|  |
| --- |
|  |
| **北京数字创新科技有限公司**  **微服务监控应用系统V1.0使用说明书** |
|  |

**二〇一七年二月**

**目 录**

[第1章 概述 3](#_Toc479179337)

[1.1 背景 3](#_Toc479179338)

[1.2 系统简介 3](#_Toc479179339)

[1.3 系统运行环境 4](#_Toc479179340)

[1.4 版权说明 5](#_Toc479179341)

[第2章 软件安装与配置 5](#_Toc479179342)

[2.1 MySQL Server安装 5](#_Toc479179343)

[2.2 JDK8安装 7](#_Toc479179344)

[2.3 Redis集群搭建 8](#_Toc479179345)

[2.4 Zookeeper集群搭建 9](#_Toc479179346)

[2.5 Kafka集群搭建 11](#_Toc479179347)

[2.6 Elasticsearch集群搭建 12](#_Toc479179348)

[2.7 Storm 集群搭建 13](#_Toc479179349)

[2.8 Tomcat 安装 15](#_Toc479179350)

[2.9 微服务监控系统部署 16](#_Toc479179351)

[第3章 软件功能介绍与使用说明 17](#_Toc479179352)

[3.1 监控面板 17](#_Toc479179353)

[3.2 系统管理 18](#_Toc479179354)

[3.2.1 用户管理 18](#_Toc479179355)

[3.2.2 菜单管理 19](#_Toc479179356)

[3.2.3 用户组管理 20](#_Toc479179357)

[3.3 统一权限管理 22](#_Toc479179358)

[3.3.1 职务信息管理 22](#_Toc479179359)

[3.3.2 组织信息管理 23](#_Toc479179360)

[3.3.3 系统信息管理 24](#_Toc479179361)

[3.3.4 角色信息管理 26](#_Toc479179362)

[3.3.5 小组信息管理 27](#_Toc479179363)

[3.4 系统监控 28](#_Toc479179364)

[3.4.1 实例状态 28](#_Toc479179365)

[3.4.2 异常信息查询 35](#_Toc479179366)

[3.5 监控管理 36](#_Toc479179367)

[3.5.1 系统管理 36](#_Toc479179368)

[3.5.2 预警配置管理 38](#_Toc479179369)

[3.5.3 系统组-用户组管理 39](#_Toc479179370)

[3.5.4 业务管理 40](#_Toc479179371)

[3.5.5 短信配置管理 41](#_Toc479179372)

[3.5.6 邮件配置管理 42](#_Toc479179373)

[3.6 消息监控 43](#_Toc479179374)

[3.6.1 MQ总体概览图 43](#_Toc479179375)

[3.6.2 MQ连接信息 44](#_Toc479179376)

[3.6.3 MQ队列信息 44](#_Toc479179377)

[3.6.4 MQ交换机信息 44](#_Toc479179378)

[3.6.5 MQ通道 45](#_Toc479179379)

[3.6.6 MQ消息跟踪 45](#_Toc479179380)

[3.7 消息管理 46](#_Toc479179381)

[3.7.1 集群管理 46](#_Toc479179382)

[3.7.2 用户管理 47](#_Toc479179383)

[3.7.3 数据字典 48](#_Toc479179384)

[3.7.4 接出服务注册 50](#_Toc479179385)

[3.7.5 队列配置 51](#_Toc479179386)

[3.7.6 介入服务注册 52](#_Toc479179387)

[3.7.7 交换机配置 53](#_Toc479179388)

[3.7.8 Direct方式注册 54](#_Toc479179389)

[3.8 行为管理 55](#_Toc479179390)

[3.8.1 访问分析 56](#_Toc479179391)

[3.8.2 行为分析 56](#_Toc479179392)

# 概述

## 背景

随着业务发展和计算机应用技术的飞速发展，企业信息化建设步伐越来越快，相应的业务应用系统逐渐增多，线上系统的数量也在不断增加，依靠过去人工巡检系统的方式发现系统故障、潜在风险及安全隐患的方式效率越来越低下且运维人员的工作强度及压力也在不断增加，为了提高发现系统故障的及时性、系统维护的专业性、规范化、科学性同时也能把运维人员从重复的工作中解放出来去做更多有意义的事情，因此亟需引入新的监控手段、工具来协助运维工程师解决当前的问题。

因此数字创新微服务监控应用系统就在此前提下诞生、成长最终发展为可进行IT硬件资源监控、服务跟踪和服务调用链监控的应用软件系统。

## 系统简介

为保证自有软件平台运行稳定性，对线上平台进行自动化监控，合理设置监控粒度及监控对象。尽可能的把潜在问题在萌芽状态解决及消除隐患，以此提高IT技术支持部门的整体集成能力和交付系统运行质量。

数字创新微服务监控应用系统的主要应用目标是：

1. 进行业务系统的服务跟踪和服务调用链监控，为系统故障及性能瓶颈排查，业务系统性能优化提供直观参考依据。
2. 不仅仅面向运维人员、开发人员而且也面向测试人员，例如根据监控平台采集到JVM Young GC/Full GC次数及时间、耗时Top 10线程堆栈等信息，经常查看监控平台，开发、测试人员便可以评估代码质量，排除隐患，同时也能提高系统维护的专业性和规范性。
3. 通过系统监控实时告警可以帮助运维人员实时发现定位问题及时处理，保证系统良好运行，提高用户体验，降低服务宕机时间。

## 系统运行环境

数字创新微服务监控应用系统运行的软件件环境如下表所示：

1、硬件环境

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **项目** | **最低硬件配置** | **备注** |
|  | Zookeeper集群 | CPU：2.0 GHZ  内存：4Gb  存储：40Gb |  |
|  | Elasticsearch集群 | CPU：2.0 GHZ  内存：4Gb  存储：40Gb |  |
|  | Kafa集群 | CPU：2.0 GHZ  内存：4Gb  存储：20Gb |  |
|  | Storm集群 | CPU：2.0 GHZ  内存：4Gb  存储：20Gb |  |
|  | 数据库服务器 | CPU：2.4 GHZ  内存：4Gb  存储：40Gb |  |
|  | Web服务器 | CPU：2.4 GHZ  内存：8Gb  存储：40Gb |  |
|  | Redis 集群 | CPU：2.4 GHZ  内存：4Gb  存储：40Gb |  |

2、软件环境

|  |  |  |
| --- | --- | --- |
| **序号** | **项目** | **备注** |
|  | Linux6.4，CentOS6.4，Windows7/8/10/2008/2012 | 操作系统 |
|  | JDK8 |  |
|  | zookeeper-3.4.9 |  |
|  | kafka\_2.10-0.8.2.2 |  |
|  | elasticsearch-2.4.1 |  |
|  | apache-storm-0.10.0 |  |
|  | Ruby22 |  |
|  | Python 2.7 |  |
|  | apache-tomcat-9.0.0 |  |
|  | MySQL-5.6 |  |
|  | redis-3.0.6 |  |

## 版权说明

（1）限制

禁止将数字创新IT监控应用系统之拷贝以盈利为目的传播于他人。您不得对数字创新IT监控应用系统进行反向工程(Reverse engineer)、反向编译(Decompile)或反汇编(Disassemble)。您必须遵守所有数字创新IT监控应用系统产品使用之相关法律。此限制如与法律法规冲突，则以法律法规为准。

（2）、著作权。

凡与本数字创新IT监控应用系统及其拷贝有关之所有权与著作权均属北京数字创新科技优先公司所有。凡与因透过本数字创新IT监控应用系统而存取之资料内容之所有权以及知识产权，均属各该资料之所有权人，并受相关著作权法或其他知识产权法律与条约之保护。

（3）、其他规定。

本版权说明适合中国（包括香港、澳门、台湾）法律。

# 软件安装与配置

## MySQL Server安装

1、下载安装包“mysql-5.6.34-linux-glibc2.5-i686.tar.gz”

# 安装依赖

yum -y install perl perl-devel autoconf

2、把下载的安装包移动到/usr/local/下。

3、解压

tar zxvf mysql-5.6.34-linux-glibc2.5-i686.tar.gz

4、复制解压后的mysql目录到系统的本地软件目录

cp mysql-5.6.33-linux-glibc2.5-x86\_64 /usr/local/mysql -r

5、添加系统mysql组和mysql用户

groupadd mysql

useradd -r -g mysql mysql

6、进入安装mysql软件目录，修改目录拥有者为mysql用户

cd mysql/

chown -R mysql:mysql ./

7、安装数据库，此处可能出现错误。

./scripts/mysql\_install\_db --user=mysql

8、修改当前目录拥有者为root用户

chown -R root:root ./

9、修改当前data目录拥有者为mysql用户

chown -R mysql:mysql data

10、启动mysql服务和添加开机启动mysql服务

添加开机启动，把启动脚本放到开机初始化目录。

cp support-files/mysql.server /etc/init.d/mysql

11、启动mysql服务

#创建缺少的文件夹

mkdir /var/log/mariadb

service mysql start

正常提示信息：Starting MySQL. SUCCESS!

通过使用 mysql -uroot -p 连接数据库（默认数据库的root用户没有密码，这个需要设置一个密码）。

错误信息：ERROR 2002 (HY000): Can't connect to local MySQL server through socket '/tmp/mysql.sock' (2)

解决方法：打开/etc/my.cnf,看看里面配置的socket位置是什么目录。“socket=/var/lib/mysql/mysql.sock”

路径和“/tmp/mysql.sock”不一致。建立一个软连接：ln -s /var/lib/mysql/mysql.sock /tmp/mysql.sock

12、采用MySQL客户端软件连接MySQL将itmonitor.sql导入，完成数据库itmonitor的创建和数据初始化。

## JDK8安装

1、 登录Linux，切换到root用户

su root 获取root用户权限，当前工作目录不变(需要root密码)，或sudo -i 不需要root密码直接切换成root（需要当前用户密码）。

2、 在usr目录下建立java安装目录

cd /usr

mkdir java

3、将jdk-8u121-linux-i586.tar.gz拷贝到java目录下

4、解压jdk到当前目录

tar -zxvf jdk-8u121-linux-i586.tar.gz，得到文件夹 jdk1.8.0\_121

5.安装完毕为他建立一个链接以节省目录长度

ln -s /usr/java/ jdk1.8.0\_121/ /usr/jdk

6.编辑配置文件，配置环境变量

vim /etc/profile

添加如下内容：JAVA\_HOME根据实际目录来

JAVA\_HOME=/usr/java/ jdk1.8.0\_121

CLASSPATH=$JAVA\_HOME/lib/

PATH=$PATH:$JAVA\_HOME/bin

export PATH JAVA\_HOME CLASSPATH

7.重启机器或执行命令 ：source /etc/profile

sudo shutdown -r now

8.查看安装情况

java -version

java version "1.8.0\_121"

Java(TM) SE Runtime Environment (build 1.8.0\_121-b13)

Java HotSpot(TM) Server VM (build 25.121-b13, mixed mode)

## Redis集群搭建

1.解压安装redis：

tar -zxvf redis-3.0.6.tar

mv redis-3.0.6 redis-3

cd redis

make&& make install

注意：

redis集群要最低必须是3个主或者3主3从(每一个文件夹对应一个端口)

mkdir -p /opt/redis/cluster/7000

mkdir -p /opt/redis/cluster/7001

mkdir -p /opt/redis/cluster/7002

mkdir -p /opt/redis/cluster/7003

mkdir -p /opt/redis/cluster/7004

mkdir -p /opt/redis/cluster/7005

cp /opt/soft/redis/redis/redis.conf /opt/redis/cluster/7000

vi /opt/redis/cluster/7000/redis.conf

修改下面几项：

daemonize yes //redis后台运行

pidfile /var/run/redis\_7000.pid //pidfile文件对应7000

port 7000 //端口7000

cluster-enabled yes //开启集群 把注释#去掉

cluster-config-file nodes.conf //集群的配置 配置文件首次启动自动生成

cluster-node-timeout 5000 //请求超时 设置5秒够了

appendonly yes //aof日志开启有需要就开启，它会每次写操作都记录一条日志

配置好了，就相应地把这个修改后的配置文件拷贝到 7001 7002目录，注意要修改监听端口port 和pidfile

2、到每一个文件夹启动redis

redis-server redis.conf

3、创建集群

前面已经准备好了搭建集群的redis节点，接下来我们要把这些节点都串连起来搭建集群。官方提供了一个工具：redis-trib.rb （/usr/local/redis/src/redis-trib.rb） 看后缀就知道这鸟东西不能直接执行，它是用ruby写的一个程序，所以我们还得安装ruby。

yum -y install ruby ruby-devel rubygems rpm-build

再用 gem 这个命令来安装 redis接口 gem貌似是ruby的一个工具包 反正没错就是了。

gem install redis

4、 确认所有的节点都启动，接下来使用参数create 创建

/opt/soft/redis/redis/src/redis-trib.rb create --replicas 1 127.0.0.1:7000 127.0.0.1:7001，127.0.0.1:7002 ，127.0.0.1:7003，127.0.0.1:7004， 127.0.0.1:7005

## Zookeeper集群搭建

由于zookeeper集群用于提供众多服务的监控配置，需要集群吞吐量、性能较优，所以需要根据需要调整集群大小。

1、预备

安装java环境，并配置好java相关的环境变量$JAVA\_HOME

2、下载3.4.9版并解压

tar -zxvf zookeeper-3.4.9.tar.gz

可执行文件都在bin/目录下，配置都在conf/目录下

3、生成配置

conf/目录下有个zoo\_sample.cfg，是样板配置文件

复制一份成zoo.cfg，里面有两个比较重要的配置：

dataDir=/var/lib/zookeeper # 数据存放位置，可根据需要修改

clientPort=2181 # 服务监听端口，可根据需要修改

4、启动服务

bin/zkServer.sh start

如果未启动成功，一个检查JAVA\_HOME是否正确设置，二个看端口是否冲突

5、启动客户端测试

启动命令：

bin/zkCli.sh -server 127.0.0.1:2181

显示根目录下、文件：

ls /

创建文件，并设置初始内容：

create /shenjian hello

获取文件内容：

get /shenjian

修改文件内容：

set /shenjian world

删除文件：

delete /shenjian

6、集群搭建

这里我使用三台主机进行Zookeeper的分布式配置。

（1）首先，我们在之前配置的单机基础上继续修改配置文件：

（2）在之前的配置文件末尾添加如下内容：

server.0=master01:2888:3888 //master01为你的主机名，前面的0代表的是给这台主机配的ID；

server.1=slave01:2888:3888 //同理slave01也是主机名，可以使用IP；

server.2=slave02:2888:3888

（3）然后在我们之前建立的data文件中使用命令：

echo 0 myid //在data目录建立myid文件，并写入0（与上面的配置中server.0中的0对应）

（4）将这台主机配置好的Zookeeper整个目录都拷贝到另外两台主机的相同路径下，并在那两台上修改myid为1和2。

（5）修改另外两台主机/etc/profile文件，使得它们的Zookeeper环境配置和第一台主机一样

（6）在每台主机上面开启Zookeeper后，查看状态

开启三台主机中Zookeeper后，查看状态会发现有一个leader以及两个follower，此时我们的Zookeeper分布式环境以及搭建好了。

## Kafka集群搭建

由于kafka的版本间差异较大，建议版本按照要求使用kafka\_2.10-0.8.2.2，避免因为kafka版本间差异、API变化、客户端差异引起的未知异常

1、kafka 集群安装环境准备

集群环境:192.168.139.130 master、192.168.139.131 node1、192.168.139.132 node2

JDK8以及zookeeper 集群。

2、创建kafka的安装目录

（1）mkdir /usr/local/kafka

（2）把kafka\_2.11-0.10.1.1.tgz移动到安装目录kafka解压缩

tar -zxvf kafka\_2.11-0.10.1.1.tgz

（3）配置集群文件

vi server.properties

#broker id,每个broker id在集群中必须唯一

broker.id=1

#监听端口

listeners=PLAINTEXT://:9091

#端口

port=9092

host.name=192.168.139.130

zookeeper.connect=192.168.139.130:2181,192.168.139.131:2181,192.168.139.132:2181/kafka

3、配置成功以后按顺序启动kafka,master->node1->node2

bin/kafka-server-start.sh config/server.properties &

4、启动成功后就是验证kafka集群安装是否成功

创建一个名称为my-replicated-topic5的Topic，5个分区，并且复制因子为3

bin/kafka-topics.sh --create --zookeeper 192.168.139.130:2181,192.168.139.131:2181,192.168.139.132:2181/kafka

--replication-factor 3 --partitions 5 --topic my-replicated-topic5

## Elasticsearch集群搭建

可以使用2.2、2.3、2.4几个版本，由于项目中需要用到groovy脚本功能，所以需要在elasticsearch集群配置过程中打开groovy脚本可运行开关。

1. 下载安装包，并解压

elasticsearch-2.4.1.tar.gz

tar –zxvf elasticsearch-2.4.1.tar.gz

2、启动

进入bin目录

$ ./elasticsearch –d

3、配置

（1）Client Node

ela-client.example.com:192.168.8.10(client node)

mv /opt/ elasticsearch-2.4.1 /config/elasticsearch.yml{,.default}

cluster.name: elasticsearch\_cluster

node.name: ${HOSTNAME}

node.master: false

node.data: false

path.data: /opt/ elasticsearch-2.4.1 /data

path.logs: /opt/ elasticsearch-2.4.1 /logs

bootstrap.mlockall: true

network.host: 127.0.0.1,192.168.8.10

http.port: 9200

index.refresh\_interval: 5s

script.inline: true

script.indexed: true

（2）Master Node

ela-master1.example.com:192.168.8.101(master node)

mv /opt/ elasticsearch-2.4.1/config/elasticsearch.yml{,.default}

cat >/opt/ elasticsearch-2.4.1/config/elasticsearch.yml <<HERE

cluster.name: elasticsearch\_cluster

node.name: ${HOSTNAME}

node.master: true

node.data: false

path.data: /opt/ elasticsearch-2.4.1/data

path.logs: /opt/ elasticsearch-2.4.1/logs

bootstrap.mlockall: true

network.host: 127.0.0.1,192.168.8.101

http.port: 9200

index.refresh\_interval: 5s

script.inline: true

script.indexed: true

## Storm 集群搭建

storm版本间存在较大差异，建议使用0.10.0。

1、安装JDK安装jdk8。

不管你用什么方法，最后需要配置JAVA\_HOME并检测当前jdk8版本：

Java HotSpot(TM) 64-Bit Server VM (build 20.6-b01, mixed mode)

2、安装Zookeeper集群。

3. 安装Python。

4、安装Storm

下载apache-storm-0.10.0.tar.gz，并解压:

tar –zxvf apache-storm-0.10.0.tar.gz

接下来，配置环境变量：

export STORM\_HOME=/usr/lib/storm

export PATH=$PATH:$STORM\_HOME/bin

建立storm存储目录：

$ mkdir /tmp/storm

修改配置文件/usr/lib/storm/conf/storm.yaml，修改为如下：

最后配置完毕后：

########### These MUST be filled in for a storm configuration

storm.zookeeper.servers:

- "master"

- "slave1"

- "slave2"

storm.local.dir: "/data/storm/data"

nimbus.host: "master"

supervisor.slots.ports:

- 6700

- 6701

- 6702

- 6703

5、启动

以下是启动Storm各个后台进程的方式：

（1） Nimbus: 在Storm主控节点上运行 "storm nimbus &" 启动Nimbus后台程序，并放到后台执行；

（2）Supervisor: 在Storm各个工作节点上运行" storm supervisor &"启动Supervisor后台程序，并放到后台执行；

（3）UI: 在Storm主控节点上运行 "storm ui &" 启动UI后台程序，并放到后台执行，启动后可以通过http://192.168.8.94:8080/index.html观察集群的worker资源使用情况、Topologies的运行状态等信息。

## Tomcat 安装

1、安装JDK8，并设置好环境变量

JAVA\_HOME=/usr/local/services/java/jdk1.8.0\_121

CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar

PATH=$PATH:$JAVA\_HOME/bin

export JAVA\_HOME

export CLASSPATH

export PATH

2、下载安装tomcat（http://tomcat.apache.org/）

[root@master ~]# unzip apache-tomcat-9.0.0.M17.zip

[root@master ~]# mv apache-tomcat-9.0.0.M17/ /usr/local/

[root@master ~]cd /usr/local/

[root@master local]# ln -s /usr/local/ apache-tomcat-9.0.0.M17/ /usr/local/tomcat

[root@master local]# cd tomcat/bin/

[root@master bin]#ls

[root@master bin]#vim catalina.sh

添加以下内容：

CATALINA\_HOME=/usr/local/ apache-tomcat-9.0.0.M17/

[root@master local]#chmod +x \*.sh

3、启动tomcat服务器

[root@master tomcat]# /usr/local/tomcat /bin/catalina.sh start

Using CATALINA\_BASE: /usr/local/ apache-tomcat-9.0.0.M17/

Using CATALINA\_HOME: /usr/local/ apache-tomcat-9.0.0.M17/

Using CATALINA\_TMPDIR: /usr/local/ apache-tomcat-9.0.0.M17//temp

Using CLASSPATH: /usr/local/ apache-tomcat-9.0.0.M17//bin/bootstrap.jar

[root@master logs]# cd /usr/local/tomcat/logs/

[root@master logs]# tail -f catalina.out

4、在浏览器中输入

http://localhost:8080/（如果不是本机，则输入对应的ip地址）

测试出现tomcat页面则测试成功

1)webapps文件夹主要用于web应用程序部署，比如你可以把你的应用程序包，如war文件拷到该目录下，容器会自动部署。

　　2)conf文件夹下主要是放置tomcat的服务器的相关配置文件

## 微服务监控系统部署

以上本系统所依赖的运行支撑软件安装部署完成后，进行本系统的安装部署，将如下4个war包发布到tomcat的webapps下，重新启动Tomcat即可完成系统的部署。

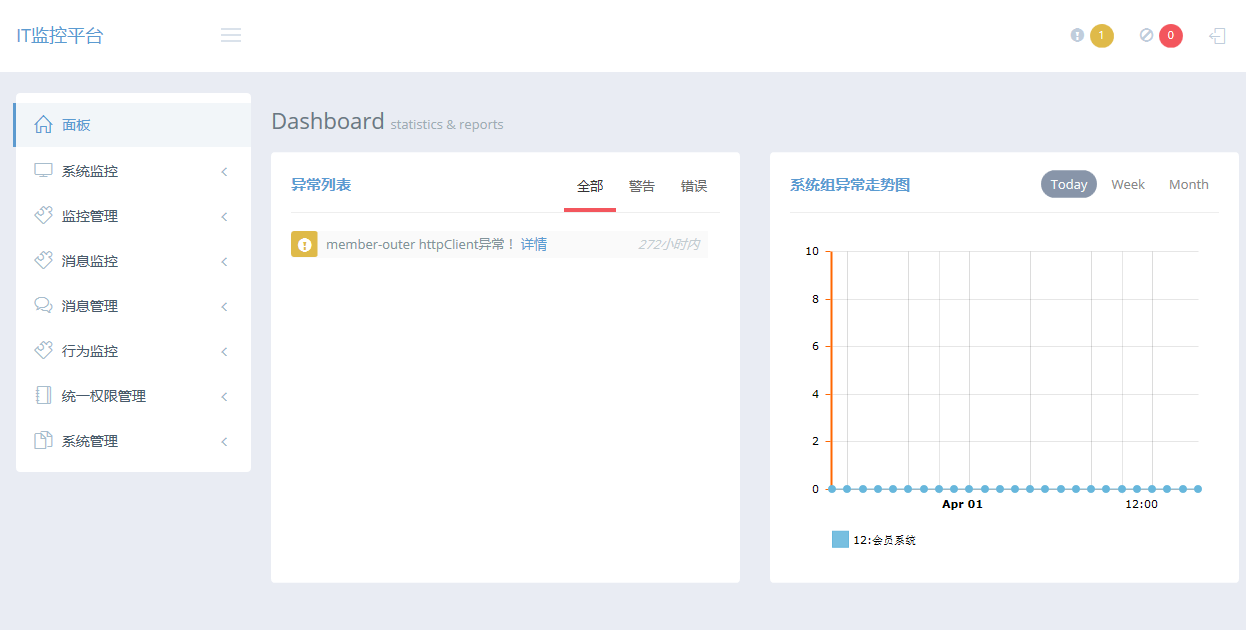
common-service-1.2.0-SNAPSHOT.war

console-module-main-1.2.0-SNAPSHOT.war

exception-service-1.2.0-SNAPSHOT.war

monitor-service-1.2.0-SNAPSHOT.war

启动浏览器在地址栏中输入如下地址<http://192.168.10.95:8081/console/>即可看到本系统的登陆页面，登陆后显示系统的主界面：



# 软件功能介绍与使用说明

数字创新微服务监控系统1.0采用大数据分析手段，提供整合基础设施组件的日志监控工具、系统资源监控、实例管理、应用管理、请求监控统计、数据源访问监控统计、行为监控，实时报警等功能。

## 监控面板



监控面板主要以图表方式显示异常列表和系统组异常走势图。

## 系统管理

采用大数据实时分析手段，根据预先配置的系统响应时间、实例节点判活、系统异常、系统错误的阈值，实时数据上报并通过短信及邮件的通知系统相关负责人，及时响应处理，真正做到7\*24小时监控告警。

### 用户管理



系统管理员可以对用户进行管理和维护，主要功能有用户添加、用户删除、用户信息编辑和用户添加以及用户密码重置等。

**用户信息编辑**



### 菜单管理

系统管理员可进行系统菜单的配置与维护，主要功能有菜单项新建、菜单项信息浏览、菜单项编辑、菜单项删除等功能。



**菜单创建**



**子菜单列表**

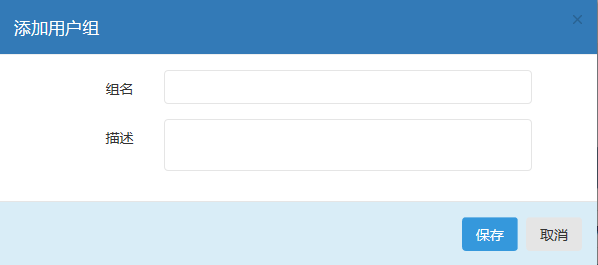


### 用户组管理

系统管理员可以进行用户组的管理与维护，主要功能包括用户组列表、用户组创建、用户组信息维护、用户组删除、用户组成员维护等功能。



**创建用户组**



**添加用户组成员**



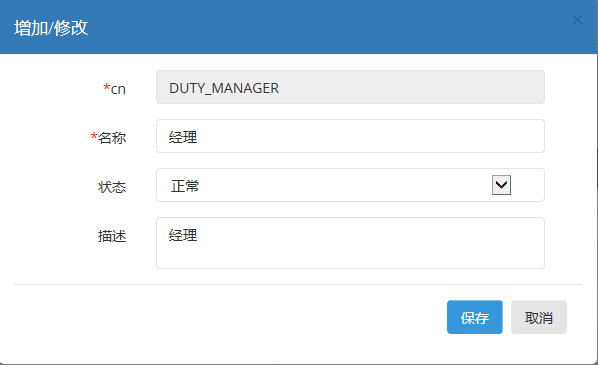
## 统一权限管理

### 职务信息管理

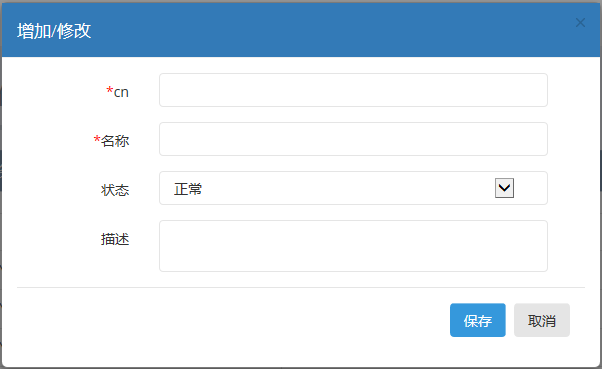
系统管理员可以进行职务信息的维护，主要功能包括，职务创建、职务信息维护、职务删除等。



**添加职务信息**



**修改职务信息**



### 组织信息管理

系统管理员可以进行组织信息的管理与维护，主要功能包括组织信息的创建、编辑、删除等。



**添加组织信息**

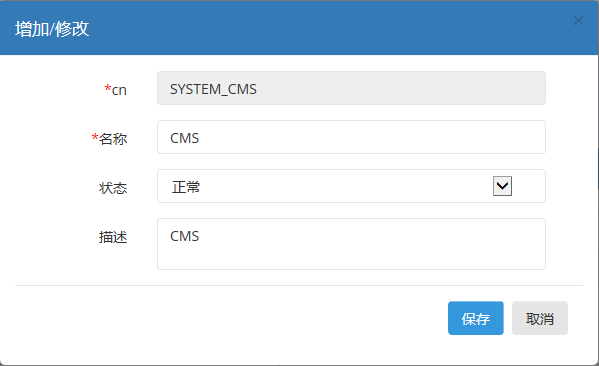


### 系统信息管理

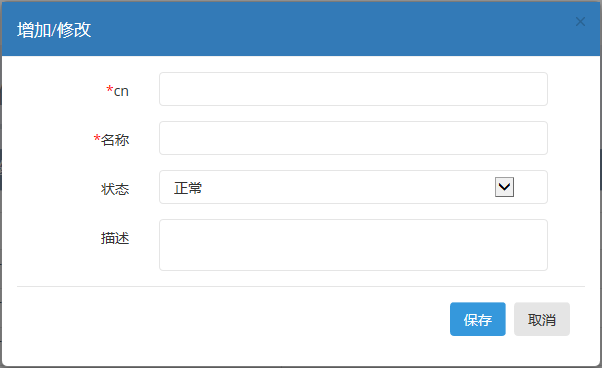
系统管理员可以进行系统信息的管理与维护，主要功能包括系统信息的创建、信息维护、系统信息删除以及角色关联等。



**添加系统信息**



**修改系统信息**



**添加关联角色**

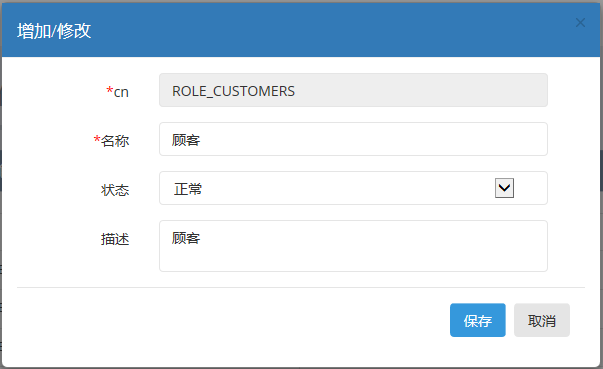


### 角色信息管理

系统管理员可以进行角色信息的管理与维护，主要功能包括角色创建、角色删除、角色信息维护以及用户关联等。



**修改角色信息**

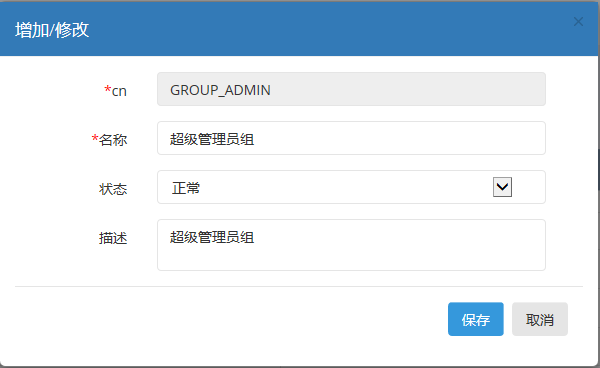


### 小组信息管理

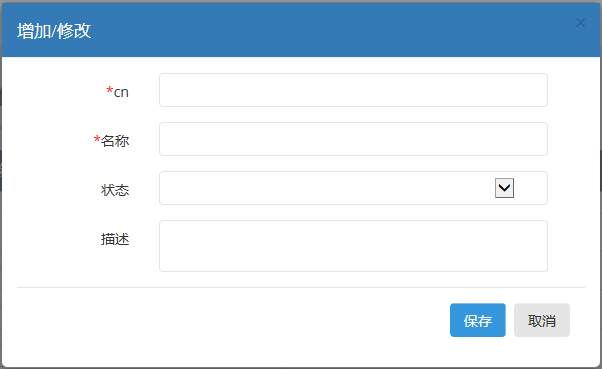
系统管理员可以进行小组信息的管理与维护，主要的功能有小组信息创建、小组信息删除、小组信息维护、关联用户等。



**编辑小组信息**



**添加小组信息**



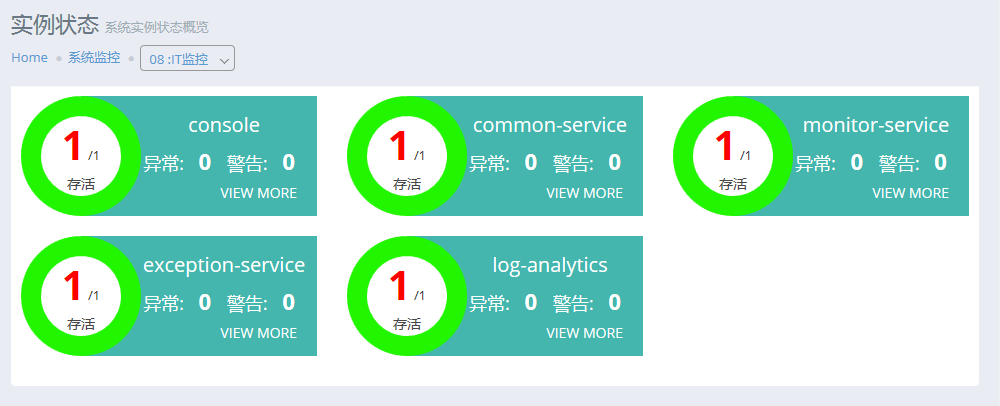
## 系统监控

采用大数据分析手段，提供整合基础设施组件的日志监控工具、系统资源监控、实例管理、应用管理、请求监控统计、数据源访问监控统计、行为监控，实时报警等功能。

### 实例状态

实例状态主要是以图表形式只管展示本系统所监控的各应用实例及其运行服务器的状态，包括应用实例存活数、异常及警告。

**实例概况**



**异常情况概览**



可以以图表方式直观的展示业务警告或者业务异常的状态及发生时间

**业务异常列表**



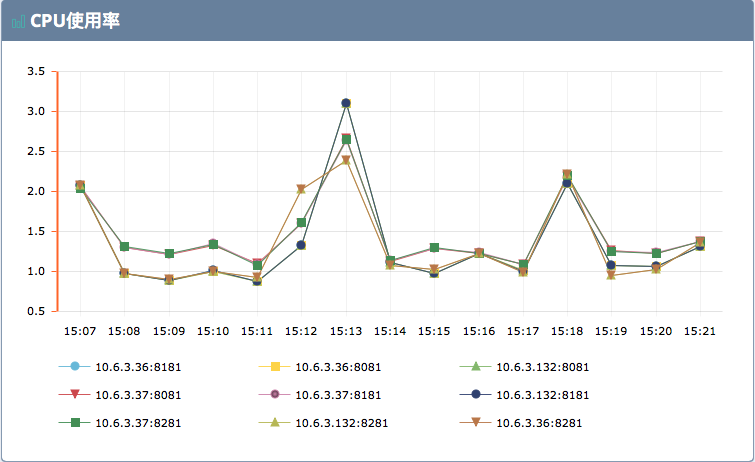
应用业务异常列表会以业务码的形式，实时展示出当前应用存在哪些异常和警告，以及异常和警告的处理率。

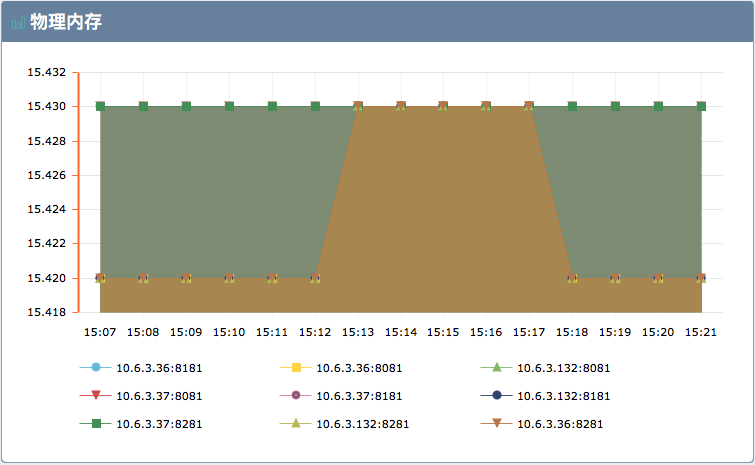
**应用实例监控**

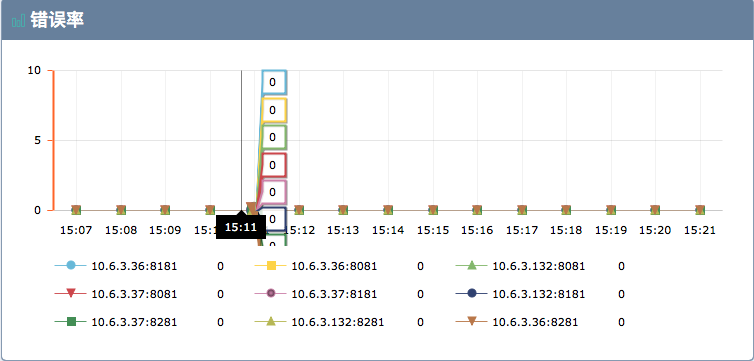


应用实例监控实时监控应用实例的当前响应时间、平均响应时间、吞吐率、错误率、CPU利用率、内存使用量等，并以图文报表的形式展示。











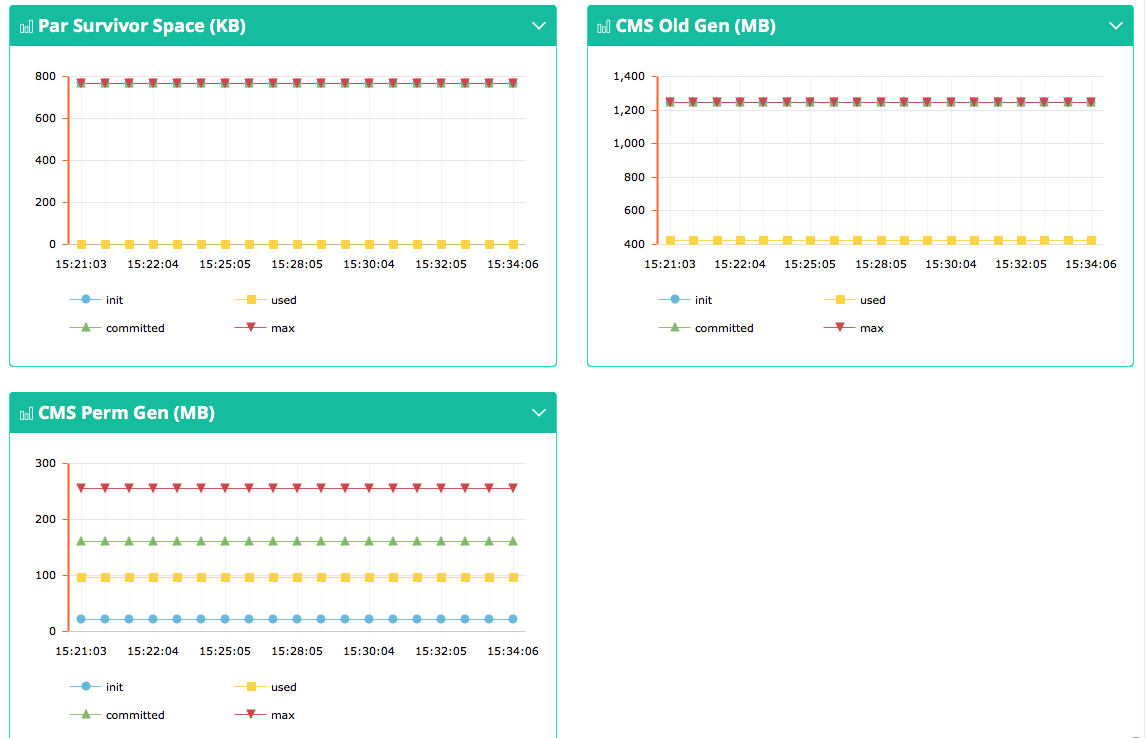
**应用实例节点监控**



应用实例节点监控会监控具体到应用的某一台具体ip的实例，针对其JVM内存使用量、线程数、数据源、实例明细、请求耗时等进行监控分析，为开发人员提供对JVM参数调优，数据库慢sql监控调优提供有效手段。







**SQL慢查询TOP N监控**



实时展现最近系统SQL慢查询TOP N，方便系统数据库（SQL）调优。

**实时报警**



### 异常信息查询

**异常信息查询**



**异常信息浏览**



## 监控管理

### 系统管理

系统管理包括系统组信息维护与管理，包括系统组创建、编辑、删除、系统添加等功能。



**系统组添加**



**系统组信息维护**



**添加系统**



### 预警配置管理

预警配置管理主要是对本平台所监控的各系统异常报警阀值的设置，包括警告数最小值、警告数最大值、异常数阀值最大值、实例节点阀值最小值、实例节点阀值、响应时间最小值、响应时间最大值、异常、警告开启状态、节点数预警开关等。



### 系统组-用户组管理

系统组-用户组管理主要是进行与系统组相关的用户组维护。包括用户组的创建、用户添加、用户删除等功能。



**添加用户组**



### 业务管理

业务管理主要是对本平台所监控的各类业务进行维护，包括业务信息查询、业务信息编辑、业务添加和业务删除等功能。



**业务添加**



**业务信息编辑**



### 短信配置管理

用于与业务相关的短信接口配置，包括短信接口用户名、短信接口密码、短信产品ID、短信发送地址、使用状态等。主要功能有短信配置信息添加、编辑与删除等功能。



**添加业务短信配置信息**

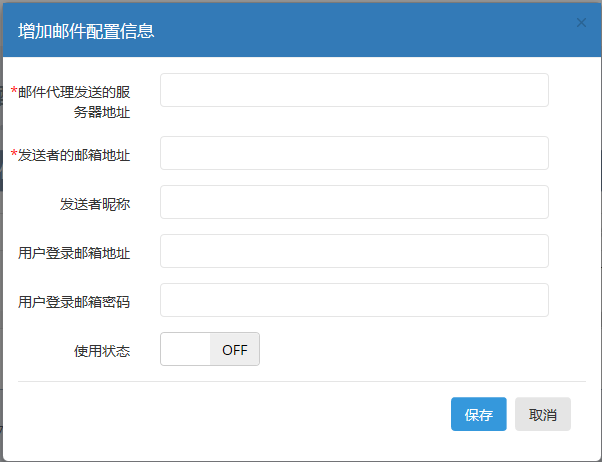


### 邮件配置管理

用于用户邮件接口信息的配置，包括mail代理发送的地址、发送者的邮箱地址、发送者昵称、户登录邮箱地址、用户登录邮箱密码、使用状态。主要功能有邮件接口信息的添加、删除、编辑等功能。



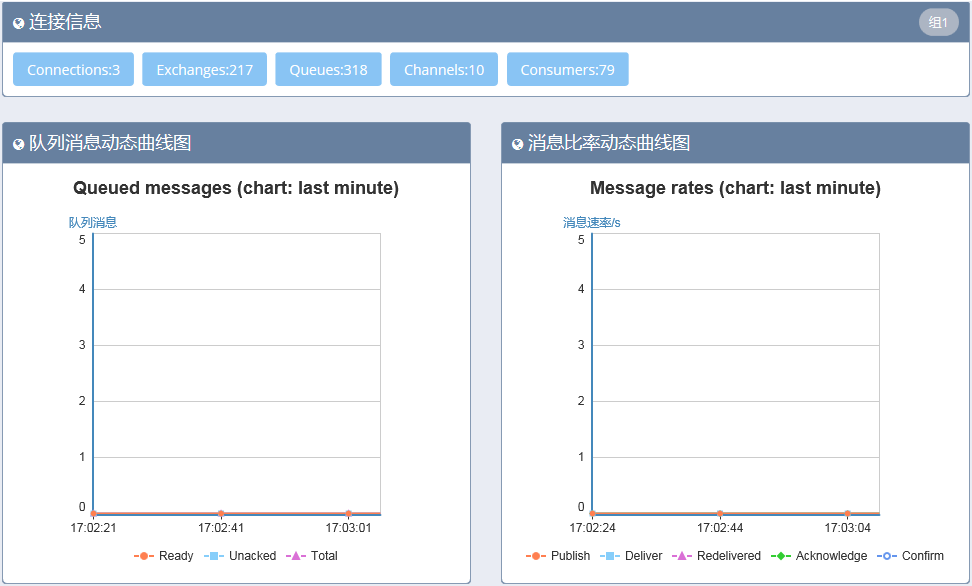
**新增邮件配置信息**



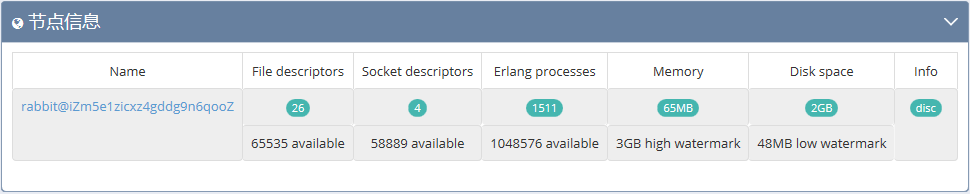
## 消息监控

### MQ总体概览图

总体展示MQ的运行状况。包括连接信息，消息队列状况以及节点信息等。

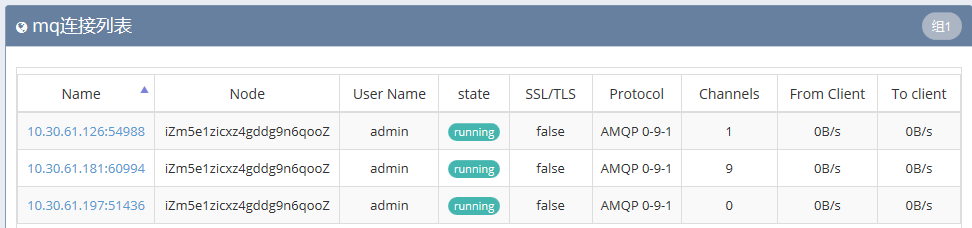


**节点信息**



### MQ连接信息

以图表形式展示消息队列的连接队列和状态。



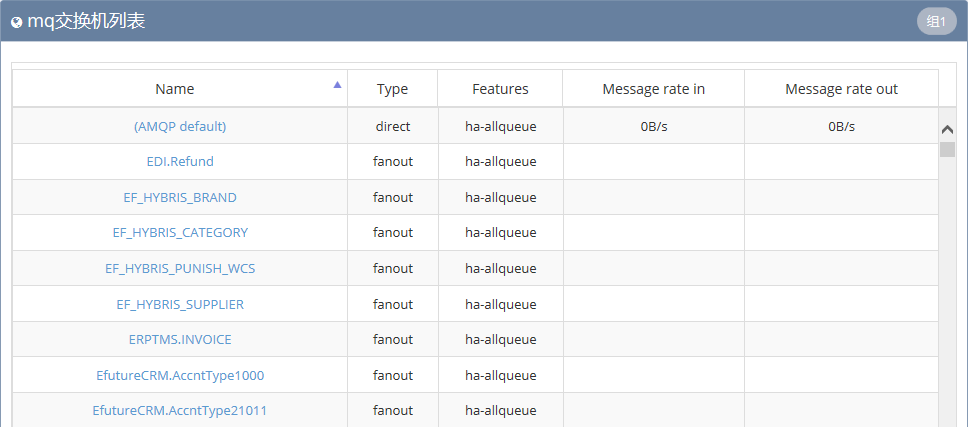
### MQ队列信息

展示消息队列状态和各项参数，包括队列状态、消息数、等待数、未收到消息数、持久化、自动删除、占用内存等。



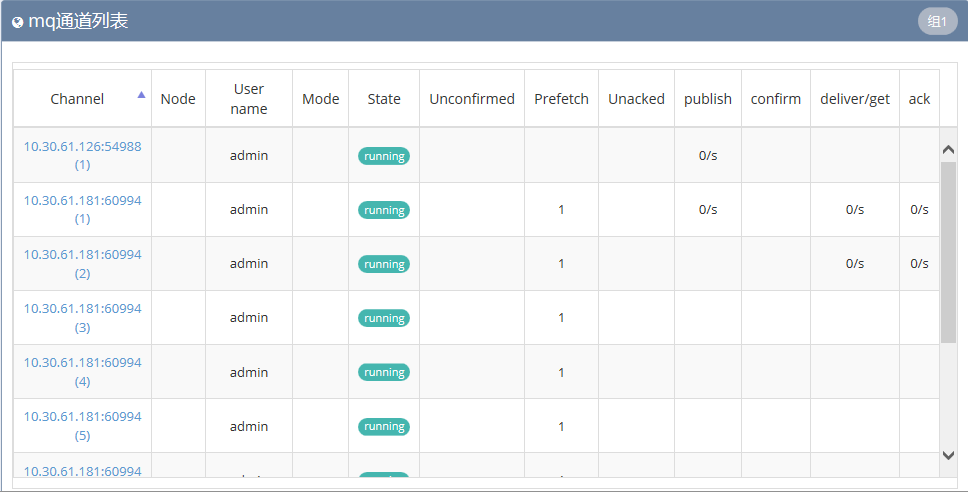
### MQ交换机信息

以表格形式展示MQ交换机的状态和各项参数。



### MQ通道

以表的方式展示MQ通道信息及各项参数。



### MQ消息跟踪

进行MQ消息的查询与跟踪，包括消息号、消息内容、发送方编号、消息状态（全部、发送中...、消息已到达、重试发送中...、发送失败）、接收方编号、查询归档等消息状态。



## 消息管理

### 集群管理

主要用于集群信息的管理与维护，包括消息集群的各项状态和参数以及消息集群的添加、编辑和删除。



**集群信息编辑**

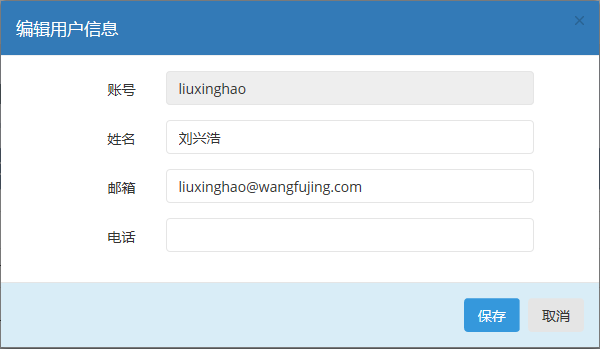


### 用户管理

主要进行MQ集群用户的管理与维护，主要功能包括用户信息维护、用户添加、用户同步、用户删除等。



**用户维护**



**用户同步**



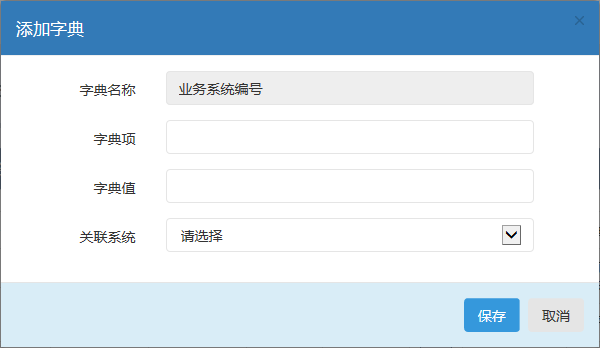
### 数据字典

主要用于业务系统字典维护和业务代码字典维护。包括各项代码的创建和删除功能。

**业务系统字典**



**业务系统编号添加**



**业务代码**



**业务代码添加**



### 接出服务注册

该项功能主要用于接出服务的注册，包括接出服务的描述以及添加新的接出服务。



**添加接出服务**



### 队列配置

用于消息队列的配置。



**队列信息编辑**

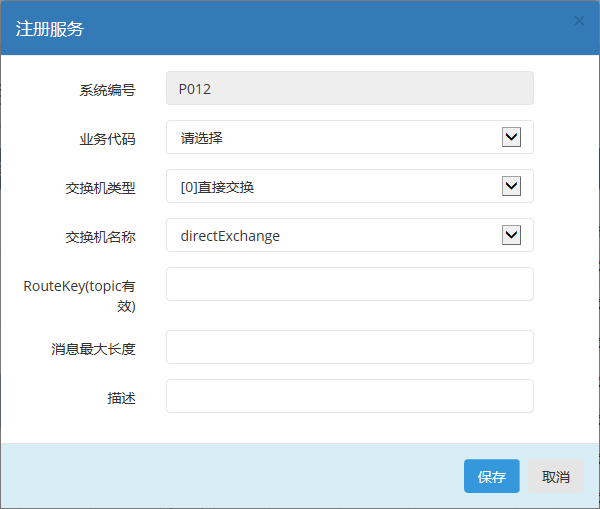


### 介入服务注册

用于接入服务的注册与配置。



添加注册服务

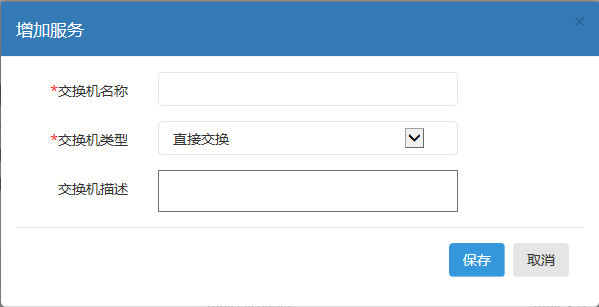


### 交换机配置

配置消息队列交换机。



**添加新的交换机**

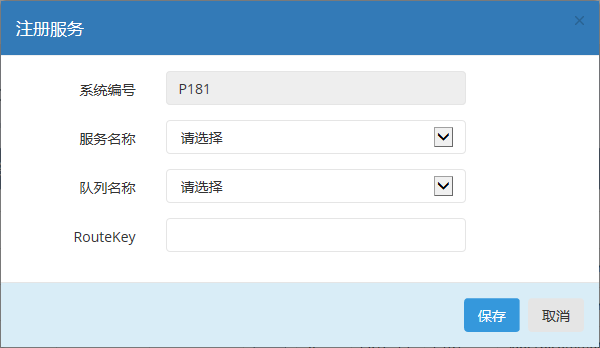


### Direct方式注册

以Direct方式 注册服务。



**添加注册服务**

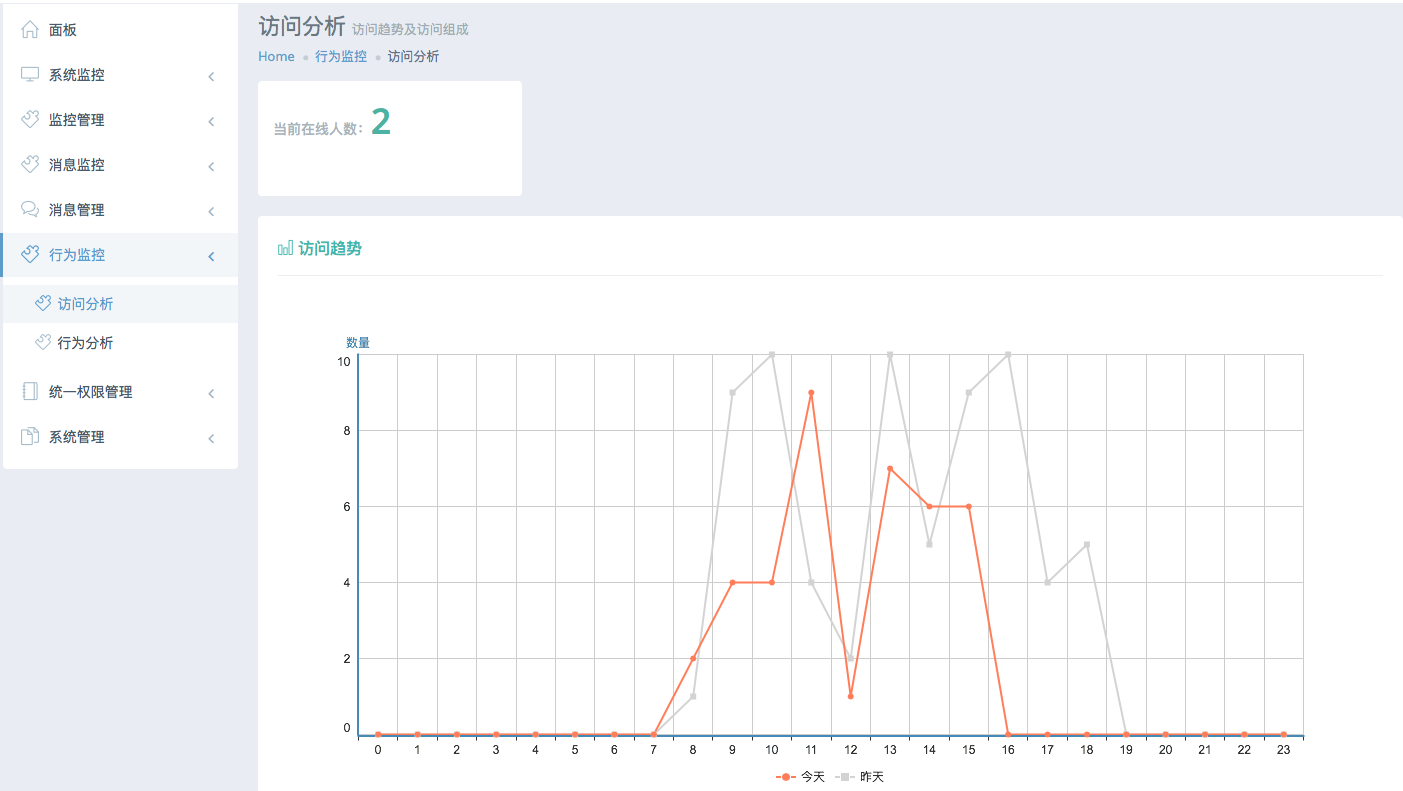


## 行为管理

行为监控可以对公司对内对外系统进行实时监控分析，可以对系统功能使用情况如系统访问趋势、访问热度、访问组成等进行分析。

### 访问分析

访问分析主要是对访问趋势和访问组成进行分析。



### 行为分析

行为分析主要是对用户的基本信息进行统计和分析。



