CS 425 MP 1 Report

Xilun Jin and Yuchen Liang

I. Introduction

In this MP, we implemented a distributed log system, where a central query machine receives and then outputs logs combined from each distributed machine in the cluster. We use Python as our main language for development.

II. Algorithm and Analysis

We use socket to transmit log files between different machines. There are two scripts in the repository, *new_server.py* and *new_client.py*. The former script is run as the server process, where it sets up a socket listening to connection. Whenever a connection request is sent to the server, it executes the message as a command and sends back the result line by line, ending with byte 0x0. The latter script is run as the client process, where it sets up threads, each of which independently controls a connection to one of the cluster machines. The thread sends command (i.e. grep) to the machine, receives feedback line by line and outputs to terminal accordingly, and returns a byte message (i.e. 0x1) after receiving each line. This acts as an ACK signal.

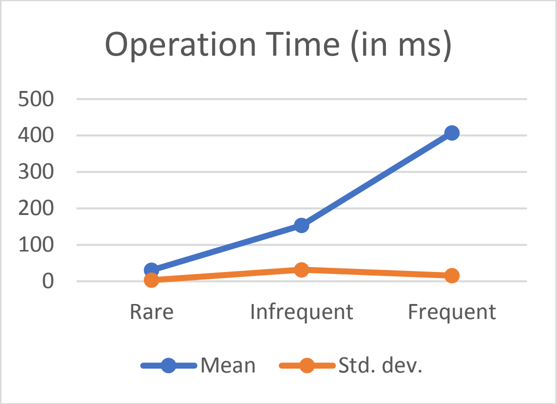
If one of the server crashes, the thread that controls that server will output an error message, but other threads controlling the rest of the cluster machines are running independently and will not be affected.

We also implemented another script, *log_data_generator.py*, to unit-test the distributed grep functionality. This function generates logs of frequent and infrequent patterns to a specified number of machines. The logs are generated as a random permutation of words in a given list, where some words are specified to occur more frequently than others. The resulted line count and output are compared on the distributed system and the local system.

III. Test Result

	Rare		Infrequent		Frequent	
	Time (ms)	# logs	Time (ms)	# logs	Time (ms)	# logs
Measure #1	25	6	168	5069	433	14899
Measure #2	35	8	203	6798	344	12724
Measure #3	30	7	118	3651	393	15872
Measure #4	33	6	125	3875	418	15668
Measure #5	29	9	179	4193	435	15328
Measure #6	31	5	127	3250	421	16275
Mean	30.5	6.8	153.3	4472	407.3	15127
Std. dev.	3.1	1.3	31.8	1180	15.7	1157

From the table and plot, we see that the time to run the distributed grep algorithm depends on the number of logs generated on each machine. As the text grepped becomes more frequent, the time required also increases linearly. This makes sense because transmission is more time consuming than producing the grepped result within each machine, where the time necessary to transmit the log file grows linearly with the number of lines of the file.


Operation Time (in ms)
Rare
Infrequent
Frequent
Mean
Std. dev.