服务器备份及恢复guide

1. 本文基于服务器环境如下：







前端

Mysql cluster

Web cluster

前端目前使用物理机，ip: 192.168.5.40

Web group 使用物理机, ip: 192.168.5.240, 192.168.5.241

Mysql group 目前在建。

Web group 使用rsync机制，保证web目录的数据同步。并搭载keepalived虚拟IP对外服务

Mysql group 使用一主双从机制，并搭载MHA+keepalived的高可用及虚拟IP对外服务。

工作模式为访问db.capvision.com，则解析到前端server (192.168.5.40) 前端server 路由到目前的工作web server(192.168.5.241), 数据库连接则选中mysql group的Master server。

1. web cluster宕机分析

1．硬件crash

这种故障导致本机所有服务不可用，虚拟ip自动漂移到backup设备，则backup设备成为master server,代表web cluster对外提供服务。

2．系统crash

这种故障将导致本机某些服务不可用，如果keepalived保持活跃，则通过wget访问db的login界面。

a.如果返回 Connecting to uat.ndb.capvision.com (uat.ndb.capvision.com)|192.168.5.37|:80... failed: Connection refused , 则表明nginx崩溃，即通过keepalived 切换虚拟IP到backup server， 由backup server 继续提供服务。

b.如果返回HTTP request sent, awaiting response... 500 Internal Server Error 则表明mysql数据库连接出错，则每隔5分钟 执行wget 获取返回值，一直到状态正常或者connection refused 为止。Mysql数据库问题交由mysql cluster解决。

注：以上crash的虚拟ip漂移的自动完成都是最终目标，目前的方案是使用前端配置来确定访问哪个web server，因为在Master server上会有计划任务需要停止，以及master , backup之间的rsync服务需要停止（待补充）

1. Mysql cluster宕机分析

1. 硬件crash , 系统crash

Mysql的crash由 MHA维护，MHA有一个管理节点，一旦mysql 的master server 出现不能提供服务（硬件级，系统级），MHA管理工具将会从slave中选出一个做为新的master， 并且把另一个slave挂载到这个新的master上，成为单主单从的结构。配合keepalived的虚拟IP服务，虚拟IP漂移到新的master上，提供数据库虚拟IP接口。

注：以上过程完全自动化完成，整个服务正在完成中，并且需要一段时间的测试。

1. web/mysql cluster 恢复

1.系统恢复

一旦工作稳定，需要恢复损坏的server。

目前我们对web/mysql服务器提供全盘备份，备份命令是：

tar -czvpf backup.tgz -g /home/capvision/backup/snapshot --exclude=/home/capvision/backup --exclude=/proc --exclude=/lost+found --exclude=/sys --exclude=/mnt --exclude=/media --exclude=/dev --exclude=/tmp /

--exclude指令表示这些目录不需要备份。

生成tgz文件群，由一个base的全盘备份文件和几个增量的备份文件。每周一个全盘备份文件，其余执行增量备份。

备份文件通过网络传输，存储到192.168.5.100/sharebackup/NDB/ 下。

如果是硬件崩溃或者系统奔溃，不能短时间通过debug手段恢复，则能启动server之后，重新安装操作系统，版本号为Ubuntu 12.04.3 LTS

而后把tgz文件放到用户/home/capvision下。

scp xxxx.tgz capvision@192.168.5.241/240:/home/cavpsion

然后需要添加用户，我们在DB prod环境上部署了服务有mysql,zabbix,memcache,执行以下命令，添加

sudo groupadd mysql

sudo useradd -g mysql mysql

sudo groupadd zabbix

sudo useradd -g zabbix zabbix

sudo groupadd memcache

sudo useradd -g memcache memcache

添加的用户加入到文件/etc/password中。

然后执行恢复命令，先恢复全盘备份文件，然后再依次恢复增量备份文件。

tar --warning=no-timestamp -xkvpz -f backup.tgz -C /

--warning=no-timestamp 取消关于时间戳的警告，linux有些文件的时间戳比较奇异，一旦解压缩会报错，所以添加这个选项

-p 保留原来的文件属主属性

-k 保留原来存在的文件，这点很重要，存在的文件不要覆盖否则会破坏原来linux的生态环境，所以我们要恢复的linux设备安装的软件越少越好，只安装基本系统和ssh，所有新的软件和配置全部通过tar来恢复出来。

正是使用了-K选项，不覆盖/etc/password文件，这样恢复之后的mysql,zabbix,memcache用户都不存在的，user:group都是对应原来的数字，不能正常工作，所以我们在使用tar恢复之前，需要先添加软件需要的用户和组。

执行上述命令之后，重启，一般都能正常恢复，如果有问题，也只会是mysql有问题，这就需要debug调试恢复下。实在不行，执行sudo apt-get autoremove mysql-server 删除mysql数据库，然后重新安装mysql-server, 数据还是存在的。

注：以后mysql和web app分离，则这里的mysql用户和组的添加以及mysql的恢复就不需要了。

2.数据库恢复

如果需要恢复损坏的数据库，可以使用xtrabackup 工具

我们在数据库服务器上部署了xtrabackup工具，对数据库进行备份，每天一个全量备份，而后每小时执行一次增量备份。

备份文件通过网络传输，存储到192.168.5.100 /sharebackup/DBbackup下。

全盘备份命令：

innobackupex --user=dbuser --password=dbpassword --defaults-file=/etc/mysql/my.cnf backfileDirectory --no-timestamp

增量备份命令：

innobackupex --incremental incrementalDirectory --incremental-basedir=/ incrementalDirectoryPre --user=dbuser --password=dbpassword --no-timestamp

可以看到每一个增量备份都是基于前一个增量备份的内容进行，这样每一次增量都是前一次增量的增量数据。

恢复命令：

Xtrabackup的恢复包括log和data, 需要把每一个增量的log和data都恢复到全量备份的目录中，然后再使用全量备份的内容来恢复整个mysql系统。

首先停掉数据库，然后删除数据库目录下的所有数据库文件.

cd /data/mysql\_data

rm -rf \* # 删除数据目录里的所有文件

而后执行命令：

xtrabackup --prepare --apply-log-only --target-dir=/ BaseMysqlDir

把增量备份恢复到全量备份目录中：

sudo innobackupex --user=dbuser --password=dbpassword --apply-log /BaseMysqlDir \

--incremental-dir=/IndrementalDir1/

sudo innobackupex --user=dbuser --password=dbpassword --copy-back /BaseMysqlDir \

--incremental-dir=/IndrementalDir1/

把全量备份恢复到系统：

sudo innobackupex --user=root --apply-log /BaseMysqlDir

sudo innobackupex --user=root --copy-back /BaseMysqlDir

而后

先切换到root的权限 su root

进入lib的目录 cd /var/lib

将mysql 的目录的用户和组的权限都改为mysql chown -R mysql:mysql mysql

重启mysql服务 sudo /etc/init.d/mysql start

3.文件损坏，误删

使用bacula进行快速恢复。目前还在搭建。

1. 备份性能分析:
2. tar

tar备份是实时打包根目录下所有文件，

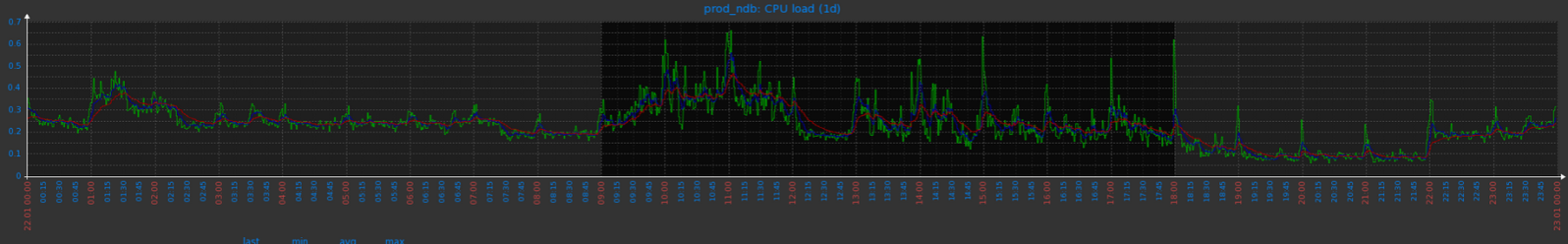
以192.168.5.241上的备份为例，经过测试，这个命令表现为两种action

Tar命令在一开始，会占用比较多的CPU，但不会超过100% ，于此同时mysqld的服务占用CPU一直处于150%-250%之间。

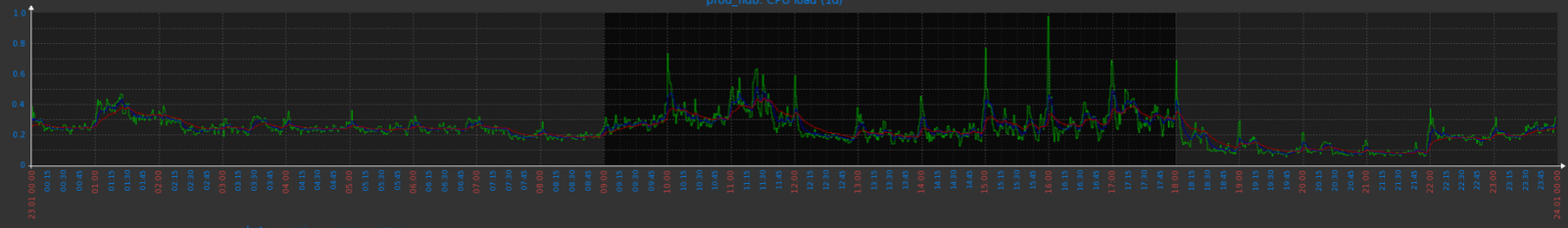
但约1分钟以后, tar占用的CPU时间明显下降，只有3-5%。

以下是cpu load graph on ndb prod:

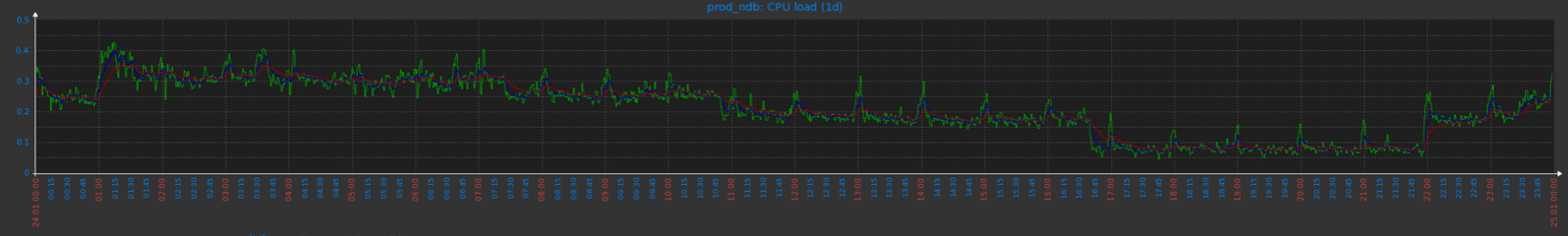
2015-1-22



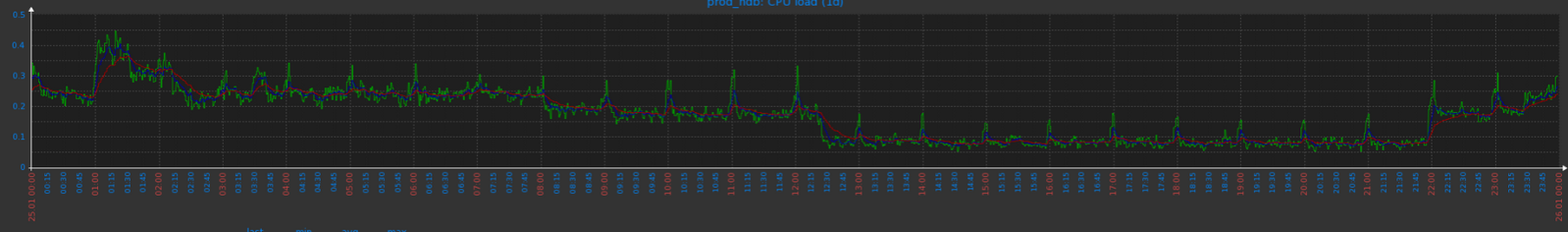
2015-1-23



2015-1-24



2015-1-25



其中,只有2014-1-24日是在凌晨1点执行了全盘备份，其他全部在1点执行了增量备份。

增量备份在凌晨1点有一个高峰之后缓慢下降，全量备份相对来说都处于比较高的峰值。

但都处于可接受的范围之内，cpu load average 基本都在0.4以下。

说明Tar 命令的可用性还是比较高的，cpu占用率比ndb prod上正常工作状态的cpu load average要低。

全盘备份，实际花费时间为11小时

增量备份，实际划费时间为1小时左右（10G文件为1小时左右，20G文件也要9小时，机制待调查）

以上时间包括网络传输

2.xtrabackup

上图还叠加了xtrabackup的任务，图中看到的叠加之后的load average.

全盘备份 花费时间30分钟左右

增量备份 花费时间10分钟左右

以上时间包括网络传输

小结：

所以即使备份任务对正常工作流程产生影响的话，也只可能在数据传输的时候，对网络传输的压力过大，占用带宽高，影响正常工作。

Mysql MHA 数据loss测试

环境：

现在有DB1(67),DB2(68),DB3(70) 以及 MHA manager (72) 以及一个VIP:78

操作：

在72上首先测试MHA的情况：

masterha\_check\_ssh --conf=/etc/app1.cnf

测试ssh 免 key登陆是不是correct。

masterha\_check\_repl --conf=/etc/app1.cnf

测试mha 的设置状态是不是都correct。

之后 启动MHA:

masterha\_manager --conf=/etc/app1.cnf。

在72上再开一个console, 输入测试命令

测试1:

进入mysql 控制台: mysql –uroot –p –h 192.168.5.78

此时，连接的78主机会虚拟连接到DB1(67)

在mysql> 提示符下, 执行source xxxx.sql;

测试数据为1w5 条insert 语句。

在 DB1(67) 上 停掉mysql,

在72的mysql控制台上不断输出 connection lost。

MHA manager 执行failover操作，把candidate manager选为manager。并做为主ip，映射到78。此过程标称是30s

在72上再看，等到slave变为master并提供ip服务之后,insert 语句顺序执行到结束。

结果：slave中间数据丢失一大段，即 connection lost 是以不断的执行insert语句，不断的抛出错误来实现的。

测试2:

在shell命令行下，执行 mysql –uroot –p –e ‘source xxx.sql’;

转到DB1(67)上，执行service mysql stop

则72的shell上执行的mysql 立即connection lost, 并且控制权限交还给shell。

结果：slave 只有前面部分数据，后面数据没有。

对于数据完整性来说，这两个action都是不能接受的。

测试3:

事务

在source 文件里面的insert语句部分 执行事务提交

1. 数据5W条，最后一条出错

执行过程如上，但是期间不断开数据库，对于事务来说，时间是消耗在事务的log上，提交是很快的，我们没有办法模拟在提交的瞬间断开数据库。

结果：master库recover，不可用，但数据库不奔溃，所以mha不切换。

期间要花30分钟以上来recover 回滚的数据库，并且该数据库不可访问。不可接受。

1. 数据1千条，最后一条出错

经过分析，一次写入5w条数据对于业务系统来说基本不可能存在，一次写入1千条数据已经是极端情况了。

结果：基本瞬间就recover 了数据库，切换操作台的手速不能发现该数据库是否发生短暂的不可用。

最终讨论结果，app端使用事务写入表，并且发现事务的recover可能和内存大小有关。