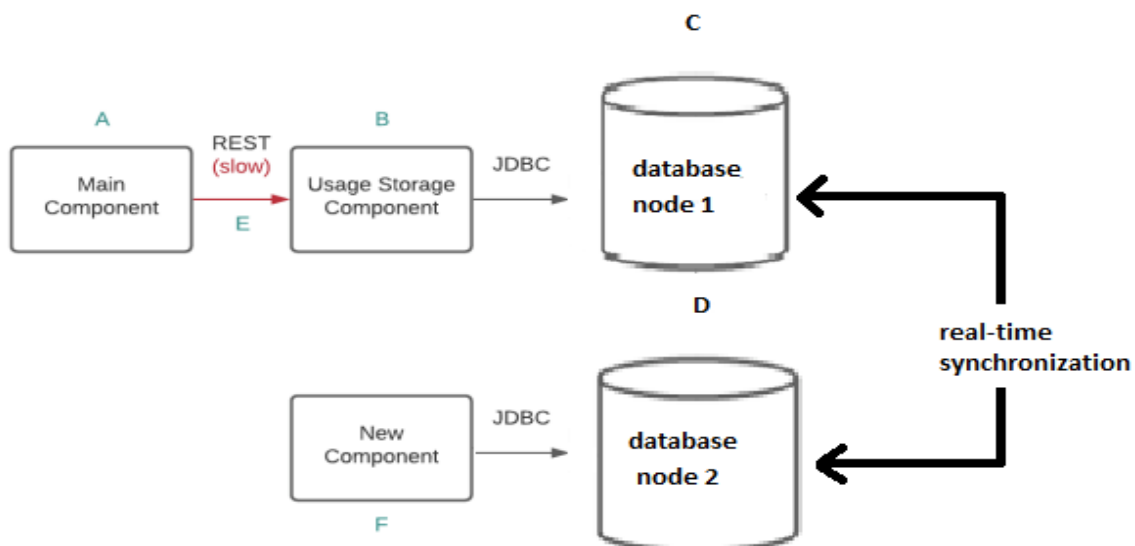
The existing architecture is solely hosting on AWS, not hybrid

The batch process may involve write process, like create or update existing record

The database is RDBMS

Description

Adopt AWS Aurora. AWS Aurora supports multi-master cluster. All database instance inside the same cluster can perform read and write



Potential drawbacks

Currently, Aurora multi-master only supports for single region only. That means all database instances inside in the same cluster must be located within the same region.

Expensive - it is more costly than adopt a read-replica in solution 1

Conflict - You have to handle the concurrency issue inside the application logic. Let say if there are different database instances try to update the same record, it would result in rollback of both transactions.

Reference

<https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/aurora-multi-master.htm>

!

Other suggested solutions

Dynamodb - Dynamodb is a AWS managed no-sql database. It supports for multi-master database instance locate in multi region. As this is a nosql database, it requires significant effort to convert the existing database to nosql database (if the existing database is a RDBMS)

Cloudfront - Cloudfront is the AWS managed CDN service. We can implement the cloudfront between component A and B. That means when ever we call the REST API from component A, we would go through the cloudfront. Cloudfront would cache the response of recent request, so it will save some workload of component B. However, the question already mentioned the bottleneck at the database, so this solution probably is not best fit for this situation.

Load balancer and multi server - We can have multi component B and a load balancer between component A and B. That means when ever we call the REST API from component A, we would go through the load balancer, so the load balancer will help us to distribute the workload equally. However, the question already mentioned the bottleneck at the database, so this solution probably is not best fit for this situation.