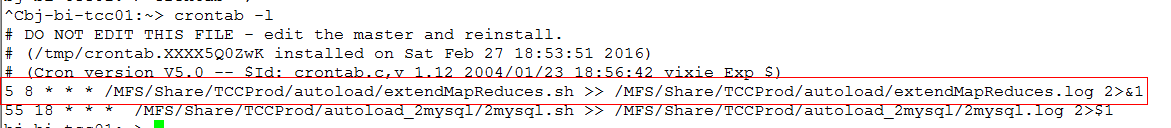
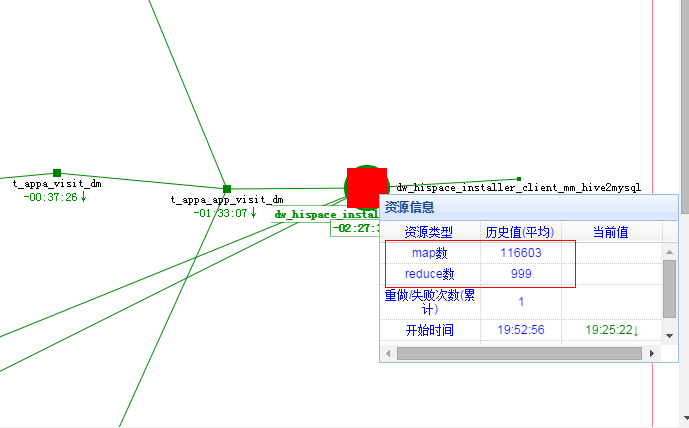
TCC获取任务执行MR资源解读

服务器（[TCCProd@10.41.24.83）上有一个例行程序，每天8点5](mailto:TCCProd@10.41.24.83）上有一个例行程序，每天8点5)分执行，用于获取TCC任务使用的MR资源。



这个程序会去jobtracker服务器（10.41.24.35）获取日志信息，并计算每个任务最近一个月的MR平均值，然后写到TCC的数据库中。TCC的任务依赖图中的资源消耗数据就是这么来的。



例行程序（/MFS/Share/TCCProd/autoload/extendMapReduces.sh）的源码如下：

源码解读

首先，调用PUSH工具，将一天的运行任务从TCC数据库中取出，并写到服务器的tasks目录。从数据库取数据的sql如下（AppConf.xml）：



select '21' as db\_id,f.\*,date\_format(start\_time,'%Y%m%d') as run\_date,unix\_timestamp(f.end\_time) - unix\_timestamp(f.start\_time) as execute\_seconds from

(select b.service\_name,

a.task\_name,

c.cycle\_id as period\_id,

c.state,

min(e.Running\_Start\_Time) as start\_time,

c.Running\_End\_Time as end\_time,

group\_concat(e.Running\_Job\_ID) as job\_Ids,

group\_concat(d.Job\_Input) as file\_names,

a.serviceid as service\_id,

c.task\_id,

a.Multi\_Batch\_Flag as ods\_flag,

c.Return\_Times-1 as redo\_times

from tcc\_task a,

service\_defination b,

tcc\_task\_running\_state c,

tcc\_batch\_running\_state d,

tcc\_step\_running\_state e

where a.task\_id = c.task\_id and

a.serviceid=b.service\_id and

c.task\_id = d.task\_id and

c.cycle\_id = d.cycle\_id and

d.task\_id = e.task\_id and

d.cycle\_id = e.cycle\_id and

d.batch\_id = e.batch\_id and

c.Running\_Start\_Time>=date\_format(DATE\_ADD('${startTime}',interval -1 day),'%Y-%m-%d 08:00:00') and

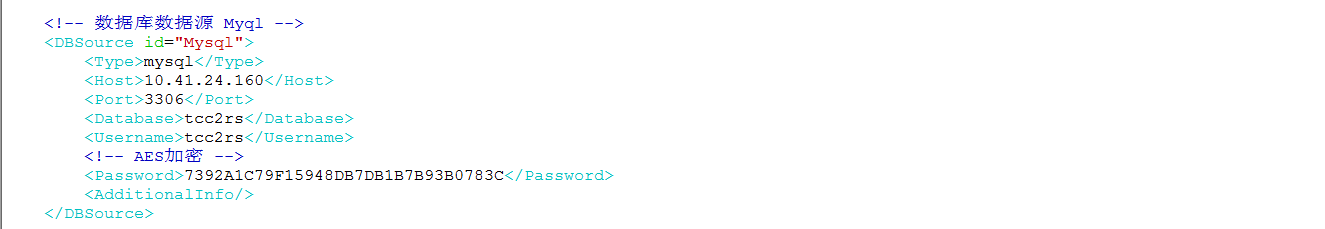
c.Running\_Start\_Time<date\_format('${startTime}','%Y-%m-%d 08:00:00')

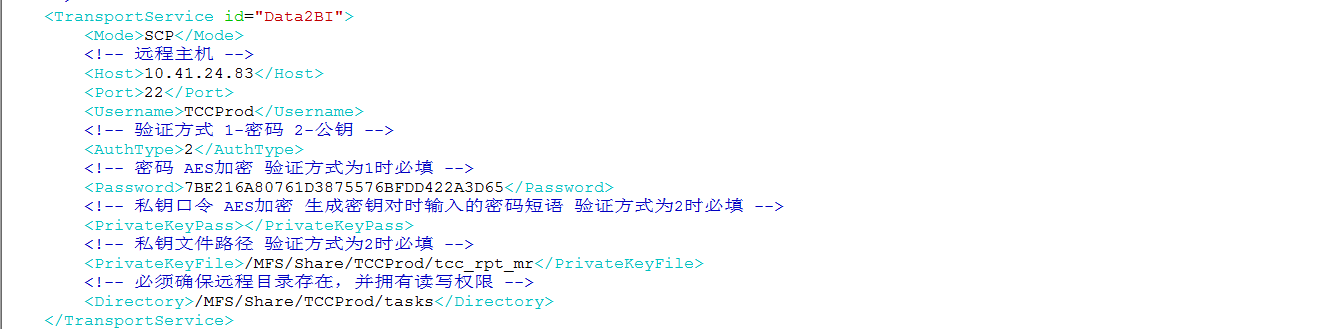
group by c.task\_id,c.cycle\_id

order by start\_time) f

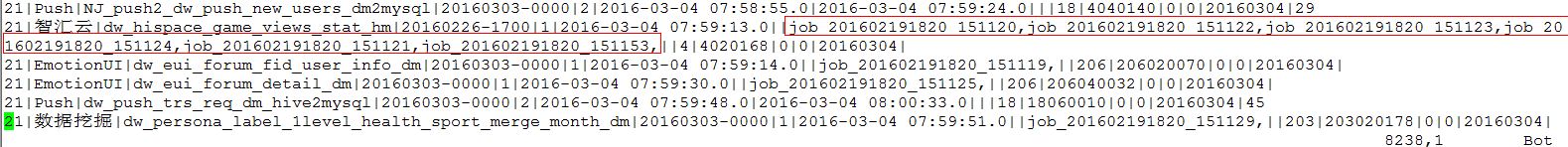
从sql中可以看出，PUSH工具将昨天8:00到今天8：00之间的所有任务信息（包括hadoop的任务ID）导出到文档task\_instance\_xxx.txt文件中。

PUSH工具用到的数据库配置和文件服务器配置在（EnvConfig.xml）：

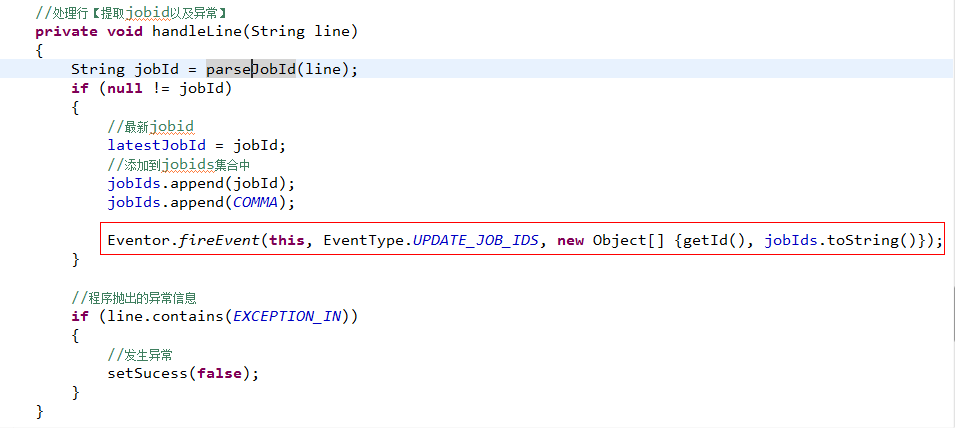




task\_instance\_xxx.txt中的内容如下：

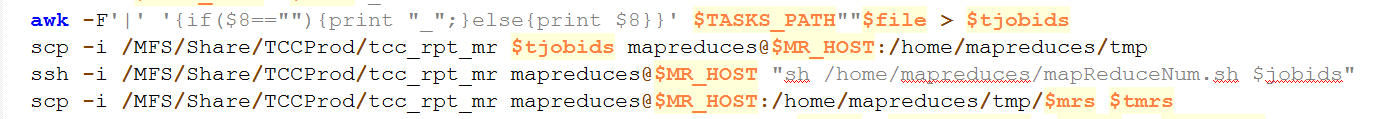


那么这个jobid是怎么得到的呢？其实这个jobid是TCC在调用ssh命令执行脚本时，会记录壳日志，并解析壳日志，将jobid解析出来，更新到tcc\_task\_running\_state表中。



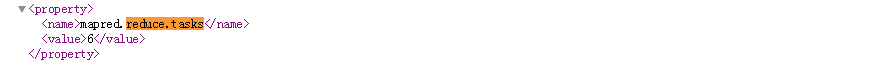
然后，将所有的jobid解析出来，并上传至jobtracker服务器，然后执行jobtracker服务器的脚本mapReduceNum.sh，并将结果复制到本地。





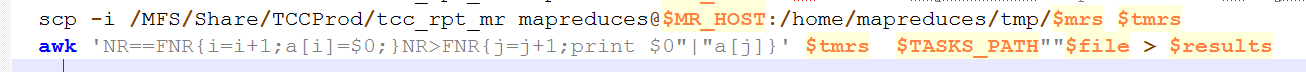
Jobtracker服务器根据每个jobid的日志文件，获取相应的map和reduce数，并写到文件mr文件中。没有jobid的设为0 。

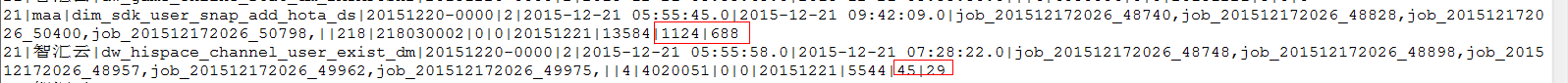




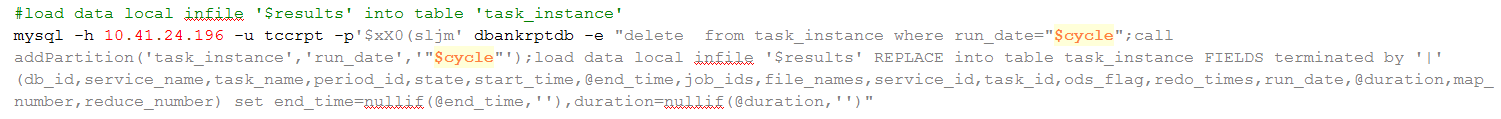


将jobtracker服务器的的mr文件复制到本地，并将map|reduce数量追加到相应的任务中。





将结果文件写到mysql数据库的task\_instance表中。



最后，计算最近一个月所有任务的平均值（调度时长、map数、reduce数、重做次数、平均开始时间、平均结束时间），并写到res\_statistics表中。