

CS7NS1 Module Final Report

Student name: Roman Shaikh

Student number: 18300989

Date: 17th November 2018

Contents

CS7NS1 Module Final Report	1
InfernoBall team self-evaluation	2
What I learned.....	2
What I did.....	3
Module evaluation	5

InfernoBall team self-evaluation

- Team number: Team4
- Team members: - Roman Shaikh, Rujul Kapadia, Rory Hughes, Jordan Mayers.
- Effort must be high/medium/low/zero
- Effectiveness must be high/medium/low/zero

	Roman	Rory	Jordan	Rujul
effort	High	Medium	Low	High
effectiveness	High	High	Low	High

What I learned

This module has been taught me the power of scaling a problem. I learned about the various industries and applications like autonomous vehicle networks, the cryptocurrency market in which scaling is used as an efficient way of solving large problems quickly. The module also thought me to answer the right questions what to scale, when to scale and how to scale when thinking about scaling a problem.

The hash cracking assignments of Prof. Stephen were effective in teaching many skills like OpenCL and GPU scaling. It also thought me the power of Linux shell scripting and automating the tasks. Apart from this the last part of the assignment being a group assignment, it gave us a chance to experience how to work in teams and distribute a given task effectively.

The classes of Prof. Ciaran covered a ton of interesting topics like CPU/GPU scaling, cloud, and edge computing, IoT devices and their network etc. All this gave me a better understanding of how on a large scale a system works by effectively utilizing and distributing the tasks in hand.

The assignments for reading a research paper and summarizing it in a few sentences was defiantly an interesting one. It taught me to focus on the subject more and digging the academic resources thoroughly to find the correct points.

Apart from all this I also learned a lot about new tools and services which help us with scaling in practice. Like John the Ripper and hashcat which are effective in cracking the hashes using scaling techniques like OpenCL and OpenMP. The cloud computing services

AWS and Google Cloud Platform has been a major addition to my knowledge of various cloud resource available. Before taking this module, I had very limited knowledge about these services and were only a buzzword to me.

The guest lectures on vehicular IoT, WSN, and industry 4.0 by Ricardo Simon Carbajo were very informative and gave a sense into the current and future trends in the industry. They also helped in understanding the current industry needs which will (I hope) be helpful in searching and getting a good industry job after my course. Another interesting lecture on OpenCL programming by Andrew gave me a sense of confidence to work on OpenCL API before this OpenCL was like a big maze to me.

In conclusion, the module effectively taught me many aspects of scalable computing. From IoT to scaling on CPUs and GPUs, vertical and horizontal scaling, Cloud computing services and many more. All in all, this has been a great learning experience for me and I hope will be useful for my future professional life.

What I did

Ciaran's assignment: In this, we were required to read various research papers, surveys, and Transactions and after reading we had to summarize the paper with respect to key findings, key technological insights and relevance to cloud computing.

Throughout the three-part assignment I had gone and summarized the below mentioned academic references:

1. Tutorial Paper: Mobile Big Data: The Fuel for Data-Driven Wireless [DOI: 10.1109/JIOT.2017.2714189]
2. Review Paper: Big Sensor Data Systems for Smart Cities [DOI: 10.1109/JIOT.2017.2695535]
3. ICDCS Paper: Energy-Aware CPU Frequency Scaling for Mobile Video Streaming [DOI: 10.1109/ICDCS.2017.74]
4. Review Paper: GPU Based Strategies for Distance-Based Outlier Detection [DOI: 10.1109/TPDS.2016.2528984]
5. Survey Paper: Survey of Fog Computing: Fundamental, Network Applications, and Research Challenges
6. Survey Paper: A Survey on Resiliency Techniques in Cloud Computing Infrastructures and Applications

All the summarized papers are uploaded on google drive and can be accessed by links:

- <https://drive.google.com/open?id=1a-FHxzyajECKZvaGokcqNNeAki4CIEQA>
- <https://drive.google.com/open?id=1gzdSluRaqokfTog3pMGVAOedckltvTPY>
- <https://drive.google.com/open?id=1-lyHjBBBoQckNIVRgysvK1Cr2OoCBn84>

Stephan's assignments: In this assignments, we were given a task of cracking randomly distributed hashes. This assignment was incremental in terms of increasing complexity and challenges. Starting with the setting of rosettahub accounts and tools required for the assignment. All of the tasks were graded using submitted.

1. Assignment 1:

- a. The task was to set up an AWS instance with Rosettahub and install a community version of John the Ripper (a tool used for cracking hashes) run a benchmarking script to validate all this.
- b. This was relatively easy and the most challenging part was figuring out the Linux prerequisites/dependencies for JTR. Once I got to know the JTR Github repo. It had a readme file with all the information.

2. Assignment 2:

- a. In this assignment, we were given a set of 1000 hashes and were supposed to find the unhashed/plain text values of them.
- b. All of the hashes given were in MD5, this was figured out by the indicator "\$1" at the beginning and some googling.
- c. Next, we needed to identify the best / fastest method to crack the hashes. Using JTR and the proper wordlist I was able to crack all 1000 hashes in no time. (Rockyou.txt being the most popular choice on the internet.)

3. Assignment 3:

- a. The task for this assignment was to crack 2000 hashes of 6 different types.
- b. Following the same approach as the previous one, I identified the types based on the identifier of each hash. To automate this task, I wrote a shell script to sort the hashes by type and place them in a different file.
- c. I started to write another shell script to automate the process of hashcat based on the hash type and best-suited method. But eventually gave up because I didn't feel the effort was worth it. And started manually running the process which is much more efficient since we can apply different methods of cracking each time.
- d. I was able to crack 1024/2000 hashes with this.
- e. Two more python scripts were written to process the cracked hashes and remove them from the input file.
- f. Because the Rosettahub instance was getting shut down randomly another script was written to back up the pot file every minute to my blinkenshell community server.

- g. All the scripts for this assignment can be found at:
<https://github.com/romaan7/scalable-computing-ass/tree/master/Scripts>

4. Assignment 4:

- a. This being the final assignment was a bit challenging. The tasks were to crack the password to get into a Dante InfernoBall like JSON structure to get to the next level.
- b. The hashes were of the 4 types PBKDF2, SHA512Crypt, SHA1Crypt- Cisco, Argon2.
- c. After examining the code shared, I wrote a python script for validating all the cracked passwords and if the next level has been unlocked.
- d. Since this was a team task, we divide divided the easiest hashes to crack in terms of time complexity (PBKDF2 and SHA1) among ourselves and each one using his own GCP or AWS instance to start cracking.
- e. There was another script which collated all the hashes from everyone's instance and checked for level if it cracked.
- f. There were very few levels where we had to go for SHA512 and Argon because we didn't hit the K value for Shamir secret sharing.
- g. Github was of great help in keeping the track of all the passwords and distributing/ collating the work of each teammate.
- h. All the scripts and code used in this assignment can be found at below link:
<https://github.com/romaan7/scalable-ass-5>

Module evaluation

The overall structure of the module was good. I feel it covered all the relevant topics concerning scalable computing. The assignments were well designed to give a scene and develop an understanding of scaling huge problems. I developed the habit of reading research paper only after doing the assignments.

Although the hash cracking assignments were designed to explain the scaling problems, I feel that there could have been a better implementation of it. Just because it took a lot of my time running hashcat. People have only learned how to successfully run tools like hashcat or JTR in this assignment. This could be changed to something useful like developing a small but scalable OpenCL program to crack one algorithm, in which the students can learn more scaling.

Also, please switch from RosettaHub to something else like Google Cloud platform. ☹️