# 组件简介

## 组件功能介绍

DSF动态流控功能包含在commons.configuration.jar中，他主要提供了以下能力：

流控的服务及方法的初始值在dsf.dynamic.config.properties里面配置。后续可以在不中断系统运行的情况下，通过修改Zookeeper上对应节点的值来动态修改，实时生效。

# 涉及到的外部组件

## 前置条件

Zookeeper服务启动正常

DSF服务发布正常

# 配置文件和API

## 配置文件

### DSF服务发布的XML文件

以sub组件的subscribe.monitor.DSFHealthService服务为例。

#### 服务级流控：

<dsf:exporter name="subscribe.monitor.DSFHealthService"

interface="com.huawei.jaguar.commons.monitor.client.dsf.DSFHealthService" ref="ereading.subscribe.subDsfHealthCheck

executes="${subscribe.monitor.DSFHealthService.executes}">

</dsf:exporter>

#### 方法级流控：

<dsf:exporter name="subscribe.monitor.DSFHealthService" interface="com.huawei.jaguar.commons.monitor.client.dsf.DSFHealthService" ref="ereading.subscribe.subDsfHealthCheck">

<dsf:method name="checkStatus" executes="${subscribe.monitor.DSFHealthService.checkStatus.executes}" />

</dsf:exporter>

#### 说明

红色部分的${subscribe.monitor.DSFHealthService.executes}，${subscribe.monitor.DSFHealthService.checkStatus.executes}要和dsf.dynamic.config.properties里面的配置保持一致

### dsf.dynamic.config.properties配置文件

#### dsf.dynamic.config.properties文件路径

在dsf.properties同级目录增加一个dsf.dynamic.config.properties

#### dsf.dynamic.config.properties配置说明

#服务级别流控配置示范，红色部分为dsf服务名。.executes为固定部分，

#0是流控值，流控值不配置默认是0

subscribe.monitor.DSFHealthService.executes=0

#方法级别流控配置示范，红色部分为dsf服务名，蓝色部分为方法名称。.executes为固定部分，

#0是流控值，流控值不配置默认是0

subscribe.monitor.DSFHealthService.checkStatus.executes=0

#### 全局默认流控配置（不需要可以不关注）

可以配置一个全局的默认流控值

1. 在dsf.dynamic.config.properties文件增加如下配置项

executes.default=0

2.如果第一步没有配置，则默认为0

### spring bean配置

#### 配置示范

需要加入一个如下的spring bean配置。红色部分是以sub组件为示范。

<bean id=*"DsfFlowControlBean"*

class=*"com.huawei.commons.configuration.spring.DsfFlowControlBean"*

init-method=*"initConnectionListener"*>

<property name=*"connectionName"*>

<value>subscribe.configerZKClient</value>

</property>

<property name=*"flowControlService"*>

<value>/config/sub/flowcontrol/service</value>

</property>

<property name=*"flowControlMethod"*>

<value>/config/sub/flowcontrol/method</value>

</property>

</bean>

#### 参数说明

1.subscribe.configerZKClient：对应了\*.zk-cluster.xml文件的name节点值



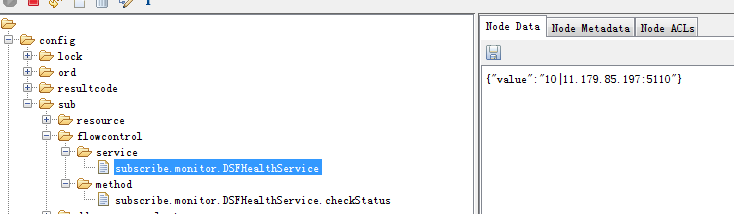
2./config/sub/flowcontrol/service：

服务级监控路径，该节点的每一个子节点对应了一个服务级的流控配置

注意子节点名称所对应的服务名不要加executes

3./config/sub/flowcontrol/method：

方法级监控路径，该节点的每一个子节点对应了一个方法级的流控配置



4.服务级监控路径和方法级监控路径要是平级关系，不能前缀包含

## 使用介绍

下面分别介绍统一配置组建的使用

### 组件的启动

组件启动时候，

1.DSF服务会自动根据executes="${subscribe.monitor.DSFHealthService.executes}">去dsf.dynamic.config.properties找到对应的流控值 进行发布。

2.会在ZK上创建对应的监控节点

节点的监控路径由spring bean 的*flowControlService和flowControlMethod两个配置项决定*

系统会自动监控这两个节点下的所有子节点的增加，删除和修改事件。

当修改和增加事件会调用DSF的流控API去修改流控值，同时更新dsf.dynamic.config.properties。

当发生删除事件仅仅记录一下日志，什么也不做。

### 流控值的动态修改

**1.服务级流控的修改**

组件自己选择命令方式或者可视化的ZK客户端工具，

在/config/sub/flowcontrol/service节点下新增一个节点：节点的名称为服务名;对应的值为一个json字符串，包括了流控值和不需要流控的IP列表。

修改该节点的值会动态调整DSF内存中的流控值，并把新的流控值写入dsf.dynamic.config.properties对应的配置项。后续即使组件重启，流控值依然有效。

配置示范：

服务名：subscribe.monitor.DSFHealthService

对应的值：{"value":"10|11.179.85.197:5110,12.179.85.197:5110"}

**2.方法级流控的修改**

组件自己选择命令方式或者可视化的ZK客户端工具，

在/config/sub/flowcontrol/method节点下新增一个节点：节点的名称为服务名.方法名;对应的值为一个json字符串，包括了流控值和不需要流控的IP列表。

修改该节点的值会动态调整DSF内存中的流控值，并把新的流控值写入dsf.dynamic.config.properties对应的配置项。后续即使组件重启，流控值依然有效。以达到永久修改流控值的效果

配置示范：

服务名：subscribe.monitor.DSFHealthService.checkStatus

对应的值：{"value":"10|11.179.85.197:5110,12.179.85.197:5110"}

**参数说明**

{"value":"10|11.179.85.197:5110,12.179.85.197:5110"}

**a.10**

流控值,指的是并发数，表示最多可以同时处理多少个请求。达到上限后，再过来的请求会被服务端拒绝

**b.11.179.85.197:5110,12.179.85.197:5110**

例外的IP列表：IP:port, IP:port, IP:port 。IP之间 以英文逗号分隔。在IP列表中的节点不使用流控

**c.如果没有例外的IP列表，直接写\*。形如**

{"value":"10|\*"}

### 确认是否修改成功

可以看一下dsf.dynamic.config.properties，找到对应的流控值，看看是否修改成功

## 使用场景

【注意】优先级来说：方法级 > 服务级

对于能力中心基线部分的DSF服务全部配到了方法级别，不再使用服务级。

对于新的DSF服务:发布的时候。可以参考下面的使用场景自行决定如何配置dsf.dynamic.config.properties

现在有几种场景：

#### 服务级流控

目标：对该服务下的所有方法生效。

约束：dsf.dynamic.config.properties只设置服务级配置，不能设置方法级配置

#### 方法级流控

目标：对指定服务的某一个方法使用流控

约束:   dsf.dynamic.config.properties只配置该方法的流控值

#### 服务级和部分方法级

目标：对某一个服务下的某些方法使用方法级流控，其他使用服务级流控

约束：dsf.dynamic.config.properties同时配置服务级流控和需要使用流控的方法

**综上所述：组件在使用的时候，根据自己的需要设置好配置文件的流控值。然后ZK上可以动态调整配置文件里面的流控值**

# Maven依赖

场景样例为组件插件的一类功能实现过程，包括场景描述、开发过程。

## 依赖说明

1.动态流控的功能是基于DSF **1.1.00.SPC500版本.**低于此版本不支持。

2.Commons组件规划了一个commons.denpengcy-management工程来规范所有基于commons组件开发jar包的使用，为了防止jar包冲突以及参考公司的优选库，commons.denpengcy-management里面的pom.xml定义了所有优选的jar和对应的版本。请尽量使用commons.denpengcy-management里面的jar。如果确实需要使用不在commons.denpengcy-management范围内的jar。请将jar和对应版本反馈给（朱元庆z00278802）

## 依赖示例

<properties>

<commons.version> V300R003C60B603-SNAPSHOT</commons.version>

</properties>

<dependencyManagement>

<dependencies>

<dependency>

<groupId>com.huawei.jaguar</groupId> <artifactId>commons.denpengcy-management</artifactId>

<version>${commons.version}</version>

<type>pom</type>

<scope>import</scope>

</dependency>

</dependencies>

</dependencyManagement>

<dependencies>

<dependency>

<groupId>com.huawei.jaguar</groupId>

<artifactId>commons.configuration</artifactId>

</dependency>

</dependencies>

# 附件

## 一些示范配置文件





# FAQ

插件过程中常见问题。