分布式任务调度在电力市场交易系统中的应用设计

承 林 王海宁 高春成
(南瑞集团有限公司 江苏 南京 211000)
(北京科东电力控制系统有限责任公司 北京 100192)

摘 要 为应对电力交易规模不断扩大,电力交易系统已开始引入云计算、微服务、大数据等最新 IT 技术。使用微服务技术会进一步增加系统的分布式架构特性,这种架构的演进会对系统的任务调度功能提出更高的要求。在分析电力交易系统集群规模小、任务绝对量少、策略复杂等特点的基础上,参考互联网行业的解决思路,提出一个基于分布式控制的任务调度解决方案。利用 Leader 选举的方式,解决互斥任务的调度,并且有效规避脑裂问题。采用易于同 Spring 整合的触发机制,降低介入的复杂度,解决新型分布式电力交易系统的任务调度功能面临的难题。通过压力测试和实践证明,该方案具有可实施性。

关键词 分布式系统 任务调度 电力交易系统

中图分类号 TP3 **文献标识码** A DOI:10.3969/j.issn.1000-386x.2018.11.027

APPLICATION DESIGN OF DISTRIBUTED TASK SCHEDULING IN POWER MARKET TRADING SYSTEM

Cheng Lin Wang Haining Gao Chuncheng
(*Nari Group Co., Ltd., Nanjing 211000, Jiangsu, China*)
(*Beijing Kedong Power Control System Co., Ltd., Beijing 100192, China*)

Abstract In order to cope with the continuous expansion of the scale of power trading, the electric power trading system has introduced the latest IT technology, such as cloud computing, micro service and big data. With the use of micro service technology, the distributed architecture characteristic of the system will be further increased, and the evolution of this architecture will set higher requirements for the task scheduling function of the system. In this paper, on the basis of the analysis of the small cluster size of the power trading system, the few task absolute quantity and the complex characteristics of the strategy, a task scheduling solution, based on distributed control, was proposed by referring to the solution of the Internet industry. We used Leader election to solve the scheduling of mutual excluded tasks and effectively avoided the split brain problem. The trigger mechanism, which was easy to integrate with Spring, was adopted to reduce the complexity of the intervention and solve the task scheduling problem of the new distributed power transaction system. The scheme is proved to be feasible through pressure testing and practice.

Keywords Distributed system Task scheduling Power market trading system

0 引 言

随着电力改革的推进,电力交易规模的不断扩大,参与电力市场的各类用户快速增长,电力市场也正在从中长期市场向现货市场逐步扩展。这些都对系统架构、并发响应能力、系统资源分布式调度使用能力、大数据并行计算能力提出了更新更高的要求。因此借助云计算技术、服务化设计思想,对传统单体架构的电力交易系统进行改造已成为一种必然趋势[1-3]。

任务调度[4-5]作为信息系统中最重要的功能之一,是管理信息系统中各种任务优先级安排以及任务

收稿日期:2018-03-17。南瑞集团"适应改革发展的新型电力交易系统"项目资助。**承林**,工程师,主研领域:电力市场信息化技术。**王海宁**,研究员级高工。**高春成**,高工。

执行的中枢。任务调度算法一般分为事件驱动调度算法、时钟驱动调度算法[6]。目前,时钟驱动调度算法在电力交易系统平台[7-8]中有着广泛的应用。例如横向数据同步、纵向数据同步、电力交易日清算、电力交易月结算、交易对账、日志轮转、定期数据统计分析、定期数据校验等功能,这些都属于时钟驱动调度算法的任务调度处理方式。然而在分布式微服务化的信息系统架构中,不仅存在着基于时钟驱动调度算法的业务处理需求,在各微服务之间的数据请求、数据处理的信息交互下,基于事件驱动调度算法的任务调度也将普遍存在。从资源分布、调度算法、任务执行控制、数据一致性等方面看,传统的单体系统架构中任务调度框架已无法适应分布式系统架构中对任务调度的要求。在这种情况下,一些互联网企业基于自身的需求开发出了一系列分布式调度系统,如淘宝网的 TBschedule 和当当网的 Elastic-Job 等技术架构,但是这些系统应对业务规模和基础设施与电力交易系统存在的差异较大,关注的问题也往往在负载均衡上。直接将这些系统架构应用于集群规模小,任务绝对量少,策略复杂电力交易系统中并不适合,因此急需一种针对电力交易业务应用特点和分布式系统架构的任务调度解决方案[9-10]。

1 应用现状

在传统的单体结构中,单机任务调度获得广泛应用,操作系统和各种语言的调用库,都提供了良好的实现机制。当前基于 F5 负载均衡的多节点电力交易系统中,仍采用单体调度任务模式,其中:有的调度任务部署在一个节点;有的调度任务部署在多个节点同时重复执行。前者容易出现单点故障,一旦配置调度任务的节点宕机,将会导致整个任务调度的失效;后者不仅增加无效的负载,而且容易出现数据一致性的问题。

通过将函数封外部接口、轮训方式调用或者利用虚拟 IP 实现主备机,实现一些简单分布式任务调度的能力,本质上仍是将分布式任务调度转化为单机任务调度问题。这不仅容易出现单点故障,而且对复杂业务的调度任务分配也难以应对,调度任务需要手动注册在每一个节点上,配置和维护也十分繁琐。

1.1 分布式任务调度场景分类

随着电力交易系统业务的扩展和系统架构向云计算、微服务方向演进,任务调度的场景也将日益复杂化。归纳起来,可以考虑如下三种场景:场景 A:任务一执行失败,写入部分数据,任务二读取到任务一产生的脏数据导致不一致。场景 B:任务一先于任务二执行,而任务二先于任务一完成,旧数据覆盖新数据同样导致数据不一致问题。场景 C:执行任务节点异常,系统未能成功唤起其他节点执行。

在实践中,无论是单机调度还是分布式调度,对于事物的控制通常由业务逻辑本身支持,不同业务调度之间通常是业务数据依赖。场景 A 和场景 B 中,数据一致性主要是同一调度任务不同批次之间的数据一致性。对于需用各台机器执行相同的任务,本质上属于单机调度的范畴。对于主控性任务,需要多台机器中选出一台执行的任务。如果只在一台机器上执行,那么此时分布式调度也退化成单机调度。对于场景 C 在不考虑分布式事物情况下,可以视为主备问题。

1.2 分布式任务调度主要解决方式

解决分布式系统中任务调度问题,通常有调度集中式控制和分布式规划控制式两种。

(1) 集中式控制 是任务的集中触发控制,由独立的控制模块控制,各个节点只提供任务触发的接口。

(2) 分布式控制 有各个节点独立的维护任务触发逻辑,控制中心只起到协调的作用。

集中式控制是出现较早也容易实现的方式,但容易出现单点故障。比如基于虚拟 IP 进行轮询就是一种简单实现,但虚拟 IP 失效时会引起任务调度系统整体失效。在单机任务调度应用广泛的 Quartz 框架也基于数据库行锁机制提供了一套分布式解决方案,但是仍然无法避免单点故障问题。

分布式控制将任务调度控制权分担到各个节点上,来避免单点故障问题。但是这种设计引入了复杂性,需要解决分布式系统中的协调问题。淘宝网的 TBschedule 和当当网的 Elastic-Job 也主要是通过引入 Zookeeper 技术来进行解决。

2 解决方案

2.1 关键技术与设计思路

(1) Zookeeper 与 Leader 选举 Zookeeper[11-14]是一个分布式的协调工具,通过 zab 算法来解决分布式系统一致性问题。Zookeeper 分布式系统中解决统一配置、分布式命名空间、分布式队列、Leader 选举等功能。

在分布式系统中,Leader 选举是在一个跨越几台机器(节点)的分布式任务中,指定一台机器作为任务组织者,在选举进行之前各个节点并不知道哪台机器将会成为 Leader。在 Leader 选举之后各台机器都将知道集群中唯一的 Leader。因为 Zookeeper 保证节点之

间的数据一致性和顺序性,使用 Zookeeper 可以满足 Leader 选举的要求。创建一个节点 election 通知相关机器参与选举,各台机器接到通知后在 election 节点下方建立顺序临时节点,然后选取序列号最小的节点作为 Leader。Leader 选举结束后,对 Leader 进行监听,一旦发现 Leader 节点被删除,重新发起 Leader 选举。但当 Leader 失去时,所有节点就会同时拉取 election 节点下的所有子节点,来重新进行选举。这就会对 Zookeeper 集群产生很大的压力。一种改进方法就是:任何节点只监听下一个兄弟节点,一旦出现 Leader 失效,监听 Leader 的节点必然成为 Leader,因为没有序号比它更小。

(2) 任务分配策略考虑　传统任务分配策略类似操作系统的作业调度,主要解决同构任务在不同节点的分配,关心任务执行的效率和负载均衡问题。比如在分布式计算中,子计算任务的分配策略侧重考虑的是各个节点的 CPU、内存等资源如何得到充分的利用以及如何在任务失败后重新分配。

对电力交易系统来说,分配策略的复杂性在于分配策略的多样性。一种任务分配策略是各个节点都需要同时执行的,如定时拉取缓存,访问数据库;另一种是互斥执行的,如定时结算等,这种需要在数个节点中选出一个执行,属于一种互斥性任务。

对于涉及的节点数目较少,不需过分考虑各个节点之间的均衡问题,只保证不出现单点故障的情况,本文采用 Leader 选举方式解决互斥问题。当 Leader 失去服务能力时候,进行 Leader 切换,互斥任务也迁移到新的 Leader 上,形成主从备份。

对于任务失败的处理策略,各种任务也不同,有些任务需要重复执行,有些需要任务放弃执行。各种任务失败策略以配置形式进行注册,以满足各种任务的需要。

(3) 脑裂问题的预防和解决思路　在实际的生产环境中,网络震荡是随时可能出现的,如果 Leader 所在机器出现短暂网络中断,集群则会认为 Leader 已经宕机,从而重新发起 Leader 选举。旧的 Leader 本身并不知道集群已经产生了新 Leader,这种情况常被称为脑裂。虽然 Zookeeper 本身保证了脑裂问题不会长期出现,但是需要旧 Leader 等待集群的一个通知。在(2)中介绍的互斥任务是在 Leader 上进行执行的,即使短暂脑裂,也可能引起任务重复执行,对于计费、清算这种业务来说这是不可接受。所以 Leader 需要对网络事件进行监听,一旦产生网络中断,立即释放 Leader,同时重新进行选举时候,等待一定的时间间隔,保证失去网络的 Leader 完成释放。

(4) 任务的触发控制　分布式调度任务触发控制可以分为时钟控制和任务触发两部分。

时钟触发:通过时间满足时钟条件时,激活相关动作。激活条件可以使用类似 Crotnab 的时间表达式注册于任务调度中心,表达式采用字符形式表达时间条件,每个字符分别表达秒、分钟、小时、日、月、星期、年等。这种表达式可以清晰表达时间条件。

任务控制:任务的方法、类、执行对象通过字符串的形式进行注册,满足时钟条件的时候,通过反射技术进行调用。在 Java 语言中提供了原生的反射功能支持。另外,SpingTask[15] 和 Quartz[16] 也提供完整时钟机制和反射调用框架[17],并且容易同基于 Spring 架构的电力交易系统进行集成,降低了实用和开发的难度。

2.2　系统设计

电力市场交易逻辑复杂,调度任务场景类型多,特别是分布式服务化架构的引入,使得电力交易系统对分布式任务调度存在较迫切的应用需求。

本文依据电力交易系统的应用场景特点、现有技术架构特点以及分布式架构的技术要求,并参考互联网行业的成熟解决方案,提出一个基于分布式控制的任务调度解决方案。其架构如图 1 所示。

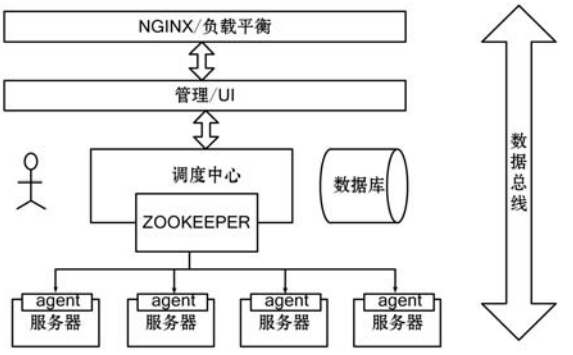

图 1　分布式调度方案架构图

该方案主要由管理界面、任务调度中心、任务调度客户端三个部分组成。用户界面提供给用户注册、编辑任务功能;任务调度中心对任务分配进行管理;任务调度客户端负责任务触发和执行。

(1) 管理界面　管理界面提供一个可视化的交互平台,提供调度任务注册、暂停、取消等功能,监控任务的执行的结果。

任务注册:任务执行是将所在机器(节点)具体指定的类和方法,以及任务执行的 Crontab 时间表达式注册于系统。对于注册于多台机器(节点)的任务,还需指定是互斥任务还是并发任务。互斥任务为在满足任务条件时选出一台机器(节点)执行任务,其他机器作为备份;并发任务为这些机器(节点)同时执行的任务。

任务修改:管理界面提供功能启动和暂停任务,可

以在任务执行时刻前暂停任务。

任务监控:通过管理界面查看任务的状态、历史执行时间、执行结果、互斥任务显示、执行的实际机器等信息。

(2) 任务调度中心 如图 2 所示,任务调度中心为调度系统核心,控制各个节点任务的实际执行行为。本文提出的调度中心使用基于 Zookeeper 的方案,用户通过管理界面注册任务时,是注册在各个机器对应的节点下面,并在该节点下面建立子节点。客户修改任务时,调度中心就更新对应节点下面的内容。管理端删除任务时候,将对应任务节点删除。

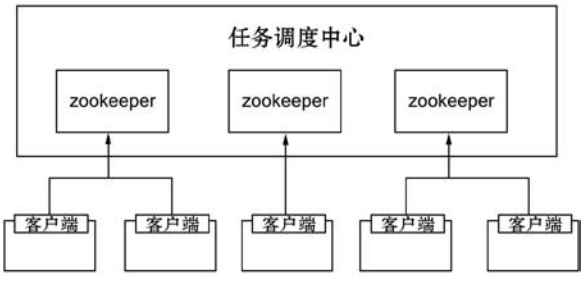

图 2 任务调度中心

(3) 任务调度客户端 任务调度客户端部署执行的应用,客户端部分负责调度的具体执行,客户端的主要结构有监听器、任务容器、任务调度器,如图 3 所示。

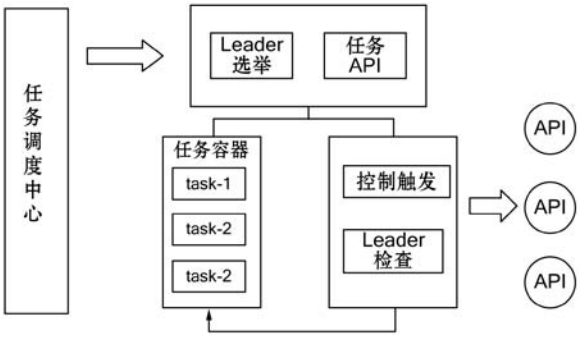

图 3 任务客户端示意图

监听器在客户端所在应用进程初始化开始时对调度中心对应的 IP 节点开始监听,并从该节点拉取任务数据保存到任务容器中。如果子节点发生增加、删除、修改就要对任务容器中对应的任务信息进行增加、删除、修改,并且通知任务控制器进行相应的处理。

任务容器:用来存储本地任务的具体信息,如任务名称、任务类型、任务调度方法、调度表达式、任务状态等信息。

任务调度器:初始化时任务调度从任务容器获取任务,依据任务的时间表达式来启动任务,并将执行状态写到任务容器中。当监听器监测到新增或者启动事件时,调度器将会从任务容器中取出任务,检查任务状态,如果任务尚未启动就启动它,并更新状态到任务容器;如果任务执行失败,则根据配置判断是否需要重复执行。

如果监听到暂停或者删除事件,将首先修改状态,再将其从任务容器中删除。

如果任务的类型是互斥任务,客户端初始化时,就会发起 Leader 选举,从多台备用机中选出一台,作为实际执行调度的机器。调度器在加载任务时,如果检查到自身是 Leader 就正常启动,如果不是 Leader 就放弃启动。任务启动后 Leader 选举的状态保持监听,如果监听到丧失 Leader 权限,就暂停任务,如果监听到获得 Leader 身份,则重启动调度任务。

(4) 效果监控功能 为了调度任务进行管理和监控,任务的注册、执行、完成或者异常等数据将实时写入消息总线,并通过消息总线同步到数据库中。管理界面通过发布订阅机制与数据总线保持监听,并将接收到的消息实时同步到管理界面。

监控模块对写入消息总线的数据进行过滤,当监控到任务异常消息时触发报警通知管理员处理。由于本文阐释系统不考虑对分布式事物的控制,即使在单机系统的调度中,调度系统本身也不牵涉到回滚等事物操作的,所以在实践中采用报警触发的机制是能满足实际业务需要的。

3 方案验证

基于本文提出的方案构建的系统,在实践中运行了三个月,其中普通节点 4 台,执行主控性任务机器 2 台,任务执行的成功率为 99.98%,未发生因 Leader 切换失败或任务失效等情况,也未发生因短期脑裂产生任务重复执行的情况。

对于电力交易系统而言,负载性能并不是业务痛点所在。为应对更大规模的集群介入,对该系统进行了压力测试,结果如表 1 所示。

表 1 压力测试结果

每秒查询率/(次·s^{-1})	响应速度/秒
100	0.232
200	0.342
400	0.42

本系统采用分布式的技术架构,对于任务调用中心的 Zookeeper,可以通过扩充集群来提高负载性能,对于管理端界面,也可以通过负载均衡的手段,实现水平扩展提高吞吐量。

在系统运行的几个月中,除了系统上线引起 Leader 切换外,因网络问题出现几次意外的切换,但都通过本

文的 Leader 选举改进机制和任务注册管理功能,防止了分布式系统中 Leader 选举的脑裂问题,从而保护了任务不会因为网络震荡被重复调度执行。

4 结　语

本文提出了一种基于改进式 Leader 选举的分布式任务调度系统,解决了电力交易系统从单体架构演进到分布式架构中的复杂任务调度问题。在分析了电力交易业务和电力交易系统的基础上,利用改进式 Leader 选举方式解决了互斥任务的调度问题,提供了可配置的失败任务处理方式。为多样性的电力交易系统提供了灵活的支持,并通过方案验证和压力测试,证明该方案不仅能够满足当前系统的需要,而且在面对更大规模业务需要时,依然能够良好运行。

参 考 文 献

[1] 杨平,史连军,邵平,等. 新一代电力市场交易平台架构探讨[J]. 电力系统自动化,2017,41(24):67-76.

[2] 严宇,李庚银,李国栋,等. 新一轮电改形势下电力直接交易组织情况分析[J]. 中国电力,2017,50(7):33-37.

[3] 承林,王海宁,高春成. 微服务在电力交易系统中的应用研究[J]. 电网技术,2018,42(2):442-446.

[4] 曹阳,高志远,杨胜春,等. 云计算模式在电力调度系统中的应用[J]. 中国电力,2012,45(6):14-17.

[5] Schwiegelshohn U, Yahyapour R. Attributes for communication between Grid scheduling instances[M]//Grid resource management. Kluwer Academic Publishers,2004:41-52.

[6] Fischetti M, Martello S, Toth P. The Fixed Job Schedule Problem with Working-Time Constraints[J]. Operations Research,1989,37(3):395-403.

[7] 杨争林,曹帅,郑亚先,等. 电力市场全景实验平台设计[J]. 电力系统自动化,2016,40(10):97-102.

[8] 张显,郑亚先,耿建,等. 支持全业务运作的电力用户与发电企业直接交易平台设计[J]. 电力系统自动化,2016,40(3):122-128.

[9] 王晓川,叶超群. 一种基于分布式调度机制的集群体系结构[J]. 计算机工程,2002,28(8):232-234.

[10] 王德文,刘杨. 一种电力云数据中心的任务调度策略[J]. 电力系统自动化,2014,38(8):61-66,97.

[11] Prisco R D, Lampson B W, Lynch N A. Revisiting the Paxos algorithm[C]//Proceedings of the 11th International Workshop on Distributed Algorithms. Springer-Verlag,1997:111-125.

[12] 唐海东,武延军. 分布式同步系统 Zookeeper 的优化[J]. 计算机工程,2014,40(4):53-56.

[13] Apache Software Foundation. Apache Zookeeper[EB/OL]. (2013-02-01). http://Zookeeper.apache.org/.

[14] Hunt P, Konar M, Junqueira F P, et al. ZooKeeper: Wait-free Coordination for Internet-scale Systems[C]//Proceedings of the 2010 USENIX conference on USENIX annual technical conference. USENIX Association,2010:11.

[15] Apache Softwarev Foundation, Execution and Scheduling [EB/OL]. https://docs.spring.io/spring/docs/3.2.x/spring-framework-reference/html/scheduling.html.

[16] 朱哲明. 基于 Quartz 的消息沟通平台的研究[D]. 北京:北京邮电大学,2013.

[17] 朱跃龙,韦敏,冯钧. 使用 Java 反射的可扩充水利数据库应用系统[J]. 计算机工程,2006,32(22):96-98.

(上接第 152 页)

[2] Dong X, Zhao Y, Xu Y, et al. Design of PSO fuzzy neural network control for ball and plate system[J]. International Journal of Innovative Computing Information & Control Ijicic, 2011, 7(12):7091-7103.

[3] 梁琛,王鹏,韩肖清,等. 基于间歇性风速的风力发电机功率输出模型研究[J]. 电网技术,2017,41(5):1370-1374.

[4] Munteanu I, Cutululis N A, Bratcu A I, et al. Optimization of variable speed wind power systems based on a LQG approach[J]. Control Engineering Practice, 2005, 13(7):903-912.

[5] 田兵,赵克,孙东阳,等. 改进型变步长最大功率跟踪算法在风力发电系统中的应用[J]. 电工技术学报,2016,31(6):227-230.

[6] Boukhezzar B, Siguerdidjane H. Nonlinear Control of a Variable-Speed Wind Turbine Using a Two-Mass Model[J]. IEEE Transactions on Energy Conversion, 2011, 26(1):149-162.

[7] Leithead W E, Connor B. Control of variable speed wind turbines: Design task[J]. International Journal of Control, 2000, 73(13):1189-1212.

[8] Song Z, Shi T, Xia C, et al. A novel adaptive control scheme for dynamic performance improvement of DFIG-Based wind turbines[J]. Energy, 2012, 38(1):104-117.

[9] Vihriala H, Perela R, Makila P, et al. A gearless wind power drive: part 2: performance of control system[C]//Proceedings of the Wind Energy for the New Millennium European Conference(EWCE'01). 2001:1090-1093.

[10] 张洪新, 涂群章, 蒋成明,等. 基于模糊 PI 控制器的 PMSM 矢量控制[J]. 装备制造技术,2017(7):127-130.

[11] 于子捷,魏晨曦,田芳芳,等. 一种改进型扰动观测法在最大功率点跟踪中的应用[J]. 电测与仪表,2017,54(15):113-119.

[12] 孙超. 基于改进 MPPT 算法的风光互补发电功率最大功率跟踪系统研究[D]. 天津:天津理工大学,2017.