

# 集市管理配置文档

## 前言

集市管理系统 (MPP Console) 是 Yonghong Z-Suite 一个独立的功能模块。它被用来管理 Yonghong MPP 数据集市。可以监测集市中各台机器的 CPU、内存、磁盘的使用情况和网络速度；监测各个节点的 JVM (Java Virtual Machine) 状态；可以修改全局参数和每个节点自己的参数；可以启动、停止、重启节点；可以更新集市里各个节点用到的 jar 文件。集市管理使用到的端口：5057和5032。

注意事项：只有linux集群才可以配置集市管理

本文配置流程：配置分节点进行，分为N节点和非N节点，以下：CM、CR、CMR、MR、M、R、C中未包含N的都属于非N节点。

## 一、N节点配置manager和console

### 1. 配置manager

manager路径：`永洪安装目录/manager/bin/console.properties`

配置如下：

```
1 # 指定os版本（操作系统）
2 console.os=linux
3 # 数据库驱动
4 console.db.driver=com.mysql.jdbc.Driver
5 # 数据库厂商，如oracle等
6 console.db.vendor=mysql
7 # 连接信息：mysql是：jdbc:mysql://ip:port/database
8 # Oracle为：jdbc:oracle:thin:@ip:port/database
9 console.db.url=jdbc:mysql://192.168.227.1:3307/yonghong1
10 # N节点IP
11 console.manager=192.168.227.131
12 # 数据库连接用户名
13 console.db.user=root
14 # 数据库连接密码
15 console.db.passwd=admin1234
16 # 本节点或本机IP
17 console.io.ip=192.168.227.131
18 # 指定本节点的bihome目录
19 console.node.home=/usr/bi9/Yonghong/bihome
20 # 指定本节点的工作路径
21 console.node.path=/usr/bi9/tomcat/bin
```

```
#utf-8
#Fri Apr 02 11:56:06 CST 2021
console.os=linux
console.db.driver=com.mysql.jdbc.Driver
console.db.vendor=mysql
console.db.url=jdbc:mysql://192.168.227.1:3306/yonghong1
console.manager=192.168.227.131
console.db.user=root
console.io.ip=192.168.227.131
console.node.home=/usr/bi9/Yonghong/bihome
console.db.passwd=admin1234
console.node.path=/usr/bi9/tomcat/bin
```

## 2. 配置console

console路径: 永洪安装目录/console/bin/concole.properties

配置如下:

```
1 # 指定os版本（操作系统）
2 console.os=linux
3 # 端口偏移量，如果一台机器启动多个Console，需要指定偏移量。比如N节点，已经有一个
  manager了，所以对应的console一定要加偏移
4 console.port.offset=1
5 # N节点IP
6 console.manager=192.168.227.131
7 # 本节点IP
8 console.io.ip=192.168.227.131
9 console.node.home=/usr/bi9/Yonghong/bihome
10 console.node.path=/usr/bi9/tomcat/bin
```

```
#utf-8
#Fri Apr 02 11:56:06 CST 2021
console.os=linux
console.port.offset=1
console.manager=192.168.227.131
console.io.ip=192.168.227.131
console.node.home=/usr/bi9/Yonghong/bihome
console.node.path=/usr/bi9/tomcat/bin
~
```

## 3. 非N节点只要配置console即可

console路径: 永洪安装目录/console/bin/concole.properties

配置如下:

```

1 # 指定os版本（操作系统）
2 console.os=linux
3 # 端口偏移量，如果一台机器启动多个Console，需要指定偏移量。比如N节点，已经有一个
  manager了，所以对应的console一定要加偏移
4 console.port.offset=1
5 # N节点IP
6 console.manager=192.168.227.131
7 # 本节点IP
8 console.io.ip=192.168.227.133
9 console.node.home=/usr/BI9/Yonghong/bihome
10 console.node.path=/usr/BI9/tomcat/bin

```

```

#utf-8
#Wed Apr 07 14:39:31 CST 2021
console.os=linux
console.port.offset=1
console.manager=192.168.227.131
console.io.ip=192.168.227.133
console.node.home=/usr/BI9/Yonghong/bihome
console.node.path=/usr/BI9/tomcat/bin

```

## 二、启动顺序

启动顺序不可随意，需严格按照顺序启动：

先启动N节点的manager，再启动各个非N节点的console，最后启动产品。

注意事项：产品tomcat重启不需要重启集市管理。

- 启动方式

N节点manager启动：永洪安装路径/manager/bin/ 下执行 sh startConsole.sh 启动

N节点console启动：永洪安装路径console/bin/ 下执行 sh startConsole.sh 启动

非N节点console启动：永洪安装路径console/bin/ 下执行 sh startConsole.sh 启动

启动后看看nohup.out日志是否有报错，如果有，后续相应处理。

- 启动成功后，去集群C节点访问系统监控-集市管理模块，可直观的查看当前每个节点状态、cpu使用率、磁盘空间等等

The screenshot shows a web management interface with a navigation menu on the left. The 'Market Management' (集市管理) option is highlighted. The main content area is titled 'Market Parameters' (集市参数) and contains a table of global parameters (全局参数) and a section for parameter modification results (参数修改结果).

全局参数名	全局参数值	详细参数值
compare tent local	true	true
dc io local	false	false
dc node naming	192.168.1.211	192.168.1.211
rpc repeat	false	false

参数名称	初始	状态

### 三、常见报错处理

#### 1. ifconfig未安装

```
18:07:42.149[SEVERE]StateWorker.work: For input string: "ifconfig:"
java.lang.NumberFormatException: For input string: "ifconfig:"
    at sun.misc.FloatingDecimal.readJavaFormatString(FloatingDecimal.java:2043)
    at sun.misc.FloatingDecimal.parseDouble(FloatingDecimal.java:110)
    at java.lang.Double.parseDouble(Double.java:538)
    at g5.console.worker.impl.NetWorker.parseOnLinux(Unknown Source)
    at g5.console.worker.StateWorker.parse(Unknown Source)
    at g5.console.worker.StateWorker.work0(Unknown Source)
    at g5.console.worker.StateWorker.work(Unknown Source)
    at g5.console.worker.WorkerTask.run(Unknown Source)
    at java.util.TimerThread.mainLoop(Timer.java:555)
    at java.util.TimerThread.run(Timer.java:505)
```

安装ifconfig: `yum install net-tools`

#### 2. 端口被占用

检查偏移是否配置对了, ip配置是否正确。

```
18:07:59.544[INFO]StateWorker.<init>: cmd is sh /opt/yh75n/console/Script/disk.sh
java.net.BindException: 地址已在使用
    at sun.nio.ch.Net.bind0(Native Method)
    at sun.nio.ch.Net.bind(Net.java:433)
    at sun.nio.ch.Net.bind(Net.java:425)
    at sun.nio.ch.ServerSocketChannelImpl.bind(ServerSocketChannelImpl.java:223)
    at sun.nio.ch.ServerSocketAdaptor.bind(ServerSocketAdaptor.java:74)
    at g5.dc.io.IOServant.<init>(Unknown Source)
    at g5.dc.io.IOServant.init(Unknown Source)
    at g5.console.util.ConsoleUtil.init(Unknown Source)
    at g5.console.ConsoleMain.main(Unknown Source)
```

#### 3. Script路径下脚本执行无报错

N节点路径manager/Script、console/Script与非N节点console/Script下都有四个.sh文件(cpu.sh、disk.sh、mem.sh、net.sh), 分别执行以下四个脚本: sh cpu.sh; sh disk.sh; sh mem.sh; sh net.sh, 要确保执行无报错。集市管理是通过定时执行这四个脚本搜集服务器cpu,磁盘, 内存, 网络信息。如果以上脚本执行出现报错, 对应文件打开检查一下。

最常见的错误是net.sh执行报错"Device not found", 原因是网卡不对, 命令ifconfig查询一下网卡, 然后修改。报错如下截图:

```
ether: error fetching interface information: Device not found
ether: error fetching interface information: Device not found
ether: error fetching interface information: Device not found
ether: error fetching interface information: Device not found
net.sh: line 36: (-)*5: syntax error: operand expected (error token is ")*5")
```

修改网卡信息请参考以下截图例子: 如查询当前服务器ip的网卡是wifi0, 则需要将net.sh中的eno和eth全部替换为wifi0, 分别使用命令: %s/eno/wifi0/g与%s/eth/wifi0/g全部进行替换。替换后再次执行脚本sh net.sh检查是否还有报错。

```
root@DESKTOP-43CVL1V:/#
root@DESKTOP-43CVL1V:/# ifconfig
lo: flags=73<UP, LOOPBACK, RUNNING> mtu 1500
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0xfe<compat, link, site, host>
    loop (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wifi0: flags=4163<UP, BROADCAST, RUNNING, MULTICAST> mtu 1500
    inet 192.168.1.211 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 2409:8a62:3e6:2ba0:c834:7ff1:6b7d:27d6 prefixlen 64 scopeid 0x0<global>
    inet6 2409:8a62:3e6:2ba0:a1e2:7a70:b0cf:9487 prefixlen 128 scopeid 0x0<global>
    inet6 fe80::c834:7ff1:6b7d:27d6 prefixlen 64 scopeid 0xfd<compat, link, site, host>
    ether 70:c9:4e:ea:ce:33 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```
#!/bin/bash
isCentOs=`ls /sys/class/net | grep wifi0`
if [ -n "$isCentOs" ];then
#for centos 7.0
eth=`ifconfig | grep wifi0 | awk '{print $1}' | awk -F":" '{print $1}'`
for ee in ${eth}
do
RXpre=$(ifconfig ${ee} | grep 'RX packets' | awk '{print $5}')
TXpre=$(ifconfig ${ee} | grep 'TX packets' | awk '{print $5}')
sleep 0.2
RXnext=$(ifconfig ${ee} | grep 'RX packets' | awk '{print $5}')
TXnext=$(ifconfig ${ee} | grep 'TX packets' | awk '{print $5}')
echo "${ee} $(((($RXnext)-$RXpre)*5)) $(((($TXnext)-$TXpre)*5))"
done
else
eth=`ifconfig | grep eth | awk '{print $1}'`
for ee in ${eth}
do
RXpre=$(ifconfig ${ee} | grep bytes | awk '{print $2}' | awk -F":" '{print $2}')
TXpre=$(ifconfig ${ee} | grep bytes | awk '{print $6}' | awk -F":" '{print $2}')
: %s/eth/wifi0/g_

```

替换脚本中所有eth为wifi0

```
#!/bin/bash
isCentOs=`ls /sys/class/net | grep eno`
if [ -n "$isCentOs" ];then
#for centos 7.0
eth=`ifconfig | grep eno | awk '{print $1}' | awk -F":" '{print $1}'`
for ee in ${eth}
do
RXpre=$(ifconfig ${ee} | grep 'RX packets' | awk '{print $5}')
TXpre=$(ifconfig ${ee} | grep 'TX packets' | awk '{print $5}')
sleep 0.2
RXnext=$(ifconfig ${ee} | grep 'RX packets' | awk '{print $5}')
TXnext=$(ifconfig ${ee} | grep 'TX packets' | awk '{print $5}')
echo "${ee} $(((($RXnext)-$RXpre)*5)) $(((($TXnext)-$TXpre)*5))"
done
else
eth=`ifconfig | grep eth | awk '{print $1}'`
for ee in ${eth}
do
RXpre=$(ifconfig ${ee} | grep bytes | awk '{print $2}' | awk -F":" '{print $2}')
TXpre=$(ifconfig ${ee} | grep bytes | awk '{print $6}' | awk -F":" '{print $2}')
: %s/eno/wifi0/g_

```

将当前net.sh脚本中所有eno替换为