Cluster and Cloud Computing
Assignment 2

#TheDeadlySins#2019

Victoria Social Media Analysis

Team 52

Nitish Mathur (Student ID: 954892)
Yash Shinde (Student ID: 920691)
Nakia Silva (Student ID: 796504)
Shreyas Walvekar (Student ID: 961621)
Krishna Srinivas Sundararajan(Student ID: 977863)

Git repository:
https://github.com/knightmatish/CC_assignment_2

Table of Contents

1. Introduction

There has been a massive proliferation in the usage of social media platforms globally over the past decade. The magnitude of active users and the platform utilization that contemporary mainstream social media platforms boast is truly staggering. As of 2019, Facebook reports a venerable 1.2 billion active users per day [1]. On the other hand, Twitter reports a total of 126 million monthly active users that on an average tweet approximately 500 million tweets every day [1][2]. As a result of these astounding numbers, the sheer amount of social media data generated is colossal and is increasing with every passing second. The availability of this data through the APIs that these social media platforms render coupled with the realization that social media data analytics can enable the identification of inconspicuous trends and provide invaluable insights which can in turn enable better strategic decision making, validation of proposed hypotheses and the understanding of the inherent human behavior has propelled research in the pertinent area.

While there has been an enormous amount of research in the area of sentiment analysis e.g. determining if people are happy or sad based on their tweets, the work undertaken on other aspects of the human nature and emotion such as pride, greed, wrath, lust etc., has been far less [3]. In recognition of this the report endeavors to describe the cloud based social media data analytics system that aims to explore how social media data can be harnessed to gain insights about the human emotion and how these insights relate to actual human behavior captured in the form of real-world statistical data. The aforementioned software system was developed as part of a mandated deliverable for the second assignment of COMP90024 as proposed in the assignment specification [3]. The report comprehensively documents the developed system and its core functions. In doing so it provides a high-level overview of the system's architecture while alluding to the various design decisions taken, the rationale that led to it and the various idiosyncrasies that the system manifests. Additionally, the report includes a simple user guide for testing including information regarding the systems deployment, invocation and usage. Furthermore, having used the UniMelb Research Cloud for the deployment of the system, the report addresses its various pros and cons. The report also attempts to explain the kind of social media data analytics that this system performs, the datasets used, the methodology employed, and the results obtained. Finally, the report delves into a comprehensible analysis discussing the reason behind the obtained results and how it relates to the proposed hypothesis relating to the existence of a correlation between the popularity of "Game of Thrones" on social media and the amount of violent crimes recorded.

Various studies done over the years argue the existence of a correlation between the portrayal of violence in mass media and the amount of violence, aggression and the transpiration of germane crimes in the real-world [4][5]. The aforementioned system was developed with the intent to explore this notion further by analyzing social media data to identify the ratio of users that tweet about "Game of Thrones", arguably one of the most popular TV series in today's time that famously portrays numerous graphically violent scenes. These ratios are found for different cities in Australia and are then used to validate the existence of a correlation between the relative popularity of "Game of Thrones" in these cities and the amount of violent crimes in these respective cities. Hence, the developed software systems aims to explore an aspect of wrath in the context of the seven deadly sins [6] by finding the relative popularity of a prominent graphically violent television series in various cities in Australia and its relation to violent crimes in these areas by trying to validate the existence of a trend by using the result of the social media analytics done by the system and comparing to the data related to violent retrieved from AURIN [7] in the pertinent cities.

2. System Description

This system starts the process with live streaming the tweets that are filtered by the coordinates. User Id is determined from the filtered tweet and is stored to a database. The document id ("_id") is set as the User Id to avoid any duplicated entry of the same user in that database. Once the user is successfully entered to the database, the system performs the search operation, in that another program is called which looks up for the timeline for that particular user. Whole timeline of that user is saved to another database. As per our hypothesis, CouchDB performs the data analytics operation. The system uses the Mapreduce functionality of CouchDB to create a view that filters out the tweet by looking for particular keywords and on the search process we identify if the user can be classified as Game of Thrones fan or not. Once the map reduce operation is completed, the web server calls the reduced view which updates a data structure which maintain the record of game of thrones fan in the duration of (2012-2016) and the number of user created in the same duration.
The processed data is then compared with the data extracted from AURIN and we try to find a correlation between the data to prove our hypothesis.

2.1 System Overview

The cloud based social media data analytics software system developed in adherence to the assignment specifications provided [3] as the name suggests performs data analytics on social media data specifically Tweets harvested by the means of the consumption of the Twitter API. After the system harvests the required twitter data it uses it to reckon the relative popularity of “Game of Thrones” on the germane social media platform (Twitter) in various cities in Australia by the means of data analytics performed on the harvested tweets. Having done that the system provides a web-based data visualization capability (a web interface) that facilitates the comparison between these results that are obtained with the real-world data about actual crimes in these cities retrieved from AURIN [7]. The engineered application is a cloud-based distributed system with scripted deployment and configuration capabilities. Some of the other salient features of the software system include:

- The processing of data in a distributed manner using the MapReduce paradigm for data processing over a cluster of nodes to facilitate scalability.
- The replication and subsequent storage of data over multiple nodes to increase data availability and augment a sense of fault tolerance etc.
- Harvesting of twitter data using both the search as well as the streaming

2.2 System Functionality

This system starts the process with live streaming the tweets that are filtered by the coordinates. User Id is determined from the filtered tweet and is stored to a database. The document id (“_id”) is set as the User Id to avoid any duplicated entry of the same user in that database. Once the user is successfully entered to the database, the system performs the search operation, in that another program is called which looks up for the timeline for that particular user. Whole timeline of that user is saved to another database. As per our hypothesis, CouchDB performs the data analytics operation. The system uses the Mapreduce functionality of CouchDB to create a view that filters out the tweet by looking for particular keywords and on the search process we identify if the user can be classified as Game of Thrones fan or not. Once the map reduce operation is completed, the web server calls the reduced view which updates a data structure which maintain the record of game of thrones fan in the duration of (2012-2016) and the number of user created in the same duration.

The processed data is then compared with the data extracted from AURIN and we try to find a correlation between the data to prove our hypothesis.

2.3 System Elucidation

This section attempts to address the various decision taken in relation to the system design and the usage of the technologies while detailing the rationale that led to it. Each of the issues were undertaken by the project team using a methodical decision-making framework leveraging SWOT analysis [8] that took into consideration various different factors including the requirements encompassed within the assignment specification that was provided [3], the skill set of individual project team members, the project schedule dictated by assignment deadline among others.

2.3.1 CentOS:

Each of the virtual machines (cloud instances) that runs part of the developed software system run CentOS as its operating system. While other potential alternatives were explored the reasons that the project team opted to use CentOS as the operating system of choice over the others are as follows:

- The project team members are quite well acquainted with Linux as well as the RHEL (Red Hat Enterprise Linux) architecture having been exposed to it in previous academic and professional endeavours making CentOS a tenable candidate for the job.
- CentOS is considered to be a more stable distribution over some of the other Linux distributions such as Ubuntu as a result of less frequent package update, hence reducing some overheads related to the maintenance of the system.

2.3.2 CouchDB:

CouchDB is an open-source document-based NoSQL database [9]. The system uses CouchDB in the cluster mode to store and manage the data related to the social media data analytics performed as part of the core function of the developed system. Following are the reasons for the selection of CouchDB in the pertinent project:

- Using CouchDB in the cluster mode enables the replication of the data stored within the CouchDB cluster which increases the magnitude of data-availability due the storage on redundant data over multiple nodes within that cluster thereby making the system more fault-tolerant. Therefore, given a CouchDB cluster made up of multiple nodes the malfunction or failure of a single node would not necessarily entail the failure of the entire CouchDB cluster.
- Using CouchDB in the cluster mode enables the distribution of data over multiple nodes on the cluster which in turn provides the opportunity to create more scalable application by exploiting a distributed processing paradigm such as MapReduce [10].
- Data stored in a CouchDB cluster is amalgamated and accessible by each node with the cluster.
- The usage of CouchDB is a mandatory requirement as specified in the assignment specification [3].

2.3.3 Python:

Python scripts are used as part the developed software system for consuming the twitter APIs in order to harvest tweets and for subsequently pushing them into CouchDB. The reasons for the selection python over other languages was because:

- The CouchDB python module is quite easy to use.
- The project team as whole unanimously choose python as their preferred language to code in.

2.3.4 Twitter API:

As mentioned in subsection 2.3.3, the application consumes the twitter APIs so as to retrieve tweets with the intent to process and analyse them. The software solution developed uses both the streaming as well as the search APIs for harvesting data. Following is a high-level abstraction of these two approaches in the context of the developed system:

2.3.4.1 Search API:

For the purposes of the developed software system, the Twitter Search API is consumed with the intent to retrieve user timelines i.e. the 3200 most recent tweets of a specific user irrespective of how old the tweets are.

2.3.4.2 Twitter Stream API

- In addition to the Twitter Search API the Twitter Stream API is used for harvesting tweets as the consumption of the latter enables the capabilities of live streaming the tweets.
- The Geo box location filter is used in this approach for tweet harvesting.

2.3.5 Tweepy:

Tweepy was selected as a means for consuming the Twitter APIs in this application as:

- Tweepy is an easy to use API wrapper to call the twitter API which is rendered as a python library.
- Tweepy provides support for consuming both the streaming as well as the searching Twitter API.

2.3.6. Ansible:

Ansible is a simple and powerful tool to leverage virtualized environment (server virtualization for this scenario) [14]. It is a great automation tool to have ubiquitous configuration across all the servers. It enables:

- Configuration Management

- Application management/deployment
- Easy Orchestration
- Continuous delivery

For our setup, we are using the Push configuration architecture. For this architecture the ansible master server does the following:

1. Spins up four VM’s and setup the basic configuration.
2. Mounts the volume and attaches the required security groups.
3. Install and update all the required packages
4. Configures docker and required containers.
5. Configure CouchDB on 3 instances and Web hosting services on the 4th VM.
6. Ships harvester on 3 instances and analyser on the 4th.
7. Configures the cluster of CouchDB instances.

2.3.7 Docker:

Docker is DevOps tool designed to make it easy to create, deploy, and run applications by using the technology containers [13]. Docker leverages the concept of containerization by allowing easy packaging of an application and use it across any virtual machines. Docker uses a common repositories where the packaged application are stored and can be accessed. We are using docker for improving the flexibility of our application deployment process using containers and VMs. We are using docker for containerising CouchDB.

2.3.8 Handling Duplicate Tweets:

The twitter harvester is designed in such a way that, it retrieves entire timeline of users. Using CouchDB, a dataset is multiplied across instances for concurrent storage. The streamer handles incoming streams and checks if a particular user has already been processed by querying a replicated database. This prevents a single user’s timeline being added to the processing database a second time.

3. System Design And Architecture

The developed cloud-based system exploits a multitude of Virtual Machines for the purposes of performing the following sequence of tasks:

i. Harvesting Tweets using Twitter APIs
ii. Storage of the required social media data on the CouchDB cluster in order to perform data analytics.
iii. Performing the required social media data analytics.
iv. Visualization of the results obtained from the previous step against real-world data obtained from AURIN [7] using a web interface.

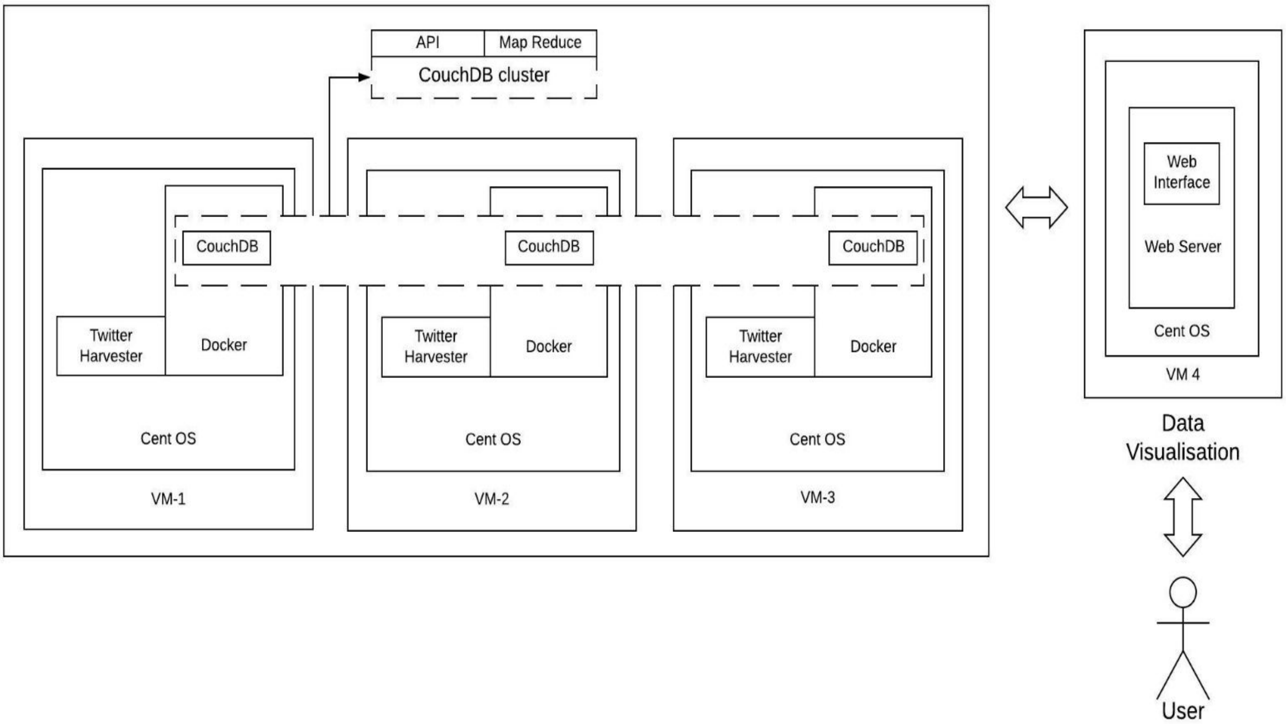

Fig 3.1. System Architecture Diagram

The architecture used to develop the system so as to the carry out the aforementioned tasks efficaciously is depicted in Fig 3.1. Based on the functions that the subsystems deployed on the virtual machines carry out the virtual machines that the systems utilise can be broadly bifurcated into:

3.1 Cluster Node VMs:

These are the Virtual machines that are responsible for performing the first three tasks mentioned above. There exists multiple nodes of this type. These types of nodes are labelled as VM-1, VM-2 and VM-3 in Fig 3.1. Each Virtual Machine of the type Cluster Node runs a docker container that in turn encompasses and runs CouchDB in the cluster mode. CouchDB that runs within the containers that run on each of these VMs together form a single CouchDB Cluster. Additionally, each of these VMs run twitter harvesters that simultaneously run and push the retrieved tweets into the CouchDB cluster in order to perform the required social media data analytics which is carried out on the data stored within the CouchDB cluster using MapReduce. The total number of Cluster node VMs is set to three owing to the constraints imposed on the maximum number of instances (virtual machines) per team that the UniMelb Research Cloud allows to spin up for the purposes of this assignment (a maximum of four instances). That being said it is important to note that the number of Cluster Node

VMs could be easily increased by making minimal changes to the ansible scripts that orchestrate the setup of the system given additional resources. Given the way the system is architected an increase in the number of Cluster Node VMs would entail an increase in the total number of simultaneously running twitter harvester processes as well as an increase in the size of the CouchDB cluster. This would in turn lead to a greater number of tweets being harvested per unit time (due to the increase in the number of twitter harvesters) and would also results in the distribution of the processing associated with the social media data analytics performed on the data contained with the CouchDB cluster over a larger number of nodes (owing to the increase in the size of the cluster and the usage of MapReduce for processing data parallely on the cluster). Thus augmenting a sense of scalability to the system. Given a large set of data (tweets) to be processed, an increase in the computing resources (number of Cluster Node VMs) would improve the system's performance. Furthermore as the data within the cluster is replicated and stored over multiple, it makes this part of the system more fault tolerant. Hence, the failure of a cluster node VM would not necessitate the failure of the entire cluster or the loss of data as a result of the redundant storage of data.

3.2 The Data Visualization VM:

This type of VM is responsible for running the web server and hosting a web interface for visualizing the correlation between the results of the social media data analytics and the real-world statistical data by strategically rendering them in graphical format. Fig 3.2 shows a screenshot of the web interface hosted on the VM of this type. There is no more than one VM of this type. As the VM acts as the interaction point between the user and the core analytical subsystem (the cluster) that stores data that may be construed as being sensitive in nature, it does not form part of the cluster in addition to hosting the web interface of the system as while this approach may increase system performance, it raises some security concerns. While a VM of this time can be considered as being a single point of failure for the system, it should be noted that the failure of this VM does not affect the functions of the cluster but rather just precludes the end user from viewing the results. Additionally, in the case of the failure of this VM, a new Data Visualization VM could be spun up and configured to replace the old faulty VM quite easily using the Ansible script.

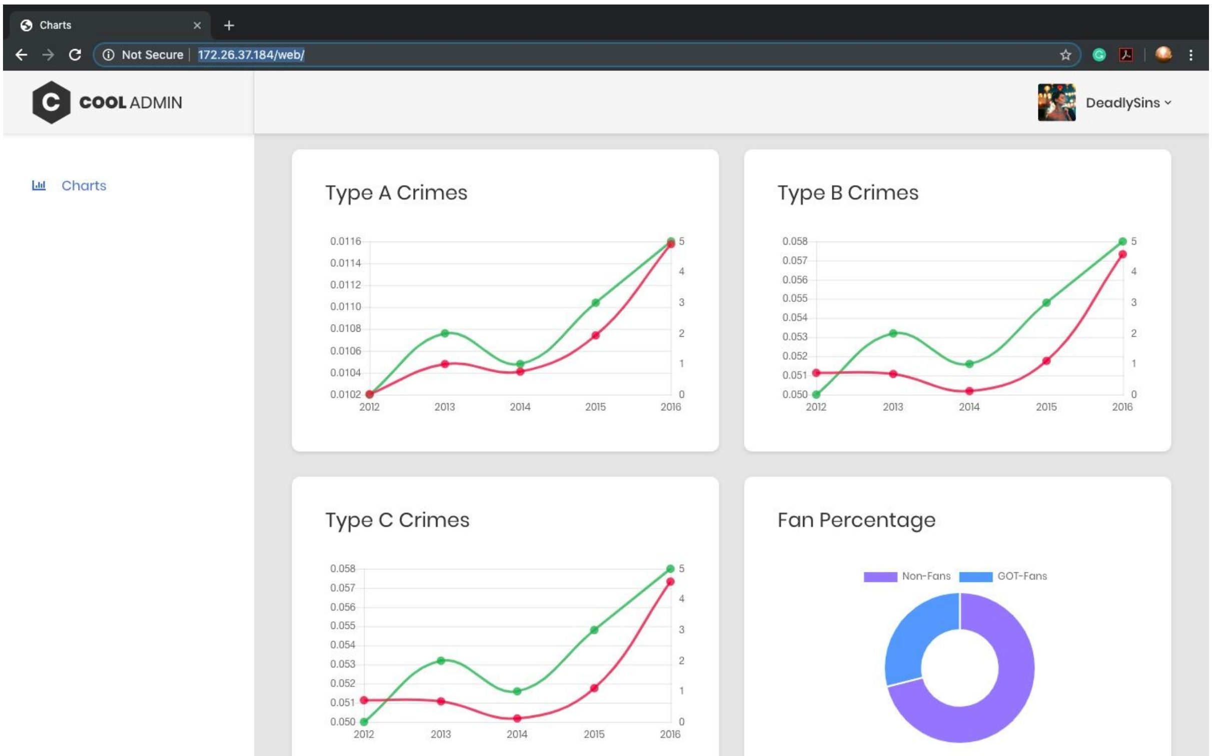

Fig 3.2 The Web Interface for Data Visualization

In summary, the system is made up of four cloud instances (Virtual Machines) as shown in Fig 3.1. Three of these instances are used to form a CouchDB cluster and are responsible for harvesting tweets using the twitter API and subsequently storing then within the CouchDB cluster as documents. Having stored all the required data within the CouchDB cluster, this data is then processed using map reduce on this cluster. The fourth instance that does not form a part of the CouchDB cluster runs a web server that host a web interface that enables the visualisation of results of the analysed data.

4. The UniMelb Research Cloud

Having used the UniMelb Research Cloud for infrastructure provisioning, image creation and deployment. As a result of this exposure, the team was able to discern various different advantages and disadvantages of using the UniMelb Research Cloud as part of this software development endeavour. Following are the various pros and cons of the using the UniMelb Research Cloud in the context of image creation and system deployment:

4.1 Pros of the Unimelb Research Cloud:

a. Dynamic Provisioning of computing resources:

The UniMelb research cloud provides on demand dynamic provisioning of computing infrastructure. The provisioning process in instantaneous and hence images can be launched and instances can be created in a matter of seconds.

b. Don’t have to worry about infrastructure maintenance:
The UniMelb Research cloud allows the usage of infrastructure without having to worry about the maintenance of the underlying hardware.

c. Allows for application scalability:
Applications deployed on UniMelb Research Cloud can be scaled up or down quite easily as new instances can be spun up almost instantaneously and unused instances can be released.

d. Network access:
The UniMelb Research Cloud renders its services over the internet. Hence, one can conveniently launch and access instances using any remote computer that has access to the internet.

e. Easy to interface:
It is relatively easy to interact with and avail to the services that the UniMelb Research cloud renders. For example, one could quite easily create an instance by conveniently interacting with the management console (web interface) or by consuming the APIs that it provides.

f. No need to purchase hardware:
Deploying an application using the UniMelb Research Cloud enables the usage of the pre-existing infrastructure and hence there is no need to purchase hardware .

g. Cost effective (for the purposes of this project):
One of the biggest advantages of using the UniMelb Research for this project is that the cloud based services required to accomplish the pertinent requirements of the assignment specifications [3] could be avail to for free. If the offerings a commercial IaaS were to be used for instead of the UniMelb Research, there would be a monetary cost associated with it which would increase with the magnitude of the services used.

4.2 Cons of the Unimelb Research Cloud:

a. Loss of control:
   Using the services of the Unimelb Research Cloud has certain implication on the amount of control we have on the application. For example, if the Unimelb Research Cloud ceases to exists tomorrow all the instances that are application deployed on would be destroyed and the data contained there within would be lost (unless it is backed up).

b. Slow interface/dashboard:
   The Management console can be really slow and unresponsive at times.

c. d. Auto time synchronization problem:
   Privileged port 123, a requirement for using NTP update is blocked by default. Need to add a temp security group during first launch to enable this update.

d. No pre-configured proxy settings:
   The Unimelb Research cloud doesn't come with pre-configured proxy settings which make it harder to update/install the packages on the fly.

   e. Group key pair management potentially missing:
   The key pairs available to users during instance creation are limited to the ones in the user's own account. Even when working on a project in a group, creating new keys in the project's workspace doesn't work as the keys are just created in the user's workspace. When a new user wants to launch another instance in the same project, the previously shared key file needs to be imported again into the new user's workspace, potentially causing naming conflicts even among authenticated users.

5. Ansible User guide

The ansible contains multiple files which are required to run the script. These files are categorised as required files, inventory files and scripts.

The Required files to run the ansible scripts are:

5.1. key pair file (sins101.pem)
5.2. openrc.sh file (unimelb-comp90024-group-52-openrc.sh)
5.3. password for openrc.sh file (Password.txt)

The Inventory file- host.ini file contains 3 groups:

5.1. [all_servers_common]:

Contains the IP addresses of the VM's which requires all common configuration for our cloud-based application.

5.2. [couch_cluster_node]:

Contains configuration of our couchDB cluster node.

5.3. [web_server]:

Contains configuration of our data visualising node.

The invoking scripts are as follows:

5.1. nectar.yaml:

This script contains 4 roles for basic setup

5.1.a. openstack-image:

It retrieves all available openstack images and selects the desired image for our application (that is centOS).

5.1.b. openstack-volume:

It creates extra non-volatile storage to the instance.

5.1.c. openstack-security-group:

It sets up the network configuration for our application.

5.1.d. openstack-instance:

It creates the instance by installing the defined or provided image and attaches the created volume and security groups.

5.2. mount_volume_config_proxy:

This playbook ssh into the created instance and has two roles.

5.2.a. openstack-mount-volume:

This creates a directory /mnt/storage and mount volume /dev/vdb with file system type ext4. This also updates the UTC time.

5.2.b. openstack-common:

This sets the proxies http, htpps, ftp, no_proxy into the file /etc/environment.

5.3. package-configure:

This file majorly updates all the packages and sets up the couchdb cluster node and webserver.

5.3.a. openstack-package-configure:

This update "yum" abd installs the packages: yun_utils, development tools, python, python 36-pip, jq, docker, docker-composer, vim, lvm2.

It downloads the node.js package and configures npm. It also installs python related packages like ijson, couchdb and tweepy. It configures the docker setup and enables the required docker proxies.

5.3.b. openstack-start-harvester:

This creates our directory mnt/storage/project for twitter harvester. It clones the git repository where all files are stored and maintained. It also creates the couchdb cluster by using the file create_cluster.sh.

5.3.c. openstack-configure-cluster:

It creates 3 docker containers for couchdb and groups them into a cluster.

The command used to execute the above 3 script is ". ./unimelb-comp90024-group-52-openrc.sh; ansible-playbook --ask-become-pass <file_name>.yaml -i hosts.ini". This command is executed 3 times with the appropriate file name in the <file_name> argument. The order of execution is shown in the below table.

Sequence	<file_name>
1	nectar
2	mount_volume_config_proxy
3	package_configure

Table 5.1

6. Data Analytics

6.1 Proposed Hypothesis:

We believe that there is a positive correlation between the number of violent crimes and number of people exposed to graphic content on television. As mentioned before, one of the sources of said graphic content is identified as a popular TV show, Game of Thrones. The thing that can come closest to the actual number of violent crimes in an area is a count of reported crimes of a certain kind of the area. The kind of data made available to us has crime numbers segregated into various categories (described in the next point). Among these categories, it is our hypothesis that the ratio of twitter users exposed to 'Game of Thrones' to total twitter users will have the highest correlation with the category of that describes violent crimes to the size of the population.

6.2 Dataset retrieved using AURIN:

AURIN (Australian Urban Research Infrastructure Network) is a cooperative national project collaborating with researches and data providers across government, academic and other sectors across Australia to provide data sets, protocols, and tools to researchers across Australia for understanding patterns in urban development [11]. A dataset related to criminal offences recorded in the selected cities was retrieved using AURIN so as to validate the proposed hypothesis by comparing this data with the results of the social media data analytics performed by the developed system to find a correlation between the two. The retrieved dataset classifies these offences into six divisions developed by Crime statistics Agency (CSA) [12]. Table 5.2 outlines the offenses in each division. As it is quite evident from Table 5.2 the offences classified as Type A and Type C have a relatively greater association to violence and aggression. In addition to the offences specific to the selected cities, the retrieved dataset also provides the total population of the associated region.

Crime Statistics Agency offense division	Offense
A	Homicide Assault Sexual offense Abduction
B	Bribery Theft Property damage
C	Drug related offences Public nuisance offences
D	Illegal possession of weapons and explosives
E	Justice procedure offences
F	Driving Transport other

Table 5.2 [12]

6.3 Methodology:

The project involves two sources of data, AURIN and Twitter. These are processed in the following way :-

- Twitter Data
    - Stream twitter until you encounter a new user.
    - Retrieve the user's timeline.

        - Process individual tweets in timeline to check if user knows about Game of Thrones, if he does, enter the year of the tweet along with the year of creation of the user's profile into the results table.
        - Follow process for any historical twitter data by retrieving the user's timelines.
- AURIN Data
    - Retrieve the data from AURIN
    - Since the dataset in question returns a city wise crime report of the State of Victoria, aggregate the results of the cities that form a part of Melbourne region.
    - Store this aggregate result to use for creation of graphs.
- Analysis
    - Get two year wise counts of :
        - Users that existed in the specified years.
        - Users that knew about Game of Thrones in specified years.
    - This means that if a user profile was created in year 2010, all the succeeding year (including) counters will be incremented, [2010, 2016].
    - Similarly, once a user becomes aware about GoT, say, in 2014, GoT counters of years 2014, 2015 and 2016 will be incremented.
    - We now have two series of data points. Plot these on a graph to see which of the type crimes (from AURIN) matches best.

6.4 Results

Factors affecting results :

1. Game of thrones search parameters, they are potentially hundreds of keywords that a user can post. It is not easy to implement all of these for identification.
2. False positives, although can be minimised, there may be people who might tweet 'red wedding' without knowing about the show.

7. Conclusion

In conclusion the report documents the cloud based social media data analytics system developed in adherence to the assignment specification provided [3]. In doing so we proposed a hypothesis related to the existence of a correlation between the popularity of game of thrones in various cities of Australia and the crimes recorded within this cities. The system facilitates this by performing social media analytics and comparing these two real world statistical data. We were able to do this by the means of systematic data analysis documented within this report.

8. References

[1] H. Shaban, “Twitter reveals its daily active user numbers for the first time,” The Washington Post, 07-Feb-2019. [Online]. Available: https://www.washingtonpost.com/technology/2019/02/07/twitter-reveals-its-daily-active-user-numbers-first-time/?noredirect=on&utm_term=.9758316ced77. [Accessed: 01-May-2019].

[2] “Twitter Usage Statistics,” Twitter Usage Statistics - Internet Live Stats. [Online]. Available: https://www.internetlivestats.com/twitter-statistics/#trend. [Accessed: 01-May-2019].

[3] R. Sinnott, “Cluster and Cloud Computing Assignment 2 – #DeadlySins#2019,” Retrived from https://app.lm s.unimelb.edu.au/bbcswebdav/pid17408844-dt-contentrid-59591441_2/courses/COMP90024_2019_SM1/Assi gnment1%2811%29.pdf

[4] Anderson, Craig A.; Berkowitz, Leonard; Donnerstein, Edward; Huesmann, L. Rowell; Johnson, James D.; Linz, Daniel; Malamuth, Neil M.; Wartella, Ellen (2003). "The influence of media violence on youth". Psychological Science in the Public Interest. 4 (3): 81–110. doi:10.1111/j.1529-1006.2003.pspi_1433.x. PMID 26151870

[5] Media Violence Commission, International Society for Research on Aggression (Isra) (2012). "Report of the Media Violence Commission". Aggressive Behavior. 38 (5): 335–41. doi:10.1002/ab.21443. PMID 22886500.

[6] Aquinas, Thomas (2013). Summa Theologica (All Complete & Unabridged 3 Parts + Supplement & Appendix + interactive links and annotations). ISBN 9788074842924.

[7] Aurin.org.au. (2019). AURIN. Australian Urban Research Infrastructure Network – Australia's Urban Intelligence Network. [online] Available at: https://aurin.org.au/ [Accessed 1 May 2019].

[8] Dess, Gregory (2018). Strategic Management. United States: McGraw-Hill. p. 73. ISBN 9781259927621.

[9] Apache Software Foundation. "Apache CouchDB". Retrieved 1 May 2019.

[10] Jeffrey Dean and Sanjay Ghemawat. 2004. MapReduce: simplified data processing on large clusters. In Proceedings of the 6th conference on Symposium on

Opearting Systems Design & Implementation - Volume 6 (OSDI'04), Vol. 6. USENIX Association, Berkeley, CA, USA, 10-10.

[11] Sinnott, R. O., Galang, G., Tomko, M., & Stimson, R. (2011, December). Towards an e-infrastructure for urban research across Australia. In 2011 IEEE Seventh International Conference on eScience (pp. 295-302). IEEE.

[12] Crime Statistics Agency Victoria. (2019). Retrieved from https://www.crimestatistics.vic.gov.au

[13] O'Gara, Maureen (2013). "Ben Golub, Who Sold Gluster to Red Hat, Now Running dotCloud". SYS-CON Media.

[14] Staff writer. "Overview - How Ansible Works". ansible.com. Red Hat, Inc. p. 1.