



(12)发明专利申请

(10)申请公布号 CN 110266771 A

(43)申请公布日 2019. 09. 20

(21)申请号 201910466088.9

(22)申请日 2019.05.30

(71)申请人 天津神兔未来科技有限公司  
地址 300000 天津市滨海新区自贸试验区  
(中央商务区)新华路3699号宝元大厦  
16层Y0-16002

(72)发明人 王静逸

(74)专利代理机构 北京市兰台律师事务所  
11354

代理人 张峰

(51) Int. Cl.  
H04L 29/08(2006.01)  
H04L 12/24(2006.01)  
G06N 3/00(2006.01)

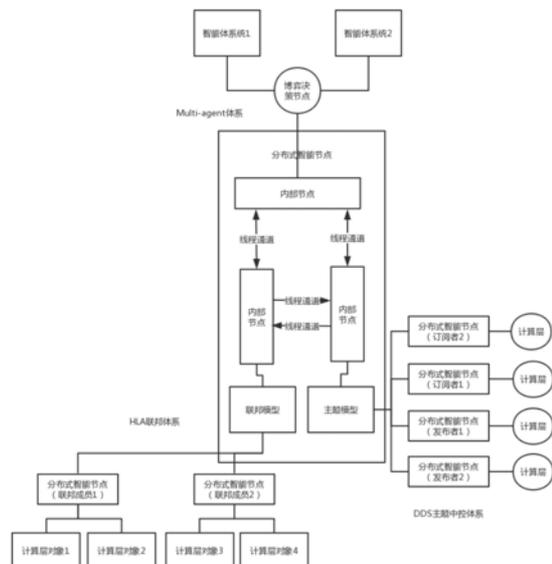
权利要求书2页 说明书9页 附图12页

(54)发明名称

分布式智能节点及分布式群体智能统部署方法

(57)摘要

本发明是关于一种分布式智能节点,所述分布式智能节点通过线程分布式的方式形成内部节点,所述内部节点分别通过加载HLA的联邦模型、DDS的主题模型,可至少支持HLA、DDS、Multi-Agent三种以上的分布式体系,不同的内部节点通过线程通道进行信息的交换和同步。此发明还提供一种部署分布式群体智能系统的方法,包括设计系统及部署分布式智能节点两大环节。本发明解决了规模化的计算、计算模型的拆分、多智能专家系统的协作、群体智能决策和智能系统决策灵活组织等问题。



1. 一种分布式智能节点,其特征在于,所述分布式智能节点通过线程分布式的方式形成内部节点,所述内部节点分别通过加载HLA的联邦模型、DDS的主题模型,可至少支持HLA、DDS、Multi-Agent三种以上的分布式体系,不同的内部节点通过线程通道进行信息的交换和同步;

所述HLA体系,一个体系整体视为一个联邦,一个联邦包含多个联邦成员,一个联邦成员包含多个对象,所述联邦成员遵循统一的联邦对象模型和成员对象模型,所述联邦对象模型和所述成员对象模型定义了体系的属性、接口、映射、对象和交互标准;

所述DDS体系,包括主题、数据发布模块、数据订阅模块;所述主题用于定义数据模型;所述数据发布模块用于发布数据到所述主题中;所述数据订阅模块用于从所述主题订阅数据模型;

所述Multi-Agent体系,拥有多个智能体,所述智能体是独立的,拥有自己的计算逻辑和智能,不同智能体之间可以进行数据和计算模型的沟通。

2. 根据权利要求1所述的分布式智能节点,其特征在于,在所述DDS体系中,所述数据发布模块包括:数据写入器及发布者,所述数据写入器,用于将数据输入所述DDS体系中;所述发布者,用于将数据传输发送到所述DDS体系中。

3. 根据权利要求1所述的分布式智能节点,其特征在于,在所述DDS体系中,所述数据订阅模块包括:数据读取者及订阅者,所述数据读取者,用于读取所述订阅者订阅的数据,所述订阅者,用于向所述DDS体系订阅数据,通过积极的轮询,获取所述数据读取者的新数据。

4. 根据权利要求1所述的分布式智能节点,其特征在于,所述Multi-Agent体系,包括三种组织结构:

集中式结构,不同的智能体分为多个群组,每个群组都有一个领队智能体,所述领队智能体负责协调统一本组智能体的智能决策,在与其他领队智能体相互协调,形成最终决策;或

分布式结构:每个智能体完全平级,没有上下级关系;或

混合式结构:混合了所述集中式结构和分布式结构。

5. 根据权利要求1所述的分布式智能节点,其特征在于,在开放性网络中,通过与其他分布式智能节点形成联邦分布式体系或DDS分布式体系或Multi-Agent分布式体系,此时不同类型的分布式体系也可作为一个大型分布式智能节点。

6. 根据权利要求1所述的分布式智能节点,其特征在于,通过连接外置的软件开发工具包运行具有计算能力的计算程序形成计算层,所述外置的软件开发包提供适用不同环境下的计算机编程SDK。

7. 根据权利要求1-6任意一种所述的分布式智能节点,其特征在于,对需要由复杂计算程序提供的数据,所述分布式智能节点通过消息协议,定义代理智能体,所述代理智能体使用所述SDK连接运行复杂的计算程序,通过设置拦截器,拦截一些复杂的计算程序中非法数据和不必要的结果,确定需要同步的数据和结果形成所述复杂计算层,进行计算任务。

8. 一种部署分布式群体智能系统的方法,其特征在于包括如下步骤,

1) 设计分布式体系步骤:分布式体系至少包括,HLA体系、DDS体系以及Multi-Agent体系的一种或多种,分布式体系由分布式智能节点构成;

2) 设计分布式模型步骤:设计分布式联邦模型,编写模型任务文件,

所述分布式联邦模型包括分布式智能节点的树形结构;数据,计算模型之间的交互方式;以及所述计算模型中的数据结构;

3) 设计计算层步骤:根据分布式模型,设计计算层和相关计算层算法模块,

所述计算层为,利用软件开发包连接到所述分布式智能节点,提供计算能力的计算程序;

所述计算层算法为所述计算层中计算程序,根据它所申请的联邦模型中的任务对象,所设计的特定的、完成某一能力的计算机程序步骤。

4) 设计同步与异步步骤:设计所述计算层的协调接口、数据一致性算法、数据类型、同步和异步方式;

5) 部署节点步骤:根据设计的分布式群体智能体系,部署分布式智能节点;

6) 启动节点步骤:启动联邦中央节点,启动其他分布式智能节点,组成分布式联邦网络;

7) 计算层连接步骤:根据设计的计算层,发布所述分布式联邦模型,使用对外软件开发包编写计算层算法模块,并且连接到对应的分布式智能节点;

8) 运行系统步骤:运行所述分布式群体智能系统,启动中控系统,监听运行环境,在使用中迭代更新智能参数、调整所述分布式群体智能系统。

9. 根据权利要求8所述的部署分布式群体智能系统的方法,其特征在于,

在所述设计同步与异步步骤后,还包括设计代理层步骤:定义代理智能体的属性,用于连接负责的计算层,代理智能体的数据同步和过滤。

10. 根据权利要求9所述的部署分布式群体智能系统的方法,其特征在于,

在所述计算层连接步骤中,当所需要的计算程序为复杂程序时,使用特定的SDK编写委托代理智能体模块,通过SDK建立所述代理智能体与复杂的计算程序的连接,通过设置拦截器,拦截一些复杂的计算程序中非法数据和不必要的结果,确定需要同步的数据和结果形成复杂计算层,所述分布式智能节点通过所述对外软件开发包连接所述复杂计算层,进行计算任务。

## 分布式智能节点及分布式群体智能统部署方法

### 技术领域

[0001] 本发明专利涉及计算机分布式系统领域,尤其涉及不同分布式体系下的分布式智能节点设计及分布式群体智能系统部署的方法。

### 背景技术

[0002] 在最近的几年中,人工智能随着神经网络的突破,得到了巨大的发展,特别在图像、分析、推荐等领域。在人工智能快速发展的同时,计算规模不断扩大、专家系统过于单一、神经网络模型的灵活性、应用领域的复杂性等问题,也在不断升级。在这样的环境下,分布式人工智能的发展被研究机构 and 大型企业提上日程。现有的较为流行的分布式系统主要包含以下三种。

[0003] 一、Tensorflow的分布式方案

[0004] Tensorflow底层是高性能的GRPC库,分布式集群的组件主要是三个部分:client、master和worker process。运用它的组件,可以形成两种主要的分布式部署模式:单机多卡(single process)和多机多卡(multi-device process)。方案出于Martin Abadi、Ashish Agarwal、PaulBarham论文《TensorFlow:Large-Scale Machine Learning on Heterogeneous Distributed Systems》。

[0005] 它的部署和运作模式,如图1所示,对于普通的机器学习,单机单卡就可以进行运算,在运算规模不断升级的时候,就需要对tensorflow的训练设计分布式结构。机器学习的参数训练,主要是两大过程:(1)使用卷积参数训练梯度;(2)根据梯度再优化更新参数。在大规模的计算过程中,就需要进行集群计算。在小型规模的时候,可以使用单个机器,多个CPU/GPU来进行计算;在规模较大的时候,可以使用多个机器进行并行计算。

[0006] 根据Tensorflow的分布式机器学习方案,我们可以总结它存在的一些缺点:(1)Tensorflow在分布式的计算方式中,分为了训练和参数更新服务,极大地扩展了规模化的计算能力,但是在数据归并和多级分层上,并没有提供非常好的解决方案。没有较好的计算模型,去提供多层次的分布式结构。(2)Tensorflow在分布式的计算模型中,主要分为了训练任务、梯度计算、参数更新等,但是却没有提供一个较好的计算模型构造方式,去提供分布式AI所需要的任务分发、数据归并、模型更新等能力。(3)Tensorflow在分布式的组织方式上,更像是一个平级层次的分布式,最终统一归并成为一个最终模型,它最终完成的是一个集中化的单一专家系统,无法提供群体智能能力,也无法形成多群体博弈对抗的协调决策,无法支持多专家系统的兼容协调。(4)Tensorflow需要高性能计算机,高性能的GPU,难以支持小型设备、物联网等多设备的边缘化集群计算。

[0007] 二、Spark分布式机器学习

[0008] 在Spark中,计算模型可以设计成为有向无环图DAG,无环图的顶点是RDD,它是Spark的核心组件。RDD是一种弹性的分布式数据集,它可以支持多个RDD分片的依赖、变换(transformation)和动作(action),它可以从RDD\_A变换成为RDD\_B,transformation就是DAG无环图的边。通过DAG的表示方法,建立计算模型,并且编译成为Stage。如图2-1所示:

Spark运用它的分布式机制,可以支持多个Stage并行计算,也能支持Stage下属Stage模型分层结构。Spark基于它的master-worker架构,可以将DAG中的stage分割,指定到不同的机器上面执行任务。它的分布式并行运算流程,如图2-2所示:它的驱动器driver负责协调任务和调度器组件scheduler,调度器分为DAG和Task调度器,用来分配Task到不同的运算单元。

[0009] 根据Spark的分布式机器学习机制,它存在如下缺点:(1) Spark支持大规模的模型计算分层,但是计算消耗较大,不适合用较小规模的分布式方案。(2) Spark在节点和节点之间的数据路由不是非常灵活。(3) Spark难以在小型化设备进行联合学习运算,无法部署在小型化设备之上。(4) Spark的框架是一个并行计算框架,它的模式是任务确认调度-模型划分-分片计算-归并总结,更适合大数据集的机器学习,不适合多节点博弈等强化学习方法。

[0010] 三、Google联合学习方案

[0011] 针对小型化设备,google在一篇文章《Federated Learning: Collaborative Machine Learning without Centralized Training Data》中,提出了联邦联合学习的概念。

[0012] 它的工作原理为:(1) 手机或者小型设备,下载云端的共享模型。(2) 每个小型设备的用户,通过自己的历史数据,训练和更新模型。(3) 将用户个性化更新后的模型,抽取成为一个小的更新文件。(4) 提取模型的差异化部分,进行加密,上传到云端。(5) 在云端将新用户的差异化模型和其他用户模型,进行Average平均化,然后更新改善现有共享模型。

[0013] 这样工作的好处有:(1) 聚合了边缘的小型化设备(比如手机),增加了AI的数据来源和计算能力来源。(2) 在机器学习的模型结果上,更加适应广泛用户的行为数据模型。(3) 不同群体之间,可以产生博弈和不断强化的模型结果,模型可以在广泛的分布式基础下,不断迭代更新。它实际上就是DDS中心订阅模型分发、然后进行联合学习的模型。它并不是特别支持HLA联邦模型以及委托代理模型。

## 发明内容

[0014] 为了解决集中化人工智能的几个主要问题:(1) 规模化的计算问题。(2) 计算模型的拆分训练。(3) 多智能专家系统的协作。(4) 多智能体博弈和训练演化,解决数据集不足的问题。(5) 群体智能决策和智能系统决策树的灵活组织,适应复杂的应用场景,比如工业、生物、航天、社群经济研究等领域。(6) 适应物联网和小型智能设备,联合更多的边缘计算单元和移动设备。

[0015] 本发明提供一种分布式智能节点,分布式智能节点之间相互连通形成分布式网络,通过分布式群体智能系统可以控制分布式智能节点。如图3所示,分布式智能节点通过线程分布式的方式,形成内部节点,所述内部节点分别通过加载HLA的联邦模型、DDS的主题模型,可至少支持HLA、DDS、Multi-Agent三种以上的分布式体系,不同的内部节点通过线程通道进行信息的交换和同步;

[0016] 所述HLA体系,如图4所示,一个体系整体视为一个联邦,一个联邦包含多个联邦成员,一个联邦成员包含多个对象,所述联邦成员遵循统一的联邦对象模型和成员对象模型,所述联邦对象模型和所述成员对象模型定义了体系的属性、接口、映射、对象和交互标准;

[0017] 在HLA的联邦分层基础下,传统的机器学习模型,可以通过HLA联邦对象模型的规

则,进行分布式任务的划分,将任务向下分发,形成多任务并行计算,并且可以归并群体决策的HLA联邦AI智能模型;

[0018] 所述DDS体系,包括主题、数据发布模块以及数据订阅模块;所述主题用于定义数据模型;所述数据发布模块用于发布数据到所述主题中;所述数据订阅模块用于从所述主题订阅数据模型;

[0019] DDS体系是一种基于分布式系统通讯模型,它基于发布/订阅模式,是相对比较简洁和直观的分布式系统结构,它适合用在数据的分发、同步等领域。在DDS体系中,所有成员都是实体数据角色。它类似联合学习的分布式结构,它可以通过分布式系统的的一个核心节点作为云服务,提供共享模型,联合大量设备联合学习和更新模型。

[0020] 所述Multi-Agent体系,拥有多个智能体,所述智能体是独立的,拥有自己的计算逻辑和智能,不同智能体之间可以进行数据和计算模型的沟通。

[0021] 在Multi-Agent体系中,多个智能体,它们可以协调服务,通过群体智能完成同一个任务。它最主要的能力,是将一个复杂巨大的任务拆分,形成不同的小型任务,多个智能体分别完成,并且相互协调,共同管理。每个智能体是独立的,拥有自己的计算逻辑和智能,他们之间的数据交互是异步通信的。它在独立思考的情况下,也可以和其他成员进行数据和计算模型沟通,相互协调,解决矛盾,最终达到复杂问题解决的一致性。Multi-Agent体系的数据和处理有以下特点:数据和知识具有分散性系统是完全分散的,没有唯一的全局中控节点。智能体都具有独立解决一个任务的能力和知识,可以支持智能体随时加入与退出,它具有良好的模块性,即拔即插,可以将巨大复杂的系统,拆分成多个子任务,降低管理难度和成本。

[0022] 在上述分布式智能节点,可同时运行三种分布式计算体系中的一种或多种(HLA、DDS、Multi-Agent),极大的提高了大规模分布式智能计算的体系,使得任何一种分布式结构在本系统中均能找到其应用架构。在实际应用环节当中,我们需要结合不同的应用场景和计算组合方式,通过开放软件层连接其他人工智能库(比如Tensorflow)进行架构,做到大规模、分布式、单体AI节点计算,做到集群人工智能的总体决策,并且互相激发、互相协作。

[0023] 进一步地,如图5所示,在所述DDS体系中,为了解决数据订阅模块的数据读取问题,所述数据订阅模块包括:数据读取者及订阅者;所述数据读取者,用于读取所述订阅者订阅的数据;所述订阅者,用于向所述DDS体系订阅数据,通过积极的轮询,获取所述数据读取者的新数据。

[0024] 进一步地,为了系统的高度协调性,突破传统人工智能的单一专家系统,形成多个专家系统,所示分布式智能节点适用所述Multi-Agent体系,包括三种组织结构:

[0025] (1)集中式结构,如图6所示,不同的智能体(agent)分为多个群组,每个群组都有一个领队智能体(agent),所述领队智能体(agent)负责协调统一本组智能体(agent)的智能决策,在与其他领队智能体(agent)相互协调,形成最终决策;该结构具有易于管理、易于调度的优势;

[0026] (2)分布式结构:如图7所示,每个智能体(agent)完全平级,没有上下级关系,在本结构中,智能体(agent)的激活和工作,都是受整体外部的系统数据所驱动,它的优势是让整个系统更加灵活,具有较高的自主性;

[0027] (3) 混合式结构:如图8所示,混合了所述集中式结构和分布式结构;在分布式群体智能系统当中,一些复杂的集中化的群体智能结构,可以使用集中式结构来构建,一些微小服务和个体智能,通过分布式结构来工作,再通过中控节点来总体决策。这种结构具有高复杂性和高灵活度等优点。

[0028] 在实际的分布式人工智能系统解决方案中,Multi-Agent结构的系统,具有以下优势:(1) 在此分布式系统中,智能体的独立性,可以很好的解决子问题,也能通过自己影响周围的智能体决策;(2) 此分布式系统不追求单个智能体的复杂性,它在设计理念上,追求多层次、多智能体的架构,降低单个智能体求解的难度;(3) 此分布式系统提供是一个高度协调的系统,由多个智能体协调决策,通过群体智能求解高层的决策方法,通过信息集成,形成复杂的大规模运算架构;(4) 此分布式系统突破了传统人工智能的,单一的专家系统,它可以通过多个专家系统,协调决策,提高决策的适应性和处理能力;(5) 此分布式系统中的智能体是异步且分布的,它可以是一个组织(由多个智能体形成的更大智能体,也可以是小型个体智能体,它们可以是多语言的、多设计模式的);(6) 在此系统中,智能体的处理是异步的,互相之间的进程和数据的协调,可以用不同的算法来进行处理。

[0029] 进一步地,分布式智能节点在开放性网络中,通过与其他分布式智能节点形成联邦分布式体系或DDS分布式体系或Multi-Agent分布式体系,此时不同类型的分布式体系也可作为一个大型分布式智能节点。

[0030] 如果分布式网络存在多个分布式体系下,分布式智能节点可以通过内部线程多任务的方式,兼容多个分布式体系的模型和功能,分布式节点可以支持外部网络分布式和内部线程分布式。当分布式智能节点属于体系的中心节点时,通过线程分布式的方式,在一个节点中形成多体系任务;当分布式智能节点属于体系的分布式节点时,加载各自负责分布式体系的模型文件;中心节点中多体系之间的交互,通过分布式节点中线程通道进行信息的交换和同步;各自的分布式节点,再和外部的节点进行当前体系的模型加载、任务分发、计算层处理等工作。

[0031] 具体而言,在单个分布式智能节点中,通过多线程的方式,进行一个运行进程中的多个内部节点的同步处理;在局域网络或者小规模网络中,根据HLA、DDS、Multi-Agent等体系,形成自己的小型网络联邦,并且组成独立的分布式计算模型。

[0032] 进一步地,分布式智能节点通过连接外置的软件开发工具包,运行具有计算能力的计算程序,形成计算层,所述外置的软件开发包提供适用不同环境下的计算机编程SDK,包括Python、C++、C#、JAVA和图形引擎Unreal。

[0033] 进一步地,如图9所示,为了简化群体智能计算中的数据,对需要由复杂计算程序提供的数据,所述分布式智能节点通过消息协议定义代理智能体,所述代理智能体使用SDK连接运行复杂的计算程序,如图10所示,通过设置拦截器,拦截一些复杂的计算程序中非法数据和不必要的结果,确定需要同步的数据和结果,形成所述复杂计算层,进行计算任务。

[0034] 为什么要采取代理的方式,因为在HLA体系的联邦模型或者DDS分布式体系的IDL模型中,定义的参数和模型方式,相对比较简单一些,而神经网络的模型,或者其他的复杂的智能系统,他们的计算参数和计算方式,或者直接是一个小系统(复杂),相对比较复杂,在整个群体智能计算中,我们实际上不需要那么多的数据同步,或者只需要部分的计算结果数据,或者需要其他系统的合作的方式,那么我们就选择智能委托代理,然后同步和协调

我们的计算层工作。总之,不太方便全部的计算在我们的整个群体智能中进行完全和全部计算的,就可以采用代理的方式进行。

[0035] 为解决上述问题,本发明还提供了一种部署分布式群体智能系统的方法,包括如下步骤,

[0036] (1) 设计分布式体系步骤:分布式体系至少包括,HLA体系、DDS体系以及Multi-Agent体系的一种或多种,分布式体系由分布式智能节点构成;

[0037] (2) 设计分布式模型步骤:设计分布式联邦模型,编写模型任务文件,

[0038] 所述分布式联邦模型包括分布式智能节点的树形结构;数据,计算模型之间的交互方式;以及所述计算模型中的数据结构;

[0039] (3) 设计计算层步骤:根据分布式模型,设计计算层和相关计算层算法模块。

[0040] 所述计算层为利用软件开发包连接到所述分布式智能节点,提供计算能力的计算程序;

[0041] 所述计算层算法为所述计算层算法为所述计算层中计算程序,根据它所申请的联邦模型中的任务对象,所设计的特定的、完成某一能力的计算机程序步骤;计算层中的算法,是为联邦模型中某一个对象进行服务,接收来自整体联邦的数据输入,输出特定联邦模型对象所需要的计算结果,同步到其他联邦节点中。

[0042] (4) 设计同步与异步步骤:设计所述计算层的协调接口、数据一致性算法、数据类型、同步和异步方式;

[0043] (5) 部署节点步骤:根据设计的分布式群体智能体系,部署分布式智能节点;

[0044] (6) 启动节点步骤:启动联邦中央节点,启动其他分布式智能节点,组成分布式联邦网络;

[0045] (7) 计算层连接步骤:根据设计的计算层,发布所述分布式联邦模型,使用SDK编写计算层算法模块,并且连接到对应的分布式智能节点。

[0046] (8) 运行系统步骤:运行所述分布式群体智能系统,启动中控系统,监听运行环境,在使用中迭代更新智能参数、调整所述分布式群体智能系统。

[0047] 进一步地,在所述设计同步与异步步骤后,还包括设计代理层步骤:定义代理智能体的属性,用于连接负责的计算层,代理智能体的数据同步和过滤。

[0048] 进一步地,在所述计算层连接步骤中,当所需要的计算程序为复杂程序时,使用特定的SDK编写委托代理智能体模块,通过SDK建立所述代理智能体与复杂的计算程序的连接,通过设置拦截器,拦截一些复杂的计算程序中非法数据和不必要的结果,确定需要同步的数据和结果形成复杂计算层,所述分布式智能节点通过所述SDK连接所述复杂计算层,进行计算任务。

[0049] 应当理解的是,以上的一般描述和后文的细节描述仅是示例性和解释性的,并不能限制本公开。

## 附图说明

[0050] 此处的附图被并入说明书中并构成本说明书的一部分,示出了符合本发明的实施例,并与说明书一起用于解释本发明的原理。

[0051] 图1为Tensorflow单机单卡与多级多卡部署模式示意图

- [0052] 图2-1为Spark计算模型和Stage形成的DAG图
- [0053] 图2-2为Spark分布式并行计算分层示意图
- [0054] 图3为分布式智能节点多分布式体系兼容示意图
- [0055] 图4为HLA联邦体系示意图
- [0056] 图5为DDS数据订阅发布体系示意图
- [0057] 图6为Multi-Agent体系集中式结构
- [0058] 图7为Multi-Agent分布式结构
- [0059] 图8为Multi-Agent混合式结构
- [0060] 图9为委托代理关系图
- [0061] 图10为委托代理方式示意图
- [0062] 图11为实施例1示意图
- [0063] 图12为实施例2示意图
- [0064] 图13为分布式机器学习系统结构图
- [0065] 图14为RTOSNode的三层原理示意图
- [0066] 图15为多智能体分布式体系示意图

### 具体实施方式

[0067] 为更容易理解本发明的优点、特征以及达到技术效果的技术方法将参照例示性实施例进行更详细地描述,且本发明可以不同形式来实现,故不应被理解为本发明仅限于此处所陈述的实施例。相反地,对本领域的技术人员,所提供的实施例将更加透彻、全面且完整地传达本发明的范畴,且本发明将以申请专利文件的权利要求确定保护范围。

#### [0068] 实施例1

[0069] 如图11所示,这个架构中属于典型的HLA体系架构,其中联邦1计算中心1和联邦2计算中心2分别为两个联邦的联邦中心节点;在联邦1中,组织计算单元1和组织计算单元2属于联邦成员,摄像头作为实体计算单元属于对象,每个分布式智能节点通过安装本系统,相互连通,交互。系统运用不同的监控摄像头,形成自有的小型组网,再通过组网结合,形成更大的组织网络。组织网络和中心化云平台结合,充分发挥摄像头的计算能力。系统通过并行计算识别任务,通过组网共享机器学习模型、视频编解码计算资源和共享数据。再通过组网计算层归纳、总结、决策,和中心化云服务交互,形成群体智能决策,还可以减少中心化云服务的计算和带宽压力,有效提高整个系统的计算能力,极大地减少系统成本。

[0070] 多个摄像头之间,通过分布式智能系统,主要解决了分布式人工智能的几个问题:

[0071] (1) 不同区域的摄像头,可以通过自组局域网络,形成不同区域的组网。(2) 组网之间的摄像头,都有自己的独立计算层,支持机器学习和视频编解码。(3) 不同摄像头可以通过分布式智能系统,在组网之间并行计算,共享计算模型和数据。(4) 组网和组网之间,也可以通过分布式智能系统并行计算,共享组网之间的计算模型和数据。

[0072] 通过HLA联邦对象模型的定义,与AI模型结合形成分布式AI智能模型。

#### [0073] 实施例2

[0074] 如图12所示,这是典型的DDS体系,一个平面地形环境,协调多实体(多人、多车辆等情况)的计算,包括AI模拟、AI协调(交通协调)等方式的架构。

[0075] 架构把整个地形分为4个区域,不同区域发布不同的DDS主题,每个区域的计算平台单元是主题,不停实体计算单元是发布者或者订阅者,相关的计算实体(车辆)可以通过各自的区域的主题,进行分布式数据的同步、模型计算。

[0076] 进一步地,因为分布式智能系统也支持其他体系,也可以通过分布式智能系统和其他主题同步和并行计算,实现分布式的群体AI决策。

[0077] 实施例3

[0078] 在分布式群体智能系统中,通过软件开发包(SDK)与智能体计算层的结合,支持多计算实体与智能体的协调操作,通过分布式智能节点作为Multi-Agent共享计算模型和数据的消息路由通道。

[0079] 分布式群体智能系统将会支持tensorflow与智能体的异步计算和数据同步机制,支持分布式机器学习,做到多智能体间的协作机制。具体的分布式机器学习系统结构图,如图13所示:

[0080] 其中常用的机器学习算法类别,比如归纳学习、类比学习、强化学习、分析学习、遗传算法等。每种算法都利用一个智能体处理,通过分布式模型任务拆分的方式,让多智能体之间独立工作,数据和计算模型相互协调。在此分布式系统中的智能体是异步和分布的,它可以是一个组织(由多个智能体形成的更大智能体,也可以是小型个体智能体,它们可以是多语言的、多设计模式的。Multi-Agent之间的协作体系结构,主要有以下几种:

[0081] 智能体间的网络,将所有的智能体通过分布式智能节点连接成整体的网络,不管远近距离的,都直接通信,两两之间的数据和计算模型,都可以互相获取和影响,如17图中所示的监督智能体与执行智能体。

[0082] 黑板结构,联盟系统,如13图中所示的圈内的各种学习算法形成一个联盟,通过联盟和联盟之间的数据进行影响,各个联盟之间,通过局域网进行协调。联盟对外,只开放局部数据和计算模型。

[0083] Multi-Agent是一种重要的分布式人工智能体系,分布式群体智能系统中会重点支持。这种计算模式,大量应用在航天、生物细胞、环境模拟、经济体决策等大规模人工智能计算领域。

[0084] 实施例4

[0085] SintolRTOS系统的RTOSNode节点

[0086] 此系统中的RTOSNode即为本发明分布式智能节点的一种,在SintolRTOS的分布式节点原理的理解上,可以有三种理解方式:

[0087] 一、在单个RTOSNode中,通过多线程的方式,进行一个运行进程中的多个Node同步处理;

[0088] 二、在局域网络,或者小规模网络中,根据HLA、DDS、Agent等体系,形成自己的小型网络联邦,并且组成独立的分布式计算模型;

[0089] 三、在开放性网络中,不同联邦、区域计算的Node节点,通过开放性网络与其他联邦和区域代表形成更大的分布式模型。在这种网络中,可以把一个联邦、一个区域计算看作一个大型Node节点。

[0090] 如图14所示:Abstract Node:抽象了节点相关的功能,包括数据队列、异构处理等。AbstractServer:抽象了相关RTOSNode节点,作为分布式节点服务的功能,包括连接、操

作、数据推送、消息分发、数据队列、数据压缩等。Thread Sever:作为第一层的分布式节点,继承了AbstractServer,实现了节点功能,提供了RTOSNode单节点内部的分布式结构,可以通过进程间的管道进行通讯,充分发挥单节点的多CPU计算核心。RTOSNode:作为Sinto1RTOS中分布式节点的标准单位,通过分布式联邦模型进行数据同步、并行计算、智能决策,通过小规模网络,在一个联邦集团中,形成小规模集团计算。联邦网络:通过RTOSNode中的联邦模型,和多个其余的联邦单位,进行多联邦模型之间的协调计算。这个部分可以通过互联网,或多网络结构进行路由通信。

[0091] 关于Multi-Agent形式,Agent的形式多种多样,在Sinto1RTOS中,主要作为智能代理模型的形式,与FED、IDL模型等协作,将不同智能系统代理起来,与其他代理系统协同工作。Sinto1RTOS提供了Protobuf作为消息协议体,用来编写Agent的智能代理属性,多语言、多平台地支持各种智能系统协同。

[0092] Multi-Agent没有具体的定义方式,通过消息协议,我们只需要定义对外的委托人,然后通过委托互相调用和协调各自的分布式智能系统。

[0093] 在Sinto1RTOS中,通过Agent调用Modle的体系,主要有三个部分:目标对象、拦截机对象、目标对象代理。如21图所示:目标对象就是联邦智能模型中的联邦对象;拦截机对象就是拦截机负责拦截联邦模型中的一些无用或者非法数据;目标代理对象就是Agent联邦代理负责代理联邦智能模型中的某一个联邦对象。

[0094] 在Sinto1RTOS中,在RTOSNode之间,通过TCP连接,并且通过Protobuf实现Agent代理信息和协议的编写;在Model值上,实现Agent的拦截以及和Model层的链接和调度;Model中定义FED联邦模型、DDS主题模型等开放了代理接口,对于非法调度,拦截层应该进行拒绝和错误反馈,实现Agent代理。

[0095] 实施例5

[0096] 定义实施例的AI模型和分布式计算体系。在本实施例中,我们需要使用多个AI人物互相对抗,它们拥有四种状态:原地监视、寻路找人、跟随活动和丢失目标。在运动过程当中,AI智能体需要绕开障碍,或者跳跃通过障碍。在工作过程中,智能体需要不断训练,不断提升,智能体之间的运算模型和数据都需要通过Sinto1RTOS来进行协调运算和群智处理。

[0097] 定义联邦计算模型

[0098] 联邦定义模型,需要定义在整个联邦中出现对象类和交互接口,在本实施例中,主要是智能体的属性,包括位置、速度、方向、运动状态、状态机参数等。智能体的交互,包括跟随、停止、跳跃等。根据这些计算,我们定义了联邦计算模型multiAI.xml。

[0099] 定义强化学习DQN神经网络模型的智能代理Agent

[0100] Agent描述了在Tensorflow中,DQN(强化学习神经网络)中计算的智能实体,它所计算的参数需要根据神经网络的参数表进行修正,但是数据的输入和输出都需要转化,和MultiAI中定义的智能体一致。

[0101] 每个Agent运算了一个DQN神经网络,如何将多个计算实体结合起来,形成一个规模化的分布式神经网络,还需要使用Sinto1SDK去接入Agent,在Fed联邦中实例化多个AI计算实体,这里我们使用PSinto1SDK,来接入Tensorflow的Python计算环境。

[0102] 重构联邦实体的处理类

[0103] 在Sinto1SDK创建或者加入联邦时,需要传入联邦实体的处理类,用于接收回调

callback信息,包括其他实体的数据、联邦模型和实体的改变等。

[0104] DQN神经网络与PSintolSDK构建计算层

[0105] 编写完联邦处理类以后,再通过PSintolSDK,发布联邦、加入联邦、同步联邦、订阅AI实体类和生成发布实体类,并且监听和订阅联邦中其他计算实体的数据和模型,接受来自联邦的计算数据。

[0106] 在DQN计算层中,Agent的训练模型和数据,需要来自于实体数据的传入和代理,驱动DQN强化神经网络的学习训练的计算层,并且在实际应用中对抗博弈。

[0107] 实施例6

[0108] 在本实施例中,分布式体系网络结构如图15所示,主要体现了在Sintol RTOS的分布式集群中,多智能体在同一个联邦下,进行协作、对抗、寻路、寻人、学习和强化。对此需要设计联邦体下的联邦结构、分布式体系、物理仿真计算层、智能DQN神经网络计算层、联邦节点等。其中整体体系都在一个联邦体系下,所有智能体都在一个联邦模型中,作为一个模型实体,进行运算和协调。每个智能体通过UnrealRTOS和CSintolSDK组成实体,处理它的仿真、状态机、物理碰撞和图形渲染。智能体通过PSintolSDK与Agent形成智能代理机制,智能体通过DQN神经网络进行强化学习,形成不同智能体的智能对抗训练和升级。整个联邦可以容纳多个智能体,它们共同形成了这个小型联邦的群体智能。

[0109] 本实施例使用了Unreal引擎,进行图形化场景和智能体的可视化开发。对此,SintolRTOS提供了Unreal使用的SDK和使用案例UnrealRTOS工程,我们将用它和CSintolSDK搭建多智能体对抗的仿真演练场景。

[0110] 运行UnrealRTOS多智能体进行联邦对抗

[0111] 运行2个智能实体,当没有接入Agent代理智能决策层时,几个智能实体,在场景中会各自根据状态机算法和寻路判定算法,随机的进行寻找、跟随其他智能体、越过障碍等行为,以上智能体是根据联邦模型和状态机进行运行的,运行久了以后,可以发现智能体实际上在寻找其他智能体的情况上,没有AI能力的提高。当我们使用PSintolSDK和Agent运行DQN强化神经网络,经过多智能体的互动和博弈训练,经过长时间的训练,会发现智能体在寻找其他联邦对象和寻路能力上,得到了极大的提高,最终形成群体智能队列的共同行为,最终提高决策能力。

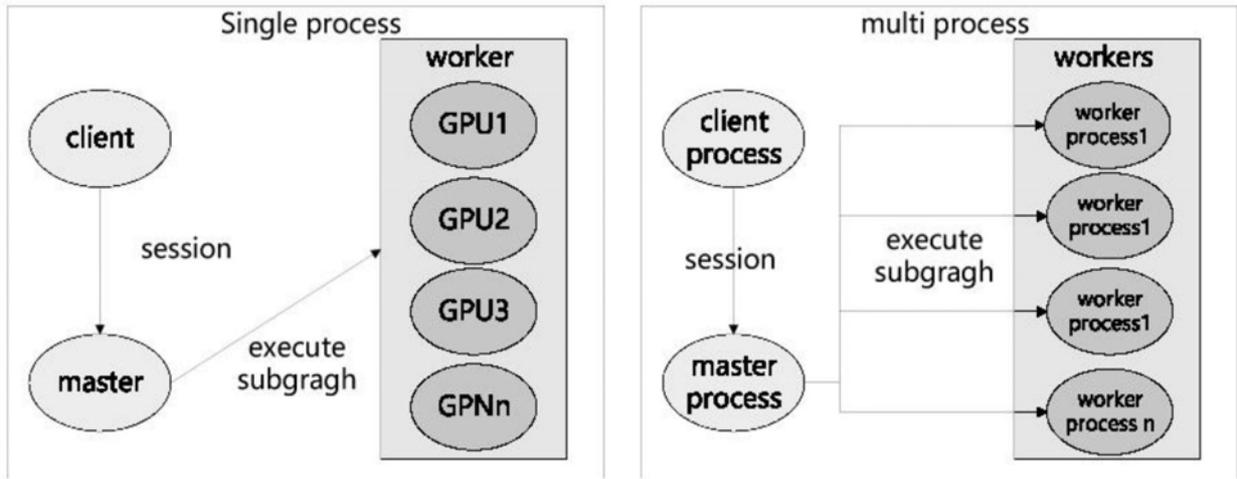


图1

Stage\_3

AI Modle

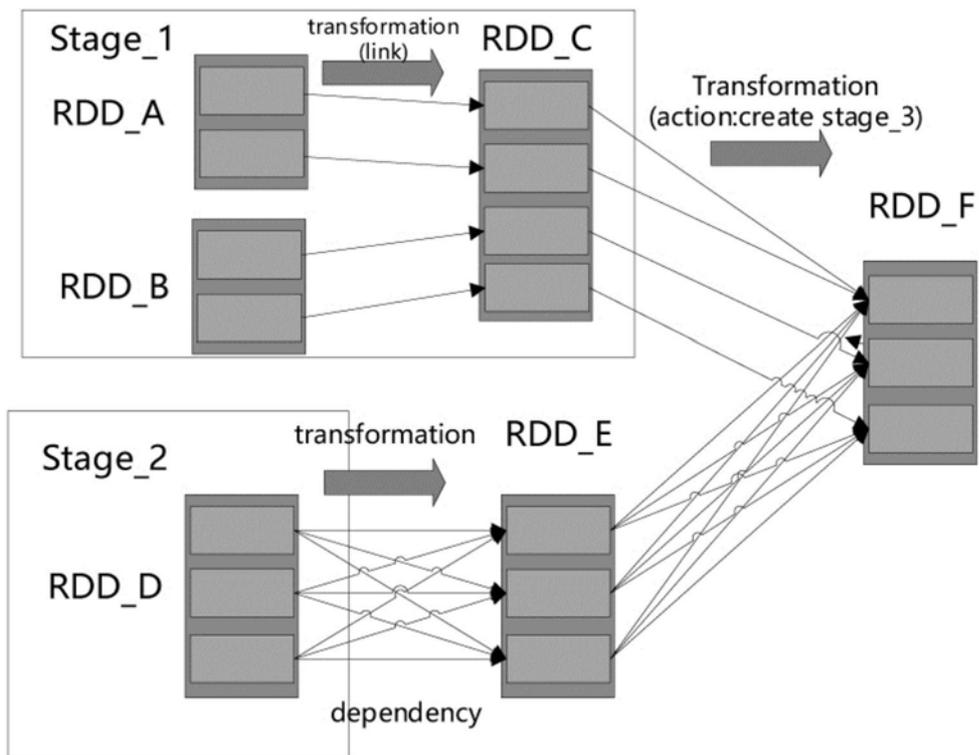


图2-1

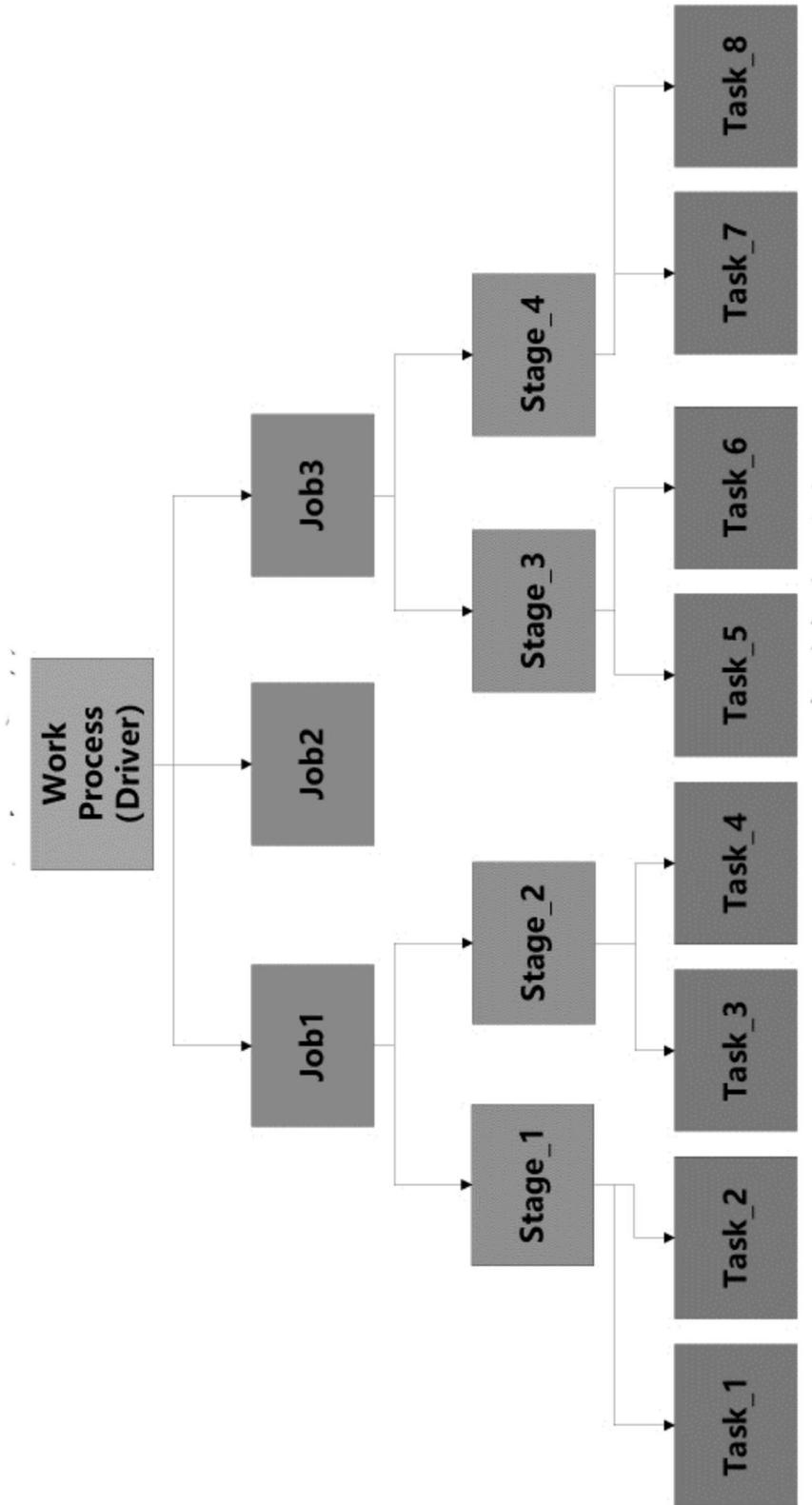


图2-2

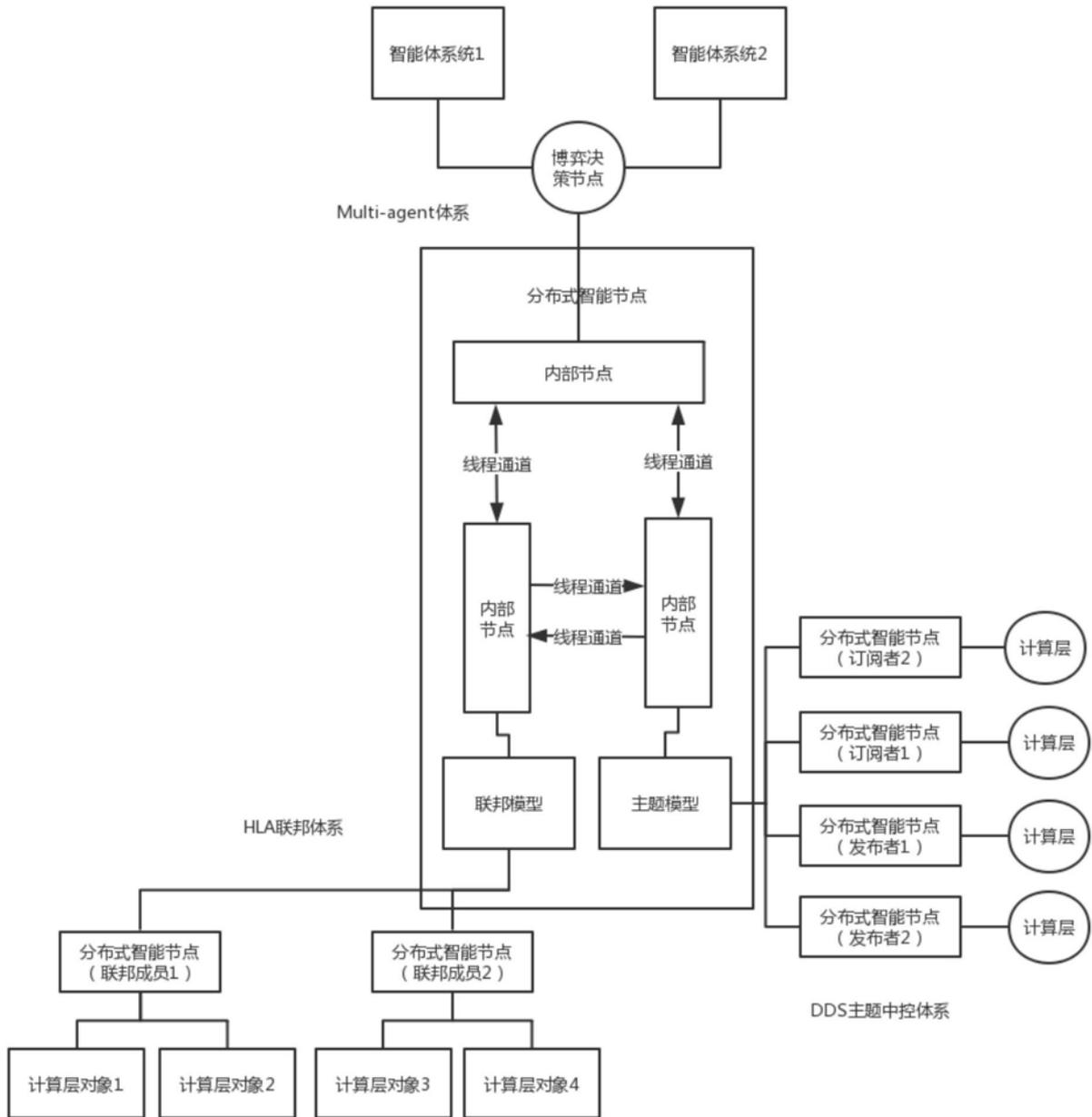


图3

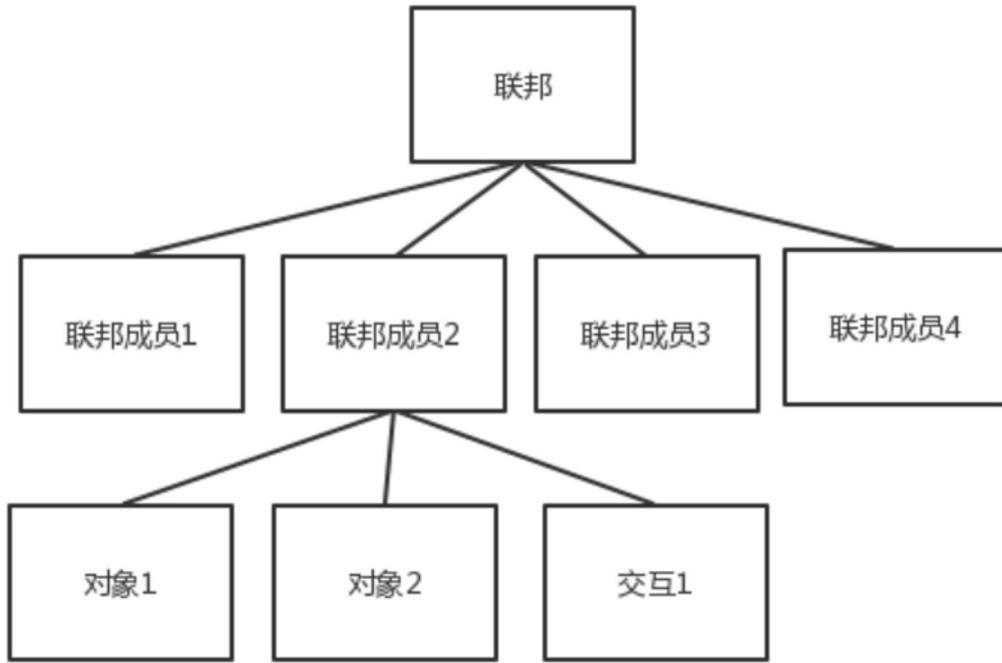


图4

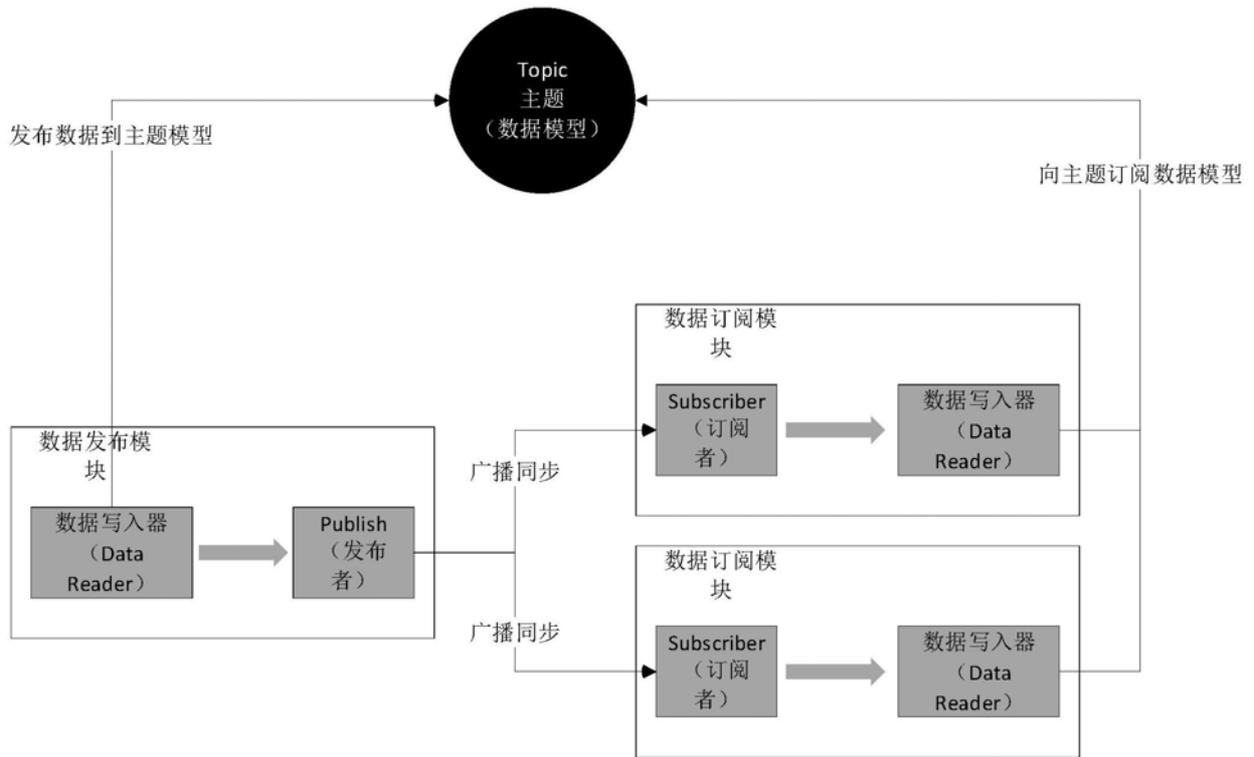


图5

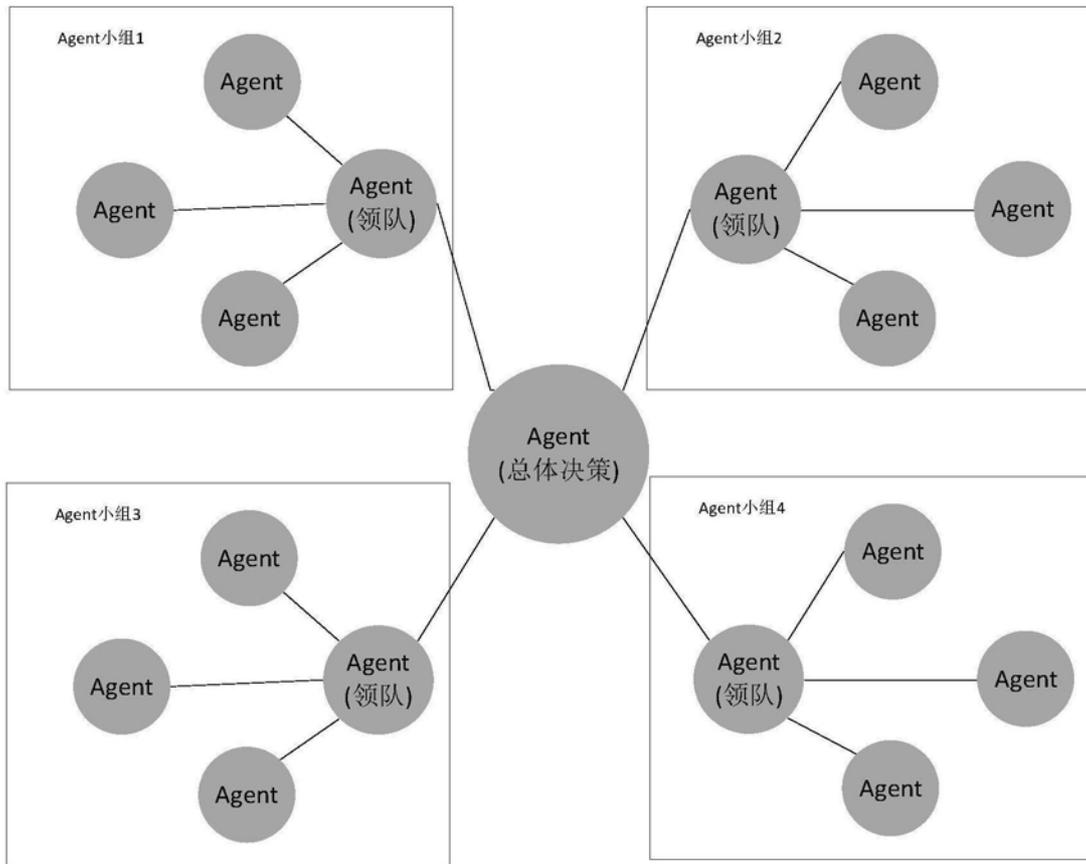


图6

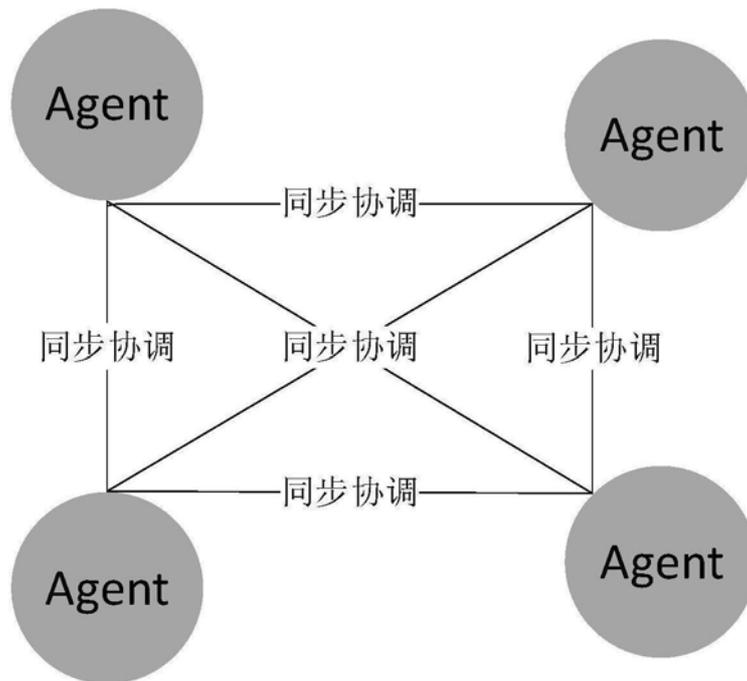


图7

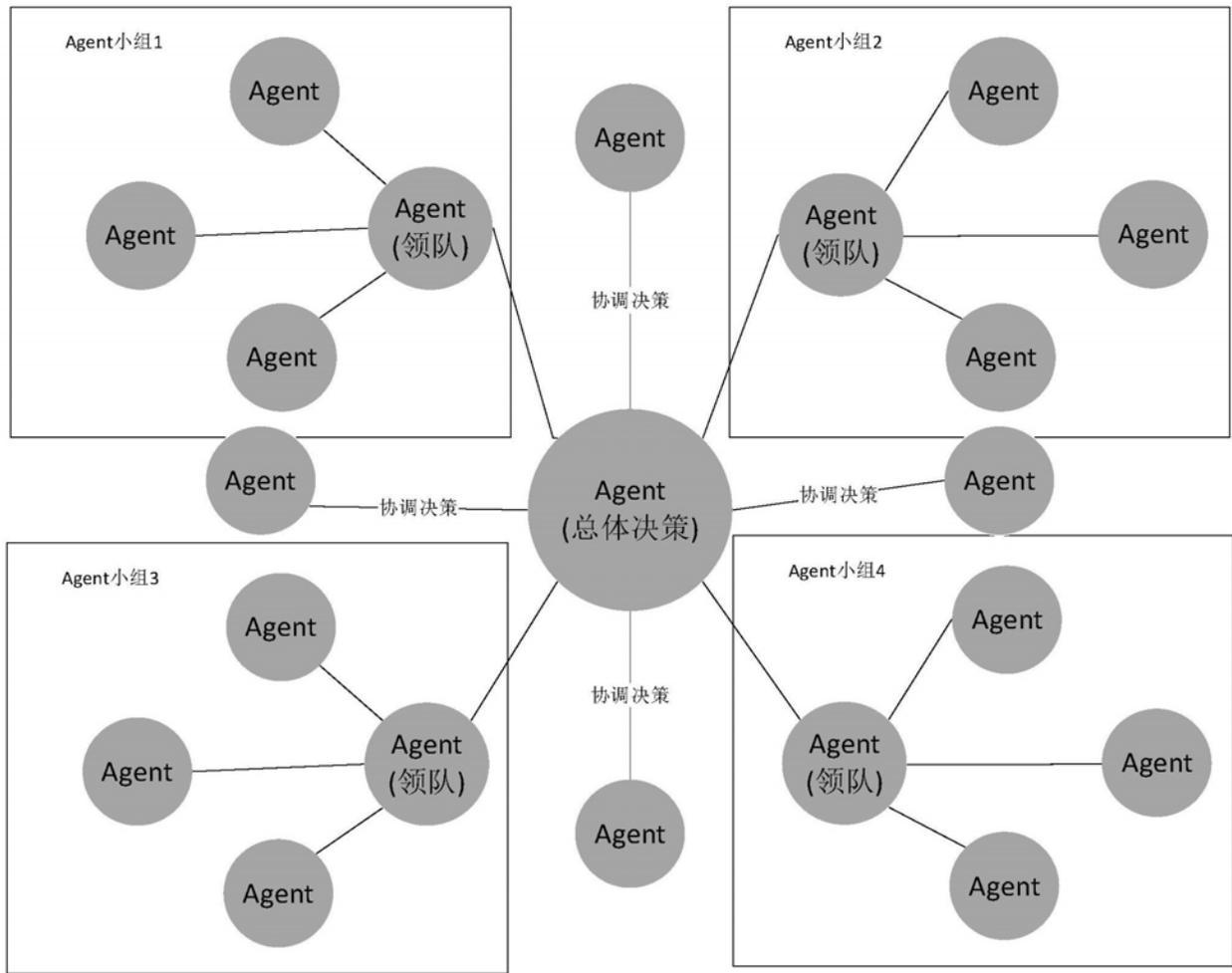


图8

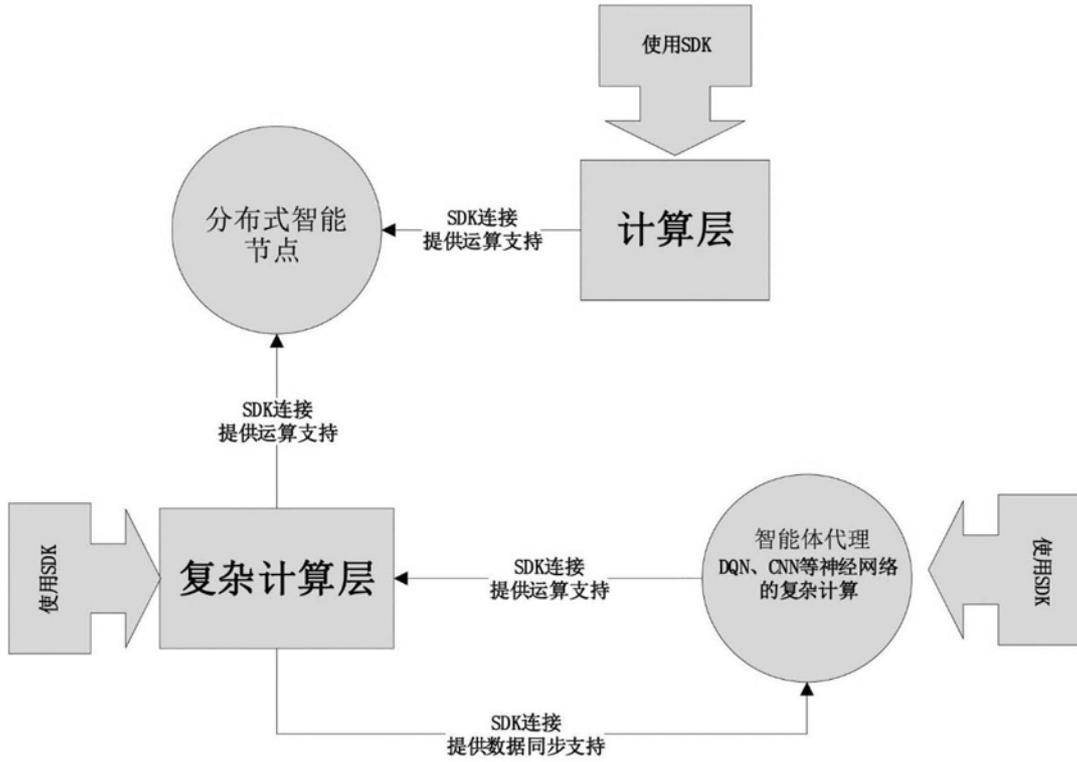


图9

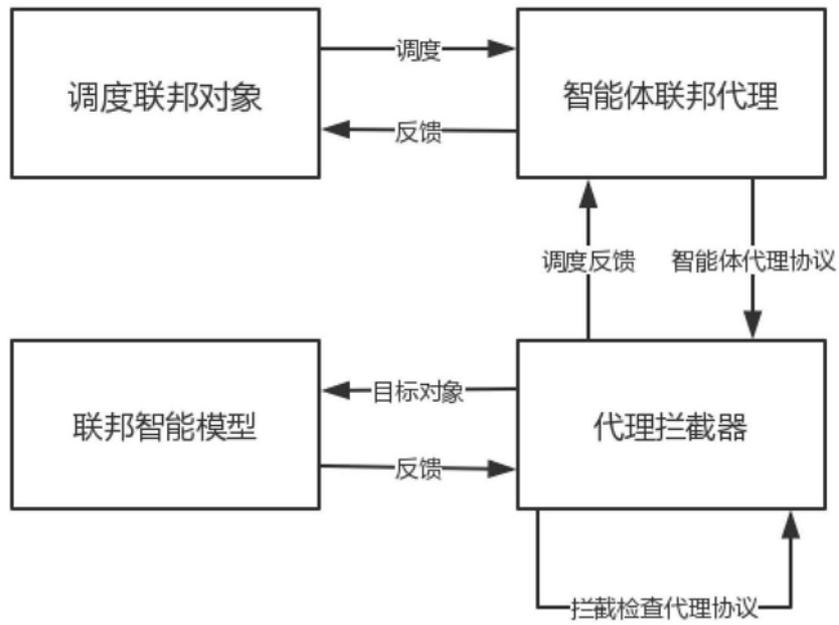


图10

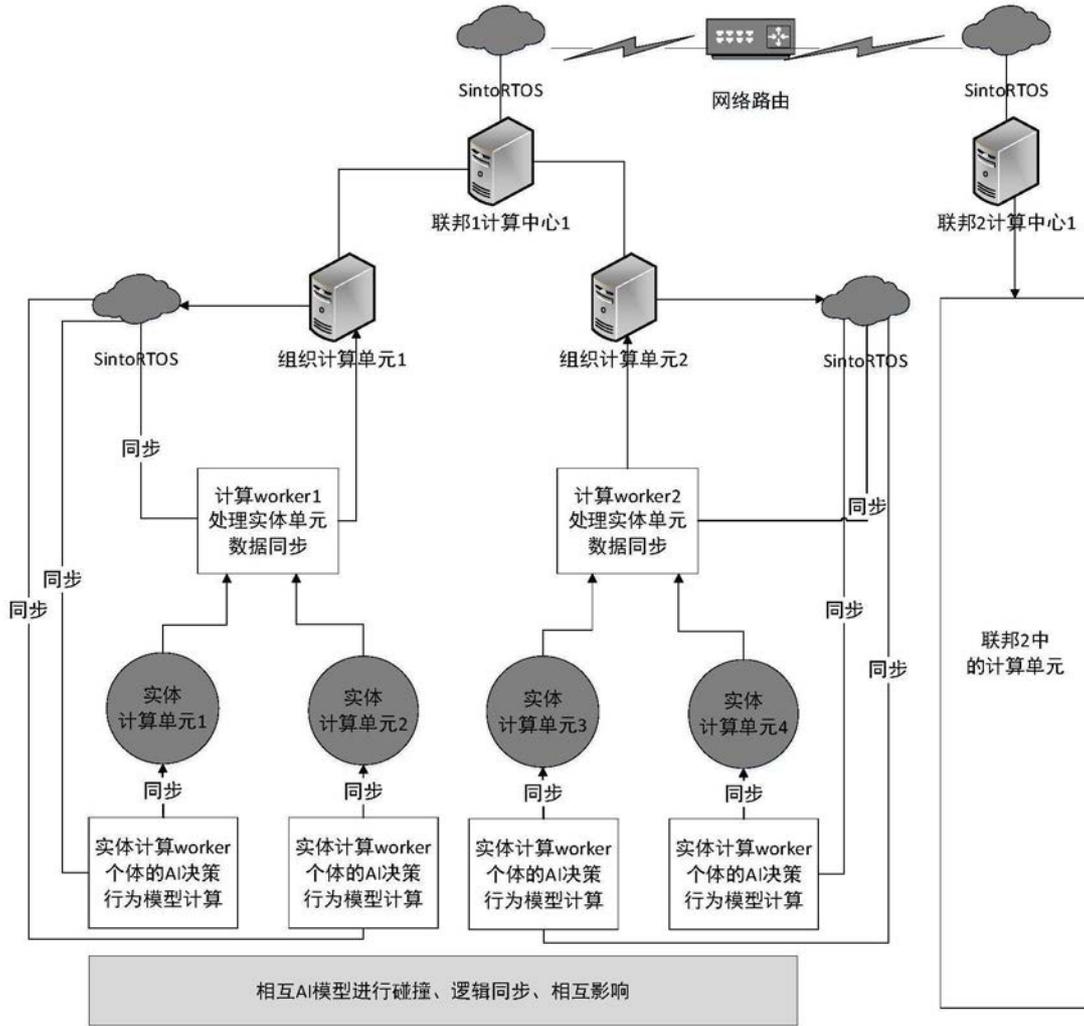


图11

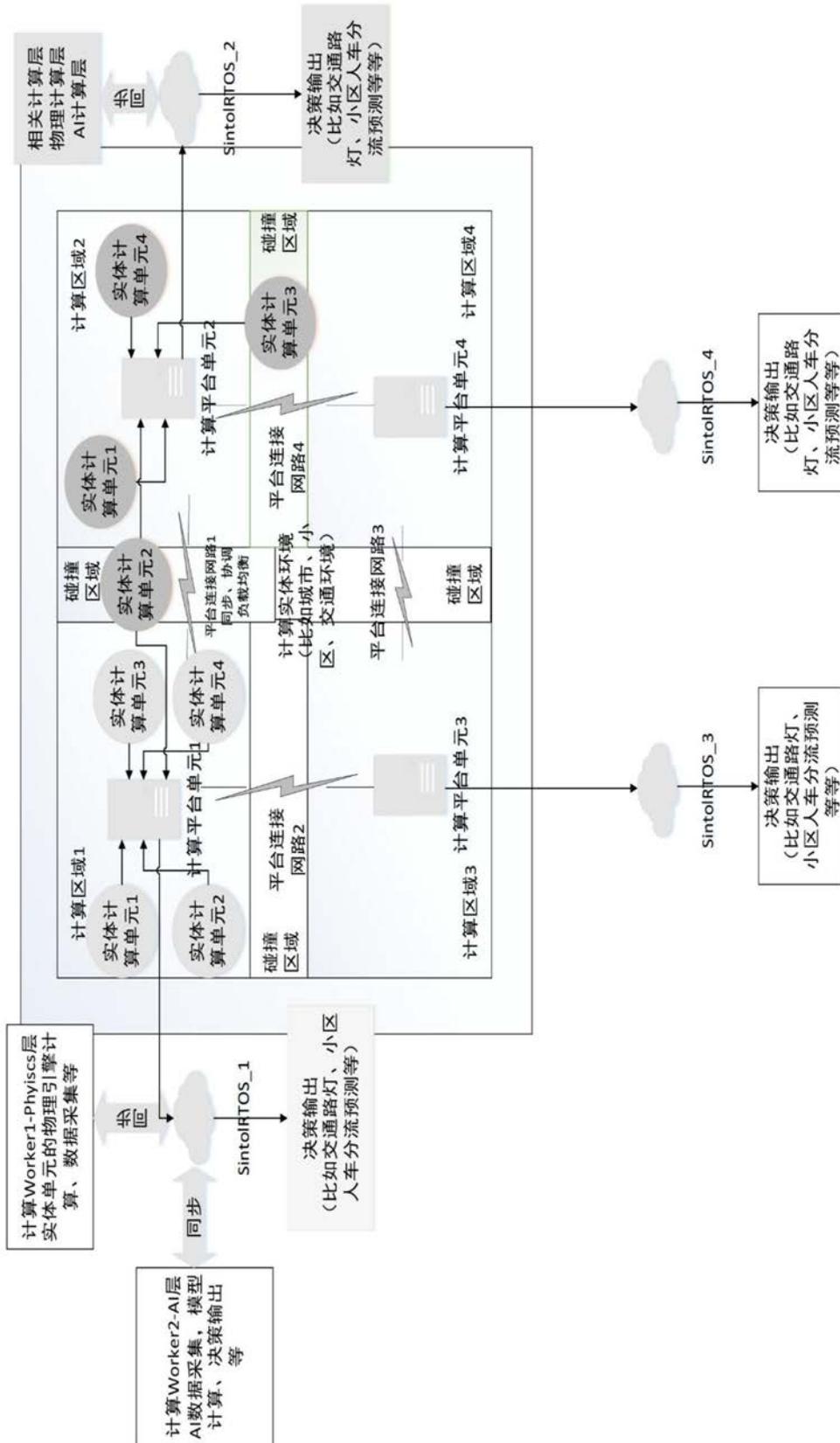


图12

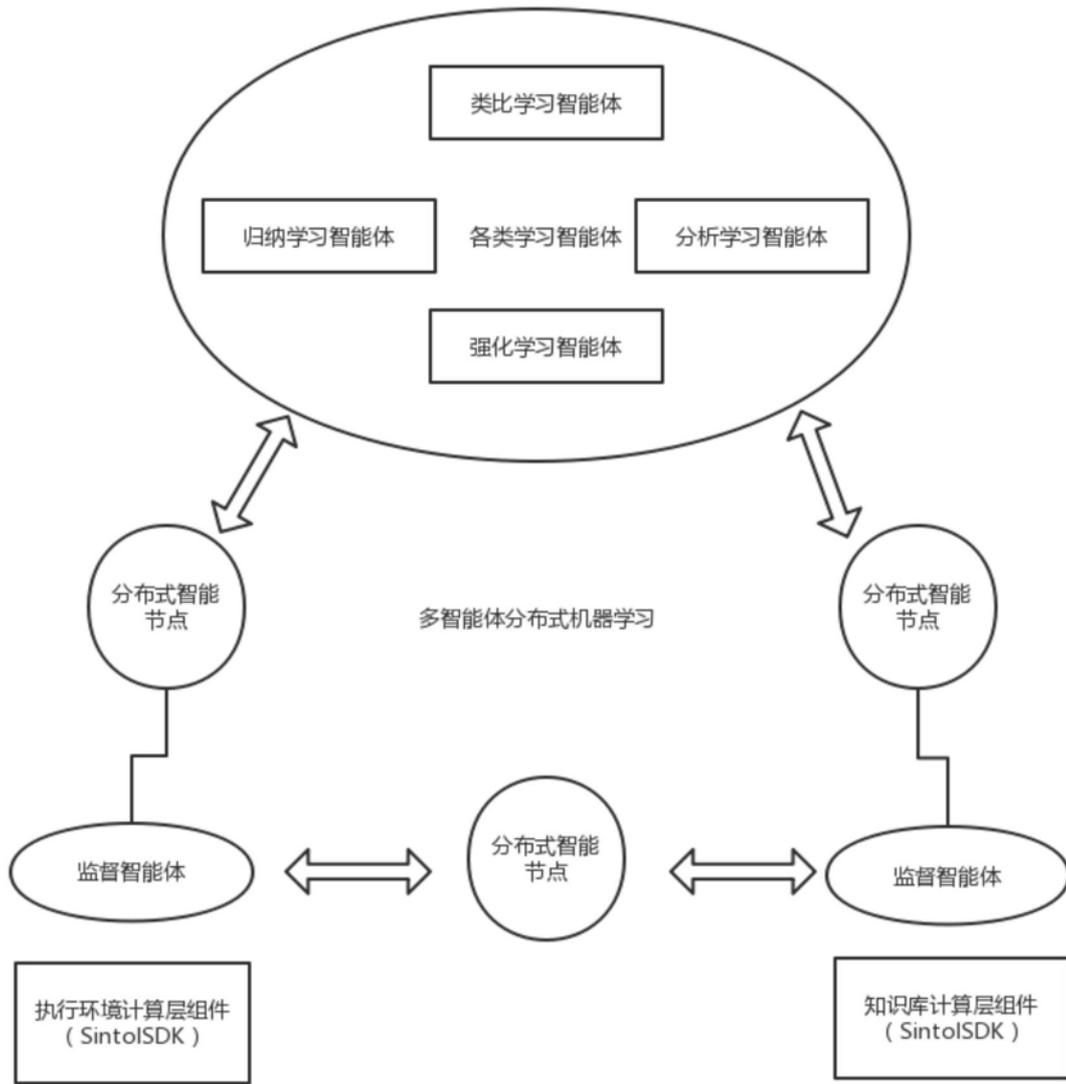


图13

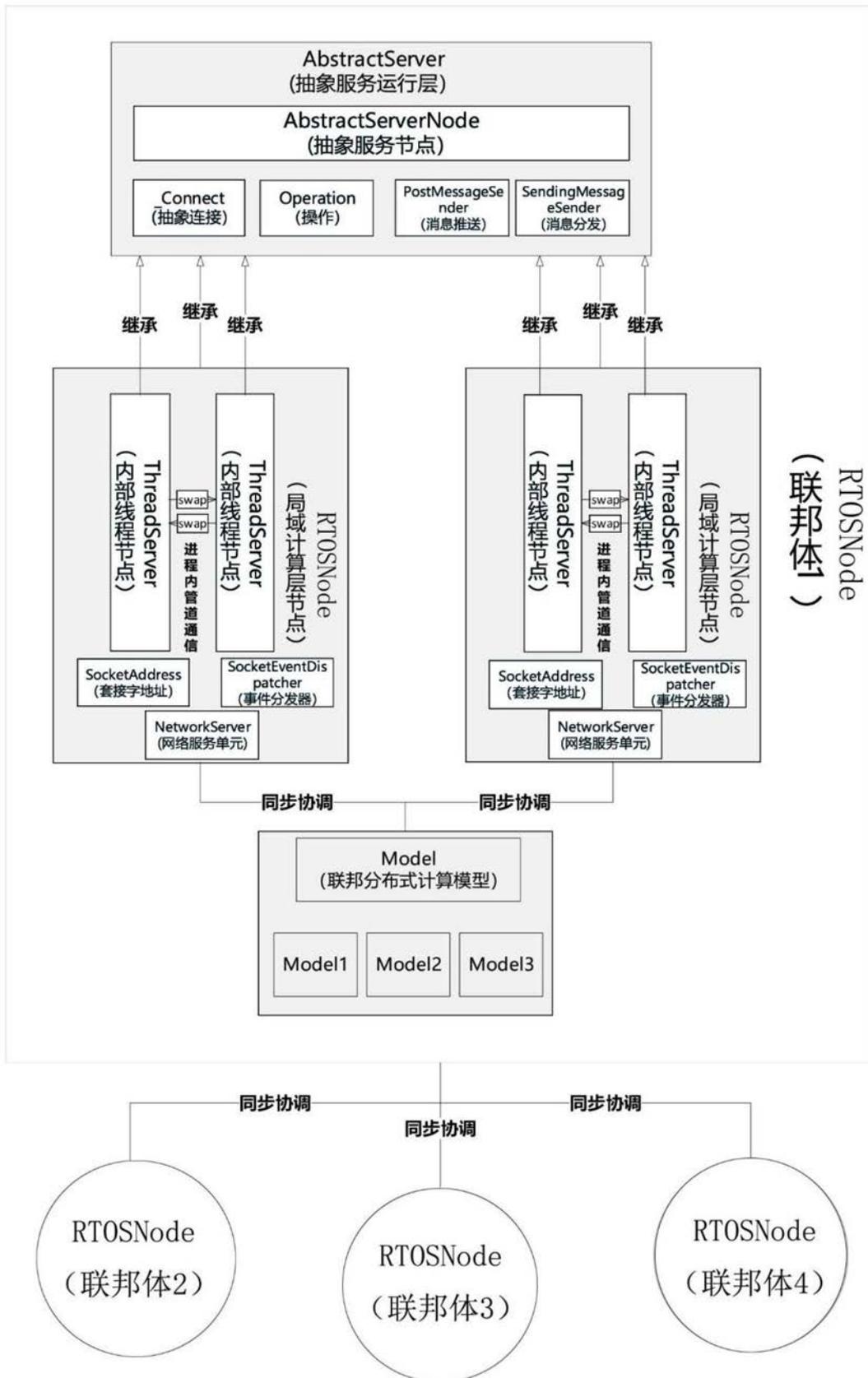


图14

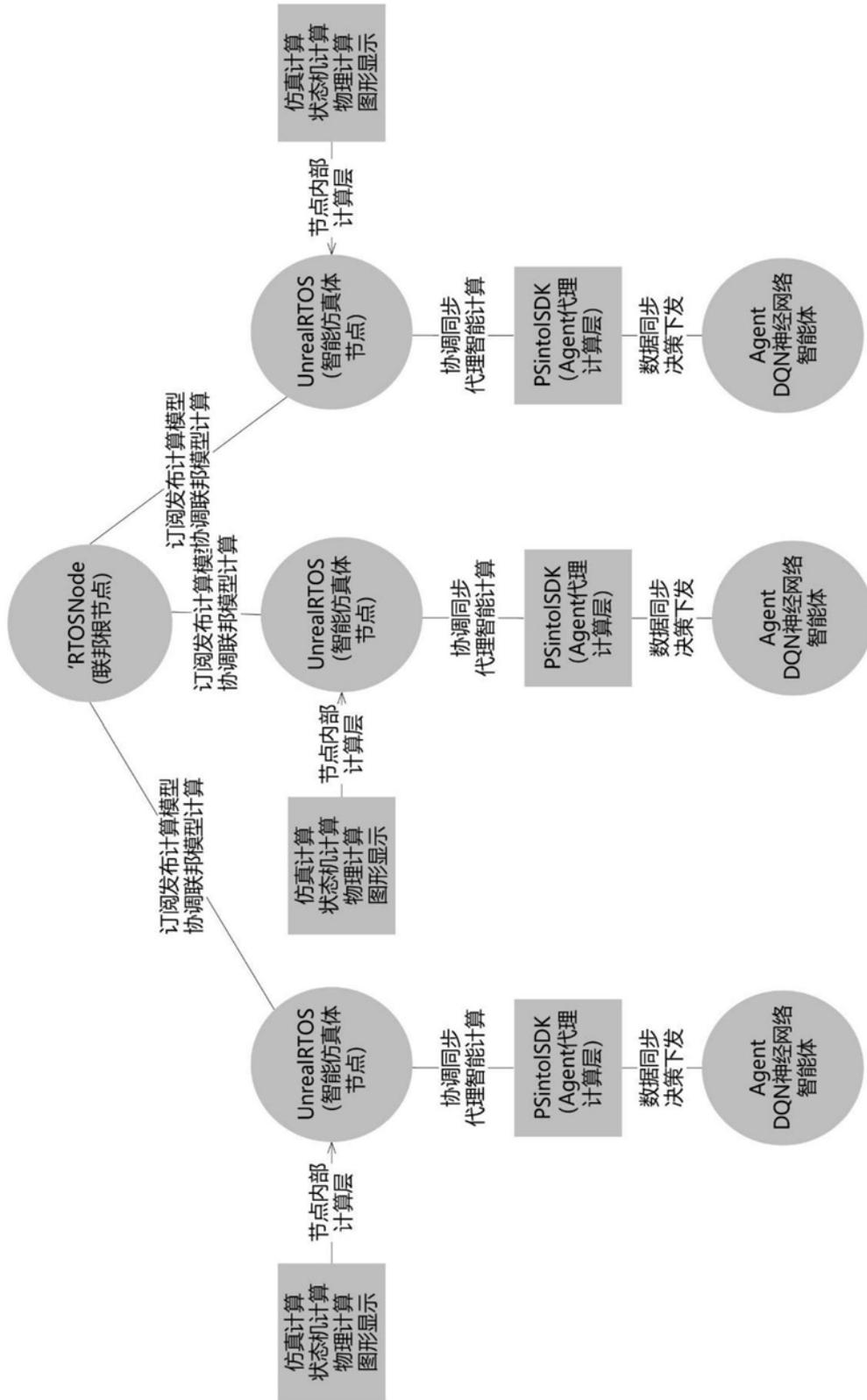


图15