* 1. What is the difference between inputsplit and Blcok size in Hadoop/Hive?

Block size is a physical concept . Its an actual storage 258MB.

Input split is a locaical concept used for Data processing.

2) Can We set increase/decrease the no of mappers in hive program?

Yes, by defaults mapper = no of blocks . There are certain input file parameters which can be set in the hive programs to change the default behavior and increase/decrease mappers.

3) Hive Performance -

1) Merging small files can reduce the number of maps, but it will increase network IO.

2) Try to make the split block size and hdfs block size close to each other, to avoid multiple hdfs blocks on a split block size being located on different data nodes, thereby reducing network IO.

3)

* 1. What map and reduce does?

Map -> does transformations

Reduce -> Aggregations

* 1. Difference between Managed table and External Table ?

For External Tables ,Hive does not move the data into its warehouse directory. If the external table is dropped, then the table metadata is deleted but not the data.

For Internal tables , Hive moves data into its warehouse directory. If the table is dropped, then the table metadata and the data will be deleted.

* 1. What is Hive ?

Hive was created to allow non-programmers familiar with SQL to work with petabytes of data, using a SQL-like interface called HiveQL. Traditional relational databases are designed for interactive queries on small to medium datasets and do not process huge datasets well. Hive instead uses batch processing so that it works quickly across a very large distributed database. Hive transforms HiveQL queries into MapReduce or Tez jobs that run on [Apache Hadoop’s](https://aws.amazon.com/emr/details/hadoop/what-is-hadoop/) distributed job scheduling framework, Yet Another Resource Negotiator (YARN). It queries data stored in a distributed storage solution, like the Hadoop Distributed File System (HDFS) or [Amazon S3](https://aws.amazon.com/s3/). Hive stores its database and table metadata in a metastore, which is a database or file backed store that enables easy data abstraction and discovery.

Hive includes HCatalog, which is a table and storage management layer that reads data from the Hive metastore to facilitate seamless integration between Hive, Apache Pig, and MapReduce. By using the metastore, HCatalog allows Pig and MapReduce to use the same data structures as Hive, so that the metadata doesn’t have to be redefined for each engine. Custom applications or third party integrations can use WebHCat, which is a RESTful API for HCatalog to access and reuse Hive metadata.

* 1. Can hive Queries data from [Amazon S3](https://aws.amazon.com/s3/)?

Yes , it can query on top of S3.

* 1. What is Hcatcalog and what are its significance ?

Hive includes HCatalog, which is a table and storage management layer that reads data from the Hive metastore to facilitate seamless integration between Hive, Apache Pig, and MapReduce. By using the metastore, HCatalog allows Pig and MapReduce to use the same data structures as Hive, so that the metadata doesn’t have to be redefined for each engine. Custom applications or third party integrations can use WebHCat, which is a RESTful API for HCatalog to access and reuse Hive metadata.

* 1. Why to use S3 over HDFS?

You want very cheap storage

You don't have a long running Hadoop cluster with HDFS

You want to access the data anytime and from anywhere

<https://community.cloudera.com/t5/Support-Questions/which-way-is-the-best-when-using-hive-to-analyse-S3-data/td-p/167074>

* 1. What are the different ways to load a table from relational DB to S3)

Use Sqoop - from relation dbs to s3/hdfs . By default 4 mappers are created , can be adjsted.

Use Spark/MR to connect to RD API process the data and load to S3/Hdfs. Apachi Nifi can also do the same job.

Use informatica/ETL tool or python script -> Extract to local file -> localfile to HDFS.

* 1. Does hive/spark on s3 is faster than hive/spark on HDFS?
  2. Performance will be optimized when you colocate storage and compute - i.e. store in HDFS on a cluster using direct attached storage and access via Spark or Hive. In this type of architecture Spark or Hive on HDFS will outperform Spark or Hive on S3. The data is not traveling far when reading from disk and loading into memory. On the other hand, there are economies of scale with S3 (object store) that outweigh the performance benefits of a block storage file system like HDFS. Doing analysis directly on the data in HDFS will give better performance

* 1. What are the ways S3 data can be queries using SQL.

Create EC2 instance and install hive and connect to S3 using hive tables

Using Amazon Athena can be connected directly to S3 without Hadoop . This uses presto.

* 1. Hdfs is designed for high throughput and for batch processing and not for latency and interactive processing.
  2. When MapReduce is useful?

MR is useful when we need to analyze large datasets spread across machines . MR is suited for process and Write once and Read many times of applications.

* 1. What is Hbase and Can Hbase use S3 as storage or by default HDFS storage only needs to be used?

A distributed, column-oriented database. HBase uses HDFS for its underlying

storage, and supports both batch-style computations using MapReduce and point

queries (random reads).

HBase can use any filesystem supported in Hadoop, including S3 but if you do not use EMR it will be too slow.

For using S3 as the datasource, you need to configure your Hadoop filesystem to be s3. cloudera did this package of using s3 as storage for Hbase.

* 1. Scenorio where you used Hbase in your application?
  + Transaction lookup for Random reads.

19 ) what is Yarn and what is its general usage ? Can we run Hadoop appplications withot Yarn?

We need yarn(resource manger) if we are using different processing (MR/spark/hive/hbase) in the same cluster.

Spark and MR jobs can be run without Yarn and submit to cluster . Only issue is the resource management will be there . If you are able to able the resource management inside application code you can use without Yarn. Yarn streamlines the process of resource management ensuring all applications are queuing correctly.

* + Explain how the Map and reduce steps working in MR

Map process the steps and load to the Local disk(not hdfs).

Reduce steps process the local files and aggregate accordingly and load to HDFS.

* 1. What is the advantage of combiners in Hive/MR? when to use and what are their usage ?

Combiner is not a replacement of reducer. It acts as a reduces within the same node. Based on the keys it combines the tasks(map) within the same node. There by reducing the data shuffled in the shuffle phase. Combiner might not be suitable for high cardinality map keys.

22)MSCK REPAIR is used to synch Hive Metastore with the HDFS data.

* 1. What is Serde in Hive and why its needed?

When we want Hive to process our own files as tables (external tables) we have to let him know - how to translate data in files into records. This is exactly the role of SerDe. You can see it as plug-in which enables Hive to read / write your data.

For example - you want to work with CSV. Here is example of CSV\_Serde

* 1. Why we need compression Files ? Advantages and Disadvanatges .

General trade off for compression -

higher the compression rate ,less the speed.

higher the speed , less the compression and more space.

**Advantages** -

Less Space

**Disadvantages** -

Increase in CPU load due to encode/decode.

The different tools have very different compression characteristics. Gzip is a general purpose compressor, and sits in the middle of the space/time trade-off. Bzip2 compresses more effectively than gzip, but is slower. Bzip2’s decompression speed is faster than its compression speed, but it is still slower than the other formats. LZO and Snappy, on the other hand, both optimize for speed and are around an order of magnitude faster than gzip, but compress less effectively. Snappy is also significantly faster than LZO for decompression.

* 1. IO bound and CPU bound -

Reasons to compress:

a) Data is mostly stored and not frequently processed. It is usual DWH scenario. In this case space saving can be much more significant then processing overhead

b) Compression factor is very high and thereof we save a lot of IO.

c) Decompression is very fast (like Snappy) and thereof we have a some gain with little price

d) Data already arrived compressed

Reasons not to compress

a) Compressed data is not splittable. Have to be noted that many modern format are built with block level compression to enable splitting and other partial processing of the files. b) Data is created in the cluster and compression takes significant time. Have to be noted that compression usually much more CPU intensive then decompression.

c) Data has little redundancy and compression gives little gain.

* 1. High level of Kafka ?

Kafka, a system developed at LinkedIn, is essentially a messaging system that is designed to support aggregation of high throughput log messages arriving from different applications.

In Kafka a Topic is the container with which messages are associated.

One Topic has many partitons(unit of parallelism) .

Producer put the message to Topic and Kafka internally split to partitions and loads . Subscriber will subscribe to the topic and receive the message as needed (can be set synchronously and asynchronously )

Kafka Broker -> Broker needs servers. Kafka have cluster of machines(brockers).

Kafka only guarantees at-least-once delivery. In most cases a message is delivered exactly once to its consumers. It also guarantees that messages from a single partition are delivered to a consumer in order. Across partitions no such guarantee is made.

The user's events on your web or app can be logged by your Web sever and then sent to Kafka broker through the producer. In producer, you could specific the partition method, for example : event type (different event is saved in different partition) or event time (partition a day into different period according your app logic) or user type or just no logic and balance all logs into many partitions.

About your case in question, you can create one topic called "page-view-event", and create N partitions through hash keys to distribute the logs into all partitions evenly. Or you could choose a partition logic to make log distributing by your spirit.

* 1. Do we have any relation to CPU and RAM(Memory)?

Not necessarily . CPU is for processing/operation and RAM is for Storage.

|  |  |
| --- | --- |
| 6 down vote | RAM is used to save data. CPU time is used to process data.  There is no relationship between CPU and memory usage. A process can occupy all CPUs of a system but use only a minimal amount of memory. Also, a process can allocate all memory available on a system but only use minimal CPU time. So there is no relation between both. |

No relationship. Think of it in terms of doing some heavy-duty number-crunching. RAM is how many of those numbers can be stored for faster retrieval, whereas CPU is how fast those numbers can have mathematical operations performed on them.

For how one affects the other, let's assume that you have a slow CPU but lots of RAM. That means that you can store quite a few of those numbers in memory - about half a billion of them on a typical 32 bit machine. But processing/computation/etc on those numbers will be slower.

On the other hand if you have a fast CPU but a small amount of RAM, processing of the numbers will be fast, but from time to time the machine will need to drag them from disk (and put them back again to make room for more), which is many times slower than taking them from RAM.

So really, **each affects the other,** and for best performance you need both to be good. However, this totally depends on what each application is actually *doing*, and there are other factors (such as disk speed, processor caches, etc) which we needn't go into too much detail about for now, but which you should be aware will complicate things.

* 1. What is parquet predicate pushdown?

Predicate pushdown deals with what values will be scanned and not what columns. So, if you apply filter on column A to only return records with value V, the predicate push down will make parquet read only blocks that may contain values V. Parquet holds min/max statistics in several levels, and it will compare the value V to the those min/max headers, and only scan blocks where min/max contains the value V. This is for predicate push down.

* 1. What is vectorization in hive ?

Vectorized query execution is a Hive feature that greatly reduces the CPU usage for typical query operations like scans, filters, aggregates, and joins. A standard query execution system processes one row at a time. This involves long code paths and significant metadata interpretation in the inner loop of execution. Vectorized query execution streamlines operations by processing a block of 1024 rows at a time. Within the block, each column is stored as a vector (an array of a primitive data type). Simple operations like arithmetic and comparisons are done by quickly iterating through the vectors in a tight loop, with no or very few function calls or conditional branches inside the loop. These loops compile in a streamlined way that uses relatively few instructions and finishes each instruction in fewer clock cycles, on average, by effectively using the processor pipeline and cache memory.https://docs.cloudera.com/documentation/enterprise/6/6.3/topics/hive\_query\_vectorization.html

* 1. difference between hive metastore and hiveserver2

Diagram

Description automatically generated

<https://docs.cloudera.com/documentation/enterprise/5-8-x/topics/cdh_ig_hive_metastore_configure.html>

**What is HiveServer :** It is a service that allows a remote client to submit requests to hive. Using a variety of programming languages, and retrieve results.

What is Hive Service : Hive Service is nothing but daemon which runs on your client node which sends requests to Hive Server.

Relation between all these:

* + The Thrift-based Hive service is the core of HS2 and responsible for servicing the Hive queries (e.g., from Beeline). In simple terms Hive server is based on thrift protocols which sends queries from hive client i.e., your command line interface or from HUE interface to the underlying data which can be in your HDFS or any other data sources.

Usage:

* + When you query any hive tables or database, in background automatically your requests is transferred between hive service and hive server

* + Hive Metastore -

Diagram

Description automatically generated

* 1. Difference between Hive CLI and beeline CLI?

1.cli accesses metadata through metaServer directly.

* 1. Beeline accesses metadata through hiverserver2
  2. Cli uses thirt based API
  3. Beeline use jdbc client api.

* 1. What is Tez and their Usage ?

<https://devstacks.wordpress.com/2017/04/06/apache-tez-vs-spark/>

Tez can run only on Yarn which is kind of disadvantage.

Tez backward compatibility is really an advantage (you just need to change the config in existing hive codes).

Tez is a DAG-based system, it's aware of all opération in such a way that it optimizes these operations before starting execution.

MapReduce model simply states that any computation can be performed by two kinds of computation steps – a map step and a reduce step. One pair of map and reduce does one level of aggregation over the data. Complex computations typically require multiple such steps.

<https://www.sjsu.edu/people/robert.chun/courses/CS259Fall2013/s3/F.pdf> - very good

<https://www.infoq.com/articles/apache-tez-saha-murthy/>

* 1. When to use hive external table ?

Use external tables when files are already present or in remote locations(azure/aws storage), and the files should remain even if the table is dropped.

* 1. When to use msck repair table?
  2. Use MSSCK table - when the new partitions are added and the hive metastore is not updated. This ismainly for external table. I believe , internal table partition add will be added to the metastore automatically.

Use the MSCK REPAIR TABLE command to update the metadata in the catalog after you add Hive compatible partitions.

The MSCK REPAIR TABLE command scans a file system such as Amazon S3 for Hive compatible partitions that were added to the file system after the table was created. MSCK REPAIR TABLE compares the partitions in the table metadata and the partitions in S3. If new partitions are present in the S3 location that you specified when you created the table, it adds those partitions to the metadata and to the Athena table

* 1. What is the difference between hive on spark and spark sql?

In a nutshell, with Hive on Spark, your queries are optimized by Hive optimizer and finally a Spark job is executed

Thats what Hive on Spark is. Runnning Hive queries on Spark.

Coming to SparkSQL, It is part of the Spark core framework which also runs on the spark core. Using Sparksql we can create structured data frames that we use in spark and also we can invoke Hive by creating Hivecontext. Once Hivecontext is created, just like a link to the hive metastore, you can access and query tables in Hive.

* 1. When hive sort\_by clause is used?

Sort\_by is used while loading the table . The reducers will arrange the data and load the table so when we select the table sorted values will be used for predicate pushdown.

* 1. How to increase the no of mappers

<https://cloudera.ericlin.me/2015/05/how-to-control-the-number-of-mappers-required-for-a-hive-query/>

Increase the min/max split size to increase the mappers

org.apache.hadoop.hive.ql.io.CombineHiveInputFormat will combine the small files whose size is less than the split value mentioned. But the combining might not work if the blocks are in different data nodes.

* 1. What are the disadvanatge of small files issues which you can think of ?

Namenode overload

No of mapper increase, increase tasks start up time

Same no of files generated after processing small files (without reducers).

Storage is wasted.

* 1. What are the parameters will be checked in the logs ?

<https://www.bbvaaifactory.com/self-service-performance-tuning-for-hive/>

Number of tasks

Avg number of bytes per tasks

Avg task run time

* 1. What is the reason for generating small files ?

Excessive no of reducers processing small amount of data

High cardinality partitions (which in turn generate increase number of partitions).

* 1. What are the paramters needs to be set for mapper and reducer memory increase?

<https://www.bbvaaifactory.com/self-service-performance-tuning-for-hive/>

Check above

* 1. Skew Join parameters.

* 1. What is Hive Metastore and Hcatclog ?

<https://www.linkedin.com/pulse/hive-metastore-hcatalog-hcat-haotian-zhang/>

* 1. Default Hive metadata store is Derby . This has certain limitation like only one session can open the metastore. We can change the Metadata to Mysql.
  2. HCatalog is a tool for accessing metadata that reside in Hive metastore. It acts as an API to expose the metastore as REST interface to external tools such as Pig:
  3. Diagram

     Description automatically generated
  4. In short Hcatalog, used to expose the Hive Metastore details to other big data applications.
  5. Every mapreduce tool has its own notion about HDFS data (example Pig sees the HDFS data as set of files, Hive sees it as tables). With having table based abstraction, HCatalog supported mapreduce tools do not need to care about where the data is stored, in which format and storage location (HBase or HDFS).
  6. Diagram

     Description automatically generated