CSDN zeus:

1. 负载均衡管理。

我们目前的系统负载均衡功能没有运行起来，非常影响整个系统的性能。

zeus/zeus2 的负载均衡只依赖于内存的使用率。

具体机制是：zeus会有一个Master （host） 和多个work (host), 多个host一起起来的时候会通过获取distribute\_lock抢占Master位置,抢占成功的话会把自己的host IP写到zeus\_lock表中。

其他的host自动成为worker，并通过心跳消息向Master注册自己。 这样Master就拥有了完整的worker host list.

Master在分配任务的时候从worker host list是找到一个内存使用率低于maxMemRate的host， 把这个任务分配给这个host.

ctrip的版本是用mem+cpu来做负载均衡判断。(需要改进： 选负载最小的woker)

相关代码：Master.java (getRunableWorker)

2.Dos2Unix 没有安装是会导致任务进入队列后不执行的情况。

3. host 分组管理

ctrip版本用两张表zeus\_host\_group zeus\_host\_relation来对所有host（master and workers host）进行分组管理.

一个任务只能在一个host group里面分配，这其实也是对任务进行分组。

每个任务在创建的时候必须指定一个host group ID, 或者使用缺省的host group ID.

由于我们的任务相对比较单一， 可以只使用一个缺省的分组，以简化实现。

需要增加： 自动在数据库表zeus\_host\_group中写一个缺省的host group, 把所有的worker host IP写到zeus\_host\_relation表中。

相关代码： MysqlHostGroupManager.java, Master.java (getRunableWorker)

4.用户管理不完善

ctrip版本增加了用户管理功能。

需要考虑CSDN版本的用户管理和ctrip版本集成的问题

5.性能问题 （需要数据分析师提供输入），个人手动的大任务的处理。

6.稳定性问题 （需要数据分析师提供输入）

7. 有关调度：

定时扫描(每分钟)zeus\_job表，生成zeus\_action表。

自动调度的时候使用zeus\_action表。

8. 有关配置：

在antx.properties 中可以配置邮件告警和SMS告警相关的配置。

#邮件发送的相关配置

email.host=

email.port=

email.username=

email.password=

email.sendFrom=

#Noc报警的相关配置

###测试

noc.notifyUrl=

noc.accessToken=

9. 有关failover:

当master/worker下线之后系统的行为。

Log:

Master/Worker: CSDN\_HDP-ZSS-01: 192.168.103.103

Worker: Failed to connect to Master:

2015-06-24 16:16:41,911 [INFO ] com.taobao.zeus.schedule.mvc.ScheduleInfoLog - worker start connect by DistributeLocker

2015-06-24 16:16:41,912 [ERROR] com.taobao.zeus.schedule.mvc.ScheduleInfoLog - start up worker fail

java.lang.RuntimeException: connect server fail 192.168.103.103

at com.taobao.zeus.socket.worker.ClientWorker.connect(ClientWorker.java:368)

at com.taobao.zeus.schedule.DistributeLocker.update(DistributeLocker.java:159)

at com.taobao.zeus.schedule.DistributeLocker.access$0(DistributeLocker.java:114)

at com.taobao.zeus.schedule.DistributeLocker$1.run(DistributeLocker.java:100)

at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:441)

at java.util.concurrent.FutureTask$Sync.innerRunAndReset(FutureTask.java:317)

at java.util.concurrent.FutureTask.runAndReset(FutureTask.java:150)

at java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.access$101(ScheduledThreadPoolExecutor.java:98)

at java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.runPeriodic(ScheduledThreadPoolExecutor.java:180)

at java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.run(ScheduledThreadPoolExecutor.java:204)

at java.util.concurrent.ThreadPoolExecutor$Worker.runTask(ThreadPoolExecutor.java:886)

at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:908)

at java.lang.Thread.run(Thread.java:662)

Caused by: java.net.ConnectException: Connection refused

at sun.nio.ch.SocketChannelImpl.checkConnect(Native Method)

at sun.nio.ch.SocketChannelImpl.finishConnect(SocketChannelImpl.java:567)

at org.jboss.netty.channel.socket.nio.NioClientSocketPipelineSink$Boss.connect(NioClientSocketPipelineSink.java:396)

at org.jboss.netty.channel.socket.nio.NioClientSocketPipelineSink$Boss.processSelectedKeys(NioClientSocketPipelineSink.java:358)

at org.jboss.netty.channel.socket.nio.NioClientSocketPipelineSink$Boss.run(NioClientSocketPipelineSink.java:274)

zeus2: (May 2014)

1.支持多时区，对不同时区提供支持

2、支持任务指定机器执行（提交）

3、支持MR2,Hive0.12

修改BUG：

1、数据库字段较短导致添加资源文件失败

2、心跳程序占满整个/tmp目录，导致目录下文件数过多，无法打目录

3、修复某个Work下线后，仍然向该Worker下发任务的BUG