# An Experimental Key-Value Database Using Serverless Cloud Functions

Chung-An Huang | Austin Patello | Beshani Weralupitiya

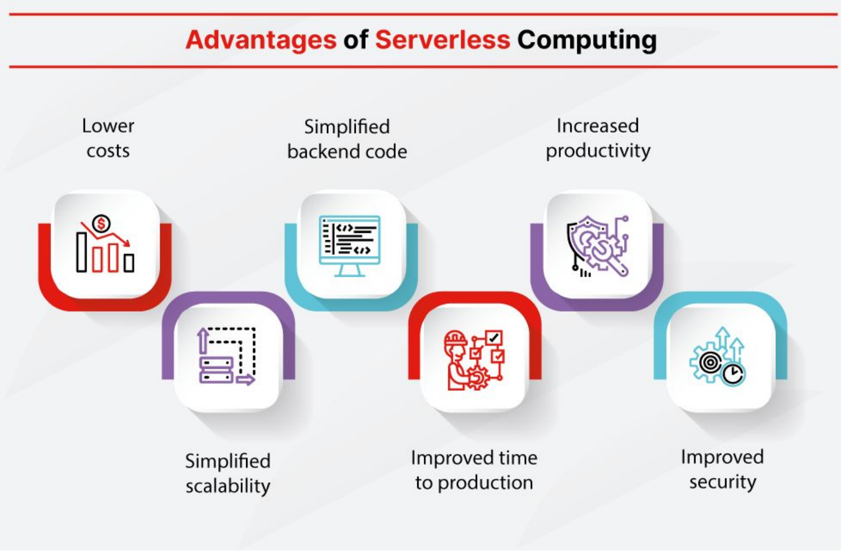
# Outline

- Introduction
- Methodology
- Evaluation
- Conclusion
- References
- Q & A

# Introduction - Background

- Serverless computing has recently attracted a lot of attention from research and industry.

- The capacity of cloud function services like AWS Lambda or Azure Functions to execute precise, small-scale tasks positions them as an intuitive option for query processing.

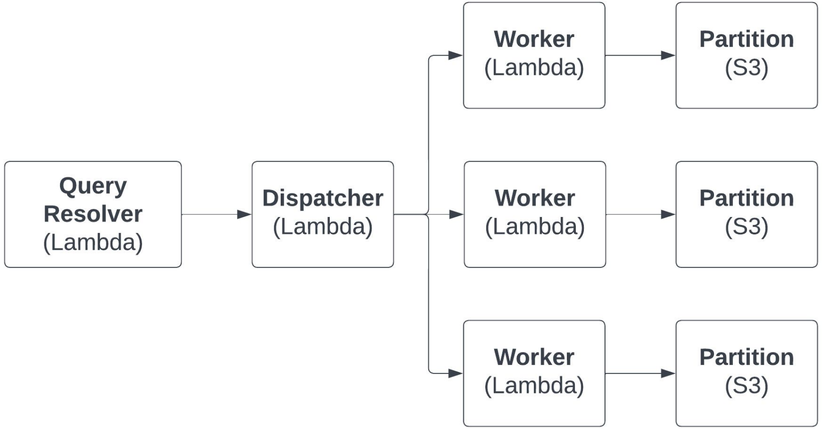


# Introduction - Related Works

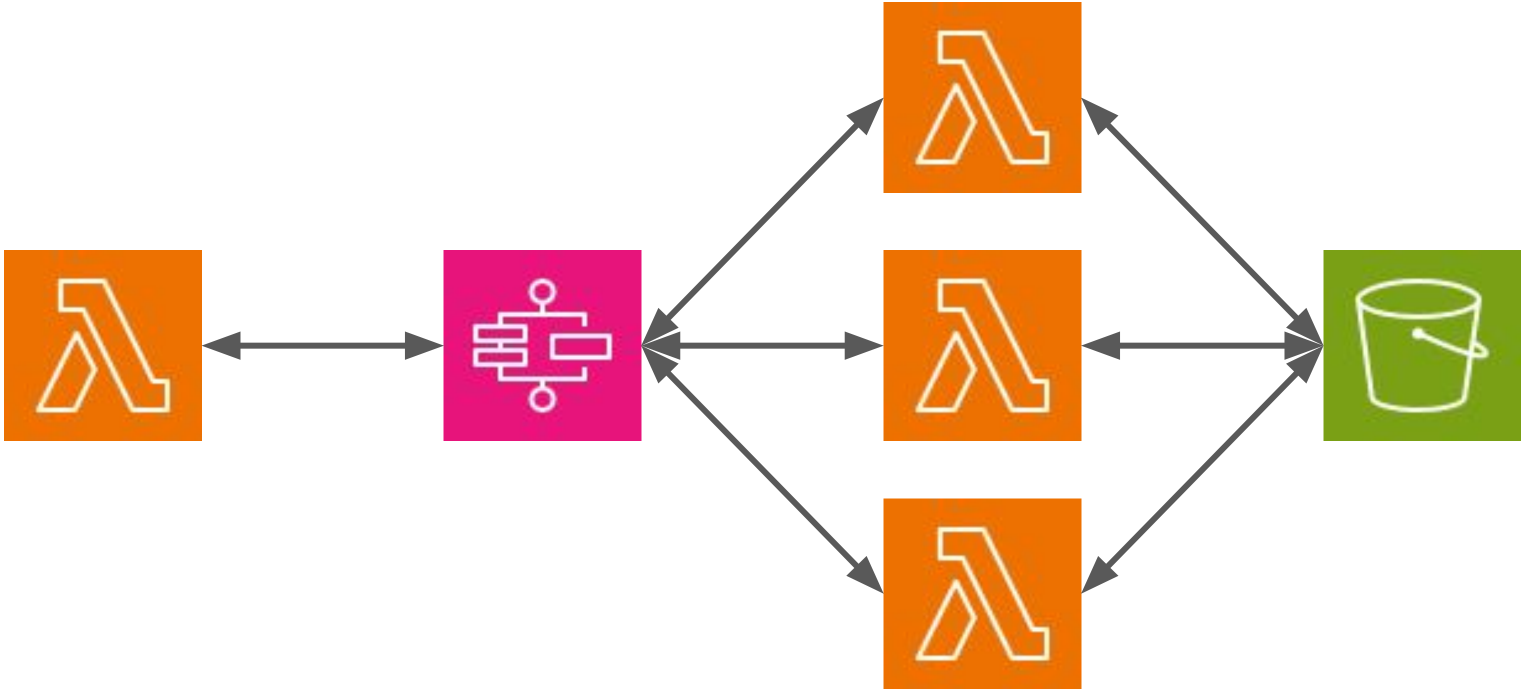
- Starling: A Scalable Query Engine on Cloud Functions
  - A query engine built to provide ad-hoc query processing, which allows users to only pay for resources the user actually uses. This focuses on analytical workloads.
- Building a database on S3
  - Discuss the ability of constructing B+ Tree Indexes on S3 and consistency in a completely stateless cloud setting.
- Lambada: Interactive Data Analytics on Cold Data Using Serverless Cloud Infrastructure
  - A serverless distributed data processing framework

# Introduction - Our Experiment

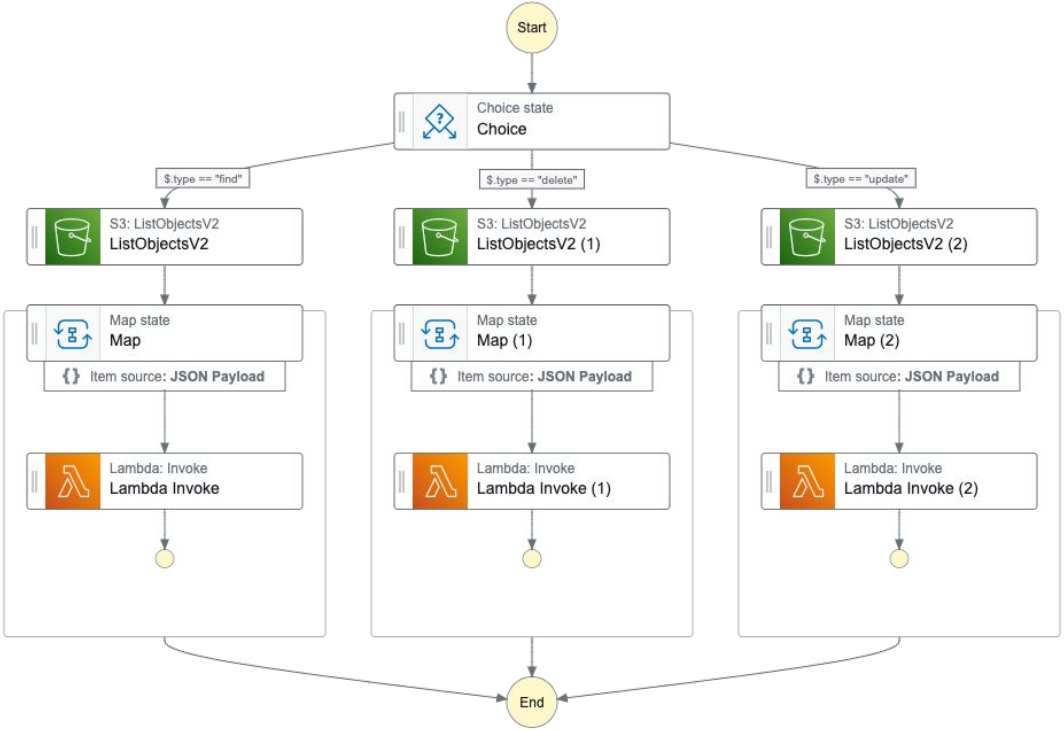
- Our main goal is to explore the application of AWS Lambda functions to simulate a database system in a cloud environment.
- This  Project will review the proposed system with the traditional on premises databases, DynamoDB and DocumentDB with regards to their performance.



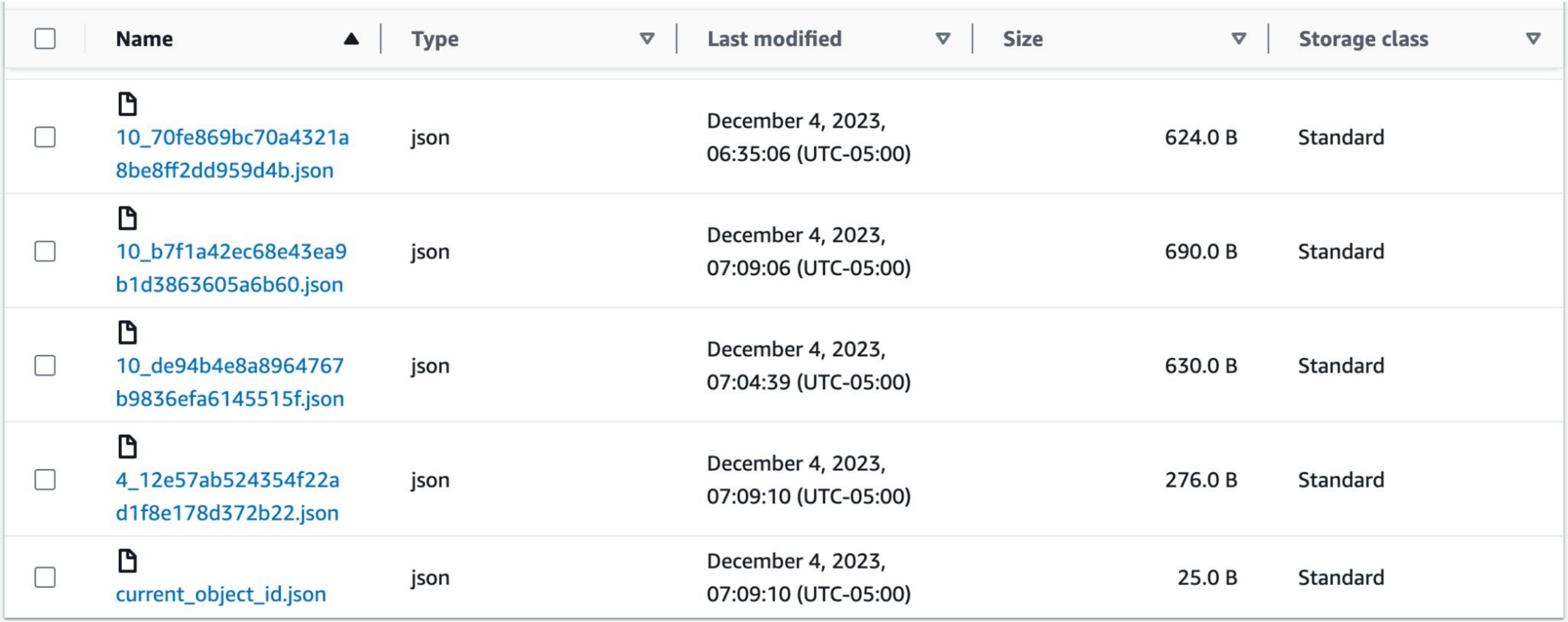
# Methodology - Our Approach



# Methodology - Our Approach



# Methodology - Our Approach



| □ | Name ▲ | Type ▽ | Last modified ▽ | Size ▽ | Storage class ▽ |
|---|---|---|---|---|---|
| □ | 10_70fe869bc70a4321a8be8ff2dd959d4b.json | json | December 4, 2023, 06:35:06 (UTC-05:00) | 624.0 B | Standard |
| □ | 10_b7f1a42ec68e43ea9b1d3863605a6b60.json | json | December 4, 2023, 07:09:06 (UTC-05:00) | 690.0 B | Standard |
| □ | 10_de94b4e8a8964767b9836efa6145515f.json | json | December 4, 2023, 07:04:39 (UTC-05:00) | 630.0 B | Standard |
| □ | 4_12e57ab524354f22ad1f8e178d372b22.json | json | December 4, 2023, 07:09:10 (UTC-05:00) | 276.0 B | Standard |
| □ | current_object_id.json | json | December 4, 2023, 07:09:10 (UTC-05:00) | 25.0 B | Standard |

# Methodology - DynamoDB

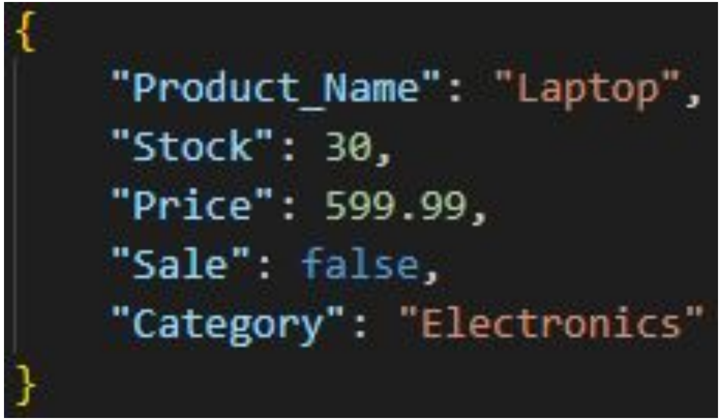
- Main NoSQL solution provided by AWS
- Baseline to compare our implementation against
- Set-up with default settings
- Connected to tables from local machine

# Methodology - DocumentDB

- The AWS DocumentDB, which has MongoDB compatibility, is a fully managed native JSON document database.
- Even though it is listed under AWS Free Tier, it has several limitations.
- Amazon DocumentDB is a virtual private cloud (VPC) only service, and doesn't support the use of public endpoints.

# Evaluation: Setup and Tests

- Setting up each of the different databases
    - Making a table for DynamoDB
    - Attempted to create DocumentDB cluster
- Online store test case
    - Insert/deleting products
    - Updating stock
    - Toggling sale
    - Getting price
- Performing Random Operations
    - Assigned different chances to different operations



```
{
    "Product_Name": "Laptop",
    "Stock": 30,
    "Price": 599.99,
    "Sale": false,
    "Category": "Electronics"
}
```

# Evaluation: Results

- Results for table with 1000 records

| | Lambda Serverless | DynamoDB |
|---|---|---|
| Record Count | 1000 | |
| Insert | > 500 ms | 20 ms |
| Search | > 1 s | 120 ms |
| Update | > 1 s | 240 ms |
| Delete | > 1 s | 20 ms |

# Conclusion - Summary

- Inspired by past works, the Starling system in particular
- Experimental serverless architecture of a NoSQL database
- The proposed system was benchmarked against DynamoDB, which performs significantly better than our system in each of the basic CRUD operations.

# Conclusion - Limitations

- AWS free tier
- Only supports equality conditions
- Does not operate on multiple search/update/delete conditions
- Lack of fine-grained security policies

# Conclusion - Future Work

- Testing for more records.
- Incorporate DocumentDB for benchmark.
- Concurrency control.
- Recovery management and availability.

# References

- Amazon S3. https://aws.amazon.com/s3/
- Amazon IAM. https://aws.amazon.com/iam/
- Amazon DynamoDB. https://aws.amazon.com/dynamodb/
- Amazon DocumentDB. https://aws.amazon.com/pm/documentdb/
- Get Started with Amazon DocumentDB. https://docs.aws.amazon.com/documentdb/latest/developerguide/get-started-guide.html
- Matthew Perron, Raul Castro Fernandez, David DeWitt, and Samuel Madden. 2020.Starling: A Scalable Query Engine on Cloud Functions. In Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data (Portland, OR,USA) (SIGMOD '20). Association for Computing Machinery, New York, NY, USA,131–141. https://doi.org/10.1145/3318464.3380609.
- Matthias Brantner, Daniela Florescu, David A. Graf, Donald Kossmann, and Tim Kraska. 2008. Building a database on S3. In Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC,Canada, June 10-12, 2008, Jason Tsong-Li Wang (Ed.). ACM, 251–264. https://doi.org/10.1145/1376616.1376645.
- Ingo Müller, Renato Marroquín, and Gustavo Alonso. 2020. Lambada: Interactive Data Analytics on Cold Data Using Serverless Cloud Infrastructure. In Proceedings of the 2020 International Conference on Management of Data, SIGMOD Conference 2020, online conference [Portland, OR, USA], June 14-19, 2020, David Maier, Rachel Pottinger, AnHai Doan, Wang-Chiew Tan, Abdussalam Alawini, and Hung Q.Ngo (Eds.). ACM, 115–130. https://doi.org/10.1145/3318464.3389758.

# Thank You

# Q & A