Big Data? How do you run capacity planning?

Posted by **datumengineering** on February 15, 2013

Most of Datawarehouse folks are very much accustomed with the term "Capacity Planning", **Read Inmon**. This is widely used process for DBA's and Datawarehouse Architects. In an typical project of data management and warehouse wide variety of audience is involved to drive the capacity planning. It involves everyone from Business Analyst to Architect to Developer to DBA and finally Data Modeler.

This practice which has had wide audience in typical Datawarehouse world, how this has been driven in Big Data? I have hardly heard noise around this in any Hadoop driven project which had started with an intention to handle growing data. I have met pain bearers DBA/Architects who have been facing challenges at all stages of data management when data outgrows. They are the main players who advocates bringing Hadoop **ASAP.** Crux of their problem is not growing data. But the problem is, they didn't have mathematical calculation which advocate the growth rate. All we talk about is: How much percentage it is going? Most of the time that percentage also come from experience :)

Capacity planning should be explore more than just calculating the percentage and experience.

1. *It should be more mathematical calculation of every byte of the data sources coming into the system.*
2. *How about designing a predictive model which will confirm my data growth with an accuracy until 10 years?*
3. *How about involving business to confirm the data growth drivers and feasibility of future born data sources ?*
4. *Why don't consider compression factor and purging into the calculation to reclaim the space for data grow.*
5. *Why we consider only disk utilization and why there is no consideration about other hardware resources like memory, processor, cache? After all, it is all about data processing environment.*

- I think this list of consideration can still grow….

I know building robust Capacity planning is not a task of day or month. One to two year of time frame data is good enough to understand this trend and develop a algorithm around it. Consider 1-2 years as a learning data set and take some months of data as training data set and start analyzing the trend, start building the

model which can predict the growth after 3rd or 4th year. Because as per Datawarehouse gurus bleeding starts after 5th year age.

I'll leave up to you to design the solution and process for capacity capacity to claim your DATA as BIG DATA.

Remember, disk space is cheap but not the disk seek.

Posted in **Big Data**, **Hadoop** | Tagged: **Big Data**, **Hadoop** | **Leave a Comment »**

Data analysis drivers

Posted by **datumengineering** on February 11, 2013

I have been exploring data analysis and modeling techniques since months. There are lots of topics floating around in the space of data analysis like statistical modeling, predictive modeling. There have always been questions in mind which technique to choose? which is preferred way for data analysis? Some articles and lecture highlight machine learning or mathematical model over statistics modeling limitations. They mention mathematical modeling as a next step of accuracy and prediction. This kind of articles create more questions in mind of naive user.

Finally, i would thank to **coursera.org** for zero down this confusion and stating a clear picture of Data Analysis drivers. Now, things are pretty clear in terms of How to proceed on data analysis? Rather, defining "DATA ANALYSIS DRIVERS". In one liner the answer is simple "**Define a question or problem**". So, all depend upon how you define the problem.

To start with data analysis drivers here are steps in a data analysis

1. *Define the question*
2. *Define the ideal data set*
3. *Determine what data you can access*
4. *Obtain the data*
5. *Clean the data*
6. *Exploratory data analysis*
7. *Statistical prediction/modeling*
8. *Interpret results*
9. *Challenge results*
10. *Synthesize/write up results*
11. *Create reproducible code*

- Defining the question means how the business problem has stated and how you proceed on story telling on this problem. Story telling on the problem will take you to the structuring the solution. So you should be good in story telling on the problem statement.
- Defining the solution will help you to prepare the data (data set) for the solution.
- Profile the source to identify what data you can access.
- Next step is cleansing the data.
- Now, once the data is cleansed it is either in one of the following standard: txt, csv, xml/html, json and database.
- Based on the solution need we start building the model. Precisely, the solution will have requirement of Descriptive analysis, Inferential analysis or predictive analysis.

Henceforth, The data set and model may depend on your goal:

1. Descriptive – a whole population.
2. Exploratory – a random sample with many variables measured.
3. Inferential – the right population, randomly sampled.
4. Predictive – a training and test data set from the same population.
5. Causal – data from a randomized study.
6. Mechanistic – data about all components of the system.

From here knowledge on statistics, machine learning and mathematical algorithm works :)

Posted in **Big Data**, **Data Analysis** | Tagged: **Big Data**, **Data Analysis** | **2 Comments »**

Data flow: Web log analysis on a Hive-way

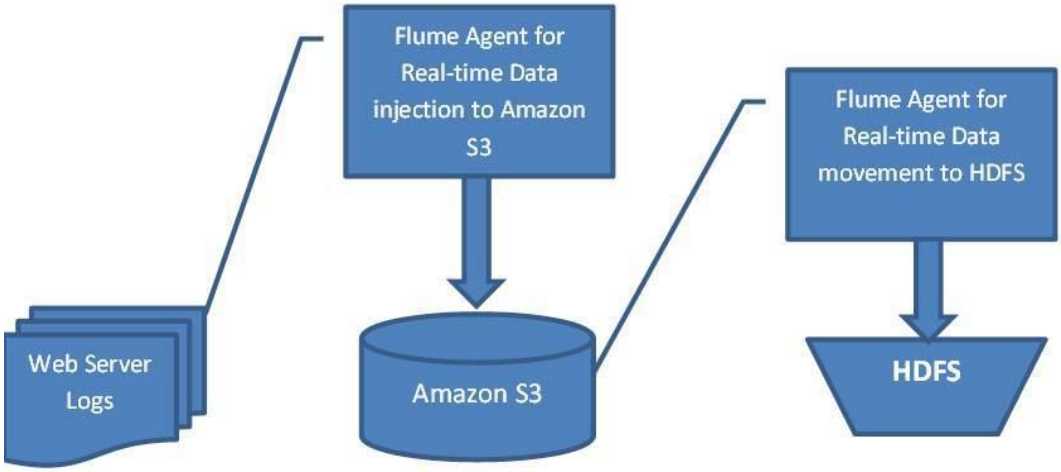
Posted by **datumengineering** on February 8, 2013

Data flow design to get an insight of user behavior on web site. Data flow explains the method of flattening up all elements in web log which can support detail user analysis and behavior.

Technology & Skills: Hadoop-Hive, HiveQL (+ Rich set of UDF in HiveQL) .

Infrastructure: Amazon Web Services (AWS).

Process -1 Moving data from Web Server to Amazon Simple Storage Services (S3) to HDFS.



Process -2 Start EC2 instance type : small to run Map Reduce job to parse log file.

To run jobs on AWS we should have EBS and EC2 both instance running.

Process -3 Prepare for Elastic Map Reduce to run the jobs from command line.

To run the EMR from command line we use an Amazon EMR credentials file to simplify job flow creation and authentication of requests. The credentials file provides information required for many commands. The credentials file is a convenient place to store command parameters so you don't have to repeatedly enter the information. The Amazon EMR CLI automatically looks for these credentials in the file credentials.json.

To install the Elastic MapReduce CLI1. Navigate to your elastic-mapreduce-cli directory.
2. Unzip the compressed file: Linux and UNIX users, from the command-line prompt, enter the following:$ unzip elastic-mapreduce-ruby.zip**Configuring Credentials**
The Elastic MapReduce credentials file can provide information required for many commands. It is convenient to store command parameters in the file to save you from the trouble of repeatedly entering the information. Your credentials are used to calculate the signature value for every request you make. Elastic MapReduce automatically looks for your credentials in the file credentials.json. It is convenient to edit the credentials.json file and include your AWS credentials. An AWS key pair is a security credential similar to a password, which you use to securely connect to your instance when it is running.*To create your credentials file:*1. Create a file named credentials.json in the elastic-mapreduce-cli/elastic-mapreduce-ruby directory.2. Add the following lines to your credentials file:

```
{
"access_id": "[Your AWS Access Key ID]",
"private_key": "[Your AWS Secret Access Key]",
"keypair": "[Your key pair name]",
"key-pair-file": "[The path and name of your PEM file]",
"log_uri": "[A path to a bucket you own on Amazon S3, such as, s3n://myloguri/]",
```

“region”: “[The Region of your job flow, either us-east-1, us-west-2, uswest-1, eu-west-1, ap-northeast-1, ap-southeast-1, or sa-east-1]“ }Note the name of the Region. You will use this Region to create your Amazon EC2 key pair and your Amazon S3 bucket.

Process -4 Prepare Hive table for data analysis. Create landing table to load log data.

We create schema for tokenizing the string. So MAP and COLLECTION is used to build key-value array.

CREATE TABLE logdata (

C_2 STRING,

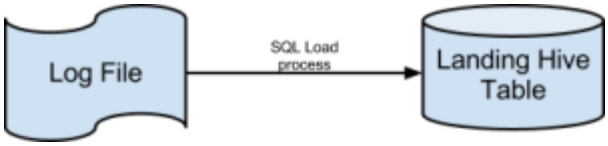
C_3 MAP<STRING, STRING>,

C_4 STRING,

C_21 STRING)

ROW FORMAT DELIMITED FIELDS TERMINATED BY ‘ ‘ COLLECTION ITEMS TERMINATED BY ’73’ MAP KEYS TERMINATED BY ‘=’ STORED AS textfile;

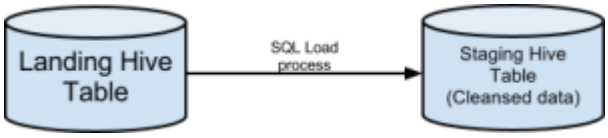
Process -6 Load Hive landing table with log file data from HDFS.



LOAD DATA INPATH ‘hdfs://10.130.86.181:9000/input/log.txt’ OVERWRITE INTO TABLE `logdata`;

Process -7 Load Hive stage table from landing table.

This stage table will have the data from landing. Stage table is used to load cleansed data without any junk character (Log has some # characters which we remove when load into staging).

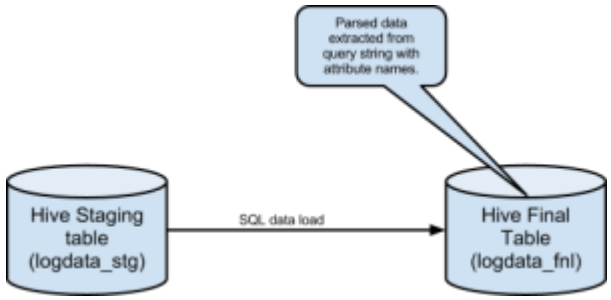


create table logdata_stg

comment ‘log data’ stored as sequencefile as

*select * from logdata where C_0 not like '%#%';*

Process -8 Load Hive final table from staging table.



This process will create flatten structure of complete log file into final table. This table will be used in all over the analysis. This table is created with actual column names identified in the log file. Final table load happen using UDF to parse query string, host name and category tree in browse data.

Source: http://datumengineering.wordpress.com/2013/02/