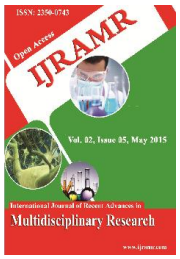


International Journal of Recent Advances in Multidisciplinary Research
Vol. 02, Issue 05, pp.0407-0412, May, 2015

Research Article

CONTEXT AWARE DYNAMIC WEB DESIGN USING HADOOP

Dr. Sreenivasa Ravi, K., *Abhiram, D.V., Santosh, K.S.S.K. and Bharathi Devi, A.

K L University, Vijayawada, India

ARTICLE INFO

Article History:
Received 19th February 2015
Received in revised form
21th March, 2015
Accepted 25th April, 2015
Published online 31th May, 2015

Keywords:
Hadoop,
Web Programming,
Big Data,
Dynamic Web Page Design.

ABSTRACT

Context aware web design is not a new technology and its been in the use from many years. However the algorithms designed for that purpose are written in jsp or similar other server programming languages which includes bringing the huge amounts of the transactional data into a single computing unit (which is usually of very high computing capacity) and run analytics that are necessary. In this paper we have tried to implement the similar website which changes dynamically according to the users latest interests. The web content to be displayed is computed by the Hadoop instead of traditional servlet programs .Hadoop relies on taking the computing power to the data instead of data to the computing source which reduces the cost of the maintenance by one tenth of the present value and the processing power varies linearly with the input data, Unlike in the traditional processing where time varies exponentially with the data. Although the idea presented is being used, this is an implementation of the Hadoop's ability to adapt to the traditional database structures and run quite efficiently to produce effective results.

INTRODUCTION

Hadoop is a software framework initially designed by Yahoo which deals with processing big data (in order of hundreds of petabytes) in a distributed fashion on a large amounts of commodity hardware. The main reason why Hadoop became so popular is because of its effective fault tolerant model. When we deal with several thousands of nodes the probability of occurrence of fault increases dramatically and Hadoop has a mechanism recover from these without affecting the overall execution time exponentially. It also limits bandwidth usage unlike other distributed systems.

Components of Hadoop

The Apache Hadoop project has two core components, the file store called Hadoop Distributed File System (HDFS), and the programming framework called MapReduce. There are a number of supporting projects that leverage HDFS and MapReduce. This article will provide a summary

***HDFS*:** If 4000+ computers to work on the data, then spreading the data across 4000+ computers is the best possible way. HDFS does this for us. HDFS has a few moving parts. The Datanodes store the data, and the Namenode keeps track of where data is stored.

Corresponding author: Abhiram, D.V.,
K L University, Vijayawada, India.

MapReduce: This is the programming model for Hadoop. There are two phases, called Map and Reduce. Everything in Hadoop is done in this two parts.

Hadoop Streaming: A utility to enable MapReduce code in any language: C, Perl, Python, C++, Bash, etc. for example a Python mapper and an AWK reducer. Hive and Hue: Hive can convert sql query into a MapReduce job. Hue provides a browser-based graphical interface to do hive work.

Pig: A higher-level programming environment to do MapReduce coding. The Pig language is called Pig Latin.

Sqoop: a tool for transferring data between Hadoop and relational databases. You can use Sqoop to import data from a MySQL or Oracle database into HDFS, run MapReduce on the data, and then export the data back into an RDBMS. Sqoop automates these processes, using MapReduce to import and export the data in parallel with fault-tolerance.

Oozie: Manages Hadoop workflow. This doesn't replace scheduler or BPM tooling, but it does provide if-then-else branching and control within Hadoop jobs.

HBase: A super-scalable key-value store. It works very much like a persistent hash-map. It is not a relational database.

FLUME: is a distributed system for collecting, aggregating, and moving large amounts of data from multiple sources into

HDFS or another central data store. Enterprises typically collect log files on application servers or other systems and archive the log files in order to comply with regulations. Being able to ingest and analyze that unstructured or semi-structured data in Hadoop can turn this passive resource into a valuable asset.

FlumeNG: A real time loader for streaming your data into Hadoop. It stores data in HDFS and HBase. It is advanced version of the flume server.

Whirr: Cloud provisioning for Hadoop.

Mahout: Machine learning for Hadoop. Used for predictive analytics and other advanced analysis.

Fuse: Makes the HDFS system look like a regular file system like the use of ls, rm, cd, and others on HDFS data

Zookeeper: Used to manage synchronization for the cluster.

Why All This Change Now

In the current era Big Data is becoming common and predominant. Big Data refers to the large amounts, at least terabytes, of poly-structured data that flows continuously through and around organizations, including video, text, sensor logs, and transactional records. The business benefits of analyzing this data can be significant. According to a recent study by the MIT Sloan School of Management, organizations that use analytics on their businesses are twice as likely to be top performers in their industry as those that won't. Business analysts at a large company such as Intel or amazon, for example, with its global market and Complex supply chain, have long sought insight into customer demand by analyzing far flung Data points culled from market information and business transactions. Increasingly, the data we need is embedded in economic reports, discussion forums, news sites, social Networks, weather reports, wikis, tweets, and blogs, as well as transactions. By analyzing all the data available, decision-makers can better assess competitive threats, anticipate changes in customer behavior, strengthen supply chains, improve the effectiveness of marketing campaigns, and enhance business continuity.

Many of these benefits are not new to organizations that have mature Processes for incorporating business intelligence (BI) and analytics into their decision-making. However, most organizations have yet to take full advantage of new technologies for handling big data. Put simply, the cost of the technologies needed to store and analyze large volumes of diverse data has dropped, thanks to open source software running on industry-standard hardware. The cost has dropped so much, in fact, that the key strategic question is no longer what data is relevant, but rather how to extract the most value from all the available data. Rapidly ingesting, storing, and processing big data requires a cost-effective infrastructure that can scale with the amount of data and the scope of analysis. Most organizations with traditional data Platforms—typically relational database management systems (RDBMS) coupled to enterprise data warehouses (EDW) using ETL tools—find that their legacy infrastructure is either technically incapable or financially impractical for storing and analyzing big data. A traditional ETL process extracts data from multiple sources, then cleanses, formats, and loads it into a data warehouse for analysis. hence the source data sets are large, fast, and unstructured, traditional ETL can become the bottleneck, because it is too complex to develop, too Expensive to operate, and takes too long to execute.

How hadoop fits in existing system and the work flow

The developers of Hadoop are clever enough to know that the enterprise systems do not have time or resources to switch to a completely new architecture suddenly just because it has better resource management. The architectural changes cost them some decades of this performance costs hence the Hadoop is designed to fit in the work flow of the present system and has greater flexibility to program in any language comfortable. The following section shows the mapping program which is written in python and the reducer program written in awk (UNIX tool)

Hadoop Streaming

Hadoop streaming is a utility that comes with the Hadoop distribution. The utility allows you to create and run Map/Reduce jobs with any executable or script as the mapper and/or the reducer. For example:

Python Mapper:

```
#!/usr/bin/env python
import sys
# read stdin
for linein in sys.stdin:
# strip blanks
 linein = linein.strip()
# split into words
 mywords = linein.split(",")
# loop on mywords, output the length of each word
 for word in mywords:
# the reducer just cares about the first column,
# normally there is a key - value pair
  print '%s %s' % ((word), 1)
 AWK REDUCER :
com# awk is calculating
# NR - the number of words in total
# sum/NR - the average word length
# sqrt(mean2/NR) - the standard deviation
$ cat statsreducer.awk
awk '{delta = $1 - avg; avg += delta / NR; \mean2 += delta * ($1 - avg); sum=$1+sum } \END { print NR, sum/NR, sqrt(mean2 / NR); }'
```

Working of hadoop streaming

In the given example both the mapper and the reducer read input from stdin line to line and output to stdout. The cluster utility will create a map job and submit it to the appropriate cluster node and monitor the progress of the job until the job gets completed. When an executable is specified for mappers, each mapper task will be launched as a separate process. As the mapper task runs, it converts its inputs into lines and will feed the lines to the stdin of the specific process. In the meantime, the mapper collects the line oriented outputs from the stdout of the process and converts each line into a key/value pair (as per the program), and is collected as the output of the mapper. By default, the prefix of a line up to the first tab character is the

key and the rest of the line (excluding the tab character) will be the value. If there is no tab or space character in the line, then entire line is considered as key and the value is null. However, this can be customized. When an executable is specified for reducers, each reducer task will be launched as a separate process then the reducer is initialized. The input of the reducer is always key value pairs is the reducer runs, it converts its input key/values pairs into lines and feeds the lines to the stdin of the specific process. In the meantime, the reducer collects the line oriented outputs from the stdout of the process and converts each line into a key/value pair, which is collected as the output of the reducer. By default, the prefix of a line up to the first space character is the key and the rest of the line (excluding the tab character) is the value. However, this also can be customized. This is the basis for the communication protocol between the Map/Reduce framework and the streaming mapper/reducer.

MATERIALS AND MATHODS

There are so many resources to start working the Hadoop and its distributed file system. Cloud era is one of the best example and provides a good starting point to get running. However we have built our own single node cluster setup from apache using virtual machine which runs Ubuntu 14.04 (LTS) on windows 7 operating system. The experiment is done on a mysql database which stores and collects the users interest in songs from the website we have already developed.

Automated periodic refresh of latest information

The database is continuously flooded with the user statistics and the hadoop has to keep up with the database. For importing the data from the database to the HDFS sqoop is the best solution and it initializes a mapper and reducer jobs to bring the data from the database and stores it in the HDFS. Here is the small insight on the sqoop command.

$ sqoop import – connect jdbc:mysql://10.45.16.xxx:3306/adda – table statistics – username abhiram –P – target-dir input/database/ -m 10 [12]

The above sqoop command imports the table statistics from the database named adda and from the specified ip address using username as abhiram for authentication. This command initially helps us to bring all the data from the database but as the data gets incremented the only way to bring the extra data is to use – check-column and last-value attributes using which it checks for the latest data and stores it in a new file. However the table must have a primary key to use this feature and the check-column attribute takes only primary key. The table we have imported does not contain any primary key so we have imported the entire table again when we wanted to get new data and having the number of map tasks as one. As technical support knowledge sets are updated daily with new solutions derived, a mechanism to mechanically update the technical support information from that the clustered results are derived to confirm correct output is needed. Given the character of the information, it's spare to perform such a procedure on a commonplace. To avoid uploading the whole technical support knowledge assault a commonplace, a crone job may be scheduled to mechanically backup the changes that occurred over the preceding twenty four hours.

It's projected that a separate Hadoop job is accountable for uploading the CSV files containing the main points of the support requests received that day into HDFS.

Parallalized Clustering

In order for information to be processed by the driver cluster algorithms, it should initial be reborn to vector format. Technical support knowledge keep in HDFS is reborn into vectors victimization Mahout's existing statement for consequent agglomeration analysis. significantly, the challenge of characteristic connected support calls supported their downside description is resolved by victimization Mahout's distributed agglomeration machine learning algorithms to analyze the information set, thereby characteristic support calls with the same description. Multiple agglomeration algorithms are evaluated within the projected system with reference to system performance and accuracy. the particular details of the agglomeration analysis and results are mentioned more in Section V. it's on the far side the scope of this paper to produce an in depth description of every agglomeration algorithmic program with reference to a parallelized Map scale back job. But to as a sign, the k-means algorithmic program represented from the angle of a MapReduce job is outlined: every map task receives a set of the initial centroids and is accountable for assign every input data-point i.e. text vector to its nearest cluster i.e. centroid. For every information, the clerk generates a key/value combine, wherever the secret is the cluster symbol and therefore the worth corresponds to the coordinates of that time. The algorithmic program uses a combiner to cut back the number of information to be transferred from the clerk to the reducer. The combiner receives all key/value pairs from the clerk and produces partial sums of the input vectors for every cluster. All values related to constant cluster are sent to constant reducer, so every reducer receives the partial sums of all points of 1 or additional clusters and computes the new cluster centroids. The driving force program iterates over the points and clusters till all clusters have converged or till the utmost variety of iterations has been reached.

Real Time Platform Access

Importantly, Hadoop doesn't give period of time access and is intended for instruction execution. Thus, once the analysis part is completed, a Hadoop MapReduce job stores the agglomeration results into a non-relational information in order that technical support engineers will question the data in period of time. The data keep includes the support call mark, the cluster symbol and therefore the chance of that support decision happiness to a given cluster. Given its native integration with Hadoop, HBase is chosen for this purpose. once Associate in Nursing engineer requests the support calls associated with a selected case, the cluster to that such a case ought to belong is known in HBase, with all support calls at intervals that cluster sorted supported their cluster membership chance. Associate in Nursing ordered list containing the support decision symbol and its associated cluster membership chance is came back to the technical support engineer. The support calls displayed at the highest of such list are additional possible to contain similar issues to the desired unresolved technical support case and as a result are additional possible to share constant resolution.

Experimental Details

The enforced resolution made public within the previous Section is currently evaluated employing a real-world VMware

desc adda.store

Page Size: 20 Total Rows: 3 Page: 1 of 1 Matching Rows:

#	Field	Type	Null	Key	Default	Extra
1	user	varchar(40)	YES		<NULL>	
2	id	int(11)	NO		<NULL>	
3	serial	int(11)	NO	PRI	<NULL>	auto_increment

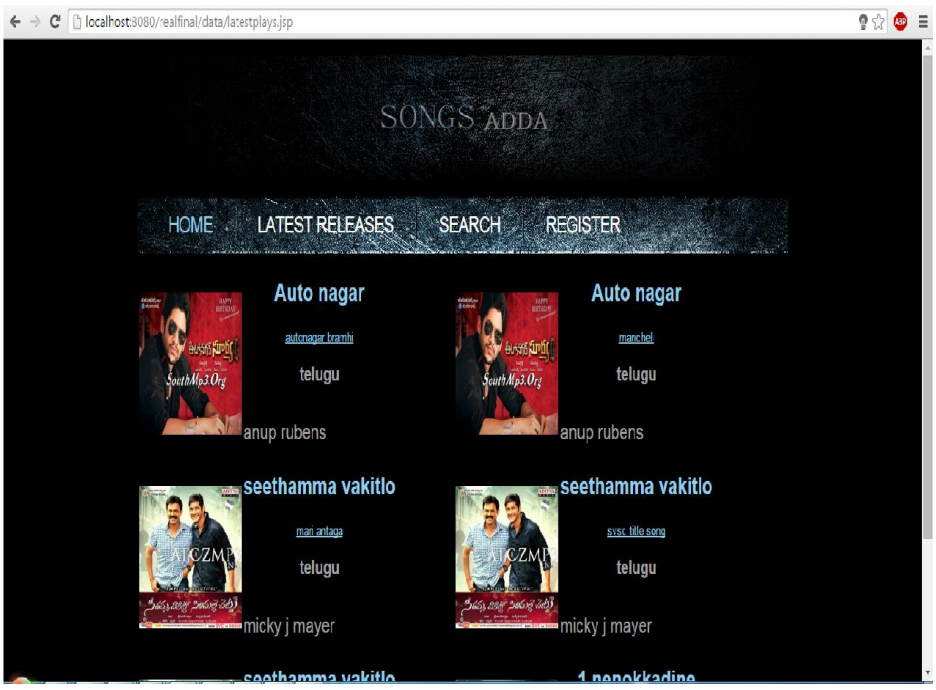
Snap shot of the description of table store

select * from adda.store

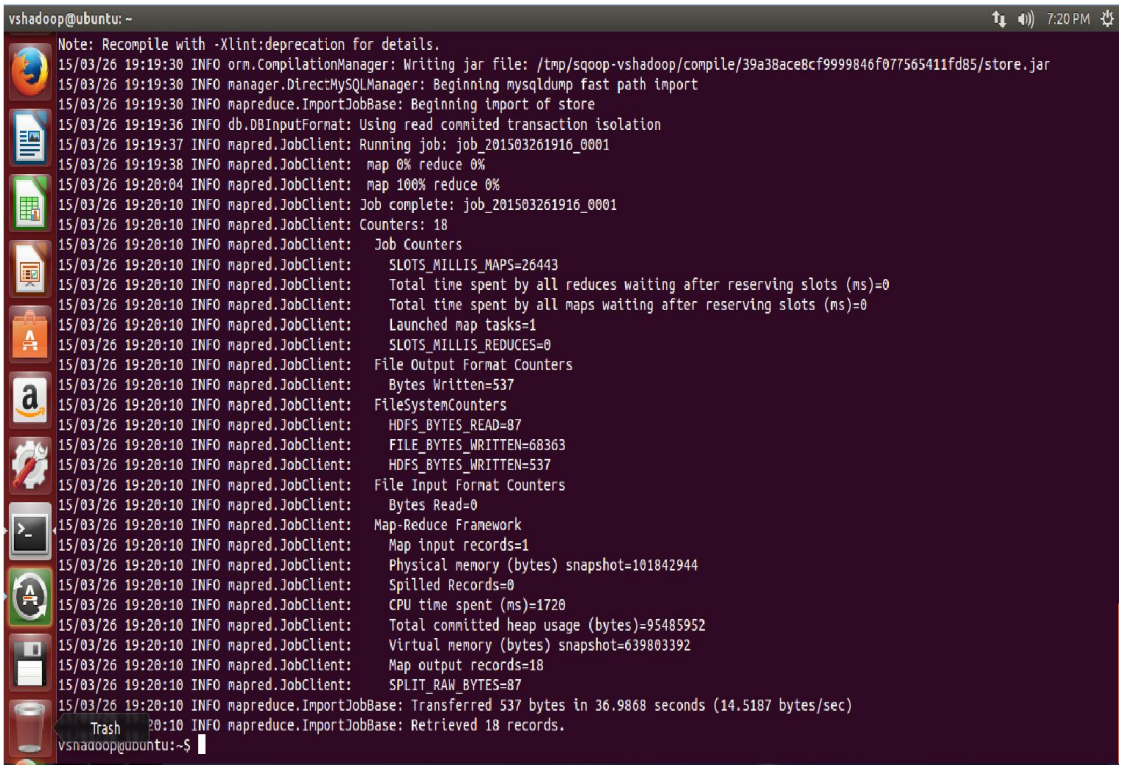
Page Size: 20 Total Rows: 18 Page: 1 of 1 Matching Rows:

#	user	id	serial
1	abhiram@adda.com	101	1
2	abhiram@adda.com	114	2
3	abhiram@adda.com	112	3
4	abhiram@adda.com	117	4
5	test@test.com	112	5
6	test@test.com	117	6
7	venkata.abhiram144@gmail.com	112	7
8	venkata.abhiram144@gmail.com	114	8
9	venkata.abhiram144@gmail.com	101	9
10	venkata.abhiram144@gmail.com	120	10
11	venkata.abhiram144@gmail.com	121	11
12	venkata.abhiram144@gmail.com	124	12
13	venkata.abhiram144@gmail.com	125	13
14	venkata.abhiram144@gmail.com	107	14
15	venkata.abhiram144@gmail.com	110	15

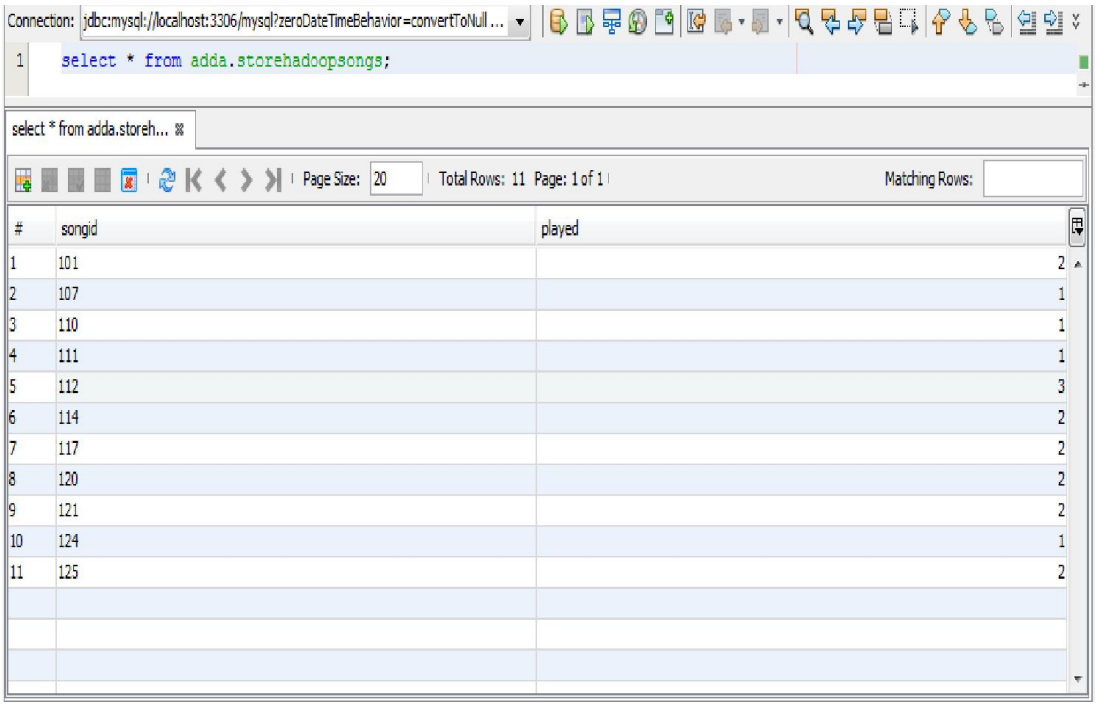
Snap shot of sample data of table store



Snap Shot of the website showing recent view history of users



Snapshot of Hadoop running analytics on files in HDFS



Snapshot of Hadoop storing results in the database

technical support information set of roughly 0.052 TB. This can be supported a mean decision size of sixty KB from four main decision centers, with associate calculable 1900 calls per week. The four node Hadoop cluster relies on artefact laborious ware with every node following an equivalent configuration: 6GB RAM; two TB hard drive; four MB Cache; one central processing unit Intel Core i5; two cores at two.20 GHz; aboard 100Mbps LAN; UNIX Ubuntu 14.04(LTS) with Hadoop v2.2.The analysis is mentioned with reference to the process performance of the particular parallelized bunch algorithms and conjointly with reference to the bunch accuracy. For each map job that included minimum of 30 records the time taken by the sqoop command to generate java file and bring it into HDFS is 36 seconds this time then varies linearly with the number of the records that has to be bought to HDFS. Then the analysis of the url frequency by the users has been analyzed by writing a customized counting mapper and reducer program in python using the Hadoop streaming feature. After this the data is

exported to the sql database using sqoop export command. The output of the agglomeration algorithms is currently examined yet because the accuracy of the clusters... the output of all thought of agglomeration algorithms for clusters together with support calls associated with "performance" problems. It may be noted that the results area unit quite similar for k-means, k-means with a pre-processing step exploitation cover agglomeration and fuzzy k-means. The amount of support calls enclosed within the clusters area unit comparable exploitation all techniques. The output of the probabilistic agglomeration algorithms is that the most applicable however given the prohibitory delays incurred with Dirichlet, the output of the LDA agglomeration formula is deemed the foremost acceptable with the relevant generated topics issues matched with those often encountered by VMware customers. However, it may be noted that current agglomeration output has limitations. The connectedness of the output is proscribed. This downside isn't associated with the planned system most because the methodology wont to generate the text vectors. At present, driver solely supports TF or TF-IDF coefficient (Owen *et al*. [21]). Though TD-IDF coefficient was wont to convert the support calls descriptions into vectors, stop words still appeared within the agglomeration results. Whereas not presently supported in driver, a lot of advanced techniques exist to remodel text documents into vectors is needed. As delineate in supervised coefficient ways like Gain magnitude relation supported applied math confidence intervals can be wont to improve the standard of the text vectors and, as a consequence, the agglomeration results. What is more it's necessary to contemplate collocation. Generate the text vectors. At present, driver solely supports TF or TF-IDF coefficient (Owen et al. [21]). Though TD-IDF coefficient was wont to convert the support calls descriptions into vectors, stop words still appeared within the agglomeration results. Whereas not presently supported in driver, a lot of advanced techniques exist to remodel text documents into vectors is needed. As delineate in supervised coefficient ways like Gain magnitude relation supported applied math confidence intervals can be wont to improve the standard of the text vectors and, as a consequence, the agglomeration results. What is more it's necessary to contemplate collocation.

RESULTS

This table represents the data which is collected from each user after a particular song is played User attribute is the identifier that uniquely identifies different users Id is the song id of the particular song

REFERENCES

Aberdeen Group: Unlocking Business Intelligence in the Contract Center.

Apache Hadoop http://hadoop.apache.org/

Apache Hbase http://hbase.apache.org/

Apache Hive http://hive.apache.org/ Esteves RM, Pais R, Rong C: K-means Clustering in the Cloud – A Mahout Test.

Apache Mahout https://mahout.apache.org/

Bhandarkar M: MapReduce programming with apache Hadoop.

Cisco Internet Business Solutions Group (IBSG): The Internet of Things: How the Next Evolution of thenternet is Changing Everything.

Digital Universe Study (on behalf of EMC Corporation): Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East.

Encyclopaedia of Computer Science and Technology 2000, 42:113-157

Ghemawat S, Gobioff H, Leung S: The Google File System.

http://idcdocserv.com/1414

http://www.cisco.com/web/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf

http://www.mckinsey.com/insights/business_technology/big_data_the_next_frontier_for_innovation

Huang DW, Sherman BT, Lempicki RA: Systematic and integrative analysis of large gene lists using DAVID Bioinformatics Resources.

IBM: The Essential CIO. http:// www935.ibm.com/ services/ uk/cio/pdf/ CIE03073-GBEN-01.pdf

IEEE 24th International Symposium on Parallel & Distributed Processing (IPDPS' 10) 2010.

IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST' 10) 2010. ACM pp 1–10. http://dl.acm.org/citation.cfm?id=1914427

IEEE International Conference on Advanced Information Networking and Applications (WAINA '11) 2011.

Karmasphere: Deriving Intelligence from Big Data in Hadoop: A Big Data Analytics Primer.

Karmasphere: Understanding the Elements of Big Data: More than a Hadoop Distribution.

McKinsey Global Institute: Big data: The next frontier for innovation, competition, and productivity.

Nat Protoc 2009, 4(1):44-57. PubMed Abstract | Publisher Full Text Lavrac N, Keravnou E, Zupan B: Intelligent data analysis in medicine.

Shvachko K, Kuang H, Radia S, Chansler R: The Hadoop Distributed File System.

SOSP '03 Proceedings of the 19th ACM symposium on Operating Systems Principles 2003.

****