CS-E4660 - Advanced Topics in Software Systems D

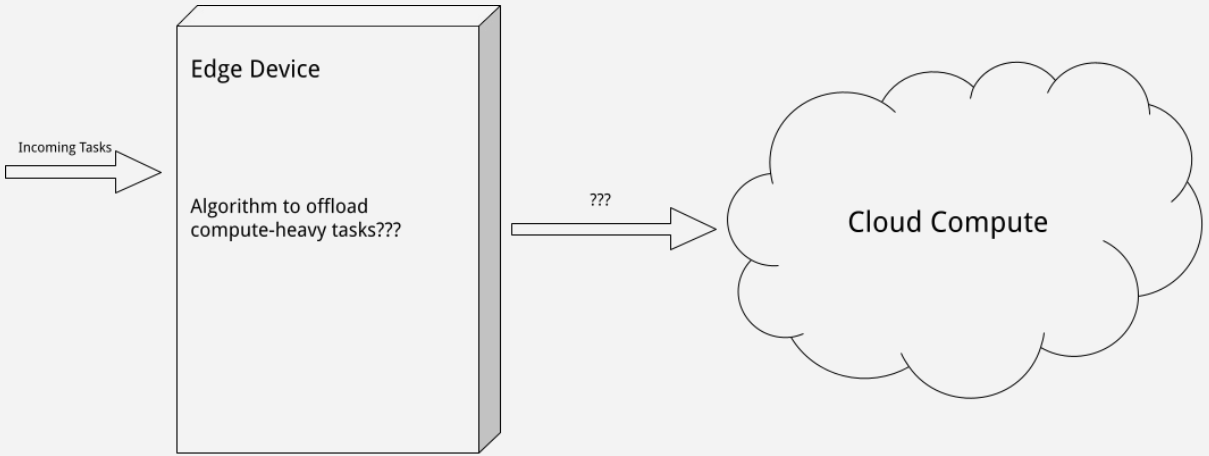
Dynamic Predictive Edge Offloading

Linus Jern

Research Question

Evaluate the *performance increase* and *consistency of offloading choice* of a dynamic prediction offloading algorithm.

Will different Cloud Compute design choices affect the *reliability*?



Learning Objectives

- Determine if it's possible to easily develop an algorithm that is able to *reliably* offload the correct tasks to decrease total system latency.

Assumptions

- The compute will happen on computationally constrained "edge" device.
  - In the real world, this might be a drone, robot, camera etc.
  - In my example project, I plan on using Kubernetes pods with restricted resources.
- In addition to the "edge" devices, cloud compute is available on demand if needed.
  - Could be Serverless, Workers, Fixed Server etc.

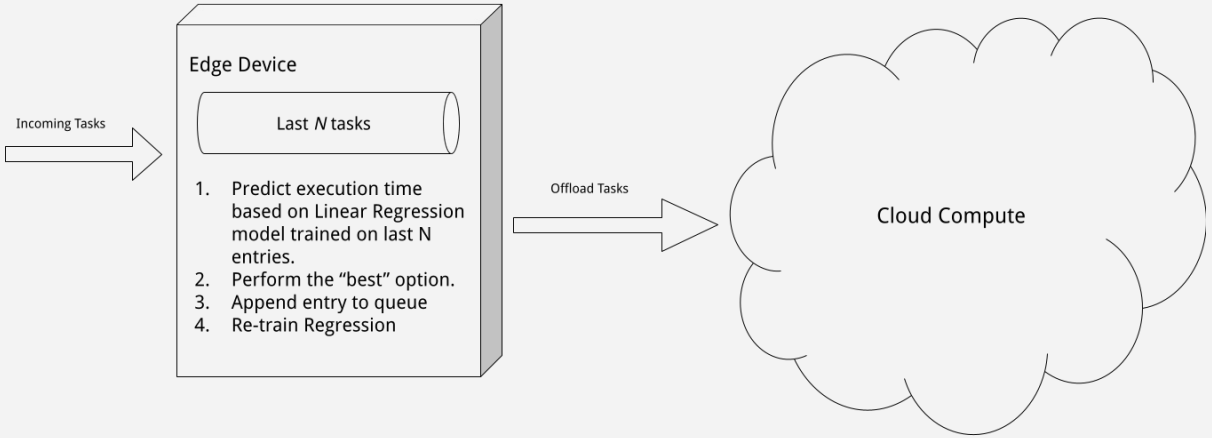
Details

Tasks:

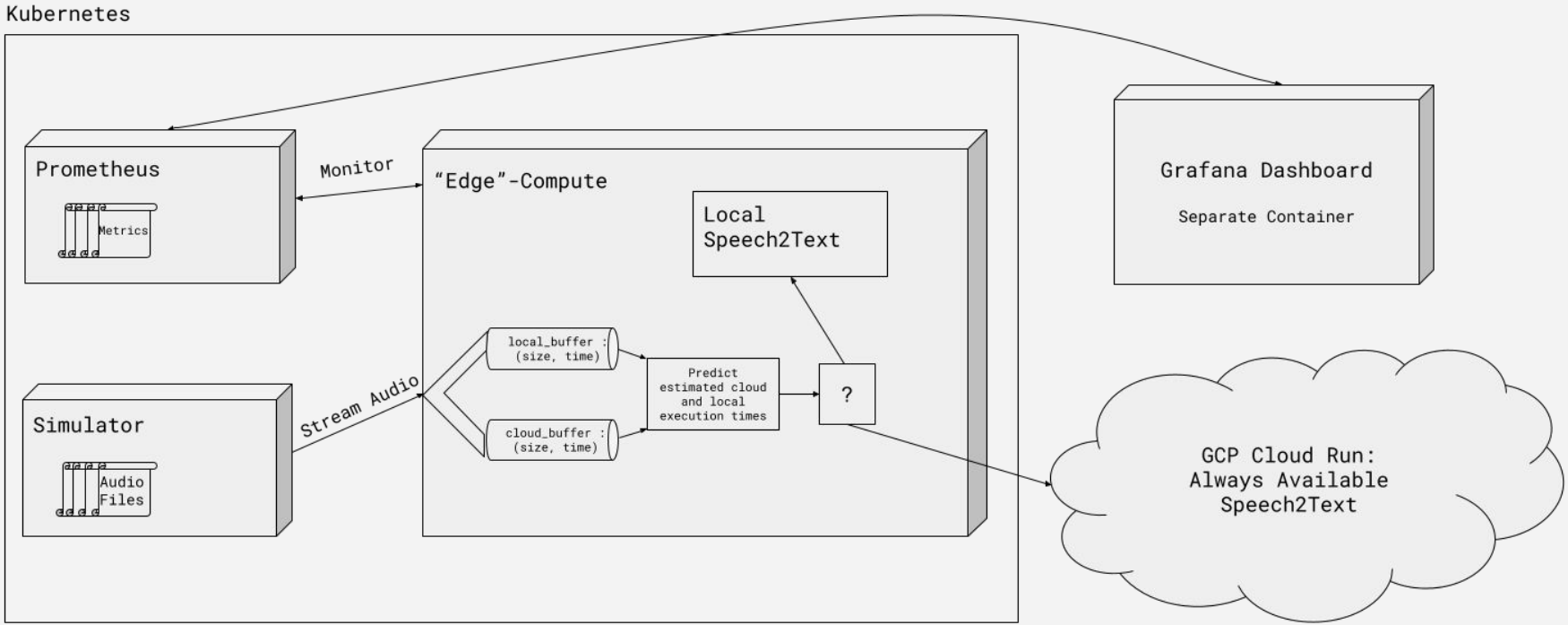
- NLP Speech-to-text
- Variable input length -> variable compute requirements

Implementation:

- Algorithm
  - Dynamically train a ML based regression to estimate t_l and t_c and determine which to choose by total time.
- Train the, very lightweight (and hopefully quick), ML regressor after each task on the last N tasks performed.
- Store relevant metadata from last N tasks in local queue. Example of metadata:
  - Task size (character length etc.)
  - Actual time of completion
  - Etc.
- Monitor this decision and accuracy after retraining to gain insight into reliability



Implementation



Algorithm - Pseudocode

```
Init buffers B_l and B_c
For each audio file:
    fit model m_l and m_c
    pred_l, pred_c <- predict incoming
    If B_l == B_c == N:
        if pred_l < pred_c:
            execute locally
            add to B_l
        else:
            execute cloud
            add to B_c
    else:
        Execute both
        add to both Bs
```

Demo

- Source Code: https://github.com/LinuzJ/sys4bigml-work-repo/tree/main/project