Project Proposal

Aidai Beishekeeva, ab5248
Yaxin Chen, yc3995

Team Name: dsh

Topic: Distributed Shell

Motivations:

1. Aidai
   I want to study the use and functionality of distributed shells because I believe they tremendously increase the efficiency of work for a programmer. I also think that the topic provides deep insights into how distributed systems work because me and Yaxin will be studying the implementation of existing distributed shell tools.

2. Yaxin
   I am interested in distributed shells because I want to study how software can be extended by distributed systems. Shell is a powerful tool used by programmers to execute commands and manage systems, and it can be distributed in several ways. I would like to explore the features of existing distributed shells and learn about their implementations.

Project Description:
We would like to compare and analyze features of commonly used distributed shell tools. For now, we are thinking of comparing these three: Dash, Dsh, Posh.
Some of the implementations we want to compare and their impact on features include handling jobs in parallel, configuration complexity, ability to handle connection timeouts, number of target machines it can be used on.
To compare these tools we will most likely set up a server cluster on Google Cloud and test the functionalities of the shells on that cluster.

Optional:
If time permits, we would like to build a distributed shell on top of Raft. This distributed shell is expected to take shell command from clients (server managers) and execute this command on multiple servers simultaneously. There are some existing distributed shells that have similar features, such as Dsh and Dash, but most of them deal with the single-client case. We want to extend it to multiple clients, i.e., multiple managers may send commands at the same time to the servers, and the servers must reach

consensus on the execution order of these commands. We plan to use Raft as the consensus protocol. As for programming language, we plan to use Go to implement this shell, because Go is good at dealing with concurrency and widely used in distributed systems development. What's more, we could utilize the Raft library in Go (https://pkg.go.dev/github.com/hashicorp/raft).