#org.quartz.scheduler.instanceName属性可为任何值，用在 JDBC JobStore 中来唯一标识实例，但是所有集群节点中必须相同。

org.quartz.scheduler.instanceName = TestScheduler1

#org.quartz.scheduler.instanceId　属性为 AUTO即可，基于主机名和时间戳来产生实例 ID。

org.quartz.scheduler.instanceId = AUTO

org.quartz.threadPool.class = org.quartz.simpl.SimpleThreadPool

org.quartz.threadPool.threadCount = 10

org.quartz.threadPool.threadPriority = 5

org.quartz.threadPool.threadsInheritContextClassLoaderOfInitializingThread = true

#org.quartz.jobStore.class属性为 JobStoreTX，将任务持久化到数据中。因为集群中节点依赖于数据库来传播 Scheduler 实例的状态，你只能在使用 JDBC JobStore 时应用 Quartz 集群。这意味着你必须使用 JobStoreTX 或是 JobStoreCMT 作为 Job 存储；你不能在集群中使用 RAMJobStore。

org.quartz.jobStore.misfireThreshold = 60000

org.quartz.jobStore.class = org.quartz.impl.jdbcjobstore.JobStoreTX

org.quartz.jobStore.driverDelegateClass=org.quartz.impl.jdbcjobstore.StdJDBCDelegate

org.quartz.jobStore.tablePrefix = QRTZ\_

org.quartz.jobStore.maxMisfiresToHandleAtATime=10

#org.quartz.jobStore.isClustered 属性为 true，你就告诉了 Scheduler 实例要它参与到一个集群当中。这一属性会贯穿于调度框架的始终，用于修改集群环境中操作的默认行为。

org.quartz.jobStore.isClustered = true

#org.quartz.jobStore.clusterCheckinInterval 属性定义了Scheduler 实例检入到数据库中的频率(单位：毫秒)。Scheduler 检查是否其他的实例到了它们应当检入的时候未检入；这能指出一个失败的 Scheduler 实例，且当前 Scheduler 会以此来接管任何执行失败并可恢复的 Job。通过检入操作，Scheduler 也会更新自身的状态记录。clusterChedkinInterval 越小，Scheduler 节点检查失败的 Scheduler 实例就越频繁。默认值是 15000 (即15 秒)。

org.quartz.jobStore.clusterCheckinInterval = 20000