1. **What is Hbase?**

   Hbase is a column-oriented database management system which runs on top of HDFS (Hadoop Distribute File System). Hbase is not a relational data store, and it does not support structured query language like SQL.

   In Hbase, a master node regulates the cluster and region servers to store portions of the tables and operates the work on the data.

2. **What are the advantages of Hbase?**

   - High capacity storage system
   - Distributed design to cater large tables
   - Column-Oriented Stores
   - Horizontally Scalable
   - High performance & Availability
   - Base goal of Hbase is millions of columns, thousands of versions and billions of rows
   - Unlike HDFS (Hadoop Distribute File System), it supports random real time CRUD operations

3. **What are the key components of Hbase?**

   - **Zookeeper:** It does the co-ordination work between client and Hbase Maser
   - **Hbase Master:** Hbase Master monitors the Region Server
   - **RegionServer:** RegionServer monitors the Region
   - **Region:** It contains in memory data store(MemStore) and Hfile.
   - **Catalog Tables:** Catalog tables consist of ROOT and META

4. **Explain what does Hbase consists of?**

   - Hbase consists of a set of tables

- And each table contains rows and columns like traditional database
- Each table must contain an element defined as a Primary Key
- Hbase column denotes an attribute of an object

5. **How many operational commands are in Hbase?**

   Operational command in Hbases are five types

   - Get
   - Put
   - Delete
   - Scan
   - Increment

6. **What is WAL and Hlog in Hbase?**

   WAL (Write Ahead Log) is similar to MySQL BIN log; it records all the changes occur in data. It is a standard sequence file by Hadoop and it stores HLogkey's. These keys consist of a sequential number as well as actual data and are used to replay not yet persisted data after a server crash. So, in cash of server failure WAL work as a life-line and retrieves the lost data's.

7. **When you should use Hbase?**

   - **Data size is huge:** When you have tons and millions of records to operate
   - **Complete Redesign:** When you are moving RDBMS to Hbase, you consider it as a complete re-design then mere just changing the ports
   - **SQL-Less commands:** You have several features like transactions; inner joins, typed columns, etc.
   - **Infrastructure Investment:** You need to have enough cluster for Hbase to be really useful

8. **In Hbase what is column families?**

Column families comprise the basic unit of physical storage in Hbase to which features like compressions are applied.

9. **Explain what is the row key?**

   Row key is defined by the application. As the combined key is pre-fixed by the rowkey, it enables the application to define the desired sort order. It also allows logical grouping of cells and make sure that all cells with the same rowkey are co-located on the same server.

10. **Explain deletion in Hbase? Mention what are the three types of tombstone markers in Hbase?**

    When you delete the cell in Hbase, the data is not actually deleted but a tombstone marker is set, making the deleted cells invisible. Hbase deleted are actually removed during compactions.

    Three types of tombstone markers are there:

    - Version delete marker: For deletion, it marks a single version of a column
    - Column delete marker: For deletion, it marks all the versions of a column
    - Family delete marker: For deletion, it marks of all column for a column family

11. **Explain how does Hbase actually delete a row?**

    In Hbase, whatever you write will be stored from RAM to disk, these disk writes are immutable barring compaction. During deletion process in Hbase, major compaction process delete marker while minor compactions don't. In normal deletes, it results in a delete tombstone marker- these delete data they represent are removed during compaction.

    Also, if you delete data and add more data, but with an earlier timestamp than the tombstone timestamp, further **Gets** may be masked by the delete/tombstone marker and hence you will not receive the inserted value until after the major compaction.

12. Explain what happens if you alter the block size of a column family on an already occupied database?

When you alter the block size of the column family, the new data occupies the new block size while the old data remains within the old block size. During data compaction, old data will take the new block size. New files as they are flushed, have a new block size whereas existing data will continue to be read correctly. All data should be transformed to the new block size, after the next major compaction.

13. Mention the difference between Hbase and Relational Database?

Hbase	Relational Database
• It is schema-less • It is a column-oriented data store • It is used to store de-normalized data • It contains sparsely populated tables • Automated partitioning is done in Hbase	• It is a schema based database • It is a row-oriented data store • It is used to store normalized data • It contains thin tables • There is no such provision or built-in support for partitioning

14. What is the history of HBase?

2006: BigTable paper published by Google.

2006 (end of year): HBase development starts.

2008: HBase becomes Hadoop sub-project.

2010: HBase becomes Apache top-level project.

15. What is Apache HBase?

Apache Hbase is one the sub-project of Apache Hadoop,which was designed for NoSql database(Hadoop Database),bigdata store and a distributed,

scalable.Use Apache HBase when you need random, realtime read/write access to your Big Data.A table which contain billions of rows X millions of columns -atop clusters of commodity hardware. Apache HBase is an open-source, distributed, versioned, non-relational database modeled after Google's Bigtable. Apache HBase provides Bigtable-like capabilities run on top of Hadoop and HDFS.

16. What is NoSql?

Apache HBase is a type of "NoSQL" database. "NoSQL" is a general term meaning that the database isn't an RDBMS which supports SQL as its primary access language, but there are many types of NoSQL databases: BerkeleyDB is an example of a local NoSQL database, whereas HBase is very much a distributed database. Technically speaking, HBase is really more a "Data Store" than "Data Base" because it lacks many of the features you find in an RDBMS, such as typed columns, secondary indexes, triggers, and advanced query languages, etc.

17. What are the main features of Apache HBase?

Apache HBase has many features which supports both linear and modular scaling,HBase tables are distributed on the cluster via regions, and regions are automatically split and re-distributed as your data grows(Automatic sharding).HBase supports a Block Cache and Bloom Filters for high volume query optimization(Block Cache and Bloom Filters).

18. What is the difference between HDFS/Hadoop and HBase?

HDFS doesn't provides fast lookup records in a file,IN Hbase provides fast lookup records for large table.

19. Is there any difference between HBase datamodel and RDBMS datamodel?

In Hbase,data is stored as a table(have rows and columns) similar to RDBMS but this is not a helpful analogy. Instead, it can be helpful to think of an HBase table as a multi-dimensional map.

20. What are key terms are used for designing of HBase datamodel?

- table(Hbase table consists of rows)
- row(Row in hbase which contains row key and one or more columns with value associated with them)

- column(A column in HBase consists of a column family and a column qualifier, which are delimited by a : (colon) character)
- column family(having set of columns and their values,the column families should be considered carefully during schema design)
- column qualifier(A column qualifier is added to a column family to provide the index for a given piece of data)
- cell(A cell is a combination of row, column family, and column qualifier, and contains a value and a timestamp, which represents the value's version)
- timestamp(represents the time on the RegionServer when the data was written, but you can specify a different timestamp value when you put data into the cell)

21. What are datamodel operations in HBase?

- Get(returns attributes for a specified row,Gets are executed via HTable.get)
- put(Put either adds new rows to a table (if the key is new) or can update existing rows (if the key already exists). Puts are executed via HTable.put (writeBuffer) or HTable.batch (non-writeBuffer))
- scan(Scan allow iteration over multiple rows for specified attributes)
- Delete(Delete removes a row from a table. Deletes are executed via HTable.delete)

HBase does not modify data in place, and so deletes are handled by creating new markers called tombstones. These tombstones, along with the dead values, are cleaned up on major compaction.

22. How should filters are useful in Apache HBase?

Filters In Hbase Shell,Filter Language was introduced in APache HBase 0.92. It allows you to perform server-side filtering when accessing HBase over Thrift or in the HBase shell.

23. How many filters are available in Apache HBase?

Total we have 18 filters are support to hbase.They are:

ColumnPrefixFilter

TimestampsFilter

PageFilter

MultipleColumnPrefixFilter

FamilyFilter

ColumnPaginationFilter

SingleColumnValueFilter

RowFilter

QualifierFilter

ColumnRangeFilter

ValueFilter

PrefixFilter

SingleColumnValueExcludeFilter

ColumnCountGetFilter

InclusiveStopFilter

DependentColumnFilter

FirstKeyOnlyFilter

KeyOnlyFilter

24. How can we use MapReduce with HBase?

Apache MapReduce is a software framework used to analyze large amounts of data, and is the framework used most often with Apache Hadoop. HBase can be used as a data source, TableInputFormat, and data sink, TableOutputFormat or MultiTableOutputFormat, for MapReduce jobs. Writing MapReduce jobs that read or write HBase, it is advisable to subclass TableMapper and/or TableReducer.

25. How do we back up my HBase cluster?

There are two broad strategies for performing HBase backups: backing up with a full cluster shutdown, and backing up on a live cluster. Each approach has pros and cons.

1)Full Shutdown Backup

Some environments can tolerate a periodic full shutdown of their HBase cluster, for example if it is being used a back-end analytic capacity and not serving front-end web-pages. The benefits are that the NameNode/Master are RegionServers are down, so there is no chance of missing any in-flight changes to either StoreFiles or metadata. The obvious con is that the cluster is down.

2)Live Cluster Backup

live clusterbackup-copytable:copy table utility could either be used to copy data from one table to another on the same cluster, or to copy data to another table on another cluster.

live cluster backup-export:export approach dumps the content of a table to HDFS on the same cluster.

26. Does HBase support SQL?

Not really. SQL-ish support for HBase via Hive is in development, however Hive is based on MapReduce which is not generally suitable for low-latency requests.By using Apache Phoenix can retrieve data from hbase by using sql queries.

27. What are the different commands used in Hbase operations?

There are 5 atomic commands which carry out different operations by Hbase.

Get, Put, Delete, Scan and Increment.

28. How to connect to Hbase?

A connection to Hbase is established through Hbase Shell which is a Java API.

29. What is the role of Master server in Hbase?

The Master server assigns regions to region servers and handles load balancing in the cluster.

30. What is the role of Zookeeper in Hbase?

The zookeeper maintains configuration information, provides distributed synchronization, and also maintains the communication between clients and region servers.

31. **When do we need to disable a table in Hbase?**

    In Hbase a table is disabled to allow it to be modified or change its settings. .When a table is disabled it cannot be accessed through the scan command.

32. **Give a command to check if a table is disabled.**

    Hbase > is_disabled “table name”

33. **What does the following table do?**

    hbase > disable_all 'p.*'

    The command will disable all the tables starting with the letter p

34. **Name three disadvantages Hbase has as compared to RDBMS?**

    - Hbase does not have in-built authentication/permission mechanism
    - The indexes can be created only on a key column, but in RDBMS it can be done in any column.
    - With one HMaster node there is a single point of failure.

35. **Is Hbase a scale out or scale up process?**

    Hbase runs on top of Hadoop which is a distributed system. Haddop can only scale uo as and when required by adding more machines on the fly. So Hbase is a scale out process.

36. **What are the steps in writing something into Hbase by a client?**

    In Hbase the client does not write directly into the HFile. The client first writes to WAL(Write Access Log), which then is accessed by Memdtore. The Memstore Flushes the data into permanent memory from time to time.

37. **What are catalog tables in Hbase?**

    The catalog tables in Hbase maintain the metadata information. They are named as −ROOT− and .META. The −RROT− table stores information about location of .META> table and the .META> table holds information about all regions and their locations.

38. **What is compaction in Hbase?**

As more and more data is written to Hbase, many HFiles get created. Compaction is the process of merging these HFiles to one file and after the merged file is created successfully, discard the old file.

39. What are the different compaction types in Hbase?

There are two types of compaction. Major and Minor compaction. In minor compaction, the adjacent small HFiles are merged to create a single HFile without removing the deleted HFiles. Files to be merged are chosen randomly.

In Major compaction, all the HFiles of a column are emerged and a single HFiles is created. The delted HFiles are discarded and it is generally triggered manually.

40. What is the difference between the commands delete column and delete family?

The Delete column command deletes all versions of a column but the delete family deletes all columns of a particular family.

41. What is the role of the class HColumnDescriptor in Hbase?

This class is used to store information about a column family such as the number of versions, compression settings, etc. It is used as input when creating a table or adding a column.

42. What is the lower bound of versions in Hbase?

The lower bound of versions indicates the minimum number of versions to be stored in Hbase for a column. For example If the value is set to 3 then three latest version wil be maintained and the older ones will be removed.

43. What is TTL (Time to live) in Hbase?

TTL is a data retention technique using which the version of a cell can be preserved till a specific time period.Once that timestamp is reached the specific version will be removed.

44. Does Hbase support table joins?

Hbase does not support table jons. But using a mapreduce job we can specify join queries to retrieve data from multiple Hbase tables.

45. What is a rowkey in Hbase?

Each row in Hbase is identified by a unique byte of array called rowkey.

46. What are the two ways in which you can access data from Hbase?

The data in Hbase can be accessed in two ways.

- Using the rowkey and table scan for a range of row key values.
- Using mapreduce in a batch manner.

47. What are the two types of table design approach in Hbase?

They are − (i) Short and Wide (ii) Tall and Thin

48. In which scenario should we consider creating a short and wide Hbase table?

The short and wide table design is considered when there is

- There is a small number of columns
- There is a large number of rows

49. In Which scenario should we consider a Tall-thin table design?

The tall and thin table design is considered when there is

- There is a large number of columns
- There is a small number of rows

50. Give a command to store 4 versions in a table rather than the default 3.

hbase > alter 'tablename', {NAME => 'ColFamily', VERSIONS => 4}

51. What does the following command do?

hbase > alter 'tablename', {NAME => 'colFamily', METHOD => 'delete'}

This command deletes the column family form the table.

52. Give the commands to add a new column family “(newcolfamily”) to a table (“tablename”) which has a existing column family(“oldcolfamily”). ?

Hbase > disable ‘tablename’

Hbase > alter ‘tablename’ {NAME => ‘oldcolfamily’,NAME=>’newcolfamily’}

Habse > enable ‘tablename’

53. What is the Hbase shell command to only 10 records form a table?

scan 'tablename', {LIMIT=>10,

STARTROW=>"start_row",

STOPROW=>"stop_row"}

54. What does the following command do?

major_compact 'tablename'

Run a major compaction on the table.

55. How does Hbase support Bulk data loading?

There are two main steps to do a data bulk load in Hbase.

- Generate Hbase data file(StoreFile) using a custom mapreduce job) from the data source. The StoreFile is created in Hbase internal format which can be efficiently loaded.
- The prepared file is imported using another tool like comletebulkload to import data into a running cluster. Each file gets loaded to one specific region.

56. How does Hbase provide high availability?

Hbase uses a feature called region replication. In this feature for each region of a table, there will be multiple replicas that are opened in different RegionServers. The Load Balancer ensures that the region replicas are not co-hosted in the same region servers.

57. What is HMaster?

The Hmaster is the Master server responsible for monitoring all RegionServer instances in the cluster and it is the interface for all metadata changes. In a distributed cluster, it runs on the Namenode.

58. What is HRegionServer in Hbase?

HRegionServer is the RegionServer implementation. It is responsible for serving and managing regions. In a distributed cluster, a RegionServer runs on a DataNode.

59. What are the different Block Caches in Hbase?

HBase provides two different BlockCache implementations: the default on-heap LruBlockCache and the BucketCache, which is (usually) off-heap.

60. How does WAL help when a RegionServer crashes?

The Write Ahead Log (WAL) records all changes to data in HBase, to file-based storage. if a RegionServer crashes or becomes unavailable before the MemStore is flushed, the WAL ensures that the changes to the data can be replayed.

61. Why MultiWAL is needed?

With a single WAL per RegionServer, the RegionServer must write to the WAL serially, because HDFS files must be sequential. This causes the WAL to be a performance bottleneck.

62. In Hbase what is log splitting?

When a region is edited, the edits in the WAL file which belong to that region need to be replayed. Therefore, edits in the WAL file must be grouped by region so that particular sets can be replayed to regenerate the data in a particular region. The process of grouping the WAL edits by region is called log splitting.

63. How can you disable WAL? What is the benefit?

WAL can be disabled to improve performance bottleneck.

This is done by calling the Hbase client field Mutation.writeToWAL(false).

64. When do we do manual Region splitting?

The manual region splitting is done we have an unexpected hotspot in your table because of many clients querying the same table.

65. What is a Hbase Store?

A Habse Store hosts a MemStore and 0 or more StoreFiles (HFiles). A Store corresponds to a column family for a table for a given region.

66. Which file in Hbase is designed after the SSTable file of BigTable?

The HFile in Habse which stores the Actual data(not metadata) is designed after the SSTable file of BigTable.

67. Why do we pre-create empty regions?

Tables in HBase are initially created with one region by default. Then for bulk imports, all clients will write to the same region until it is large enough to split and become distributed across the cluster. So, empty regions are created to make this process faster.

68. What is hotspotting in Hbase?

Hotspotting is asituation when a large amount of client traffic is directed at one node, or only a few nodes, of a cluster. This traffic may represent reads, writes, or other operations. This traffic overwhelms the single machine responsible for hosting that region, causing performance degradation and potentially leading to region unavailability.

69. What are the approaches to avoid hotspotting?

Hotspotting can be avoided or minimized by distributing the rowkeys across multiple regions. The different techniques to do this is salting and Hashing.

70. Why should we try to minimize the row name and column name sizes in Hbase?

In Hbase values are always freighted with their coordinates; as a cell value passes through the system, it'll be accompanied by its row, column name, and timestamp. If the rows and column names are large, especially compared to the size of the cell value, then indices that are kept on HBase storefiles (StoreFile (HFile)) to facilitate random access may end up occupying large chunks of the HBase allotted RAM than the data itself because the cell value coordinates are large.

71. What is the scope of a rowkey in Hbase?

Rowkeys are scoped to ColumnFamilies. The same rowkey could exist in each ColumnFamily that exists in a table without collision.

72. What is the information stored in hbase:meta table?

The Hbase:meta tables stores details of region in the system in the following format.

info:regioninfo (serialized HRegionInfo instance for this region)

info:server (server:port of the RegionServer containing this region)

info:serverstartcode (start-time of the RegionServer process containing this region)

73. What is a Namespace in Hbase?

A Namespace is a logical grouping of tables. It is similar to a database object in a Relational database system.

74. How do we get the complete list of columns that exist in a column Family?

The complete list of columns in a column family can be obtained only querying all the rows for that column family.

75. When the records are fetched form a Hbase tables, in which order are the sorted?

The records fetched form Hbase are always sorted in the order of rowkey-> column Family-> column qualifier-> tiestamp.

76. What is importance of ColumnFamily in Hbase?

A logical deviation of a data represented by a key is called column family. Virtually column families form dynamically based on data, which holds the multiple columns of related data. All column members of a column family have the same prefix. For example Vehicle is a column Maruthi, Tata, Hero are the sub column of the Vehicle. So here Vehicle is considered as column family.

Eg: Hbase > put ‘cars’, ‘price’, ‘ Vehicle:Maruthi’, ‘1,00,000’ // The syntax should be in order table, row, column family, value.

put ‘cars’, ‘price’, ‘Vehicle:Tata’,’2,00,000′

put ‘cars’, ‘price’,’Vehicle:Hero’,’3,00,000′

Here, cars is a table, Vehicle is a column family and 1,00,000 value.

77. Why Hbase instead of Hadoop?

Hbase suitable for low latency request, but mapreduce is high latency. Hadoop not support updates, but Hbase can support. Hadoop can store matadata only, but hbase can index the data.

78. What type of datatypes supports and not supports in Hbase?

Hbase has Put and Result interface which converts bytes and stored in an array as a value. So it can support any datatype like string, number, image or anything that can rendered as bytes. Typecasting is always possible.

79. What are different types of block cache?

Hbase provides 2 different block cache, such as onheap and offheap cache also called LruBlockCache (default) and bucketCache. Onheap cache is implemented from Java heap, where as bucketCache implemented from fileblock cache.

80. Elaborate very important commands in Hbase?

create ‘table’, ‘columnFamily’

put ‘table’, ‘rwo’, ‘columnFamily’, ‘value’

get’table’, ‘row’, ‘columnfamily’, ‘value’

scan ‘table’, ‘row’, ‘columnfamily’, ‘value’

list ‘tablename’

disable ‘table’

drop ‘table’

describe ‘table’

81. What is Memstore?

Menstore is a temporary repository in Hbase, which holds data inmemory modifications to the Store. It’s store Maximum HDFS block size data once reaches maximum size(64MB), it flushes the data into a HDFS.

82. What is DFSClient functions?

DFSCllient handels all remote server's interactions. It means to communicate with NameNode, Datanodes or JobTracker/YARN, required DFSClient. Hbase persists the data in HDFS via DFS client.

83. What is auto–sharding in Hbase?

Hbase dynamically distributes by the system when it is getting huge amount of data, this feature called autosharding.

84. What are Ulimit and nproc of Hbase?

ulimit is a upper bound of the process.

nproc can limiting the maximum number of processes available for a particular application, which restrict the processes.

85. What are Bloom Filters?

Bloom filters are filtering out blocks that you don't need which can save your disk and improve read latency.

86. What is the importance of MemStore and BlockCache?

Memory utilization and caching structures are too important in Hbase. To archive it's goal, HBase maintain two cache structures called MemStore and BlockCache. MemStore is a temporary repository and buffering in memory. Block cache keeps data blocks in memory after read.

87. What are different types of blocks in Hbase?

Block is a single smallest amount/unit of data. There are 4 type of veriets such as: Data, Meta, Index and Bloom. Data locks store User data. Index and Bloom blocks serve to speed up the read path. Index provides index of the particular Data blocks. Bloom block contain a bloom filter, that filters the data and display desired data quickly. Meta blocks store information about Hfile.

88. What are the core components in Hbase?

Hmaster serves one or more HRegion Servers.

Each HRegion Server serves one or more Region.

Each Region serves one Hlog and multiple Stores.

Each Store serves one MemStore and multiple StoreFile.

Each Store file has only one Hfile.

Each Hfile can hold 64kb of data.

89. How to write a file?

First the client written the data to HregionServer. First data stored data in (write ahead log) Hlog file, then the data is written to MemStore. Memstore temporary holds the data. If Memstore is full, it flush the data to Hfile. The data is ordered in Memstore and Hfile. Which is the temporary repository in the Hbase. Which persist the data on HDFS via DFS client.

90. If you forget syntax what should you do?

use help followed by command, for example, help 'scan'

91. When CRUD operations are not applicable?

When schema level updates / alternations done, it's not possible to run CRUD operations. To alter schema level updations first disable the table, it's mandatory.

92. How to open a connection in Hbase?

If you are going to open connection with the help of Java API, the following code provides the connection

Configuration myConf = HBaseConfiguration.create();

HTableInterface usersTable = new HTable(myConf, "users");

93. Are there any Schema Design examples?

Theres a very big difference between storage of relational/row-oriented databases and column-oriented databases. For example, if I have a table of users and I need to store friendships between these users… In a relational database my design is something like:

Table: users(pkey = userid) Table: friendships(userid,friendid,…) which contains one (or maybe two depending on how its impelemented) row for each friendship.

In order to lookup a given users friend, SELECT * FROM friendships WHERE userid = myid;

The cost of this relational query continues to increase as a user adds more friends. You also begin to have practical limits. If I have millions of users, each with many thousands of potential friends, the size of these indexes grow

exponentially and things get nasty quickly. Rather than friendships, imagine Im storing activity logs of actions taken by users.

In a column-oriented database these things scale continuously with minimal difference between 10 users and 10,000,000 users, 10 friendships and 10,000 friendships.

Rather than a friendships table, you could just have a friendships column family in the users table. Each column in that family would contain the ID of a friend. The value could store anything else you would have stored in the friendships table in the relational model. As column families are stored together/sequentially on a per-row basis, reading a user with 1 friend versus a user with 10,000 friends is virtually the same. The biggest difference is just in the shipping of this information across the network which is unavoidable. In this system a user could have 10,000,000 friends. In a relational database the size of the friendship table would grow massively and the indexes would be out of control.

94. Can you please provide an example of good de-normalization in HBase and how its held consistent (in your friends example in a relational db, there would be a cascadingDelete)? As I think of the users table: if I delete an user with the userid=123, do I have to walk through all of the other users column-family friends to guaranty consistency?! Is de-normalization in HBase only used to avoid joins? Our webapp doesnt use joins at the moment anyway.

You lose any concept of foreign keys. You have a primary key, thats it. No secondary keys/indexes, no foreign keys.

Its the responsibility of your application to handle something like deleting a friend and cascading to the friendships. Again, typical small web apps are far simpler to write using SQL, you become responsible for some of the things that were once handled for you.

Another example of good denormalization would be something like storing a users favorite pages. If we want to query this data in two ways: for a given user, all of his favorites. Or, for a given favorite, all of the users who have it as a favorite. Relational database would probably have tables for users, favorites, and userfavorites. Each link would be stored in one row in the userfavorites table. We would have indexes on both userid and favoriteid and

could thus query it in both ways described above. In HBase wed probably put a column in both the users table and the favorites table, there would be no link table.

That would be a very efficient query in both architectures, with relational performing better much better with small datasets but less so with a large dataset.

Now asking for the favorites of these 10 users. That starts to get tricky in HBase and will undoubtedly suffer worse from random reading. The flexibility of SQL allows us to just ask the database for the answer to that question. In a small dataset it will come up with a decent solution, and return the results to you in a matter of milliseconds. Now lets make that userfavorites table a few billion rows, and the number of users youre asking for a couple thousand. The query planner will come up with something but things will fall down and it will end up taking forever. The worst problem will be in the index bloat. Insertions to this link table will start to take a very long time. HBase will perform virtually the same as it did on the small table, if not better because of superior region distribution.

95. How would you design an Hbase table for many-to-many association between two entities, for example Student and Course?

By designing two tables:

Student: student id student data (name, address, …) courses (use course ids as column qualifiers here) Course: course id course data (name, syllabus, …) students (use student ids as column qualifiers here)

96. What is the maximum recommended cell size?

A rough rule of thumb, with little empirical validation, is to keep the data in HDFS and store pointers to the data in HBase if you expect the cell size to be consistently above 10 MB. If you do expect large cell values and you still plan to use HBase for the storage of cell contents, you'll want to increase the block size and the maximum region size for the table to keep the index size reasonable and the split frequency acceptable.

97. Why can't I iterate through the rows of a table in reverse order?

Because of the way HFile works: for efficiency, column values are put on disk with the length of the value written first and then the bytes of the actual value written second. To navigate through these values in reverse order, these length values would need to be stored twice (at the end as well) or in a side file. A robust secondary index implementation is the likely solution here to ensure the primary use case remains fast.

98. What happens if we change the block size of a column family on an already populated database?

When we change the block size of the column family, the new data takes the new block size while the old data is within the old block size. When the compaction occurs, old data will take the new block size. “New files, as they are flushed, will have the new block size, whereas existing data will continue to be read correctly. After the next major compaction, all data should be converted to the new block size.”

99. Is NoSQL follow relational DB model?

No.

100. Why would NoSQL be better than using a SQL Database? And how much better is it?

It would be better when our site needs to scale so massively that the best RDBMS running on the best hardware we can afford and optimize as much as possible simply can’t keep up with the load. How much better it is depends on the specific use case (lots of update activity combined with lots of joins is very hard on “traditional” RDBMSs) – could well be a factor of 1000 in extreme cases

101. What is the difference between HBase and Hive?

Hive doesn’t support record level operations but HBase support record level operations.

102. How Does HBase supports scalability and why RDBMS can’t?

HBase has many features which supports both linear and modular scaling. HBase clusters expand by adding RegionServers that are hosted on commodity class servers. If a cluster expands from 10 to 20 RegionServers, for example, it doubles both in terms of storage and as well as processing capacity.

RDBMS can scale well, but only up to a point – specifically, the size of a single database server – and for the best performance requires specialized hardware and storage devices.

103. What is S3?

S3 stands for simple storage service and it is a one of the file system used by hbase.

104. What is the use of get() method?

get() method is used to read the data from the table.

105. In how many modes Hbase can run?

There are two run modes of Hbase i.e. standalone and distributed.

106. Define standalone mode in Hbase?

It is a default mode of Hbase .In standalone mode, HBase does not use HDFS—it uses the local filesystem instead—and it runs all HBase daemons and a local ZooKeeper in the same JVM process.

107. What is decorating Filters?

It is useful to modify, or extend, the behavior of a filter to gain additional control over the returned data.

108. What is the full form of YCSB?

YCSB stands for Yahoo! Cloud Serving Benchmark.

109. What is the use of YCSB?

It can be used to run comparable workloads against different storage systems.

110. Which operating system is supported by Hbase?

Hbase supports those OS which supports java like windows, Linux.

111. What is the most common file system of Hbase?

The most common file system of HBase is HDFS i.e. Hadoop distributed file system

112. Define Pseudodistributed mode?

A pseudodistributed mode is simply a distributed mode that is run on a single host.

113. What is regionserver?

It is a file which lists the known region server names.

114. Which command is used to show the version?

Version command is used to show the version of hbase.

Syntax – hbase> version

115. What is use of tools command?

This command is used to list the hbase surgery tools.

116. What is the use of shutdown command?

It is used to shut down the cluster.

117. What is the use of truncate command?

It is used to disable, recreate and drop the specified tables.

118. Which command is used to run HBase Shell?

$ ./bin/hbase shell command is used to run the hbase shell.

119. Which command is used to show the current Hbase user?

whoami command is used to show Hbase user.

120. How to delete the table with the shell?

To delete table first disable it then delete it.

121. Define LZO?

Lempel-Ziv-Oberhumer (LZO) is a lossless data compression algorithm that is focused on decompression speed, and written in ANSIC.

122. What is use of InputFormat in MapReducr process?

InputFormat the input data, and then it returns a RecordReader instance that defines the classes of the key and value objects, and provides a next() method that is used to iterate over each input record.

123. What is Hbase Fsck?

HBase comes with a tool called hbck which is implemented by the HBaseFsck class. It provides various command-line switches that influence its behaviour.

124. What is REST?

Rest stands for Representational State Transfer which defines the semantics so that the protocol can be used in a generic way to address remote resources. It also provides support for different message formats, offering many choices for a client application to communicate with the server.

125. Define Thrift?

Apache Thrift is written in C++, but provides schema compilers for many programming languages, including Java, C++, Perl, PHP, Python, Ruby, and more.

126. What are the fundamental key structures of Hbase?

The fundamental key structures of Hbase are row key and column key.

127. What is JMX?

The Java Management Extensions technology is the standard for Java applications to export their status.

128. What is nagios?

Nagios is a very commonly used support tool for gaining qualitative data regarding cluster status. It polls current metrics on a regular basis and compares them with given thresholds.

129. What is the use of ZooKeeper?

The zookeeper is used to maintain the configuration information and communication between region servers and clients. It also provides distributed synchronization.

130. What is HBase Shell?

HBase shell is a java API by which we communicate with Hbase.

131. What the use of exists command?

exists command is used to check that the specified table is exists or not.

132. How do I upgrade Maven-managed projects from HBase 0.94 to HBase 0.96+?

In HBase 0.96, the project moved to a modular structure. Adjust your project's dependencies to rely upon the hbase-client module or another module as appropriate, rather than a single JAR. You can model your Maven depency after one of the following, depending on your targeted version of HBase. See Section 3.5, "Upgrading from 0.94.x to 0.96.x" or Section 3.3, "Upgrading from 0.96.x to 0.98.x" for more information.

Maven Dependency for HBase 0.98

org.apache.hbase

hbase-client

0.98.5-hadoop2

Maven Dependency for HBase 0.96

org.apache.hbase

hbase-client

0.96.2-hadoop2

Maven Dependency for HBase 0.94

org.apache.hbase

hbase

0.94.3

133. How can I troubleshoot my HBase cluster?

Always start with the master log (TODO: Which lines?). Normally it's just printing the same lines over and over again. If not, then there's an issue. Google or search-hadoop.com should return some hits for those exceptions you're seeing.

An error rarely comes alone in Apache HBase, usually when something gets screwed up what will follow may be hundreds of exceptions and stack traces coming from all over the place. The best way to approach this type of problem is to walk the log up to where it all began, for example one trick with RegionServers is that they will print some metrics when aborting so grapping for Dump should get you around the start of the problem.

RegionServer suicides are 'normal', as this is what they do when something goes wrong. For example, if ulimit and max transfer threads (the two most

important initial settings, see [ulimit] and dfs.datanode.max.transfer.threads) aren't changed, it will make it impossible at some point for DataNodes to create new threads that from the HBase point of view is seen as if HDFS was gone. Think about what would happen if your MySQL database was suddenly unable to access files on your local file system, well it's the same with HBase and HDFS. Another very common reason to see RegionServers committing seppuku is when they enter prolonged garbage collection pauses that last longer than the default ZooKeeper session timeout.