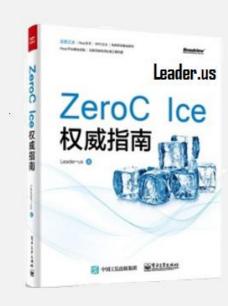
# Zeroc Ice微服务架构企业应用实践指南

八: Mycat Ice企业框架(下)





主讲老师: Leader.us 联系QQ:719867650

Leader.us高端架构师精品系列课程

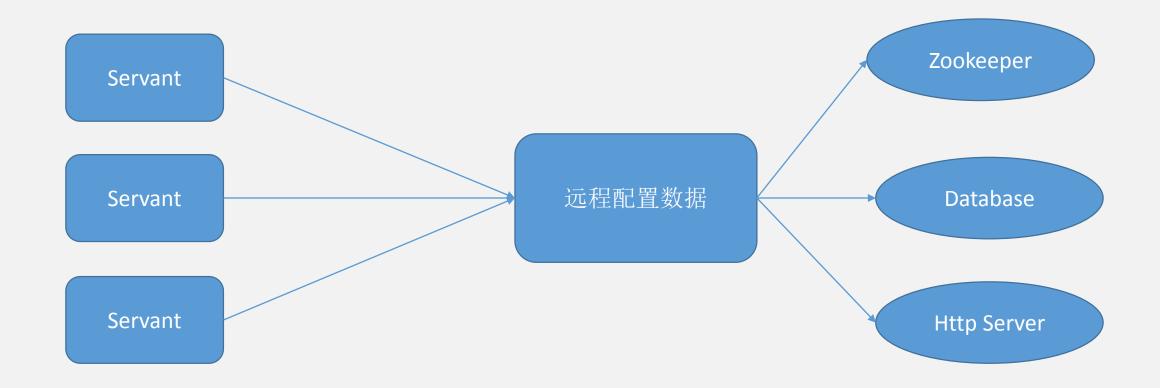
#### 本集视频主要内容

- 配置集中化的框架代码
- 远程动态加载服务代码
- Servant生命周期接口



**产果学院** 主讲老师: Leader.us roncoo.com 联系QQ:719867650

# 配置集中化的思路





**产果学院** 主讲老师: Leader.us oncoo.com 联系QQ:719867650

# 配置集中化的框架代码

ZookeeperStore ConfigTool.getPropertiesHolder(String phld,String defaultphld) → PropertiesStore ConfigTool DataBaseStore PropertiesHolder Client App HttpServerStore

PropUpdatedHandler



#### 远程动态加载服务代码



程序中用的Jar包分为公共部分以及每个Servant私有的两种,存放在一个Web Server里,各个节点都可以访问



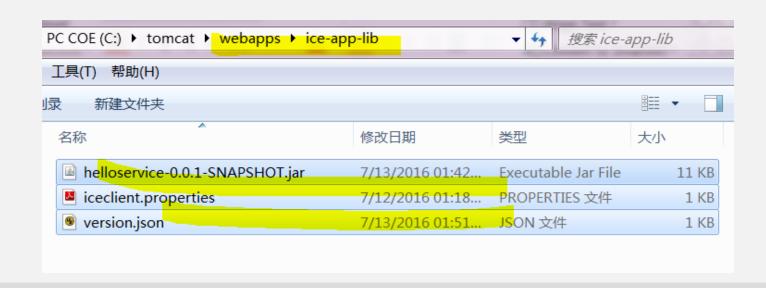
#### Grid.xml配置

```
cproperties id="LoadJarProps">
   cproperty name="LoadJarsFromRemote.Enabled" value="true"/>
   property name="LoadJarsFromRemote.SharedJars" value="iceclient.properties">
   property
name="LoadJarsFromRemote.AutoUpdate" value="true">
</properties>
<server-template id="MyHelloServerTemplate">
   <parameter name="id" />
   <icebox id="MyHelloServer${id}" exe="java" activation="on-demand">
      properties>
          properties refid="props" />
      </properties>
      <option>io.mycat.ice.server.Sl4jIceBoxServer</option>
      <env>CLASSPATH=C:\ZeroC\Ice-3.6.1\lib\*;C:\project\Ice 8\bin;C:\project\Ice 8\starter-lib\*</env>
      <service name="MyService" entry="io.mycat.ice.server.GenIceBoxService">
      properties>
          properties refid="LoadJarProps" />
          property name="myjars" value="helloservice" />
          property name="jdbc url" value="jdbc://mysql:localhost" />
      </properties>
          <adapter name="MyService" id="MyService${id}" endpoints="default"</pre>
             replica-group="MyServiceRep">
          </adapter>
      </service>
   </icebox>
</server-template>
```



# 使用指导

程序中的Jar包整理出来,放入一个Tomcat 编写version.json文件,记录Ice Servant的Jar文件的版本信息 修改grid.xml,启用远程动态加载能力



重新发布程序

>>> application list

>>> application create c:\project\ice-8\grid.xml

>>> server list

MyHelloServer1

>>> server start MyHelloServer1



### 使用指导(二)

version.json

{"helloservice":"0.0.1-SNAPSHOT"}

helloservice-0.0.1-SNAPSHOT.jar
iceclient.properties
version.json

だ果学院 roncoo.com

# 使用指导(三)

19:04:38.436 [main] INFO io.mycat.ice.server.GenIceBoxService - load jars from remote for service MyService

19:04:38.687 [main] INFO i.m.i.server.RemoteJarServiceLoader - load remote jars for MyService,from http://localhost:8080/ice-app-lib ,jars:helloserviceiceclient.properties 19:04:38.702 [main] INFO com.my.demo.MyServiceImpl - create jdbc pool for jdbc://mysql:localhost

19:05:41.802 [main] INFO io.mycat.ice.server.GenIceBoxService - MyService service started successful, class name:com.my.demo.MyServiceImpl



## Servant生命周期接口

```
package io.mycat.ice.server;
import Ice.ObjectAdapter;
public interface ServantLifcycle {
       /**
       * Servant提供服务之前会被触发此接口调用,可
以用于初始化资源, 比如建立连接池等
       public void init(ObjectAdapter adapter);
       * Servant被销毁的时候触发此调用,用于释放资
源
       */
       public void destroy();
```



#### Servant生命周期接口Demo

```
public class MyServiceImpl extends MyServiceDisp implements ServantLifcycle {
         private static final long serialVersionUID = 1L;
         private static org.slf4j.Logger logger = LoggerFactory.getLogger(MyServiceImpl.class);
         @Override
         public String hellow(Current current) {
                   return "Hello world";
         @Override
         public void init(ObjectAdapter adapter) {
                   String jdbcURL = adapter.getCommunicator().getProperties().getProperty("jdbc_url");
                   logger.info("create jdbc pool for " + jdbcURL);
         @Override
         public void destroy() {
                   logger.info("close idbc pool ");
```



# 下一集: Docker方式部署IceGrid

