Answers

1. Why Map-reduce program is needed in Pig Programming?

Ans: pig uses MapReduce to execute all of its data processing. It compiles the Pig Latin scripts that users write into a series of one or more MapReduce jobs that it then executes.

2. What are advantages of pig over MapReduce?

Ans: Pig Latin provides all of the standard data-processing operations, such as join, filter, group by, order by, union, etc. MapReduce provides the group by operation directly (that is what the shuffle plus reduce phases are), and it provides the order by operation indirectly through the way it implements the grouping. Filter and projection can be implemented trivially in the map phase. But other operators, particularly join, are not provided and must instead be written by the user.

Pig provides some complex, nontrivial implementations of these standard data operations.

In MapReduce, the data processing inside the map and reduce phases is opaque to the system. This means that MapReduce has no opportunity to optimize or check the user’s code. Pig, on the other hand, can analyze a Pig Latin script and understand the data flow that the user is describing. That means it can do early error checking (did the user try to add a string field to an integer field?) and optimizations (can these two grouping operations be combined?).

In MapReduce, the data processing inside the map and reduce phases is opaque to the system. This means that MapReduce has no opportunity to optimize or check the user’s code. Pig, on the other hand, can analyze a Pig Latin script and understand the data flow that the user is describing. That means it can do early error checking (did the user try to add a string field to an integer field?) and optimizations (can these two grouping operations be combined?).

3. What is pig engine and what is its importance ?

Ans: Pig provides an engine for executing data flows in parallel on Hadoop.It includes a language, *Pig Latin*, for expressing these data flows. Pig Latin includes operators for many of the traditional data operations (join, sort, filter, etc.), as well as the ability for users to develop their own functions for reading, processing, and writing data.

4. What are the modes of Pig execution?

Ans: There are two modes of Pig execution:-Local mode and MapReduce Mode.

5. What is grunt shell in Pig?

Ans: *Grunt* shell is Pig’s interactive shell. It enables users to enter Pig Latin interactively and provides a shell for users to interact with HDFS. Grunt provides command-line history and editing, as well as Tab completion.

6. What are the features of Pig Latin language?

Ans: Pig Latin fall into three separate categories: traditional extract transform load (ETL) data pipelines, research on raw data, and iterative processing.

The largest use case is data pipelines. A common example is web companies bringing in logs from their web servers, cleansing the data, and precomputing common aggregates before loading it into their data warehouse.

Traditionally, ad-hoc queries are done in languages such as SQL that make it easy to quickly form a question for the data to answer. However, for research on raw data, some users prefer Pig Latin. Because Pig can operate in situations where the schema is unknown, incomplete, or inconsistent, and because it can easily manage nested data, researchers who want to work on data before it has been cleaned and loaded into the warehouse often prefer Pig. Researchers who work with large data sets often use scripting languages such as Perl or Python to do their processing. Users with these backgrounds often prefer the dataflow paradigm of Pig over the declarative query paradigm of SQL.

Users building iterative processing models are also starting to use Pig. Consider a news website that keeps a graph of all news stories on the Web that it is tracking. In this graph each news story is a node, and edges indicate relationships between the stories. For example, all stories about an upcoming election are linked together. Every five minutes a new set of stories comes in, and the data-processing engine must integrate them into the graph. Some of these stories are new, some are updates of existing stories, and some supersede existing stories. Some data-processing steps need to operate on this entire graph of stories. For example, a process that builds a behavioral targeting model needs to join user data against this entire graph of stories. Rerunning the entire join every five minutes is not feasible because it cannot be completed in five minutes with a reasonable amount of hardware. But the model builders do not want to update these models only on a daily basis, as that means an entire day of missed serving opportunities.

To cope with this problem, it is possible to first do a join against the entire graph on a regular basis, for example, daily. Then, as new data comes in every five minutes, a join can be done with just the new incoming data, and these results can be combined with the results of the join against the whole graph. This combination step takes some care , as the five-minute data contains the equivalent of inserts, updates, and deletes on the entire graph. It is possible and reasonably convenient to express this combination in Pig Latin.

7. Is Pig Latin commands case sensitive?

Ans: Keywords in Pig Latin are not case-sensitive; for example, LOAD is equivalent to load.

8. What is a data flow language?

Pig Latin is a dataflow language. This means it allows users to describe how data from one or more inputs should be read, processed, and then stored to one or more outputs in parallel.