Spring-boot整合elastic-job分布式调度解决方案

一、Spring boot 整合  
1. 添加依赖【在此只列出额外需要添加的elastic-job依赖的jar】

1. *<!-- ElasticJobAutoConfiguration自动配置类作用-->*
2. <dependency>
3. <groupId>com.github.kuhn-he</groupId>
4. <artifactId>elastic-job-lite-spring-boot-starter</artifactId>
5. <version>2.1.5</version>
6. </dependency>

2. 添加相应配置项

1. elaticjob.zookeeper.server-lists=127.0.0.1:2181
2. elaticjob.zookeeper.namespace=my-project

3. 创建定时任务

1. import com.dangdang.elasticjob.lite.annotation.ElasticSimpleJob;
2. import org.springframework.stereotype.Component;
3. import com.dangdang.ddframe.job.api.ShardingContext;
4. ​
5. @ElasticSimpleJob(cron = "\* \* \* \* \* ?", jobName = "test123", shardingTotalCount = 2, jobParameter = "测试参数", shardingItemParameters = "0=A,1=B")
6. @Component
7. public class MySimpleJob implements com.dangdang.ddframe.job.api.simple.SimpleJob {
8. ​
9. @Override
10. public void execute(ShardingContext shardingContext) {
11. System.out.println(String.format("------Thread ID: %s, 任务总片数: %s, " +
12. "当前分片项: %s.当前参数: %s," +
13. "当前任务名称: %s.当前任务参数: %s"
14. ,
15. Thread.currentThread().getId(),
16. shardingContext.getShardingTotalCount(),
17. shardingContext.getShardingItem(),
18. shardingContext.getShardingParameter(),
19. shardingContext.getJobName(),
20. shardingContext.getJobParameter()
21. ​
22. ));
23. ​
24. }
25. }

4. 启动2个不同端口，查看执行结果  
执行： java -jar xxx.jar --server.port=8081

1. ------Thread ID: 83, 任务总片数: 2, 当前分片项: 0.当前参数: A,当前任务名称: com.willow.elasticJob.MySimpleJob.当前任务参数:
2. ------Thread ID: 84, 任务总片数: 2, 当前分片项: 1.当前参数: B,当前任务名称: com.willow.elasticJob.MySimpleJob.当前任务参数:

1.2动态添加elastic-job任务  
当前暂未解决的问题： 动态添加的任务只能在添加的机器上运行，平行部署的其他机器上不会运行该任务

在上边配置的基础上添加以下配置：

1. 添加zookeeper配置类，和动态添加方法

1. @Configuration
2. public class ElasticJobConfig {
3. ​
4. @Bean(initMethod = "init")
5. public ZookeeperRegistryCenter regCenter(@Value("${elaticjob.zookeeper.server-lists}") final String serverList, @Value("${elaticjob.zookeeper.namespace}") final String namespace) {
6. return new ZookeeperRegistryCenter(new ZookeeperConfiguration(serverList, namespace));
7. }
8. ​
9. @Autowired
10. private ZookeeperRegistryCenter regCenter;
11. ​
12. */\*\**
13. *\* 动态添加*
14. *\* @param jobClass*
15. *\* @param cron*
16. *\* @param shardingTotalCount*
17. *\* @param shardingItemParameters*
18. *\*/*
19. public void addSimpleJobScheduler(final Class<? extends SimpleJob> jobClass,
20. final String cron,
21. final int shardingTotalCount,
22. final String shardingItemParameters){
23. JobCoreConfiguration coreConfig = JobCoreConfiguration.newBuilder(jobClass.getName(), cron, shardingTotalCount).shardingItemParameters(shardingItemParameters).jobParameter("job参数").build();
24. SimpleJobConfiguration simpleJobConfig = new SimpleJobConfiguration(coreConfig, jobClass.getCanonicalName());
25. JobScheduler jobScheduler = new JobScheduler(regCenter, LiteJobConfiguration.newBuilder(simpleJobConfig).build());
26. jobScheduler.init();
27. ​
28. }
29. }

2. 动态添加任务逻辑

1. import com.willow.elasticJob.MySimpleJob;
2. import org.springframework.beans.factory.annotation.Autowired;
3. import org.springframework.web.bind.annotation.RequestMapping;
4. import org.springframework.web.bind.annotation.RestController;
5. ​
6. @RestController
7. public class TestController {
8. ​
9. @Autowired
10. private ElasticJobConfig elasticJobConfig;
11. ​
12. @RequestMapping("/addJob")
13. public void addJob() {
14. int shardingTotalCount = 2;
15. elasticJobConfig.addSimpleJobScheduler(new MySimpleJob().getClass(),"\* \* \* \* \* ?",shardingTotalCount,"0=A,1=B");
16. ​
17. }
18. ​
19. }