Index:

1.所需要的软件
1-1.系統初始化
1-2.軟件清單
1-3.系统大体架构
2.安装
2-1.安装jemalloc-3.4.0
2-2.安装mysql
2-3.安装jdk
2-4.安装tomcat
2-5.安装redis
2-6.安装nginx
3.配置
3-1.配置mysql
3-2.配置nginx
3-3.配置程序的数据库连接
3-4.Atlas数据库读写分离配置

1.所需要的软件

1-1.系統初始化

在□□安装的系统中,执行安装lrzsz命令并上传初始化脚本、执行。

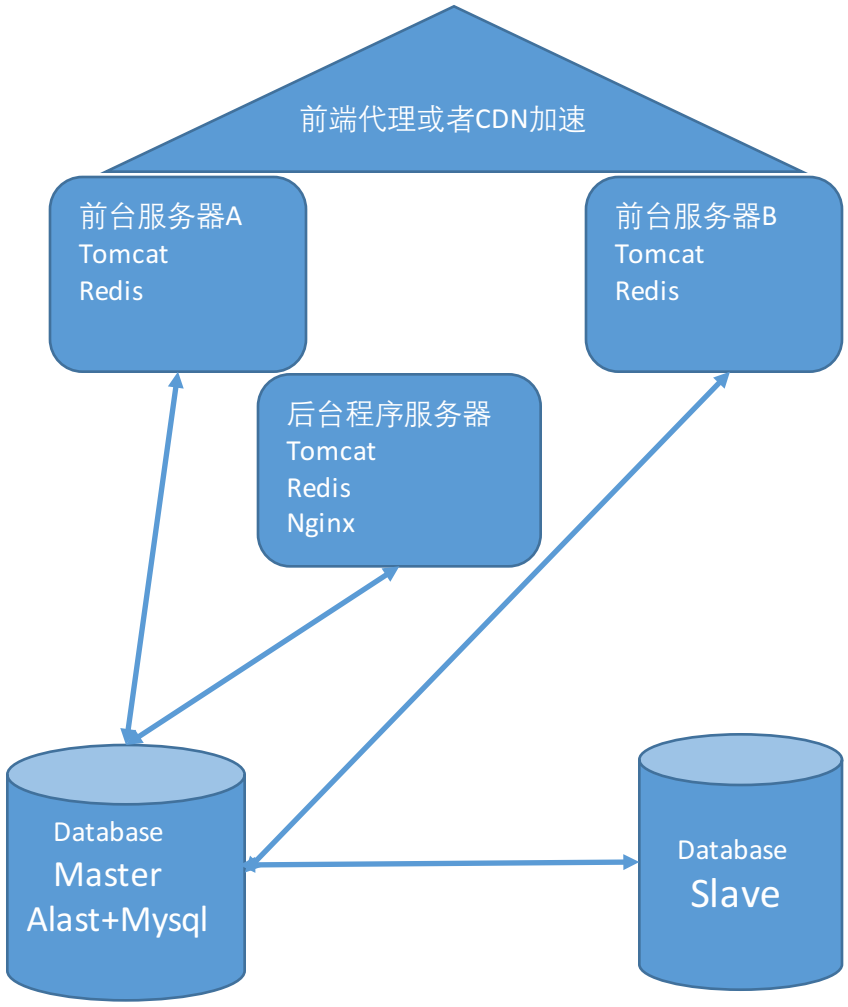
```
#  bash System_initialize.sh
```

注：执行完这个脚本，基本的依赖性软件安装了95%以上，如果在安装过程中还有其他依赖性的软件，请自行安装

1-2.軟件清單

軟件名稱	版本號	功能	備註
mysql	5.1以上	数据库	N/a
jdk	1.7版本	Java开发运行环境	此系统必须使用5.3版本的php,不支持最新版本php的某些函数
nginx	无限制	web功能	N/a
jemalloc	无限制	内存优化	N/a
redis	3.0以上	缓存	本实例中使用3.2.0版本
tomcat	7.0版本	Java解释器	本实例中使用3.0.8版本
rsync	3.0版本	前后台数据同步	本实例中使用3.0.6版本,系统中的默认安装版本
Atlas	2.21版本	数据库读写分离	本实例中使用2.21版本,系统中的默认安装版本

1-3.系统大体架构



注：前端使用Nginx反向代理或者租用其他公司的CDN防护和加速服务

注：前端的tomcat要做集群，使用redis做session的共享

注：后台服务器的Redis服务器主要跑封锁表

注：后台数据库的读写分离,Slave服务器主要提供读的服务,Master服务器主要提供一小部分表的读写和大部分表的写的服务

注：上图为生成环境的最简架构图,生产环境中为了更好的提供服务,读写分离一定要做。还有做数据库归档服务器。

2.安装
2-1.安装jemalloc-3.4.0

下载源码安装包

```
# wget http://www.canonware.com/download/jemalloc/jemalloc-3.4.0.tar.bz2
```

解压安装包

```
# tar  -jxvf  jemalloc-3.4.0.tar.bz2
```

进入解压的安装包

```
# cd  jemalloc-3.4.0
```

编译

```
./configure
```

安装

```
# make && make install
```

添加链接库文件

```
# echo '/usr/local/lib' > /etc/ld.so.conf.d/local.conf
```

加载库文件

```
# ldconfig
```

2-2.安装mysql

下载源码安装包

```
# wget http://mirrors.neusoft.edu.cn/mariadb//mariadb-10.2.13/source/mariadb-10.2.13.tar.gz
```

解压安装包

```
# tar  -zxvf   mariadb-10.2.13.tar.gz
```

进入解压的安装包

```
# cd  mariadb-10.2.13
```

编译

```
#  cmake -DCMAKE_INSTALL_PREFIX=/usr/local/webserver/mysql -DMYSQL_DATADIR=/data/mysql -DDEFAULT_CHARSET=utf8
-DDEFAULT_COLLATION=utf8_general_ci -DMYSQL_UNIX_ADDR=/tmp/mysql.sock  -DWITH_MYISAM_STORAGE_ENGINE=1
 -DENABLE_DOWNLOADS=1 -DCMAKE_EXE_LINKER_FLAGS="-ljemalloc" -DWITH_SAFEMALLOC=OFF
```

安装

```
# make
```

```
# make install
```

2-3.安装jdk

下载jdk1.7源码安装包

```
# wget http://download.oracle.com/otn/java/jdk/7u80-b15/jdk-7u80-linux-x64.tar.gz
```

移动压缩包到特定路径并解压安装包

```
#  mv  jdk-7u80-linux-x64.tar.gz  /usr/local/
```

```
# tar  -jxvf  jdk-7u80-linux-x64.tar.gz
```

重命名jdk目录并设置jdk的环境变量

```
#  mv   jdk-7u80   jdk1.7
```

配置环境变量

```
# vim  /root/.bash_profile
```

Befor:

```
# .bash_profile
# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi
# User specific environment and startup programs
PATH=$PATH:$HOME/bin
export PATH
```

Affter:

```
# .bash_profile
# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi
# User specific environment and startup programs
PATH=$PATH:$HOME/bin:/usr/local/jdk1.7/bin:/usr/local/webserver/mysql/bin
export PATH
```

使用如下命令确认jdk是否安装成功

```
#.  java -version
```

显示如下：

```
java version "1.7.0_45"
Java™ SE Runtime Environment (rhel-2.4.3.3.el6-x86_64 u45-b15)
Java HotSpot(TM) 64-Bit Server VM (build 24.45-b08, mixed mode)
```

注：显示如上内容，表示jdk环境安装完毕

2-4.安装tomcat

下载tomcat

```
wget   http://www-us.apache.org/dist/tomcat/tomcat-7/v7.0.85/bin/apache-tomcat-7.0.85.tar.gz
```

移动到/usr/local/webserver/

```
# mv  apache-tomcat-7.0.85.tar.gz   /usr/local/webserver/
```

解压apache-tomcat-7.0.85.tar.gz

```
# tar -zxvf  apache-tomcat-7.0.85.tar.gz
```

检查是否安装成功

```
./version.sh
```

显示如下：

```
Using CATALINA_BASE:   /usr/local/webserver/tomcat_ht
Using CATALINA_HOME:   /usr/local/webserver/tomcat_ht
Using CATALINA_TMPDIR: /usr/local/webserver/tomcat_ht/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /usr/local/webserver/tomcat_ht/bin/bootstrap.jar:/usr/local/webserver/tomcat_ht/bin/tomcat-juli.jar
Server version: Apache Tomcat/7.0.75
Server built:   Jan 18 2017 20:54:42 UTC
Server number:  7.0.75.0
OS Name:        Linux
OS Version:     2.6.32-431.el6.x86_64
Architecture:   amd64
JVM Version:    1.7.0_45-mockbuild_2013_11_22_18_30-b00
JVM Vendor:     Oracle Corporation
```

注：Tomcat 的运行模式请使用apr模式,这样并发内存的优化更加明显。详细请更加tomcat官方网站的官方文档去配置。

2-5.安装redis

下载redis-server

```
# wget http://download.redis.io/releases/redis-4.0.8.tar.gz
```

```
#  tar xzf redis-4.0.8.tar.gz
```

```
# cd redis-4.0.8
```

```
# make
```

检查是否安装成功

```
# redis-server  -v
```

显示如下

```
Redis server v=3.2.5 sha=00000000:0 malloc=jemalloc-4.0.3 bits=64 build=e9dbdc75e63d2168
```

添加redis-server运行的内核参数在/etc/sysctl.conf 文件

```
#  vm.overcommit_memory = 1
```

```
# sysctl -p
```

2-6.安装nginx

下载源码包

```
# wget http://nginx.org/download/nginx-1.12.0.tar.gz
```

解压安装包

```
 # tar -zxvf nginx-1.12.0.tar.gz
```

进入解压的安装包

```
# cd nginx-1.12.0
```

编译

```
# ./configure --user=nginx --group=nginx --prefix=/usr/local/webserver/nginx --with-http_stub_status_module

--with-http_ssl_module --with-ld-opt="-ljemalloc" --with-http_v2_module --with-http_realip_module --with-debug
```

安装

```
# make
```

```
# make install
```

2-7.Atlas安装

下载Atlas

```
# wget  https://github.com/Qihoo360/Atlas/archive/2.2.1.tar.gz
```

注意:安装Atlas的步骤和设置步骤详细见官方网站

3.配置

3-1.配置mysql

创建mysql用户

```
#  useradd  mysql
```

创建mysqldata的目录

```
# mkdir -p  /data/mysql
```

配置权限

```
#  chonw -R mysql:mysql  /data/mysql
```

复制my.cnf文件到/data/mysql

```
#  mv  my.cnf  /data/mysql/
```

初始化数据库

```
# /usr/local/webserver/mysql/scripts/mysql_install_db  --basedir=/usr/local/webserver/mysql/   --datadir=/data/mysql/
 --defaults-file=/data/mysql/my.cnf --user=mysql
```

启动数据库

```
#  /usr/local/webserver/mysql/bin/mysqld_safe --defaults-file=/data/mysql/my.cnf  --datadir=/data/mysql --user=mysql
```

创建数据库localssc,并导入数据库文件

```
mysql> create database lottery;
```

```
# mysql -u root  -p   localssc  < /data/lottery.sql
```

修改mysql的root用户密码

```
# mysqladmin -u root password "rootroot"
```

注：数据库基本配置完毕，配置主从服务、数据库归档、mysql的自启动文件和用户权限的配置请自己去配置。

3-3.配置程序的数据库连接

进入程序所在目录

```
# /usr/local/webserver/tomcat_ht/webapps/ROOT/WEB-INF
```

修改数据库的配置文件db.properties

```
db.driver=com.mysql.jdbc.Driver
db.url=jdbc:mysql://IP.Address:3306/lottery?createDatabaseIfNotExist=true&autoReconnect=true&useUnicode=true
&characterEncoding=utf8
db.user=db.user.name
db.pass=db.user.password
db.init=100
db.min=100
db.max=2000
redis.maxIdle=200
redis.maxWaitMillis=5000
redis.maxTotal=2000
redis.testOnBorrow=true
redis.testOnReturn=true
```

```
redis.ip=127.0.0.1
redis.port=6379
```

注：这里的redis服务的ip地址要设置成后端的redis服务的ip，如果设置错误，或出现封锁表无效;如果业务量大的话，请单独配置redis集群服务。

3-4.Atlas数据库读写分离配置

注：这里配置Atlas读写分离，默认是数据库的Master-Slave都配置完毕的状态下。

my.conf 可以根据实际情况去优化

```
[client]
default-character-set = utf8
port = 7502
socket  = /tmp/mysql.sock

[mysql]
#prompt="(\u:db1@yejr.com:\R:\m:\s)[\d]> "
#pager="less -i -n -S"
#no-auto-rehash

[mysqld_safe]
socket= /tmp/mysql.sock
nice= 0

[mysqld]
#监听地址的修改
bind-address= 0.0.0.0
#开启数据库表名忽略大小写
lower_case_table_names=1
#explicit_defaults_for_timestamp=true
wait_timeout=7200 #等待时间
skip-name-resolve
#default-character-set = utf8
log-error=/data/logs/mysql_error.log
#开启慢查询日志功能
slow_query_log = 1
long_query_time = 1
#log-slow-queries= /data/logs/slowquery.log
slow_query_log_file=/data/logs/slowquery.log
```

```
#log_long_format
user = mysql
port = 7502
socket  = /tmp/mysql.sock
basedir = /usr/local/webserver/mysql
datadir = /data/mysql
open_files_limit = 10240
back_log = 600
max_connections = 5000
max_connect_errors = 60

table_open_cache = 5120
external-locking = FALSE
max_allowed_packet = 32M
sort_buffer_size = 64M
join_buffer_size = 64M
thread_cache_size = 300
###开启查询缓冲
query_cache_type = on
query_cache_size = 128M
query_cache_limit = 4M
query_cache_min_res_unit = 2k
thread_stack = 192K
transaction_isolation = READ-COMMITTED
#tmp_table_size和max_heap_table_size要相等
tmp_table_size = 2048M
max_heap_table_size = 2048M
log-bin = /data/mysql/binlog
binlog_cache_size = 4M
binlog_format = MIXED
max_binlog_cache_size = 512M
```

```
max_binlog_size = 512M
expire_logs_days = 7
#140507 Ô´ó_length_for_sort_data
max_length_for_sort_data = 8096

#配置主从的配置
server-id=1
#master-host=192.168.1.53
#master-user=repadmin
#master-password=rep@adminS.123
#master-port=7502

#根据自身情况去设置key_buffer_size
key_buffer_size = 1000M
myisam-recover-options = BACKUP
#read_buffer_size = 1M
read_buffer_size = 2M
#read_rnd_buffer_size = 16M
read_rnd_buffer_size = 32M
#bulk_insert_buffer_size = 64M
bulk_insert_buffer_size = 128M
#myisam_sort_buffer_size = 128M
myisam_sort_buffer_size = 4096
#myisam_max_sort_file_size = 5G
#myisam_max_extra_sort_file_size = 5G
myisam_repair_threads = 1
myisam-recover-options = 1
#innodb_buffer_pool_size一般为内存的75%-80%
innodb_buffer_pool_size = 20000M
innodb_data_file_path = ibdata1:1024M:autoextend
#innodb_file_io_threads = 4
```

```
innodb_read_io_threads = 8
innodb_write_io_threads = 64
innodb_thread_concurrency = 32
innodb_flush_log_at_trx_commit = 2
innodb_log_buffer_size = 16M
innodb_log_file_size = 128M
innodb_log_files_in_group = 3
innodb_max_dirty_pages_pct = 90
innodb_lock_wait_timeout = 120
innodb_file_per_table = 1
#innodb_use_sys_malloc = 1
[mysqldump]
quick
max_allowed_packet = 1024M
socket  = /tmp/mysql.sock
```