Software Requirements Specification (SRS) Document

Control transfer between Lambda Functions for Cost Optimisation

Team 48

Maneesh Manoj

Kavish Kapoor

Aravind Jagannathan Tearle

Nikhil Sriram

Umang Patel

Brief Problem Statement

Ensuring optimum use of computing power when executing operations on a serverless function requires full prior knowledge of the task being executed. However, in cases this is not known it is not feasible to re-run the task on a function with larger resources if it does not complete on one with lesser, due to large costs involved. The goal is to transfer control seamlessly between lambda functions keeping current program state at every stage to minimise costs as much as possible.

System Requirements

- AWS Lambda
- AWS Cloudwatch

Users profile

User running a program on a Serverless function.

Feature requirements

No.	Use Case Name	Description	Release
1.	Transfer program state information	The current program state information is transferred to the larger lambda function, in case the previous function has run out of allocated computing resources	R2
2.	Randomly generate Van Der Monde Matrix	Generate the random matrix as a way to use computing power	R2
3.	UI (Unconfirmed)	Have a User interface in order to set memory between the lambda functions manually	-

Use Case Description

Use Case Number:	UC-01
Use Case Name:	Transfer program state information
Overview	The current program state information is transferred to the larger lambda function, in case the previous function has run out of allocated computing resources
Actors:	AWS Lambda function 1, AWS Lambda function 2, AWS Cloudwatch
Pre-Condition:	Function 1 is running
Flow:	1. Function 1 exhausts its resources 2. Current program state is recorded 3. Program state is stringified and sent as a JSON payload to function 2
Alternate Flow:	**-**
Post Condition	Function 2 continues where the program state had left off

Use Case Number:	UC-02
Use Case Name:	Randomly generate Van Der Monde Matrix
Overview	Generate the random matrix as a way to use computing power
Actors:	AWS Lambda function 1
Pre-Condition:	Function 1 is running
Flow:	1. Function 1 runs python code to generate random matrix cell
Alternate Flow:	1. Function 1 runs out of computing resources. 2. Current program state is recorded and UC-01 is executed
Post Condition	Function 1 continues until it has finished generating the matrix