

Industrial strength Java/JEE Career Companion to open more doors

search here ...

Go

Home

Java FAQs

600+ Java Q&As

Career

Tutorials

Member

Why?

Can u Debug?

Java 8 ready?

Top X

Productivity Tools

Judging Experience?

Home › Interview › Hadoop & BigData Interview Q&A › 07: Q62 – Q70 HDFS
blocks Vs. splits & Spark partitions Interview Questions & Answers

07: Q62 – Q70 HDFS blocks Vs. splits & Spark partitions Interview Questions & Answers

Posted on June 15, 2016 by Arulkumaran Kumaraswamipillai



Q62. Can you explain the difference between HDFS **blocks** and input **splits**?

A62. A **block** is a physical representation of data, and a **Split** is a logical division of your data or records. For example, an input split might split a large text file at the end of a record using the “END OF LINE” character, but a block has no knowledge of the contents or records and splits the file by the physical size say 64MB, 128MB, 256MB, etc. This means you may have a half of a record spilling over to the next block. In

**600+ Full
Stack
Java/JEE
Interview
Q&As ♥Free
♦FAQs**

open all | close all

- ✚ Ice Breaker Interview
- ✚ Core Java Interview C
- ✚ JEE Interview Q&A (3
- ✚ Pressed for time? Jav
- ✚ SQL, XML, UML, JSC
- ✚ Hadoop & BigData Int
- ♥ 01: Q1 – Q6 Had
- 02: Q7 – Q15 Hadc
- 03: Q16 – Q25 Hac
- 04: Q27 – Q36 Apa
- 05: Q37 – Q50 Apa
- 05: Q37-Q41 – Dat
- 06: Q51 – Q61 HBa
- 07: Q62 – Q70 HDI
- ✚ Java Architecture Inte
- ✚ Scala Interview Q&As
- ✚ Spring, Hibernate, & I
- ✚ Testing & Profiling/Sa
- ✚ Other Interview Q&A 1
- ✚ 📺 Free Java Interview

other words, first half of a CSV line in end of “Block 1” and the remaining half of a CSV line in the beginning of “Block2”. A split can logically group 2 blocks of data.

Q63. Do the input splits contain the actual data?

A63. No. The “input splits” don’t contain the actual data. But store the locations to the actual data on HDFS. The “**map**” phase of a “map-reduce” job reads data from the HDFS blocks through the “splits”. The splits act as an intermediary between a Mapper and a block.

Q64. How do you verify the block size of a Hadoop file?

A64. The “hdfs” command with “fsck” and “-files -blocks”

```
1
2 $ hdfs fsck /user/john/basic-data/ -files -block
3
```

The output:

```
1
2 Connecting to namenode via http://mynamenode-ser
3 FSCK started by john (auth:SIMPLE) from /10.10.1
4 /user/john/basic-data/ <dir>
5 /user/john/basic-data/_SUCCESS 0 bytes, 0 block(
6
7 /user/john/basic-data/part-00000 215856020 bytes
8 0. BP-856152091-10.10.128.241-1416054677660:blk_
9
10 /user/john/basic-data/part-00001 215846999 bytes
11 0. BP-856152091-10.10.128.241-1416054677660:blk_
12
13 /user/john/basic-data/part-00002 215867307 bytes
14 0. BP-856152091-10.10.128.241-1416054677660:blk_
15
16 /user/john/basic-data/part-00003 215873401 bytes
17 0. BP-856152091-10.10.128.241-1416054677660:blk_
18
19 /user/john/basic-data/part-00004 215874370 bytes
20 0. BP-856152091-10.10.128.241-1416054677660:blk_
21
22 Status: HEALTHY
23 Total size: 1079318097 B
24 Total dirs: 1
25 Total files: 6
26 Total symlinks: 0
27 Total blocks (validated): 5 (avg. block si
28 Minimally replicated blocks: 5 (100.0 %)
29 Over-replicated blocks: 0 (0.0 %)
30 Under-replicated blocks: 0 (0.0 %)
31 Mis-replicated blocks: 0 (0.0 %)
32 Default replication factor: 3
```

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 \(1\)](#)
- [Low Latency \(7\)](#)
- [Memory Management \(1\)](#)
- [Performance \(13\)](#)
- [QoS \(8\)](#)
- [Scalability \(4\)](#)
- [SDLC \(6\)](#)
- [Security \(13\)](#)
- [Transaction Management \(1\)](#)

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 Tutorials \(1\)](#)
- [JEE Tutorials \(19\)](#)
- [Scala Tutorials \(1\)](#)
- [Spring & Hibernate Tutorials \(1\)](#)
- [Tools Tutorials \(19\)](#)
- [Other Tutorials \(45\)](#)

```

33 Average block replication:      3.0
34 Corrupt blocks:                0
35 Missing replicas:              0 (0.0 %)
36 Number of data-nodes:         11
37 Number of racks:              1
38 FSCK ended at Wed Jun 15 16:28:52 AEST 2016 in 2
39

```

As you can see “**repl**” is replication factor. 3 copies of data block stored on 3 different nodes in the cluster. The above cluster has 11 data nodes.

Q65. When HDFS has a file stored in 5 splits, can you repartition the data after you read it from HDFS in Apache Spark?

A65. Yes. By default spark creates one partition for each input split. You can **repartition** or split the data after you have read the data from HDFS a number of ways.

1. HDFS configuration

dfs.block.size – The default value in Hadoop 2.0 is 128MB.

2. Whilst reading the file in a Spark job

```

1
2 JavaSparkContext context = new JavaSparkContext(c
3 JavaRDD<String> textFileRdd = context.textFile("/")
4

```

3. Repartitioning a Spark RDD

Repartitioning will shuffle the data on different data nodes.

```

1
2 textFileRdd.repartition(12); // creates 12 input
3

```

or

coalesce, and coalesce will not shuffle data across network.

100+ Java pre-interview coding tests

[open all](#) | [close all](#)

- [Can you write code? \(1\)](#)
- [Complete the given](#)
- [Converting from A to I](#)
- [Designing your classe](#)
- [Java Data Structures](#)
- [Passing the unit tests](#)
- [What is wrong with th](#)
- [Writing Code Home A](#)
- [Written Test Core Jav](#)
- [Written Test JEE \(1\)](#)

How good are your?

[open all](#) | [close all](#)

- [Career Making Know-](#)
- [Job Hunting & Resum](#)

```
1  
2 textFileRdd.coalesce(12);  
3
```

The following code creates 4 partitions

```
1  
2 JavaSparkContext context = new JavaSparkContext(c  
3 context.parallelize(1 to 100, 4); //4 partitions  
4
```

You can print the number of partitions with

```
1  
2 System.out.println(textFileRdd.partitions().size(  
3
```

Not all files are splittable. The above examples used raw text files (e.g. a CSV payload). The `TextInputFormat` class that Spark utilizes inside the `SparkContext.textFile(...)` function is capable of splitting raw text files.

If the input files are compressed, not all compression algorithms are splittable. An example of anon-splittable compression scheme is Gzip. A gzipped file may spill over multiple blocks, which in turn could be stored on multiple nodes. So, gzipped files affect parallelism in Spark. Some of the splittable compression schemes are **Bzip2**, **Snappy** and **Lzo**

Q66. What are the different types of partitioning in Spark RDD?

A66. In Spark, you control the parallelism by controlling the “**input splits**”. Partitions don’t spill over multiple machines. This means tuples in the same partition are guaranteed to be on the same machine. The number of partitions can be configured.

There are 2 types of partitions in Spark RDD:

1) Hash partition, which attempts to spread data evenly across partitions based on the key.

```
1  
2 ordersRdd.map( e -> (e.customerId, e.price)).g  
3
```

partition = key.hashCode() % numberOfPartitions

2) Range partition, which partitions based on a sorted key order. E.g. Set ranges of [1,1000], [1001, 2000], [2001, 3000]

Q67. Do Spark RDD operations like map, join, groupBy maintain and propagate the partitions ?

A67. There are transformation operations in Spark that maintain & propagate the partitions.

foldByKey, partitionBy, join, groupByKey, reduceByKey, cogroup, groupWith, sort, etc

The following will maintain & propagate the partitions if parent has a partition:

mapValues, flatMapvalues, and filter

All other Spark RDD operations like “**map**” will produce a result without partitions.

Why doesn't the “map” operation maintain the partition?

A map operation can change the keys, hence it does not make sense to maintain the partitions as the keys have changed.

Q68. What is shuffling in Spark?

A68. Spark run jobs in stages. Spark stage are built by the DAGScheduler. Shuffling means the reallocation of data between multiple Spark stages. Typically shuffling is triggered by transformation operations like **foldByKey, partitionBy, join**, etc. These operations create ShuffleRDD, CoGroupRDD, etc under the covers. In MapReduce, shuffling takes place in between “map” and “reduce” tasks.

Shuffling Example

Say, you want to calculate the **number of occurrences of each** word in a text document.

You will be using a “word” as the **key** and “1” as the **value**. This is known as the “map” task. After this map task, you would **sum up values** for each key. This is known as the “reduce” task. But when you store the data across the cluster, how can you sum up the values for the same key stored on different machines? The only way to achieve this is to make all the values for the same key be on the same machine, after this you would be able to sum them up. So, before the “reduce” task, shuffling takes place. There are many operations like join, groupBy, etc trigger a shuffling process.

Q69. What does a shuffling process involve?

A69. It can involve expensive data sorting & partitioning, data serialization & deserialization to move data across the network, data compression to reduce I/O bandwidth, and disk I/O for merging.

Q70. What the different types of shuffling implementations?

A70. **1)** Hash Shuffle **2)** Sort Shuffle.

— Hash shuffle outputs one separate file for each of the “reducers”. More suited when smaller number of reducers are required.

— sort shuffle outputs a single file ordered by “reducer” id, and ids are indexed. This allows you to fetch any chunk of the data related to “reducer x”, and perform a single fseek before fread.

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

401 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](https://www.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](#)

◀ 02: Scenarios based Java OO concepts & GoF design patterns – naive Template Method pattern

03: Scenarios based Java OO concepts & GoF design patterns – Strategy Pattern ▶

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?](#)

© Disclaimer

The contents in this Java-Success are copy righted. The author has the right to correct or enhance the current content without any prior notice.

These are general advice only, and one needs to take his/her own circumstances into consideration. The author will not be held liable for any damages caused or alleged to be caused either directly or indirectly by these materials and resources. Any trademarked names or labels used in this blog remain the property of their respective trademark owners. No guarantees are made regarding the accuracy or usefulness of content, though I do make an effort to be accurate. Links to external sites do not imply endorsement of the linked-to sites.