业务接入说明：  
flasher基于redis3集群服务器，java客户端基于jedis，现提供功能如下：  
1、redis3集群的客户端操作(兼容gcache)；  
2、调用客户端方法的指标监控(使用次数及使用时长)，监控方式目前提供2种：1.HTTP被动请求；2.HTTP主动上报；  
  
目前gedis-java客户端spring配置，具体配置如下：  
1、使用gedis客户端的redis3集群服务  
 1.1.1 方式一 <bean id="redisClusterConnectionFactory"  
 class="RedisClusterConnectionFactory">  
 <!--<property name="hostPorts">-->  
 <!--<set>-->  
 <!--<value>10.58.47.100:7010</value>-->  
 <!--<value>10.58.47.100:7011</value>-->  
 <!--<value>10.58.47.100:7012</value>-->  
 <!--<value>10.58.47.101:7010</value>-->  
 <!--<value>10.58.47.101:7011</value>-->  
 <!--<value>10.58.47.101:7012</value>-->  
 <!--</set>-->  
 <!--</property>-->  
 <property name="hostPort" value="10.58.47.100:7010,10.58.47.100:7011,10.58.47.100:7012,10.58.47.101:7010,10.58.47.101:7011,10.58.47.101:7012" />  
 </bean>  
 1.1.2 方式二:提供虚ZK的IP,自动获取redis IP组  
   
 <bean id="redisClusterConnectionFactory"  
 class="RedisClusterConnectionVHFactory">  
 <property name="hostPort" value="127.0.0.1:2181" /> <!--hostPort 为虚 ZK IP, -->  
 <property name="jedisPoolConfig" ref="jedisPoolConfig" />  
 </bean>  
   
 1.2   
 <bean id="redisCluster"  
 class="RedisClusterImpl">  
 <constructor-arg index="0" ref="redisClusterConnectionFactory" />  
 </bean>  
 总结   
 redis集群组可采用3种方式注入：  
 1.hostPorts，Set类型,使用(RedisClusterConnectionFactory 类)；使用RedisClusterConnectionVHFactory 属性为hostPort 这样可以提供虚IP访问redis集群(详细可参考test/resources/applicationContext-VH.xml)  
 2.hostPort，String类型。若想自定义连接池属性，增加如下配置：  
 在bean id="redisClusterConnectionFactory"中增加<property name="jedisPoolConfig" ref="jedisPoolConfig" />，增加bean注入  
 <bean id="jedisPoolConfig" class="redis.clients.jedis.JedisPoolConfig">  
 <property name="maxTotal" value="500" />  
 <property name="maxIdle" value="100" />  
 <property name="minIdle" value="10" />  
 <property name="maxWaitMillis" value="2000" />  
 <property name="testOnBorrow" value="true" />  
 </bean>  
 JedisPoolConfig的详细参数可参看属性。  
   
  
2、flasher-java客户端提供调用方法的监控，  
 监控服务spring配置如下：  
 <!-- 1-监控拦截器 -->  
 <bean id="monitorInterceptor" class="MonitorInterceptor" />  
  
 <!-- 2-配置拦截器代理 -->  
 <bean id="autoProxyCreator" class="org.springframework.aop.framework.autoproxy.BeanNameAutoProxyCreator">  
 <!-- 设置目标对象 -->  
 <property name="beanNames">  
 <list>  
 <value>redisCluster</value>  
 </list>  
 </property>  
 <!-- 代理对象所使用的拦截器 -->  
 <property name="interceptorNames">  
 <list>  
 <value>monitorInterceptor</value>  
 </list>  
 </property>  
 </bean>  
  
 监控数据获取目前有2种：  
 1.HTTP被动请求，配置web.xml如下：  
 <servlet>  
 <servlet-name>monitoring</servlet-name>  
 <servlet-class>MonitorService</servlet-class>  
 </servlet>  
 <servlet-mapping>  
 <servlet-name>monitoring</servlet-name>  
 <url-pattern>/monitoring</url-pattern>  
 </servlet-mapping>  
 暴露出来的url地址为/monitoring，监控服务会定时请求并记录指标数据，(监控系统参见其他说明)，清除监控数据目前提供2种方式：  
 1.自动清除，在<servlet>中配置  
 <init-param>  
 <param-name>AUTO\_CLEAR\_TIME</param-name>  
 <param-value>60</param-value>  
 </init-param>  
 <load-on-startup>1</load-on-startup>  
 AUTO\_CLEAR\_TIME为自动清除时间，单位为秒(s)  
 2.手动清除，每次监控系统请求url时增加HAND\_CLEAR参数，值为true  
 bean id="monitorService"中的url为主动上报的监控服务地址，protocol为监控服务支持的协议类型，目前监控服务采用G-Falcon，监控数据协议为FalconProtocol，MonitorService还有其他属性，详细见代码。  
 bean id="protocol"为协议注入，目前支持G-Falcon，其中参数endpoint为监控服务实例别名，唯一，必须。