目录

[集群准备 5](#_Toc37431040)

[测试准备 5](#_Toc37431041)

[应用JVM 5](#_Toc37431042)

[功能测试步骤 5](#_Toc37431043)

[集群 5](#_Toc37431044)

[多主机平台支持 5](#_Toc37431045)

[双集群数据同步 5](#_Toc37431046)

[安全方案 5](#_Toc37431047)

[连接能力 6](#_Toc37431048)

[监控 6](#_Toc37431049)

[热更新 6](#_Toc37431050)

[数据均衡 6](#_Toc37431051)

[事件 6](#_Toc37431052)

[缓存事件 6](#_Toc37431053)

[分区丢失系统事件 7](#_Toc37431054)

[缓存 7](#_Toc37431055)

[SQL能力与语法 7](#_Toc37431056)

[动态修改缓存配置 7](#_Toc37431057)

[单cache熔断 8](#_Toc37431058)

[数据导出 8](#_Toc37431059)

[数据导出一致性 8](#_Toc37431060)

[数据失效策略 9](#_Toc37431061)

[事物 9](#_Toc37431062)

[缓存一致性 9](#_Toc37431063)

[EP原子性 9](#_Toc37431064)

[全复制一致性 9](#_Toc37431065)

[近缓存一致性 10](#_Toc37431066)

[分区缓存一致性 10](#_Toc37431067)

[cachestore 10](#_Toc37431068)

[二级索引支持 10](#_Toc37431069)

[第三方持久化 10](#_Toc37431070)

[事物 11](#_Toc37431071)

[性能测试步骤 11](#_Toc37431072)

[随机写 11](#_Toc37431073)

[Partition put 11](#_Toc37431074)

[Partition put 单条 11](#_Toc37431075)

[Partition put 同key 11](#_Toc37431076)

[Partition stream 12](#_Toc37431077)

[Partition stream单条 12](#_Toc37431078)

[Near put 12](#_Toc37431079)

[大对象Partition put 12](#_Toc37431080)

[大对象Partition stream 12](#_Toc37431081)

[Partition多表多次put 13](#_Toc37431082)

[EP put(多笔数据) 13](#_Toc37431083)

[EP put 单条 13](#_Toc37431084)

[随机读 13](#_Toc37431085)

[Partition get 13](#_Toc37431086)

[Partition get 单条 14](#_Toc37431087)

[Partition get 同表同key 14](#_Toc37431088)

[Near get 14](#_Toc37431089)

[大对象Partition get 14](#_Toc37431090)

[Partition多表多次get 15](#_Toc37431091)

[EP get(多笔数据) 15](#_Toc37431092)

[EP get 单条 15](#_Toc37431093)

[随机读写 15](#_Toc37431094)

[Partition get&put 1主1备 15](#_Toc37431095)

[Partition get&put 同表同key 1主1备 16](#_Toc37431096)

[Partition get&put stream 1主1备 16](#_Toc37431097)

[Near get&put 1主1备 16](#_Toc37431098)

[大对象Partition get&put 1主1备 16](#_Toc37431099)

[大对象Partition get&put stream 1主1备 17](#_Toc37431100)

[Partition多表多次get&put 1主1备 17](#_Toc37431101)

[EP get&put(多笔数据) 主1备 17](#_Toc37431102)

[Partition get&put 1主2备 17](#_Toc37431103)

[Partition get&put 同表同key 1主2备 18](#_Toc37431104)

[Partition get&put stream 1主2备 18](#_Toc37431105)

[Near get&put 1主2备 18](#_Toc37431106)

[大对象Partition get&put 1主2备 18](#_Toc37431107)

[大对象Partition get&put stream1主2备 19](#_Toc37431108)

[Partition多表多次get&put 1主2备 19](#_Toc37431109)

[EP get&put(多笔数据) 1主2备 19](#_Toc37431110)

[再平衡 19](#_Toc37431111)

[同步 19](#_Toc37431112)

[异步 20](#_Toc37431113)

[数据膨胀率 本地可测，生产未知 20](#_Toc37431114)

[java对象 1JVM 20](#_Toc37431115)

[Json 对象 1JVM 21](#_Toc37431116)

[二进制 对象 1JVM 21](#_Toc37431117)

[异步EP 21](#_Toc37431118)

[EP get&put(多笔数据) 同步 21](#_Toc37431119)

[EP get&put(多笔数据) 异步 21](#_Toc37431120)

[扩展性 22](#_Toc37431121)

[Partition多表多次put 1节点2JVM 22](#_Toc37431122)

[EP put(多笔数据) 1节点2JVM 22](#_Toc37431123)

[Partition多表多次get 1节点2JVM 22](#_Toc37431124)

[EP get(多笔数据) 1节点2JVM 22](#_Toc37431125)

[Partition多表多次put 1节点3JVM 23](#_Toc37431126)

[EP put(多笔数据) 1节点3JVM 23](#_Toc37431127)

[Partition多表多次get 1节点3JVM 23](#_Toc37431128)

[EP get(多笔数据) 1节点3JVM 23](#_Toc37431129)

[Partition多表多次put 2节点2JVM 24](#_Toc37431130)

[EP put(多笔数据) 2节点2JVM 24](#_Toc37431131)

[Partition多表多次get 2节点2JVM 24](#_Toc37431132)

[EP get(多笔数据) 2节点2JVM 24](#_Toc37431133)

[Partition多表多次put 3节点3JVM 25](#_Toc37431134)

[EP put(多笔数据) 3节点3JVM 25](#_Toc37431135)

[Partition多表多次get 3节点3JVM 25](#_Toc37431136)

[EP get(多笔数据) 3节点3JVM 25](#_Toc37431137)

[稳定性 26](#_Toc37431138)

[get/put& ep get/put 500W数据量 6JVM 26](#_Toc37431139)

[get/put& ep get/put 1000W数据量 6JVM 26](#_Toc37431140)

[cachestore写入方式 26](#_Toc37431141)

[Partition get&put 非后写 1JVM 26](#_Toc37431142)

[Partition get&put 后写 1JVM 26](#_Toc37431143)

[cachestore数据载入 1JVM 27](#_Toc37431144)

[不持久化 数据库串行 27](#_Toc37431145)

[不持久化 数据库并行 27](#_Toc37431146)

[持久化 数据库串行 27](#_Toc37431147)

[持久化 数据库并行 27](#_Toc37431148)

[Failover 27](#_Toc37431149)

[Get同时增加节点 1M流量再平衡 27](#_Toc37431150)

[Get同时减少节点 1M流量再平衡 28](#_Toc37431151)

[Put同时增加节点 1M流量再平衡 28](#_Toc37431152)

[Put同时减少节点 1M流量再平衡 29](#_Toc37431153)

[Put stream同时增加节点 1M流量再平衡 29](#_Toc37431154)

[Put stream同时减少节点 1M流量再平衡 29](#_Toc37431155)

[EP get同时增加节点 1M流量再平衡 30](#_Toc37431156)

[EP get同时减少节点 1M流量再平衡 30](#_Toc37431157)

[EP put同时增加节点 1M流量再平衡 31](#_Toc37431158)

[EP put同时减少节点 1M流量再平衡 31](#_Toc37431159)

[Get同时增加节点 5M流量再平衡 32](#_Toc37431160)

[Get同时减少节点 5M流量再平衡 32](#_Toc37431161)

[Put同时增加节点 5M流量再平衡 32](#_Toc37431162)

[Put同时减少节点 5M流量再平衡 33](#_Toc37431163)

[Put stream同时增加节点 5M流量再平衡 33](#_Toc37431164)

[Put stream同时减少节点 5M流量再平衡 34](#_Toc37431165)

[EP get同时增加节点 5M流量再平衡 34](#_Toc37431166)

[EP get同时减少节点 5M流量再平衡 34](#_Toc37431167)

[EP put同时增加节点 5M流量再平衡 35](#_Toc37431168)

[EP put同时减少节点 5M流量再平衡 35](#_Toc37431169)

[Get同时增加节点 10M流量再平衡 36](#_Toc37431170)

[Get同时减少节点 10M流量再平衡 36](#_Toc37431171)

[Put同时增加节点 10M流量再平衡 37](#_Toc37431172)

[Put同时减少节点 10M流量再平衡 37](#_Toc37431173)

[Put stream同时增加节点 10M流量再平衡 37](#_Toc37431174)

[Put stream同时减少节点 10M流量再平衡 38](#_Toc37431175)

[EP get同时增加节点 10M流量再平衡 38](#_Toc37431176)

[EP get同时减少节点 10M流量再平衡 39](#_Toc37431177)

[EP put同时增加节点 10M流量再平衡 39](#_Toc37431178)

[EP put同时减少节点 10M流量再平衡 39](#_Toc37431179)

[关联表 40](#_Toc37431180)

[并置小对象导入 非测试项 40](#_Toc37431181)

[并置小对象查询 40](#_Toc37431182)

[非并置小对象查询 40](#_Toc37431183)

[并置大对象导入 非测试项 41](#_Toc37431184)

[并置大对象查询 41](#_Toc37431185)

[非并置大对象查询 41](#_Toc37431186)

[区间计算 41](#_Toc37431187)

[小对象导入 非测试项仅做数据准备 41](#_Toc37431188)

[小对象非索引查询 41](#_Toc37431189)

[小对象索引查询 42](#_Toc37431190)

[大对象导入 非测试项仅做数据准备 42](#_Toc37431191)

[大对象非索引查询 42](#_Toc37431192)

[大对象索引查询 42](#_Toc37431193)

[模糊查询 43](#_Toc37431194)

[小对象导入 非测试项仅做数据准备 43](#_Toc37431195)

[小对象模糊查询 43](#_Toc37431196)

[大对象导入 非测试项仅做数据准备 43](#_Toc37431197)

[大对象模糊查询 43](#_Toc37431198)

[持久化 44](#_Toc37431199)

[SSD盘写 44](#_Toc37431200)

[SSD盘读 44](#_Toc37431201)

[NAS盘写 44](#_Toc37431202)

[NAS盘读 44](#_Toc37431203)

[本地盘写 45](#_Toc37431204)

[本地盘读 45](#_Toc37431205)

[持久化恢复 2节点2JVM 45](#_Toc37431206)

[SSD盘恢复 45](#_Toc37431207)

[NAS盘恢复 45](#_Toc37431208)

[本地盘恢复 46](#_Toc37431209)

# 集群准备

export JVM\_OPTS="-Djava.net.preferIPv4Stack=true -Xmx8G -Xms8G -XX:MaxMetaspaceSize=512m -XX:MaxDirectMemorySize=512m -XX:+AlwaysPreTouch -XX:+UseG1GC -XX:+ScavengeBeforeFullGC"

nohup /apphome/ignite/ignite-2.8.0/bin/ignite.sh /apphome/ignite/config/node-config.xml >> /apphome/ignite/log/log/nohup.out &

tail -f /apphome/ignite/log/log/nohup.out

# 测试准备

## 应用JVM

测试环境：

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomw.nearsmallput.NearSmallPutScript 2000 1 1 1

说明：2000代表总数据量 1 并发数 1 批次数量 1 循环次数

正式环境数据量较大需要增加内存，添加命令

java -Xmx1G -Xms1G -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomw.nearsmallput.NearSmallPutScript

# 功能测试步骤

## 集群

### 多主机平台支持

### 双集群数据同步

### 安全方案

账号验证：配置文件中增加

<property name="authenticationEnabled" value="true"/>

账号密码:ignite/ignite

control.sh 脚本需要增加--user ignite --password ignite

jdbc连接需要使用账号密码连接

### 连接能力

### 监控

集群默认使用jmx监控，可以从rest api 和 jconsole 查看指标

Rest方式

<http://172.32.148.244:8080/ignite?cmd=top&attr=true>

jconsole 方式

使用jconsole输入172.32.148.244:{port} port从进程中查找

### 热更新

1. 集群设置1个节点 打开目录
2. cust目录 3节点分别执行

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.ep.RunEp

日志输出1

1. 单节点停止 切换到cust2目录 执行

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.ep.RunEp

日志输出 1或者2

所有客户端节点更新完成后 日志输出2

### 数据均衡

1.准备数据

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.export.ExportPutScript 2000 1 1 1

2.新节点加入集群，加入基线拓扑，持久化文件中会有新的均衡数据。

## 事件

### 缓存事件

1.2节点集群

2.创建分区缓存，缓存数据，副本数为0

3.打开监听程序

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.event.TestEventListener

4.运行缓存添加删除ep等操作

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.event.TestEventScript

5.在监控控制台中可以看到缓存除clear操作外的所有事件

### 分区丢失系统事件

1.紧跟缓存事件

2.杀掉一个节点

监控控制台会输出 非常多的分区丢失的事件

## 缓存

### SQL能力与语法

使用Dbeaver

### 动态修改缓存配置

1.Cust目录下执行 导入数据

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.structure.ChangeDataPutScript

2.执行查询数据

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.structure.ChangeDataGetScript

3.在dbeaver中执行sql

**ALTER** **TABLE** NEWLAND.CHANGEDATA **DROP** **COLUMN** REMARK1;

**ALTER** **TABLE** NEWLAND.CHANGEDATA **DROP** **COLUMN** REMARK2;

**ALTER** **TABLE** NEWLAND.CHANGEDATA **DROP** **COLUMN** REMARK4;

**ALTER** **TABLE** NEWLAND.CHANGEDATA **DROP** **COLUMN** birthday;

**ALTER** **TABLE** NEWLAND.CHANGEDATA **ADD** **COLUMN** REMARK3 **VARCHAR**;

**INSERT** **INTO** NEWLAND.CHANGEDATA (\_**key**,ID, AINT, ALONG, ADOUBLE) **VALUES** ('32','32',**32**,**32**,**32**)

**UPDATE** NEWLAND.CHANGEDATA **SET** REMARK3='ffffff' **WHERE** id='31'

4.在cust2目录下执行查询数据

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.structure.ChangeDataGetScript

结论

1. 不支持修改副本数，不支持修改字典属性
2. SQL方式支持动态添加字段，删除字段
3. API方式不完全支持动态添加字段，删除字段

### 单cache熔断

1.至少2节点环境 准备数据

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.fuse.FuseInit

1个全复制缓存 1个副本数为0的缓存 1个副本数为1的缓存

2.执行命令

查询全复制缓存

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.fuse.FuseReplicatedScript

查询副本数为0的缓存

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.fuse.FusePartitionOneScript

查询副本数为1的缓存

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.fuse.FusePartitionTwoScript

3.停一个节点 重复步骤2

### 数据导出

1.准备数据

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.export.ExportPutScript 2000 1 1 1

2.表 FREERESOURCE 字段id String,name String,age int

3.执行

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.export.ExportScript /apphome/ignite/cust/1.csv "select count(1), age from FREERESOURCE group by age"

第一个参数为导出文件，第二个参数为可运行sql

### 数据导出一致性

1.准备数据

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.export.ExportPutScript 2000 1 1 1

2.表 FREERESOURCE 字段id String,name String,age int

3.执行

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.export.ExportScript /apphome/ignite/cust/1.csv "select id,name,age from FREERESOURCE"

第一个参数为导出文件，第二个参数为可运行sql

### 数据失效策略

默认20秒数据会失效 自动删除 内存和内持久化

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.TestExpired

### 事物

乐观锁

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.transcation.TranscationOptimisticScript

悲观锁

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.transcation.TranscationPessimisticScript

SQL 事务不建议生产使用

## 缓存一致性

### EP原子性

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.ep.EpScript 20 1 1 1

### 全复制一致性

两节点，插入数据后，修改数据 干掉一个节点，输出读取

1.执行 全复制脚本 长连接客户端

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.consistency.TestReplicatedModeScript

2.干掉一个节点

看步骤1的输出是否一致

### 近缓存一致性

插入数据后修改再读取，是否显示修改过后数据

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.consistency.TestNearModeScript

### 分区缓存一致性

两节点，插入数据后，修改数据 干掉一个节点，输出读取

1.执行 分区缓存脚本 长连接客户端

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.consistency.TestPartitionedModeScript

2.干掉一个节点

看步骤1的输出是否一致

## cachestore

### 二级索引支持

1.执行

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.cachestore.CacheStoreSecondScript

2在DBeave中查看 sql：**SELECT** ID, NAME, REMARK, AUTOMATION\_NAME, AUTOMATION\_AGE, AUTOMATION\_REMARK

**FROM** NEWLAND.EXPIRY;

mysql：select \* from ignite. expiry;

### 第三方持久化

后写：

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.cachestore.CacheStoreBackWriteScript

非后写：

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.cachestore.CacheStoreNoBackWriteScript

### 事物

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.cachestore.CacheStoreTranscationScript

# 性能测试步骤

## 随机写

### Partition put

场景1

for i in {1..1} ; do echo "nohup java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomw.partitionsmallput.PartitionSmallPutScript 100000 1 1 200 "$i" > $i.out &" | bash ;done

场景2

for i in {1..5} ; do echo "nohup java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomw.partitionsmallput.PartitionSmallPutScript 100000 10 1 200 "$i" > $i.out &" | bash ;done

场景3

for i in {1..10} ; do echo "nohup java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomw.partitionsmallput.PartitionSmallPutScript 100000 10 1 200 "$i" > $i.out &" | bash ;done

场景4

节点1

for i in {1..10} ; do echo "nohup java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomw.partitionsmallput.PartitionSmallPutScript 100000 10 1 200 "$i" > $i.out &" | bash ;done

节点2

for i in {1..10} ; do echo "nohup java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomw.partitionsmallput.PartitionSmallPutScript 100000 10 1 200 "$i" > $i.out &" | bash ;done

### Partition put 单条 废弃

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomw.partitionsmallputone.PartitionSmallPutOneScript 2000 1 1 1

场景2

场景3

### Partition put 同key

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomw.partitionsmallputsamekey.PartitionSmallPutSameKeyScript 2000 1 1 1

场景2

for i in {1..5} ; do echo "nohup java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomw.partitionsmallput.PartitionSmallPutScript 100000 10 1 200 "$i" > $i.out &" | bash ;done

场景3

for i in {1..10} ; do echo "nohup java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomw.partitionsmallput.PartitionSmallPutScript 100000 10 1 200 "$i" > $i.out &" | bash ;done

场景4

节点1

for i in {1..10} ; do echo "nohup java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomw.partitionsmallput.PartitionSmallPutScript 100000 10 1 200 "$i" > $i.out &" | bash ;done

节点2

for i in {1..10} ; do echo "nohup java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomw.partitionsmallput.PartitionSmallPutScript 100000 10 1 200 "$i" > $i.out &" | bash ;done

### Partition stream

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomw.partitionsmallstreamput.PartitionSmallSteamPutScript 2000 1 1 1

场景2

场景3

### Partition stream单条

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomw.partitionsmallstreamputone.PartitionSmallSteamPutOneScript 2000 1 1 1

场景2

场景3

### Near put

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomw.nearsmallput.NearSmallPutScript 2000 1 1 1

场景2

场景3

### 大对象Partition put

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomw.partitionbigput.PartitionBigPutScript 2000 1 1 1

场景2

场景3

### 大对象Partition stream

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomw.partitionbigstreamput.PartitionBigSteamPutScript 2000 1 1 1

场景2

场景3

### Partition多表多次put

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomw.partitionmanyput.PartitionmanyPutScript 2000 1 5 1

场景2

场景3

### EP put(多笔数据)

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomw.partitionsmallEPput.PartitionSmallEpPutScript 2000 1 1 1

场景2

场景3

### EP put 单条

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomw.partitionsmallEPputOne.PartitionSmallEpPutOneScript 2000 1 1 1

场景2

场景3

## 随机读

### Partition get

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomr.partitionsmallget.PartitionSmallGetScript 2000 1 1 1

场景2

场景3

### Partition get 单条

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomr.partitionsmallgetone.PartitionSmallGetOneScript 2000 1 1 1

场景2

场景3

### Partition get 同表同key

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomr.partitionsmallgetsamekey.PartitionSmallGetSameKeyScript 1 1 1 1

场景2

场景3

### Near get

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomr.nearsmallget.NearSmallGetScript 2000 1 1 1

场景2

场景3

### 大对象Partition get

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomr.partitionbigget.PartitionBigGetScript 2000 1 1 1

场景2

场景3

### Partition多表多次get

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomr.partitionmanyget.PartitionmanyGetScript 2000 1 5 1

场景2

场景3

### EP get(多笔数据)

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomr.partitionEPget.PartitionEpGetScript 2000 1 1 1

场景2

场景3

### EP get 单条

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomr.partitionEPgetOne.PartitionEpGetOneScript 2000 1 1 1

场景2

场景3

## 随机读写

### Partition get&put 1主1备

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomwr.partitionsmallpg.PartitionSmallPGScript\_2 2000 1 1 1

场景2

场景3

### Partition get&put 同表同key 1主1备

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomwr.partitionsmallpgsamekey.PartitionSmallPGSameKeyScript\_2 2000 1 1 1

场景2

场景3

### Partition get&put stream 1主1备

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomwr.partitionsmallpgstream.PartitionSmallSGScript\_2 2000 1 1 1

场景2

场景3

### Near get&put 1主1备

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomwr.nearsmallpg.NearSmallPGScript\_2 2000 1 1 1

场景2

场景3

### 大对象Partition get&put 1主1备

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomwr.partitionbigpg.PartitionBigPGScript\_2 2000 1 1 1

场景2

场景3

### 大对象Partition get&put stream 1主1备

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomwr.partitionbigpgstream.PartitionBigSGScript\_2 2000 1 1 1

场景2

场景3

### Partition多表多次get&put 1主1备

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomwr.partitionmanypg.PartitionManyPGScript\_2 2000 1 5 1

场景2

场景3

### EP get&put(多笔数据) 主1备

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomwr.partitionsmallEPPG.PartitionSmallEPPGScript\_2 2000 1 1 1

场景2

场景3

### Partition get&put 1主2备

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomwr.partitionsmallpg.PartitionSmallPGScript\_3 2000 1 1 1

场景2

场景3

### Partition get&put 同表同key 1主2备

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomwr.partitionsmallpgsamekey.PartitionSmallPGSameKeyScript\_3 2000 1 1 1

场景2

场景3

### Partition get&put stream 1主2备

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomwr.partitionsmallpgstream.PartitionSmallSGScript\_3 2000 1 1 1

场景2

场景3

### Near get&put 1主2备

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomwr.nearsmallpg.NearSmallPGScript\_3 2000 1 1 1

场景2

场景3

### 大对象Partition get&put 1主2备

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomwr.partitionbigpg.PartitionBigPGScript\_3 2000 1 1 1

场景2

场景3

### 大对象Partition get&put stream1主2备

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomwr.partitionbigpgstream.PartitionBigSGScript\_3 2000 1 1 1

场景2

场景3

### Partition多表多次get&put 1主2备

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomwr.partitionmanypg.PartitionManyPGScript\_3 2000 1 5 1

场景2

场景3

### EP get&put(多笔数据) 1主2备

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomwr.partitionsmallEPPG.PartitionSmallEPPGScript\_3 2000 1 1 1

场景2

场景3

## 流导入

### Put 批量

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.stream.PartitionPutScript 2000 1 1 1

场景2

场景3

### Stream Put 批量

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.stream.PartitionStreamPutScript 2000 1 1 1

场景2

## 再平衡

### 全复制同步

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.rebalance.Rebalance1PutScript 2000 1 1 1

场景2

场景3

### 全复制异步

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.rebalance.Rebalance2PutScript 2000 1 1 1

场景2

场景3

### 分区同步

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.rebalance.Rebalance3PutScript 2000 1 1 1

场景2

场景3

### 分区异步

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.rebalance.Rebalance4PutScript 2000 1 1 1

场景2

场景3

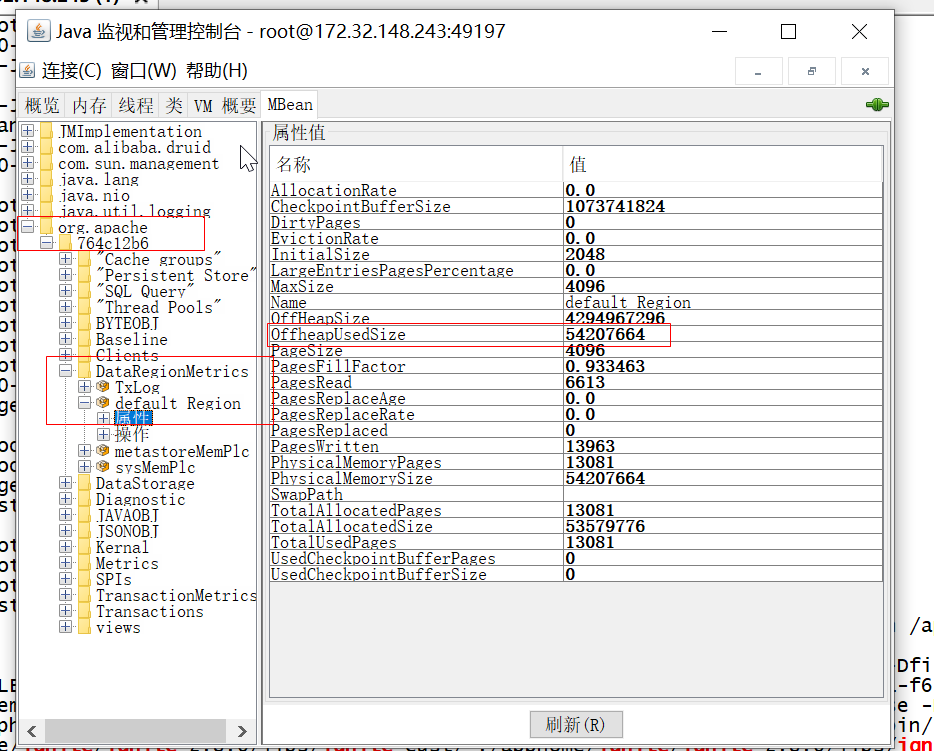
## 数据膨胀率 本地可测，生产未知

步骤1.重启单节点，配置需要开启JMX

步骤2.运行 目的:确定基准值

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.TestExpired

步骤3.启动JDK工具jconsole.exe 连接节点JMX节点（ps 指令查看启动命令，查找JMX端口）:



### java对象 1JVM

步骤4：

运行

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.bulk.JavaPutStreamScript 2000 1 1 1

步骤5:记录OffheapUsedSize大小

### Json 对象 1JVM

步骤6：

运行

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.bulk.JsonPutStreamScript 2000 1 1 1

步骤7:记录OffheapUsedSize大小

### 二进制 对象 1JVM

步骤6：

步骤8：

运行

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.bulk.BytePutStreamScript 2000 1 1 1

步骤9:记录OffheapUsedSize大小

步骤10：等待5分钟到持久化目录下，查看对应缓存占用磁盘空间 du . 命令

步骤11：依次算出内存占用，磁盘占用。膨胀率为 磁盘/内存

## 异步EP

### EP get&put(多笔数据) 同步

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.epasyn.PartitionEPsynScript 2000 1 1 1

场景2

场景3

### EP get&put(多笔数据) 异步

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.epasyn.PartitionEPasynScript 2000 1 1 1

场景2

场景3

## 扩展性

### Partition多表多次put 1节点2JVM

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomw.partitionmanyput.PartitionmanyPutScript 2000 1 1 1

场景2

场景3

### EP put(多笔数据) 1节点2JVM

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomw.partitionsmallEPput.PartitionSmallEpPutScript 2000 1 1 1

场景2

场景3

### Partition多表多次get 1节点2JVM

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomr.partitionmanyget.PartitionmanyGetScript 2000 1 1 1

场景2

场景3

### EP get(多笔数据) 1节点2JVM

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomr.partitionEPget.PartitionEpGetScript 2000 1 1 1

场景2

场景3

### Partition多表多次put 1节点3JVM

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomw.partitionmanyput.PartitionmanyPutScript 2000 1 1 1

场景2

场景3

### EP put(多笔数据) 1节点3JVM

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomw.partitionsmallEPput.PartitionSmallEpPutScript 2000 1 1 1

场景2

场景3

### Partition多表多次get 1节点3JVM

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomr.partitionmanyget.PartitionmanyGetScript 2000 1 1 1

场景2

场景3

### EP get(多笔数据) 1节点3JVM

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomr.partitionEPget.PartitionEpGetScript 2000 1 1 1

场景2

场景3

### Partition多表多次put 2节点2JVM

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomw.partitionmanyput.PartitionmanyPutScript 2000 1 1 1

场景2

场景3

### EP put(多笔数据) 2节点2JVM

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomw.partitionsmallEPput.PartitionSmallEpPutScript 2000 1 1 1

场景2

场景3

### Partition多表多次get 2节点2JVM

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomr.partitionmanyget.PartitionmanyGetScript 2000 1 1 1

场景2

场景3

### EP get(多笔数据) 2节点2JVM

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomr.partitionEPget.PartitionEpGetScript 2000 1 1 1

场景2

场景3

### Partition多表多次put 3节点3JVM

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomw.partitionmanyput.PartitionmanyPutScript 2000 1 1 1

场景2

场景3

### EP put(多笔数据) 3节点3JVM

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomw.partitionsmallEPput.PartitionSmallEpPutScript 2000 1 1 1

场景2

场景3

### Partition多表多次get 3节点3JVM

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomr.partitionmanyget.PartitionmanyGetScript 2000 1 1 1

场景2

场景3

### EP get(多笔数据) 3节点3JVM

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomr.partitionEPget.PartitionEpGetScript 2000 1 1 1

场景2

场景3

## 稳定性

### get/put& ep get/put 500W数据量 6JVM

此场景 只有前三个参数有用，会稳定跑满8小时,测试时手动干掉

场景1

nohup java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.stable.PartitionStableScript 2000 2 5 1 >> 1.out &

场景2

场景3

数据一致性 使用导出CSV

### get/put& ep get/put 1000W数据量 6JVM

此场景 只有前三个参数有用，会稳定跑满8小时,测试时手动干掉

场景1

nohup java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.stable.PartitionStableScript 2000 2 5 1 >> 1.out &

场景2

场景3

## cachestore写入方式

### Partition get&put 非后写 1JVM

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.cachestore.CacheStoreNoWBPutScript 2000 1 1 1

场景2

场景3

### Partition get&put 后写 1JVM

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.cachestore.CacheStoreWBPutScript 2000 1 1 1

场景2

场景3

## cachestore数据载入 1JVM

### 不持久化 数据库串行

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.cachestore.load.CacheStoreSerialLoadScript

### 不持久化 数据库并行

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.cachestore.load.CacheStoreParallelLoadScript

### 持久化 数据库串行

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.cachestore.load.CacheStoreSerialLoadScript

### 持久化 数据库并行

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.cachestore.load.CacheStoreParallelLoadScript

## Failover

### Get同时增加节点 1M流量再平衡

测试步骤

1.清空所有表

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.DestoryScript

2.准备测试数据

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.stream.PartitionStreamPutScript 1000000 1000 1 1

3.启动监控程序

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.EventRebalanceScript

4.增加一台节点的同时启动get操作

Get操作命令

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.level1.PartitionLevel1GetScript 2000 1 1 1 1

5.看下监控程序的耗时

### Get同时减少节点 1M流量再平衡

测试步骤

1.清空所有表

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.DestoryScript

2.准备测试数据

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.stream.PartitionStreamPutScript 1000000 1000 1 1

3.启动监控程序

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.EventRebalanceScript

4.减少一台节点的同时启动get操作

Get操作命令

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.level1.PartitionLevel1GetScript 2000 1 1 1

5.看下监控程序的耗时

### Put同时增加节点 1M流量再平衡

测试步骤

1.清空所有表

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.DestoryScript

2.准备测试数据

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.stream.PartitionStreamPutScript 1000000 1000 1 1

3.启动监控程序

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.EventRebalanceScript

4.增加一台节点的同时启动put操作

put操作命令

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.level1.PartitionLevel1PutScript 2000 1 1 1

5.看下监控程序的耗时

### Put同时减少节点 1M流量再平衡

测试步骤

1.清空所有表

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.DestoryScript

2.准备测试数据

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.stream.PartitionStreamPutScript 1000000 1000 1 1

3.启动监控程序

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.EventRebalanceScript

4.减少一台节点的同时启动put操作

Put操作命令

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.level1.PartitionLevel1PutScript 2000 1 1 1

5.看下监控程序的耗时

### Put stream同时增加节点 1M流量再平衡

测试步骤

1.清空所有表

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.DestoryScript

2.准备测试数据

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.stream.PartitionStreamPutScript 1000000 1000 1 1

3.启动监控程序

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.EventRebalanceScript

4.增加一台节点的同时启动put操作

Put stream操作命令

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.level1.PartitionLevel1PutStreamScript 2000 1 1 1

5.看下监控程序的耗时

### Put stream同时减少节点 1M流量再平衡

测试步骤

1.清空所有表

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.DestoryScript

2.准备测试数据

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.stream.PartitionStreamPutScript 1000000 1000 1 1

3.启动监控程序

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.EventRebalanceScript

4.减少一台节点的同时启动put操作

Put stream操作命令

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.level1.PartitionLevel1PutStreamScript 2000 1 1 1

5.看下监控程序的耗时

### EP get同时增加节点 1M流量再平衡

测试步骤

1.清空所有表

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.DestoryScript

2.准备测试数据

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.stream.PartitionStreamPutScript 1000000 1000 1 1

3.启动监控程序

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.EventRebalanceScript

4.增加一台节点的同时启动EP get操作

EP get操作命令

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.level1.PartitionLevel1EPGetScript 2000 1 1 1

5.看下监控程序的耗时

### EP get同时减少节点 1M流量再平衡

测试步骤

1.清空所有表

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.DestoryScript

2.准备测试数据

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.stream.PartitionStreamPutScript 1000000 1000 1 1

3.启动监控程序

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.EventRebalanceScript

4.减少一台节点的同时启动EP get操作

EP get操作命令

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.level1.PartitionLevel1EPGetScript 2000 1 1 1

5.看下监控程序的耗时

### EP put同时增加节点 1M流量再平衡

测试步骤

1.清空所有表

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.DestoryScript

2.准备测试数据

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.stream.PartitionStreamPutScript 1000000 1000 1 1

3.启动监控程序

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.EventRebalanceScript

4.增加一台节点的同时启动EP put操作

EP put 操作命令

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.level1.PartitionLevel1EPPutScript 2000 1 1 1

5.看下监控程序的耗时

### EP put同时减少节点 1M流量再平衡

测试步骤

1.清空所有表

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.DestoryScript

2.准备测试数据

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.stream.PartitionStreamPutScript 1000000 1000 1 1

3.启动监控程序

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.EventRebalanceScript

4.减少一台节点的同时启动EP put操作

EP put 操作命令

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.level1.PartitionLevel1EPPutScript 2000 1 1 1

5.看下监控程序的耗时

### Get同时增加节点 5M流量再平衡

测试步骤

1.清空所有表

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.DestoryScript

2.准备测试数据

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.stream.PartitionStreamPutScript 1000000 1000 1 1

3.启动监控程序

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.EventRebalanceScript

4.增加一台节点的同时启动get操作

Get操作命令

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.level2.PartitionLevel2GetScript 2000 1 1 1

5.看下监控程序的耗时

### Get同时减少节点 5M流量再平衡

测试步骤

1.清空所有表

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.DestoryScript

2.准备测试数据

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.stream.PartitionStreamPutScript 1000000 1000 1 1

3.启动监控程序

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.EventRebalanceScript

4.减少一台节点的同时启动get操作

Get操作命令

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.level2.PartitionLevel2GetScript 2000 1 1 1

5.看下监控程序的耗时

### Put同时增加节点 5M流量再平衡

测试步骤

1.清空所有表

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.DestoryScript

2.准备测试数据

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.stream.PartitionStreamPutScript 1000000 1000 1 1

3.启动监控程序

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.EventRebalanceScript

4.增加一台节点的同时启动put操作

put操作命令

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.level2.PartitionLevel2PutScript 2000 1 1 1

5.看下监控程序的耗时

### Put同时减少节点 5M流量再平衡

测试步骤

1.清空所有表

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.DestoryScript

2.准备测试数据

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.stream.PartitionStreamPutScript 1000000 1000 1 1

3.启动监控程序

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.EventRebalanceScript

4.减少一台节点的同时启动put操作

Put操作命令

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.level2.PartitionLevel2PutScript 2000 1 1 1

5.看下监控程序的耗时

### Put stream同时增加节点 5M流量再平衡

测试步骤

1.清空所有表

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.DestoryScript

2.准备测试数据

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.stream.PartitionStreamPutScript 1000000 1000 1 1

3.启动监控程序

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.EventRebalanceScript

4.增加一台节点的同时启动put操作

Put stream操作命令

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.level2.PartitionLevel2PutStreamScript 2000 1 1 1

5.看下监控程序的耗时

### Put stream同时减少节点 5M流量再平衡

测试步骤

1.清空所有表

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.DestoryScript

2.准备测试数据

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.stream.PartitionStreamPutScript 1000000 1000 1 1

3.启动监控程序

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.EventRebalanceScript

4.减少一台节点的同时启动put操作

Put stream操作命令

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.level2.PartitionLevel2PutStreamScript 2000 1 1 1

5.看下监控程序的耗时

### EP get同时增加节点 5M流量再平衡

测试步骤

1.清空所有表

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.DestoryScript

2.准备测试数据

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.stream.PartitionStreamPutScript 1000000 1000 1 1

3.启动监控程序

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.EventRebalanceScript

4.增加一台节点的同时启动EP get操作

EP get操作命令

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.level2.PartitionLevel2EPGetScript 2000 1 1 1

5.看下监控程序的耗时

### EP get同时减少节点 5M流量再平衡

测试步骤

1.清空所有表

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.DestoryScript

2.准备测试数据

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.stream.PartitionStreamPutScript 1000000 1000 1 1

3.启动监控程序

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.EventRebalanceScript

4.减少一台节点的同时启动EP get操作

EP get操作命令

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.level2.PartitionLevel2EPGetScript 2000 1 1 1

5.看下监控程序的耗时

### EP put同时增加节点 5M流量再平衡

测试步骤

1.清空所有表

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.DestoryScript

2.准备测试数据

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.stream.PartitionStreamPutScript 1000000 1000 1 1

3.启动监控程序

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.EventRebalanceScript

4.增加一台节点的同时启动EP put操作

EP put 操作命令

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.level2.PartitionLevel2EPPutScript 2000 1 1 1

5.看下监控程序的耗时

### EP put同时减少节点 5M流量再平衡

测试步骤

1.清空所有表

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.DestoryScript

2.准备测试数据

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.stream.PartitionStreamPutScript 1000000 1000 1 1

3.启动监控程序

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.EventRebalanceScript

4.减少一台节点的同时启动EP put操作

EP put 操作命令

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.level2.PartitionLevel2EPPutScript 2000 1 1 1

5.看下监控程序的耗时

### Get同时增加节点 10M流量再平衡

测试步骤

1.清空所有表

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.DestoryScript

2.准备测试数据

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.stream.PartitionStreamPutScript 1000000 1000 1 1

3.启动监控程序

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.EventRebalanceScript

4.增加一台节点的同时启动get操作

Get操作命令

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.level3.PartitionLevel3GetScript 2000 1 1 1

5.看下监控程序的耗时

### Get同时减少节点 10M流量再平衡

测试步骤

1.清空所有表

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.DestoryScript

2.准备测试数据

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.stream.PartitionStreamPutScript 1000000 1000 1 1

3.启动监控程序

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.EventRebalanceScript

4.减少一台节点的同时启动get操作

Get操作命令

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.level3.PartitionLevel3GetScript 2000 1 1 1

5.看下监控程序的耗时

### Put同时增加节点 10M流量再平衡

测试步骤

1.清空所有表

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.DestoryScript

2.准备测试数据

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.stream.PartitionStreamPutScript 1000000 1000 1 1

3.启动监控程序

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.EventRebalanceScript

4.增加一台节点的同时启动put操作

put操作命令

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.level3.PartitionLevel3PutScript 2000 1 1 1

5.看下监控程序的耗时

### Put同时减少节点 10M流量再平衡

测试步骤

1.清空所有表

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.DestoryScript

2.准备测试数据

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.stream.PartitionStreamPutScript 1000000 1000 1 1

3.启动监控程序

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.EventRebalanceScript

4.减少一台节点的同时启动put操作

Put操作命令

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.level3.PartitionLevel3PutScript 2000 1 1 1

5.看下监控程序的耗时

### Put stream同时增加节点 10M流量再平衡

测试步骤

1.清空所有表

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.DestoryScript

2.准备测试数据

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.stream.PartitionStreamPutScript 1000000 1000 1 1

3.启动监控程序

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.EventRebalanceScript

4.增加一台节点的同时启动put操作

Put stream操作命令

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.level3.PartitionLevel3PutStreamScript 2000 1 1 1

5.看下监控程序的耗时

### Put stream同时减少节点 10M流量再平衡

测试步骤

1.清空所有表

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.DestoryScript

2.准备测试数据

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.stream.PartitionStreamPutScript 1000000 1000 1 1

3.启动监控程序

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.EventRebalanceScript

4.减少一台节点的同时启动put操作

Put stream操作命令

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.level3.PartitionLevel3PutStreamScript 2000 1 1 1

5.看下监控程序的耗时

### EP get同时增加节点 10M流量再平衡

测试步骤

1.清空所有表

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.DestoryScript

2.准备测试数据

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.stream.PartitionStreamPutScript 1000000 1000 1 1

3.启动监控程序

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.EventRebalanceScript

4.增加一台节点的同时启动EP get操作

EP get操作命令

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.level3.PartitionLevel3EPGetScript 2000 1 1 1

5.看下监控程序的耗时

### EP get同时减少节点 10M流量再平衡

测试步骤

1.清空所有表

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.DestoryScript

2.准备测试数据

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.stream.PartitionStreamPutScript 1000000 1000 1 1

3.启动监控程序

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.EventRebalanceScript

4.减少一台节点的同时启动EP get操作

EP get操作命令

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.level3.PartitionLevel3EPGetScript 2000 1 1 1

5.看下监控程序的耗时

### EP put同时增加节点 10M流量再平衡

测试步骤

1.清空所有表

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.DestoryScript

2.准备测试数据

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.stream.PartitionStreamPutScript 1000000 1000 1 1

3.启动监控程序

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.EventRebalanceScript

4.增加一台节点的同时启动EP put操作

EP put 操作命令

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.level3.PartitionLevel3EPPutScript 2000 1 1 1

5.看下监控程序的耗时

### EP put同时减少节点 10M流量再平衡

测试步骤

1.清空所有表

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.features.DestoryScript

2.准备测试数据

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.stream.PartitionStreamPutScript 1000000 1000 1 1

3.启动监控程序

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.EventRebalanceScript

4.减少一台节点的同时启动EP put操作

EP put 操作命令

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.failover.level3.PartitionLevel3EPPutScript 2000 1 1 1

5.看下监控程序的耗时

## 索引随机读

### 导入1000万 非测试项

先导入数据：不用并发循环，直接导入

for i in {1..10} ; do echo "nohup java -Xms4G -Xmx4g -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.index.IndexPutScript 5000 5 1 20 "$i" > $i.out &" | bash ;done

for i in {11..20} ; do echo "nohup java -Xms4G -Xmx4g -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.index.IndexPutScript 5000 5 1 20 "$i" > $i.out &" | bash ;done

### get方式

1并发

for i in {1..1} ; do echo "nohup java -Xms4G -Xmx4g -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.index.IndexGetScript 500000 1 1 20 "$i" > $i.out &" | bash ;done

100并发

for i in {1..10} ; do echo "nohup java -Xms4G -Xmx4g -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.index.IndexGetScript 5000 5 1 20 "$i" > $i.out &" | bash ;done

for i in {11..20} ; do echo "nohup java -Xms4G -Xmx4g -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.index.IndexGetScript 5000 5 1 20 "$i" > $i.out &" | bash ;done

200并发

for i in {1..10} ; do echo "nohup java -Xms4G -Xmx4g -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.index.IndexGetScript 2500 5 1 20 "$i" > $i.out &" | bash ;done

for i in {11..20} ; do echo "nohup java -Xms4G -Xmx4g -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.index.IndexGetScript 2500 5 1 20 "$i" > $i.out &" | bash ;done

for i in {21..30} ; do echo "nohup java -Xms4G -Xmx4g -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.index.IndexGetScript 2500 5 1 20 "$i" > $i.out &" | bash ;done

for i in {31..40} ; do echo "nohup java -Xms4G -Xmx4g -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.index.IndexGetScript 2500 5 1 20 "$i" > $i.out &" | bash ;done

### SQL PK 1K

1并发

for i in {1..1} ; do echo "nohup java -Xms4G -Xmx4g -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.index.IndexGetPrimaryIndexScript 500000 1 1 20 "$i" > $i.out &" | bash ;done

100并发

for i in {1..10} ; do echo "nohup java -Xms4G -Xmx4g -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.index.IndexGetPrimaryIndexScript 5000 5 1 20 "$i" > $i.out &" | bash ;done

for i in {11..20} ; do echo "nohup java -Xms4G -Xmx4g -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.index.IndexGetPrimaryIndexScript 5000 5 1 20 "$i" > $i.out &" | bash ;done

200并发

for i in {1..10} ; do echo "nohup java -Xms4G -Xmx4g -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.index.IndexGetPrimaryIndexScript 2500 5 1 20 "$i" > $i.out &" | bash ;done

for i in {11..20} ; do echo "nohup java -Xms4G -Xmx4g -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.index.IndexGetPrimaryIndexScript 2500 5 1 20 "$i" > $i.out &" | bash ;done

for i in {21..30} ; do echo "nohup java -Xms4G -Xmx4g -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.index.IndexGetPrimaryIndexScript 2500 5 1 20 "$i" > $i.out &" | bash ;done

for i in {31..40} ; do echo "nohup java -Xms4G -Xmx4g -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.index.IndexGetPrimaryIndexScript 2500 5 1 20 "$i" > $i.out &" | bash ;done

### SQL first index 1K

1并发

for i in {1..1} ; do echo "nohup java -Xms4G -Xmx4g -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.index.IndexGetFirstIndexScript 500000 1 1 20 "$i" > $i.out &" | bash ;done

100并发

for i in {1..10} ; do echo "nohup java -Xms4G -Xmx4g -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.index.IndexGetFirstIndexScript 5000 5 1 20 "$i" > $i.out &" | bash ;done

for i in {11..20} ; do echo "nohup java -Xms4G -Xmx4g -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.index.IndexGetFirstIndexScript 5000 5 1 20 "$i" > $i.out &" | bash ;done

200并发

for i in {1..10} ; do echo "nohup java -Xms4G -Xmx4g -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.index.IndexGetFirstIndexScript 2500 5 1 20 "$i" > $i.out &" | bash ;done

for i in {11..20} ; do echo "nohup java -Xms4G -Xmx4g -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.index.IndexGetFirstIndexScript 2500 5 1 20 "$i" > $i.out &" | bash ;done

for i in {21..30} ; do echo "nohup java -Xms4G -Xmx4g -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.index.IndexGetFirstIndexScript 2500 5 1 20 "$i" > $i.out &" | bash ;done

for i in {31..40} ; do echo "nohup java -Xms4G -Xmx4g -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.index.IndexGetFirstIndexScript 2500 5 1 20 "$i" > $i.out &" | bash ;done

### SQL second index 1K

1并发

for i in {1..1} ; do echo "nohup java -Xms4G -Xmx4g -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.index.IndexGetSecondIndexScript 500000 1 1 20 "$i" > $i.out &" | bash ;done

100并发

for i in {1..10} ; do echo "nohup java -Xms4G -Xmx4g -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.index.IndexGetSecondIndexScript 5000 5 1 20 "$i" > $i.out &" | bash ;done

for i in {11..20} ; do echo "nohup java -Xms4G -Xmx4g -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.index.IndexGetSecondIndexScript 5000 5 1 20 "$i" > $i.out &" | bash ;done

200并发

for i in {1..10} ; do echo "nohup java -Xms4G -Xmx4g -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.index.IndexGetSecondIndexScript 2500 5 1 20 "$i" > $i.out &" | bash ;done

for i in {11..20} ; do echo "nohup java -Xms4G -Xmx4g -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.index.IndexGetSecondIndexScript 2500 5 1 20 "$i" > $i.out &" | bash ;done

for i in {21..30} ; do echo "nohup java -Xms4G -Xmx4g -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.index.IndexGetSecondIndexScript 2500 5 1 20 "$i" > $i.out &" | bash ;done

for i in {31..40} ; do echo "nohup java -Xms4G -Xmx4g -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.index.IndexGetSecondIndexScript 2500 5 1 20 "$i" > $i.out &" | bash ;done

## 关联表

### 并置小对象导入100万 非测试项

先导入数据：不用并发循环，直接导入

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.affinity.AffinityStreamPutSmallScript 2000 1 1 1

### 并置小对象查询

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.affinity.AffinityGetScript 2000 1 1 1

场景2

场景3

### 非并置小对象查询

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.affinity.AffinityGetNoScript 2000 1 1 1

场景2

场景3

### 并置大对象导入 非测试项

先导入数据：不用并发循环，直接导入

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.affinity.AffinityStreamBigSmallScript 2000 1 1 1

### 并置大对象查询

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.affinity.AffinityGetScript 2000 1 1 1

场景2

场景3

### 非并置大对象查询

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.affinity.AffinityGetNoScript 2000 1 1 1

场景2

场景3

## 区间计算

### 小对象导入100万数据量 非测试项仅做数据准备

先导入数据：不用并发循环，直接导入

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.affinity.AffinityStreamPutSmallScript 10000 1 100 1000

### 小对象非索引查询

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.rangecalc.RangeCalcNoIndexScript 2000 1 1 1

场景2

场景3

### 小对象索引查询

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.rangecalc.RangeCalcIndexScript 2000 1 1 1

场景2

场景3

### 小对象主键索引查询

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.rangecalc.RangeCalcKeyIndexScript 2000 1 1 1

场景2

场景3

### 小对象二级索引查询

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.rangecalc.RangeCalcSecIndexScript 2000 1 1 1

场景2

场景3

### 大对象导入100万数据量非测试项仅做数据准备

先导入数据：不用并发循环，直接导入

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.affinity.AffinityStreamBigSmallScript 2000 1 1 1

### 大对象非索引查询

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.rangecalc.RangeCalcNoIndexScript 2000 1 1 1

场景2

场景3

### 大对象索引查询

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.rangecalc.RangeCalcIndexScript 2000 1 1 1

场景2

场景3

### 大对象主键索引查询

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.rangecalc.RangeCalcKeyIndexScript 2000 1 1 1

场景2

场景3

## 模糊查询

### 小对象导入 非测试项仅做数据准备

先导入数据：不用并发循环，直接导入

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.affinity.AffinityStreamPutSmallScript 2000 1 1 1

### 小对象模糊查询

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.likequery.LikeQueryScript 2000 1 1 1

场景2

场景3

### 大对象导入 非测试项仅做数据准备

先导入数据：不用并发循环，直接导入

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.affinity.AffinityStreamBigSmallScript 2000 1 1 1

### 大对象模糊查询

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.likequery.LikeQueryScript 2000 1 1 1

场景2

场景3

## 持久化

### SSD盘写

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomw.partitionsmallput.PartitionSmallPutScript 2000 1 1 1

场景2

场景3

### SSD盘读

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomr.partitionsmallget.PartitionSmallGetScript 2000 1 1 1

场景2

场景3

### NAS盘写

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomw.partitionsmallput.PartitionSmallPutScript 2000 1 1 1

场景2

场景3

### NAS盘读

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomr.partitionsmallget.PartitionSmallGetScript 2000 1 1 1

场景2

场景3

### 本地盘写

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomw.partitionsmallput.PartitionSmallPutScript 2000 1 1 1

场景2

场景3

### 本地盘读

场景1

java -Dfile.encoding=UTF-8 -cp ignite.jar com.newland.boss.script.performance.randomr.partitionsmallget.PartitionSmallGetScript 2000 1 1 1

场景2

场景3

## 持久化恢复1节点1JVM

使用数据盘不同，其他步骤都一样。

1. 清除所有缓存数据
2. 预加载 10000000数据

java -Dfile.encoding=UTF-8 -Xms4G -Xmx4g -cp ignite.jar com.newland.boss.script.performance.stream.PartitionStreamPutScript 1000000 1 1000 1 1

1. 修改配置文件

<!--网络超时时长 默认5000-->

<property name="networkTimeout" value="60000"/>

网络超时时间同级目录下 增加启动事件配置

<property name="lifecycleBeans">

<list>

<bean class="com.newland.boss.script.performance.loaddata.LoadData"/>

</list>

</property>

1. 启动节点打开日志，日志中会有消耗时间。