

郑州大学电气工程学院公共安全大 数据与人工智能平台设备采购项目

Insight HD 集群性能测试报告

2021 年 5 月

1 概述

1.1 文档目的

本文档主要描述一体机 Spec 测试活动的测试内容。

本次测试在特定软硬件配置下，通过测试评估 Insight 产品的极限性能。结合 Insight 已有的性能测试数据，进行对比分析，形成一体机 Spec 规格，为前端推广和宣传作参考。

1.2 文档范围

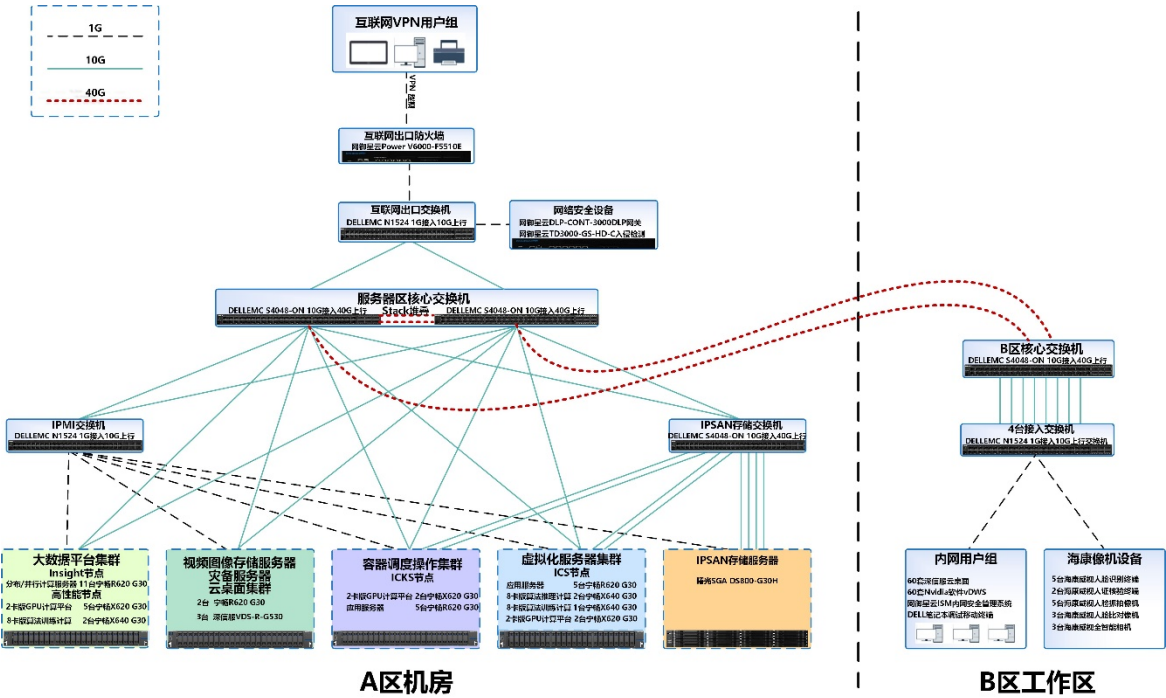
文档主要涉及一体机 Spec 相关软硬配置、测试策略、测试结果及分析。

1.3 术语、定义和缩略语

序号	术语/缩略语	解释
1.	存储性能	用来评价大数据平台 Insight 在分布式文件系统存储、NoSQL 列式存储、内存数据库等应用中并发 IO 能力。
2.	数据分析性能	用来评价大数据平台 Insight 对不同应用场景 SQL 的处理能力，如 CPU 密集型、IO 密集型
3.	并行计算能力	用来评价大数据平台 Insight 中内存计算 Spark 和离线计算 MapReduce 平均每节点处理能力
4.	流处理性能	用来评价大数据平台 Insight 在实时流处理应用方面的能力
5.	检索性能	用来评价大数据平台 Insight 在搜索引擎应用方面的能力

2 环境及工具

2.1 环境拓扑



2.2 硬件配置

产品名称	配置信息	招标名称	备注	数量
X640 G30_4U	XEON 5218R*2 /DDR4 32G*8 600G 2.5 吋 10K 12Gb SAS 硬盘*3 /1.2TB 2.5 吋 10K 12Gb SAS 硬盘*5 1600W 电源模块*4 /双口 10G 光纤网卡*2 NV TESLA T4 -E3x16 16GB 75W GPU 卡*8	高性能 GPU 计算平台 (8 卡版) (算法推理计算)	8 卡 T4-4 光口	2
X640 G30_4U	XEON 6240R*2 /DDR4 32G*12 600G 2.5 吋 10K 12Gb SAS 硬盘*3 /2.4TB 2.5 吋 10K 12Gb SAS 硬盘*5 1600W 电源模块*4 /双口 10G 光纤网卡 NV TESLA V100S -E3x16 32GB 250W 双宽 GPU 卡*8	高性能 GPU 计算平台 (8 卡版) (算法训练计算)	8 卡 V100S-2 光口	2
X640 G30_4U	XEON 6240R*2 /DDR4 32G*12 600G 2.5 吋 10K 12Gb SAS 硬盘*3 /2.4TB 2.5 吋 10K 12Gb SAS 硬盘*5 1600W 电源模块*4 /双口 10G 网卡*2 NV TESLA V100S -E3x16 32GB 250W 双宽 GPU 卡*8	高性能 GPU 计算平台 (8 卡版) (算法训练计算)	8 卡 V100S-4 光口	1

X620 G30_2U	XEON 5218R*2 /DDR4 2933 32G*8 2.4TB 2.5 吋 10K 12Gb SAS 硬盘*10 /480G 2.5 SATA 6G R SSD*2 1200W 电源模块*2 /双口 10G 光纤网卡 NV TESLA T4 -E3x16 16GB 75W 单宽 GPU 卡*2	高性能 GPU 计 算平台（2 卡 版）	2 卡 T4-2 光 口	4
X620 G30_2U	XEON 5218R*2*2 /DDR4 32G*8 / 2.4TB 2.5 吋 10K 12Gb SAS 硬盘*10 /480G 2.5 SATA 6G R SSD*2 / 1200W 电源模块*2 /双口 10G 光纤网卡*2 NV TESLA T4 -E3x16 16GB 75W 单宽 GPU 卡*2	高性能 GPU 计 算平台（2 卡 版）	2 卡 T4-4 光 口	4
X620 G30_2U	XEON 5218R*2*2 /DDR4 32G*8 / 2.4TB 2.5 吋 10K 12Gb SAS 硬盘*12 /480G 2.5 SATA 6G R SSD*2 / 1200W 电源模块*2 /双口 10G NV TESLA T4 -E3x16 16GB 75W 单宽 GPU 卡*2	高性能 GPU 计 算平台（2 卡 版）	2 卡 T4-2 光 口 (2.4T*12)	1
R620 G30 2U	XEON 5218R*2*2 /DDR4 32G*4 1.2TB 2.5 吋 10K 12Gb SAS 硬盘*12 /240G 2.5 SATA 6Gb R SSD*2 / 550W 电源模块*2 /双口 10G*2	应用服务器	4 光口	10
R620 G30 2U	XEON 5218R*2*2 /DDR4 32G*16 6TB 3.5 吋 7.2K 6Gb SATA 硬盘*6 /960G 2.5 6Gb R SSD*6 /240G 2.5 SATA 6Gb R SSD*2 550W 电源模块*2 /双口 10G SFP+光纤 OCP 网卡	分布式计算服 务器（用于 Hadoop 发行版 分布式非关系 型计算）并行 计算服务器 （用于数据仓 库等并行计 算）	2 光口	11

2.3 软件配置

序号	产品名称	规格型号	用途备注
1	InsightV4.6	(1) 安装镜像: InCloudInsight-V4R02A001-b35.centos7.x86_64.tar 【20200331 发行版】 (2) 部署模式采用 3+X 方式。其中 1 个 manager, 2 个 master, 8 个 DataNode	大数据集群
2	操作系统	CentOS Linux release 7.4.1708 (Core)	

3	浏览器	Google Chrome 版本 63.0.3239.108	
---	-----	--------------------------------	--

2.4 测试工具

序号	工具名称	版本	用途说明
1	TestDFSIO	3.1.1	Hadoop 自带基准测试工具，用于测试 HDFS I/O 性能；
3	YCSB	V0.17.0	雅虎开发的用来对云服务进行基础测试的工具,用于测试 Hbase 读写查询性能；
4	Hibench	V7.0	Intel 开源的大数据基准测试工具，用于测试 Spark 计算框架的处理速度。

2.5 关键指标统计

分类	测试项	测试目标值
HDFS	写 20000 个 100M 文件耗时	<1447.527s
	读 20000 个 100M 文件耗时	<1266.533s
Mapreduce	Wordcount: 平均每节点处理能力 (1T)	>8.23 G/min/node
	TeraSort: 平均每节点处理能 (300G)	>7.78 G/min/node
Spark	Wordcount: 平均每节点处理能力 (1T)	>50.4 G/min/node
	TeraSort: 平均每节点处理能 (300G)	>6.9 G/min/node
HBase	Load: 平均每节点写入记录条数	>66, 283 Records/s/Node

	100%随机读：平均每节点写入记录条数	> 150,000 Records/s/Node
	读写混合：平均每节点写入记录条数	> 60,000 Records/s/Node
	Readmostly：平均每节点写入记录条数	> 79,986 Records/s/Node
	100%随机写：平均每节点写入记录条数	> 64,000 Records/s/Node
	顺序扫描场景：平均每节点写入记录条数	> 46,000 Records/s/Node
	50%随机写+50%扫描读写场景：平均每节点写入记录条数	> 61,000 Records/s/Node

3 性能测试

3.1 测试概览

测试采用黑盒测试方法，从 Insight 集群资源、任务过程参数、底层系统参数 3 个维度进行调优，以测试目标值为参考，尽可能挖掘本次一体机配置在测试场景下的最优性能。

“√”：测试通过，优于目标值

“×”：测试失败，未达到目标值

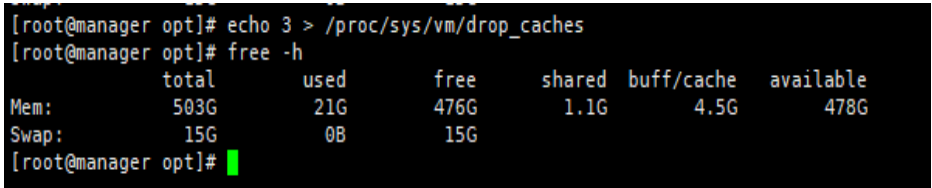
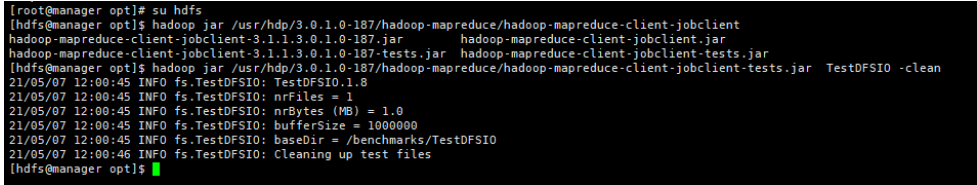
分类	测试项	本次测试值	测试结论
HDFS	写 2000 个 100M 文件耗时	126.64s	√
	读 2000 个 100M 文件耗时	125.12s	√
Mapreduce	Wordcount: 平均每节点处理能力 (1T)	12.88 G/min/node	√
	TeraSort: 平均每节点处理能 (300G)	15.65 G/min/node	√
Spark	Wordcount: 平均每节点处理能力 (1T)	96.39 G/min/node	√
	TeraSort: 平均每节点处理能 (300G)	16.5 G/min/node	√
HBase	Load: 平均每节点写入记录条数	105764.15 Records/s/Node	√
	100%随机读: 平均每节点写入记录条数	146669.57 Records/s/Node	×
	读写混合: 平均每节点写入记录条数	100728.87 Records/s/Node	√
	Readmostly: 平均每节点写入记录条数	60643.5 Records/s/Node	√
	100%随机写: 平均每节点写入记录条数	255357.39 Records/s/Node	√
	顺序扫描场景: 平均每节点写入记录条数	417379.92 Records/s/Node	√

	50%随机写+50%扫描读写场景：平均每节点写入记录条数	298067.46 Records/s/Node	√
--	------------------------------	--------------------------	---

3.2 存储性能测试

3.2.1 分布式文件系统并发读写（HDFS）

3.2.1.1 测试用例

性能标识	DFP-001	编写人员	申晓青
测试目标	测试当前环境并发读写文件的耗时是否满足测试目标值，挖掘最优的性能		
前提条件	1. 环境已安装 HDFS、Hadoop yarn、mapreduce 2. 已知需求并发文件数和文件大小（本次测试为并发读写 2000 个 100M 文件）		
测试步骤	1. 执行清除缓存操作（每个节点）： <pre>echo 3 > /proc/sys/vm/drop_caches</pre>  2. 切换用户至 hdfs: <pre>su hdfs</pre> 3. 执行 clean 操作，清理原有测试数据及缓存 <pre>hadoop jar /usr/hdp/3.0.1.0-187/hadoop-mapreduce/hadoop-mapreduce-client-jobclient-tests.jar TestDFSIO -clean</pre>  4. 执行 write 操作，如写入 2000 个文件，每个文件 100MB: <pre>hadoop jar /usr/hdp/3.0.1.0-187/hadoop-mapreduce/hadoop-mapreduce-client-jobclient-tests.jar TestDFSIO -write -nrFiles 2000 -fileSize 100MB -compression org.apache.hadoop.io.compress.SnappyCodec -resFile /tmp/result.log</pre>		


```
[hdfs@manager opt]$ hadoop jar /usr/hdp/3.0.1.0-187/hadoop-mapreduce/hadoop-mapreduce-client-jobclient-tests.jar TestDFSIO -write -nrfiles 2000 -filesize 100MB -compression org.apache.hadoop
io.compress.SnappyCodec -resFile /tmp/result.log
21/05/07 12:02:40 INFO fs.TestDFSIO: TestDFSIO:1.0
21/05/07 12:02:40 INFO fs.TestDFSIO: nrfiles = 2000
21/05/07 12:02:40 INFO fs.TestDFSIO: nrbytes (MB) = 100.0
21/05/07 12:02:40 INFO fs.TestDFSIO: bufferSize = 1000000
21/05/07 12:02:40 INFO fs.TestDFSIO: baseDir = /benchmarks/TestDFSIO
21/05/07 12:02:40 INFO fs.TestDFSIO: compressionClass = org.apache.hadoop.io.compress.SnappyCodec
21/05/07 12:02:41 INFO fs.TestDFSIO: creating control file: 104857600 bytes, 2000 files
21/05/07 12:02:50 INFO fs.TestDFSIO: created control files for: 2000 files
21/05/07 12:02:50 INFO client.AHSProxy: Connecting to Application History server at master1.zzdx.com/172.28.92.2:10200
21/05/07 12:02:50 INFO client.AHSProxy: Connecting to Application History server at master1.zzdx.com/172.28.92.2:10200
21/05/07 12:03:00 INFO mapreduce.JobResourcePlanner: Disabling Erasure Coding for path: /user/hdfs/.staging/job_1620291850698_0001
21/05/07 12:03:00 INFO mapred.FileInputFormat: Total input files to process : 2000
21/05/07 12:03:00 INFO mapreduce.JobSubmitter: number of splits:2000
21/05/07 12:03:00 INFO mapreduce.JobSubmitter: Submitting Tokens for job: job_1620291850698_0001
21/05/07 12:03:00 INFO mapreduce.JobSubmitter: Executing with tokens: []
21/05/07 12:03:00 INFO conf.Configuration: found resource resource-types.xml at file:/etc/hadoop/3.0.1.0-187/0/resource-types.xml
21/05/07 12:03:01 INFO impl.YarnClientImpl: Submitted application application_1620291850698_0001
21/05/07 12:03:01 INFO mapreduce.Job: The url to track the job: http://master1.zzdx.com:8088/proxy/application_1620291850698_0001/
21/05/07 12:03:01 INFO mapreduce.Job: Running job: job_1620291850698_0001
21/05/07 12:03:06 INFO mapreduce.Job: Job job_1620291850698_0001 running in uber mode : false
21/05/07 12:03:06 INFO mapreduce.Job: map 0% reduce 0%
21/05/07 12:03:12 INFO mapreduce.Job: map 2% reduce 0%
21/05/07 12:03:13 INFO mapreduce.Job: map 3% reduce 0%
21/05/07 12:03:15 INFO mapreduce.Job: map 4% reduce 0%
21/05/07 12:03:16 INFO mapreduce.Job: map 5% reduce 0%
21/05/07 12:03:18 INFO mapreduce.Job: map 7% reduce 0%
21/05/07 12:03:19 INFO mapreduce.Job: map 8% reduce 0%
```

5. 记录并发写测试结果：记录 Test exec time

```
Shuffled Maps =2000
Failed Shuffles=0
Merged Map outputs=2000
GC time elapsed (ms)=267220
CPU time spent (ms)=6202180
Physical memory (bytes) snapshot=3008898084864
Virtual memory (bytes) snapshot=43518384734208
Total committed heap usage (bytes)=5331986939904
Peak Map Physical memory (bytes)=2522152960
Peak Map Virtual memory (bytes)=21745049600
Peak Reduce Physical memory (bytes)=635367424
Peak Reduce Virtual memory (bytes)=40298090496

Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0

File Input Format Counters
Bytes Read=228890
File Output Format Counters
Bytes Written=90
21/05/07 12:05:06 INFO fs.TestDFSIO: ----- TestDFSIO ----- : write
21/05/07 12:05:06 INFO fs.TestDFSIO: Date & time: Fri May 07 12:05:06 CST 2021
21/05/07 12:05:06 INFO fs.TestDFSIO: Number of files: 2000
21/05/07 12:05:06 INFO fs.TestDFSIO: Total MBytes processed: 200000
21/05/07 12:05:06 INFO fs.TestDFSIO: Throughput mb/sec: 1142.88
21/05/07 12:05:06 INFO fs.TestDFSIO: Average IO rate mb/sec: 1182.79
21/05/07 12:05:06 INFO fs.TestDFSIO: IO rate std deviation: 193.52
21/05/07 12:05:06 INFO fs.TestDFSIO: Test exec time sec: 126.64
21/05/07 12:05:06 INFO fs.TestDFSIO:
[hdfs@manager opt]$
```

6. 执行 read 操作，如读 2000 个文件，每个文件 100B:

hadoop jar /usr/hdp/3.0.1.0-187/hadoop-mapreduce/hadoop-mapreduce-client-jobclient-tests.jar TestDFSIO -read -nrFiles 2000 -fileSize 100MB -compression org.apache.hadoop.io.compress.SnappyCodec -resFile /tmp/result.log

```
[hdfs@manager opt]$ hadoop jar /usr/hdp/3.0.1.0-187/hadoop-mapreduce/hadoop-mapreduce-client-jobclient-tests.jar TestDFSIO -read -nrfiles 2000 -filesize 100MB -compression org.apache.had
io.compress.SnappyCodec -resFile /tmp/result.log
21/05/07 12:07:40 INFO fs.TestDFSIO: TestDFSIO:1.0
21/05/07 12:07:40 INFO fs.TestDFSIO: nrfiles = 2000
21/05/07 12:07:40 INFO fs.TestDFSIO: nrbytes (MB) = 100.0
21/05/07 12:07:40 INFO fs.TestDFSIO: bufferSize = 1000000
21/05/07 12:07:40 INFO fs.TestDFSIO: baseDir = /benchmarks/TestDFSIO
21/05/07 12:07:40 INFO fs.TestDFSIO: compressionClass = org.apache.hadoop.io.compress.SnappyCodec
21/05/07 12:07:40 INFO fs.TestDFSIO: creating control file: 104857600 bytes, 2000 files
21/05/07 12:07:50 INFO fs.TestDFSIO: created control files for: 2000 files
21/05/07 12:07:50 INFO client.AHSProxy: Connecting to Application History server at master1.zzdx.com/172.28.92.2:10200
21/05/07 12:07:50 INFO client.AHSProxy: Connecting to Application History server at master1.zzdx.com/172.28.92.2:10200
21/05/07 12:07:50 INFO mapreduce.JobResourcePlanner: Disabling Erasure Coding for path: /user/hdfs/.staging/job_1620291850698_0002
21/05/07 12:07:50 INFO mapred.FileInputFormat: Total input files to process : 2000
21/05/07 12:07:50 INFO mapreduce.JobSubmitter: number of splits:2000
21/05/07 12:07:50 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1620291850698_0002
21/05/07 12:07:50 INFO mapreduce.JobSubmitter: Executing with tokens: []
21/05/07 12:07:50 INFO conf.Configuration: found resource resource-types.xml at file:/etc/hadoop/3.0.1.0-187/0/resource-types.xml
21/05/07 12:07:50 INFO impl.YarnClientImpl: Submitted application application_1620291850698_0002
21/05/07 12:07:50 INFO mapreduce.Job: The url to track the job: http://master1.zzdx.com:8088/proxy/application_1620291850698_0002/
21/05/07 12:07:50 INFO mapreduce.Job: Running job: job_1620291850698_0002
21/05/07 12:08:02 INFO mapreduce.Job: Job job_1620291850698_0002 running in uber mode : false
21/05/07 12:08:02 INFO mapreduce.Job: map 0% reduce 0%
21/05/07 12:08:09 INFO mapreduce.Job: map 2% reduce 0%
21/05/07 12:08:10 INFO mapreduce.Job: map 3% reduce 0%
21/05/07 12:08:15 INFO mapreduce.Job: map 8% reduce 0%
21/05/07 12:08:16 INFO mapreduce.Job: map 9% reduce 0%
21/05/07 12:08:18 INFO mapreduce.Job: map 10% reduce 0%
```

7. 记录并发读测试结果：记录 Test exec time

	<pre>Shuffled Maps =2000 Failed Shuffles=0 Merged Map outputs=2000 GC time elapsed (ms)=275904 CPU time spent (ms)=4864420 Physical memory (bytes) snapshot=3946777882624 Virtual memory (bytes) snapshot=43512111677440 Total committed heap usage (bytes)=5352480833536 Peak Map Physical memory (bytes)=2501877760 Peak Map Virtual memory (bytes)=21741895680 Peak Reduce Physical memory (bytes)=620388352 Peak Reduce Virtual memory (bytes)=40295272448 Shuffle Errors BAD_ID=0 CONNECTION=0 IO_ERROR=0 WRONG_LENGTH=0 WRONG_MAP=0 WRONG_REDUCE=0 File Input Format Counters Bytes Read=228890 File Output Format Counters Bytes Written=90 21/05/07 12:10:01 INFO fs.TestDFSIO: ----- TestDFSIO ----- : read 21/05/07 12:10:01 INFO fs.TestDFSIO: Date & time: Fri May 07 12:10:01 CST 2021 21/05/07 12:10:01 INFO fs.TestDFSIO: Number of files: 2000 21/05/07 12:10:01 INFO fs.TestDFSIO: Total MBytes processed: 200000 21/05/07 12:10:01 INFO fs.TestDFSIO: Throughput mb/sec: 1187.85 21/05/07 12:10:01 INFO fs.TestDFSIO: Average IO rate mb/sec: 1245.62 21/05/07 12:10:01 INFO fs.TestDFSIO: IO rate std deviation: 255.23 21/05/07 12:10:01 INFO fs.TestDFSIO: Test exec time sec: 125.12 21/05/07 12:10:01 INFO fs.TestDFSIO: [hdfs@manager opt]\$</pre>
完成标准	<ol style="list-style-type: none">1. 测试正常结束2. 记录测试任务调优过程参数和任务耗时3. 关键参数已调优；测试值优于目标值；集群资源已合理利用；已到测试 deadline；满足测试退出标准4. 记录最优测试结果截图和集群资源占用截图
备注	

3.2.1.2 测试脚本



- 说明：
- (1) 对于 yarn 相关参数如 yarn.scheduler.minimum-allocation-mb 和 yarn.scheduler.maximum-allocation-mb 需要通过 web UI 进行调参操作，Map 内存/Vcores 等参数可以通过-D 传参的方式调优。
 - (2) 脚本仅含调优中部分关键调优参数，具体调参和调参值需要根据具体环境配置以及业务运行情况而定。
 - (3) 若确认参数调整范围，可优化脚本，通过传参等方式实现自动化运行。

3.2.1.3 测试结果

测试项	目标值	测试值	测试结果
HDFS 写 2000 个 100M 文件耗时	<144.752s	126.64s	通过
HDFS 读 2000 个 100M 文件耗时	<126.653s	125.12s	通过

3.2.2 Nosql 数据库并发读写和查询（HBase）

3.2.2.1 测试用例




性能标识	DFP-002	编写人员	申晓青
测试目标	测试当前环境 HBase 数据入库、随机读、随机写、随机扫描、读写混合、readmostly 场景是否满足目标值，挖掘最优性能		
前提条件	1. 环境已安装 HDFS、Hadoop yarn、mapreduce、HBase 2. YCSB 已上传并配置完成		
测试步骤	<p>1. 执行清除缓存操作（每个节点）：</p> <pre>echo 3 > /proc/sys/vm/drop_caches</pre> <p>2. 执行 hbase shell 进入 hbase 客户端</p> <p>3. 创建 hbase 分区表</p> <pre>create 'usertable',{NAME => 'cf',COMPRESSION=>'GZ',{SPLITS => (1...200).map{ i "user#{1000+i*(9999-1000)/200}"}}</pre> <p>说明：可使用 GZ 压缩也可不使用，根据调优情况决定；分区可为 200 也可为其他的如 80,120 等</p> <p>4. 执行 hbase_load.sh 脚本进行数据入库操作</p> <pre>./hbase_load.sh 并发数 表名 列簇名 记录条数 线程数</pre> <p>5. 记录 Load 数据入库结果：记录任务执行时间</p> <p>说明：测试 load 入库极限性能采用多并发多线程的方式进行，此处的任务执行时间为随机选取某一进程的运行时间即可。</p> <p>6. 执行 hbase_read.sh 脚本进行 100%随机读操作：</p> <pre>./hbase_read.sh 并发数 表名 列簇名 操作条数 线程数</pre>		

	<p>7. 记录 100%随机读执行结果：记录任务执行时间</p> <p>说明：测试 100% 随机读极限性能采用多并发多线程的方式进行，此处的任务执行时间为随机选取某一进程的运行时间即可。</p> <p>8. 执行 hbase_write.sh 脚本进行 100%随机写操作：</p> <p>./hbase_write.sh 并发数 表名 列簇名 操作条数 线程数</p> <p>9. 记录 100%随机写执行结果：记录任务执行时间【把对应字段加一下】</p> <p>说明：测试 100% 随机写极限性能采用多并发多线程的方式进行，此处的任务执行时间为随机选取某一进程的运行时间即可</p> <p>10. 执行 hbase_50%read&50%write.sh 脚本进行读写混合操作（50%读 50%写）：</p> <p>./hbase_50%read&50%write.sh 并发数 表名 列簇名 操作条数 线程数</p> <p>11. 记录读写混合执行结果：记录任务执行时间【把对应字段加一下】</p> <p>说明：测试读写混合极限性能采用多并发多线程的方式进行，此处的任务执行时间为随机选取某一进程的运行时间即可</p> <p>12. 执行 hbase_readmostly.sh 脚本进行读多写少操作（95%读 5%写）：</p> <p>./ hbase_readmostly.sh 并发数 表名 列簇名 操作条数 线程数</p> <p>13. 记录 readmostly 执行结果：记录任务执行时间【把对应字段加一下】</p> <p>说明：测试 readmostly 极限性能采用多并发多线程的方式进行，此处的任务执行时间为随机选取某一进程的运行时间即可</p> <p>14. 执行 hbase_scan.sh 脚本进行顺序扫描操作：</p> <p>./ hbase_scan.sh 并发数 表名 列簇名 操作条数 线程数</p> <p>15. 记录顺序扫描执行结果：记录任务执行时间【把对应字段加一下】</p> <p>说明：测试顺序扫描极限性能采用多并发多线程的方式进行，此处的任务执行时间为随机选取某一进程的运行时间即可</p> <p>16. 执行 hbase_50%write&50%scan.sh 脚本进行 50%随机写+50%扫描读操作：</p> <p>./ hbase_50%write&50%scan.sh 并发数 表名 列簇名 操作条数 线程数</p> <p>17. 记录 50%随机写+50%扫描读执行结果：记录任务执行时间【把对应字段加一下】</p> <p>说明：测试 50% 随机写+50% 扫描读极限性能采用多并发多线程的方式进行，此处的任务执行时间为随机选取某一进程的运行时间即可</p>
完成标准	<p>1. 测试正常结束</p> <p>2. 记录测试任务调优过程参数和任务耗时，计算平均每节点写入记录条数</p> <p>3. 关键参数已调优；测试值优于目标值；集群资源已合理利用；已到测试 deadline；满足测试退出标准</p> <p>4. 记录最优测试结果截图</p>

备注	Hbase 相关读/写/扫描/读写扫描混合场景测试均需建立在 load 操作执行的基础上
----	--

3.2.2.2 测试脚本

脚本示例：




hbase_load.sh hbase_50%read&hbase_read.sh.sh
50%write.sh

说明：（1）每个场景测试脚本的差异性在于 workloads/workloadX 文件中的 read、update、insert、scan 等配比不一致。

（2）在测试过程中，可直接修改对应 workloadX 后使用同一脚本测试；也可使用传参的方式覆盖配置文件原有参数值。

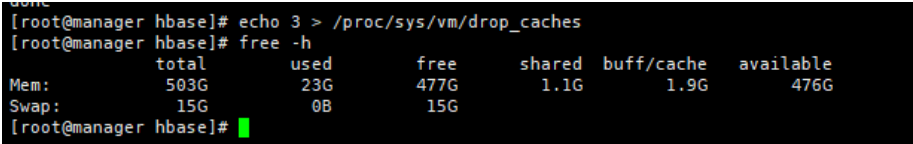
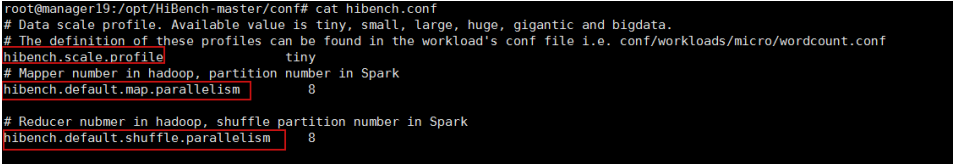
3.2.2.3 测试结果

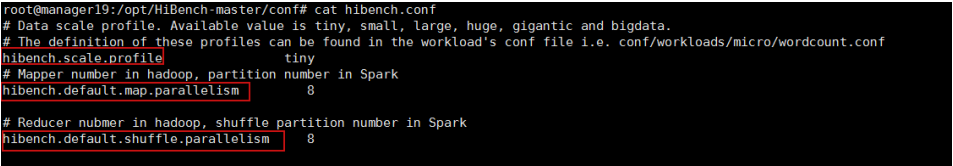
测试项	目标值	测试值	比较 (测试值/目标值)
Load	66283 Records/s/Node	105764.15 Records/s/Node	1.6
100%随机读	150000 Records/s/Node	146669.57 Records/s/Node	0.98
读写混合	60000 Records/s/Node	100728.87 Records/s/Node	1.68
Readmostly	79986 Records/s/Node	80662.51 Records/s/Node	1.01
100%随机写	64000 Records/s/Node	255357.39 Records/s/Node	3.99
顺序扫描场景	46000 Records/s/Node	417379.92 Records/s/Node	9.07
50%随机写 + 50%扫描场景	61000 Records/s/Node	304359.84 Records/s/Node	4.99

3.3 计算性能测试

3.3.1 离线计算能力（MapReduce）

3.3.1.1 测试用例

性能标识	DFP-003	编写人员	申晓青
测试目标	测试当前环境 MR 在词频统计和排序场景应用中每节点处理能力是否满足目标值，挖掘最优的性能		
前提条件	1. 环境已安装 HDFS、Hadoop yarn、mapreduce 2. HiBench 工具已上传并配置		
测试步骤	<div>1. 执行清除缓存操作（每个节点）：</div> <div>echo 3 > /proc/sys/vm/drop_caches</div> <div></div> <div>2. 设置 wordcount 操作要处理数据量大小、map/reduce 参数：</div> <div>vim \$HiBench_Home/conf/hibench.conf</div> <div></div> <div>3. 生成 wordcount 测试数据数据</div> <div>su hdfs</div> <div>cd \$HiBench_Home/bin/workloads/micro/wordcount/prepare</div> <div>sh prepare.sh</div> <div>4. 执行清除缓存操作（每个节点）：</div> <div>echo 3 > /proc/sys/vm/drop_caches</div> <div>5. 运行词频统计任务：</div> <div>sh \$HiBench_Home/bin/workloads/micro/wordcount/hadoop/run.sh</div> <div>6. 查看生成的测试报告 report/hibench.report，记录 wordcount 任务运行时间</div> <div>cat \$HiBench_Home/report/hibench.report</div> <div>7. 执行清除缓存操作（每个节点）：</div> <div>echo 3 > /proc/sys/vm/drop_caches</div> <div>8. 设置 terasort 操作要处理数据量大小、map/reduce 参数：</div>		

	<pre>vim \$HiBench_Home/conf/hibench.conf</pre>  <pre>root@manager19:/opt/HiBench-master/conf# cat hibench.conf # Data scale profile. Available value is tiny, small, large, huge, gigantic and bigdata. # The definition of these profiles can be found in the workload's conf file i.e. conf/workloads/micro/wordcount.conf hibench.scale.profile tiny # Mapper number in hadoop, partition number in Spark hibench.default.map.parallelism 8 # Reducer nubmer in hadoop, shuffle partition number in Spark hibench.default.shuffle.parallelism 8</pre> <p>9. 生成 terasort 测试数据数据</p> <pre>su hdfs cd \$HiBench_Home/bin/workloads/micro/terasort/prepare sh prepare.sh</pre> <p>10. 执行清除缓存操作（每个节点）：</p> <pre>echo 3 > /proc/sys/vm/drop_caches</pre> <p>11. 运行排序任务：</p> <pre>sh \$HiBench_Home/bin/workloads/micro/terasort/hadoop/run.sh</pre> <p>12. 查看生成的测试报告 report/hibench.report，记录 terasort 任务运行时间</p> <pre>cat \$HiBench_Home/report/hibench.report</pre>
完成标准	<p>1. 测试正常结束</p> <p>2. 记录测试任务调优过程参数和任务耗时</p> <p>3. 关键参数已调优；测试值优于目标值；集群资源已合理利用；已到测试 deadline；满足测试退出标准</p> <p>4. 记录最优测试结果截图和集群资源占用截图</p>
备注	

3.3.1.2 测试结果

测试项	目标值	测试值	比较 测试值/目标值
Wordcount: 平均每节点处理能力（1T）	>8.23 G/min/node	12.88 G/min/node	1.5
TeraSort: 平均每节点处理能（300G）	>7.78 G/min/node	15.65 G/min/node	2

3.3.2 内存计算能力（Spark）

3.3.2.1 测试用例

性能标识	DFP-004	编写人员	申晓青
测试目标	测试当前环境 Spark 在词频统计和排序场景应用中每节点处理能力是否满足目标值，挖掘最优的性能		
前提条件	1. 环境已安装 HDFS、Hadoop yarn、mapreduce、spark 2. HiBench 工具已上传并配置		
测试步骤	<div>1. 执行清除缓存操作（每个节点）：</div> <div>echo 3 > /proc/sys/vm/drop_caches</div> <div>2. 设置 wordcount 操作要处理数据量大小、map/reduce 参数：</div> <div>vim \$HiBench_Home/conf/hibench.conf</div> <div><pre>root@manager19:/opt/HiBench-master/conf# cat hibench.conf # Data scale profile. Available value is tiny, small, large, huge, gigantic and bigdata. # The definition of these profiles can be found in the workload's conf file i.e. conf/workloads/micro/wordcount.conf hibench.scale.profile tiny # Mapper number in hadoop, partition number in Spark hibench.default.map.parallelism 8 # Reducer nubmer in hadoop, shuffle partition number in Spark hibench.default.shuffle.parallelism 8</pre></div> <div>3. 生成 wordcount 测试数据数据</div> <div>su hdfs</div> <div>cd \$HiBench_Home/bin/workloads/micro/wordcount/prepare</div> <div>sh prepare.sh</div> <div>4. 执行清除缓存操作（每个节点）：</div> <div>echo 3 > /proc/sys/vm/drop_caches</div> <div>5. 设置任务运行关键资源参数配置：</div> <div>vim \$HiBench_Home/conf/spark.conf</div> <div>配置合适的参数值：hibench.yarn.executor.num、hibench.yarn.executor.cores、spark.executor.memory、spark.driver.memory</div> <div>6. 运行词频统计任务：</div> <div>sh \$HiBench_Home/bin/workloads/micro/wordcount/spark/run.sh</div> <div>7. 查看生成的测试报告 report/hibench.report，记录 wordcount 任务运行时间</div> <div>cat \$HiBench_Home/report/hibench.report</div> <div>8. 执行清除缓存操作（每个节点）：</div> <div>echo 3 > /proc/sys/vm/drop_caches</div> <div>9. 设置 terasort 操作要处理数据量大小、map/reduce 参数：</div> <div>vim \$HiBench_Home/conf/hibench.conf</div>		

	<pre>root@manager19:/opt/HiBench-master/conf# cat hibench.conf # Data scale profile. Available value is tiny, small, large, huge, gigantic and bigdata. # The definition of these profiles can be found in the workload's conf file i.e. conf/workloads/micro/wordcount.conf hibench.scale.profile tiny # Mapper number in hadoop, partition number in Spark hibench.default.map.parallelism 8 # Reducer nubmer in hadoop, shuffle partition number in Spark hibench.default.shuffle.parallelism 8</pre> <p>10. 生成 terasort 测试数据数据</p> <pre>su hdfs cd \$HiBench_Home/bin/workloads/micro/terasort/prepare sh prepare.sh</pre> <p>11. 执行清除缓存操作（每个节点）：</p> <pre>echo 3 > /proc/sys/vm/drop_caches</pre> <p>12. 设置任务运行关键资源参数配置：</p> <pre>vim \$HiBench_Home/conf/spark.conf</pre> <p>配置合适的参数值：hibench.yarn.executor.num、hibench.yarn.executor.cores、spark.executor.memory、spark.driver.memory</p> <p>13. 运行词频统计任务：</p> <pre>sh \$HiBench_Home/bin/workloads/micro/terasort/spark/run.sh</pre> <p>14. 查看生成的测试报告 report/hibench.report，记录 terasort 任务运行时间</p> <pre>cat \$HiBench_Home/report/hibench.report</pre>
完成标准	<p>1. 测试正常结束</p> <p>2. 记录测试任务调优过程参数和任务耗时</p> <p>3. 关键参数已调优；测试值优于目标值；集群资源已合理利用；已到测试 deadline；满足测试退出标准</p> <p>4. 记录最优测试结果截图和集群资源占用截图</p>
备注	

3.3.2.2 测试结果

测试项	目标值	测试值	比较 测试值/目标值
Wordcount: 平均每节点处理能力 (1T)	>50.4 G/min/node	96.39 G/min/node	1.9

TeraSort: 平均每节点 处理能 (300G)	>6.9 G/min/node	16.5 G/min/node	2.3
-------------------------------	-----------------	-----------------	-----

4 用户确认

测试结论	合格
测试人员签字	荆菲
用户签字	任超
日期	21.5.7