

# CS 6480: Project Update 7

## Serverless or Function as a service

Aakanksha Saha

School of Computing, University of Utah

November 10, 2017

The following tasks were performed by me this week:

- Had a meeting with Kobus and Aisha to discuss the possible future direction. The problem of resource management of containers in OpenLambda was highlighted.
- It was discussed that compute nodes will be divided into three segments light, medium and hard weight.
- Similarly, the request workload (curl request) will be formulated in three ways light, medium and hard.
- For doing so I used the time tool in Ubuntu to time the events called in OpenLambda. Any event call taking less than 500ms will be assigned to the light weight container and so on.
- For distinguishing the containers based on their resource requirement I have looked through resource requests and limits option in containers.
- CPU resources- Each container is assigned a “share” of the CPU. CPU share is set using the `-c` or `-cpu-shares` flag when launching the container.
- Memory resources- Currently controlling the amount of allocated memory and swap separately is not possible in Docker. By default, when a container is launched there are no set memory limits, which can lead to issues where a single container can hog up all the memory and make the system unstable. Memory can be limited with a short command (`-m` flag).

For next week, with these resource optimization options I plan to integrate the memory and cpu optimization of the containers in OpenLambda basic code. I also plan to formulate a script to differentiate workloads running in the containers.