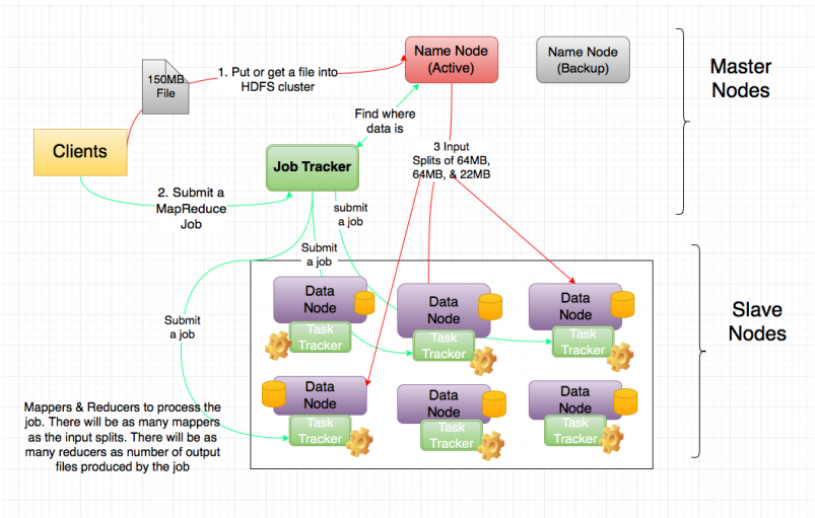
02: Q7 – Q15 Hadoop overview & architecture interview questions & answers

Posted on April 29, 2016 by Arulkumaran Kumaraswamipillai

This extends Q1 – Q6 Hadoop Overview & Architecture interview Q&As.

Q7. What are the major machine roles in a Hadoop cluster?
A7. The three major categories of machine roles in a Hadoop cluster are

1) Client machines.
2) Masters nodes.
3) Slave nodes.



Hadoop Cluster with Master & Slave Nodes

The **master nodes** manage the **slave nodes**. The client & master nodes will be comprised of more expensive high-end machines, and the slave nodes will be comprised of cheap commodity hardwares.

The client nodes are

neither master nor slave, and tasked with

1) loading the data into data nodes in the cluster, and

2) submitting jobs for processing the data,

3) and then retrieving the data after job completion.

The slave nodes are tasked with

a) storing lots of data on HDFS **data nodes**.

b) running parallel computations on all that data via the **Task Trackers**.

The master nodes are tasked with

a) overseeing and coordinating the data storage function on HDFS withe the **Name Node**.

16 Technical Key Areas

open all | close all

- Best Practice (6)
- Coding (26)
- Concurrency (6)
- Design Concepts (7)
- Design Patterns (11)
- Exception Handling (3
- Java Debugging (21)
- Judging Experience I
- Low Latency (7)
- Memory Managemen
- Performance (13)
- QoS (8)
- Scalability (4)
- SDLC (6)
- Security (13)
- Transaction Managen

80+ step by step Java Tutorials

open all | close all

- Setting up Tutorial (6)
- Tutorial - Diagnosis (2
- Akka Tutorial (9)
- Core Java Tutorials (2
- Hadoop & Spark Tuto
- JEE Tutorials (19)
- Scala Tutorials (1)
- Spring & HIbernate Tu
- Tools Tutorials (19)
- Other Tutorials (45)

b) overseeing and coordinating the parallel processing of data using Map Reduce with the **Job Tracker**.

Q8. What are edge nodes in a hadoop cluster?
A8. **Edge nodes** are the interface between the Hadoop cluster and the outside network. Edge nodes are used as staging area to ingest data into a Hadoop cluster with tools such as Sqoop, Pig, Flume, Kafka, Oozie (i.e. Workflow scheduler), etc. The management tools such as Hue (i.e. open source Web interface for analyzing data with any Apache Hadoop), Ambari (i.e. provisioning, managing, and monitoring Apache Hadoop clusters), etc.

Q9. What are some of the design decisions in extracting insights into user behavior by trawling through application log files? The insights to be extracted include page views per month, number of unique visits, average visit duration per user, checkouts per day, etc

A9. The key design decisions are:

1. **Storage Decision**: HDFS Vs HBase? File formats e.g. **Raw data** in Sequence File format, and then **columnar data** in Avro or Parquet format for efficient querying.

2. **Data ingestion decision**: How to **e**xtract the raw data from the log files & **l**oad the data into HDFS as say sequence files. This is basically the "**EL**" (i.e. Extract & Load) part of the **ELT** process. Flume streaming can be used for the data ingestion.

3. **Data wrangling decision**: refers to **t**ransforming the raw data into datasets so that they can be used for querying and reporting. This is the "**T**" in the "ELT" (i.e. Extract, Load & Transform) process. The transformation tasks include:

1) Data cleansing & validation. Cleansing includes deduplication of data, filtering out bad data & characters, etc.

2) Converting the raw data in sequence file format into columnar formats like Avro or Parquet. Selection of Avro or Parquet depends on the usage patterns. If you plan to use only 5 to 10 columns from a 50+ columns then Parquet is more suited. If you plan to use most of the columns then Avro format is more suited.

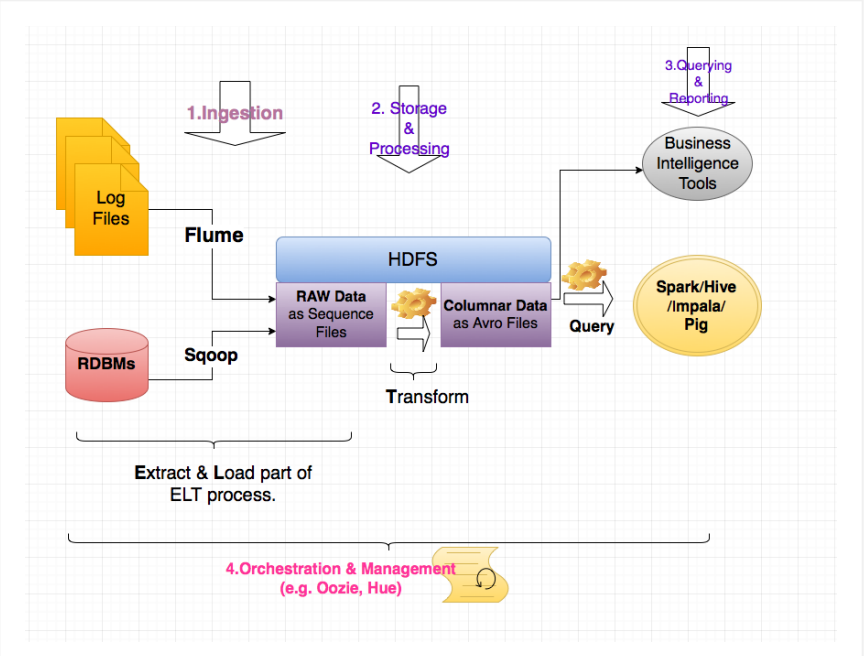
3) Enriching the data via RDBMs & Restful web service look-ups.

4) Grouping & aggregating the data to be able to extract pages visited by a single user, per location, etc. This grouping can also be accomplished by applying the schema **on read**.

4. **Querying & analyzing the data**: in columnar formats to extract insights into user behaviors. Tools like Spark, Impala, Hive, Pig, etc can be used.

5. **Orchestration and workflow management**: involving automation, coordination, monitoring, & management of the various ELT processes/tasks.

Q10. Can you draw the high level architectural diagram for the above BigData scenario in Q9?
A10. Here is a high level diagram showing the key components, stages and the tools that can be used.



Ingesting, Processing & Querying Log Files

Q11. When would you favor ELT over ETL?
A11. ETL architecture was discussed in detail at ETL architecture interview Q&As. **ELT** allows raw data to be loaded directly into the target and transformed there. This capability is very useful for processing large volumes of data (i.e. BigData). One of the key benefits of ELT is its reduction in load times relative to the ETL approach.

ELT is parallelized according to the data set, and disk I/O is usually optimized at the engine level for faster throughput. ELT scales better on commodity hardwares.

ELT is beneficial over ETL when:

1) There are big volumes of data.
2) There are structured, semi-structured & unstructured data to be processed from multiple streams in near real-time or real-time.

Often, you will be making use of hybrid architectures involving both ETL & ELT.

Q12. Why does hadoop have small files problem?
A12. Hadoop is designed for larger files. The default file split size i2 64MB. Smaller files can adversely affect performance as descibed below.

1) NameNode memory management: NameNode stores metadata about every directory, file, and block in Hadoop. As a rule of thumb, each block of metadata requires 150 to 200 bytes of memory. For example, If you have 0.5 billion block of metadata (i.e 0.5 billion files), it may require 150GB memory. This means the more memory occupation takes longer startup times when a NameNode restarts, as it must read the metadata of every file from a local disk. Also, during the normal operation, the NameNode must constantly track and check where every block of data is stored in the cluster. This is accomplished by listening for data nodes to report back on

all of their blocks of data. The more blocks a data node must report back on, the more network bandwidth it will consume. So, if you reduce the number of files to be stored, it will have lesser number of blocks of meta data to be stored.

2) MapReduce performance: The larger number of small files will degrade the performance of MapReduce, Hive, Pig, and Cascading Java/Scala performance due to a larger number of random disk IO. It is also better to have 200 map tasks working on each file of 128MB size as opposed to 2400 map tasks working on each file of 10MB size. The more mappers will require more nodes as usually 10 to 20 mappers run per node. So, larger files (e.g 128MB) will perform better than smaller files (e.g. 10MB).

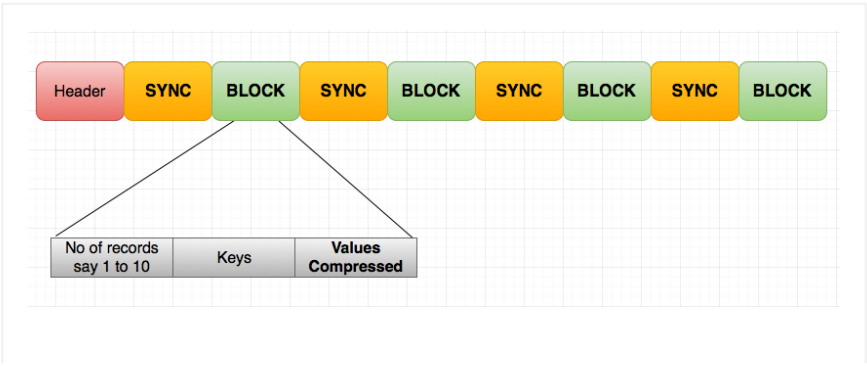
Q13. How would you go about dealing with smaller files in Hadoop?
A13. Hadoop is suited for larger files (i.e 64MB+). Smaller files put more load on the name nodes. So, how will you go about handling smaller files in Hadoop?

1. HAR (Hadoop ARchive Files)

HAR files work by building a layered filesystem on top of HDFS. An HAR file is created using the “hadoop archive” command.

2. Sequence Files

can be used to store the filename as the key and the file contents as the value. This is a better approach than HAR, as the sequence files are splittable for parallel processing and supports block compression.



Sequence Files are splittable & compressable at record or block level.
Stores data as key/value.

3. HBase

HBase stores data in MapFiles (i.e. indexed SequenceFiles) for random access.

Q14. What is the difference between **blocks** & **input splits**?
A14. Input splits are a **logical division** of your data whereas HDFS blocks are a **physical division** of the input data.

HDFS breaks down a large file say “1 GB” into blocks of say “64 MB” and stores 3 copies of each block on different nodes in the cluster. So, there will be 16 blocks (i.e 1024 MB / 64 MB). Since, HDFS does not know what is inside a block, when the blocks are broken, **some records may spill over blocks**.

If each map task processes all records in a specific data block, what happens to those **records that span block boundaries**? Hadoop uses a logical representation of the data stored in file blocks, known as **input splits** by figuring out where the first whole record in a block begins and where the last record in the block ends.

In situations where the last record in a block is incomplete, the input split includes location information for the next block and the byte offset of the data needed to complete the record. The **number of mappers** that run the task will be equal to the **number of input splits**.

Q15. Hadoop runs tasks in parallel across many nodes. What mechanism does Hadoop provide to combat the problem of one slow running task slowing down the entire job?
A15. Speculative Execution. A task may run slowly due to hardware or configuration issues at a particular node. Hadoop doesn’t try to diagnose and fix slow-running tasks. It tries to detect when a task is running slower than expected and

launches another equivalent task as a backup. This approach is known as “speculative execution of tasks”

Popular Posts

♦ 11 Spring boot interview questions & answers

828 views

♦ Q11-Q23: Top 50+ Core on Java OOP Interview Questions & Answers

768 views

18 Java scenarios based interview Questions and Answers

400 views

001A: ♦ 7+ Java integration styles & patterns interview questions & answers

389 views

01b: ♦ 13 Spring basics Q8 – Q13 interview questions & answers

296 views

♦ 7 Java debugging interview questions & answers

293 views

01: ♦ 15 Ice breaker questions asked 90% of the time in Java job interviews with hints

286 views

♦ 10 ERD (Entity-Relationship Diagrams) Interview Questions and Answers

280 views

♦ Q24-Q36: Top 50+ Core on Java classes, interfaces and generics interview questions & answers

240 views

001B: ♦ Java architecture & design concepts interview questions & answers

202 views

Bio | Latest Posts

Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and

often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.**945+** paid members. join my LinkedIn Group. **Reviews**

About Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.**945+** paid members. join my LinkedIn Group. **Reviews**

‹ ♥ 01: Q1 – Q6 Hadoop overview & architecture interview questions & answers

01: Convert XML file To Sequence File – writing & reading ›

Posted in Hadoop & BigData Interview Q&A, member-paid

Empowers you to open more doors, and fast-track

Technical Know Hows

☀ Java generics in no time ☀ Top 6 tips to transforming your thinking from OOP to FP ☀ How does a HashMap internally work? What is a hashing function?
☀ 10+ Java String class interview Q&As ☀ Java auto un/boxing benefits & caveats ☀ Top 11 slacknesses that can come back and bite you as an experienced Java developer or architect

Non-Technical Know Hows

☀ 6 Aspects that can motivate you to fast-track your career & go places ☀ Are you reinventing yourself as a Java developer? ☀ 8 tips to safeguard your Java career against offshoring ☀ My top 5 career mistakes

Prepare to succeed

☀ Turn readers of your Java CV go from “Blah blah” to “Wow”? ☀ How to prepare for Java job interviews? ☀ 16 Technical Key Areas ☀ How to choose from multiple Java job offers?

Select Category

© Disclaimer

 ↑