15745 Project Proposal
Compiler optimization for micro services in Cloud

Sanil Rao (sanilr) Pil Jae Jang (piljaej),

March 2020

URL

Project Web Page:
`https://piljaej.github.io/15745_project/index.html`

Project Description

Since its inception, the cloud has been an integral part of many modern businesses. The cloud allows corporations to scale out their applications without having to spend large sums of money acquiring all the underlying hardware. As such, many popular applications have moved from locally hosted to cloud hosted using one of the large cloud providers.

When moving to a cloud environment the application itself also has to change. It is no longer ideal to have it as a giant monolithic application. Instead, the application should be broken up into smaller micro-services that communicate with one another. This reconfiguration better fits the underlying structure of modern cloud computing centers. From a compiler perspective these new micro-service applications provide an interesting challenge for optimization. Given how novel this application design approach is, not much time has been spent on exposing optimizations to the compiler. For example, these micro-services are constantly communicating with one another, which could be very high cost. Exposing which services talk most frequently to each other could allow the compiler to better schedule/place those services, minimizing the communication cost.

The goal of our project is to implement a compiler optimization that finds high communication between different micro-services and minimize this communication cost at run time. Our 75% goal is the implementation of a compiler pass that will list, in order of time spent, the remote procedure calls between micro-services. Our 100% goal is to use this information in a meaningful way to provide better placing of these micro-services within the cloud environment. Our 125% goal is to extend this framework beyond the one application we currently have.

Logistics

Plan of Attack

- **Week 3/23** - Be able to run the application and run the framework
- **Week 3/30** - Learn the framework in more detail / learn how to program java / begin writing pass
- **Week 4/6** - Continue to write the pass and look for how that pass can be useful
- **Week 4/13** - Have finished the pass and focus on scheduling and next steps with Professor Mowry
- **Week 4/20** - Work on suggestions from milestone meeting
- **Week 4/27** - Wrap up work and put into poster

The schedule for our project is described at a high level above. Our work will be divided in that we will each write our own compiler passes initially and come together to see how they worked or didn't work. Essentially seeing if we can get our implementations to work similarly, and see what extra features we were able to get form our framework.

Milestone

We hope that by April 16th, we will have successfully implemented a compiler pass for our application that will be able to list all remote procedure calls in the function in order of time spent. Meaning the calls with the largest amount of communication first to lowest amount of communication. From here we can go on to model how this would be beneficial in a cloud ecosystem and how this can be useful in a JIT context.

Literature Search

We were pointed to a few resources by Professor Mowry for this project. One was a talk given by a compiler group from IBM that discussed the compiler framework we would be using. Its design is tailored for micro-service based applications.

Resources Needed

At this time we don't need any specific hardware resources as we can perform this project on our local machines. We will be using the IBM compiler infrastructure Eclipse OMR for our implementation and the application ACME Air as our benchmark. We currently have all this software available to us.

Getting Started

At this time we have downloaded all the necessary software and are currently setting it up on our respective machines. We expect to be up to speed with the software relatively quickly. Our project has begun as of this week March 23rd.