

项目性能测试报告

笔记本: homework

创建时间: 2022/8/28 23:04

作者: 职业失踪人员

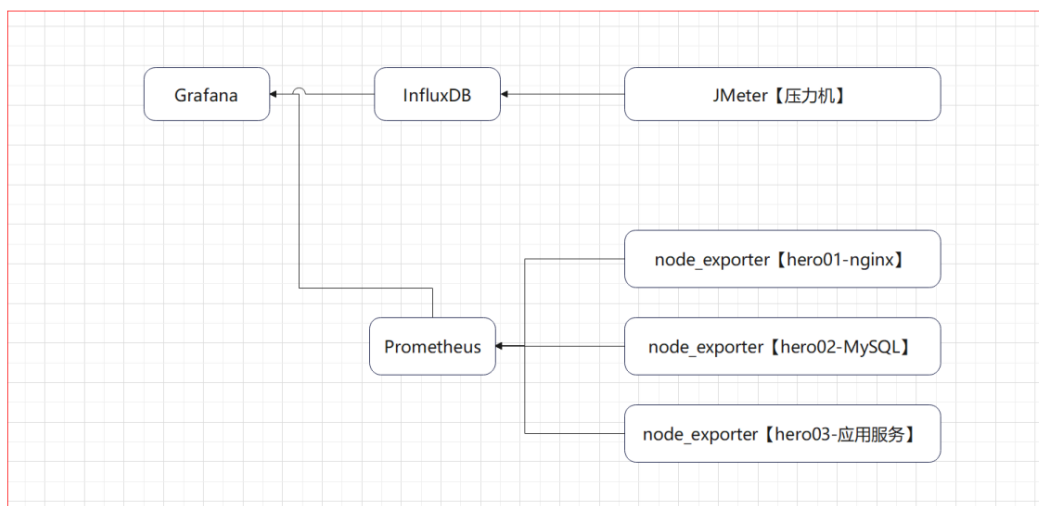
更新时间: 2022/8/28 23:48

项目性能测试报告

01-测试目的

通过搭建一套压力测试平台，对 hero_mall 项目的性能负载和容量有个准确的认知，以便更为直观的了解性能优化的各项参数，能够正确的评价一套系统在性能方面的优劣。通过压力测试结果分析存在的性能问题，为后续性能优化提供有利的解决方案。

02-测试工具



03-测试环境

3.1 环境

指标	参数
机器	4C8G
集群规模	单机
hero_mall_one版本	1.0
数据库	MySQL

3.2 设置启动参数

```
export JAVA_HOME

export JRE_HOME=${JAVA_HOME}/jre

export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib

export SERVER_NAME="hero_web"

export JAVA="${JAVA_HOME}/bin/java"

export BASE_DIR=`cd $(dirname $0)/. ; pwd`

export

DEFAULT_SEARCH_LOCATIONS="classpath:/,classpath:/config/,file:./,file:./conf

ig/"

export

CUSTOM_SEARCH_LOCATIONS=${DEFAULT_SEARCH_LOCATIONS},file:${BASE_DIR}/conf/

JAVA_OPT="${JAVA_OPT} -server -Xms512m -Xmx512m -Xmn256 -

XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=320m"

JAVA_OPT="${JAVA_OPT} -XX:-OmitStackTraceInFastThrow -

XX:+HeapDumpOnOutOfMemoryError -

XX:HeapDumpPath=${BASE_DIR}/logs/java_heapdump.hprof"

JAVA_OPT="${JAVA_OPT} -XX:-UseLargePages"
```

```

JAVA_OPT="${JAVA_OPT} -jar ${BASE_DIR}/${SERVER_NAME}*.jar"

JAVA_OPT="${JAVA_OPT} ${JAVA_OPT_EXT}"

JAVA_OPT="${JAVA_OPT} --spring.config.location=${CUSTOM_SEARCH_LOCATIONS}"

if [ ! -d "${BASE_DIR}/logs" ]; then

mkdir ${BASE_DIR}/logs

fi

echo "JAVA ${JAVA_OPT}"

if [ ! -f "${BASE_DIR}/logs/${SERVER_NAME}.out" ]; then

touch "${BASE_DIR}/logs/${SERVER_NAME}.out"

fi

echo "JAVA ${JAVA_OPT}" > ${BASE_DIR}/logs/${SERVER_NAME}.out 2>&1 &

nohup JAVA ${JAVA_OPT} hero_web.hero_web >>

${BASE_DIR}/logs/${SERVER_NAME}.out 2>&1 &

echo "server is starting, you can check the

${BASE_DIR}/logs/${SERVER_NAME}.out"

```

04-测试场景

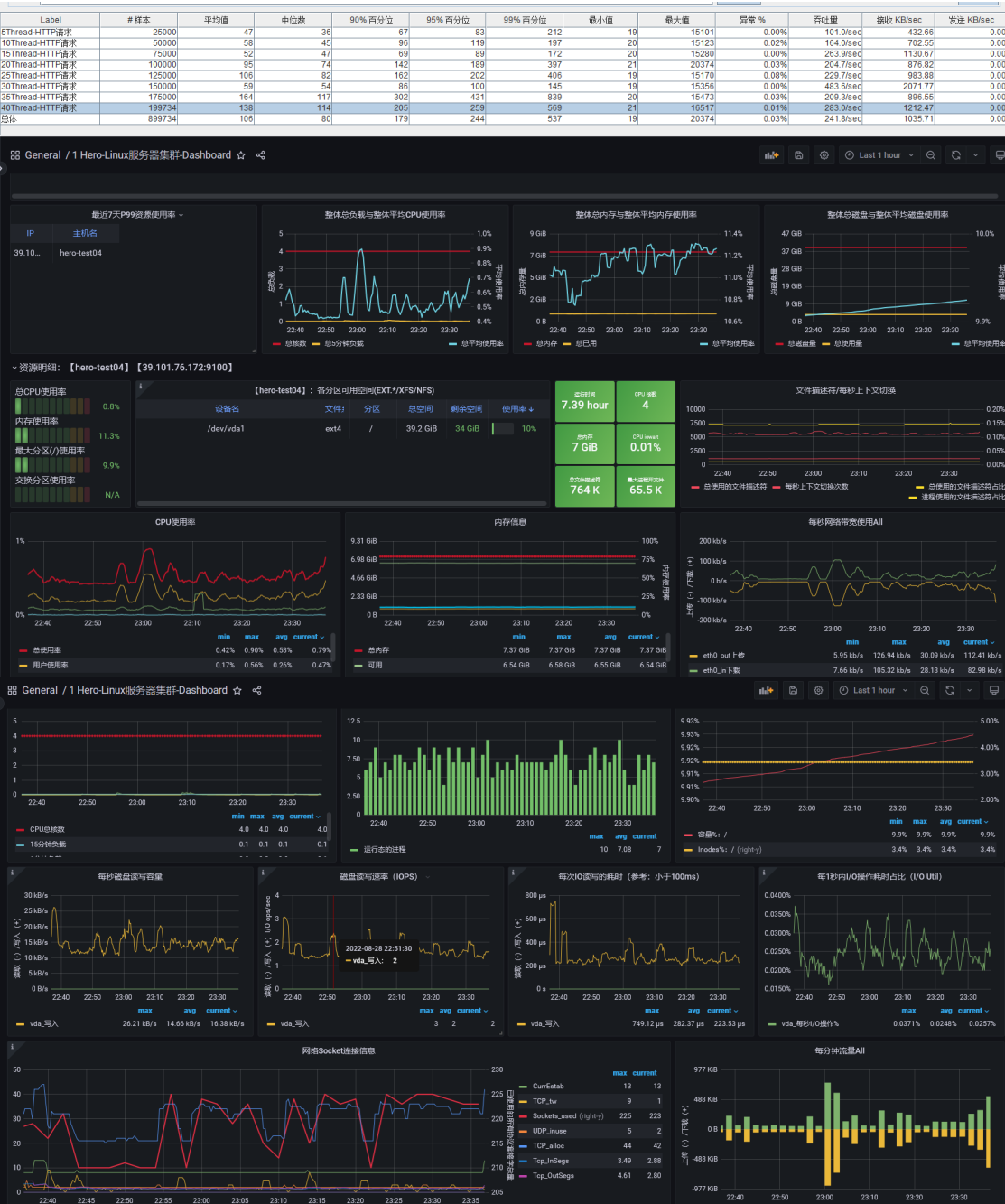
测试场景一般情况下都是最重要接口：验证hero_mall服务获取商品信息接口在不同并发规模的表现

情况01-模拟低延时场景，用户访问接口并发逐渐增加的过程。接口的响应时间为20ms，线程梯度：5、10、15、20、25、30、35、40个线程，5000次；

- 时间设置：Ramp-up period(inseconds)的值设为对应线程数
- 测试总时长：约等于20ms x 5000次 x 8 = 800s = 13分

05-核心接口的测试结果

验证hero_mall服务获取商品信息接口能力性能



06-测试结论

hero_web性能测试是针对重点功能，单机单节点服务进行压测，可以看到各个接口容量。本次测试结果显示，5、10、15、20、25、30、35个线程的测试均正常通过，40个线程的测试在第199734次请求后停滞不前。

