CSP Test VASTech Documentation

This is thr documentation for the CSP test by VASTech. This document contains both parts of the test broken in to two sections, Design- and Coding Challenge.

Marcell Oosthuizen

VASTech, 2019
Technical Report

Abstract

Reading through the test PDF that was sent I found that this test relies heavily on the understanding of how to configure a distributed system and why it is important to configure it correctly. In the first section "Design Challenge" the test enforces the importance of Network Time Protocol with the implementation of the ChronyD method. I was further assessed on different aspects of the configuration of the source file and some other questions relating to the subject of NTP importance. After that I was tasked to develop multiple services for which, I was given the choice to use Java,Golang and Python for which I opted to use the latter. These services included but not limeted to temperature-service, memory-usage-service, cpu-load-service, byte-count-service, packet-length-service and a nasty service. I was tasked to Implement them as a single process and then split up into multiple.

Contents

1 Design Challenge **1**

1.1 Introduction 1

1.1.1 Question 1 1

1.1.2 Question 2 2

1.1.3 Question 3 2

2 Coding Challenge **4**

2.0.1 Introduction 4

3 Thoughts and experiences **7**

3.0.1 Design Challenge 7

3.0.2 Coding Challenge 7

Chapter 1

Design Challenge

1.1 Introduction

The approach I took to do this Challenge was research in nature, at first I read the questions and then scheduled time sections for me to familiarise myself with the NTP protocol and since it has been a while since I had Networking. I did this by looking at white papers that I will add as references. After that, I studied more practical setups like the manual and blog-posts and further documentation to ensure I am fully familiarized with the subject. Then I switched over to Chrony and worked through how-to guides as well as the user manual to ensure I understand how it works. Only thereafter I attempted the questions.

A time Breakdown that represents this section :

- Research : 8 hours
- Questions : 2 hours
- Setting up of latex document and style : 1 hour
- Total : 11 hours

1.1.1 Question 1

1 : List the failures that could cause a single host to loose time synchronisation.

Normally a host can lose synchronization due to changes like the temperature of the crystal oscillator and quality [1]of the processor, but this is counteracted when having it synced to a ntp-server. So other failures that can make a host fail in synchronization is :

- A host that has one network interface and loses connection to the internal network for a prolonged period of time and does not have a second network interface.
- Strong Firewall keeping the host from synchronizing
- If somehow ip port 123 is closed

2: Suggest an alternative to the time sources configuration.

Master 1	Master 2	All other hosts
server ntp-server trust prefer	server ntp-server trust prefer	server master-1
server 127.127.1.0	server 127.127.1.0	server master-2

This configuration allows for less traffic through the firewall and reduces UDP traffic on the ntp-server(Stratum 1 server) I also removed prefer [2] from the master one since I believe this will split the load more evenly as well as let the hosts choose the best option for themselves in terms of delay.(Prefer would be canceled if the delay is of a significant difference)

1.1.2 Question 2

1: Would you say that this configuration is an improvement on the configuration provided in question 1? Why?

Yes, it is an and improvement on the first given configuration. One reason is that it is not pointing all the hosts in the network to the stratum 1 NTP server, which will decrease traffic through the firewall and help decrease the load on the stratum 1 reference server.

2: List the failures that could cause a single host to loose time synchronisation.

- A host that has one network interface and loses connection to the internal network for a prolonged period of time and does not have a second network interface.
- Strong Firewall keeping the host from synchronizing
- If somehow ip port 123 is closed

3: Suggest an alternative to the time sources configuration.

Master 1	Master 2	All other hosts
server ntp-server trust prefer	server ntp-server trust prefer	server master-1
server 127.127.1.0	server 127.127.1.0	server master-2

Made the Master 2 server also look at the external ntp server for time since if Master 1 has a problem it would not have to revert to 127.127.1.0 this makes master two more accurate. Other than that having master 2 look at master 1 makes it a Stratum 3 server, which is less accurate that Stratum 2 which it will be when preferring the NTP server. This makes sense because master 2 will be a client of master 1 and a client cannot be more accurate than the server. [3]

1.1.3 Question 3

1 : Is it a requirement for the hosts to always be synchronised to actual time? Why?

Yes, the reason for this is simple. As stated the reconstruction of events in a logical order is important when it comes to security and post processing. In terms of security it is used to restrict DDos attacks if the ntp server has restricted use of certain commands like monlist(returns administrative information of the recent clients that interacted with the server) and version [4]. Also in post processing like billing and event generation. Most financial firms require to be set up to a stratum 1 time source by law. [5]

2 Suppose the ntp-server was disconnected for some time. It is reconnected when all the hosts are synchronised to master-1, but master-1 is 60 seconds behind actual time. Describe the important factors that should be taken into account and how should the cluster by synchronised with actual time?

The Chrony daemon will slowly bring back the time of the master to the actual time of the NTP-server. As a network administrator, it is not needed to interfere with the system since the time of 60 seconds is not much of a worry for the NTP protocol. ChronyD will start to bring the time closer over a few iterations without causing confusion within the system by speeding up the system clock to gradually bring it back to actual time. The threshold time in terms of difference between the NTP source and client is normally 1000 seconds(insane time) [1] to value that the Administrator needs to use NTP date command to bring the time closer before the system can take over. But one of the differences is that in Chrony you can configure it to be even more. Something to remember is that the bigger the difference the longer Chrony will take to synchronize the time.

Chapter 2

Coding Challenge

2.0.1 Introduction

temperature-service
For this part of the problem given the nature of the task that have been assigned I thought I might extend the scenario to a raspberry pi which is stationed within the airport that collects these kind of data points. The temperature within the airport , adding notifications to it when temp gets to high.
memory-usage-service
The next service I will talk about is the memory usage, this service will post to the database on the state of RAM usage of the raspberry pi at that given moment. this process is not that important and will only be post to the database in increments of 10 seconds.
byte-count-service
The next service will be the CPU usage service, this service will monitor the CPU usage of the pi, this I service is a bit more important and will send updates ever second to the database.
packet-length-service NOT COMPLETED
Another service this host will be running is the receive byte counter, this counter will be looking at how many bytes of data this device is receiving though one of its network interfaces.
temperature-service
The last of the functions that will be run is the measuring of packet length but this has not been completed
Time Breakdown

- Planning : 2 hour
- Question 1 : 9 hours
- Question 2 : 4 hours
- Setting up of latex document and style : 2 hour
- total : 17 hours

Question 1

Planning
My initial idea was to have the functions in parallel run with pythons multi-threading library but got stuck with the fact that the functions was not behaving in the manner I have hoped it would work but sadly it did not. After this I tried quite a few ways using the python thread and multi-threading libraries but did not succeed so currently this questions processes is running sequentially. Another Idea I had was to run the processes concurrently using docker compose. At first I had problem dockerizing the files but I have got it working now. Both the python script as well as the database has been dockerized.

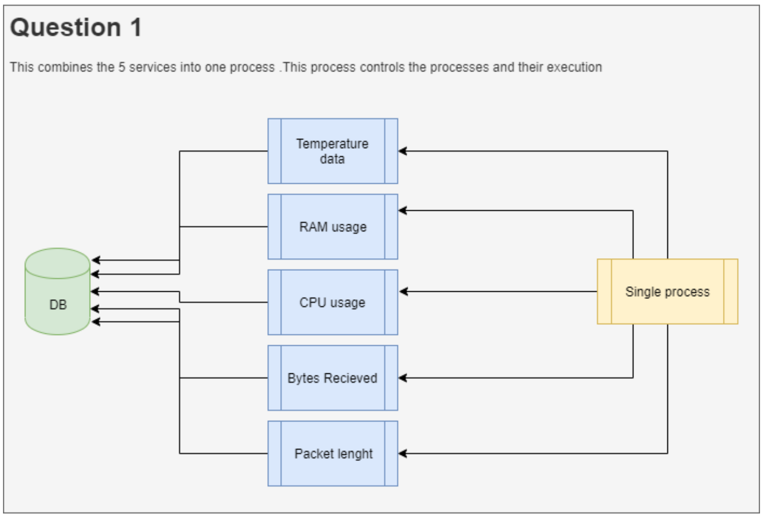
Q1.png 
Question 1
This combines the 5 services into one process .This process controls the processes and their execution
Temperature data
RAM usage
CPU usage
Bytes Recieved
Packet lenght
DB
Single process

Figure 2.1: Idea of a single process system

Implementation
My initial idea was to have the functions in parallel run with pythons multi-threading library but got stuck with the fact that the functions was not behaving in the manner I have hoped it would work but sadly it did not. After this I tried quite a few ways using the python thread and multi-threading libraries but did not succeed so currently this questions processes is running sequentially. Another Idea I had was to run the processes concurrently using docker compose. I did manage to get the files Dockerized but when I was attempting the implementation of running it it was not working as I expected it to. I also tried running it concurrently using a bash script that adds the processes to the background but this also ended up running sequentially.

```
#!/bin/bash

python cpu_usage.py &
python  Ram_usage.py &
python bytesRecieved.py &
python tempreture.py &
```

This is why I kept it as is and ran it as a single sequential process. **Requirements**
The imports that you require to run this code is (automatically handled by docker):

- time
- mysql.connector
- psutil
- datetiem
- multiprocessing
- and the various classes of the services

To run This is written in python2.7. Instructions on the execution is stated in the README file

Improvement to be Done
Improvement needs to be done to make the processes run concurrently alongside each other.

Question 2

Planning

I planned on using a flask server that acts as an interface between the processes and the database. This will be done by setting up routes for each of the processes. Then when the user enters the correct url the process will start running. **Implementation**

Q2.png

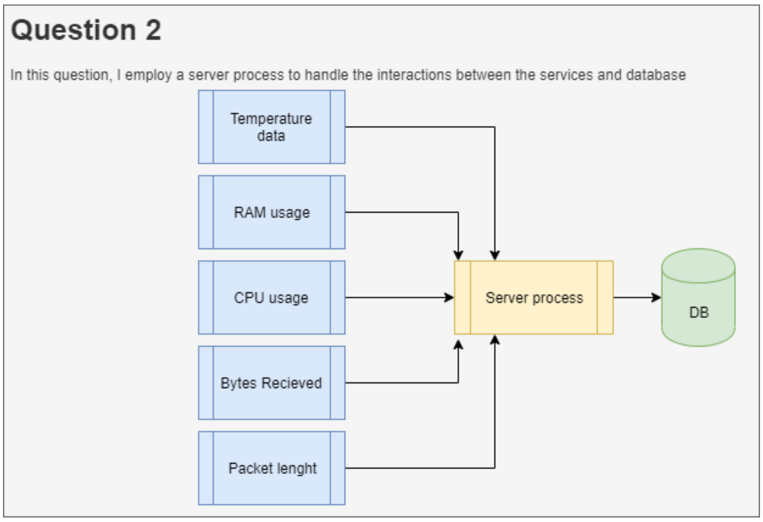

Question 2
In this question, I employ a server process to handle the interactions between the services and database
Temperature data
RAM usage
CPU usage
Bytes Recieved
Packet lenght
Server process
DB

Figure 2.2: Idea of a flask server

This implementation went better than the single process question. I was able to make a server using Flask that executes the the processes when called upon. **Requirements**
These are the packages that are required for this server to run :

```
from flask import Flask, request
from flaskext.mysql import MySQL
import time
import psutil
import mysql.connector
from datetime import datetime
```

To run

This is written in python3.6 . Instructions on execution is layed out within the README file.

Question3

This question I did not attempt since when the deadline was to close I thought it would be better to finish up what I have done and put more time into the report

Chapter 3

Thoughts and experiences

3.0.1 Design Challenge

This section was really interesting since I took quite some time to read up and refresh my mind on how NTP works. From reading white papers on why NTP is this critical protocol. The hard part for me was to read into how chronyd works since I have not worked with it yet.

3.0.2 Coding Challenge

First of I intended to use my raspberry pi as a remote device that sends the data and my laptop as the database. This worked well until I needed SU privileges to do some tasks and found out the default username and password has been changed(equipment of the university). Further more during this challenge I fell in to some unexpected traps. These traps include perusing paths to fix a problem to long. Further more since I haven worked with multi-threading that much I spent a lot of time trying to fix these errors. The second part of the question where I had to use a server I found a bit easier. I ran out of time coming to the end and feel like I could have done this task better.

Bibliography

[1] chrony, “Comparison of ntp implementations,” 2018.

[2] D. Both, “Manage ntp with chrony,” 2018.

[3] EndRunTechnologies, “Introduction to ntp,” pp. 1–4, 02 2004.

[4] T. Rytilahti, D. Tatang, J. KÃ¶pper, and T. Holz, “Masters of time: An overview of the ntp ecosystem,” pp. 1–15, 04 2018.

[5] Microsemi, “The importance of network time synchronization for enterprise solutions,” pp. 1–10, 2016.