Web Harvesting, Scrapping and Map Reduce
Common Crawl Report
Dr. Bina Ramamurthy

Komas Aryal

2019

Contents

1 Introduction

1.1 Common Crawl

Common Crawl is a nonprofit 501 organization that crawls the web and freely provides its archives and datasets to the public. Common Crawl's web archive consists of petabytes of data collected since 2011. It completes crawls generally every month. Common Crawl was founded by Gil Elbaz. There are different ways to use the data from the Common Crawl such as using their DATA format, WARC format, WAT response format, or even the WET Response Format.

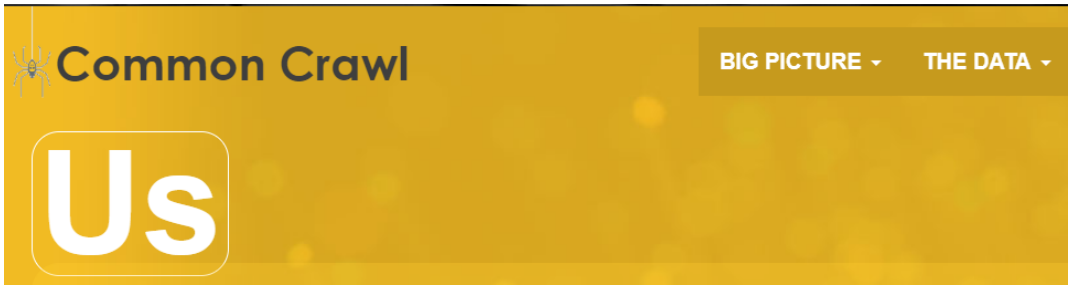

Fig 1: Common Crawl Home Page

1.2 Amazon Web Services

Amazon Web Services (AWS) is a subsidiary of Amazon that provides on-demand cloud computing platforms to individuals, companies and governments, on a metered pay-as-you-go basis. In aggregate, these cloud computing web services provide a set of primitive, abstract technical infrastructure and distributed computing building blocks and tools. One of these services is Amazon Elastic Compute Cloud, which allows users to have at their disposal a virtual cluster of computers, available all the time, through the Internet. AWS's version of virtual computers emulate most of the attributes of a real computer including hardware (CPU(s) GPU(s) for processing, local/RAM memory, hard-disk/SSD storage); a choice of operating systems; networking; and pre-loaded application software such as web servers, databases, CRM, etc.

Fig 2: AWS Web services

1.2.1 S3 Bucket

An Amazon S3 bucket is a public cloud storage resource available in Amazon Web Services' (AWS) Simple Storage Service (S3), an object storage offering. Amazon S3 buckets, which are similar to file folders, store objects, which consist of data and its descriptive metadata. For this project, one can store the mapper and reducer of their choice in the bucket for the creation of the cluster.

Fig 3: S3 Bucket

1.2.2 Amazon Elastic Map Reduce

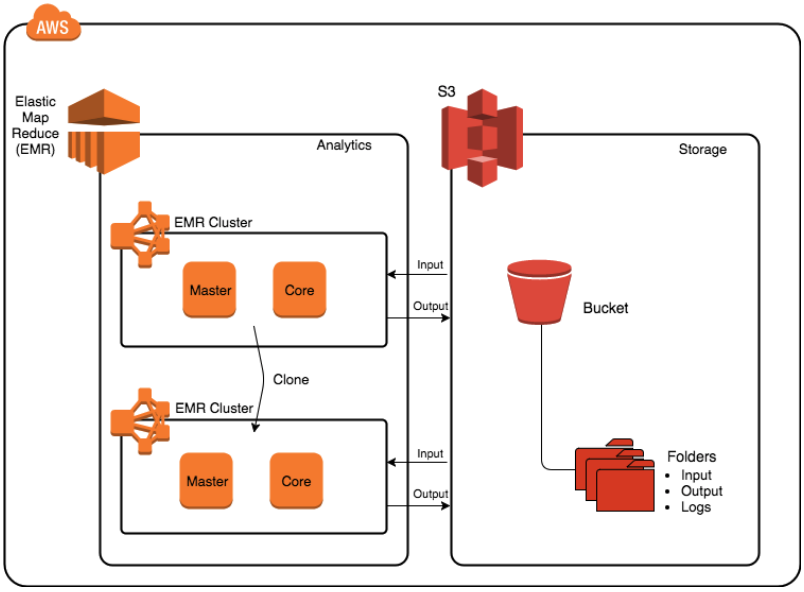

Fig 4: Elastic Map Reduce

Map Reduce and Hadoop Structure Amazon Web Services Elastic Map Reduce and S3 was used for the cluster infrastructure. Hadoop is a framework for running applications on large cluster built of commodity hardware. It frameworks transparently provides applications both reliability and data motion and implements a computational paradigm named Map/Reduce, where the application is divided into many small fragments of work, each of which may be executed or re-executed on any node in the cluster. For more information visit: https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-what-is-emr.html

2 Purpose

The purpose of the project was to use the idea of web-scrapping in one of the rapidly growing big-data set i.e. Common Crawl. The idea is to utilize the AWS service's EMR and use mapper and reducer files created in Python and input the comma-separated values to generate the required information. Alternatively, we can use the beautiful soup library in python to scan words in the given API for keywords.

3 Safety

1. **If you use the AWS services makes sure to understand the encryption keys concept.**
   - Elastic map reduce may require the keys to create the cluster structure.
   - AWS key is an example of private key, public key concept. Dr.Bina's Distributed System course has a separate chapter dedicated to this concept where she goes over this valuable concept.
   - https://aws.amazon.com/kms/
2. **University at Buffalo offeres their students credit code worth $ 100.00 of credit for AWS services.**
   - It is valid for a year once created, so make sure to take full advantage of it instead of using money from your pocket.
   - Further instructions on signing up for the AWS services with the credit code can be found on the university website.
   - http://www.buffalo.edu/ubit/service-guides/teaching-technology/aws.html
3. **Make sure to terminate clusters when using the EMR service when creating the cluster infrastructure.**
   - By enabling that option, the cluster is terminated if any errors is encountered.
   - If it is not enabled AWS charges for the broken clusters.

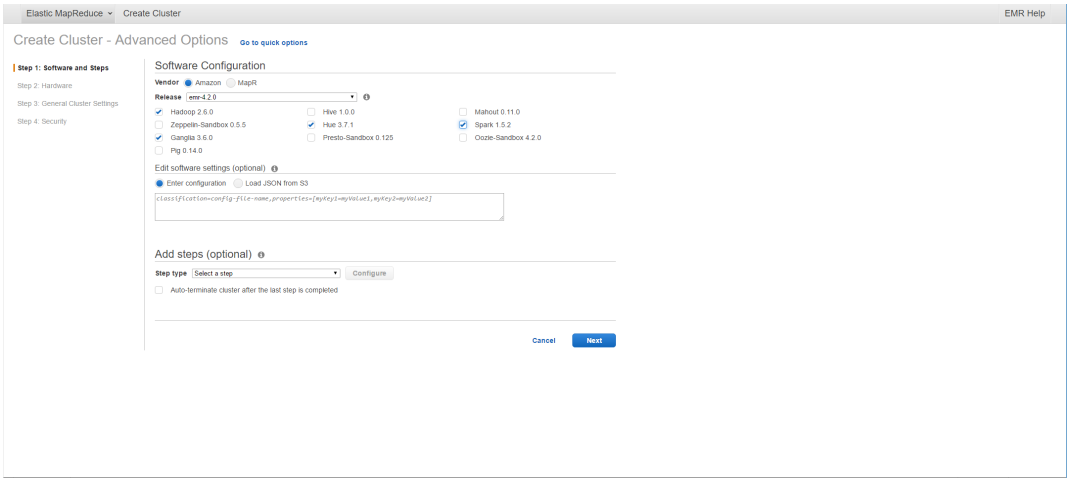

Fig 5: Cluster Clause

4. **Be cautious of when the cluster goes wrong.**
   - The cluster may take up to an hour to two depending on the mapper and reducer.
   - If anything goes wrong, the errors can be traced from the "Steps" option on your cluster options as shown in the following figure.

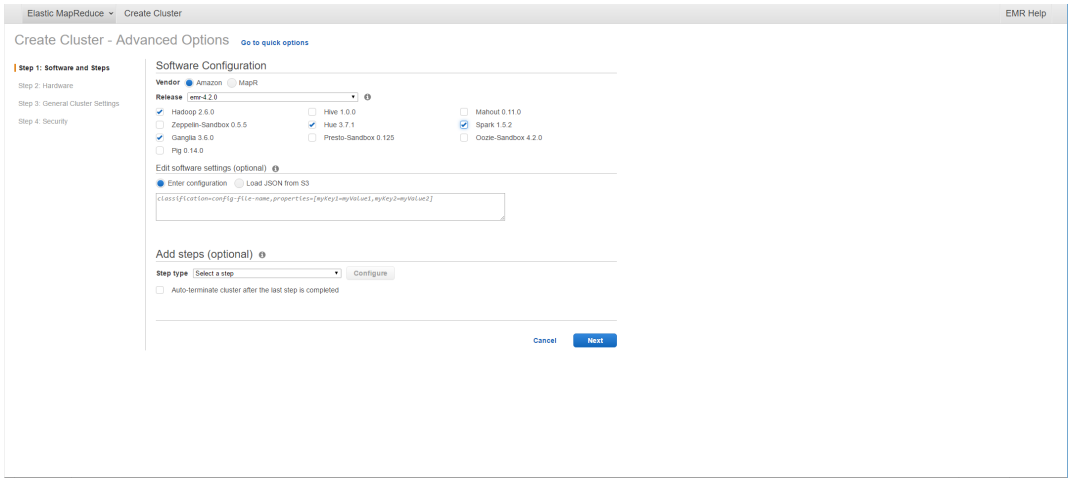

Fig 6: Tracing errors

5. **Be cautious of the bootstrap actions in AWS while creating the clusters**
   - These basically installs the library that are needed for the cluster formation.
   - If not enabled, the cluster will terminate with errors and it would difficult to trace them.

4 Usage Instructions

1. **Create your AWS service account.**
   - Make sure to use the student credit code. Now let us dive into the steps.
   - There are several ways to create your cluster in AWS services.
     You could use a streaming program using mapper and reducer as a python files and give your arguments in it. You could also use Apache Hive or pig, if it makes it easier for you. But the most popular option is to use the custom Java ARchive or commonly known as Jar file for the process. If the credit code is not activated, you can only launch a 10-node EMR cluster with applications such as Apache Spark, and Apache Hive, for as little as $0.15 per hour. For this report, the streaming program will be used as an example.

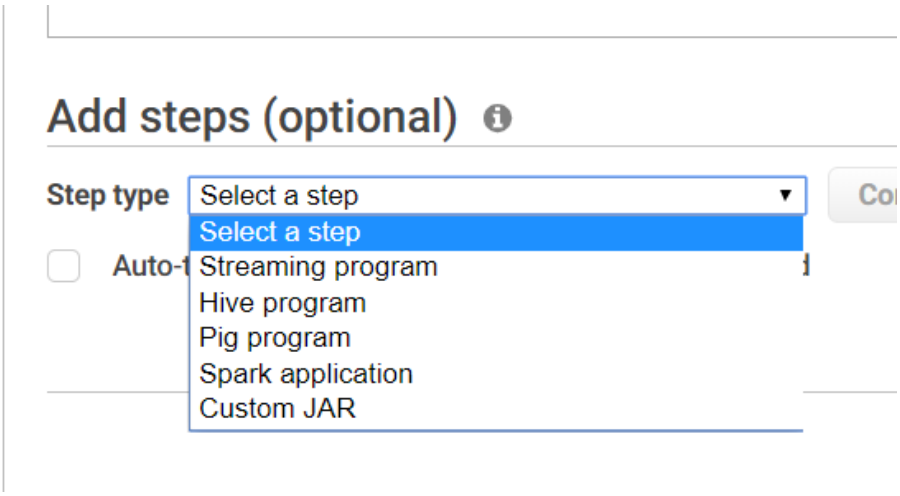

Fig 7: Steps and Options

2. **Search for SW2 bucket in the search bar.**
   - Create a new bucket and name is appropriately, the naming convention is really important in this process.
   - Upload the mapper and reducer in the bucket with proper name convention.

3. **After successful upload and set you are now ready for the next step.**
   - Search EMR in the search bar and you will be directed to a new page.
   - Before clicking on create cluster make sure you understand the use and the concept of cluster, which can be found on the AWS EMR page linked above.
   - When creating a cluster go to advanced options because, we are going to be using streaming program for this project.
   - For the software configuration, let it all be on default and release is usually at emr-5.24.0
   - Do not enable multi-master support.
   - As for the AWS Glue Data Catalog settings, you can enable hive table metadata, if you are using hive program
   - The software setting should be on default, unless you are planning to load a JSON file from the S3.

4. **Streaming program configuration**
   - On the bottom of the screen, on the step type option choose streaming program,
   - A dialog box will appear with name, reducer, mapper and other options.
   - For the name, use a proper naming convention for the cluster.
   - For the mapper options, use the S3 location of the map function or the name of the Hadoop streaming command to run.
   - Similarly, for the reducer options, do the same use the S3 location of the map function or the name of the Hadoop streaming command to run.
   - Upload the mapper and reducer in the same bucket so that it easier to work with.
   - For the Input S3 location, you could use the wet.gz link from the common crawl or you can upload the comma separated file(csv) to find the desired results.
   - For the output makes sure you chose a new folder that does not exists because this is where you will obtain the result.

5. **EC2 key pair**
   - It may asks to create the key pair before proceeding if it was your first time creating a cluster.
   - Make sure to read about the key configuration before going to the next step.
   - Create cluster and wait the appropriate time to get your results. If anything goes wrong, the cluster will terminate with errors and the errors can be traced from the steps options as mentioned above.
   - Your s3 output has the desired result.

6. **Once you are ready with the EC2 key press the create cluster.**
   - The process that takes some time depending on the output that is required.
   - The output can be found on the SW3 output bucket that was assigned on the output.
   - The success bucket should look something like the following:

Fig 8: Example of successful cluster

5 Troubleshooting

If you are encountering issues with the common crawl or AWS web services try the following solutions.

- **Getting to know Common crawl.**
  - Make sure you know what common crawl is and how it works for this project to even begin. For more information on common crawl visit http://commoncrawl.org/about/
- **Types of file to use extract from common crawl.**
  - Use "index" files if you are new to the common crawl. This can act as a the building block.
  - For more information on properly using the data contact http://commoncrawl.org/connect/contact-us/
- **Problems with creating AWS account with credit code.**
  - Be careful when creating the account with the credit code, if you run into any problem with the code try contacting the UBIT team. cse-consult@buffalo.edu.
- **AWS and SW2 related problems.**
  - Make sure you are logged in too the correct account.
  - If UBIT can not fix the problem contact the AWS team.
- **Most importantly, clusters related issues.**
  - **Gather data about the Issue.** The first step in troubleshooting a cluster is to gather information about what went wrong and the current status and configuration of the cluster. This information will be used in the following steps to confirm or rule out possible causes of the issue.
  - **Check the Environment.** Secondly,amazon EMR operates as part of an ecosystem of web services and open-source software. Things that affect those dependencies can impact the performance of Amazon EMR.
  - **Look at the Last State Change** The last state change provides information about what occurred the last time the cluster changed state. This often has information that can tell you what went wrong as a cluster changes state to FAILED. For example, if you launch a streaming cluster and specify an output location that already exists in Amazon S3, the cluster will fail with a last state change of "Streaming output directory already exists".You can locate the last state change value from the console by viewing the details pane for the cluster, from the CLI using the list-steps or describe-cluster arguments, or from the API using the DescribeCluster and ListSteps actions.
  - **Examine the Log Files** The next step is to examine the log files in order to locate an error code or other indication of the issue that your cluster experienced. For information on the log files available, where to find them, and how to view them, see View Log Files.

- **Test the Cluster Step by Step.** A useful technique when you are trying to track down the source of an error is to restart the cluster and submit the steps to it one by one. This lets you check the results of each step before processing the next one, and gives you the opportunity to correct and re-run a step that has failed. This also has the advantage that you only load your input data once.

  To test a cluster step by step:

  1. Launch a new cluster, with both keep alive and termination protection enabled. Keep alive keeps the cluster running after it has processed all of its pending steps. Termination protection prevents a cluster from shutting down in the event of an error.
  2. Submit a step to the cluster.
  3. When the step completes processing, check for errors in the step log files. For more information, go to your "Examine the Log Files." The fastest way to locate these log files is by connecting to the master node and viewing the log files there. The step log files do not appear until the step runs for some time, finishes, or fails.
  4. If the step succeeded without error, run the next step. If there were errors, investigate the error in the log files. If it was an error in your code, make the correction and re-run the step. Continue until all steps run without error.
  5. When you are done debugging the cluster, and want to terminate it, you will have to manually terminate it. This is necessary because the cluster was launched with termination protection enabled.

6 Further Expansion

The same idea can be used to trace data in databases like New York times data, twitter data and even the Google big datas. You can use the word count option to sort the data and count the word you desire to search. Further examples of using common crawls can be found in: http://commoncrawl.org/the-data/examples/

7 References

Aws General , docs.aws.amazon.com/general/latest/gr/aws-general.pdf.
Common Crawl, commoncrawl.org/.