15-319 / 15-619
Cloud Computing

Recitation 6

October 6th, 2020

Overview

- **Last week's reflection**
  - OLI Unit 3, Module 7, 8, 9
  - Quiz 4
  - Project 2.2
- **This week's schedule**
  - OLI Unit 3 - Module 10, 11, 12
  - Quiz 5
  - Project 2.3
  - Project 3.1 (Optional)
  - Project 2.1 Code Review
  - OPE Session
  - 15-619 Team Formation

Last Week

- **Unit 3: Virtualizing Resources for the Cloud**
  - Module 7: Introduction and Motivation
  - Module 8: Virtualization
  - Module 9: Resource Virtualization - CPU
- **Quiz 4**
- **Project 2.2, Containers: Docker and Kubernetes**
  - Docker Intro / Embedded Profile Service
  - Intro to Helm Charts / Deploying MySQL
  - WeCloud Chat Microservices Architecture
    - Autoscaling, Multi-Cloud and Fault Tolerance to Azure

This Week

- **Unit 3: Virtualizing Resources for the Cloud**
  - Module 10: Resource virtualization (Memory)
  - Module 11: Resource virtualization (I/O devices)
  - Module 12: Case Study
- **Quiz 5**
- **Project 2.3, Functions as a Service (FaaS)**
  - Task 1, Explore functions on various CSPs
    - Azure Functions, GCP Cloud Functions, AWS Lambda
  - Task 2, Extract thumbnails from video stream
    - Azure Functions and FFmpeg
  - Task 3, Get image labels and index
    - Azure Computer Vision, Azure Search

Also This Week

- Primers released this week
  - Introduction to multithreaded programming in Java
  - Profiling a Cloud Service
  - Storage I/O benchmarking
- Project 2.1 Code Review due on Sunday 10/11
- OPE training session due on Sunday 10/11
- [for 15-619 students] Make sure your teams are formed **on TPZ** by Friday 10/09
- [optional] Project 3.1

Team Project - Time to Team Up

15-619 Students:

- Start to form your teams
  - Choose carefully as you cannot change teams
  - Look for a mix of skills in the team
    - Web tier: web framework performance
    - Storage tier: deploy and optimize MySQL and HBase
    - Extract, Transform and Load (ETL)

- Create a new AWS account only for the team project

15-319 Students:

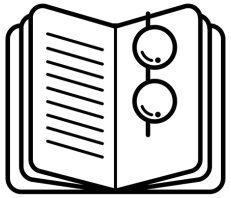
- You are allowed to participate in the team project *if* you are an undergraduate
- **Once committed to a team, you cannot quit**
- You will earn a significant bonus for participating in the team project
- If you are a 15-319 undergraduate student and want to participate in the team project, please make a private post on piazza.

Team Formation - Deadlines

Follow the instructions on Piazza @346 carefully

- **By Friday 10/09 at 11:59 PM ET**
  - Identify your team members
  - One team member should form a team on TPZ and all other team members should accept the invitation
    - Completing this step will freeze your team
- **By Saturday 10/10 at 11:59 PM ET**
  - Create a new AWS account ⇒ only used for the team project
  - Update the team profile in TPZ with the new AWS ID `aws-id`
- **By Sunday 10/11 at 11:59 PM ET**
  - Finish reading the Profiling a Cloud Service primer to get yourself prepared for the team project

This Week: Conceptual Content



UNIT 3: Virtualizing Resources for the Cloud

Module 7: Introduction and Motivation
Module 8: Virtualization
Module 9: Resource Virtualization - CPU
Module 10: Resource Virtualization - Memory
Module 11: Resource Virtualization – I/O
Module 12: Case Study
Module 13: Storage and Network Virtualization

OLI Module 10 - Memory Virtualization

- A process that cannot fit into the physical memory? To run or not to run?
- Page Table
  - Per process
  - Maps virtual addresses to physical addresses
- One level vs. two levels mapping
- Virtual, Real, Physical address spaces
- Memory reclamation
- Ballooning

OLI Module 11 - I/O Virtualization

- How?
  - Construct a virtual version of the device
  - Virtualize the I/O activity routed to the device
- I/O Basics
- System call interface, device driver interface, and operation-level interface

OLI Module 12 - AWS Case Study

This Week's Project

- P2.1: Horizontal Scaling and Autoscaling
  - Horizontal scaling in / out using AWS APIs
  - Load balancing, failure detection, and cost management on AWS
  - Infrastructure as Code (Terraform)
- P2.2: Docker Containers and Kubernetes
  - Building your own container-based microservices
  - Docker containers
  - Manage multiple Kubernetes Cluster
  - Multi Cloud deployments
- P2.3: Functions as a Service
  - Explore Cloud Functions provided by Azure, GCP, Azure
  - Building a serverless application using Azure Functions
  - Deploy multiple functions to build a video processing pipeline

Project 2

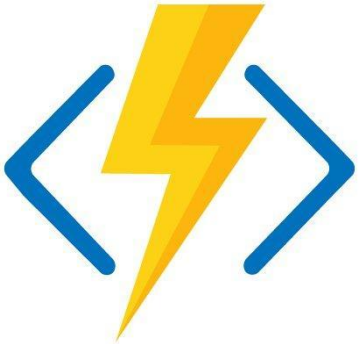
Running Theme:
Automating and scaling distributed systems

“Serverless” Computing

- The Cloud Service Provider (CSP) provides the server to run the application.
- Develop and run applications on servers without having to provision or manage servers or worry about scaling.
- Pay-per-invocation model.
- Functions-as-a-Service (FaaS) is a use-case of serverless computing.
- Applications can have one or more functions.

Cloud Functions (1/2)

- Possible use cases
  - Chat bots
  - Mobile backends
- How about Mapreduce?
  - Not suitable for FaaS
  - EMR is PaaS
- FaaS, typically stateless functions
- Pay per number of function invocations + running time
- Scalability is automatic through provider



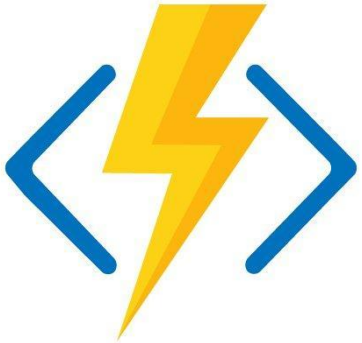
Cloud Functions (2/2)

- High availability
- Resiliency
  - Each execution is contained and isolated, and thus has no impact on other executions
- Built in logging and monitoring
  - Easily accessible logs ensure traceability
- Event-driven
  - Functions can be triggered by events like Blob storage file uploads

Triggers

- Events trigger functions
- Events are passed to function as input parameters
- Event sources publish events that cause the cloud function to be invoked
  - Azure: HTTPTrigger, BLOBTrigger, Event Grid etc.
  - AWS Lambda: Create object in S3 bucket, SNS topic etc.
  - GCP: File upload to Cloud Storage, message on a Cloud Pub/Sub topic etc.

Azure Functions



- Only pay for the number of invocations and the running time (i.e., when your code runs)
- Stateless…
  - No data is persisted across invocations
  - You can use persistent storage to maintain the state
- Debugging?
  - Use Application Insights

Azure Function Model

- Triggers and events (HTTP Triggers, Blob Storage Triggers etc.)
- Context
  - Allows function to interact with Azure Function execution environment e.g. to get function name, function directory, ... etc.
- Logging and Monitoring

Project 2.3 - Overview

- Task 1
  - Fibonacci function in Azure Functions
  - Power sets function in GCP Cloud Functions
  - CIDR block statistics in AWS Lambda
- Task 2
  - Azure Functions, Azure Event Grid and FFmpeg to generate thumbnails
- Task 3
  - Azure Cognitive Computer Vision and Azure Cognitive Search to label thumbnails and create index for video search

P2.3 Task 1: Cloud Functions

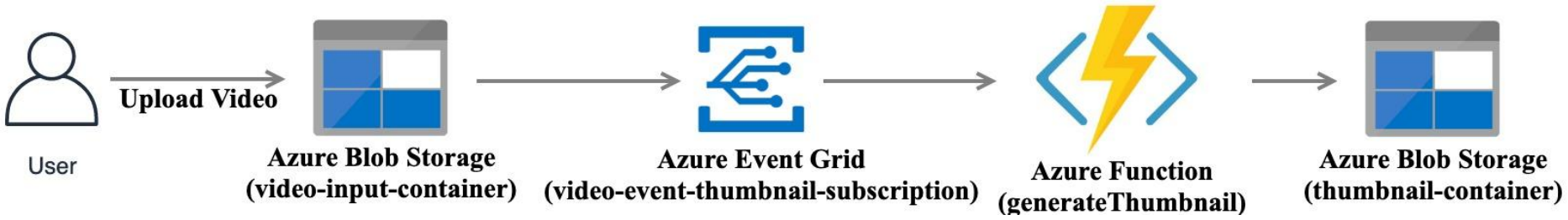
Goal: Explore HTTP Triggered Functions

- Subtask 1
  Fibonacci - Azure Functions
- Subtask 2
  PowerSets - GCP Functions
- Subtask 3
  CIDR - AWS Lambda
  - Java: use SubnetUtils class from Apache's Commons Net.
  - Python users should consider the python-iptools package or ipaddress from the Python standard library.

P2.3 Task 2: Event Driven Functions

Goal: Build a Serverless Processing Pipeline using Event driven functions and Azure Event Grid

- Azure Event Grid to trigger the function
  - An event delivery mechanism which allows Functions to subscribe to event topics.
  - Provides a reliable event delivery and event filtering functionality.
- Azure Functions and FFmpeg to process videos

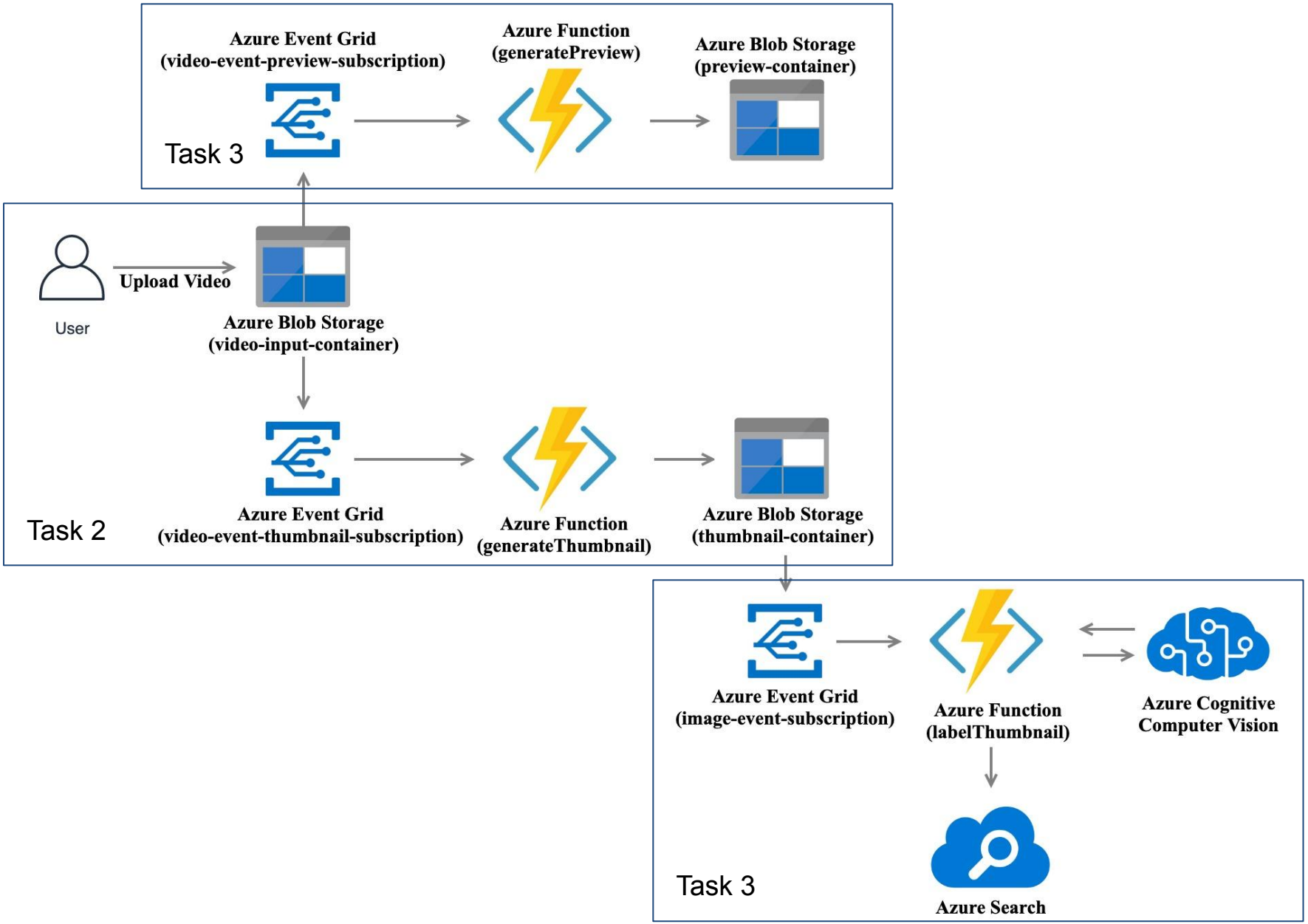


Project 2.3 - Task 3

Goal: Add two more functions into the Pipeline to assemble a Video Processing Pipeline

- Use FFmpeg to generate preview from videos
- Azure Cognitive Computer Vision for labeling thumbnails
- Azure Cognitive Search to index videos based on thumbnail labels

Project 2.3 - Task 3



Azure Cognitive Computer Vision

- Image recognition service

- Pricing!!
  - $1.00 per 1,000 images
  - During testing BE CAREFUL of the budget
  - Be careful not to exhaust your budget

Azure Monitoring and Alerts

- To mitigate overspending on Azure Computer Vision, we have provided you with an ARM template for Azure Alerts that **you must apply.**
- This template will create an alert that sends an email if you make more than 1000 Cognitive CV calls within 1 min.

Azure Cognitive Search

- Fully managed Azure service for maintaining search indexes
- Offers REST APIs for updating/accessing the indexes
- Upload thumbnail ids and tags

Project 2.3 - Hints

- You must authenticate the CLIs of Azure, GCP and AWS on the student VM before you submit your Task 1 solution.
- Review the recommended libraries in the writeup
- For example, ffmpeg + ffmpy works well in Python
- Remember to add dependencies to requirements.txt before deploying Azure Functions
- Test your functions and triggers manually by uploading video files to blob storage
- Review the Azure App Insights and Monitoring Logs
- Ensure that the output file name adheres to the specified format, i.e., “video_n.png” and “video.gif”; e.g., NOT “video.mp4_n.png”

Project 2.3 - Hints (Continued)

- If the submitter VM is not authenticated with Azure and is unable to delete, put, and get files from your blob container, or is not able to invoke your Azure functions, you will not be graded
- Make sure that you generate the labels using the visual features that are provided by Azure Cognitive CV, not the other features or products
- Use temporary folders to store images and videos during a function execution
  - Also, remember that functions are stateless.

Project 2.3 - Reminders

- Tag your resources with:
  Key: Project,
  Value: 2.3 on AWS and Azure, 2-3 on GCP
- Azure Cognitive Services are very expensive.
- Your subscription will be disabled if you run out of your subscription budget. Please exercise caution and **plan the budget**.
- Remember to delete the Azure resource groups to clean up all the resources after completion.

Project 3.1 - Files v/s Databases

- Primary learning objectives: Explain and compare the advantages and disadvantages of utilizing flat files, SQL databases, and NoSQL database solutions.

- This project is optional, but you are strongly encouraged to finish it,
  - if you want to learn topics about databases
  - if you want to get **hints for team projects**
  - if you want to get a higher final grade by using it to substitute a failed project.

Project 3.1 - Task 1

- Task 1 - Flat Files
  - Utilize tools such as awk, grep, and Python pandas library to analyze flat files.
  - You will use awk and pandas to execute queries on the yelp datasets in flat file format.
  - You should answer q1-q3.

Project 3.1 - Task 2

- Task 2 - MySQL
  - Utilize JDBC driver to write MySQL queries.
  - Implement and evaluate MySQL index.
  - Deploy Hibernate Application to learn how to use Object-Relational Mapping (ORM).
  - You should answer q4 and questions in MySQLTasks.java using MySQL Python and Java API, and also complete Business.java for ORM.

Project 3.1 - Task 3

- Task 3 - Redis
  - You will implement your own in-memory key-value store to understand how Redis functions.
  - Adopt test-driven development with JUnit.

Project 3.1 - Task 4

- Task 4 - HBase
  - Compare the schema design and data loading between MySQL and HBase.
  - You need design a good rowkey for HBase to avoid hotspotting.
  - You will answer the questions in HBaseTasks.java by HBase v1 API, and explain your rowkey design.

Upcoming Deadlines

- **Quiz 5 (OLI Modules 10, 11 & 12)**
  - Due on **Friday**, Oct 9th, 2020, 11:59PM ET
- **Team Project - Team Formation**
  - Due on **Friday**, Oct 9th, 2020, 11:59PM ET
- **OPE Training Session**
  - Due on **Sunday**, Oct 11th, 2020, 11:59PM ET
- **Project 2.1 Code Review**
  - Due on **Sunday**, Oct 11th, 2020, 11:59PM ET
- **Project 2.2 Reflection**
  - Due on **Sunday**, Oct 11th, 2020, 11:59PM ET
- **Project 2.3**
  - Due on **Sunday**, Oct 11th, 2020, 11:59PM ET
- **Project 3.1 (Optional)**
  - Due on **Sunday**, Oct 11th, 2020, 11:59PM ET

Team Formation - Deadlines

Follow the instructions on Piazza @346 carefully

- **By Friday 10/09 at 11:59 PM ET**
  - Identify your team members
  - One team member should form a team on TPZ and all other team members should accept the invitation
    - Completing this step will finalize your team

- **By Saturday 10/10 at 11:59 PM ET**
  - Create a new AWS account ⇒ only used for the team project
  - Update the team profile in TPZ with the new AWS ID `aws-id`

- **By Sunday 10/11 at 11:59 PM ET**
  - Finish reading the Profiling a Cloud Service primer to get yourself prepared for the team project

Q&A