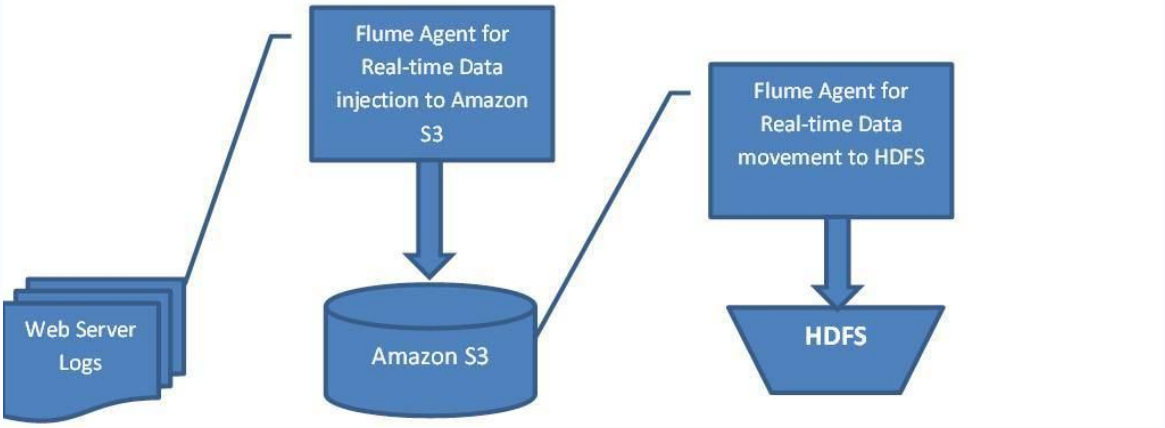
Data flow: Web log analysis on a Hive-way

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.

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.

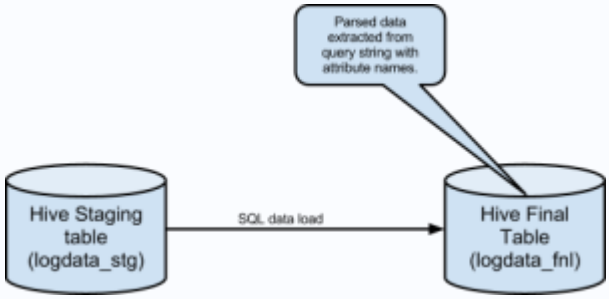
This stage table will have the data from landing. Stage table is used to load cleansed data without any junk character (Log has some # characters which we remove when load into staging).

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/2013/02/08/data-flow-web-behavior-analysis-in-a-hive-way/