

- * 打包
- * 部署
- * 目录结构
- * 脚本

打包

在工程目录下执行：

```
1 mvn clean package -Pdev
2 或者
3 mvn clean package -Ppub
```

会在工程目录的 /target目录下生成 alloc-release-x.x.x.tar.gz

部署

```
1 下载alloc最新包https://github.com/mpusher/alloc/releases
2 解压下载的tar包tar -zxvf alloc-release-x.x.x.tar.gz
3 修改 conf 目录下的 vi mpush.conf文件，修改方式参照mpush server 部署
4 给bin目录下的脚本增加执行权限chmod u+x *.sh
5 执行./mp.sh start 启动服务，查看帮助直接执行./mp.sh
6 cd logs目录，cat mpush.out查看服务是否启动成功
```

目录结构

命令输出release目录结构：

```
1 cd /d D:\mpush-alloc-0.8.0
2 tree /F >D:\mpush-alloc-0.8.0\tree.txt
```

mpush-alloc-0.8.0目录结构

```
1 | LICENSE
2 | README.md
3 | tree.txt
4 |
5 |─bin
6 | bootstrap.jar
7 | env-mp.sh
8 | mp.sh
9 | set-env.sh
10 |
11 |─conf
12 | logback.xml
13 | mpush.conf
```

```
14 |
15 |└─lib
16 |  barchart-udt-bundle-2.3.0.jar
17 |  commons-lang3-3.6.jar
18 |  commons-pool2-2.4.2.jar
19 |  config-1.3.1.jar
20 |  curator-client-2.11.1.jar
21 |  curator-framework-2.11.1.jar
22 |  curator-recipes-2.11.1.jar
23 |  curator-x-discovery-2.11.1.jar
24 |  fastjson-1.2.36.jar
25 |  guava-20.0.jar
26 |  jackson-core-asl-1.9.13.jar
27 |  jackson-mapper-asl-1.9.13.jar
28 |  javassist-3.21.0-GA.jar
29 |  jcl-over-slf4j-1.7.25.jar
30 |  jedis-2.9.0.jar
31 |  jline-0.9.94.jar
32 |  jul-to-slf4j-1.7.25.jar
33 |  log4j-1.2.16.jar
34 |  log4j-over-slf4j-1.7.25.jar
35 |  logback-classic-1.2.3.jar
36 |  logback-core-1.2.3.jar
37 |  mpush-api-0.8.0.jar
38 |  mpush-cache-0.8.0.jar
39 |  mpush-client-0.8.0.jar
40 |  mpush-common-0.8.0.jar
41 |  mpush-monitor-0.8.0.jar
42 |  mpush-netty-0.8.0.jar
43 |  mpush-tools-0.8.0.jar
44 |  mpush-zk-0.8.0.jar
45 |  netty-buffer-4.1.8.Final.jar
46 |  netty-codec-4.1.8.Final.jar
47 |  netty-codec-http-4.1.8.Final.jar
48 |  netty-common-4.1.8.Final.jar
49 |  netty-handler-4.1.8.Final.jar
50 |  netty-resolver-4.1.8.Final.jar
51 |  netty-transport-4.1.8.Final.jar
52 |  netty-transport-native-epoll-4.1.8.Final-linux-x86_64.jar
53 |  netty-transport-sctp-4.1.8.Final.jar
54 |  netty-transport-udt-4.1.8.Final.jar
55 |  slf4j-api-1.7.25.jar
56 |  zookeeper-3.4.8.jar
57 |
```

脚本

env-mp.sh

```
#!/usr/bin/env bash
```

```
# Licensed to the Apache Software Foundation (ASF) under one or more
# contributor license agreements. See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
```

```
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
```

```
# This script should be sourced into other mpush
# scripts to setup the env variables
```

```
# We use MP_CFG_DIR if defined,
# otherwise we use /etc/mp
# or the conf directory that is
# a sibling of this script's directory
```

```
MP_BIN_DIR="${MP_BIN_DIR:-/usr/bin}"    bin目录
MPUSH_PREFIX="${MP_BIN_DIR}/.."        bin的上一级目录
MPUSH_HOME=$MPUSH_PREFIX               mpush-alloc-0.8.0目录
```

```
if [ "x$MP_CFG_DIR" = "x" ]           如果MP_CFG_DIR为空
then
    if [ -e "${MPUSH_PREFIX}/conf" ]; then  如果/mpush-alloc-0.8.0/conf
        MP_CFG_DIR="${MP_BIN_DIR}/../conf"  目录存在, 设置MP_CFG_DIR目
    else
        MP_CFG_DIR="${MP_BIN_DIR}/../etc/mpush" 录
    fi
fi
```

```
if [ "x${MP_DATA_DIR}" = "x" ]  如果为空
then
    MP_DATA_DIR="${MPUSH_PREFIX}/tmp"  指定/xxxx-0.8.0/tmp
fi                                     为临时数据目录
```

```
if [ "x${MP_LOG_DIR}" = "x" ]
then
    MP_LOG_DIR="${MPUSH_PREFIX}/logs"  指定日志目录
fi
```

```
if [ -f "${MP_BIN_DIR}/set-env.sh" ]; then  如果存在此文件
    . "${MP_BIN_DIR}/set-env.sh"           执行set-env.sh
fi
```

```
if [ "x$MP_CFG" = "x" ]
```

```

then
    MP_CFG="mpush.conf"    指定配置文件
fi

MP_CFG="$MP_CFG_DIR/$MP_CFG"    指定配置文件路径

if [ -f "$MP_BIN_DIR/java.env" ]    如果bin目录中存在此
then                                文件，这执行它
    . "$MP_BIN_DIR/java.env"
fi

if [ "x${MP_LOG4J_PROP}" = "x" ]
then
    MP_LOG4J_PROP="INFO,CONSOLE"
fi

if [ "$JAVA_HOME" != "" ]; then    如果JAVA_HOME不为空
    JAVA="$JAVA_HOME/bin/java"
else
    JAVA=java
fi

#add the conf dir to classpath
CLASSPATH="$MP_CFG_DIR:$CLASSPATH"    conf目录加入到classpath

#make it work in the binary package
#(use array for LIB_PATH to account for spaces within wildcard expansion)
if [ -e "${MPUSH_PREFIX}/../lib/plugins/*.jar" ]; then
    LIB_PATH=( "${MPUSH_PREFIX}/../lib/plugins/*.jar" )
fi

for i in "${LIB_PATH[@]}"
do
    CLASSPATH="$i:$CLASSPATH"    加载plugins目录下的
done                                所有jar到classpath中

case "`uname`" in
    CYGWIN*) cygwin=true ;;    判定为cygwin则设置为true
    *) cygwin=false ;;    否则false
esac    退出case

if $cygwin
then
    CLASSPATH=`cygpath -wp "$CLASSPATH"`
fi    将cygwin路径转换为win路径

```

set-env.sh

```

#!/usr/bin/env bash
#开启远程调试
#JVM_FLAGS="$JVM_FLAGS -Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=8008"

#GC配置
#运行模式 整个堆内存大小 GC算法
#JVM_FLAGS="$JVM_FLAGS -server -Xmx1024m -Xms1024m -XX:+UseG1GC -XX:MaxGCPauseMillis=200"
#GC日志 发生OOM时创建堆内存转储文件
#JVM_FLAGS="$JVM_FLAGS -Xloggc:$MP_LOG_DIR/gc.log -XX:+PrintGCDetails -XX:+PrintGCDateStamps"
#发生OOM后的操作
#JVM_FLAGS="$JVM_FLAGS -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=$MP_LOG_DIR -XX:OnOutOfMemoryError=$MP_BIN_DIR/restart.sh"

```

mp.sh



```
#!/usr/bin/env bash

# Licensed to the Apache Software Foundation (ASF) under one or more
# contributor license agreements. See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

#
# If this scripted is run out of /usr/bin or some other system bin directory
# it should be linked to and not copied. Things like java jar files are found
# relative to the canonical path of this script.
#

# use POSTIX interface, symlink is followed automatically
MP_BIN="${BASH_SOURCE-$0}"
MP_BIN="$(dirname "${MP_BIN}")"
MP_BIN_DIR="$(cd "${MP_BIN}"; pwd)" bin目录

if [ -e "$MP_BIN../libexec/env-mp.sh" ]; then  是否存在此脚本
    . "$MP_BIN_DIR../libexec/env-mp.sh"
else
    . "$MP_BIN_DIR/env-mp.sh"  执行此脚本
fi

# See the following page for extensive details on setting
# up the JVM to accept JMX remote management:
# http://java.sun.com/javase/6/docs/technotes/guides/management/agent.html
# by default we allow local JMX connections
if [ "x$JMXLOCALONLY" = "x" ]  如果此变量为空
then
    JMXLOCALONLY=false  设置为false
fi

if [ "x$JMXDISABLE" = "x" ] || [ "$JMXDISABLE" = 'false' ]  如果JMX是开启的
then
    echo "MPush JMX enabled by default" >&2
    if [ "x$JMXPORT" = "x" ]  如果jmx port为空
    then
        # for some reason these two options are necessary on jdk6 on Ubuntu
        # accord to the docs they are not necessary, but otw jconsole cannot
        # do a local attach
        MP_MAIN="-Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.local.only=$JMXLOCALONLY"
    else
        if [ "x$JMXAUTH" = "x" ]  否则如果jmx port不为空
        then
            JMXAUTH=false
        fi
        if [ "x$JMXSSL" = "x" ]
        then
            JMXSSL=false
        fi
        if [ "x$JMXLOG4J" = "x" ]
        then
            JMXLOG4J=true
        fi
        echo "MPush remote JMX Port set to $JMXPORT" >&2  设置JMX相关参数
        echo "MPush remote JMX authenticate set to $JMXAUTH" >&2
        echo "MPush remote JMX ssl set to $JMXSSL" >&2
        echo "MPush remote JMX log4j set to $JMXLOG4J" >&2
        MP_MAIN="-Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=$JMXPORT -
Dcom.sun.management.jmxremote.authenticate=$JMXAUTH -Dcom.sun.management.jmxremote.ssl=$JMXSSL -Dmpush.jmx.log4j.disable=$JMXLOG4J"
    fi
else
    echo "JMX disabled by user request" >&2  如果JMX是禁用的
    MP_MAIN=""
fi

MP_MAIN="$MP_MAIN -jar $MP_BIN_DIR/bootstrap.jar"

if [ "x$SERVER_JVM_FLAGS" != "x" ]
then
    JVM_FLAGS="$SERVER_JVM_FLAGS $JVM_FLAGS"
fi

if [ "x$2" != "x" ]  命令的第2个参数不为空
then
    MP_CFG="$MP_CFG_DIR/$2"
fi

# if we give a more complicated path to the config, don't screw around in $MP_CFG_DIR
if [ "x$(dirname "$MP_CFG")" != "x$MP_CFG_DIR" ]  如果由命令指定的配置
```

```

then                                如果由命令行指定的配置
    MP_CFG="$2"                      置目录与当前配置目录
fi                                    不一致，则使用命令行
if $cygwin                           指定的配置目录
then
    MP_CFG=`cygpath -wp "$MP_CFG"`    将cygwin路径转换为win路径
    # cygwin has a "kill" in the shell itself, gets confused
    KILL=/bin/kill                    设定win下的kill
else
    KILL=kill                         linux kill
fi

echo "Using config: $MP_CFG" >&2

case "$OSTYPE" in
*solaris*)                          系统类型如果是solaris，则指定grep命令
    GREP=/usr/xpg4/bin/grep
    ;;
*)
    GREP=grep                        其他系统，直接指定grep
    ;;
esac 退出
if [ -z "$MP_PID_FILE" ]; then        如果MP_PID_FILE串的长度为0
#   MP_DATA_DIR=${GREP^^[:space:]}*dataDir "$MP_CFG" | sed -e 's/.*/ /'
if [ ! -d "$MP_DATA_DIR" ]; then      如果不存在xxx/tmp数据目录，则创建一个
    mkdir -p "$MP_DATA_DIR"
fi
    MP_PID_FILE="$MP_DATA_DIR/mpush_server.pid" 指定进程PID文件
else
    # ensure it exists, othw stop will fail
    mkdir -p "$(dirname "$MP_PID_FILE")"      创建进程PID文件目录
fi

if [ ! -w "$MP_LOG_DIR" ]; then
mkdir -p "$MP_LOG_DIR"                创建xxx/log日志目录
fi

_MP_DAEMON_OUT="$MP_LOG_DIR/mpush.out" 日志文件

case $1 in
start)                                mp.sh start          $0=mp.sh $1=start
    echo -n "Starting mpush ... "
    if [ -f "$MP_PID_FILE" ]; then      如果PID文件存在
        if kill -0 `cat "$MP_PID_FILE"` > /dev/null 2>&1; then 检查进程是否存在
            echo $command already running as process `cat "$MP_PID_FILE"`.
            exit 0                      正常退出
        fi
    fi
    # 核心的JAVA程序启动命令
    nohup "$JAVA" "-Dmp.home=$MPUSH_HOME" "-Dmp.conf=$MP_CFG" -cp "$CLASSPATH" $JVM_FLAGS $MP_MAIN > "$_MP_DAEMON_OUT" 2>&1 < /dev/null &
    if [ $? -eq 0 ]; then               $?是shell变量,表示"最后一次执行命令"的退出状态.0为成功,非0为失败.
    then
        case "$OSTYPE" in
*solaris*)
            /bin/echo "${!}\c" > "$MP_PID_FILE"
            ;;
*)
            /bin/echo -n $! > "$MP_PID_FILE"
            ;;
esac
        if [ $? -eq 0 ]; then           如果设置成功
            then
                sleep 1                 睡眠1S，输出STARTED
                echo STARTED
            else
                echo FAILED TO WRITE PID 否则输出失败，且异常
                exit 1                  退出
            fi
        else
            echo SERVER DID NOT START 如果JAVA程序启动失败
            exit 1                      败，则异常退出
        fi
    ;;
start-foreground)
    "$JAVA" "-Dmp.home=$MPUSH_HOME" "-Dmp.conf=$MP_CFG" -cp "$CLASSPATH" $JVM_FLAGS $MP_MAIN 前台方式启动应用
    ;;
print-cmd)                            打印启动命令
    echo "\"$JAVA\" \"$MP_MAIN\" "
    echo "\"-Dmp.home=$MPUSH_HOME -Dmp.conf=$MP_CFG\" "
    echo "$JVM_FLAGS "
    echo "-cp \"$CLASSPATH\" "
    echo "> \"$_MP_DAEMON_OUT\" 2>&1 < /dev/null"
    ;;
stop)                                  mp.sh stop
    echo "Stopping mpush ... "
    if [ ! -f "$MP_PID_FILE" ]; then    如果不存在进程PID文件
    then
        echo "no mpush to stop (could not find file $MP_PID_FILE)" 则认为没有可停止的应用
    else
        $KILL -15 $(cat "$MP_PID_FILE") 等待应用退出，然后终止进程
        SLEEP=30
        SLEEP_COUNT=1
        while [ $SLEEP -ge 0 ]; do      -ge:大于等于
            kill -0 $(cat "$MP_PID_FILE") >/dev/null 2>&1 检查进程是否存在 存在=0，不存在=1
            if [ $? -gt 0 ]; then        如果进程不存在
                rm -f "$MP_PID_FILE" >/dev/null 2>&1 删除进程PID文件
            fi
        done
    fi

```

```

rm -f "$MP_PID_FILE" >/dev/null 2>&1      删除进程PID文件
if [ $? != 0 ]; then      如果删除失败
    if [ -w "$MP_PID_FILE" ]; then      如果文件存在,且可写
        cat /dev/null > "$MP_PID_FILE"      清空文件内容
    else
        echo "The PID file could not be removed or cleared."      不能清空PID文件
    fi
fi
echo STOPPED
break
fi
if [ $SLEEP -gt 0 ]; then
    echo "stopping ... $SLEEP_COUNT"      如果睡眠时间>0, 输出睡眠次数, 继续睡
    sleep 1
fi
if [ $SLEEP -eq 0 ]; then      如果睡眠时间=0
    echo "MPUSH did not stop in time."      30S, 应用还未停止, 打印线程栈、强制退出
    echo "To aid diagnostics a thread dump has been written to standard out."
    kill -3 `cat "$MP_PID_FILE"`      打印线程栈
    echo "force stop MPUSH."
    kill -9 `cat "$MP_PID_FILE"`      强制杀进程
    echo STOPPED
fi
SLEEP=`expr $SLEEP - 1`      每次循环, 睡眠时间-1
SLEEP_COUNT=`expr $SLEEP_COUNT + 1`      睡眠次数+1
done
fi
exit 0
;;
upgrade) 更新
shift
echo "upgrading the servers to 3.*"
"$JAVA" -cp "$CLASSPATH" $JVM_FLAGS com.mpush.tools.upgrade.UpgradeMain ${@}
echo "Upgrading ..."
;;
restart) 重启
shift      shift 参数左移,每次运行shift(不带参数)销毁一个参数,后面的参数前移
"$0" stop ${@}      mp.sh xxx 经过shift移除了xxx之后, 变为执行mp.sh stop
sleep 1
"$0" start ${@}      执行mp.sh start
;;
status)
# -q is necessary on some versions of linux where nc returns too quickly, and no stat result is output
clientPortAddress=`$GREP "^[:space:]]*clientPortAddress[[:alpha:]]" "$MP_CFG" | sed -e 's/.*=//'`
if [ ! $clientPortAddress ]
then
    clientPortAddress="localhost"
fi
clientPort=`$GREP "^[:space:]]*connect-server-port[[:alpha:]]" "$MP_CFG" | sed -e 's/.*=//'\`
telnet 127.0.0.1 3002
;;
*)
echo "Usage: $0 {start|start-foreground|stop|restart|status|upgrade|print-cmd}" >&2
      如果没有匹配的命令, 则输出此条帮助信息
esac

```

shell 脚本知识

- 1 `$?` `-eq 0` `$?`是shell变量,表示"最后一次执行命令"的退出状态.`0`为成功,非`0`为失败。
- 2 `kill -0` 不发送任何信号, 检查进程是否存在, 存在=`0`, 不存在=`1`
- 3 `kill -3` 打印线程调用栈
- 4 `kill -9` 不等待应用退出, 强制杀掉进程
- 5 `kill -15` 等待应用退出, 然后终止进程
- 6
- 7 `${@}` 输出所有参数, 假如 `./test.sh a b c` 则`${@}`输出: `a b c`
- 8
- 9
- 10 `shift` 参数左移, 每次运行`shift`(不带参数)销毁一个参数, 后面的参数前移
- 11 `expr $SLEEP - 1` 表达式, 修改变量`SLEEP`值
- 12 `exit (0)`: 正常运行程序并退出程序;
- 13 `exit (1)`: 非正常运行导致退出程序;
- 14
- 15 `-eq`: 等于
- 16 `-ne`: 不等于

- 17 -le: 小于等于
- 18 -ge: 大于等于
- 19 -lt: 小于
- 20 -gt: 大于