TEAM NUMBER: Yue Ma, Yuheng Li and Conger Yang.

BASIC TASK

1)

a) Main advantage: ease the load of the central server, transfer the load of the central server to other 20 different hosts (sites); The security of data storage is improved, and the single target is divided into multiple targets. Reduce server faults caused by heavy load.

b)Potential drawback: in the process of data transmission, there may be problems in data consistency, and in the process of data transmission to other 20 hosts (sites), there may be disconnection, transmission failure and other problems; On top of that, having 20 different sites would lead to increased costs, including, of course, the cost of buying machines and maintaining them; Finally, about the maintenance of the site, 20 hosts (sites) distributed in different areas will lead to site maintenance becomes inconvenient, resulting in a reduction in efficiency.

2)

a) Main drawback: there is a certain risk in the integrity of data transmission, a and b are two different hosts (sites) and are far apart; In terms of data security, in the process of data transmission, data may be intercepted, malicious tampering and other malicious behaviors; On the cost side, transferring large files in the process of long-distance transportation of data consumes a lot of traffic, which is very unfriendly to network services that are charged by traffic.

b) b files are segmented, large files are divided into small files (mongoDB can achieve this function), and these small files are distributed to multiple different computing nodes for parallel processing. Each computing node is responsible for processing one or more file fragments and filtering data according to filtering conditions. On node a, filtering results from different compute nodes can be received asynchronously and stored using appropriate data structures (such as thread-safe queues, databases, and so on). Once all compute nodes have completed processing, the results can be summarized on node a for subsequent analysis or recording

c) Data locality means that in distributed computing tasks, computing operations and data are stored physically close to each other to reduce the overhead of data transmission and improve performance.

3.

i) **Sequential processing**: Break down matrices A and B into smaller sub-matrices, each of which can be assigned to different processors or computing nodes.

ii) **Parallel/distributed processing:** Each processor or computing node can independently compute the product of its assigned sub-matrix.

iii) **1.Determine Sub-matrix Size:** Based on the number of processors or computing nodes in the system, determine the size of each sub-matrix.

2.Task Assignment: Assign the corresponding parts of matrices A and B to different processors or computing nodes.

3.Parallel Computation:

Each processor or computing node computes the product part of its assigned sub-matrix.

For example, if matrix A is divided into four sub-matrices A1, A2, A3, A4, and matrix B is divided into B1, B2, then you can compute A1B1, A2B2, A3B3, A4B4 in parallel.

4. **Merging Results:** Merge the product results of all sub-matrices into the final matrix C.

4.

a) Independently compute (a+b)(a+b) and (c/d)(c/d), as well as (e×f)(e×f) and (g/h)(g/h)

b)

1. **Data Dependency:** Coordination is needed for the final subtraction operation.
2. **Communication Overhead:** Data transfer between nodes may introduce delays.
3. **Synchronization Overhead:** Parallel tasks need to be synchronized to complete the computation.

c)

1. **Network Latency:** Affects the efficiency of data transfer.
2. **Resource Allocation:** Need to balance the load on each node.
3. **Fault Tolerance:** Must handle node failures.

5）

Firstly, divide the large file into 50 smaller parts. Each computer node is assigned a file section and these 50 computer nodes are processed in parallel. Each node independently searches for the local maximum number in its assigned file section. Nodes can use methods such as sequential traversal to compare each number in the file section and record the maximum number currently discovered. Next, each node summarizes the local maximum number it has found and provides it to the main node. Then the master node receives the local maximum number from each child node and finds the global maximum number from it. If the method of comparing nodes with each other is adopted, the global maximum number can be gradually determined through multiple rounds of comparison. In this way, by utilizing the computing power of 50 computer nodes through distributed and parallel processing, the maximum number in very large files can be efficiently determined.

6）

The part a of the fourth question is to first search for methods that can use parallelism, and then decide on distributed methods. However, the fifth question is to allocate tasks through distributed methods first, and then quickly execute them through parallelism. And the reason why they adopt different methods, in my opinion, is the complexity of the data. The fifth question is about text data, which only contains numbers and is relatively simple, so a distributed method can be quickly adopted. The fourth question is about the structure of trees, which makes it difficult to use distributed partitioning from the beginning. Therefore, we need to first find the parts that can be used in parallel, and then use a column based method based on this part.

7）

a)

1. Relationship between management and managed:

NameNode is the manager of HDFS, responsible for managing the namespace of the file system, maintaining the directory tree structure of the file system, and mapping the relationship between files and data blocks.

DataNode is a node that stores actual data and is managed and scheduled by NameNode.

2. Communication relationship:

DataNode will regularly send heartbeat information to NameNode to inform it of its status (such as whether it is running normally, storage capacity, etc.).

The NameNode makes decisions based on the status information of the DataNode, such as allocating data blocks and placing replicas.

3. Storage division:

NameNode mainly stores metadata of the file system, rather than the actual data content. Metadata includes file names, file directory structures, location information of data blocks, etc.

DataNode is responsible for storing data blocks and performing read and write operations on data according to the instructions of NameNode.

b)

1) Due to the file size of 180MB and block size of 64MB, the number of partitions is 3. They are S1 with a size of 64MB, S2 with a size of 64MB, and S3 with a size of 52MB.

2) There are generally three replicas required in HDFS for three reasons

1. Improve data reliability: If one replica is damaged or lost, there are other replicas available to ensure that data is not easily lost.

2. Increase data availability: Even if a storage node fails, data can still be read from other normal replicas to ensure the continuous availability of the system.

3. Optimize performance: Multiple replicas can allow data to be read in parallel on different nodes, improving data access speed.

The process of implementing replication

1. The client initiates a file write request to the NameNode.

2. The NameNode determines the data block size of the file based on the file size and cluster configuration, and assigns a storage location for each data block. Usually, the first replica is placed on a node in the same rack as the client's node, the second replica is placed on a node in a different rack, and the third replica is placed on a different node in the same rack as the second replica. This layout can optimize the performance of data reading while ensuring data reliability.

3. The client writes data to the node where the first replica is located, which receives the data while passing it on to the node where the second replica is located. The second replica node then passes the data on to the node where the third replica is located.

After all replicas have been successfully written, the client confirms to the NameNode that the file write is complete.

3) Choose to split S1. Store one replica in node 1 of rack 1, one replica in node 5 of rack 2, and one replica in node 9 of rack 3. This configuration ensures that data is stored dispersedly on different racks, improving the reliability and availability of data. Even if one rack fails, data can still be recovered and accessed from replicas on other racks. At the same time, data transmission between different racks can also utilize the parallelism of the network to improve data read and write performance.

4)

1. Fault detection: NameNode will detect the fault of node 5 through heartbeat mechanism and data block reporting mechanism. The DataNode will periodically send heartbeat signals to the NameNode. If the NameNode does not receive a heartbeat from a DataNode for a period of time, it will be considered that the node has malfunctioned. At the same time, DataNode will regularly report the data block information it stores to NameNode. When node 5 fails, its data block report will not include the replica information of split S1.
2. Start replica replication: The NameNode will select a new node to create a replica of split S1 to restore data redundancy. Usually, priority is given to selecting nodes from different racks that store existing replicas to ensure data distribution across different racks and improve fault tolerance. For example, you can choose a node that is different from the rack where nodes 3 and 7 are located to create a new replica.
3. Coordinate the replication process: The NameNode will instruct the nodes that store the split S1 replica (such as node 3 and node 7) to replicate the data to the newly selected node. These nodes will start replicating data blocks and report to the NameNode after replication is complete.
4. Updating metadata: After the new replica is created, the NameNode will update the metadata of the file system and record the location information of the new replica.

8.)

a) In the traditional Hadoop MapReduce model, the JobTracker is responsible for managing and coordinating the entire job process, while the TaskTracker is responsible for executing the actual tasks. The JobTracker monitors the progress of task execution, allocates resources, and re-executes failed tasks. Each TaskTracker on the cluster nodes executes the tasks assigned by the JobTracker and reports the status and progress of the tasks.

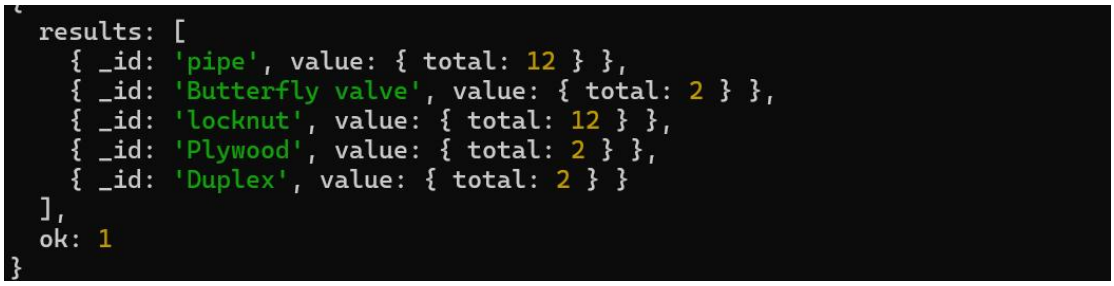
b) The MapReduce job is divided into two main stages: the Map phase and the Reduce phase.

Map phase: In this phase, the input data is processed and converted into key-value pairs. The map function receives the input record, processes it, and outputs a set of intermediate key-value pairs.

Reduce phase: This phase processes the intermediate key-value pairs output by the Map phase and merges the values with the same key to generate the final output results.

c) When dealing with a data block containing squares of various colors, we can use the MapReduce model to calculate the number of squares of each color. First, in the Map phase, we iterate through each square, using its color as the key and setting the value to 1, so that each color will correspond to a count. Then, during the Shuffle phase, the Hadoop framework automatically gathers all key-value pairs of the same color together. In the Reduce phase, we sum up the values of these color-matched key-value pairs to get the total number of squares for each color. Finally, we obtain a result set that includes each color and its corresponding number of squares.

9.



```
{
  results: [
    { _id: 'pipe', value: { total: 12 } },
    { _id: 'Butterfly valve', value: { total: 2 } },
    { _id: 'locknut', value: { total: 12 } },
    { _id: 'Plywood', value: { total: 2 } },
    { _id: 'Duplex', value: { total: 2 } }
  ],
  ok: 1
}
```

Inputkey:item.equip_key(assumption) outputkey:equip_name;

10)

Map phase

1. Read each line of data from the file and parse it into an account number and program name.
2. For each row of data, output a key value pair, where the key is the program name and the value is 1, indicating a playback.

For example, for the first line of input "121212123 Aladdin", output a key value pair ("Aladdin", 1).

Reduce phase

1. Receive key value pairs from the Map stage and group them by key (program name).
2. Sum up the values in each group to obtain the number of plays for each program.
3. Calculate the fees to be paid based on the number of views and price.

For example, if there are 5 key value pairs with the key "Aladdin", then the number of plays for "Aladdin" is 5. Assuming the price of "Aladdin" is $0.05 per play, the fee to be paid is $0.25, which is 5 * 0.05.