syncsort

DMX-h ETL Use Case Accelerator

Web Log Aggregation

Table of Contents

1 Introduction

Web log aggregation is a process in which web log data is aggregated down into a smaller, more easily analyzed dataset. This use case accelerator implements a web log aggregation in Hadoop that counts the number of occurrences of each unique URL, using a single DMX-h ETL aggregation task.

The use case accelerators are developed as standard DMExpress jobs with just minor accommodations that allow them to be run in or outside of Hadoop:

- When specified to run in the Hadoop cluster, DMX-h Intelligent Execution (IX) automatically converts them to be executed in Hadoop using the optimal Hadoop engine, such as MapReduce. In this case, some parts of the job may be run on the Linux edge node if not suitable for Hadoop execution.
- Otherwise, they are run on a Windows workstation or Linux edge node, which is useful for development and testing purposes before running in the cluster.

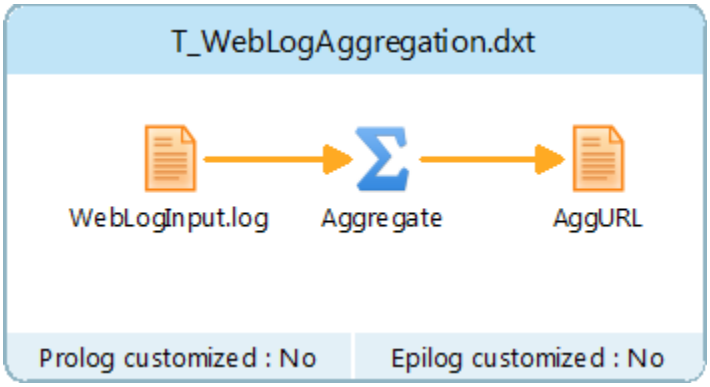
While the method presented in the next section is the simplest and most efficient way to develop this example, the more complex user-defined MapReduce solution is provided as a reference in Appendix A.

For guidance on setting up and running the examples both outside and within Hadoop, see Guide to DMExpress Hadoop Use Case Accelerators.

For details on the requirements for IX jobs, see “Developing Intelligent Execution Jobs” in the DMExpress Help.

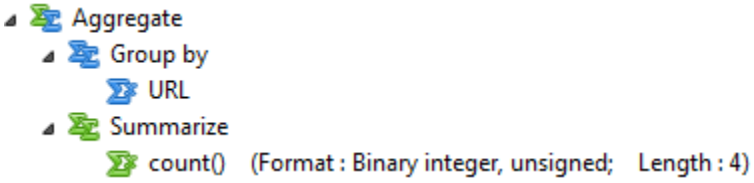
2 Web Log Aggregation with DMX-h IX

The web log aggregation solution in DMX-h ETL consists of a job, J_WebLogAggregation.dxj, with a single aggregation task.

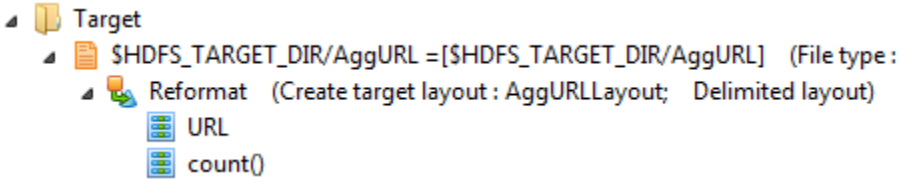


2.1 T_WebLogAggregation Task

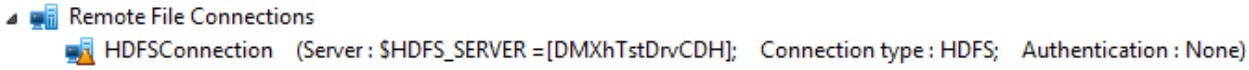
In this aggregation task, the web log data in the HDFS source WebLogInput.log is cleansed of comments (rows that start with '#') via a conditional source filter. It is then grouped by the URL field and summarized by the count of unique URL's in the Aggregate dialog. Changing the format to "binary integer, unsigned 4 bytes" improves performance over default format of decimal unsigned.



The HDFS target, AggURL, is reformatted to include only the URL and the count of unique occurrences.



The HDFS source and target files are defined with a remote file connection to the Hadoop cluster.



Appendix A Web Log Aggregation with DMX-h User-defined MapReduce

This user-defined MapReduce solution is provided as a reference in the event that particular knowledge of your application's data would benefit from manual control of the MapReduce process.

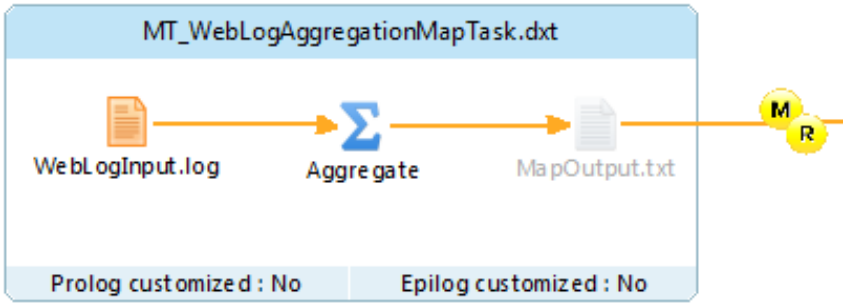
The Web Log Aggregation job in DMX-h contains a map step and a reduce step, separated by a MapReduce data flow connector between the map task and the reduce task as follows:

The map step reads the web log input file and counts the number of occurrences of each record's URL after filtering out the comments from the web log data. It then assigns partition IDs to each record such that all records with the same key will go to the same reducer.

The reduce step groups by the URL as the key, and totals the counts per URL produced by the mapper.

A.1 Map Step

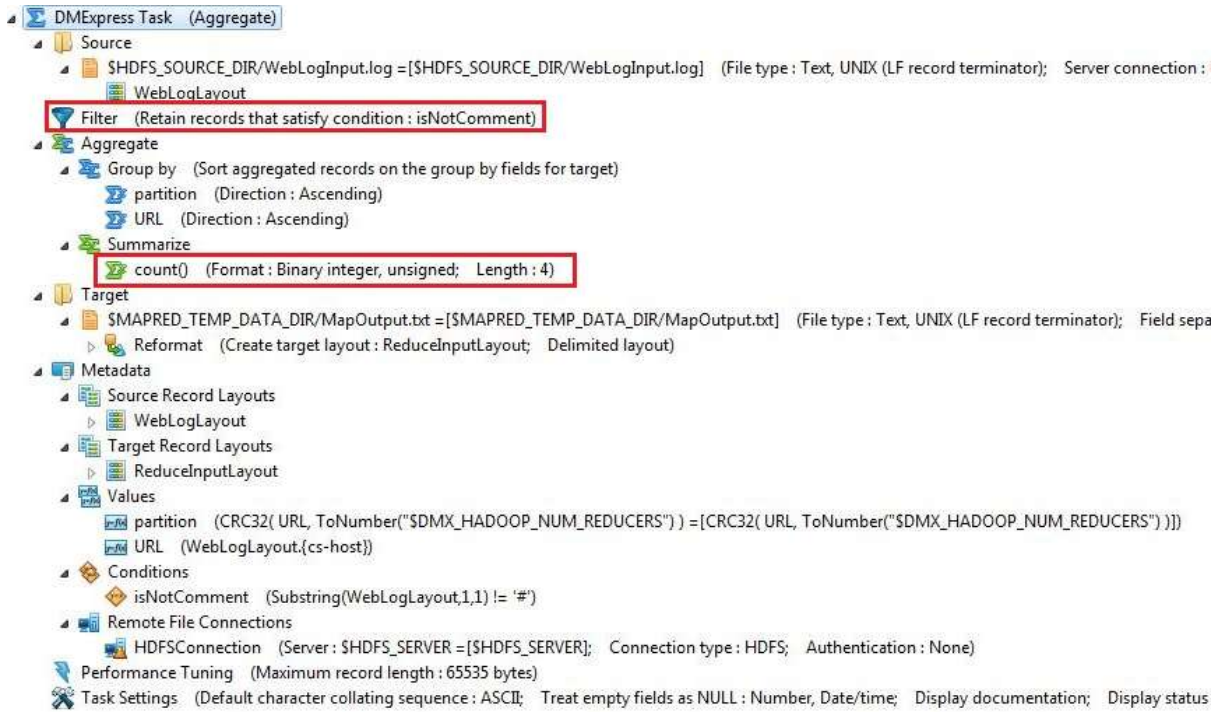
The Web Log Aggregation map step in DMX-h consists of one aggregation task that counts the number of times a URL occurs in a given input file and a filter that further reduces the amount of data that goes to the reducer by removing the comments. A partition ID is also added to the beginning of each record to direct records with the same key to the same reducer.



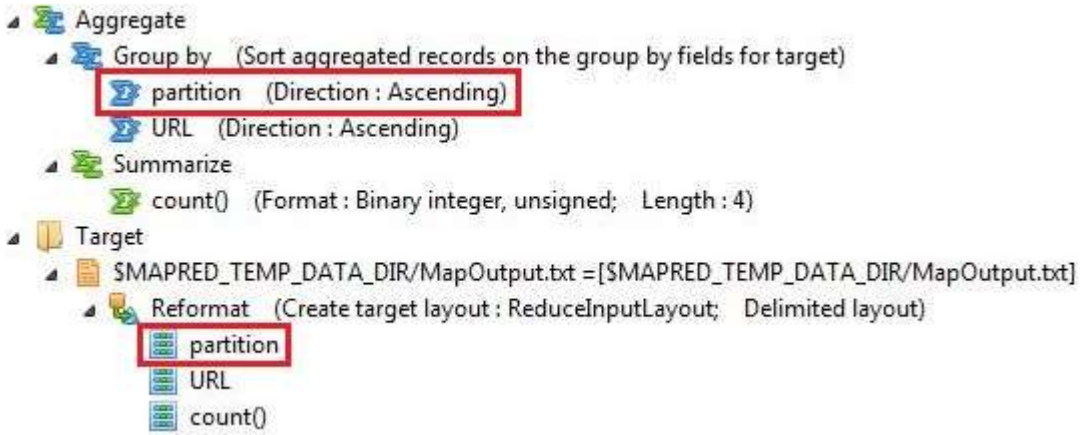
To optimize performance, the first phase of aggregation occurs in the map step as the aggregation reduces the amount of data going over the network to the reducer. As the mappers are passed only a subset of the data, the aggregation here is partial.

A.1.1 MT_WebLogAggregationMapTask.dxt

This task filters comments from the web log based on the occurrence of '#'. It then counts the number of occurrences of each URL. The count format is set to 4-byte unsigned binary integer to improve performance.



The task further assigns the partition ID (partition) as the first field in each target record and as the first field of the group by fields.



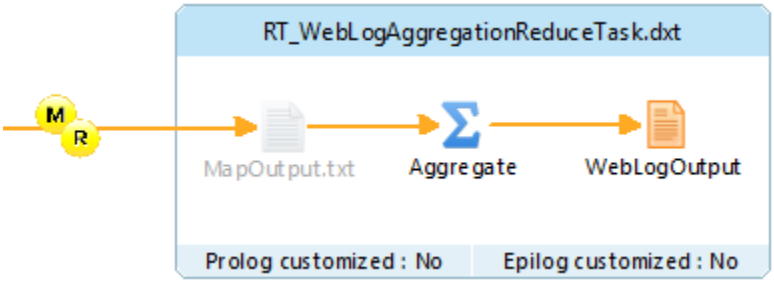
All records that share the same key must go to the same reducer. The partition ID determines the reducer to which the data goes. In this case, the records with the same URL key value must have the same partition ID; thus, we create a CRC32() hash value based on this key.

In addition to the value to be hashed, the CRC32() function allows you to provide a number to determine a range of hash values. For partition, the range is defined as the environment variable DMX_HADOOP_NUM_REDUCERS, which provides the number of reducers invoked for your MapReduce job. As a result, the CRC32() function returns a number from 0 to the value of DMX_HADOOP_NUM_REDUCERS.

The data must be ordered based on this partition ID so that the framework can properly distribute the data to the correct reducer. We include the partition ID as the first group by field and sort records based on the group by fields to ensure the data is in the correct sorted order.

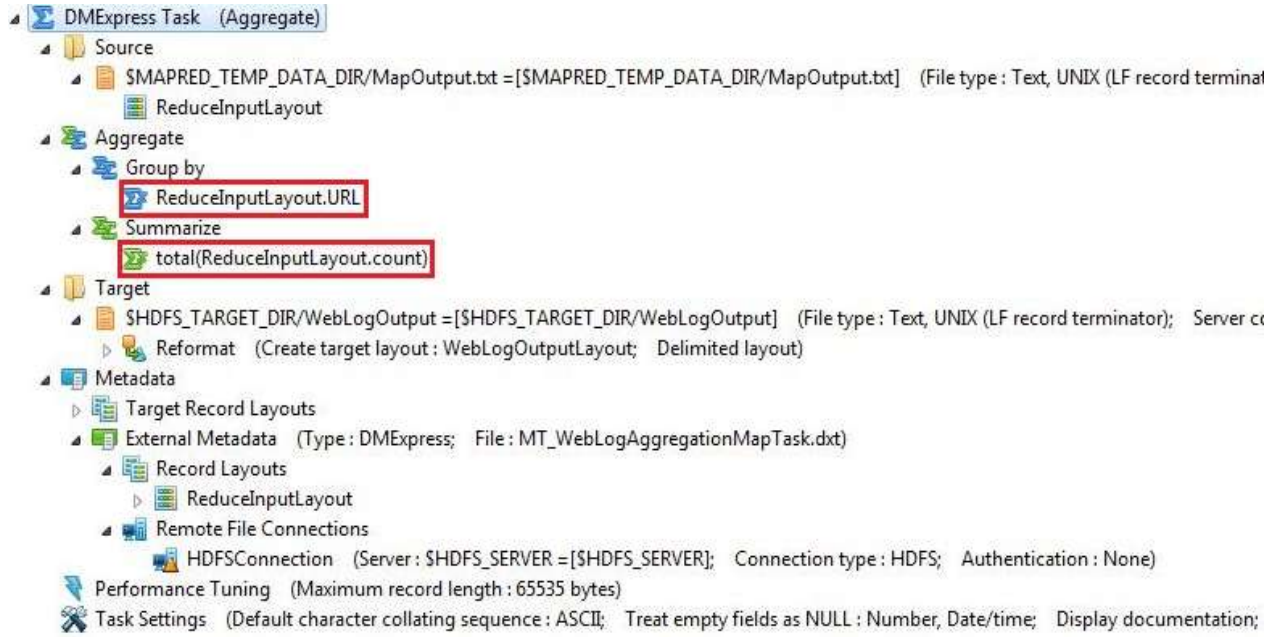
A.2 Reduce Step

The Web Log Aggregation reduce step in DMX-h ETL consists of one aggregate task that groups by the URL as the key and totals the counts per URL produced by the mapper.



A.2.1 RT_WebLogAggregationReduceTask.dxt

This task groups each record by the URL as the aggregate key and then totals the counts per URL provided by the mapper. The resulting output produced by this task is the number of times each URL occurred in the set of web logs.


DMExpress Task (Aggregate)
Source
$MAPRED_TEMP_DATA_DIR/MapOutput.txt =[$MAPRED_TEMP_DATA_DIR/MapOutput.txt] (File type : Text, UNIX (LF record terminat
ReduceInputLayout
Aggregate
Group by
ReduceInputLayout.URL
Summarize
total(ReduceInputLayout.count)
Target
$HDFS_TARGET_DIR/WebLogOutput =[$HDFS_TARGET_DIR/WebLogOutput] (File type : Text, UNIX (LF record terminator); Server c
Reformat (Create target layout : WebLogOutputLayout; Delimited layout)
Metadata
Target Record Layouts
External Metadata (Type : DMExpress; File : MT_WebLogAggregationMapTask.dxt)
Record Layouts
ReduceInputLayout
Remote File Connections
HDFSConnection (Server : $HDFS_SERVER =[$HDFS_SERVER]; Connection type : HDFS; Authentication : None)
Performance Tuning (Maximum record length : 65535 bytes)
Task Settings (Default character collating sequence : ASCII; Treat empty fields as NULL : Number, Date/time; Display documentation;

About Syncsort

Syncsort provides enterprise software that allows organizations to collect, integrate, sort, and distribute more data in less time, with fewer resources and lower costs. Thousands of customers in more than 85 countries, including 87 of the Fortune 100 companies, use our fast and secure software to optimize and offload data processing workloads. Powering over 50% of the world's mainframes, Syncsort software provides specialized solutions spanning "Big Iron to Big Data", including next gen analytical platforms such as Hadoop, cloud, and Splunk. For more than 40 years, customers have turned to Syncsort's software and expertise to dramatically improve performance of their data processing environments, while reducing hardware and labor costs. Experience Syncsort at www.syncsort.com.

Syncsort Inc. ***50 Tice Boulevard, Suite 250, Woodcliff Lake, NJ 07677*** ***201.930.8200***