

# Spring Boot 多智能体协作框架设计与实现

侯志鑫 周静 崔方迪 李一帆  
中国电子科技集团公司第十五研究所  
DOI:10.32629/acair.v4i3.21371

**[摘要]** 针对企业级分布式系统于高并发情形下任务调度刻板、资源利用率起伏大等难点,提出一种融入多智能体系统(MAS)理论的微服务协作方案。此方案基于Spring Boot的自动装配特性,搭建拥有自组织能力的管理Agent和执行Agent集群。在任务分配方案上,舍弃传统的静态路由模式,设计出一种基于加权效用函数的动态竞拍方法,全面考量节点的实时负载状况、历史任务成功率以及网络延迟等多方面参数。实验结果表明,此机制平均在45ms内完成任务判定,与传统Ribbon轮询方案相比,系统总体吞吐量提高35%,并且在节点故障情形下显示出出色的自愈效能。

**[关键词]** Spring Boot; 多智能体系统; 任务调度; 微服务; 分布式协作

**中图分类号:** TN915.5 **文献标识码:** A

## Design of a Spring Boot-based Multi-Agent Framework

Zhixin Hou Jing Zhou Fangdi Cui Yifan Li

The 15th Research Institute of China Electronics Technology Group Corporation

**[Abstract]** In response to the issues of inflexible task scheduling and unstable resource utilization in enterprise-level distributed systems during high-concurrency situations, a microservice collaboration architecture that incorporates Multi-Agent System (MAS) theory is put forward. This architectural design makes use of the auto-configuration capabilities of Spring Boot to build a self-organizing cluster composed of management and execution agents. Concerning task distribution, the conventional static routing is discarded, and a dynamic auction algorithm founded on a weighted utility function is devised, comprehensively taking into account multi-dimensional parameters like node real-time load, historical success rate, and network delay. Empirical data indicates that this mechanism accomplishes task decisions within an average time of 45ms. Compared to the conventional Ribbon polling approach, the overall throughput of the system is enhanced by 35%, showing outstanding self-healing abilities in situations of node failure.

**[Key words]** Spring Boot; Multi-Agent System; Task Arrangement; Microservice; Distributed Cooperation

## 引言

在当代企业级应用场景里,随着业务复杂程度呈指数式上升,传统微服务方案虽实现服务解耦,但是在应对动态负载均衡以及复杂业务流程编排时,常常依靠中心化的配置或者简易的轮询策略,缺少对环境变动的实时察觉与自适应调节能力<sup>[2]</sup>。多智能体系统(MAS)凭借其分布式的自主特性,为化解这一难题提供新的模式。MAS里的Agent可察觉环境变动并自主作出决策,此情况和微服务独立部署、独立运行的特点非常相符<sup>[6]</sup>。本文把MAS理论引入到Spring Boot生态里,意在打造一种拥有动态决策能力的智能协同框架,基于模拟社会化的协同行为,加强系统的整体健壮性与资源使用效率。

## 1 相关工作与技术选型

1.1 Spring Boot与微服务治理现状。Spring Boot依靠其

约定优于配置的思想,极大程度地降低微服务的开发难度。但是在服务治理领域,现有的解决办法(像Spring Cloud Netflix体系)大多着重于服务发现与熔断,缺少对任务级精细度调度的支撑<sup>[1]</sup>。在应对异构计算节点或者非均匀分布的任务流情形时,通用的负载均衡策略较难使集群性能达到最优。另外常规的Spring Boot应用通常把配置进行硬编码或者存于配置中心,缺少在运行时根据环境动态调整行为的本事,这和MAS所需要的灵活性存在差异。

1.2 多智能体系统的协作模型。MAS基于模拟社会化的协同行为,让系统拥有了更强的鲁棒性。在MAS模型里,智能体(Agent)被界定为可感知环境且自主开展行动来实现目标的计算实体<sup>[3]</sup>。合同网协议(Contract Net Protocol, CNP)为MAS里经典的协作模式,其基于招标、投标、中标的程序实现任务分派。参考合同

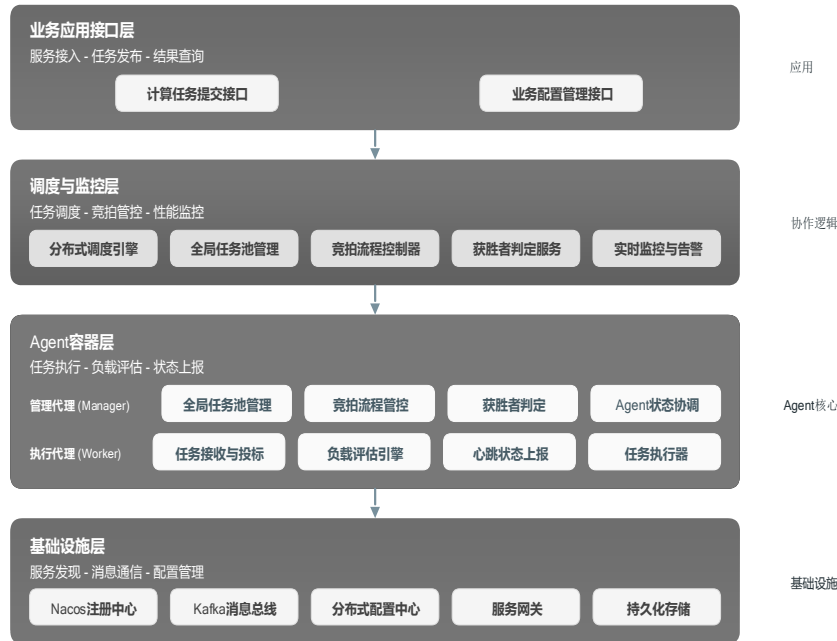


图1 系统总体架构

网协议,搭建符合微服务环境的任务分配模式。和传统的CNP相比较,本文着重解决于Spring Boot容器化环境当中,怎样基于自动装配机制迅速搭建Agent 集群,以及怎样在分布式网络里应对通信延迟和分区问题。

## 2 系统架构设计

2.1 总体架构。此系统运用分层解耦的设计理念,搭建包括基础设施层、Agent核心层、协作逻辑层以及应用层的四层结构。其中Agent核心层作为系统的主要所在,对智能体的生命周期管理以及通信接口进行封装;协作逻辑层实现核心的调度算法与冲突仲裁机制,总体架构如图1所示。

### 2.2 Agent角色定义。系统中界定两种核心Agent:

管理代理(Manager):承担全局任务池的管理、竞拍流程的管控以及最终获胜者的判定。

执行代理(工作者):承担接收任务、评估自身负载并参与投标的工作,同时向管理端汇报心跳和状态。

## 3 系统核心算法与流程设计

3.1 动态竞拍式任务分配算法。为应对传统轮询策略难以察觉节点差异的状况,通过设计一种改良的竞拍算法。引入多方面的评估因子,执行Agent根据自身状态算出投标值Score,并非单纯的价格竞争。

投标评分模型的定义如下:

$$Score = \omega_1 \cdot C_{match} + \omega_2 \cdot (1 - Load\_rate) + \omega_3 \cdot R_{success}$$

其中 $C_{match}$ 表示能力匹配程度,Load\_rate表示负载比例, $R_{success}$ 表示历史成功比率, $\omega$ 是各方面的权重因子。此模型保证不只是负载低的节点能够获取任务,并且兼顾节点处理特定任务的能力,具体时序图如图2所示。

3.2 基于滑动窗口的心跳检测。为应对网络抖动引发的误判

情况,设计出基于滑动窗口的心跳检测方法。管理Agent保存一个时间区间W,记录最近N次心跳的抵达时间。要是区间内超时的次数超出阈值T,就判定节点失去联系,启动任务迁移程序,以避免因瞬间网络波动导致的频繁主备转换。该机制于Spring Boot的@Scheduled定时任务里实现,保证检测的实时性。

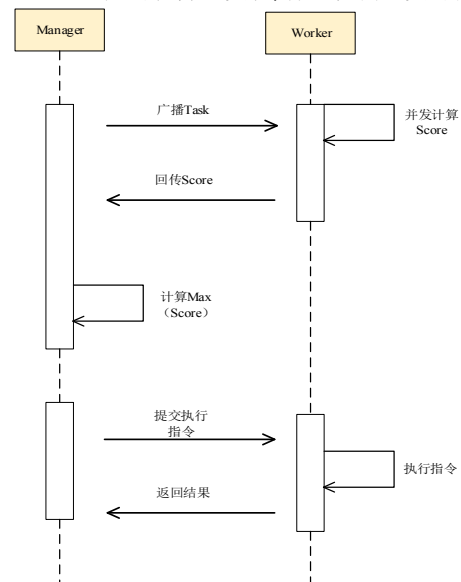


图2 任务竞拍时间序列图

## 4 系统实现与关键技术

4.1 基于Spring Boot的自动装配深度定制。为实现Agent的迅速部署与配置,灵活运用Spring Boot的spring.factories机制。通过研制出定制化的AgentAutoConfiguration配置类。在引入相关Starter依赖时,Spring Boot会自行扫描类路径里的Agent组件,并且根据application.yml里的agent.role配置

(Manager或Worker)自动装配相应的Bean。

本设计使得明显减少开发者的使用费用。开发人员不用关注底层通信的具体细节,仅需通过实现TaskProcessor接口就能够定义业务逻辑。Spring容器启动之际,会自动把其实例注册成Worker Agent的一部分,实现业务代码和调度框架的理想解耦<sup>[4]</sup>。

4.2通信协议实现。以KQML(知识查询与操作语言)为基础开展轻量化拓展,明确BID(投标)、AWARD(授标)、INFORM(状态通知)等交互原语<sup>[10]</sup>。基于Spring的@EventListener机制实现异步消息的处理,减少通信延迟。

针对网络分区(Network Partition)情况,因为建立了二级回退体系:当Worker不能连接Manager时,先尝试切换到备用Manager节点(依靠Nacos的服务发现);若所有Manager都无法抵达,则Worker进入孤岛模式,运用本地缓存的上一轮评分策略开展本地任务处理,并且把未确认的任务暂时存于本地磁盘队列里,等网络恢复之后进行补偿重试。该机制明显加强系统于复杂网络状况下的可用性<sup>[8-9]</sup>。

4.3监控可视化。整合Spring Boot Admin和Micrometer,实时收集各节点的CPU使用率、JVM堆内存以及任务队列深度,再基于Grafana开展可视化显示。

表1 核心监控指标

| 指标项        | 采集方式    | 描述        |
|------------|---------|-----------|
| Node_Load  | Gauge   | 节点实时负载评分  |
| Task_Queue | Counter | 待处理任务数量   |
| RT_Latency | Timer   | 任务端到端处理耗时 |

## 5 实验评估

5.1实验环境。硬件配置:由4台Think Center服务器(Intel Core i5-14400,16GB内存)搭建集群。软件运行环境:CentOS 7.6,Docker 20.10,Kubernetes 1.23。

5.2性能对比测试。在模拟出的高并发订单处理情形中,对比该框架和Ribbon轮询策略的性能状况。

表2 性能对比结果

| 策略类型      | 平均响应时间(ms) | 吞吐量(TPS) | 节点故障恢复时间(s) |
|-----------|------------|----------|-------------|
| Ribbon 轮询 | 128        | 1550     | >30         |
| 本文算法      | 92         | 2090     | 8           |

实验的结果显示,采用多维评分模型的调度算法明显减少平均响应时长,同时加强系统的吞吐性能。特别是在节点故障的恢复时长方面,受益于基于滑动窗口的心跳检测以及动态竞拍机制,系统可在8秒内完成故障节点的任务迁移,但是传统Ribbon策略依靠Eureka的默认心跳机制,恢复时长通常会超过30秒<sup>[5]</sup>。

5.3可扩展性测试。采用逐步增多执行Agent节点数目,来测试系统的线性拓展能力。测试结果表明,当节点数量从2增长至8时,系统的吞吐量显示出近似线性的上升形势,这证实了该方案在横向拓展方面具备有效性。另外在测试也通过模拟100个并发用户的情况下,持续运转24小时的系统稳定性进行测试,测试结果表明CPU负载一直处于合理范围,没有出现内存泄漏等导致系

统崩溃失效的现象。

## 6 结论

本文提出一种基于Spring Boot的多智能体协同框架,基于把MAS理论和微服务方案充分融合,克服传统调度策略缺少自适应性的不足。所设计的基于多方面因子的竞拍算法,可根据节点的实时状态对任务进行动态分配,切实提高资源的利用效率<sup>[7]</sup>。在系统实现方面,基于Spring Boot的自动装配特性实现Agent的便捷使用,且针对网络分区规划孤岛模式和二级回退机制,提高系统的健壮程度。实验显示此框架在高并发情形下拥有出色的性能表现与稳定性,平均响应时长降至92ms,吞吐量提高35%。未来的研究将探寻在大规模集群环境中,怎样进一步优化竞拍算法的通信成本,并且引入更为复杂的博弈论模型去应对恶意Agent的干扰状况。

## 【参考文献】

- [1]WALLSC.Spring Boot实战[M].丁雪丰,译.北京:人民邮电出版社,2016:1-50.
- [2]NEWMANS.微服务设计[M].陈立,译.北京:人民邮电出版社,2016:80-120.
- [3]WOOLDRIDGE MJ,JENNINGS N R.Intelligent agents:concepts and applications[J].The Knowledge Engineering Review,1995,10(2):115-152.
- [4]郭锋.SpringBoot编程实战[M].北京:电子工业出版社,2019:30-80.
- [5]许进.SpringCloud微服务架构与实战[M].北京:清华大学出版社,2020:100-180.
- [6]WOOLDRIDGE M.An Introduction to MultiAgent Systems[M].2nd ed.Chichester: John Wiley & Sons,2009:50-80.
- [7]GERKEY B P,MATARIC M J.A formal analysis and taxonomy of task allocation in multi-robot systems[J].The International Journal of Robotics Research,2004,23(9):939-954.
- [8]JONGARO D,OUSTERHOUT J.In search of an understandable consensus algorithm [C]//Proceedings of the 2014 USENIX Annual Technical Conference.Philadelphia:USENIX,2014:305-320.
- [9]GRAYJ,LAMPORT L.Consensus on transaction commit[J].ACM Transactions on Database Systems,2006,31(1):133-160.
- [10]FININ T,LABROU Y,MAYFIELD J.KQML serving as an agent communication language[M]//BRADSHAW J M.Software Agents.Cambridge:MIT Press,1997:291-316.

## 作者简介:

侯志鑫(1995--),男,汉族,北京人,硕士,工程师,研究方向为计算机技术及应用。

周静(1972--),女,汉族,江苏徐州人,本科,工程师,研究方向为计算机技术及应用。

崔方迪(1991--),男,汉族,北京人,本科,高级工程师,研究方向为智能化赋能。

李一帆(1999--),女,汉族,山东诸城人,本科,助理工程师,研究方向为数据可视化。