WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

CS 839: Topics in Database Management Systems

Lecture 13: Serverless-2

Xiangyao Yu
10/18/2023

Group Discussion from Lecture #12

What are the key advantages and disadvantages of classic- vs. neo-serverless? Is it possible to combine some of these advantages in a new system architecture?

Can these embedded database engines benefit from cloud computing? If so, how?

Group Discussion from Lecture #12

Advantages of classic serverless:

- Lightweight, ease of use, simplicity (no need to manage a separate database server),
- Low latency
- Offline (no network)
- More secure

Advantages of neo serverless:

- High scalability
- Good performance
- High availability and durability
- Independent failure of application and database
- Configuration-free
- Data sharing

Group Discussion from Lecture #12

Combine classic- and neo-serverless

- Disaggregate log and table storage to cloud storage -> high availability and durability
- Cloud-based database backup (e.g., S3)
- Client pushes computation down to the cloud (e.g., filters, expensive joins)
- Use external tables to enable data sharing and collaboration. Need to support writes and transaction.
- Hierarchical processing: local DB (top layer) as cache of external storage (back-end layer)
  - Top layer: handle user interactions and real-time responsiveness
  - Back-end layer: data that does not require real-time processing.
  - A form a multi-cloud database

Serverless-2– Q/A

Why need a persistent coordinator in Starling? Use a function instead?

Better cloud functions to avoid workarounds in this paper?

Workloads where Starling is worse than traditional cloud databases?

How to handle 15-min execution time cap in lambda?

How does FaaS achieve isolation guarantees?

Discussion Questions

In Lecture 6, we discussed functions that could be pushed to a "middle layer" between compute and storage. Can the middle layer be built using lambda functions? What are the advantages and disadvantages of using lambda functions to build the layer?

- Examples of middle-layer functions: Filtering and aggregation, intermediate results store, log replay, Transactions, data shuffle, access control, caching, data format translation, real-time analytics, data virtualization, ETL functions

Can a transactional database benefit from lambda functions? If so, how?