

5-2017

Web Log Data Analysis: Converting Unstructured Web Log Data into Structured Data Using Apache Pig

Neeta Niraula

Saint Cloud State University, nine1401@stcloudstate.edu

Follow this and additional works at: https://repository.stcloudstate.edu/csit_etds



Part of the [Computer Sciences Commons](#)

Recommended Citation

Niraula, Neeta, "Web Log Data Analysis: Converting Unstructured Web Log Data into Structured Data Using Apache Pig" (2017).
Culminating Projects in Computer Science and Information Technology. 19.
https://repository.stcloudstate.edu/csit_etds/19

This Starred Paper is brought to you for free and open access by the Department of Computer Science and Information Technology at theRepository at St. Cloud State. It has been accepted for inclusion in Culminating Projects in Computer Science and Information Technology by an authorized administrator of theRepository at St. Cloud State. For more information, please contact rswexelbaum@stcloudstate.edu.

**Web Log Data Analysis: Converting Unstructured Web Log Data into
Structured Data Using Apache Pig**

by

Neeta Niraula

A Starred Paper

Submitted to the Graduate Faculty of

St. Cloud State University

in Partial Fulfillment of the Requirements

for the Degree

Master of Science

in Computer Science

June, 2017

Starred Paper Committee:
Jie H. Meichsner, Chairperson
Omar Al-Azzam
Dennis C. Guster

Abstract

Data extraction and analysis have recently received significant attention due to the evolution of social media and large volume of data available in an unstructured form. Hadoop and MapReduce have been continuously implementing and analyzing large amount of data. In this paper Apache Pig, which is one of the high-level platform for analyzing large volume of data and runs on the top of Hadoop is used to analyze unstructured log files and extract information. In this paper, weblog server files are used to analyze and extract meaningful information in an unstructured form to a structured form in Apache Pig framework

The main purpose of this paper is to extract, transform and load unstructured data in an Apache Pig framework and analyze the data and its performance on local mode as well as MapReduce mode. This paper further explains in brief about the different steps required to analyze unstructured web server log files in Apache Pig. This paper also compares the efficiency when a large volume of data is processed on MapReduce mode and local mode.

Acknowledgement

I would like to express my sincere gratitude to Dr. Dennis C. Guster, Professor, Department of Information Systems for allowing me to undertake this work.

I am grateful to my advisor and supervisor, Professor Dr. Jie H. Meichsner, Department of Computer Science Information and Technology, for her continuous guidance, advice effort, and invertible suggestion throughout the research.

I am also grateful to my supervisor Dr. Omar Al-Azzam, Professor of Computer Science and Information Technology, for providing me the logistic support and his valuable suggestion to carry out my research successfully.

I would also like to thank lab consultants of the Department of Information Systems for helping to carry out my research.

I would also like to thank my friends of Computer Science for their help throughout the study.

Lastly, I would like to express my sincere appreciation to my family, especially my husband, for encouraging and supporting me throughout the study.

Table of Contents

	Page
List of Tables	5
List of Figures	6
Chapter	
1. Introduction	7
Structured Data	7
Unstructured Data	8
Apache Pig	9
Research Questions	10
Scope of Study	11
2. Literature Review	12
3. Methodology and Results	18
Pig Architecture	19
Pig Data Types	21
General Workflow of Apache Pig	23
Case Study: Access Log Report Analytics	25
4. Conclusion and Future Work	36
References	39
Appendix	41

List of Tables

Table	Page
1. Example of Pig Data Model	21
2. Performance Evaluation on Local Mode and MapReduce Mode	34

List of Figures

Figures	Page
1. Pig architecture	20
2. Workflow architecture of Apache Pig	24
3. Web server log files	26
4. Dump result to generate output	30
5. Pig output log	31
6. Output sample	32
7. Processing time of Pig Script on local	34
8. Processing time of Pig Script on MapReduce Mode	34

Chapter 1: Introduction

In fact, data is always crucial for business professionals within an enterprise to study business details, and the rise of big data has now further facilitated many other areas to explore among big business enterprises. Among these areas, one of the major area involves analysis and organization of different types of data collected in the business enterprises. Data is mainly distinguished into two different types: unstructured and structured data in an organization. Despite the fact, these data types are available in different formats and are likely to be managed differently, the role of each data type is potentially significant for an organization to keep records and make impactful business decisions.

Data is growing at the rate of 50% per year and is usually represented in five-dimensional models: volume, velocity, variety, veracity, and value, represented as 5V. Data analyst and business stakeholders rely on these data types to produce some significant results on collecting, archiving and discovering data to some useful end results.

Structured Data

Structured data refers to those data types which is available in a highly organized form. They are sometimes called traditional data and is simple to enter, store, query, and analyze. Unstructured data can be easily represented in the form of particular tables or schemas. Before the era of big data, structured data was what enterprises used for making business decisions. Structured data is defined in terms of schema and tables and is used to analyze using structured query language or excel spreadsheets to perform queries with relational database.

Structured data analysis tools and techniques is developed with the improvement of data processing by computers, lowered storage cost and new format of data. However, many data-driven companies throughout the world are nowadays beginning to take emerging data sources seriously and both structured and unstructured data are consulted, queried, assimilated and improved to make potential business decisions.

Unstructured Data

Nowadays, unstructured data is growing continually due to the increase in data storage platforms and the infinite number of complex data sources such as social media platforms, mobile application, health data, weblogs, emails, etc. Most of the business interactions now, in fact, is unstructured in nature. The fundamental challenge on managing unstructured data is the diversity in data. Unlike structured data, unstructured data need to work with some specialized tool and be structured in order to analyze.

Qazi and Sher (2016) states, “The world creates 2.5 Quintillion Bytes of data per day from unstructured data sources like sensors, social media posts, and digital photos”. The unstructured data doubles every three months and if left unmanaged the sheer volume of unstructured data will be big (*Insights on governance, risk and compliance*, 2010). In the world of Big data, the accelerating growth of data sources, volumes and configurations are mainly driven due to unstructured data. Therefore, organizations are now little more aware of the volume, composition, risk and business value of their unstructured data. Data warehouse enterprises, like IBM, now have started using big data tools and techniques like, Hadoop, Map-Reduce, etc. for analyzing all data types, which were previously only used in very specific areas to analyze the opportunities inside unstructured data.

Therefore, given these challenges for managing unstructured data, it seems reasonable to accept the challenge and investigate and analyze unstructured data to extract some useful information from it using one of the big data analytics tool, Apache Pig, which is currently trending throughout the global market.

Apache Pig

Apache Pig, an open source high-level data flow system, is an abstraction layer on top of Map-Reduce and performs effectively on any data size, type or location. This methodology is popular and implemented by Yahoo since 2006 to have an ad-hoc way of creating and executing Map-Reduce jobs on very large data sets. It appears that Pig challenges to be better than RDBMS, DBMS based on performance, storage, and transaction level fault tolerance (Minsker, 2015). Although there are other different big data analytical techniques that are used for data analysis, Apache Pig has proved itself to be most efficient for analyzing unstructured data.

Pig uses ETL process for data warehousing. ETL, which stands for “extract, transform and load” is the set of functions combined in one tool and is used to extract large amount of data from numerous database. ETL process involves loading real-time data without interrupting data processing or conducting analysis that supports decision-making process. A common use case of ETL includes converting CSV files to format readable by relational database where user’s input CSV-like data files and import it into a database.

Another advantage of Pig is that it can easily work on raw data. Unlike other big data analytics tools, Pig is the most efficient tool that can work on any sort of unstructured, semi-structured, and structured data. Pig can easily load and process data using some user-defined

functions that can be translated into a series of Map-Reduce jobs to run on an Apache Hadoop Cluster.

Although, the concept of data mining and processing has been outraged for years, extracting and analyzing raw data is always a question of challenge among data analytics. The performance, time, complexity, and the accuracy of data is vital and is needed for every business enterprise to move forward. Therefore, further research is needed that will elaborate other different findings to analyze unstructured data and extract some useful information to accelerate business plan strategies from the extracted data.

Research Questions

In this paper, the data in various unstructured forms are analyzed, examined, and converted into structured form using Apache Pig. Specifically, the following questions will be researched:

1. How to extract and transform unstructured data into a structured form using Apache Pig analysis tool?
2. How does Apache Pig, works on the top of MapReduce?
3. How to manipulate unstructured data to extract various information using Pig processing system?
4. How meaningful information can be generated from the unstructured data and how does it helps to make useful business decisions?

In the course of answering these questions, some unstructured weblog files is being used as an input and generated in a structured format using Pig processing system. The unstructured log files are extracted and stored in Hadoop Distributed File System, HDFS.

Scope of Study

To keep study compatible and functional, following parameters are used in this research:

Server Operating System: The server operating system selected for this project is Ubuntu 16.0.1. It is selected because several features of Apache Pig and Hadoop are upgraded in this version of Ubuntu.

Processing System: The processing system selected for this research is, Apache Pig. It is selected because Apache Pig supports unstructured data more efficiently than any other big data analytics tools.

Database Management System: The database management system selected for this research is Hadoop Distributed File System, HDFS. It is designed to store very large data sets reliably. The main reason for using HDFS in this research is that HDFS allows us to store filesystem metadata and application data separately within the same system.

Type and Size of Files: Unstructured data is used as input. More specifically, this research will be focused on using web server log files varying from few megabytes to gigabytes and generating results in a structured format as output.

These limitations made the study manageable in scope and will hopefully make it easier for the reader to evaluate and use the results.

Chapter 2: Literature Review

The review of literature can be broken into four parts. First, the advantage of Hadoop data analytics tool of extracting information from various data types is discussed. Second, importance of unstructured data for making business decisions is described. Third, various Hadoop analytics tool is discussed based on their functionality. Last, how Hadoop analytics tools has been successful in getting all the required information throughout the distributed data is discussed.

Minsker (2015) stated that the database's greatest power lies in the ability to process and store data that was not possible to be analyzed together due to its volume and unstructured form. She further stated that Hadoop has opened the door to analysis sentiments of society on social media providing them with the path of understanding the positively and negatively charged communications that take place throughout the social media. Lai, Chen, Wu, and Obaidat (2013) stated that Hadoop can gain response at a higher speed on implementing parallel processing over many nodes. When one node fails to work, Hadoop can obtain backup in no time and can recover data for any type of format, also known as fault tolerance. Francis and Kurian (2015) stated that Hadoop implements Map-Reduce framework which helps to retrieve and process unstructured data and map into rows and columns to a form a structured database. Map-Reduce framework can pair data as a key value pair and can be easily implemented using Apache Pig or Hive. Hive operates on the server side whereas Pig operates on the client side for data processing.

According to Quer and Sher (2016), big data is an opportunity that not only allows to deliver competitive advantages per data solutions but has also built an ecosystem throughout

the globe to improve technological power. According to the article (“Insights on governance, risk, and compliance”, 2010), the conventional relational database cannot handle unstructured data and hence framework like Hadoop has been introduced to process unstructured as well as structured data in a high-speed platform and perform a more comprehensive analysis on big data using distributed and parallel processing systems. Holzinger and Pasi (2013) stated that about 80% of data within organizations are unstructured and unfit for traditional processing. Therefore, using big data will enable the processing of unstructured data and increase system intelligence that can result in different opportunities to study business data such as, improving sales, increasing understanding on customer needs, supporting marketing initiatives and fraud monitoring. “Organisations all need to realize that every data has value”, states Al Almond, head of U.S. privacy and social media compliance at TD Bank.

Elizabeth (2013) stated that the analysis of unstructured data in business helps to discover current topics about the products from customer opinion. She also stated that the generated data in any form can be useful to find patterns in reports that may seem to predict business related activities. Business professionals can further discover and understand previously unknown issues and concerns from public feedback on different social media, social forums, etc. Herzig (2011) stated that both unstructured and structured data types are complimentary for business data and hence hybrid queries need to be implemented in order to improve data analysis. Beach and Schiefelbein (2013) stated that monitoring log files, text messages, and online customer product reviews can be the most effective medium for an organization to identify hidden risk before they emerge as full-blown crisis. They further justified that checking all types of data can provide deeper understanding of ongoing business

activities, ensure compliance with certain regulations and monitor employees' engagement and retention. According to hadoop.apache.org, big companies like IBM, Amazon, Google, Facebook, AT&T, NBCUniversal, FedEx, etc. has already adopted Hadoop to analyze data for financial, marketing, advertising, and sentiment and risk analysis purpose.

Gupta and Kiran (2014) stated that the tool used by Big Data Analytics for processing unstructured data is, Hadoop. Hadoop Implements Map-Reduce framework and can be used to filter unstructured data and semi-structured data into structured formats that can be loaded into any other analytic platforms. Hadoop consists of two components, Hadoop Map-Reduce for parallel data processing to extract higher-value data from raw files and the Hadoop distributed file system (HDFS) that supports low-cost, scale-out storage. Hadoop is much more effective than any other data analytics tools and it provides flexibility that can store any data type of any size. Hadoop provides scalability that can store data from terabytes to petabytes. Hadoop is designed to work on high volume, high velocity, and on various varieties of data. Therefore, Hadoop ecosystem is strong enough to combine any type of old and new data sets in different powerful ways. Devakunchari (2014) stated that Hadoop is designed to run on a large number of commodity servers. The servers are arranged in a system and Hadoop software runs on every server. He further stated that Hadoop is best suitable to run large datasets that are complex and computationally extensive. Nandimath et al. (2013) stated that Hadoop ecosystem consists of four main components, Hadoop Common, Hadoop distributed file system, Hadoop Map-Reduce, and Yarn. Hadoop Common also known as Hadoop core contains utilities and libraries that can be used by other modules within the

Hadoop ecosystem. It contains necessary java archives files and scripts required to start Hadoop.

Hadoop distributed file system is the default storage for data processed inside Hadoop. It creates several replicas of the data block across different Hadoop clusters to make data accessible and reliable. HDFS architecture works on master-slave model and consists of three main components, NameNode, DataNode, and Secondary NameNode. NameNode is the master node to keep track of the storage clusters and DataNode acts as a slave node performing the data processing within the Hadoop cluster. They further explained that Map-Reduce is a java-based system where actual data from HDFS gets processed. Map-Reduce further breaks down the big chunk of data into smaller sub-tasks where Map sends query input into various clusters for processing and Reduce collects all the processed data as a single output. Meanwhile, both input and output task are stored in HDFS. Another main component in Hadoop ecosystem is Yarn which is responsible for dynamic resource utilization in Hadoop framework. According to Horton works, Yarn has extended the Hadoop capabilities to adopt compelling new technologies within data center and is very cost effecting. It further provides a consistent framework to write data access applications that run in Hadoop. These are the core components in any basic Hadoop framework but there are several other components that form an integral part of the Hadoop ecosystem with the intent of enhancing the power of Apache Hadoop and providing better integration with databases.

Hadoop provides simplified access to the data stored in HDFS and sends data to data processing layer using different types of infrastructures like Hive, Pig, Mahout, Avro, etc. All of these infrastructures resides on the top of the Hadoop ecosystem to summarize the big data

concepts. Hive was originally developed by Facebook and later by Apache that allows users to write query in a SQL-like language, HiveQL, converting to Map-Reduce. It is so simple that any developer having the knowledge in SQL can write query for data processing. It first stores schema metadata in database and stores the data in HDFS and operates on the server side of cluster (“Hadoop ecosystem: An introduction”, 2016). Gates et al. (2009) states that Pig acts as a high-level data flow in between SQL and Map-Reduce. Pig was first adopted by Yahoo and has its own programming language known as, Pig Latin which can be compiled into a sequence of Map-Reduce jobs and executed in Hadoop ecosystem. Pig is a procedural programming language and supports all parallel programming conditional constructs (FOREACH, FLATTEN, GROUPBY, etc.). Gates et al. (2009) further admitted that Pig is a very useful framework for processing log files, aggregation of data warehousing, filtering media files, etc. Eluri, Ramesh, Al-Jabri, and Jane (2016) stated that in order to elevate the scalability of data clusters, Apache Mahout data clustering algorithms can be implemented on the top of Hadoop using Map-Reduce paradigm. They studied two different clustering techniques K-means clustering technique and Canopy-clustering technique using Apache Mahout. The K-means thus seems to be efficient to implement but is only suitable for globular data set whereas the Canopy clustering technique implementing Mahout was suitable for both globular and non-globular data set. Apache Mahout is, therefore, useful on filling the data from large data clusters by using recommendation engine of Apache Mahout.

Jain (2013) stated that using Apache Sqoop data can be efficiently transferred between Apache Hadoop and structured data store relational databases. Sqoop is useful when there is a need of loading bulk data into Hadoop from production systems or accessing it from Map-

Reduce applications running on a large cluster. Sqoop is a batch-oriented and thus is not suitable for low latency interactive query operations but mitigates excessive data load when required.

Information related to this research has been reviewed and studied in this chapter. In the next chapter we will focus on the research methodology applied in this research and the achieved results.

Chapter 3: Methodology and Results

As stated in earlier chapter about Hadoop ecosystem and its component, this research adopts one of the Hadoop components, Apache Pig as the research topic and unstructured data, especially some log files over the Hadoop framework. The log files is processed on Apache Pig framework and analyzed using Apache Pig scripting language, *Pig Latin* to extract useful information in a structured form. Output is based on the type of input log files used and the Apache Pig framework where the data is being processed and analyzed. Map-Reduce allows programmer to specify map function on the input data followed by reduce function to generate output. However, working on how we to fit any data to work under this pattern is itself a big challenge. In Pig, the data-structure itself are multivalued and nested thus can process any data type. Pig basically consists of two different components, *Pig Latin* and *Grunt*. Pig Latin is the scripting language that Pig uses for series of extractions, and transformations to produce output from the input on Pig environment known as Grunt. The development cycle in Map-Reduce framework is tedious and hectic due to the codes being too long and complex. In Pig Latin just using few lines of code in a simpler way can do the same job. Pig was created by Yahoo to provide data analytics to mine large amount of data. Pig divides the data transformation into series of Map-Reduce job within few lines of code which allows developers to focus on data rather than nature of execution. Pig automatically optimizes task without execution Pig Latin, on writing query supports most of the commands similar to SQL and relational operators. Pig scans large volume of dataset once which might not be suitable when we need to process data on batches. But Pig runs on client side

applications and launches jobs that interacts with HDFS from any workstation. Pig can also be implemented for data profiling using sampling and also for quick hypothesis testing.

Pig Architecture

Pig performs on the top of Hadoop and can read data from HDFS for data extract, transform and load (ETL) process. In this section, we will discuss in detail about the Pig architecture and its components. Pig uses Pig Latin as its scripting language that be further written using built-in operators that runs on Pig environment. There are mainly three ways to execute Pig Latin scripts: first, Grunt mode which is an interactive mode of Pig. In addition to this, there are certain useful shell and utility commands supported by grunt shell useful for testing syntax and ad-hoc data exploration. Second, Script mode which runs as a set of instructions from a file and is executed by the Pig server. Third is the embedded mode certain user defined functions (UDF) can be used using different other languages like Java, Ruby, Python, etc. This mode is suitable to create Pig Scripts on the fly.

As shown in Figure 1, Pig scripts from Grunt or Pig server passes through Parser. The parser parses the code by checking syntax in the script and generates a direct acyclic graph (DAG) as an output. DAG represents all the Pig Latin statements and logical operators. The logical operator acts as nodes and data flows in between the edges.

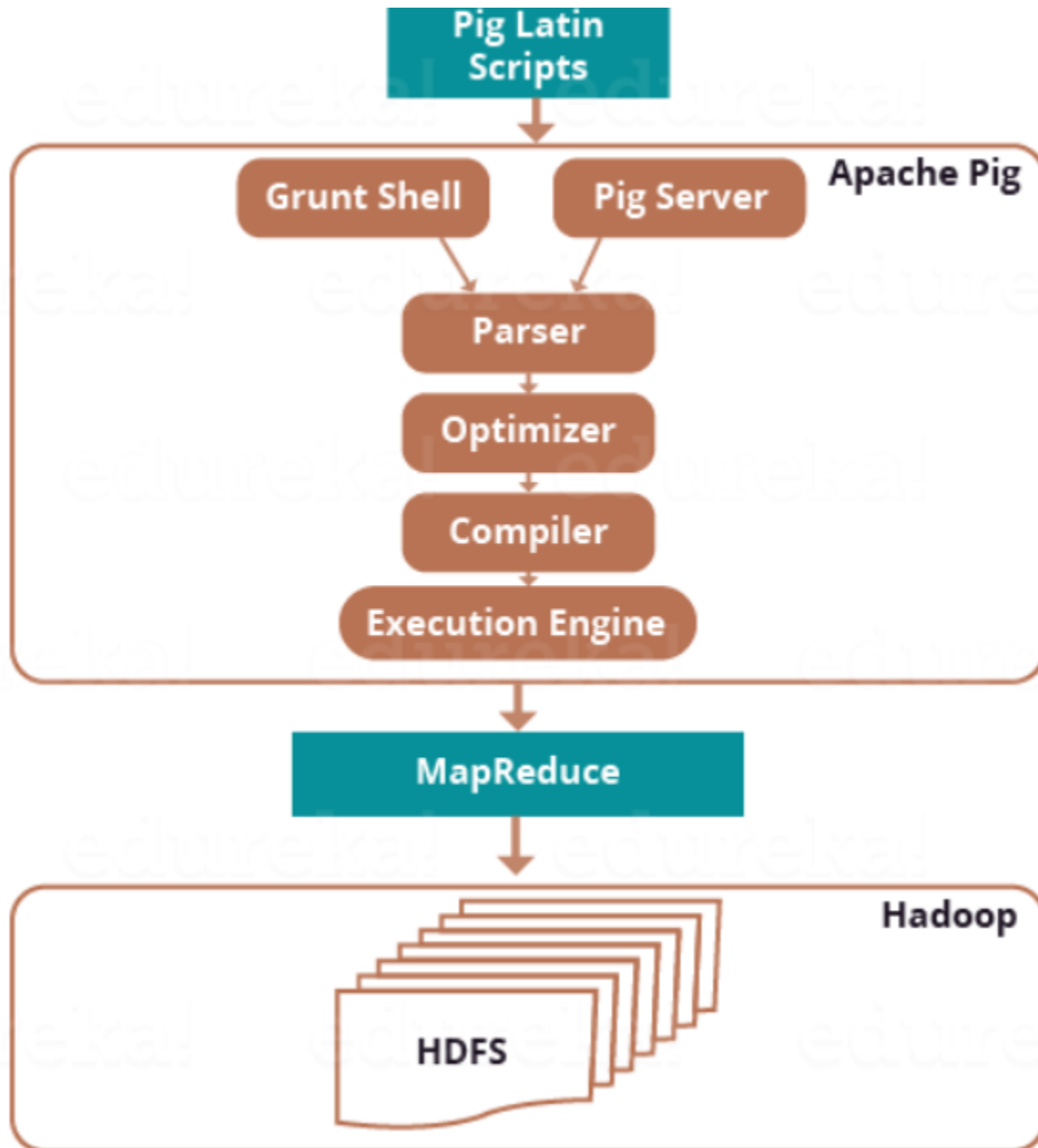


Figure 1. Pig architecture (“Hadoop ecosystem: An introduction”, 2016).

The Optimizer gets DAG as the input and optimizes the script using splits, merge, transform, reorder, etc. The optimizer also reduces the amount of data in the pipeline or reduce phase whereas for most of the Map-Reduce layer optimization rule aims to optimize

the MapReduce job properties either to include compiler or not. Additionally, at this layer query Joins, ORDER BY and GROUP BY functions are implemented.

After the script optimization, the compiler compiles the optimized code into series of MapReduce Jobs. Pig jobs can be automatically converted into MapReduce jobs by compiler. When the MapReduce jobs are submitted for execution to the execution engine either in local mode using single JVM, required result are generated as output. To view the result *DUMP* statement is required.

Pig Data Types

There are two different types of Pig's data, scalar types (single value) and complex types. Scalar data types are represented in Pig interfaces by java.lang classes and are same as the types in most programming languages. Complex data types are Tuples, Bag, and Map. A separate table is provided to explain data model in Pig as shown in Table 1.

Table 1

Example of Pig Data Model

Id	First Name	Last Name	Faculty
1	John	Doe	Management
2	Mary	Smith	IT
3	Jane	Porter	Education

Tuples

} Bag

A Tuple is a fixed length, ordered collection of fields and also represented as the record stored in relational database. For each table, we can access fields in each tuple using indexes of the fields. The fields elements can be of any type and is analogous to the the row in SQL. Fields are compared as SQL columns and referred by positions. This allows Pig to check the data in the tuple. Pig further allows user to reference the fields of the tuple by name.

Tuple constants use parentheses to indicate the tuple and commas to delimit fields in the tuple as shown in Example 1.

Example 1. {1, John, Doe, Management}

Similarly, a Bag is an unordered collection of a set of tuples. A bag might be redundant which means, it can have same tuples within the schema and hence not reference by the position. For Apache Pig to process a bag, the sequence for fields and the respective data types must be in sequence. Bag constants are constructed using braces, with tuples in the bag separated by commas as shown in Example 2. which shows the relation between student and their faculty and constructs a bag combined from many ordered tuples. In Example 2, (John, Doe, Management), (Mary, Smith, IT), (Jane, Porter, Education) are tuples and Bag is represented by parenthesis including all these tuples.

Example 2. {(John, Doe, Management), (Mary, Smith, IT), (Jane, Porter, Education)},
 {(John, Doe, Management), (Mary, Smith), (IT, Jane, Porter), (Porter, Education)}
 (John, {(Doe, Management), (Mary, Management)}).

A map is a key-value pair represented as data elements. Maps contain unique keys and are represented as chararray [] and contain unique column name that can be indexed to access the value associated with it. Because Pig does not know the type of the value, it will assume that it is a byte array. However, the actual value might be something different. If the value is of type other than the byte array, Pig will figure out the value of the data type at runtime. As shown in Example 3, Map constants are formed using brackets that delimits the map using *hash* between keys and values, and a comma in between key-value pairs is used. In

Example 3, there are two keys, “FirstName” and “Id”. The first value is a char array and the second is an integer.

Example 3. [FirstName# John, Id#1], [FirstName#Mary, Id#2]

Both simple and complex data types can be attached to the Pig schema during Load. By default, char array is the default data type for Pig.

Besides of the different Pig invocations, Pig mainly runs on two modes, Local and MapReduce mode. In this research, we will be running Pig on MapReduce mode where Pig job runs as a series of map-reduce jobs and HDFS is the storage for both input and output data file. Different types of log files are processed in Pig processing system such as web server logs files and access log files.

General Workflow of Apache Pig

When user-defined functions is implemented or any input log files is used inside the Pig GRUNT shell, the following steps are involved as explained in Figure 2.

1. Script Parsing
 - Check the syntax and valid reference variables.
 - Check whether data types are properly initialized or not.
 - Schema inference
2. Logical Optimizer
 - Pass logical plan by generating a sequence of well-founded semantics query models to solve the query.

3. Physical plan

- Translate logical plan to physical plan for each logical operator specifically by describing the physical operators that Pig will use to execute the scripts.

4. MapReduce plan and its optimization

- Assign each physical operator to a MapReduce stage (Map task and a Reduce task).
- Minimize the number of reduce stages based on the nature of operations of MapReduce optimizations.

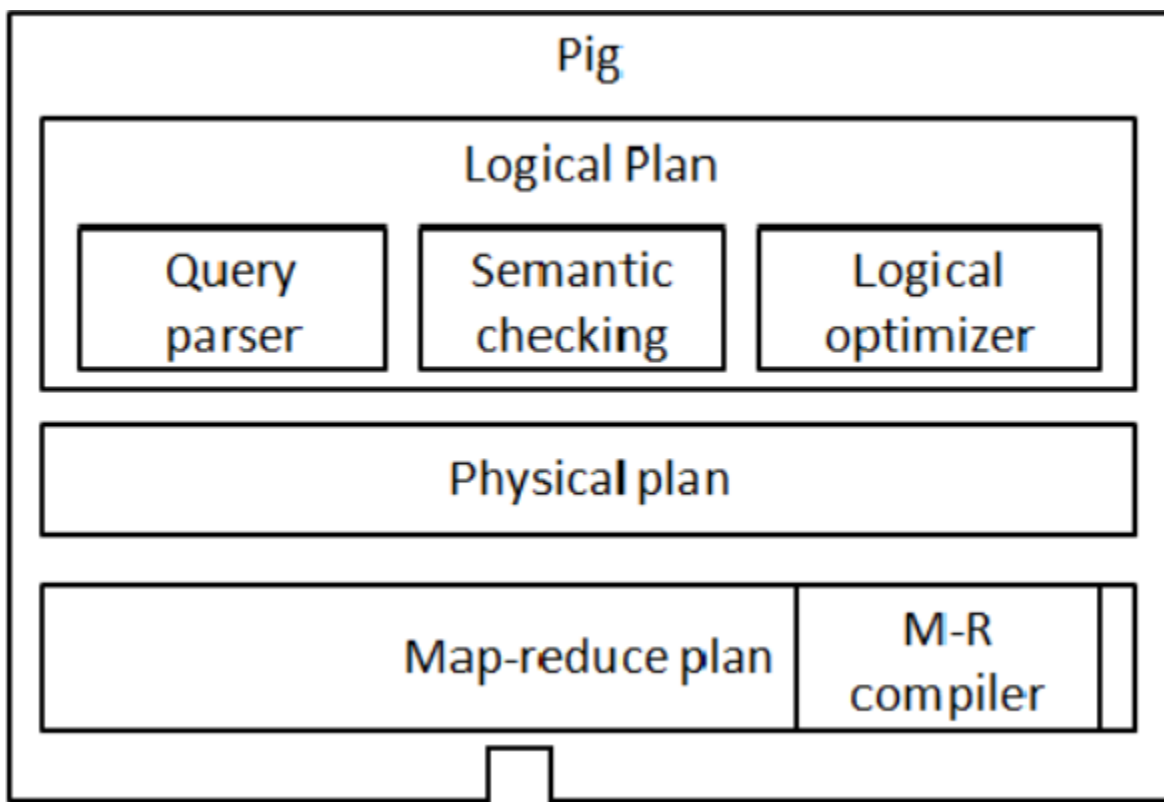


Figure 2. Workflow architecture of Apache Pig (“Hadoop ecosystem: An introduction”, 2016).

The processes involved inside the Pig architecture to transform the Pig Latin programs into executable code is explained in above Figure 2. The executable code is in the form of MapReduce tasks passing through different stages that includes a parser, logical optimizer, Physical optimizer, MapReduce optimizer and the MapReduce compiler.

Pig can also be used for column transformation, filtering, ordering, and Custom aggregation. Pig also opens up the power of MapReduce. To know the workflow of Apache Pig, our research is focused on analyzing data from web log files as explained in the test case below.

Case Study: Access Log Report Analytics

This case study includes using apache Pig commands and creating some user defined functions to process access log data files. The log reports used in analysis are the two month's HTTP requests to the NASA Kennedy Space Center WWW server in Florida . The log reports contains logs collected from July 1, 1995, through July 31, 1995, a total of 31 days. The log file contains Server IP address, Method (GET / POST), Request URI (request_link and request destination) , HTTP status code, and User-agent. The case study illustrates the methodology of processing and analytics of Apache Pig using Hadoop distributed file system.

Problem. The format of access logs are very cryptic. These files exist typically for technical site auditing and troubleshooting. In order to maintain security and investigate the incidents around the web within any organizations, these logs need to be monitored and analyzed. With proper analysis and inspection, one can identify intrusion attempts, misconfigured equipments, user behavior, and much more.

In this case, we will find the total number of times a website that has been visited in last 31 days (July 1 to July 31).

Input (Access log files). Access log files are the files generated by server and also known as server log files to provide information requested to the server from users. When an user connects to a site, the computer, browser, and network will deliver some data to the site server itself to create a record that file was requested. A sample of web server log file appears below as Figure 3.

```

AccessLogFile1 - Notepad
File Edit Format View Help
199.72.81.55 - - [01/Jul/1995:00:00:01 -0400] "GET /history/apollo/ HTTP/1.0" 200 6245unicomp6.unicomp.net - - [01/Jul/1995:00:00:06 -0400] "GET /shuttle/cou
p.net - - [01/Jul/1995:00:00:14 -0400] "GET /shuttle/countdown/count.gif HTTP/1.0" 200 40310unicomp6.unicomp.net - - [01/Jul/1995:00:00:14 -0400] "GET /image
asahi-net.or.jp - - [01/Jul/1995:00:00:19 -0400] "GET /images/launchpalms-small.gif HTTP/1.0" 200 11473205.189.154.54 - - [01/Jul/1995:00:00:24 -0400] "GET /
-0400] "GET /images/NASA-logosmall.gif HTTP/1.0" 200 786ix-orl2-01.ix.netcom.com - - [01/Jul/1995:00:00:41 -0400] "GET /shuttle/countdown/ HTTP/1.0" 200 398[
HTTP/1.0" 200 12054schesyer.clark.net - - [01/Jul/1995:00:00:58 -0400] "GET /shuttle/missions/sts-71/movies/sts-71-mir-dock-2.mpg HTTP/1.0" 200 49152ppp-nyc-
05.189.154.54 - - [01/Jul/1995:00:01:08 -0400] "GET /shuttle/missions/sts-71/images/images.html HTTP/1.0" 200 7634www-a1.proxy.aol.com - - [01/Jul/1995:00:0
istory/apollo/images/footprint-small.gif HTTP/1.0" 200 18149205.189.154.54 - - [01/Jul/1995:00:01:19 -0400] "GET /shuttle/missions/sts-71/images/KSC-95EC-04
images/launchmedium.gif HTTP/1.0" 200 11853slip1.yab.com - - [01/Jul/1995:00:01:29 -0400] "GET /shuttle/resources/orbiters/endeavour.gif HTTP/1.0" 200 169911:
0:81:43 -0400] "GET / HTTP/1.0" 200 7074link097.txdirect.net - - [01/Jul/1995:00:01:44 -0400] "GET /shuttle/missions/sts-78/mission-sts-78.html HTTP/1.0" 200
atech.org - - [01/Jul/1995:00:01:52 -0400] "GET /images/NASA-logosmall.gif HTTP/1.0" 200 786199.72.81.55 - - [01/Jul/1995:00:01:52 -0400] "GET /images/WORLD
01/Jul/1995:00:01:57 -0400] "GET / HTTP/1.0" 200 7074ix-orl0-06.ix.netcom.com - - [01/Jul/1995:00:01:57 -0400] "GET /software/win/vn/usrguide/wvnguide.gif H
.gif HTTP/1.0" 304 0netport-27.iu.net - - [01/Jul/1995:00:02:01 -0400] "GET /images/NASA-logosmall.gif HTTP/1.0" 304 0netport-27.iu.net - - [01/Jul/1995:00:0
/Jul/1995:00:02:12 -0400] "GET /shuttle/countdown/liftoff.html HTTP/1.0" 200 4538schesyer.clark.net - - [01/Jul/1995:00:02:13 -0400] "GET /shuttle/missions/st
T /shuttle/countdown/video/livevideo.gif HTTP/1.0" 200 62761lmsmith.tezcat.com - - [01/Jul/1995:00:02:16 -0400] "GET /shuttle/countdown/ HTTP/1.0" 200 3985[
sts-71/images/KSC-95EC-0918.txt HTTP/1.0" 200 1391link097.txdirect.net - - [01/Jul/1995:00:02:25 -0400] "GET /shuttle/missions/sts-65/sts-65-patch-small.gif
th-pc.moorecap.com - - [01/Jul/1995:00:02:38 -0400] "GET /history/apollo/images/apollo-small.gif HTTP/1.0" 200 9630dynip42.efn.org - - [01/Jul/1995:00:02:39
1.0" 200 509ix-orl0-06.ix.netcom.com - - [01/Jul/1995:00:02:47 -0400] "GET /icons/menu.xbm HTTP/1.0" 200 527ix-orl0-06.ix.netcom.com - - [01/Jul/1995:00:02:4
-0400] "GET /cgi-bin/imagemap/countdown?102,210 HTTP/1.0" 302 95dd11-054.compuserve.com - - [01/Jul/1995:00:03:01 -0400] "GET / HTTP/1.0" 200 7074onyx.south
" 200 12054brandt.xensei.com - - [01/Jul/1995:00:03:08 -0400] "GET /images/KSC-logosmall.gif HTTP/1.0" 200 1204piweba3y.prodigy.com - - [01/Jul/1995:00:03:08
tch-small.gif HTTP/1.0" 200 12054129.188.154.200 - - [01/Jul/1995:00:03:14 -0400] "GET /images/launchpalms-small.gif HTTP/1.0" 200 11473lmsmith.tezcat.com -
GET /images/KSC-logosmall.gif HTTP/1.0" 200 1204www-a1.proxy.aol.com - - [01/Jul/1995:00:03:20 -0400] "GET /cgi-bin/imagemap/countdown?107,144 HTTP/1.0" 302
00 49087midcom.com - - [01/Jul/1995:00:03:33 -0400] "GET /history/apollo/apollo-13/apollo-13-patch-small.gif HTTP/1.0" 200 12859ix-orl0-06.ix.netcom.com - -
" 200 17459slip1.yab.com - - [01/Jul/1995:00:03:40 -0400] "GET /images/landing-747.gif HTTP/1.0" 200 110875isdn6-34.dnai.com - - [01/Jul/1995:00:03:40 -0400]
0400] "GET /shuttle/missions/sts-71/images/KSC-95EC-0868.jpg HTTP/1.0" 200 61848205.189.154.54 - - [01/Jul/1995:00:03:51 -0400] "GET /shuttle/missions/sts-7
shuttle/missions/missions.html HTTP/1.0" 304 0teleman.pr.mcs.net - - [01/Jul/1995:00:03:57 -0400] "GET /images/launchmedium.gif HTTP/1.0" 304 0teleman.pr.mcs
995:00:04:05 -0400] "GET /images/MOSAIC-logosmall.gif HTTP/1.0" 200 363129.188.154.200 - - [01/Jul/1995:00:04:05 -0400] "GET /images/USA-logosmall.gif HTTP/
apollo/images/apollo-logo.gif mission-sts-67.html HTTP/1.0" 200 21408129.94.144.152 - - [01/Jul/1995:00:04:25 -0400] "GET /shuttle/technology/sts-newsref/stsr
01/Jul/1995:00:04:27 -0400] "GET /images/construct.gif HTTP/1.0" 200 1414slip1.yab.com - - [01/Jul/1995:00:04:28 -0400] "GET /history/history.html HTTP/1.0"
tle/missions/sts-71/movies/sts-71-hatch-hand-group.mpg HTTP/1.0" 200 49152netport-27.iu.net - - [01/Jul/1995:00:04:34 -0400] "GET /images/KSC-logosmall.gif
gosmall.gif HTTP/1.0" 200 786savvy1.savvy.com - - [01/Jul/1995:00:04:44 -0400] "GET /images/KSC-logosmall.gif HTTP/1.0" 200 1204news.ti.com - - [01/Jul/1995
.39.14 - - [01/Jul/1995:00:04:52 -0400] "GET /shuttle/countdown/ HTTP/1.0" 200 3985news.ti.com - - [01/Jul/1995:00:04:52 -0400] "GET /images/WORLD-logosmall
-71/images/images.html HTTP/1.0" 200 7634ix-sd11-26.ix.netcom.com - - [01/Jul/1995:00:05:06 -0400] "GET /cgi-bin/imagemap/countdown?107,144 HTTP/1.0" 302 96:
m.com - - [01/Jul/1995:00:05:17 -0400] "GET /icons/menu.xbm HTTP/1.0" 200 527net-1.141.eden.com - - [01/Jul/1995:00:05:17 -0400] "GET /shuttle/missions/sts-
ect.net - - [01/Jul/1995:00:05:23 -0400] "GET /icons/menu.xbm HTTP/1.0" 200 527ix-war-m11-20.ix.netcom.com - - [01/Jul/1995:00:05:23 -0400] "GET /icons/text

```

Figure 3. Web server log files.

The data as shown in above Figure 3 is space separated. It consist of IP address, timestamp , timezone, request type, requested link , request details, response code and user agent(bytes). Usually, the scale of these files is quite huge and running queries in conventional method is not possible. Therefore we will implement Apache Pig to manipulate these log files and generate necessary statistics which helps us to understand the usage of the

server, website popularity, user visit frequency, and total bytes transferred. The main purpose of using access log files is to:

- i. Identify the most popular websites to get insight using user defined functions in Pig Processing system.
- ii. Implement custom load functions to load Apache's common log format files into Apache Pig.
- iii. Use several join operators from Pig join based datasets on values common to each dataset

Expected output. The result should be able to show the total number of visits of any website within the given timestamp.

Example 4. **www.google.com has been visited 272 times from 03/15/2017 to 03/20/2017.**

Solution. The logs files are downloaded from NASA-HTTP website. In order to process unstructured web server log files, the primary step is to load the datasets into Hadoop Distributed File System (HDFS). It is done by creating a directory in HDFS and storing the access log files in HDFS for further processing.

The primary objective of this research is to implement Pig processing system to find a structured records of the URL visits throughout any given time period recorded in an unstructured data type log file. The input data from HDFS is loaded into Pig system. Pig has a special keyword called "*Load*" to load data into PigStorage. In the Pig system, data is stored, and transformed by using relations. Each transformation of data is assigned to a variable as a data as shown in the example below. The location to load file is based on that particular mode

(MapReduce or local) in which Pig is executed. We implemented both *MapReduce* mode and *Local* mode. The MapReduce mode is also known as cluster mode where the file is located inside the specified HDFS location. In given Code Snippet 1, the data is loaded from the Hadoop Distributed File System located in `‘/user/AccessLogAnalysis/WebLogs/AccessLogFile1’` file location by using the variable `‘data’`. The data is loaded specifying specific schema for the *AccessLogFile1* file. Let’s take a look at `ip_add[chararray]`, `ip_add` is the input keyword informing Pig that is of a `chararray` type.

Code Snippet 1.

```
data = LOAD
'/user/AccessLogAnalysis/WebLogs/AccessLogFile1'
usingPigStorage (' ')AS (
    ip_add:chararray,
    temp1:chararray,
    temp2: chararray,
    timestamp: chararray,
    timeZone: chararray,
    cs_method:chararray,
    cs_uri:chararray,
    request_dest:chararray,
    port: int,
    bytes: int
);
```

The relations are defined for each transformed data which act as a reference to a set of data. `Ip_add` is reference as the IP address of all the URL listed in the input *AccessLogFile1* file. In other words, these relations acts as a set of tables with user-defined functions and columns. As mentioned in Code Snippet 1, the data loaded as input is space separated and each line is a distinct event. Output generated by *Describe* command confirms same schema which is specified along *with LOAD*. After the data is loaded as shown in Code Snippet 1, in order to calculate the total bytes transferred in every interval of time, the data is aggregated based on upon the timestamp as shown in Code Snippet 2.

Code Snippet 2.

```
time_data = GROUP data BY timestamp;
  DESCRIBE time_data;
  byte_count = FOREACH time_data GENERATE group AS
  timestamp, SUM(data.bytes) AS total_bytes;
```

The above Code Snippet 2, defines a relation '*time_data*' and '*byte_count*'. The *time_data* consists of data grouped by *timestamp*. So, the new column in relation *byte_count* are *timestamp* and *total_bytes*. The data values are extracted from *time_data* relation which is grouped by *timestamp* and sum of the bytes of grouped data is calculated as *total_bytes*. Each group of data generated in this case are time stamps at which the requests were received at the server.

As shown in Code Snippet 3, we load all the *data* into *ip_data* which is grouped by *ip_add*, IP address of the access log input files. Then for each of data loaded in *ip_data*, *ip_counts* is generated as the total visits.

Code Snippet 3.

```
ip_data = GROUP data by ip_add;
DESCRIBE ip_data;
ip_count = FOREACH ip_data GENERATE group AS
timestamp, COUNT(data) AS total_visits;
```

The count of the total number of requests is received from a specific IP address and is generated to get the total number of visits by the user. The query is further executed to sort rank the data based on total visits to get the time at which maximum visits are recorded as shown in Code Snippet 4.

Code Snippet 4.

```
sort_data = RANK ip_count BY total_visits DESC;
DUMP sort_data;
STORE ip_count into
'/user/AccessLogAnalysis/OutPut3' USING PigStorage
(',');
```

Finally, *DUMP* keyword dumps the result and stores output to the specified output file location using *STORE* command as shown in Code Snippet 4.

The *PigStorage* function used in above Code Snippet 4 is an inbuilt read function with the space separated arguments to read data. Data schemas are declared when column based operations is executed. We define name of each column and column data type. The column data type by default is, *chararray* data type but there are also other built-in data type supported by Pig. *Port* and *bytes* columns are integer data type and other columns take *chararray* as default.

Log analysis is crucial and contains a lot of information. *DESCRIBE* command can describe any relation. This can be useful when we need to understand how ‘Join’ and ‘Group’ statements are used in a particular relation. *DESCRIBE* *time_data* describes *time_data* and using other statements like ‘*DISTINCT*’, ‘*FILTER*’, ‘*GROUP BY*’ are useful to eliminate data redundancy.

Filters can be used to filter data on the basis of user requirements. This case study filters all the data based IP addresses which has Port number as 200.

```
Filtered_data = FILTER ip_data by Port == 200;
```

The output here is also created in HDFS as shown in Figure 4.

```
grunt> sort_data = Rank ip_count BY numberOfVisits DESC;
grunt> DUMP sort_data;
```

Figure 4. Dump result to generate output.

Result. By analyzing and processing log files in the Pig system, we are able to get the visits of specific user, visit per unit time and failed request. As shown in Figure 4, the number

of views of per web page is retrieved by using the *DUMP* command and the result is printed in the grunt shell without storing them into a file. The analysis is further carried and stored into database to make data available for end users. When the Pig system completes its analysis, the output will be stored in HDFS and a complete report of file location will be available as shown in Figure 5.

```

HadoopVersion PigVersion  UserId  StartedAt    FinishedAt    Features
2.7.1  0.16.0  hadoop1  2017-02-08 23:44:02    2017-02-08 23:47:02    GROUP_BY,DISTINCT

Success!

Job Stats (time in seconds):
JobId  Maps  Reduces  MaxMapTime  MinMapTime  AvgMapTime  MedianMapTime  MaxReduceTime  MinReduceTime  AvgReduceTime  MedianReduceTime  Alias  Feature  Outputs
job_local825091494_0004  2      1      n/a        n/a         n/a         n/a           n/a           n/a           ip_count,ip_data  GROUP_BY,COMBINER  /user/AccessLogAnalysis/OutPut1,
job_local994152450_0003  2      1      n/a        n/a         n/a         n/a           n/a           data             DISTINCT

Input(s):
Successfully read 1891715 records (2031283308 bytes) from: "/user/AccessLogAnalysis/WebLogs/AccessLogFile1"

Output(s):
Successfully stored 81983 records (1607741976 bytes) in: "/user/AccessLogAnalysis/OutPut1"

Counters:
Total records written : 81983
Total bytes written : 1607741976
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_local994152450_0003 ->   job_local825091494_0004,
job_local825091494_0004

```

Figure 5. Pig output log.

As shown in Figure 5, after executing the *DUMP* Command, the output is stored as a output file in Hadoop Distributed File System, HDFS location. The first couple of lines gives the brief summary of the job. Hadoop version, Pig version, UserId, startedAt is the time Pig submits the job, not the time the first job starts running the Hadoop cluster. FinishedAt is the time Pig finishes processing the job, which will be slightly after the time the last MapReduce job finishes.

The section labeled as Job Stats as shown in Figure 5, gives a breakdown of each MapReduce job that was run. They include how many map and reduce tasks each job has, in our case we have one map and one reduce task. Also, include statistics on how long these

tasks took and a mapping of aliases in the Pig Latin script to the jobs. The input , output and Counters sections are self-explanatory. In the above Figure 5, total 1891715 records is from the input *AccessLogFile1* and 81983 records is stored in the output file in a structured format.

URL	Total Visits
taltal	3749
triton	3736
acm.org	3712
apt.com	3672
asi.com	3650
bix.com	3597
cml.com	3579
crl.com	3561
dfw.net	3178
diab.se	3116
efn.org	3017
flo.org	2946
fsd.com	2669
fwi.com	2597
amsa.ch	2555

Figure 6. Output sample.

As shown in Figure 6, the table shows the lists of URL's and the count of visits per URL after executing the *webServerLog.pig* file in the Pig system (on the next page).

The group of operation forces a reduce phase where we can defined the level of parallelism to be implemented. If no parallelism is specified, the number of reducers are calculated by using formula:

```
reducers = (int)Math.ceil((double)totalInputFileSize / bytesPerReducer)
TotalReducers = Math.min(maxReducers, reducers)
```

Where,

maxReducersNumber is 999 by default

bytesPerReducer is 1073741824 (1GB) bytes by default.

$$\begin{aligned} \text{TotalReducers} &= \text{Min}(999, (2031283308 / 1073741824)) \\ &= 1.81978 \\ &\sim 1 \text{ reducers} \end{aligned}$$

In the above screenshot on Figure 5, the number of maps are 2 and reducers is 1 and on input 1891715 records were read whereas on the output 81983 records were reduced and stored in HDFS location as an output file.

Code (webservlog.Pig). The code shown below is the Pig Latin script that is used to load *AccessLogFile3* file and defined the our own schema for the input which is space delimited . The ouput is generated and stored in HDFS file location as a *OutPut3* file in a strcutured format seperated by comma.

```
data = load
'/user/AccessLogAnalysis/WebLogs/AccessLogFile3'
using PigStorage (' ') AS (
ip_add: chararray,
temp1: chararray,
temp2: chararray,
timestamp: chararray,
timeZone: chararray,
cs_method: chararray,
cs_uri: chararray,
request_dest: chararray,
port: int,
bytes: int
);
data = DISTINCT data;
time_data = GROUP data BY timestamp;
DESCRIBE time_data;
byte_count = FOREACH time_data GENERATE group AS
timestamp, SUM(data.bytes) AS total_bytes;
ip_data = GROUP data by ip_add;
DESCRIBE ip_data;
ip_count = FOREACH ip_data GENERATE group AS
timestamp, COUNT(data) AS total_visits;
```

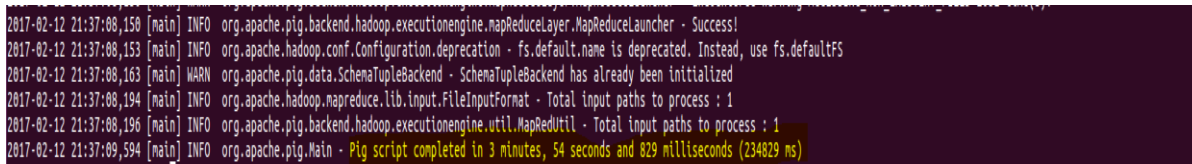
```

sort_data = RANK ip_count BY total_visits DESC;
DUMP sort_data;
STORE ip_count into
'/user/AccessLogAnalysis/OutPut3' USING PigStorage
(',',');

```

As this code is executed, the jobs runs and we see the status printed in the grunt shell as number of records generated as output, number of records written, total bytes written, and job listed Direct Acyclic Graph (DAG).

Performance. The files when used in local mode took approximately 3 minutes to process each web log files whereas when ran on the MapReduce mode it took approximately 5 minutes to process the same log file of same size as shown in Figure 7 and Figure 8.

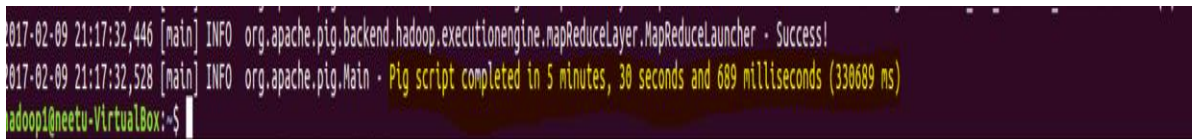


```

2017-02-12 21:37:08,150 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2017-02-12 21:37:08,153 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
2017-02-12 21:37:08,163 [main] WARN org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has already been initialized
2017-02-12 21:37:08,194 [main] INFO org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2017-02-12 21:37:08,196 [main] INFO org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
2017-02-12 21:37:09,594 [main] INFO org.apache.pig.Main - Pig script completed in 3 minutes, 54 seconds and 829 milliseconds (234829 ms)

```

Figure 7. Processing time of Pig Script on local.



```

2017-02-09 21:17:32,446 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2017-02-09 21:17:32,528 [main] INFO org.apache.pig.Main - Pig script completed in 5 minutes, 30 seconds and 689 milliseconds (330689 ms)
hadoop1@neetu-VirtualBox:~$

```

Figure 8. Processing time of Pig Script on MapReduce Mode.

Table 2

Performance Evaluation on Local Mode and MapReduce Mode

Mode	Time Taken	File Size
Local Mode	3 minutes (approx..)	283128383308 bytes
MapReduce Mode	5 minutes (approx..)	283128383308 bytes

Although the processing time in local mode seems more efficient than MapReduce mode as shown in Table 2. The results shown above is generated when the map task is only one (one server), but what happens when our job has 10,000 map tasks? Local mode fails to execute as it runs locally on only one server (single node) therefore when the map task is increased or when data is processed through large clusters (multiple node), MapReduce mode seems to be more efficient than local mode.

In all above processes of inserting unstructured input log files in Apache Pig processing system, implementing and generating some useful information from the logs, we are able to get that in Apache Pig framework using some of our user-defined function. As the input file size increases, each MapReduce stage performs certain optimizations and the job gets reduced drastically. Pig processing is useful to build behavior prediction model based on the user interaction within a website. Using Pig-based flow analyzer the analysis on web server logs files is done with increased convenience and easiness without any in-depth knowledge of programming expertise. Using typical MapReduce paradigm, MapReduce task division module and inbuilt Pig compiles to get the total visits per URL within the given time frame. In the next chapter, we will conclude the output results from the research and propose some future works from our analysis.

Chapter 4: Conclusions and Future Work

With the types of data growing everywhere, the management of data is a great challenge. To manage data, all organizations must develop a data-driven culture, where collection and storage are the integral part of the business. In order to handle this complexity, organizations need a combined knowledge of analyzing all the types of data that carry some sort of information related to the firm. This paper reveals how these complex weblog files are analyzed using one of the Hadoop data analytics tools, Apache Pig. Pig can easily process data that were extracted from diverse sources and process them using its simple scripting language known as Pig Latin. The data is processed using several Pig Latin statements and filtered out in order to remove data redundancy. The data processed in this paper is downloaded from NASA web server log files, stored in HDFS, and is accessed around the cluster to perform MapReduce tasks. From the results achieved, it shows that a large amount of unstructured weblog files will generate useful information in a structured form when used on the Apache Pig processing system.

Big data implementation is growing recently and among different types of big data frameworks, Apache Pig is a work in progress. It is an open-source project and is being actively worked on by Yahoo, Facebook, as well as several other health and external contributors to gather useful information from the unstructured data. From year 2006 when Pig was first discovered till recent, Pig itself has developed its own language that includes several Pig Latin statements, data types, general and relational operators, and user-defined functions (UDFs). By implementing Apache Pig, web server log analysis is much simpler and the time consumed for researching on the complex control loops and programming constructs for

implementing on a typical local mode, for one map job or MapReduce mode for several maps jobs decreased by a greater percentage.

Beside from processing data, the data that comes in can be in any state and might contain some useful and redundant data as well. The processing system must not only be able to process the data but also support data cleansing and profiling. Apache Pig is enriched with these features to overcome the data quality concerns. Aggregated data from Pig is a minimal subset of data and is loaded to Data Warehouse for Business Analytics and Enterprises. Reporting. Google uses the same algorithm to improve performance by examining the user behavior. Once the data gets inside the Map-Reduce cluster as input, the data goes through several processes like map and reduce and a refined output is generated in Hadoop distributed file system. The data inside Map-Reduce cluster also supports on adding additional data to the clusters without having re-index all over again. Many companies use Hadoop to analyze high-level queries that were a big challenge a few years ago. The data inside Map-Reduce cluster also generates an iterative processing model and can keep track of every new updates by joining the behavioral model with the user data. This feature is widely used in social-networking sites like Facebook and micro-blogging sites like Twitter. Pig cannot be considered as effective as MapReduce when it comes to processing small data or scanning multiple records in random order.

Future work can be extended on using Pig framework to implement unstructured weblog access files on several large clusters for prediction analysis to predict which next page will be visited by the user, implementing some artificial intelligence artifacts. Also, another

area where we can extend our future research can be, to examine the MapReduce job when the job has 10, 000 of map tasks instead of one and how efficient will be the system.

References

- Beach, C., & Schiefelbein, W. R. (2013, December 31). Unstructured data: How to implement an early warning system for hidden risks. *Journal of Accountancy*. Retrieved from <http://www.journalofaccountancy.com/issues/2014/jan/20126972.html>.
- Devakunchari, R. (2014). Handling big data with Hadoop toolkit. *International Conference on Information Communication and Embedded Systems (ICICES2014)*. doi:10.1109/icices.2014.7033839
- Elizabeth, M. (2013). *Capturing the value of unstructured data*. SAS Institute, Inc.
- Eluri, V. R., Ramesh, M., Al-Jabri, A. S. M., & Jane, M. (2016). A comparative study of various clustering techniques on big data sets using Apache Mahout. *2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC)*, Muscat, 15-16 March 2016, 1-4. doi.org/10.1109/ICBDSC.2016.7460397.
- Francis, N., & Kurian, K, S. (2015). Data processing for big data applications using Hadoop framework. *IJARCCCE*. doi:10.17148/ijarccce.2015.4343
- Gates, A. F., Natkovich, O., Chopra, S., Kamath, P., Narayanamurthy, S. M., Olston, C., ... Srivastava, U. (2009). Building a high-level data flow system on top of map-reduce. *Proceedings of the VLDB Endowment*, 2(2), 1414-1425. doi:10.14778/1687553.1687568
- Gupta, B., & Kiran, J. (2014). Big data analytics with Hadoop to analyze targeted attacks on enterprise data. *International Journal of Computer Science and Information Technologies (IJCSIT)*, 5(3), 3867-3870.
- Hadoop ecosystem: An introduction (2016). *International Journal of Science and Research (IJSR)*, 5(6), 557-562. doi:10.21275/v5i6.nov164121

- Herzig, M. D. (2011). Hybrid search ranking for structured and unstructured data. *The Institute AIFB, Karlsruhe Institute of Technology, Germany. Part II, LNCS 6644*, pp. 518-522.
- Holzinger, A., & Pasi, G. (2013). Human-computer interaction and knowledge discovery in complex, unstructured, big data. *Proceedings of the Third International Workshop, HCI-KDD 2013, SouthCHI 2013, Maribor, Slovenia, July 1-3, 2013*. Berlin: Springer.
- Insights on governance, risk and compliance*. (2010). Retrieved January 29, 2017, from <http://ey.com/GRCinsights>.
- Jain, A. (2013). *Instant Apache Sqoop*. Packt Publishing, Ltd.
- Lai, W. K., Chen, Y.-U., Wu, T.-Y., & Obaidat, M. S. (2013). Towards a framework for large-scale multimedia data storage and processing on Hadoop platform. *The Journal of Supercomputing*, 68(1), 488-507. doi:10.1007/s11227-013-1050-4
- Minsker, M. (2015, January). *Is Hadoop worth the hype*. Retrieved January 29, 2017, from www.destinationCRM.com.
- Nandimath, J., Banerjee, E., Patil, A., Kakade, P., Vaidya, S., & Chaturvedi, D. (2013). Big data analysis using Apache Hadoop. *2013 IEEE 14th International Conference on Information Reuse & Integration (IRI)*. doi:10.1109/iri.2013.6642536
- NASA-HTTP. (n.d.). Retrieved January 24, 2017, from <http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html>.
- Qazi, R. U. R., & Sher, A. (2016). Big data applications in businesses: An overview. *The International Technology Management Review*, 6(2), 50. doi:10.2991/itmr.2016.6.2.3

Appendix

A. System Specifications and Requirements

1. Ubuntu v16.0.2
2. Hadoop v2.7.1
3. Pig v0.16.0 (r1746530)

B. HDFS Web User Interface

1. Hadoop Distributed File System Overview Page

Apart from command line interface, Hadoop provides web user interface of HDFS resource manager. The HDFS interface is useful in pseudo-distributed mode and fully distributed mode. Image 1, shows the overview of the version number, cluster Id, and block pool Id of the HDFS version used.

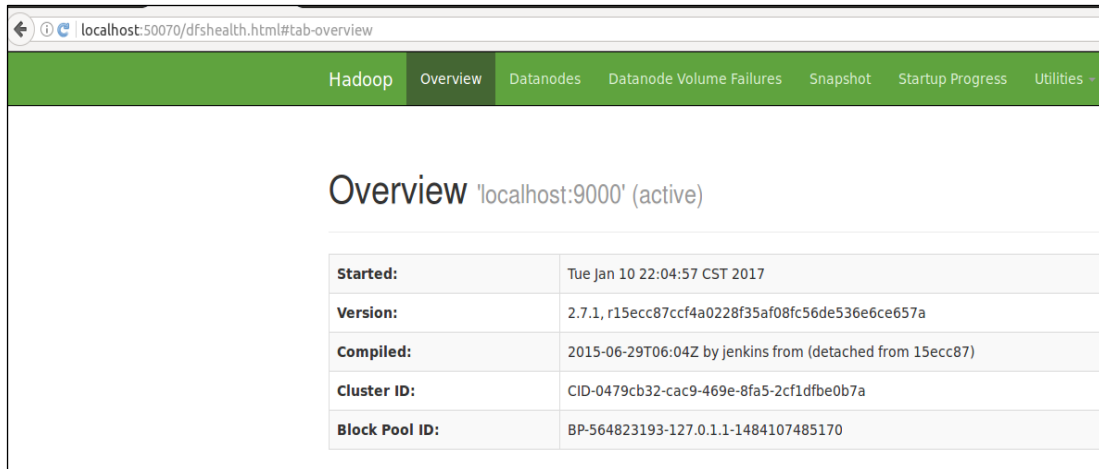


Image 1. HDFS web user interface Overview

2. Node

As shown in image 2, the summary of the configured HDFS UI which is used in our research. It includes the total size available for configuration, data node usages,

number of live nodes and dead nodes available, total number of under-replicated blocks, and block deletion start time. The live Node is the node whose children are currently being used during the process.

Summary	
Security is off.	
Safemode is off.	
105 files and directories, 49 blocks = 154 total filesystem object(s).	
Heap Memory used 43.62 MB of 65.88 MB Heap Memory. Max Heap Memory is 966.69 MB.	
Non Heap Memory used 60.84 MB of 61.84 MB Committed Non Heap Memory. Max Non Heap Memory is -1 B.	
Configured Capacity:	94.19 GB
DFS Used:	286.67 MB (0.3%)
Non DFS Used:	13.04 GB
DFS Remaining:	80.87 GB (85.85%)
Block Pool Used:	286.67 MB (0.3%)
DataNodes usages% (Min/Median/Max/stdDev):	0.30% / 0.30% / 0.30% / 0.00%
Live Nodes	1 (Decommissioned: 0)
Dead Nodes	0 (Decommissioned: 0)
Decommissioning Nodes	0
Total Datanode Volume Failures	0 (0 B)
Number of Under-Replicated Blocks	40
Number of Blocks Pending Deletion	0
Block Deletion Start Time	1/10/2017, 10:04:57 PM

Image 2. HDFS web user interface showing summary of configuration

3. Data Node Information

A DataNode stores data in HDFS. A functional file system may have more than one DataNode, with data replicated across them. The DataNode information shown in Image 3 below shows the given DataNode available in our HDFS system. “neetu-VirtualBox:50010”, is in the service state with capacity of 94.19GB and 49 Blocks. DataNode further performs read-write operations on the file system, as per client request.

Datanode Information

In operation

Node	Last contact	Admin State	Capacity	Used	Non DFS Used	Remaining	Blocks	Block pool used	Failed Volumes	Version
neetu-VirtualBox:50010 (127.0.0.1:50010)	0	In Service	94.19 GB	286.67 MB	13.04 GB	80.87 GB	49	286.67 MB (0.3%)	0	2.7.1

Image 3. HDFS web user interface showing DataNode Information

4. NameNode Information

The NameNode is the centerpiece of an HDFS. It allows user to browse the file system. NameNode acts as a master server and manages the file system namespace. NameNode also provides some logs information about current insight situation representing the transaction Id, journal manager location, and state. The edit log files contains the log records of the current situation. The NameNode directory gets configured manually on hdfs-site.xml while installing Hadoop.

NameNode Journal Status

Current transaction ID: 1075

Journal Manager	State
FileJournalManager(root=/tmp/hadoop-hadoop1/dfs/name)	EditLogFileOutputStream(/tmp/hadoop-hadoop1/dfs/name/current/edits_inprogress_000000000000001075)

NameNode Storage

Storage Directory	Type	State
/tmp/hadoop-hadoop1/dfs/name	IMAGE_AND_EDITS	Active

Hadoop, 2015.

Image 4. HDFS web user interface showing NameNode Journal Status and Storage

5. Browse directory

The image 5 shown below is the directory shown on HDFS file system. The UI has several sections, Overview, Datanodes, Snapshots, startup progress, and utilities. The browser directory listed below shows the directory created inside HDFS to store the input files and store output results. Different level of permissions can be assigned to the files and owner can be specified per each files.

The image displays two screenshots of the HDFS web user interface. The top screenshot shows the root directory listing, and the bottom screenshot shows a subdirectory listing.

Top Screenshot: Browse Directory

Path: /

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	hadoop1	supergroup	0 B	1/16/2017, 8:52:10 PM	0	0 B	tmp
drwxr-xr-x	hadoop1	supergroup	0 B	1/21/2017, 12:57:15 PM	0	0 B	user

Hadoop, 2015.

Bottom Screenshot: Browse Directory

Path: /user/AccessLogAnalysis/WebLogs

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	hadoop1	supergroup	195.73 MB	2/8/2017, 8:53:07 PM	1	128 MB	AccessLogFile1
-rw-r--r--	hadoop1	supergroup	222.63 MB	2/8/2017, 8:53:12 PM	1	128 MB	AccessLogFile2
-rw-r--r--	hadoop1	supergroup	163.04 MB	2/8/2017, 8:53:15 PM	1	128 MB	AccessLogFile3

Hadoop, 2015.

Image 5. HDFS web user interface Directory

C. MapReduce Processes

1. Logs Details

As shown in image 6 below, the log shows which files is being executed from which location. In our case, as shown in image 6, we are executing our file from HDFS location. The log shows the name of the running Job (job_145478739_8002), saved output task location, status of the reduce task execution, etc.

```

2017-02-08 22:54:32,890 [LocalJobRunner Map Task Executor #0] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.PigRecordReader - Current split being processed hdfs://localhost:9000/tmp/temp-1557744124/tmp2107943726/part-r-000000:0+134217728
2017-02-08 22:54:32,956 [LocalJobRunner Map Task Executor #0] INFO org.apache.hadoop.mapred.MapTask - (EQUATOR) 0 kvi 26214396(104857584)
2017-02-08 22:54:32,956 [LocalJobRunner Map Task Executor #0] INFO org.apache.hadoop.mapred.MapTask - mapreduce.task.io.sort.mb: 100
2017-02-08 22:54:32,956 [LocalJobRunner Map Task Executor #0] INFO org.apache.hadoop.mapred.MapTask - soft limit at 83886080
2017-02-08 22:54:32,956 [LocalJobRunner Map Task Executor #0] INFO org.apache.hadoop.mapred.MapTask - bufstart = 0; bufvoid = 104857600
2017-02-08 22:54:32,956 [LocalJobRunner Map Task Executor #0] INFO org.apache.hadoop.mapred.MapTask - kvstart = 26214396; length = 6553600
2017-02-08 22:54:32,972 [LocalJobRunner Map Task Executor #0] INFO org.apache.hadoop.mapred.MapTask - Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer
2017-02-08 22:54:32,985 [LocalJobRunner Map Task Executor #0] INFO org.apache.pig.impl.util.SpillableMemoryManager - Selected heap (Tenured Gen) of size 699072512 to monitor. collectionUsageThreshold = 489350752, usageThreshold = 489350752
2017-02-08 22:54:32,988 [LocalJobRunner Map Task Executor #0] WARN org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has already been initialized
2017-02-08 22:54:59,510 [pool-9-thread-1] WARN org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has already been initialized
2017-02-08 22:54:59,532 [pool-9-thread-1] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.PigMapReduce$Reduce - Aliases being processed per job phase (AliasName[line,of,set]): M: byte_count[6,13], time_data[4,12] C: byte_count[6,13], time_data[4,12] R: byte_count[6,13]
2017-02-08 22:55:04,164 [communication thread] INFO org.apache.hadoop.mapred.LocalJobRunner - reduce > reduce
2017-02-08 22:55:04,206 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - 98% complete
2017-02-08 22:55:04,206 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Running jobs are [job_local145478739_0002]
2017-02-08 22:55:05,155 [pool-9-thread-1] INFO org.apache.hadoop.mapred.Task - Task:attempt_local145478739_0002_r_000000_0 is done. And is in the process of committing
2017-02-08 22:55:05,168 [pool-9-thread-1] INFO org.apache.hadoop.mapred.LocalJobRunner - reduce > reduce
2017-02-08 22:55:05,168 [pool-9-thread-1] INFO org.apache.hadoop.mapred.Task - Task attempt_local145478739_0002_r_000000_0 is allowed to commit now
2017-02-08 22:55:05,165 [pool-9-thread-1] INFO org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter - Saved output of task 'attempt_local145478739_0002_r_000000_0' to hdfs://localhost:9000/tmp/temp-1557744124/tmp974198704/temporary/0/task_local145478739_0002_r_000000
2017-02-08 22:55:05,188 [pool-9-thread-1] INFO org.apache.hadoop.mapred.LocalJobRunner - reduce > reduce
2017-02-08 22:55:05,192 [pool-9-thread-1] INFO org.apache.hadoop.mapred.Task - Task 'attempt_local145478739_0002_r_000000_0' done.
2017-02-08 22:55:05,196 [pool-9-thread-1] INFO org.apache.hadoop.mapred.LocalJobRunner - Finishing task: attempt_local145478739_0002_r_000000_0
2017-02-08 22:55:05,197 [Thread-115] INFO org.apache.hadoop.mapred.LocalJobRunner - reduce task executor complete.
2017-02-08 22:55:07,713 [main] INFO org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2017-02-08 22:55:07,717 [main] INFO org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2017-02-08 22:55:07,718 [main] INFO org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2017-02-08 22:55:07,783 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - 100% complete
2017-02-08 22:55:07,804 [main] INFO org.apache.pig.tools.pigstats.mapreduce.SimplePigStats - Script Statistics:

```

Image 6. Logs while executing input file in Pig Grunt shell

2. Output log Report

As shown in image 7, the log reports:

- I. Successfully read 1891715 records
- II. Successfully stored records 1099799 files.

```

Input(s):
Successfully read 1891715 records (560849373 bytes) from: "/user/AccessLogAnalysis/WebLogs/AccessLogFile1"

Output(s):
Successfully stored 1099799 records (785716923 bytes) in: "hdfs://localhost:9000/tmp/temp-1557744124/tmp974198704"

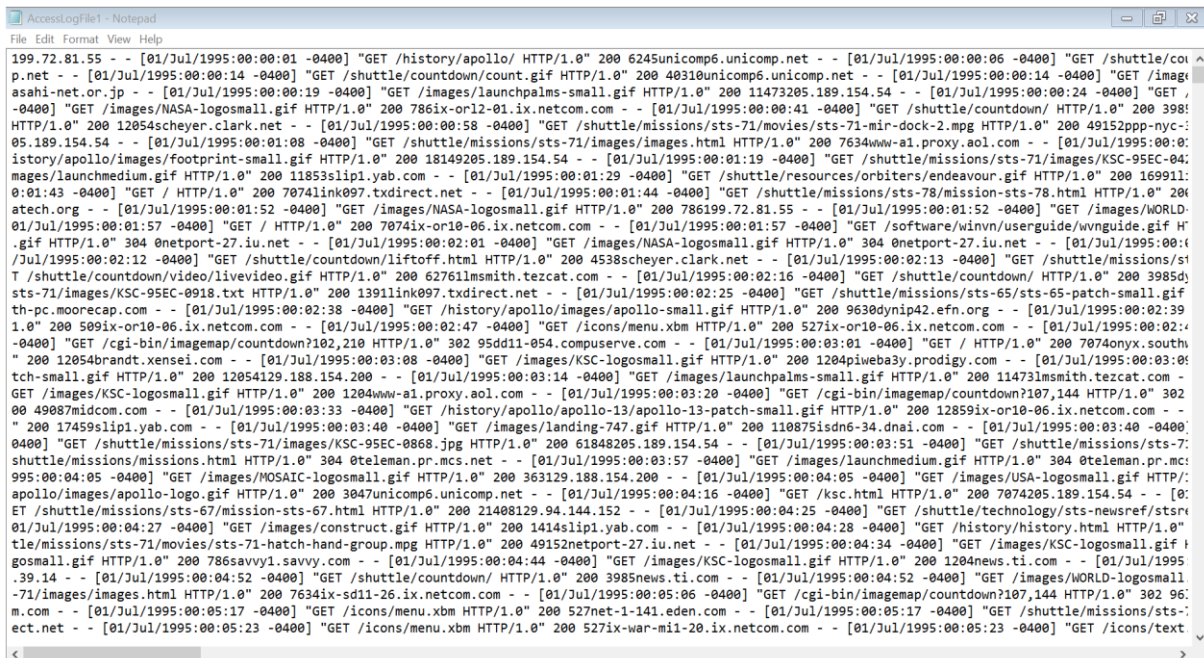
Counters:
Total records written : 1099799
Total bytes written : 785716923
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_local756095195_0001 ->      job_local145478739_0002,
job_local145478739_0002

```

Image 7. Output log report in Pig Grunt shell D.

D. Input log file sample sanpshot



```

AccessLogFile1 - Notepad
File Edit Format View Help
199.72.81.55 - - [01/Jul/1995:00:00:01 -0400] "GET /history/apollo/ HTTP/1.0" 200 6245unicomp6.unicomp.net - - [01/Jul/1995:00:00:06 -0400] "GET /shuttle/cou
p.net - - [01/Jul/1995:00:00:14 -0400] "GET /shuttle/countdown/count.gif HTTP/1.0" 200 40310unicomp6.unicomp.net - - [01/Jul/1995:00:00:14 -0400] "GET /imag
asahi-net.or.jp - - [01/Jul/1995:00:00:19 -0400] "GET /images/launchpalms-small.gif HTTP/1.0" 200 11473205.189.154.54 - - [01/Jul/1995:00:00:24 -0400] "GET
-0400] "GET /images/NASA-logosmall.gif HTTP/1.0" 200 786ix-or12-01.ix.netcom.com - - [01/Jul/1995:00:00:41 -0400] "GET /shuttle/countdown/ HTTP/1.0" 200 398:
HTTP/1.0" 200 12054scheyer.clark.net - - [01/Jul/1995:00:00:58 -0400] "GET /shuttle/missions/sts-71/movies/sts-71-mir-dock-2.mpg HTTP/1.0" 200 49152ppp-nyc-:
05.189.154.54 - - [01/Jul/1995:00:01:08 -0400] "GET /shuttle/missions/sts-71/images/images.html HTTP/1.0" 200 7634www-a1.proxy.aol.com - - [01/Jul/1995:00:0
istory/apollo/images/footprint-small.gif HTTP/1.0" 200 18149205.189.154.54 - - [01/Jul/1995:00:01:19 -0400] "GET /shuttle/missions/sts-71/images/KSC-95EC-04:
images/launchmedium.gif HTTP/1.0" 200 11853slipi.yab.com - - [01/Jul/1995:00:01:29 -0400] "GET /shuttle/resources/orbiters/endeavour.gif HTTP/1.0" 200 169911:
0:01:43 -0400] "GET / HTTP/1.0" 200 7074link097.txdirect.net - - [01/Jul/1995:00:01:44 -0400] "GET /shuttle/missions/sts-78/mission-sts-78.html HTTP/1.0" 200
atech.org - - [01/Jul/1995:00:01:52 -0400] "GET /images/NASA-logosmall.gif HTTP/1.0" 200 786199.72.81.55 - - [01/Jul/1995:00:01:52 -0400] "GET /images/WORL
01/Jul/1995:00:01:57 -0400] "GET / HTTP/1.0" 200 7074ix-or10-06.ix.netcom.com - - [01/Jul/1995:00:01:57 -0400] "GET /software/winwnv/userguide/wvnguide.gif H
.gif HTTP/1.0" 304 0netport-27.iu.net - - [01/Jul/1995:00:02:01 -0400] "GET /images/NASA-logosmall.gif HTTP/1.0" 304 0netport-27.iu.net - - [01/Jul/1995:00:0
/Jul/1995:00:02:12 -0400] "GET /shuttle/countdown/liftoff.html HTTP/1.0" 200 4538scheyer.clark.net - - [01/Jul/1995:00:02:13 -0400] "GET /shuttle/missions/st
T /shuttle/countdown/video/livevideo.gif HTTP/1.0" 200 627611msmith.tezcat.com - - [01/Jul/1995:00:02:16 -0400] "GET /shuttle/countdown/ HTTP/1.0" 200 39850:
sts-71/images/KSC-95EC-0918.txt HTTP/1.0" 200 1391link097.txdirect.net - - [01/Jul/1995:00:02:25 -0400] "GET /shuttle/missions/sts-65/sts-65-patch-small.gif
th-pc.moorecap.com - - [01/Jul/1995:00:02:38 -0400] "GET /history/apollo/images/apollo-small.gif HTTP/1.0" 200 9630dynip42.efn.org - - [01/Jul/1995:00:02:39
1.0" 200 5091ix-or10-06.ix.netcom.com - - [01/Jul/1995:00:02:47 -0400] "GET /icons/menu.xbm HTTP/1.0" 200 527ix-or10-06.ix.netcom.com - - [01/Jul/1995:00:02:4
-0400] "GET /cgi-bin/imagemap/countdown?102,210 HTTP/1.0" 302 95dd11-054.compuserve.com - - [01/Jul/1995:00:03:01 -0400] "GET / HTTP/1.0" 200 70740nyx.south
" 200 12054brandt.xensei.com - - [01/Jul/1995:00:03:08 -0400] "GET /images/KSC-logosmall.gif HTTP/1.0" 200 1204piweba3y.prodigy.com - - [01/Jul/1995:00:03:08:
tch-small.gif HTTP/1.0" 200 12054129.188.154.200 - - [01/Jul/1995:00:03:14 -0400] "GET /images/launchpalms-small.gif HTTP/1.0" 200 114731msmith.tezcat.com -
GET /images/KSC-logosmall.gif HTTP/1.0" 200 1204www-a1.proxy.aol.com - - [01/Jul/1995:00:03:20 -0400] "GET /cgi-bin/imagemap/countdown?107,144 HTTP/1.0" 302
00 49087midcom.com - - [01/Jul/1995:00:03:33 -0400] "GET /history/apollo/apollo-13/apollo-13-patch-small.gif HTTP/1.0" 200 12859ix-or10-06.ix.netcom.com - -
" 200 17459slipi.yab.com - - [01/Jul/1995:00:03:40 -0400] "GET /images/landing-747.gif HTTP/1.0" 200 110875isdnt6-34.dnai.com - - [01/Jul/1995:00:03:40 -0400]
0400] "GET /shuttle/missions/sts-71/images/KSC-95EC-0868.jpg HTTP/1.0" 200 61848205.189.154.54 - - [01/Jul/1995:00:03:51 -0400] "GET /shuttle/missions/sts-7:
shuttle/missions/missions.html HTTP/1.0" 304 0teleman.pr.mcs.net - - [01/Jul/1995:00:03:57 -0400] "GET /images/launchmedium.gif HTTP/1.0" 304 0teleman.pr.mc:
995:00:04:05 -0400] "GET /images/MOSAIC-logosmall.gif HTTP/1.0" 200 363129.188.154.200 - - [01/Jul/1995:00:04:05 -0400] "GET /images/USA-logosmall.gif HTTP/:
apollo/images/apollo-logo.gif HTTP/1.0" 200 3047unicomp6.unicomp.net - - [01/Jul/1995:00:04:16 -0400] "GET /ksc.html HTTP/1.0" 200 7074205.189.154.54 - - [0:
ET /shuttle/missions/sts-67/mission-sts-67.html HTTP/1.0" 200 21408129.94.144.152 - - [01/Jul/1995:00:04:25 -0400] "GET /shuttle/technology/sts-newsref/stsr
01/Jul/1995:00:04:27 -0400] "GET /images/construct.gif HTTP/1.0" 200 1414slipi.yab.com - - [01/Jul/1995:00:04:28 -0400] "GET /history/history.html HTTP/1.0"
tle/missions/sts-71/movies/sts-71-hatch-hand-group.mpg HTTP/1.0" 200 49152netport-27.iu.net - - [01/Jul/1995:00:04:34 -0400] "GET /images/KSC-logosmall.gif
gogsmall.gif HTTP/1.0" 200 786savvy1.savvy.com - - [01/Jul/1995:00:04:44 -0400] "GET /images/KSC-logosmall.gif HTTP/1.0" 200 1204news.ti.com - - [01/Jul/1995
.39.14 - - [01/Jul/1995:00:04:52 -0400] "GET /shuttle/countdown/ HTTP/1.0" 200 3985news.ti.com - - [01/Jul/1995:00:04:52 -0400] "GET /images/WORLD-logosmall
-71/images/images.html HTTP/1.0" 200 7634ix-sd11-26.ix.netcom.com - - [01/Jul/1995:00:05:06 -0400] "GET /cgi-bin/imagemap/countdown?107,144 HTTP/1.0" 302 96:
m.com - - [01/Jul/1995:00:05:17 -0400] "GET /icons/menu.xbm HTTP/1.0" 200 527net-1-141.eden.com - - [01/Jul/1995:00:05:17 -0400] "GET /shuttle/missions/sts-:
ect.net - - [01/Jul/1995:00:05:23 -0400] "GET /icons/menu.xbm HTTP/1.0" 200 527ix-war-m11-20.ix.netcom.com - - [01/Jul/1995:00:05:23 -0400] "GET /icons/text

```



```

AccessLogFile1 - Notepad
File Edit Format View Help

ect.net - - [01/Jul/1995:00:05:23 -0400] "GET /icons/menu.xbm HTTP/1.0" 200 527ix-war-mil-20.ix.netcom.com - - [01/Jul/1995:00:05:23 -0400] "GET /icons/text
52 - - [01/Jul/1995:00:05:35 -0400] "GET /images/KSC-logosmall.gif HTTP/1.0" 200 1204port2.electrotex.com - - [01/Jul/1995:00:05:37 -0400] "GET /shuttle/coun
.ab.ca - - [01/Jul/1995:00:05:43 -0400] "GET /shuttle/missions/sts-71/images/KSC-95EC-0876.gif HTTP/1.0" 200 51398ix-war-mil-20.ix.netcom.com - - [01/Jul/1995:00:05:51
undtown/ HTTP/1.0" 200 3985slip4068.sirius.com - - [01/Jul/1995:00:05:55 -0400] "GET /ksc.html HTTP/1.0" 200 7074pm4_23.digital.net - - [01/Jul/1995:00:05:55
:06:02 -0400] "GET /history/history.html HTTP/1.0" 200 1502news.ti.com - - [01/Jul/1995:00:06:03 -0400] "GET /shuttle/missions/sts-71/sts-71-patch-small.gif
/1995:00:06:11 -0400] "GET /images/faq.gif HTTP/1.0" 200 263burger.letters.com - - [01/Jul/1995:00:06:12 -0400] "GET /shuttle/countdown/liftoff.html HTTP/1.0"
ssions/missions.html HTTP/1.0" 200 8677gater4.sematech.org - - [01/Jul/1995:00:06:26 -0400] "GET /cgi-bin/imagemap/countdown/88,208 HTTP/1.0" 302 95remote27.
mages/kscmap-tiny.gif HTTP/1.0" 200 2537ix-wbg-va2-26.ix.netcom.com - - [01/Jul/1995:00:06:33 -0400] "GET /history/apollo/apollo11/images/ HTTP/1.0" 200 34
03985www-a1.proxy.aol.com - - [01/Jul/1995:00:06:39 -0400] "GET /images/whatsnew.gif HTTP/1.0" 200 651slip132.indirect.com - - [01/Jul/1995:00:06:41 -0400]
pc.com - - [01/Jul/1995:00:06:49 -0400] "GET /images/mercury-logo.gif HTTP/1.0" 200 6588brandt.xensei.com - - [01/Jul/1995:00:06:54 -0400] "GET /shuttle/mis:
- - [01/Jul/1995:00:07:04 -0400] "GET /images/WORLD-logosmall.gif HTTP/1.0" 200 669p06.eznets.canton.oh.us - - [01/Jul/1995:00:07:08 -0400] "GET /shuttle/mis:
P/1.0" 200 52491ppp236.iadfw.net - - [01/Jul/1995:00:07:20 -0400] "GET /software/winvn/winvn.html HTTP/1.0" 200 9867ad12-851.compuserve.com - - [01/Jul/1995
ruct.gif HTTP/1.0" 200 1414asp.erinet.com - - [01/Jul/1995:00:07:27 -0400] "GET /software/winvn/bluemarb.gif HTTP/1.0" 200 4441ppp236.iadfw.net - - [01/Jul/1995
/htbin/cdt_main.pl HTTP/1.0" 200 3214ix-sdl1-26.ix.netcom.com - - [01/Jul/1995:00:07:34 -0400] "GET /shuttle/countdown/count.gif HTTP/1.0" 200 40318ix-sdl1-
/1.0" 200 363ppp4.sunrem.com - - [01/Jul/1995:00:07:37 -0400] "GET /shuttle/countdown/ HTTP/1.0" 200 3985199.166.39.14 - - [01/Jul/1995:00:07:37 -0400]
ppp4.sunrem.com - - [01/Jul/1995:00:07:43 -0400] "GET /images/KSC-logosmall.gif HTTP/1.0" 200 1204pm4_23.digital.net - - [01/Jul/1995:00:07:43 -0400] "GET /:
0:07:50 -0400] "GET /shuttle/missions/sts-71/mission-sts-71.html HTTP/1.0" 200 12040asp.erinet.com - - [01/Jul/1995:00:07:52 -0400] "GET /images/WORLD-logos
" 200 1713ppp-mia-53.shadow.net - - [01/Jul/1995:00:08:03 -0400] "GET /images/KSC-logosmall.gif HTTP/1.0" 200 1204dnet018.sat.texas.net - - [01/Jul/1995:00:08:03
ntdown/ HTTP/1.0" 200 3985brandt.xensei.com - - [01/Jul/1995:00:08:13 -0400] "GET /images/NASA-logosmall.gif HTTP/1.0" 200 3040net-1-141.eden.com - - [01/Jul/19
1.0" 200 786tyu0.tyrell.net - - [01/Jul/1995:00:08:27 -0400] "GET /images/MOSAIIC-logosmall.gif HTTP/1.0" 200 363199.120.91.6 - - [01/Jul/1995:00:08:28 -0400
"GET /images/NASA-logosmall.gif HTTP/1.0" 200 786ix-sdl1-26.ix.netcom.com - - [01/Jul/1995:00:08:33 -0400] "GET /shuttle/missions/sts-71/images/images.html HT
/1995:00:08:52 -0400] "GET /shuttle/countdown/count.gif HTTP/1.0" 200 40310ix-dfw13-20.ix.netcom.com - - [01/Jul/1995:00:08:52 -0400] "GET /images/NASA-
IC-logosmall.gif HTTP/1.0" 200 363slip1.yab.com - - [01/Jul/1995:00:09:02 -0400] "GET /history/skylab/skylab-4.html HTTP/1.0" 200 1393pipe6.nyc.pipeline.com
HTTP/1.0" 200 20484166.79.67.111 - - [01/Jul/1995:00:09:17 -0400] "GET /shuttle/countdown/count.gif HTTP/1.0" 200 40310wwwproxy.info.au - - [01/Jul/1995:00:09:17
ages/KSC-logosmall.gif HTTP/1.0" 304 0pm110.spectra.net - - [01/Jul/1995:00:09:30 -0400] "GET /shuttle/missions/sts-71/mission-sts-71.html HTTP/1.0" 200 1204
304 0pmwwwproxy.info.au - - [01/Jul/1995:00:09:39 -0400] "GET /images/USA-logosmall.gif HTTP/1.0" 304 0pm110.spectra.net - - [01/Jul/1995:00:09:40 -0400] "GE
995:00:09:54 -0400] "GET /shuttle/countdown/ HTTP/1.0" 200 3985slip1.kias.com - - [01/Jul/1995:00:09:55 -0400] "GET /htbin/wais.pl HTTP/1.0" 200 308pm110.sp
/images/blank.xbm HTTP/1.0" 200 509129.188.154.200 - - [01/Jul/1995:00:10:07 -0400] "GET /icons/menu.xbm HTTP/1.0" 200 527129.188.154.200 - - [01/Jul/1995:00:
-0400] "GET /ksc.html HTTP/1.0" 200 7074129.188.154.200 - - [01/Jul/1995:00:10:23 -0400] "GET /shuttle/missions/sts-71/ HTTP/1.0" 200 3373ix-dfw13-20.ix.netc
:31 -0400] "GET /shuttle/missions/51-1/images/86HC159.GIF HTTP/1.0" 200 78295www-d3.proxy.aol.com - - [01/Jul/1995:00:10:32 -0400] "GET /shuttle/missions/st:
nfo.html HTTP/1.0" 200 1387ppp-mia-53.shadow.net - - [01/Jul/1995:00:10:40 -0400] "GET /shuttle/missions/sts-71/images/images.html HTTP/1.0" 200 7634brandt.
HTTP/1.0" 200 2244taiki4.envi.osakafu-u.ac.jp - - [01/Jul/1995:00:11:03 -0400] "GET /images/NASA-logosmall.gif HTTP/1.0" 200 786dynip38.efn.org - - [01/Jul/1995
.bnrc.ca - - [01/Jul/1995:00:11:14 -0400] "GET /htbin/cdt_main.pl HTTP/1.0" 200 3214129.188.154.200 - - [01/Jul/1995:00:11:15 -0400] "GET /shuttle/missions/
r.html HTTP/1.0" 200 3723slip1.kias.com - - [01/Jul/1995:00:11:24 -0400] "GET /shuttle/countdown/count.gif HTTP/1.0" 200 40310telemar.pr.mcs.net - - [01/Jul/1995:00:11:24

```

```

AccessLogFile1 - Notepad
File Edit Format View Help

t.texas.net - - [01/Jul/1995:00:11:30 -0400] "GET /icons/blank.xbm HTTP/1.0" 200 509pm28.sonic.net - - [01/Jul/1995:00:11:31 -0400] "GET /shuttle/countdown/c
efn.org - - [01/Jul/1995:00:11:36 -0400] "GET /software/winvn/release.txt HTTP/1.0" 200 23052pm28.sonic.net - - [01/Jul/1995:00:11:36 -0400] "GET /images/KSC
01/Jul/1995:00:11:42 -0400] "GET /shuttle/countdown/count.html HTTP/1.0" 200 72321ix-dfw13-20.ix.netcom.com - - [01/Jul/1995:00:11:44 -0400] "GET /cgi-bin/ir
ttle/missions/sts-74/mission-sts-74.html HTTP/1.0" 200 12054ppp3_136.bekoame.or.jp - - [01/Jul/1995:00:11:49 -0400] "GET /images/USA-logosmall.gif HTTP:
/shuttle/missions/sts-74/mission-sts-74.html HTTP/1.0" 200 3707telemar.pr.mcs.net - - [01/Jul/1995:00:11:58 -0400] "GET /images/KSC-94EC-412-small.gif HTTP:
HTTP/1.0" 200 29634slip1.kias.com - - [01/Jul/1995:00:12:11 -0400] "GET /shuttle/missions/sts-71/images/images.html HTTP/1.0" 200 7634www-b2.proxy.aol.com - - [01/Jul/1995:00:12:20
:20 -0400] "GET /images/WORLD-logosmall.gif HTTP/1.0" 200 669telemar.pr.mcs.net - - [01/Jul/1995:00:12:20 -0400] "GET /cgi-bin/imagemap/countdown/321,276 H
rim.or.jp - - [01/Jul/1995:00:12:26 -0400] "GET /images/USA-logosmall.gif HTTP/1.0" 200 234dnet018.sat.texas.net - - [01/Jul/1995:00:12:28 -0400] "GET /histo
.gif HTTP/1.0" 200 31631blv-pm0-ip28.halcyon.com - - [01/Jul/1995:00:12:36 -0400] "GET /history/history.html HTTP/1.0" 304 0205.157.131.144 - - [01/Jul/1995
/1995:00:12:45 -0400] "GET /shuttle/missions/sts-71/images/KSC-95EC-0917.gif HTTP/1.0" 200 30995dd11-054.compuserve.com - - [01/Jul/1995:00:12:49 -0400] "GE
32www-d3.proxy.aol.com - - [01/Jul/1995:00:13:02 -0400] "GET /shuttle/missions/sts-71/images/KSC-95EC-0918.jpg HTTP/1.0" 200 4688dnet018.sat.texas.net - - [
- [01/Jul/1995:00:13:11 -0400] "GET /shuttle/technology/sts-newsref/sts-1cc.html HTTP/1.0" 200 32252uconnvm.uconn.edu - - [01/Jul/1995:00:13:13 -0400] "GET
-press-kit.txt HTTP/1.0" 200 78588kenmarks-ppp.clark.net - - [01/Jul/1995:00:13:28 -0400] "GET /software/winvn/winvn.html HTTP/1.0" 200 9867ppp4.sunrem.com - - [01/Jul/1995:00:13:34
-0400] "GET /images/cor
skylab/skylab-3.html HTTP/1.0" 200 1424waters-gw.starway.net.au - - [01/Jul/1995:00:13:47 -0400] "GET /shuttle/missions/51-1/movies/ HTTP/1.0" 200 3720ahu5:
/persons/astronauts/i-to-1/ousmaJR.txt HTTP/1.0" 200 404 -waters-gw.starway.net.au - - [01/Jul/1995:00:14:18 -0400] "GET /shuttle/missions/51-1/docs/ HTTP/1.0"
00:14:26 -0400] "GET /images/KSC-logosmall.gif HTTP/1.0" 200 12044fsgp86.slip.net - - [01/Jul/1995:00:14:27 -0400] "GET /shuttle/countdown/count.gif HTTP/1.0"
00:15:55 -0400] "GET /shuttle/technology/sts-newsref/sts-1cc.html HTTP/1.0" 200 32252204.212.254.71 - - [01/Jul/1995:00:15:58 -0400] "GET /shuttle/countdown
.html HTTP/1.0" 200 1484156.151.176.30 - - [01/Jul/1995:00:16:04 -0400] "GET /ksc.html HTTP/1.0" 200 7074slip-5.io.com - - [01/Jul/1995:00:16:05 -0400] "GET
/shuttle/countdown/count.gif HTTP/1.0" 200 40310129.188.154.200 - - [01/Jul/1995:00:16:13 -0400] "GET /shuttle/missions/sts-71/movies/movies.html HTTP/1.0"
:0400] "GET /images/launch-logo.gif HTTP/1.0" 200 1713svasu.extern.ucsd.edu - - [01/Jul/1995:00:16:17 -0400] "GET /icons/text.xbm HTTP/1.0" 200 527annex-p2.s
400] "GET /shuttle/missions/sts-71/movies/sts-71-mir-dock-2.mpg HTTP/1.0" 200 65536ix-wbg-va2-26.ix.netcom.com - - [01/Jul/1995:00:16:30 -0400] "GET /history
995:00:16:37 -0400] "GET /images/NASA-logosmall.gif HTTP/1.0" 304 0156.151.176.45 - - [01/Jul/1995:00:16:37 -0400] "GET /images/ksclogo-medium.gif HTTP/1.0"
Jul/1995:00:16:43 -0400] "GET /images/launch-logo.gif HTTP/1.0" 304 0svasu.extern.ucsd.edu - - [01/Jul/1995:00:16:43 -0400] "GET /history/apollo/apollo.html
u-dialup-1005.cit.cornell.edu - - [01/Jul/1995:00:16:47 -0400] "GET /software/winvn/bluemarb.gif HTTP/1.0" 200 4441cu-dialup-1005.cit.cornell.edu - - [01/Jul
T /shuttle/countdown/ HTTP/1.0" 200 3985acs4.acs.ualgary.ca - - [01/Jul/1995:00:16:53 -0400] "GET /shuttle/missions/missions.html HTTP/1.0" 200 8677ppp111.c
ovies/astronauts.mpg HTTP/1.0" 200 106496cu-dialup-1005.cit.cornell.edu - - [01/Jul/1995:00:17:03 -0400] "GET /images/USA-logosmall.gif HTTP/1.0" 200 234129.
A-logosmall.gif HTTP/1.0" 304 0ppps5.earthlight.co.nz - - [01/Jul/1995:00:17:13 -0400] "GET /shuttle/missions/sts-70/mission-sts-70.html HTTP/1.0" 200 134692u
/Jul/1995:00:17:22 -0400] "GET /images/launch-logo.gif HTTP/1.0" 200 1713news.ti.com - - [01/Jul/1995:00:17:23 -0400] "GET /icons/blank.xbm HTTP/1.0" 200 508
tcom.com - - [01/Jul/1995:00:17:29 -0400] "GET /shuttle/missions/sts-68/images/ksc.gif HTTP/1.0" 200 152676detroit.freenet.org - - [01/Jul/1995:00:17:29 -04
1.0" 200 2261129.188.154.200 - - [01/Jul/1995:00:17:39 -0400] "GET /htbin/cdt_main.pl HTTP/1.0" 200 3214leet.cit.com - - [01/Jul/1995:00:17:42 -0400] "GET /:
1-info.html HTTP/1.0" 200 1440kuts5p06.cc.ukans.edu - - [01/Jul/1995:00:17:55 -0400] "GET /shuttle/missions/sts-77/mission-sts-77.html HTTP/1.0" 200 3071tra:
ml HTTP/1.0" 200 1395kuts5p06.cc.ukans.edu - - [01/Jul/1995:00:18:08 -0400] "GET /shuttle/missions/sts-76/mission-sts-76.html HTTP/1.0" 200 3109p-shining.no
o.u.c - - [01/Jul/1995:00:18:13 -0400] "GET /history/apollo/images/footprint-logo.gif HTTP/1.0" 200 4209slip-5.io.com - - [01/Jul/1995:00:18:15 -0400] "GET /:
/1.0" 200 509www-a1.proxy.aol.com - - [01/Jul/1995:00:18:27 -0400] "GET /facilities/vab.html HTTP/1.0" 200 4045telemar.pr.mcs.net - - [01/Jul/1995:00:18:28

```



```
AccessLogFile1 - Notepad
File Edit Format View Help
o.uk - - [01/Jul/1995:00:18:13 -0400] "GET /history/apollo/images/footprint-logo.gif HTTP/1.0" 200 4209slip-5.io.com - - [01/Jul/1995:00:18:15 -0400] "GET /:
/1.0" 200 509www-a1.proxy.aol.com - - [01/Jul/1995:00:18:27 -0400] "GET /facilities/vab.html HTTP/1.0" 200 4045teleman.pr.mcs.net - - [01/Jul/1995:00:18:28 :
m - - [01/Jul/1995:00:18:33 -0400] "GET /history/apollo/as-201/sounds/ HTTP/1.0" 200 3720tgate2.bnr.ca - - [01/Jul/1995:00:18:34 -0400] "GET /shuttle/techn
TTP/1.0" 200 786whlane.cts.com - - [01/Jul/1995:00:18:41 -0400] "GET /images/KSC-logosmall.gif HTTP/1.0" 200 1204slip-5.io.com - - [01/Jul/1995:00:18:41 -04
ages/launch-logo.gif HTTP/1.0" 200 1713ppps5.earthlight.co.nz - - [01/Jul/1995:00:18:46 -0400] "GET /shuttle/missions/sts-70/images/ HTTP/1.0" 200 966cu-dial
undtwn/ HTTP/1.0" 200 3985h-shining.norfolk.infi.net - - [01/Jul/1995:00:19:01 -0400] "GET /shuttle/resources/orbiters/atlantis-logo.gif HTTP/1.0" 200 4179r
ges/images.html HTTP/1.0" 200 7634dialup61.afn.org - - [01/Jul/1995:00:19:04 -0400] "GET /shuttle/missions/sts-71/movies/movies.html HTTP/1.0" 304 0whlane.ct
95:00:19:11 -0400] "GET /cgi-bin/imagemap/countdown?370,276 HTTP/1.0" 302 68traitor.demon.co.uk - - [01/Jul/1995:00:19:12 -0400] "GET /history/apollo/images,
"GET /history/apollo/apollo-13/apollo-13-info.html HTTP/1.0" 200 1583acs4.acs.ualgary.ca - - [01/Jul/1995:00:19:17 -0400] "GET /shuttle/missions/sts-71/new
TP/1.0" 200 48519tgate2.bnr.ca - - [01/Jul/1995:00:19:26 -0400] "GET /shuttle/technology/images/et-intertank1-small.gif HTTP/1.0" 200 79791ppp160.iadfw.n
.com - - [01/Jul/1995:00:19:35 -0400] "GET /shuttle/countdown/video/livevideo.jpeg HTTP/1.0" 200 47296204.248.98.63 - - [01/Jul/1995:00:19:36 -0400] "GET /c/
off.html HTTP/1.0" 200 4538ix-dfw11-01.ix.netcom.com - - [01/Jul/1995:00:19:41 -0400] "GET /shuttle/missions/sts-71/mission-sts-71.html HTTP/1.0" 200 12040t
ttle/missions/sts-71/sts-71-patch-small.gif HTTP/1.0" 200 12054dal31.pic.net - - [01/Jul/1995:00:19:45 -0400] "GET /images/NASA-logosmall.gif HTTP/1.0" 200
13teleman.pr.mcs.net - - [01/Jul/1995:00:19:56 -0400] "GET /history/apollo/apollo-13/apollo-13-patch-small.gif HTTP/1.0" 200 12859tempmano.netheaven.com - -
Jul/1995:00:20:05 -0400] "GET /shuttle/resources/orbiters/orbiters-logo.gif HTTP/1.0" 200 1932202.70.0.6 - - [01/Jul/1995:00:20:05 -0400] "GET /shuttle/miss
202.70.0.6 - - [01/Jul/1995:00:20:12 -0400] "GET /shuttle/missions/sts-63/sounds/ HTTP/1.0" 200 378slip-28-14.ots.utexas.edu - - [01/Jul/1995:00:20:12 -0400]
/1.0" 200 31631slip-5.io.com - - [01/Jul/1995:00:20:17 -0400] "GET /history/apollo/as-201/ HTTP/1.0" 200 1699202.70.0.6 - - [01/Jul/1995:00:20:18 -0400] "GE
/shuttle/countdown/liftoff.html HTTP/1.0" 200 4538p43.infinet.com - - [01/Jul/1995:00:20:26 -0400] "GET /shuttle/technology/sts-newsref/stsref-toc.html HTTP/
:01/Jul/1995:00:20:29 -0400] "GET /images/NASA-logosmall.gif HTTP/1.0" 200 786134.20.175.12 - - [01/Jul/1995:00:20:30 -0400] "GET /images/NASA-logosmall.g
nfi.net - - [01/Jul/1995:00:20:35 -0400] "GET /shuttle/missions/sts-70/mission-sts-70.html HTTP/1.0" 200 13469www-a1.proxy.aol.com - - [01/Jul/1995:00:20:36
1995:00:20:44 -0400] "GET /shuttle/technology/images/et1.jpg HTTP/1.0" 200 144114202.70.0.6 - - [01/Jul/1995:00:20:45 -0400] "GET /shuttle/missions/sts-63/r
-0918.jpg HTTP/1.0" 200 46888slip-5.io.com - - [01/Jul/1995:00:20:55 -0400] "GET /icons/sound.xbm HTTP/1.0" 200 530ix-phx5-17.ix.netcom.com - - [01/Jul/1995
s/sound.xbm HTTP/1.0" 200 530ix-sd9-18.ix.netcom.com - - [01/Jul/1995:00:21:01 -0400] "GET /images/MOSAIC-logosmall.gif HTTP/1.0" 200 363ix-tam1-26.ix.netco
/sts-71/images/images.html HTTP/1.0" 200 7634134.20.175.12 - - [01/Jul/1995:00:21:11 -0400] "GET /cgi-bin/imagemap/countdown?110,112 HTTP/1.0" 302 111ix-ftw
11.gif HTTP/1.0" 200 1204blv-pm2-ip16.halcyon.com - - [01/Jul/1995:00:21:23 -0400] "GET /shuttle/missions/sts-68/sts-68-patch-small.gif HTTP/1.0" 200 174591:
72204.97.234.46 - - [01/Jul/1995:00:21:38 -0400] "GET / HTTP/1.0" 200 7074midcom.com - - [01/Jul/1995:00:21:39 -0400] "GET /history/apollo/apollo-13/sounds/
363annex12-36.dial.umd.edu - - [01/Jul/1995:00:21:45 -0400] "GET /shuttle/missions/sts-71/images/KSC-95EC-0589.jpg HTTP/1.0" 200 64427ix-dfw11-01.ix.netcom.
204.97.234.46 - - [01/Jul/1995:00:21:50 -0400] "GET /images/USA-logosmall.gif HTTP/1.0" 200 234ix-phx5-17.ix.netcom.com - - [01/Jul/1995:00:21:50 -0400] "GE
00] "GET /images/ksclogosmall.gif HTTP/1.0" 200 3635ix-dfw11-01.ix.netcom.com - - [01/Jul/1995:00:21:54 -0400] "GET /shuttle/resources/orbiters/orbiters-log
ts-71/sts-71-patch-small.gif HTTP/1.0" 200 12054dal31.pic.net - - [01/Jul/1995:00:21:59 -0400] "GET /shuttle/resources/orbiters/atlantis-logo.gif HTTP/1.0"
0] "GET /images/WORLD-logosmall.gif HTTP/1.0" 200 669kuts5p06.cc.ukans.edu - - [01/Jul/1995:00:22:08 -0400] "GET /shuttle/missions/sts-71/mission-sts-71.htm
HTTP/1.0" 200 40310sam-slip-16.neosoft.com - - [01/Jul/1995:00:22:13 -0400] "GET /images/NASA-logosmall.gif HTTP/1.0" 200 786sam-slip-16.neosoft.com - - [01
,com - - [01/Jul/1995:00:22:22 -0400] "GET /shuttle/missions/sts-71/images/KSC-95EC-0912.gif HTTP/1.0" 200 48305ix-tam1-26.ix.netcom.com - - [01/Jul/1995:00
85ix-dfw11-01.ix.netcom.com - - [01/Jul/1995:00:22:29 -0400] "GET /shuttle/missions/100th.html HTTP/1.0" 200 32303chi067.wva.com - - [01/Jul/1995:00:22:31 -
00:22:39 -0400] "GET /shuttle/missions/sts-71/images/KSC-95EC-0443.gif HTTP/1.0" 200 64910dnet018.sat.texas.net - - [01/Jul/1995:00:22:42 -0400] "GET /histo
hp.com - - [01/Jul/1995:00:22:57 -0400] "GET /images/ksclogo-medium.gif HTTP/1.0" 200 0rcr14.crl.com - - [01/Jul/1995:00:22:58 -0400] "GET /shuttle/missions/
- - [01/Jul/1995:00:23:01 -0400] "GET /shuttle/countdown/count.gif HTTP/1.0" 200 40310picard.microsys.net - - [01/Jul/1995:00:23:02 -0400] "GET /shuttle/mis
/shuttle/countdown/ HTTP/1.0" 304 0palona1.cns.hp.com - - [01/Jul/1995:00:23:09 -0400] "GET /images/MOSAIC-logosmall.gif HTTP/1.0" 304 0sartre.execpc.com -
o/images/footprint-small.gif HTTP/1.0" 200 18149palona1.cns.hp.com - - [01/Jul/1995:00:23:19 -0400] "GET /images/USA-logosmall.gif HTTP/1.0" 304 0rlnport2.c
/ HTTP/1.0" 200 3985spalona1.cns.hp.com - - [01/Jul/1995:00:23:27 -0400] "GET /facts/facts.html HTTP/1.0" 200 4717chi067.wva.com - - [01/Jul/1995:00:23:27 -0
666clab040.ins.gu.edu.au - - [01/Jul/1995:00:23:34 -0400] "GET /cgi-bin/imagemap/countdown?104,169 HTTP/1.0" 302 110gclab040.ins.gu.edu.au - - [01/Jul/1995:0
6149.171.160.183 - - [01/Jul/1995:00:23:46 -0400] "GET /shuttle/missions/sts-71/sts-71-patch-small.gif HTTP/1.0" 200 12054ix-ftw-tx1-21.ix.netcom.com - - [0
y2.indy.net - - [01/Jul/1995:00:24:01 -0400] "GET /images/ksclogo-medium.gif HTTP/1.0" 200 5866gclab040.ins.gu.edu.au - - [01/Jul/1995:00:24:01 -0400] "GE
00 40310indy2.indy.net - - [01/Jul/1995:00:24:09 -0400] "GET /images/MOSAIC-logosmall.gif HTTP/1.0" 200 363indy2.indy.net - - [01/Jul/1995:00:24:10 -0400] "C
x.netcom.com - - [01/Jul/1995:00:24:17 -0400] "GET /images/kscmpap-small.gif HTTP/1.0" 200 39017149.171.160.182 - - [01/Jul/1995:00:24:18 -0400] "GET /shuttli
00:24:25 -0400] "GET /shuttle/missions/missions.html HTTP/1.0" 200 8677b1v-pm2-ip16.halcyon.com - - [01/Jul/1995:00:24:27 -0400] "GET /images/launchmedium.g
p.com - - [01/Jul/1995:00:24:39 -0400] "GET /images/WORLD-logosmall.gif HTTP/1.0" 304 0alyssa.prodigy.com - - [01/Jul/1995:00:24:44 -0400] "GET /shuttle/mis
/Jul/1995:00:24:51 -0400] "GET /shuttle/missions/sts-71/images/images.html HTTP/1.0" 200 7634blv-pm2-ip16.halcyon.com - - [01/Jul/1995:00:24:51 -0400] "GE
"GET /shuttle/missions/sts-71/sts-71-patch-small.gif HTTP/1.0" 200 12054indy2.indy.net - - [01/Jul/1995:00:25:02 -0400] "GET /shuttle/missions/sts-71/mission
huttle/missions/sts-71/movies/sts-71-tcdt-crew-walkout.mpg HTTP/1.0" 200 49152deimos.jp1.arizona.edu - - [01/Jul/1995:00:25:10 -0400] "GET /shuttle/missions,
1.0" 200 1713sartre.execpc.com - - [01/Jul/1995:00:25:20 -0400] "GET /shuttle/countdown/ HTTP/1.0" 304 0ix-phx5-17.ix.netcom.com - - [01/Jul/1995:00:25:21 -
/1.0" 200 25814boing.dgsys.com - - [01/Jul/1995:00:25:31 -0400] "GET /shuttle/missions/sts-67/sts-67-patch-small.gif HTTP/1.0" 200 17083ix-or12-16.ix.netcom
00 12054svasu.extern.ucsd.edu - - [01/Jul/1995:00:25:38 -0400] "GET /history/apollo/apollo-1/apollo-1.html HTTP/1.0" 200 38421x-or12-16.ix.netcom.com - - [0
clogosmall.gif HTTP/1.0" 200 3635149.171.160.182 - - [01/Jul/1995:00:25:44 -0400] "GET /images/launch-logo.gif HTTP/1.0" 200 1713n031681.ksc.nasa.gov - - [0
00] "GET /images/launch-logo.gif HTTP/1.0" 200 1713teleman.pr.mcs.net - - [01/Jul/1995:00:25:53 -0400] "GET /history/apollo/apollo-13/sounds/a13_002.wav HTTP
om - - [01/Jul/1995:00:26:08 -0400] "GET /shuttle/missions/sts-71/sts-71-press-kit.txt HTTP/1.0" 200 78588128.187.140.171 - - [01/Jul/1995:00:26:09 -0400] "C
0" 200 40310128.187.140.171 - - [01/Jul/1995:00:26:12 -0400] "GET /images/KSC-logosmall.gif HTTP/1.0" 200 1204slip2-42.acs.ohio-state.edu - - [01/Jul/1995:00
" 302 96cruzio.com - - [01/Jul/1995:00:26:23 -0400] "GET /shuttle/countdown/video/livevideo.jpeg HTTP/1.0" 200 47699www-b3.proxy.aol.com - - [01/Jul/1995:00
/shuttle/technology/sts-newsref/stsref-toc.html HTTP/1.0" 200 81920dal31.pic.net - - [01/Jul/1995:00:26:28 -0400] "GET /images/KSC-logosmall.gif HTTP/1.0" 20
00] "GET /images/ksclogosmall.gif HTTP/1.0" 200 3635fht.cts.com - - [01/Jul/1995:00:26:37 -0400] "GET /history/apollo/images/apollo-logo.gif HTTP/1.0" 200 3
0400] "GET /images/KSC-logosmall.gif HTTP/1.0" 200 1204ix-sd9-18.ix.netcom.com - - [01/Jul/1995:00:26:47 -0400] "GET /shuttle/resources/orbiters/atlantis.ht
rt.ab.ca - - [01/Jul/1995:00:26:52 -0400] "GET /images/ksclogosmall.gif HTTP/1.0" 200 3635ix-sd9-18.ix.netcom.com - - [01/Jul/1995:00:26:52 -0400] "GET /shu
:56 -0400] "GET /shuttle/missions/sts-71/mission-sts-71.html HTTP/1.0" 200 12040ix-or12-16.ix.netcom.com - - [01/Jul/1995:00:26:57 -0400] "GET /images/WORL
istory/apollo/apollo-goals.txt HTTP/1.0" 200 712ix-sd9-18.ix.netcom.com - - [01/Jul/1995:00:27:07 -0400] "GET /shuttle/resources/orbiters/orbiters-logo.gif
.0" 200 3985remote1-p16.ume.maine.edu - - [01/Jul/1995:00:27:14 -0400] "GET /shuttle/countdown/ HTTP/1.0" 200 3985svasu.extern.ucsd.edu - - [01/Jul/1995:00:
1/Jul/1995:00:27:22 -0400] "GET /history/apollo/apollo-13/images/index.gif HTTP/1.0" 200 99942sagami2.isc.meiji.ac.jp - - [01/Jul/1995:00:27:22 -0400] "GET
ttle/countdown/liftoff.html HTTP/1.0" 304 0dd11-006.compuserve.com - - [01/Jul/1995:00:27:27 -0400] "GET /shuttle/missions/missions.html HTTP/1.0" 200 8677d:
27:32 -0400] "GET /shuttle/missions/sts-71/images/KSC-95EC-0918.gif HTTP/1.0" 200 31631fht.cts.com - - [01/Jul/1995:00:27:32 -0400] "GET /history/apollo/apo
```

```
AccessLogFile1 - Notepad
File Edit Format View Help
85ix-dfw11-01.ix.netcom.com - - [01/Jul/1995:00:22:29 -0400] "GET /shuttle/missions/100th.html HTTP/1.0" 200 32303chi067.wva.com - - [01/Jul/1995:00:22:31 -
00:22:39 -0400] "GET /shuttle/missions/sts-71/images/KSC-95EC-0443.gif HTTP/1.0" 200 64910dnet018.sat.texas.net - - [01/Jul/1995:00:22:42 -0400] "GET /histo
hp.com - - [01/Jul/1995:00:22:57 -0400] "GET /images/ksclogo-medium.gif HTTP/1.0" 200 0rcr14.crl.com - - [01/Jul/1995:00:22:58 -0400] "GET /shuttle/missions/
- - [01/Jul/1995:00:23:01 -0400] "GET /shuttle/countdown/count.gif HTTP/1.0" 200 40310picard.microsys.net - - [01/Jul/1995:00:23:02 -0400] "GET /shuttle/mis
/shuttle/countdown/ HTTP/1.0" 304 0palona1.cns.hp.com - - [01/Jul/1995:00:23:09 -0400] "GET /images/MOSAIC-logosmall.gif HTTP/1.0" 304 0sartre.execpc.com -
o/images/footprint-small.gif HTTP/1.0" 200 18149palona1.cns.hp.com - - [01/Jul/1995:00:23:19 -0400] "GET /images/USA-logosmall.gif HTTP/1.0" 304 0rlnport2.c
/ HTTP/1.0" 200 3985spalona1.cns.hp.com - - [01/Jul/1995:00:23:27 -0400] "GET /facts/facts.html HTTP/1.0" 200 4717chi067.wva.com - - [01/Jul/1995:00:23:27 -0
666clab040.ins.gu.edu.au - - [01/Jul/1995:00:23:34 -0400] "GET /cgi-bin/imagemap/countdown?104,169 HTTP/1.0" 302 110gclab040.ins.gu.edu.au - - [01/Jul/1995:0
6149.171.160.183 - - [01/Jul/1995:00:23:46 -0400] "GET /shuttle/missions/sts-71/sts-71-patch-small.gif HTTP/1.0" 200 12054ix-ftw-tx1-21.ix.netcom.com - - [0
y2.indy.net - - [01/Jul/1995:00:24:01 -0400] "GET /images/ksclogo-medium.gif HTTP/1.0" 200 5866gclab040.ins.gu.edu.au - - [01/Jul/1995:00:24:01 -0400] "GE
00 40310indy2.indy.net - - [01/Jul/1995:00:24:09 -0400] "GET /images/MOSAIC-logosmall.gif HTTP/1.0" 200 363indy2.indy.net - - [01/Jul/1995:00:24:10 -0400] "C
x.netcom.com - - [01/Jul/1995:00:24:17 -0400] "GET /images/kscmpap-small.gif HTTP/1.0" 200 39017149.171.160.182 - - [01/Jul/1995:00:24:18 -0400] "GET /shuttli
00:24:25 -0400] "GET /shuttle/missions/missions.html HTTP/1.0" 200 8677b1v-pm2-ip16.halcyon.com - - [01/Jul/1995:00:24:27 -0400] "GET /images/launchmedium.g
p.com - - [01/Jul/1995:00:24:39 -0400] "GET /images/WORLD-logosmall.gif HTTP/1.0" 304 0alyssa.prodigy.com - - [01/Jul/1995:00:24:44 -0400] "GET /shuttle/mis
/Jul/1995:00:24:51 -0400] "GET /shuttle/missions/sts-71/images/images.html HTTP/1.0" 200 7634blv-pm2-ip16.halcyon.com - - [01/Jul/1995:00:24:51 -0400] "GE
"GET /shuttle/missions/sts-71/sts-71-patch-small.gif HTTP/1.0" 200 12054indy2.indy.net - - [01/Jul/1995:00:25:02 -0400] "GET /shuttle/missions/sts-71/mission
huttle/missions/sts-71/movies/sts-71-tcdt-crew-walkout.mpg HTTP/1.0" 200 49152deimos.jp1.arizona.edu - - [01/Jul/1995:00:25:10 -0400] "GET /shuttle/missions,
1.0" 200 1713sartre.execpc.com - - [01/Jul/1995:00:25:20 -0400] "GET /shuttle/countdown/ HTTP/1.0" 304 0ix-phx5-17.ix.netcom.com - - [01/Jul/1995:00:25:21 -
/1.0" 200 25814boing.dgsys.com - - [01/Jul/1995:00:25:31 -0400] "GET /shuttle/missions/sts-67/sts-67-patch-small.gif HTTP/1.0" 200 17083ix-or12-16.ix.netcom
00 12054svasu.extern.ucsd.edu - - [01/Jul/1995:00:25:38 -0400] "GET /history/apollo/apollo-1/apollo-1.html HTTP/1.0" 200 38421x-or12-16.ix.netcom.com - - [0
clogosmall.gif HTTP/1.0" 200 3635149.171.160.182 - - [01/Jul/1995:00:25:44 -0400] "GET /images/launch-logo.gif HTTP/1.0" 200 1713n031681.ksc.nasa.gov - - [0
00] "GET /images/launch-logo.gif HTTP/1.0" 200 1713teleman.pr.mcs.net - - [01/Jul/1995:00:25:53 -0400] "GET /history/apollo/apollo-13/sounds/a13_002.wav HTTP
om - - [01/Jul/1995:00:26:08 -0400] "GET /shuttle/missions/sts-71/sts-71-press-kit.txt HTTP/1.0" 200 78588128.187.140.171 - - [01/Jul/1995:00:26:09 -0400] "C
0" 200 40310128.187.140.171 - - [01/Jul/1995:00:26:12 -0400] "GET /images/KSC-logosmall.gif HTTP/1.0" 200 1204slip2-42.acs.ohio-state.edu - - [01/Jul/1995:00
" 302 96cruzio.com - - [01/Jul/1995:00:26:23 -0400] "GET /shuttle/countdown/video/livevideo.jpeg HTTP/1.0" 200 47699www-b3.proxy.aol.com - - [01/Jul/1995:00
/shuttle/technology/sts-newsref/stsref-toc.html HTTP/1.0" 200 81920dal31.pic.net - - [01/Jul/1995:00:26:28 -0400] "GET /images/KSC-logosmall.gif HTTP/1.0" 20
00] "GET /images/ksclogosmall.gif HTTP/1.0" 200 3635fht.cts.com - - [01/Jul/1995:00:26:37 -0400] "GET /history/apollo/images/apollo-logo.gif HTTP/1.0" 200 3
0400] "GET /images/KSC-logosmall.gif HTTP/1.0" 200 1204ix-sd9-18.ix.netcom.com - - [01/Jul/1995:00:26:47 -0400] "GET /shuttle/resources/orbiters/atlantis.ht
rt.ab.ca - - [01/Jul/1995:00:26:52 -0400] "GET /images/ksclogosmall.gif HTTP/1.0" 200 3635ix-sd9-18.ix.netcom.com - - [01/Jul/1995:00:26:52 -0400] "GET /shu
:56 -0400] "GET /shuttle/missions/sts-71/mission-sts-71.html HTTP/1.0" 200 12040ix-or12-16.ix.netcom.com - - [01/Jul/1995:00:26:57 -0400] "GET /images/WORL
istory/apollo/apollo-goals.txt HTTP/1.0" 200 712ix-sd9-18.ix.netcom.com - - [01/Jul/1995:00:27:07 -0400] "GET /shuttle/resources/orbiters/orbiters-logo.gif
.0" 200 3985remote1-p16.ume.maine.edu - - [01/Jul/1995:00:27:14 -0400] "GET /shuttle/countdown/ HTTP/1.0" 200 3985svasu.extern.ucsd.edu - - [01/Jul/1995:00:
1/Jul/1995:00:27:22 -0400] "GET /history/apollo/apollo-13/images/index.gif HTTP/1.0" 200 99942sagami2.isc.meiji.ac.jp - - [01/Jul/1995:00:27:22 -0400] "GET
ttle/countdown/liftoff.html HTTP/1.0" 304 0dd11-006.compuserve.com - - [01/Jul/1995:00:27:27 -0400] "GET /shuttle/missions/missions.html HTTP/1.0" 200 8677d:
27:32 -0400] "GET /shuttle/missions/sts-71/images/KSC-95EC-0918.gif HTTP/1.0" 200 31631fht.cts.com - - [01/Jul/1995:00:27:32 -0400] "GET /history/apollo/apo
```

```

AccessLogFile1 - Notepad
File Edit Format View Help
ttle/countdown/liftoff.html HTTP/1.0" 304 0dd11-006.comuserve.com - - [01/Jul/1995:00:27:27 -0400] "GET /shuttle/missions/missions.html HTTP/1.0" 200 8677d:
27:32 -0400] "GET /shuttle/missions/sts-71/images/KSC-95EC-0918.gif HTTP/1.0" 200 31631fht.cts.com - - [01/Jul/1995:00:27:32 -0400] "GET /history/apollo/apo:
0" 200 3985gclab040.ins.gu.edu.au - - [01/Jul/1995:00:27:37 -0400] "GET /htbin/cdt_clock.pl HTTP/1.0" 200 543ix-sd9-18.ix.netcom.com - - [01/Jul/1995:00:27:
1/Jul/1995:00:27:45 -0400] "GET /images/USA-logosmall.gif HTTP/1.0" 200 234hoflink.com - - [01/Jul/1995:00:27:45 -0400] "GET /shuttle/missions/sts-5/mission-
if HTTP/1.0" 200 669128.187.140.171 - - [01/Jul/1995:00:27:49 -0400] "GET /cgi-bin/imagemap/countdown?375,269 HTTP/1.0" 302 68sartre.execpc.com - - [01/Jul/
0 3985ix-atl12-02.ix.netcom.com - - [01/Jul/1995:00:27:57 -0400] "GET /images/MOSAIC-logosmall.gif HTTP/1.0" 200 363piweba3y.prodigy.com - - [01/Jul/1995:00
200 8677www-a1.proxy.aol.com - - [01/Jul/1995:00:28:00 -0400] "GET /images/opf-logo.gif HTTP/1.0" 200 32511blv-pm2-ip16.halcyon.com - - [01/Jul/1995:00:28:0
" 200 12040kristina.az.com - - [01/Jul/1995:00:28:12 -0400] "GET /history/apollo/apollo-13/apollo-13-info.html HTTP/1.0" 200 1583halon.sybase.com - - [01/Jul
/missions/sts-5/sts-5-info.html HTTP/1.0" 200 1405telemar.pr.mcs.net - - [01/Jul/1995:00:28:25 -0400] "GET /history/apollo/apollo-13/news/ HTTP/1.0" 200 377:
ons/sts-71/images/images.html HTTP/1.0" 200 7634149.171.160.182 - - [01/Jul/1995:00:28:33 -0400] "GET /shuttle/missions/sts-71/sts-71-day-04-highlights.html
A-logosmall.gif HTTP/1.0" 200 234kristina.az.com - - [01/Jul/1995:00:28:42 -0400] "GET /history/apollo/apollo-13/images/ HTTP/1.0" 200 1851brandt.kensei.com
] "GET /images/NASA-logosmall.gif HTTP/1.0" 200 786telemar.pr.mcs.net - - [01/Jul/1995:00:28:45 -0400] "GET /icons/image.xbm HTTP/1.0" 200 509whlanc.cts.com
pl.arizona.edu - - [01/Jul/1995:00:28:54 -0400] "GET /shuttle/missions/sts-71/sts-71-patch-small.gif HTTP/1.0" 200 12054dal08.onramp.net - - [01/Jul/1995:00
ext.xbm HTTP/1.0" 200 527hoflink.com - - [01/Jul/1995:00:29:00 -0400] "GET /shuttle/missions/sts-5/movies/ HTTP/1.0" 200 375icagen.vnet.net - - [01/Jul/1995
.gif HTTP/1.0" 200 1713eagle.co.la.ca.us - - [01/Jul/1995:00:29:08 -0400] "GET /shuttle/missions/sts-71/mission-sts-71.html HTTP/1.0" 200 12040ix-sd9-18.ix.i
com - - [01/Jul/1995:00:29:15 -0400] "GET /shuttle/missions/sts-71/news HTTP/1.0" 302 ix-ftw-tx1-21.ix.netcom.com - - [01/Jul/1995:00:29:19 -0400] "GET /cg:
/Jul/1995:00:29:28 -0400] "GET /software/winvn/wvsmall.gif HTTP/1.0" 200 13372ssandgy.scvnet.com - - [01/Jul/1995:00:29:30 -0400] "GET /shuttle/countdown HT
m - - [01/Jul/1995:00:29:35 -0400] "GET /images/KSC-logosmall.gif HTTP/1.0" 200 1204news.ti.com - - [01/Jul/1995:00:29:36 -0400] "GET /icons/movie.xbm HTTP/:
29:49 -0400] "GET /facilities/lcc.html HTTP/1.0" 200 2489telemar.pr.mcs.net - - [01/Jul/1995:00:29:50 -0400] "GET /history/apollo/apollo-13/images/index.gif
ssions/sts-71/images/KSC-95EC-0918.jpg HTTP/1.0" 200 46888ppp31.cowan.edu.au - - [01/Jul/1995:00:30:03 -0400] "GET /shuttle/countdown/ HTTP/1.0" 200 3985brar
1.0" 200 1204131.128.2.155 - - [01/Jul/1995:00:30:10 -0400] "GET /shuttle/missions/sts-71/mission-sts-71.html HTTP/1.0" 200 12040halon.sybase.com - - [01/Jul
f HTTP/1.0" 200 40310131.128.2.155 - - [01/Jul/1995:00:30:18 -0400] "GET /shuttle/missions/sts-71/sts-71-patch-small.gif HTTP/1.0" 200 12054thimble-d229.sier
az.com - - [01/Jul/1995:00:30:31 -0400] "GET /history/apollo/apollo-13/images/70HC323.GIF HTTP/1.0" 200 154223ssandgy.scvnet.com - - [01/Jul/1995:00:30:31 -0
36.dial.umd.edu - - [01/Jul/1995:00:30:39 -0400] "GET /shuttle/missions/sts-71/images/KSC-95EC-0893.jpg HTTP/1.0" 200 298302spazan.cts.com - - [01/Jul/1995:0
995:00:30:46 -0400] "GET /shuttle/countdown/liftoff.html HTTP/1.0" 304 0spazan.cts.com - - [01/Jul/1995:00:30:46 -0400] "GET /images/USA-logosmall.gif HTTP:
"GET /shuttle/countdown/count.gif HTTP/1.0" 200 40310dd11-006.comuserve.com - - [01/Jul/1995:00:30:56 -0400] "GET /shuttle/missions/sts-71/news-sts-71-mcc-t
400] "GET /shuttle/missions/sts-71/mission-sts-71.html HTTP/1.0" 200 12040spazan.cts.com - - [01/Jul/1995:00:31:06 -0400] "GET /shuttle/missions/sts-71/miss-
1 -0400] "GET /shuttle/missions/sts-71/mission-sts-71.html HTTP/1.0" 200 12040larryr.cts.com - - [01/Jul/1995:00:31:11 -0400] "GET /images/KSC-logosmall.gif
mall.gif HTTP/1.0" 200 786pma02.rt66.com - - [01/Jul/1995:00:31:15 -0400] "GET /shuttle/missions/sts-67/images/images.html HTTP/1.0" 200 4464b14.ppp.mo.net -
ET /history/apollo/apollo-13/images/70HC353.GIF HTTP/1.0" 200 153725dd11-006.comuserve.com - - [01/Jul/1995:00:31:35 -0400] "GET /icons/text.xbm HTTP/1.0" ;
1/1995:00:31:47 -0400] "GET /shuttle/countdown/tour.html HTTP/1.0" 200 4347dd11-006.comuserve.com - - [01/Jul/1995:00:31:47 -0400] "GET /shuttle/missions/st
95:00:32:02 -0400] "GET /images/NASA-logosmall.gif HTTP/1.0" 200 786piweba3y.prodigy.com - - [01/Jul/1995:00:32:02 -0400] "GET /shuttle/missions/sts-71/image
1/Jul/1995:00:32:11 -0400] "GET /cgi-bin/imagemap/countdown?97,112 HTTP/1.0" 302 111larryr.cts.com - - [01/Jul/1995:00:32:11 -0400] "GET /shuttle/countdown/
tares.physics.carleton.edu - - [01/Jul/1995:00:32:13 -0400] "GET /shuttle/technology/images/srb_mod_compare_6-small.gif HTTP/1.0" 304 0antares.physics.carlet
- [01/Jul/1995:00:32:19 -0400] "GET /images/ksclgo-medium.gif HTTP/1.0" 200 5866whlanc.cts.com - - [01/Jul/1995:00:32:22 -0400] "GET /shuttle/technology/st

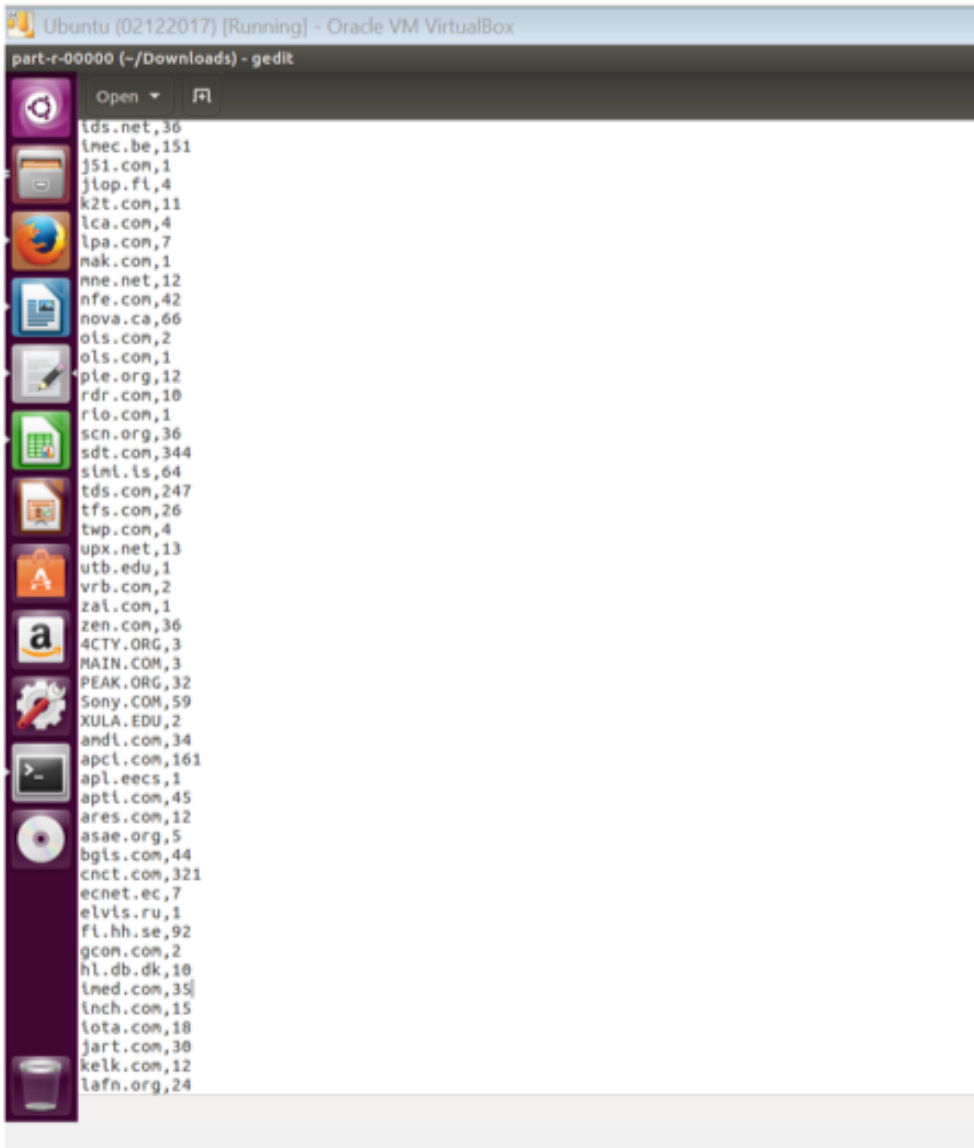
```

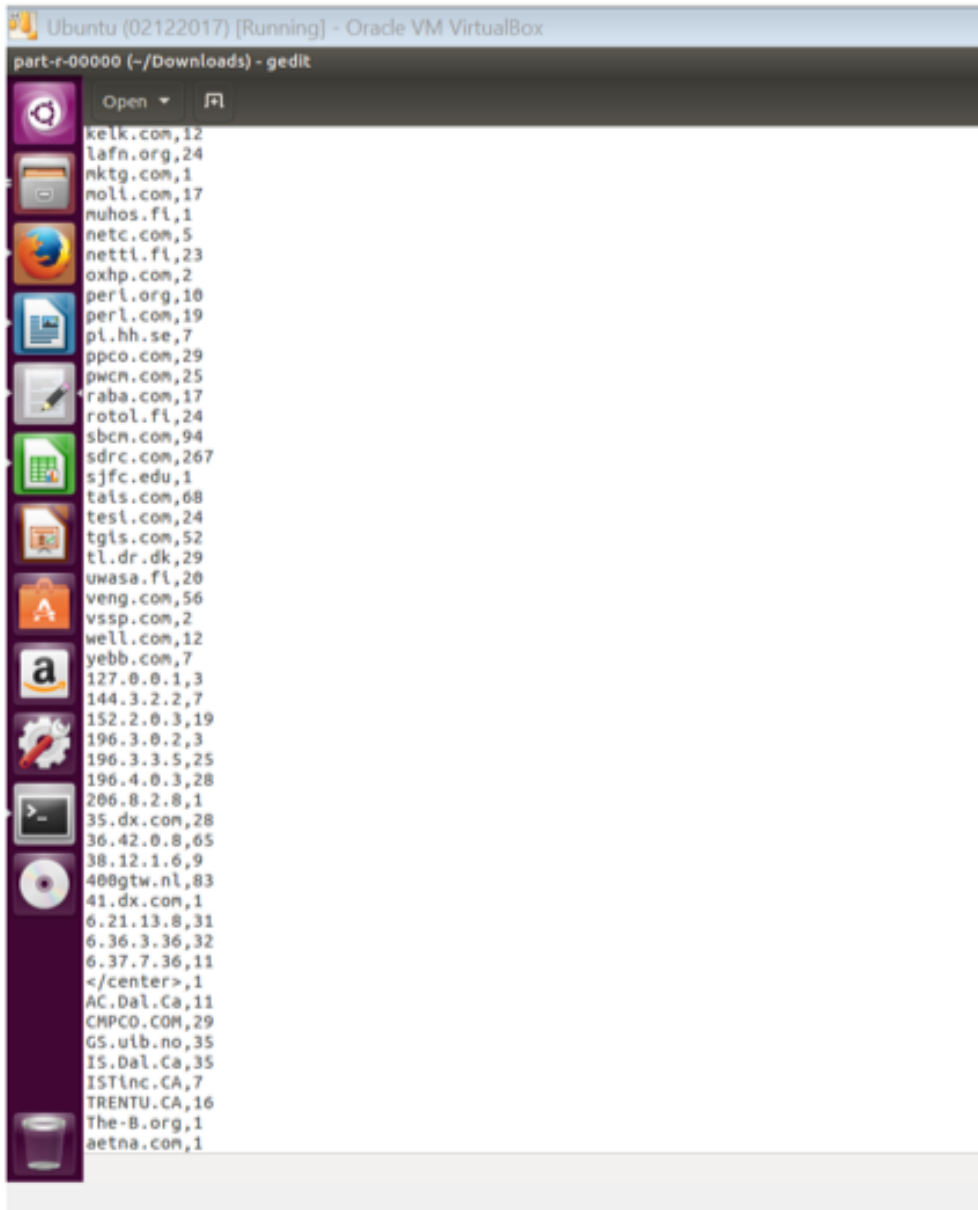
E. Output sample snapshot



The screenshot shows a gedit window titled "part-r-00000 (~/.Downloads) - gedit" running in a VirtualBox environment. The window displays a list of domain names and their associated counts, such as "BQc=,1", "cp.ca,3", "CAE.CA,10", etc. The list is displayed in a monospaced font. On the left side of the window, there is a vertical toolbar with various icons for file operations and editing.

```
part-r-00000 (~/.Downloads) - gedit
BQc=,1
cp.ca,3
CAE.CA,10
au.com,5
en.com,18
fcs.nl,7
fml.ch,17
gdc.ca,1
hel.fi,48
ksm.ny,47
ktt.fi,31
lvb.nl,1
pb.net,10
rhh.dk,114
sea.ru,18
www.sg,26
RTD.COM,1
UNAV.ES,8
aod.com,5
axs.net,3
bix.com,17
cml.com,55
crl.com,45
crx.com,1
dfw.net,48
dhp.com,1
dni.net,12
etca.fr,1
frl.com,2
fsd.com,45
gmsg.ch,12
gs1.com,35
gtx.com,35
hal.COM,73
lds.net,36
trec.be,151
js1.com,1
jlop.fi,4
k2t.com,11
lca.com,4
lpa.com,7
mak.com,1
mne.net,12
nfe.com,42
nova.ca,66
ols.com,2
ols.com,1
ple.org,12
rdr.com,18
rlo.com,1
scn.oro.36
```





Ubuntu (02122017) [Running] - Oracle VM VirtualBox

```
part-r-00000 (~/.Downloads) - gedit
Open [v] [F]
cLark.edu,2
clark.net,5607
clinet.fi,1
cpcug.org,232
cs.bc.edu,1
dames.com,3
db.sw.com,1
dmgow.com,8
ee.hit.fi,1
ee.tut.fi,2
eldec.com,42
entek.com,6
fangz.com,1
fasor.com,2
fw.itu.ch,1
g8.rmc.ca,34
genie.com,77
glock.com,11
gsv.gu.se,234
hq.si.net,13
huber.com,130
i4.auc.dk,99
iainc.com,51
iglou.com,53
inmax.com,21
innet.com,2
isgate.is,224
itp.pp.fi,9
lacma.org,10
largo.com,2
linex.com,7
lisp.eecs,8
mm.fhg.de,26
mv.MV.COM,22
ncs.co.nz,45
nexus.net,1
nic.dn.se,8
ns.bmw.de,77
ns.ilc.de,19
ns.mol.fi,20
ns.osn.de,70
ns.scn.de,188
obo.pp.fi,16
panix.com,60
ph.idg.dk,4
prose.com,5
ps.rrv.se,31
ps.utb.es,4
qdeck.com,6
quirk.com,1
ra.icl.fi,18
```