CS 340
Computer Systems

#19: Data Stores and Cloud Architectures

October 27, 2022 · Wade Fagen-Ulmschneider

Common Data Storage Options:

Scope	Data Storage	Interface	Technology
Local	**Variables**	Lang. Feature	All Programs!
	Files	File or File-like	open / fopen
Remote (Cloud)	**Object Storage**		S3 / MinIO
	Key-Value DB	Dictionary API set / get	redis
	Document DB	JSON find / update / insert	mongodb
	Relational DB	SQL	mysql / postgres
	Special Purpose	Objects	neo4j

The Illini "Coin Flip" Game Architecture:

Using Local Variable Storage for the Wallet Service:

```
19/wallet-service-local/app.py
d = {}
…
def createUser(sessionID):
  d[sessionID] = 100
…
  d[sessionID] = userData["amount"]
```

Using a Key-Value Store (redits) for the Wallet Service:

```
19/wallet-service-kvstore/app.py
kvStore = redis.Redis()
…
def createUser(sessionID):
  kvStore.set(sessionID, 100)
…
  kvStore.set(sessionID, userData["amount"])
```

Using a Document Database (mongo) for the Wallet Service:

```
19/wallet-service-documentdb/app.py
mongo = MongoClient(port=27017)
db = mongo["IlliniCoin"]["users"]
…
def createUser(sessionID):
  userData = {"amount": 100, "sessionID": sessionID}
  r = db.insert_one(userData)
…
  db.update_one({"sessionID": sessionID}, { "$set":
{"amount": userData["amount"]}})
```

Cloud Architectures

Q: What are cloud architectures?

Three Primary Design Patterns for Cloud Architectures:

[1]:

[2]:

[3]:

Monolithic Software Architecture:

Characteristics:

A (Small) Monolithic Example: Illinois Open Source Queue

Endpoint: https://queue.illinois.edu/
Source Code: https://github.com/illinois/queue

Microservice Software Architecture:

Characteristics:

Microservice Example:

PiggyMetrics by Alexander Lukyanchikov
GitHub: https://github.com/sqshq/PiggyMetrics
via: https://github.com/davidetaibi/Microservices_Project_List

Serverless Software Architecture:

Characteristics:

Serverless Examples:

adoptable-pet-bot/serverless.yml
https://github.com/lynnaloo/adoptable-pet-bot/blob/master/serverless.yml

```
provider:
  name: aws
  runtime: nodejs4.3
  region: us-east-1
…
functions:
  tweetPet:
    handler: handlers/tweetPet.tweetPet
    description: Tweets Adoptable Pets on a Schedule
    memorySize: 512
    timeout: 10
    events:
      - schedule: rate(6 hours)
```

emojibot/serverless.yaml
https://github.com/markhobson/emojibot/blob/master/serverless.yml

```
provider:
  name: aws
  region: eu-west-1
  stage: dev
  runtime: nodejs14.x
  memorySize: 128
…
functions:
  event:
    handler: src/handler.event
    events:
      - http: POST /event
  explain:
    handler: src/handler.explain
    events:
      - http: POST /explain
```

serverless-image-labeller/serverless.yml
https://github.com/nileshprasad137/serverless-image-labeller/blob/master/serverless.yml

```
provider:
  name: aws
  logs:
    restApi: true
  runtime: python3.7
…
functions:
  labelOnS3Upload:
    handler: handlers/S3UploadHandler.labelOnS3Upload
    events:
      - s3:
          bucket: ${self:provider.environment.SERVERLESS_IMAGE_LABELLING_BUCKET}
          event: s3:ObjectCreated:*
          existing: true
  getImagesByLabel:
    handler: handlers/getImagesByLabelHandler.getImagesByLabel
    events:
      - http:
          path: getImagesByLabel
          method: post
          cors: true
```