1. **Building the environment**

|  |  |
| --- | --- |
| Hardware information | SG2042/Kunpeng920 |
| architecture | riscv64/ aarch64 |
| operating system | openEuler 24.03 (LTS) |
| JDK edition | 17 |
| Maven edition | 3.3 or later |
| Boost edition | 1.72 |
| Proto buf version | 2.5.0, 3.7.1 , 3.21.7, 3.21.12 |
| GCC | 12.3.1 |
| G++ | 12.3.1 |
| Python edition | 3.11.6 |
| Kernel version | 6.6.0 |
| Perf edition | 6 .6 .0 |
| grpc-java | 1.53.0 |
| leveldbjni | 1.8 |
| leveldb-java | 0.12 |
| hawtjni-runtime | 1.16 |
| yarn | 1.22.5 |
| node | 12.22.1 |

1. **Build the process**

**1. Compile HiBench**

(1) Prepare the environment and compile with JDK8:

|  |
| --- |
| # Identify the server architecture  ARCH=$( uname -m)  # Set the jdk8 version used for compilation  if [ "$ARCH" == "riscv64" ]; then  export JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.402.b06-2.oe2403.riscv64  export PATH=$PATH:$JAVA\_HOME/bin  elif [ "$ARCH" == "aarch64" ]; then  export JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.422.b05-1.oe2403.aarch64  export PATH=$PATH:$JAVA\_HOME/bin  fi |

(2) Pull HiBench from GitHub:

|  |
| --- |
| cd $HOME  git clone git@github.com:Intel-bigdata/ HiBench.git |

(3) Start compiling:

|  |
| --- |
| cd HiBench  mvn - Dspark=2.4 - Dscala=2.11 clean package -e |

(4) Configuration Settings file (modify hibench.hadoop.home and hibench.hdfs.master according to actual situation):

|  |
| --- |
| cat <<EOF > conf/ hadoop.conf  # Hadoop home  hibench.hadoop.home / usr/local/hadoop-3.4.0  # The path of hadoop executable  hibench.hadoop.executable \${ hibench.hadoop.home}/bin/ hadoop  # Hadoop configraution directory  hibench.hadoop.configure.dir \${ hibench.hadoop.home}/ etc/ hadoop  # The root HDFS path to store HiBench data  hibench.hdfs.master hdfs://localhost:9000/user/$USER  # Hadoop release provider. Supported value: apache  hibench.hadoop.release apache  EOF  sed - i "s/ hibench.masters.hostnames/ hibench.masters.hostnames $( uname -n)/g" conf/ hibench.conf  sed - i "s/ hibench.slaves.hostnames/ hibench.slaves.hostnames $( uname -n)/g" conf/ hibench.conf |

**2. Get ready to test**

(1) Prepare data:

|  |
| --- |
| # Pull the perf script and test data from the repository  git clone git@github.com:zhangxiangbo123/ hadoop.git  cd hadoop/test  Create a local folder  mkdir output  Create the hdfs folder  hdfs dfs - mkdir -p test/input  hdfs dfs - mkdir -p test/output  # Upload the local test data to hdfs  hdfs dfs -put input/\* test/input  # Initialize yarn (for NN\* test task), all press Enter  yarn init  mkdir NN |

(2) Define functions:

|  |
| --- |
| # defined function  execute\_job() {  local job\_cmd="$1"  job=$(echo "$1" | awk {print $4})  log\_name=$(echo "$job" | sed s/ //g)  echo "Starting job: $job\_cmd"  *# Execute the job and save the log file*  bash performance\_counter\_920.sh "$job\_cmd" ./ 2>&1 | tee -a "temp.log"  awk /completed successfully/{success=1} /File System Counters/{flag=1; successFlag=success} /^$/{flag=0} flag && successFlag temp.log > temp2.log && mv temp2.log $log\_name.log  rm temp.log  } |

(3) Define the file path:

|  |
| --- |
| # Define the path of the file  hadoop\_examples\_path="/usr/local/hadoop-3.4.0/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.4.0.jar"  hadoop\_tests\_path="/usr/local/hadoop-3.4.0/share/hadoop/mapreduce/hadoop-mapreduce-client-jobclient-3.4.0-tests.jar"  hadoop\_hibench\_path="$HOME/ HiBench" |

(8) Define the task array:

|  |
| --- |
| # Define the task array  hadoop\_examples\_jobs=(  " randomwriter test/output/ randomwriter"  "sort test/output/ randomwriter test/output/sorted-data"  "grep test/input/text-data.txt test/output/grep-output th.\*"  "join test/output/sorted-data test/output/join"  " aggregatewordcount test/output/sorted-data test/output/ aggregatewordcount"  " aggregatewordhist test/output/sorted-data test/output/ aggregatewordhist"  " bbp 10 20 5 test/output/ bbp-output"  " distbbp 2 20 2 m 2 test/output/remote\_distbbp ./local\_distbbp"  " distbbp 2 20 2 m 2 test/output/remote\_distbbp ./local\_distbbp"  " multifilewc test/input/text-data\*.txt test/output/ multifilewc-output"  "pi 10 10"  " randomtextwriter test/output/ randomtextwriter"  " secondarysort test/input/secondarysort-data.txt test/output/ secondarysort-output"  "sudoku test/input/ puzzle.dta"  " teragen 10000000 test/output/ terasort-input"  " terasort test/output/ terasort-input test/output/ terasort-output"  " teravalidate test/output/ terasort-output test/output/ terasort-validate"  "wordcount test/input/text-data.txt test/output/wordcount-output"  " wordmean test/input/text-data.txt test/output/ wordmean-output"  " wordmedian test/input/text-data.txt test/output/ wordmedian-output"  " wordstandarddeviation test/input/text-data.txt test/output/ wordstandarddeviation-output"  )  hadoop\_tests\_jobs=(  " TestDFSIO -write - nrFiles 10 - fileSize 1000"  " TestDFSIO -read - nrFiles 10 -size 1000"  " TestDFSIO -clean"  "DistributedFSCheck"  " JHLogAnalyzer - historyDir test/input/log-data - resFile test/output/ JHLogAnalyzer-output"  " SliveTest"  " TestDFSIO -write - nrFiles 10 - fileSize 1000"  " TestDFSIO -read - nrFiles 10 -size 1000"  " TestDFSIO -clean"  "fail - failReducers 4"  "fail - failMappers 3"  " testfilesystem -files 10 - megaBytes 100 - noseek - noread"  " gsleep -m 5 -r 10 -mt 10000 -rt 10000 - recordt 10000"  " largesorter test/output/ randomwriter test/output/ largesorter"  " loadgen -m 200 -r 150 - outKey org.apache.hadoop.io.Text - outValue org.apache.hadoop.io.Text"  " mapredtest 5 5"  " mrbench - numRuns 50"  " nnbench -operation create\_write -maps 10 -reduces 5 - numberOfFiles 1000"  " nnbenchWithoutMR -operation createWrite - baseDir test/output/ nnbenchWithoutMR - startTime 1 - numFiles 1000 - replicationFactorPerFile 1 - blocksPerFile 1 - bytesPerBlock 1048576"  "sleep -m 5 -r 10 -mt 10000 -rt 10000 - recordt 10000"  " testbigmapoutput -input test/output/sorted-data -output test/output/ testbigmapoutput"  " testfilesystem - fastcheck"  " testmapredsort - sortInput test/output/ randomwriter - sortOutput test/output/ testmapredsort"  " testsequencefile test/output/ randomwriter/part-m-00000"  " testsequencefileinputformat test/input/ sequence.seq"  " testtextinputformat test/input/text-data.txt"  " threadedmapbench"  " timelineperformance"  )  yarn\_tests\_jobs=(  "$HADOOP\_HOME/bin/yarn jar /usr/local/hadoop-3.4.0/share/hadoop/mapreduce/hadoop-mapreduce-client-jobclient-3.4.0-tests.jar NNstructureGenerator - outDir NN"  "$HADOOP\_HOME/bin/yarn jar /usr/local/hadoop-3.4.0/share/hadoop/mapreduce/hadoop-mapreduce-client-jobclient-3.4.0-tests.jar NNdataGenerator - inDir NN -root test/output/ testLoadSpace"  "$HADOOP\_HOME/bin/yarn jar /usr/local/hadoop-3.4.0/share/hadoop/mapreduce/hadoop-mapreduce-client-jobclient-3.4.0-tests.jar NNloadGenerator -root test/output/ testLoadSpace - numOfThreads 2000"  "$HADOOP\_HOME/bin/yarn jar /usr/local/hadoop-3.4.0/share/hadoop/mapreduce/hadoop-mapreduce-client-jobclient-3.4.0-tests.jar NNloadGeneratorMR - mr 5 test/output/ NNloadGeneratorMR - elapsedTime 20"  )  hadoop\_hibench\_prepare\_jobs=(  "$hadoop\_hibench\_path/bin/workloads/micro/wordcount/prepare/prepare.sh"  "$hadoop\_hibench\_path/bin/workloads/micro/sleep/prepare/prepare.sh"  "$hadoop\_hibench\_path/bin/workloads/micro/dfsioe/prepare/prepare.sh"  "$hadoop\_hibench\_path/bin/workloads/micro/sort/prepare/prepare.sh"  "$hadoop\_hibench\_path/bin/workloads/micro/terasort/prepare/prepare.sh"  "$hadoop\_hibench\_path/bin/workloads/ml/bayes/prepare/prepare.sh"  "$hadoop\_hibench\_path/bin/workloads/ml/kmeans/prepare/prepare.sh"  "$hadoop\_hibench\_path/bin/workloads/sql/aggregation/prepare/prepare.sh"  "$hadoop\_hibench\_path/bin/workloads/ sql/join/prepare/prepare.sh"  "$hadoop\_hibench\_path/bin/workloads/ sql/scan/prepare/prepare.sh"  "$hadoop\_hibench\_path/bin/workloads/websearch/nutchindexing/prepare/prepare.sh"  "$hadoop\_hibench\_path/bin/workloads/websearch/pagerank/prepare/prepare.sh"  )  hadoop\_hibench\_jos=(  "$hadoop\_hibench\_path/bin/workloads/micro/wordcount/hadoop/run.sh"  "$hadoop\_hibench\_path/bin/workloads/micro/sleep/ hadoop/run.sh"  "$hadoop\_hibench\_path/bin/workloads/micro/ dfsioe/ hadoop/run.sh"  "$hadoop\_hibench\_path/bin/workloads/micro/sort/ hadoop/run.sh"  "$hadoop\_hibench\_path/bin/workloads/micro/terasort/hadoop/run.sh"  "$hadoop\_hibench\_path/bin/workloads/ml/bayes/ hadoop/run.sh"  "$hadoop\_hibench\_path/bin/workloads/ml/ kmeans/ hadoop/run.sh"  "$hadoop\_hibench\_path/bin/workloads/sql/aggregation/hadoop/run.sh"  "$hadoop\_hibench\_path/bin/workloads/ sql/join/ hadoop/run.sh"  "$hadoop\_hibench\_path/bin/workloads/ sql/scan/ hadoop/run.sh"  "$hadoop\_hibench\_path/bin/workloads/websearch/nutchindexing/hadoop/run.sh"  "$hadoop\_hibench\_path/bin/workloads/websearch/pagerank/hadoop/run.sh"  ) |

(9) Execute test tasks:

|  |
| --- |
| # Execute test tasks  for job in "${hadoop\_examples\_jobs[@]}"; do    execute\_job " hadoop jar $hadoop\_examples\_path $job"  done  for job in "${hadoop\_tests\_jobs[@]}"; do    execute\_job " hadoop jar $hadoop\_tests\_path $job"  done  for job in "${yarn\_tests\_jobs[@]}"; do    execute\_job "$HADOOP\_HOME/bin/yarn jar $hadoop\_tests\_path $job"  done  for job in "${hadoop\_hibench\_prepare\_jobs[@]}"; do  "$job"  done  for job in "${hadoop\_hibench\_jobs[@]}"; do  execute\_job "$job"  done |

Summary of issues

**Question 1: dbdount**

**description ：**

Calculate page views from the database

**Question two: Pentomino**

**description ：**

An example of the Pentomino problem needs to be generated first.

**Question 3: DFSCIOTest write/read**

|  |
| --- |
| Error: java.io.IOException: Cannot run program "/ tmp/ DFSCIOTest/hdfs\_read" (in directory "/ tmp/ DFSCIOTest"): error=13, Permission denied |

**description ：**

This test case requires some preparation

Create /usr/local/hadoop/libhdfs/hdfs\_read/hdfs\_write files

ln -s / usr/local/ hadoop/lib/native/libhdfs.so.0.0.0 / usr/local/ hadoop/ libhdfs/libhdfs.so.1

It needs to be executed with sudo, and the permissions of /tmp and /benchmarks in hdfs dfs should be changed to 777.

But in the end it was a matter of permissions.

**Question 4: MRReliabilityTest**

|  |
| --- |
| INFO mapred.ReliabilityTest: This must be run in only the distributed mode ( LocalJobRunner not supported). |

**description ：**

It needs to be tested in a distributed environment.

**Question 5: minicluster**

|  |
| --- |
|  |
|  |

**description ：**

1. Error java.lang.NoClassDefFoundError: org/ junit/Assert

Solution: Add juint.jar to the environment

e xportHADOOP\_CLASSPATH=$HADOOP\_CLASSPATH:$HADOOP\_HOME/share/hadoop/tools/lib/junit-4.13.2.jar

1. The storage directory /home/ zhangxiangbo/test/target/test/data/dfs/name-0-1 does not exist

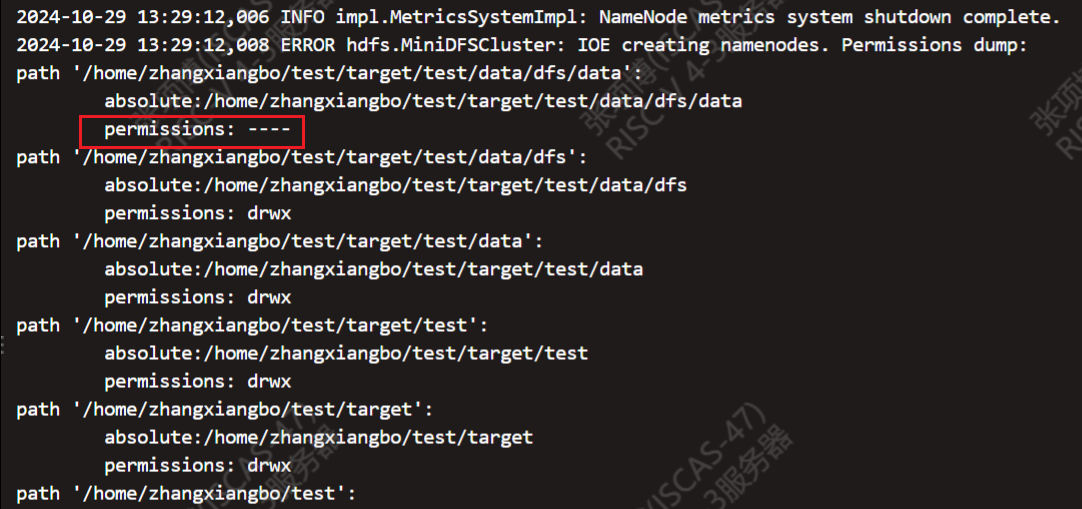
Storage directory /home/ zhangxiangbo/test/target/test/data/ dfs/name-0- 2 does not exist

Solution: Create these two folders

mkdir -p /home/ zhangxiangbo/test/target/test/data/ dfs/name-0-1

mkdir -p /home/ zhangxiangbo/test/target/test/data/ dfs/name-0- 2

1. report an error



Solution: Create a data folder and set permissions

mkdir -p /home/ zhangxiangbo/test/target/test/data/ dfs/ data

chmod -R 755 /home/ zhangxiangbo/test/target/test/data/ dfs

1. report an error

java.io.IOException: NameNode is not formatted.

Solution: The problem is still not solved after the namenode is reformatted.

**Question 6: testfilesystem**

|  |
| --- |
|  |

**description ：**

Solution: Add juint.jar to the environment, but still cannot solve the problem, the process will be stuck.

e xportHADOOP\_CLASSPATH=$HADOOP\_CLASSPATH:$HADOOP\_HOME/share/hadoop/tools/lib/junit-4.13.2.jar

**Question 7: testsequencefileinputformat**

|  |
| --- |
| java.io.EOFException  at java.io.DataInputStream.readFully(DataInputStream.java:198)  at org.apache.hadoop.io.BytesWritable.readFields(BytesWritable.java:188)  at org.apache.hadoop.io.serializer.WritableSerialization$WritableDeserializer.deserialize(WritableSerialization.java:71) |

**description ：**

This use case is used to test the sequence file input format. However, an error occurs.

**Question 8: sql.aggregation**

|  |
| --- |
|  |

**description ：**

H ibench must first configure the configuration file

Then create the folder

h dfs dfs - mkdir -p hdfs://localhost:9000/user/zhangxiangbo/HiBench/Aggregation/Output

Then it will report the above error, and never figure out what $3 is.