



(12) 发明专利

(10) 授权公告号 CN 116962425 B

(45) 授权公告日 2024. 05. 28

(21) 申请号 202310964768.X

H04L 67/145 (2022.01)

(22) 申请日 2023.08.02

H04L 67/1095 (2022.01)

(65) 同一申请的已公布的文献号

G06F 8/34 (2018.01)

申请公布号 CN 116962425 A

G06F 8/38 (2018.01)

(43) 申请公布日 2023.10.27

(56) 对比文件

(73) 专利权人 杭州和利时自动化有限公司

CN 111880904 A, 2020.11.03

地址 310018 浙江省杭州市经济技术开发区

CN 108365979 A, 2018.08.03

区12号大街M10-15-7地块

CN 109428768 A, 2019.03.05

(72) 发明人 朱珂 费盼峰

CN 116193481 A, 2023.05.30

(74) 专利代理机构 北京集佳知识产权代理有限公司

US 2018006878 A1, 2018.01.04

11227

US 2021311453 A1, 2021.10.07

专利代理师 张影

WO 2020047780 A1, 2020.03.12

审查员 朱晓旭

(51) Int. Cl.

H04L 67/104 (2022.01)

H04L 67/141 (2022.01)

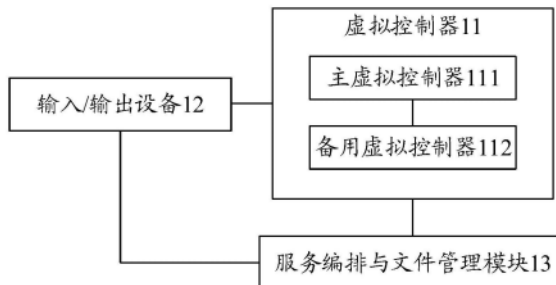
权利要求书2页 说明书11页 附图6页

(54) 发明名称

一种控制器虚拟化控制系统、方法、存储介质

(57) 摘要

本申请公开了一种控制器虚拟化控制系统、方法、存储介质,涉及通信技术领域,包括:主虚拟控制器,用于调用通用服务器的资源与位于第一独立网段中的输入/输出设备进行点对点网络通信;备用虚拟控制器,用于调用通用服务器的资源与位于第二独立网段中的输入/输出设备进行点对点网络通信;通过交叉心跳线连接的主虚拟控制器和备用虚拟控制器,用于实现主虚拟控制器和备用虚拟控制器状态互检和信息同步;服务编排与文件管理模块,用于基于控制逻辑并通过组态设置在主虚拟控制器、备用虚拟控制器、输入/输出设备之间建立数据连接,在各个设备之间进行数据传输。实现通用计算机的虚拟控制器的部署以及能够同时运行常规简单逻辑和智能算法块。



1. 一种控制器虚拟化控制系统,其特征在于,所述控制器虚拟化控制系统包括:包含布置于第一独立网段的主虚拟控制器、布置于第二独立网段的备用虚拟控制器的虚拟控制器、通过第一网口、第二网络分别连接第一独立网段和第二独立网段的输入/输出设备、位于通用PC机的服务编排与文件管理模块,其中,

所述主虚拟控制器,用于调用所述通用服务器的资源与位于所述第一独立网段中的输入/输出设备进行点对点网络通信;

所述备用虚拟控制器,用于调用所述通用服务器的资源与位于所述第二独立网段中的输入/输出设备进行点对点网络通信;

通过交叉心跳线连接的所述主虚拟控制器和所述备用虚拟控制器,用于实现所述主虚拟控制器和所述备用虚拟控制器进行状态互检和信息同步;

所述服务编排与文件管理模块,用于基于控制逻辑并通过组态设置在所述主虚拟控制器、所述备用虚拟控制器、所述输入/输出设备之间建立数据连接,以便在所述主虚拟控制器、所述备用虚拟控制器、所述输入/输出设备之间进行数据传输;

所述虚拟控制器,包括:

控制器算法应用库,对预设智能算法块进行了封装供脚本指令解析器调用;

脚本指令解析器,用于对自定义控制器指令行进行解析并进行直接机器执行或调用相应的所述预设智能算法块并执行;

通信单元,用于处理实时数据库单元与外部节点的数据通信,根据组态动态生成订阅、发布任务;

实时数据库单元,用于对输入/输出设备及所述虚拟控制器的内部数据进行内存存储和硬盘持久化;

任务调度单元,用于将组态中的计算任务按照优先级顺序和时间要求进行调度,并调用操作系统的调度器进行任务调度;

组态文件解析子模块,用于对工程师站发送的组态文件进行解析,并形成虚拟控制器内部指令;

校时单元及管理单元,用于通过GPS时间信号进行内部时钟的校准,并对虚拟控制器的状态进行监控和诊断,形成日志。

2. 根据权利要求1所述的控制器虚拟化控制系统,其特征在于,所述虚拟控制器,包括:

数据连接配置单元,用于查询所述虚拟控制器的所有数据资源并配置相应的数据连接,以便将所述虚拟控制器的数据缓存区的数据引用或发送至对应的输入/输出设备和控制器数据区。

3. 根据权利要求1所述的控制器虚拟化控制系统,其特征在于,所述服务编排与文件管理模块,包括:

服务编排子模块,用于可视化编辑以生成用于对所述控制器虚拟化控制系统软硬件的灵活配置的工程组态文件。

4. 根据权利要求3所述的控制器虚拟化控制系统,其特征在于,所述服务编排子模块,包括:

参数配置单元,用于对硬件配置、通信配置、设备连接和系统参数进行参数配置;

算法配置单元,用于通过算法块表示和算法脚本形式表示来实现控制算法文件的表

示;

数据库文件配置单元,用于对实时数据库、历史数据库、配置数据进行库文件配置;
图形界面配置单元,用于对操作界面、监视界面、报警界面以图形化界面的形式配置;
信息记录单元,用于记录包含运行日志、时间记录、性能指标的数据信息。

5. 根据权利要求3所述的控制器虚拟化控制系统,其特征在于,所述算法配置单元,包括:

界面设计单元,用于基于图形界面的工具组件提供可视化编排和解析工具;

语言设计单元,用于定义预设智能算法块的属性和方法;

有向图设计单元,用于基于有向图的数据结构表示所述预设智能算法块之间依赖关系。

6. 根据权利要求5所述的控制器虚拟化控制系统,其特征在于,还包括:

编排单元,用于通过用户端利用的输入设备将预设智能算法块拖拽至画布区,并基于多线程进行分区表示,获取多个目标分区,然后通过连线连接各个目标分区并对所述目标分区中的预设智能算法块进行属性设置;

文件解析单元,用于根据所述组态文件创建并行多线程,将输入数据流注入到预设智能算法块中;根据所述依赖关系,按照拓扑排序的顺序依次执行预设智能算法块,并将执行结果输出到相应的虚拟控制器内。

7. 一种控制器虚拟化控制方法,其特征在于,包括:

通过主虚拟控制器调用通用服务器的资源与位于第一独立网段中的输入/输出设备进行点对点网络通信;

通过备用虚拟控制器调用通用服务器的资源与位于第二独立网段中的输入/输出设备进行点对点网络通信;

通过交叉心跳线连接的主虚拟控制器和备用虚拟控制器实现主虚拟控制器和备用虚拟控制器进行状态互检和信息同步;

基于控制逻辑并通过组态设置在所述主虚拟控制器、所述备用虚拟控制器、所述输入/输出设备之间建立数据连接,以便在所述主虚拟控制器、所述备用虚拟控制器、所述输入/输出设备之间进行数据传输;

所述控制器虚拟化控制方法,还包括:对预设智能算法块进行了封装供脚本指令解析器调用;对自定义控制器指令行进行解析并进行直接机器执行或调用相应的所述预设智能算法块并执行;处理实时数据库单元与外部节点的数据通信,根据组态动态生成订阅、发布任务;对输入/输出设备及包含所述主虚拟控制器和所述备用虚拟控制器的虚拟控制器的内部数据进行内存存储和硬盘持久化;将组态中的计算任务按照优先级顺序和时间要求进行调度,并调用操作系统的调度器进行任务调度;对工程师站发送的组态文件进行解析,并形成虚拟控制器内部指令;通过GPS时间信号进行内部时钟的校准,并对虚拟控制器的状态进行监控和诊断,形成日志。

8. 一种计算机可读存储介质,其特征在于,用于存储计算机程序;其中,所述计算机程序被处理器执行时实现如权利要求7所述的虚拟化控制器通信方法的步骤。

一种控制器虚拟化控制系统、方法、存储介质

技术领域

[0001] 本发明涉及工业自动化领域,特别涉及一种控制器虚拟化控制系统、方法、存储介质。

背景技术

[0002] 目前,对于具有支持“虚拟化”部署、具有“可编程”能力及提供智能算法并行计算环境等特征且用于工业过程控制场景的虚拟控制器技术的技术中。大部分只是实现了单一的虚拟化,或者是对常规的实体控制器的软件化,并未对复杂的智能算法组态、高效运行及部署提出详细方法。现有技术中,提出了一种基于容器化技术的工业App资源开放共享方法,基于容器集群管理系统Kubernetes进行容器编排管理,基于应用容器引擎Docker进行工业app封装及镜像发布,部署工业app,提供工业app资源全生命周期的管理调用服务并对工业app状态进行监控。还提供了一种采用X86架构,具有高性能和高可靠性,能够满足工业控制系统对实时性和稳定性的要求。其特点是系统采用软件编程方式,通过软件实现PLC功能,避免了硬件开发成本高、维护困难等问题。支持多种编程语言,包括Ladder语言、SFC语言、ST语言等,方便用户进行编程。可以与多种外部设备进行通信,包括传感器、执行器、触摸屏等,具有良好的扩展性。支持远程监控和控制,方便用户进行远程操作和管理。但只实现了采用软件定义方法实现了传统PLC所具有的可编程控制功能,对当前复杂的智能控制算法的编码及并行运行没有很好的支持,且并未提及对虚拟化技术的应用。

[0003] 综上,如何实现通用计算机中的虚拟控制器的部署以及能够同时运行常规简单逻辑和智能算法块是本领域有待解决的技术问题。

发明内容

[0004] 有鉴于此,本发明的目的在于提供一种控制器虚拟化控制系统、方法、设备、存储介质,能够实现通用计算机中的虚拟控制器的部署以及能够同时运行常规简单逻辑和智能算法块。其具体方案如下:

[0005] 第一方面,本申请公开了一种控制器虚拟化控制系统,所述控制器虚拟化控制系统包括:包含布置于第一独立网段的主虚拟控制器、布置于第二独立网段的备用虚拟控制器的虚拟控制器、通过第一网口、第二网络分别连接第一独立网段和第二独立网段的输入/输出设备、位于通用PC机的服务编排与文件管理模块,其中,

[0006] 所述主虚拟控制器,用于调用所述通用服务器的资源与位于所述第一独立网段中的输入/输出设备进行点对点网络通信;

[0007] 所述备用虚拟控制器,用于调用所述通用服务器的资源与位于所述第二独立网段中的输入/输出设备进行点对点网络通信;

[0008] 通过交叉心跳线连接的所述主虚拟控制器和所述备用虚拟控制器,用于实现所述主虚拟控制器和所述备用虚拟控制器进行状态互检和信息同步;

[0009] 所述服务编排与文件管理模块,用于基于控制逻辑并通过组态设置在所述主虚拟

控制器、所述备用虚拟控制器、所述输入/输出设备之间建立数据连接,以便在所述主虚拟控制器、所述备用虚拟控制器、所述输入/输出设备之间进行数据传输。

[0010] 可选的,所述虚拟控制器,包括:

[0011] 控制器算法应用库,对预设智能算法块进行了封装供脚本指令解析器调用;

[0012] 脚本指令解析器,用于对自定义控制器指令行进行解析并进行直接机器执行或调用相应的所述预设智能算法块并执行;

[0013] 通信单元,用于处理所述实时数据单元与外部节点的数据通信,根据组态动态生成订阅、发布任务;

[0014] 实时数据库单元,用于对输入/输出设备及所述虚拟控制器的内部数据进行内存存储和硬盘持久化;

[0015] 任务调度单元,用于将组态中的计算任务按照优先级顺序和时间要求进行调度,并调用操作系统的调度器进行任务调度;

[0016] 组态文件解析子模块,用于对工程师站发送的组态文件进行解析,并形成虚拟控制器内部指令;

[0017] 校时单元及管理单元,用于通过GPS时间信号进行内部时钟的校准,并对虚拟控制器的状态进行监控和诊断,形成日志。

[0018] 可选的,所述虚拟控制器,包括:

[0019] 数据连接配置单元,用于查询所述虚拟控制器的所有数据资源并配置相应的数据连接,以便将所述虚拟控制器的数据缓存区的数据引用或发送至对应的输入/输出设备和控制器数据区。

[0020] 可选的,所述服务编排与文件管理模块,包括:

[0021] 服务编排子模块,用于可视化编辑以生成用于对所述控制器虚拟化控制系统软硬件的灵活配置的工程组态文件。

[0022] 可选的,所述服务编排子模块,包括:

[0023] 参数配置单元,用于对硬件配置、通信配置、设备连接和系统参数进行参数配置;

[0024] 算法配置单元,用于通过算法块表示和算法脚本形式表示来实现控制算法文件的表示;

[0025] 数据库文件配置单元,用于对实时数据库、历史数据库、配置数据进行库文件配置;

[0026] 图形界面配置单元,用于对操作界面、监视界面、报警界面以图形化界面的形式配置;

[0027] 信息记录单元,用于记录包含运行日志、时间记录、性能指标的数据信息。

[0028] 可选的,所述算法配置单元,包括:

[0029] 界面设计单元,用于基于图形界面的工具组件提供可视化编排和解析工具;

[0030] 语言设计单元,用于定义预设智能算法块的属性和方法;

[0031] 有向图设计单元,用于基于有向图的数据结构表示所述预设智能算法块之间依赖关系。

[0032] 可选的,所述控制器虚拟化控制系统,还包括:

[0033] 编排单元,用于通过用户端利用的输入设备将预设智能算法块拖拽至画布区,并

基于多线程进行分区表示,获取多个目标分区,然后通过连线连接各个目标分区并对所述目标分区中的预设智能算法块进行属性设置;

[0034] 文件解析单元,用于根据所述组态文件创建并行多线程,将输入数据流注入到预设智能算法块中;根据所述依赖关系,按照拓扑排序的顺序依次执行预设智能算法块,并将执行结果输出到相应的虚拟控制器内。

[0035] 第二方面,本申请公开了一种控制器虚拟化控制方法,包括:

[0036] 通过主虚拟控制器调用通用服务器的资源与位于第一独立网段中的输入/输出设备进行点对点网络通信;

[0037] 通过备用虚拟控制器调用通用服务器的资源与位于第二独立网段中的输入/输出设备进行点对点网络通信;

[0038] 通过交叉心跳线连接的主虚拟控制器和备用虚拟控制器实现主虚拟控制器和备用虚拟控制器进行状态互检和信息同步;

[0039] 基于控制逻辑并通过组态设置在所述主虚拟控制器、所述备用虚拟控制器、所述输入/输出设备之间建立数据连接,以便在所述主虚拟控制器、所述备用虚拟控制器、所述输入/输出设备之间进行数据传输。

[0040] 第三方面,本申请公开了一种计算机可读存储介质,用于存储计算机程序;其中,所述计算机程序被处理器执行时实现前述公开的虚拟化控制器通信方法的步骤。

[0041] 由此可见,本申请公开了一种控制器虚拟化控制系统,所述控制器虚拟化控制系统包括:包含布置于第一独立网段的主虚拟控制器、布置于第二独立网段的备用虚拟控制器的虚拟控制器、通过第一网口、第二网络分别连接第一独立网段和第二独立网段的输入/输出设备、位于通用PC机的服务编排与文件管理模块,其中,所述主虚拟控制器,用于调用所述通用服务器的资源与位于所述第一独立网段中的输入/输出设备进行点对点网络通信;所述备用虚拟控制器,用于调用所述通用服务器的资源与位于所述第二独立网段中的输入/输出设备进行点对点网络通信;通过交叉心跳线连接的所述主虚拟控制器和所述备用虚拟控制器,用于实现所述主虚拟控制器和所述备用虚拟控制器进行状态互检和信息同步;所述服务编排与文件管理模块,用于基于控制逻辑并通过组态设置在所述主虚拟控制器、所述备用虚拟控制器、所述输入/输出设备之间建立数据连接,以便在所述主虚拟控制器、所述备用虚拟控制器、所述输入/输出设备之间进行数据传输。可见,通过对控制器虚拟化控制系统进行系统设计,提供一种可运行于通用计算机资源的虚拟化控制器结构设计方法,它不仅具备实体控制器的所有功能,而且支持可编程且支持复杂智能算法并行计算能够运行在通用服务器,并且通过服务编排与文件管理模块能够对系统硬件进行配置及控制器的算法逻辑编排组态及文件管理,能够进一步实现通用计算机中的虚拟控制器的部署以及能够同时运行常规简单逻辑和智能算法块。

附图说明

[0042] 为了更清楚地说明本发明实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本发明的实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据提供的附图获得其他的附图。

- [0043] 图1为本申请公开的一种控制器虚拟化控制系统结构图；
- [0044] 图2为本申请公开的一种具体的可编程虚拟控制器系统架构图；
- [0045] 图3为本申请公开的一种具体的消息传递分布式通信设计示意图；
- [0046] 图4为本申请公开的一种具体的控制器虚拟化控制系统结构图；
- [0047] 图5为本申请公开的一种具体的虚拟控制器设计示意图；
- [0048] 图6为本申请公开的一种组态文件结构示意图；
- [0049] 图7为本申请公开的一种智能算法和逻辑算法混合组态示意图；
- [0050] 图8为本申请公开的一种控制器虚拟化控制方法流程图。

具体实施方式

[0051] 下面将结合本申请实施例中的附图,对本申请实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本发明一部分实施例,而不是全部的实施例。基于本发明中的实施例,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例,都属于本发明保护的范围。

[0052] 目前,对于具有支持“虚拟化”部署、具有“可编程”能力及提供智能算法并行计算环境等特征且用于工业过程控制场景的虚拟控制器技术的技术中。大部分只是实现了单一的虚拟化,或者是对常规的实体控制器的软件化,并未对复杂的智能算法组态、高效运行及部署提出详细方法。现有技术中,提出了一种基于容器化技术的工业App资源开放共享方法,基于容器集群管理系统Kubernetes进行容器编排管理,基于应用容器引擎Docker进行工业app封装及镜像发布,部署工业app,提供工业app资源全生命周期的管理调用服务并对工业app状态进行监控。还提供了一种采用X86架构,具有高性能和高可靠性,能够满足工业控制系统对实时性和稳定性的要求。其特点是系统采用软件编程方式,通过软件实现PLC功能,避免了硬件开发成本高、维护困难等问题。支持多种编程语言,包括Ladder语言、SFC语言、ST语言等,方便用户进行编程。可以与多种外部设备进行通信,包括传感器、执行器、触摸屏等,具有良好的扩展性。支持远程监控和控制,方便用户进行远程操作和管理。但只实现了采用软件定义方法实现了传统PLC所具有的可编程控制功能,对当前复杂的智能控制算法的编码及并行运行没有很好的支持,且并未提及对虚拟化技术的应用。

[0053] 为此,本申请公开了一种虚拟化控制器通信方案,能够实现通用计算机中的虚拟控制器的部署以及能够同时运行常规简单逻辑和智能算法块。

[0054] 参照图1所示,本申请实施例公开了一种控制器虚拟化控制系统,所述控制器虚拟化控制系统包括:包含布置于第一独立网段的主虚拟控制器111、布置于第二独立网段的备用虚拟控制器112的虚拟控制器11、通过第一网口、第二网络分别连接第一独立网段和第二独立网段的输入/输出设备12、位于通用PC机的服务编排与文件管理模块13,其中,

[0055] 所述主虚拟控制器111,用于调用所述通用服务器的资源与位于所述第一独立网段中的输入/输出设备13进行点对点网络通信;

[0056] 所述备用虚拟控制器112,用于调用所述通用服务器的资源与位于所述第二独立网段中的输入/输出设备12进行点对点网络通信;

[0057] 通过交叉心跳线连接的所述主虚拟控制器111和所述备用虚拟控制器112,用于实现所述主虚拟控制器111和所述备用虚拟控制器112进行状态互检和信息同步;

[0058] 所述服务编排与文件管理模块13,用于基于控制逻辑并通过组态设置在所述主虚拟控制器111、所述备用虚拟控制器112、所述输入/输出设备12之间建立数据连接,以便在所述主虚拟控制器111、所述备用虚拟控制器112、所述输入/输出设备12之间进行数据传输。

[0059] 可以理解的是,参照图2所示,构建了一个控制器虚拟化控制系统,其中,该虚拟化控制器系统包括系统软件与系统硬件,虚拟控制器为一套运行在虚拟机的软件,具体的,111表示该控制器虚拟化控制系统中的所有的主虚拟控制器,112表示该控制器虚拟化控制系统中所有的备用虚拟控制器,具体的,在该控制器虚拟化控制系统中包括多个主虚拟控制器和与各个主虚拟控制器对应的备用虚拟控制器,这样一来,该控制器虚拟化控制系统能够根据故障条件实现主、备虚拟控制器的自动切换。3表示通用服务器,作为运行虚拟控制器的硬件基础设施及计算资源虚拟化运行载体,4表示控制控制器虚拟化控制系统的工程师及操作员站,为一台通用PC机,5表示运行在工程师及操作站的系统组态软件,该系统组态软件中包含对系统硬件进行配置及虚拟控制器的服务编排与文件管理模块,具体用于对系统硬件进行参数配置以及对虚拟控制器的算法逻辑编排组态及文件管理。6表示控制器虚拟化控制系统的监控软件,可实现对控制器虚拟化控制系统中所有模块的状态监测及被控对象工艺的监控,7表示以太网网络相关的设备,为独立双网结构提高系统底层通信的可靠性,主要为交换机和光纤等构成,8为配套的IO设备,包括:IO设备的网关及连接仪表等监控对象的IO模块。通过设计一个高效、可靠、实时、灵活、可扩展和安全的数据通信系统、一个支持可编程并且支持复杂智能算法并行计算且能运行在通用计算硬件之上的虚拟化控制器载体来实现虚拟化控制器进行数据传输。

[0060] 参照图3所示,提供了一种面向数据分发的消息传递机制实现虚拟化通信系统中各控制单元及输入/输出设备8的数据交互。具体的,设置第一独立网段9和第二独立网段10作为双网冗余保障可靠性,其中,所述第一独立网段9可以为192网段,第二独立网段10可以为172,然后将主虚拟控制器和对应的备用虚拟控制器作为控制器节点和备用控制器节点,并在主控制器节点和备用控制器节点之间构建主从机制,然后分别将主控制器节点和备用控制器节点设置于第一独立网段9和第二独立网段10中,同时将输入/输出设备作为网关节点,设置于第一独立网段9和第二独立网段10中,然后,针对同一网段的输入/输出网关节点、控制器节点进行点对点网络通信,网络层采用IP协议,传输层采用UDP协议,上层采用RTPS协议进行分布式通信。其中,虚拟控制器成对布置实现互为备用,两个虚拟控制器通过一路交叉心跳连线,实现状态相互检测及同步。系统IO设备设置有连入系统网络的网关节点,设置有两个独立网口分别连入第一独立网段9和第二独立网段10。系统设置有一个全局管理节点,分别连入两个网段,并对每个节点进行心跳连接,定时获取各节点数据。网络上所有节点采用RTPS协议,各点之间可以根据控制逻辑需要通过组态设置后建立数据连接,形成发布者和订阅者,发布者可以将数据发送给订阅者。为了保证连接的可靠性,节点间通信具有心跳检测,来监测连接的状态,当一个发布者或订阅者长时间未发送数据时,对端会发送心跳包来检测连接是否还存活。如果对端没有回应,则认为连接已经断开,并且具有数据重传的机制,具体可根据项目需要开启或关闭:在数据传输过程中,如果数据包丢失或损坏,传输会自动触发数据重传机制,发布者会将丢失的数据包重新发送给订阅者,以确保数据的可靠传输。实现了在控制器节点和IO节点的解耦,提高了系统配置的灵活性,可以节省

工程实施时间,满足不同工业控制场景的需求。

[0061] 由此可见,本申请公开了一种控制器虚拟化控制系统,所述控制器虚拟化控制系统包括:包含布置于第一独立网段的主虚拟控制器、布置于第二独立网段的备用虚拟控制器的虚拟控制器、通过第一网口、第二网络分别连接第一独立网段和第二独立网段的输入/输出设备、位于通用PC机的服务编排与文件管理模块,其中,所述主虚拟控制器,用于调用所述通用服务器的资源与位于所述第一独立网段中的输入/输出设备进行点对点网络通信;所述备用虚拟控制器,用于调用所述通用服务器的资源与位于所述第二独立网段中的输入/输出设备进行点对点网络通信;通过交叉心跳线连接的所述主虚拟控制器和所述备用虚拟控制器,用于实现所述主虚拟控制器和所述备用虚拟控制器进行状态互检和信息同步;所述服务编排与文件管理模块,用于基于控制逻辑并通过组态设置在所述主虚拟控制器、所述备用虚拟控制器、所述输入/输出设备之间建立数据连接,以便在所述主虚拟控制器、所述备用虚拟控制器、所述输入/输出设备之间进行数据传输。可见,通过对控制器虚拟化控制系统进行系统设计,提供一种可运行于通用计算机资源的虚拟化控制器结构设计方法,它不仅具备实体控制器的所有功能,而且支持可编程且支持复杂智能算法并行计算能够运行在通用服务器,并且通过服务编排与文件管理模块能够对系统硬件进行配置及控制器的算法逻辑编排组态及文件管理,能够进一步实现通用计算机中的虚拟控制器的部署以及能够同时运行常规简单逻辑和智能算法块。

[0062] 参照图4所示,本发明实施例公开了一种具体的控制器虚拟化控制系统,相对于上一实施例,本实施例对技术方案作了进一步的说明和优化。具体的:

[0063] 所述虚拟控制器11,包括:

[0064] 控制器算法应用库113,对预设智能算法块进行了封装供脚本指令解析器调用;

[0065] 脚本指令解析器114,用于对自定义控制器指令行进行解析并进行直接机器执行或调用相应的所述预设智能算法块并执行;

[0066] 通信单元115,用于处理所述实时数据单元与外部节点的数据通信,根据组态动态生成订阅、发布任务;

[0067] 实时数据库单元116,用于对输入/输出设备及所述虚拟控制器的内部数据进行内存存储和硬盘持久化;

[0068] 任务调度单元117,用于将组态中的计算任务按照优先级顺序和时间要求进行调度,并调用操作系统的调度器进行任务调度;

[0069] 组态文件解析子模块118,用于对工程师站发送的组态文件进行解析,并形成虚拟控制器内部指令;

[0070] 校时单元及管理单元119,用于通过GPS时间信号进行内部时钟的校准,并对虚拟控制器的状态进行监控和诊断,形成日志。

[0071] 所述虚拟控制器11,包括:

[0072] 数据连接配置单元110,用于查询所述虚拟控制器的所有数据资源并配置相应的数据连接,以便将所述虚拟控制器的数据缓存区的数据引用或发送至对应的输入/输出设备和控制器数据区。

[0073] 可以理解的是,在虚拟控制器11中,包含对接收的或自定义的控制器指令进行解析,并在指令解析后直接机器执行或者在指令解析后调用相应的资源执行的脚本指令解析

器113,当通过脚本指令解析器113进行指令解析后,

[0074] 可以理解的是,控制器算法应用库113主要是对常用的智能算法块进行了封装供脚本指令解析器114调用,脚本指令解析器114采用类shell的解析器设计,用于对自定义的控制器指令行进行解析并进行直接机器执行或者调用相应的资源并执行。通信单元115主要是处理实时数据库单元116与外部节点的数据通信,根据组态动态生成订阅、发布任务。任务调度单元117主要负责将组态中各个计算任务按照优先级和时间要求进行调度,并调用操作系统的调度器进行任务调度。组态文件解析子模块118用于对工程师站发送的组态文件进行解析,并形成控制器内部指令。实时数据库单元116用于对I/O及虚拟控制器内部数据进行内存存储和硬盘持久化。校时单元用于通过GPS时间信号进行内部时钟的校准。管理单元是对控制器各模块的状态进行监控和诊断,并形成日志。虚拟控制器采用数据与计算相分离设计,数据连接配置单元110用于在虚拟控制器数据区可以通过查询全局节点中的数据资源进行可数据连接的配置,每个虚拟控制器中的数据缓存区的变量可以引用或发送至相应的I/O模块和控制器数据区,实现数据的分布式互联,该数据传输关系由设计的通信单元115进行解析。参照图5所示,提供了一种虚拟控制器架构模块化设计,具体的,虚拟控制器底层为虚拟机,采用Xen、KVM、VMware等虚拟化技术实现对通用it服务器资源的虚拟化。其中,所述通用it服务器资源可以包括但不限于:CPU、内存、硬盘、网络等。在虚拟机上运行RTOS即实时操作系统软件并安装上层软件所需的底层驱动及API环境,在此基础上脚本指令解析器、实时数据模块、通信单元、任务调度单元、组态文件解析、控制器算法应用库、校时单元及管理单元等组件。

[0075] 所述服务编排与文件管理模块13,包括:

[0076] 服务编排子模块131,用于可视化编辑以生成用于对所述控制器虚拟化控制系统软硬件的灵活配置的工程组态文件。

[0077] 所述服务编排子模块131,包括:

[0078] 参数配置单元1311,用于对硬件配置、通信配置、设备连接和系统参数进行参数配置;

[0079] 算法配置单元1312,用于通过算法块表示和算法脚本形式表示来实现控制算法文件的表示;

[0080] 数据库文件配置单元1313,用于对实时数据库、历史数据库、配置数据进行库文件配置;

[0081] 图形界面配置单元1314,用于对操作界面、监视界面、报警界面以图形化界面的形式配置;

[0082] 信息记录单元1315,用于记录包含运行日志、时间记录、性能指标的数据信息。

[0083] 所述算法配置单元1312,包括:

[0084] 界面设计单元13121,用于基于图形界面的工具组件提供可视化编排和解析工具;

[0085] 语言设计单元13122,用于定义预设智能算法块的属性和方法;

[0086] 有向图设计单元13123,用于基于有向图的数据结构表示所述预设智能算法块之间依赖关系。

[0087] 所述控制器虚拟化控制系统,还包括:

[0088] 编排单元13124,用于通过用户端利用的输入设备将预设智能算法块拖拽至画布

区,并基于多线程进行分区表示,获取多个目标分区,然后通过连线连接各个目标分区并对所述目标分区中的预设智能算法块进行属性设置;

[0089] 文件解析单元13125,用于根据所述组态文件创建并行多线程,将输入数据流注入到预设智能算法块中;根据所述依赖关系,按照拓扑排序的顺序依次执行预设智能算法块,并将执行结果输出到相应的虚拟控制器内。

[0090] 可以理解的是,在服务编排与文件管理模块13中,主要对虚拟控制器的算法逻辑编排组态以及文件管理,其中,对于算法逻辑编排组态主要是通过服务编排子模块131可视化系统软硬件的灵活配置的工程组态文件,具体的,在服务编排子模块131中,通过参数配置单元1311对系统的软硬件进行相应的参数配置,而算法配置单元1312,通过算法块表示和算法脚本形式表示来实现控制算法文件的表示;数据库文件配置单元1313对实时数据库、历史数据库、配置数据进行库文件配置;图形界面配置单元1314,对操作界面、监视界面、报警界面以图形化界面的形式配置;信息记录单元1315,记录包含运行日志、时间记录、性能指标的数据信息。具体的,参照图6所示,控制器虚拟化控制系统可进行文件编辑形成工程组态文件,实现对系统软硬件的灵活可配置,组态文件可以通过工程师站组态软件模块打开并进行可视化编辑。组态文件系统的设计主要涵盖:系统配置文件,具体包括硬件配置、通信配置、设备连接、系统参数;控制算法文件,包括功能块表示和脚本形式表示;数据库文件,包括实时数据库、历史数据库、配置数据;图形界面文件,包含操作界面、监视界面、报警界面等,以图形化界面的形式展现,各种报表、日志等信息,如运行日志、事件记录、性能指标等。

[0091] 可以理解的是,在算法配置单元1311中,控制算法文件除了实现对常规函数块图支持外,还提供了一个基于有向图的智能算法功能块的编排和解析工具,其设计方法如下:

(1) 界面设计:采用图形界面,为用户提供可视化的编排和解析工具。包括工具栏、画布区、属性面板、控制台等组件。(2) 语言设计:采用类似于面向对象编程的方式,将算法功能块抽象为类或对象,并定义其属性和方法。例如,可以定义一个名为“算法块”的类,其中包含了算法块的输入输出端口、属性、运行方法等。(3) 有向图设计:采用有向图的数据结构来表示算法功能块之间的依赖关系。(4) 编排功能实现:参照图7所示,用户可以通过鼠标拖拽的方式将算法功能块拖拽到画布区,画布区根据多线程实现分区表示,并通过连线将它们连接起来。连线可以表示数据流或控制流之间的依赖关系。用户还可以通过属性面板来设置算法功能块的属性,如名称、输入输出端口、参数等,具体的,在线程1-1中,将输入变量1-11与输入变量1-12进行相加,然后将输入变量的相加结果输入至智能算法块1-14,得到第一过程结果;将输入变量1-13直接输入至智能算法块1-15,得到第二过程结果,将第一过程结果和第二过程结果进行大小判断,以得到最终的判断结果作为输出变量1-16进行输出。通过线程2-1对输入变量2-11、输入变量2-12、输入变量2-13进行相应的逻辑和智能算法块处理,获取输出变量2-15。(5) 解析功能实现:系统提供组态文件解析模块对算法组态文件进行解析,根据组态文件内容创建并行多线程,将输入数据流注入到算法功能块中。解析功能将根据有向图中的依赖关系,按照拓扑排序的顺序依次执行算法功能块,并将结果输出到相应的控制器内映射中。(6) 错误处理:编排和解析过程中会涉及到许多错误处理情况,如节点不存在、输入输出端口不匹配、循环依赖等。因此,在设计中需要考虑到这些异常情况,并提供相应的错误提示和处理机制。(7) 文件格式:为了方便保存和加载算法功能块的编

排,可以定义一种文件格式来保存有向图的数据结构和算法功能块的属性,本系统采用了XML格式来保存数据。

[0092] 现有虚拟控制器的组态功能主要遵循的是IEC61131-3的国际标准编程语言,该种语言结构相对简单,只能实现一些简单的算法逻辑,对于复杂的算法实现,如深度学习、强化学习等,该语言难以进行表达,因此控制器主要支持常规的简单逻辑,很难实现智能算法的开发。虚拟控制器对于有强大算力需求的智能算法运行,没有设计相应的并行处理和加速机制,导致其仍然适合运行计算复杂度较低的逻辑运算。出现此问题的主要原因是,现有虚拟控制器技术主要是对实体控制器如PLC作了软件化,没有改变原有的单线程、顺序串行计算的框架。而本申请通过采用了虚拟化技术,能够将多个控制器虚拟化到一个物理设备上,实现了对不同控制器的统一管理和控制,提高了系统的可靠性、稳定性、安全性和管理效率。通过虚拟控制器系统化设计,不仅具备实体控制器的所有功能,更重要的是具有可为基于神经网络、大数据挖掘和先进控制等技术的智能应用算法块提供运行环境的能力,解决了当前常规控制器只能运行简单逻辑算法的问题。

[0093] 由此可见,通过设计了包含常规简单逻辑和智能算法组态的低代码系统开发方案,解决当前智能算法与常规简单逻辑不能在同一个控制器内共存现状,增加虚拟控制器的工程应用范围,为工程开发人员提供高效、便捷的工具。

[0094] 参照图8所示,本发明实施例还相应公开了一种控制器虚拟化控制方法,包括:

[0095] 步骤S11:通过主虚拟控制器调用通用服务器的资源与位于第一独立网段中的输入/输出设备进行点对点网络通信。

[0096] 本实施例中,设置第一独立网段和第二独立网段作为双网冗余保障可靠性,其中,所述第一独立网段可以为192网段,第二独立网段可以为172,然后将主虚拟控制器作为控制器节点,然后将主控制器节点设置于第一独立网段,同时将输入/输出设备作为网关节点,设置于第一独立网段,这样一来,将同处于第一独立网段192中的主虚拟控制器通过调用通用服务器资源与输入/输出设备进行点对点的网络通信。

[0097] 步骤S12:通过备用虚拟控制器调用通用服务器的资源与位于第二独立网段中的输入/输出设备进行点对点网络通信。

[0098] 本实施例中,将备用虚拟控制器作为备用控制器节点,然后将备用控制器节点设置于第二独立网段172中,然后将同处于第二独立网段192中的备用虚拟控制器通过调用通用服务器资源与输入/输出设备进行点对点的网络通信。

[0099] 步骤S13:通过交叉心跳线连接的主虚拟控制器和备用虚拟控制器实现主虚拟控制器和备用虚拟控制器进行状态互检和信息同步。

[0100] 本实施例中,在主控制器节点和备用控制器节点之间构建主从机制,通过交叉心跳线连接主控制器节点和备用控制器节点,实现状态相互检测及同步。同时,还设置全局管理节点,分别连入第一独立网段和第二独立网段,并对每个节点进行心跳连接,用于定时获取各个节点产生的节点数据,实现对全局的监控。

[0101] 步骤S14:基于控制逻辑并通过组态设置在所述主虚拟控制器、所述备用虚拟控制器、所述输入/输出设备之间建立数据连接,以便在所述主虚拟控制器、所述备用虚拟控制器、所述输入/输出设备之间进行数据传输。

[0102] 本实施例中,网络上所有节点采用RTPS协议,各点之间可以根据控制逻辑需要通

过组态设置后建立数据连接,形成发布者和订阅者,发布者可以将数据发送给订阅者。

[0103] 由此可见,本申请公开了调用所述通用服务器的资源与位于所述第一独立网段中的输入/输出设备进行点对点网络通信;调用所述通用服务器的资源与位于所述第二独立网段中的输入/输出设备进行点对点网络通信;通过交叉心跳线连接的所述主虚拟控制器和所述备用虚拟控制器,实现所述主虚拟控制器和所述备用虚拟控制器进行状态互检和信息同步;基于控制逻辑并通过组态设置在所述主虚拟控制器、所述备用虚拟控制器、所述输入/输出设备之间建立数据连接,以便在所述主虚拟控制器、所述备用虚拟控制器、所述输入/输出设备之间进行数据传输。可见,通过对控制器虚拟化控制系统进行系统设计,提供一种可运行于通用计算机资源的虚拟化控制器结构设计方法,它不仅具备实体控制器的所有功能,而且支持可编程且支持复杂智能算法并行计算能够运行在通用服务器,并且通过服务编排与文件管理模块能够对系统硬件进行配置及控制器的算法逻辑编排组态及文件管理,能够进一步实现通用计算机中的虚拟控制器的部署以及能够同时运行常规简单逻辑和智能算法块。

[0104] 进一步的,本申请还公开了一种计算机可读存储介质,用于存储计算机程序;其中,所述计算机程序被处理器执行时实现前述公开的虚拟化控制器通信方法。关于该方法的具体步骤可以参考前述实施例中公开的相应内容,在此不再进行赘述。

[0105] 本说明书中各个实施例采用递进的方式描述,每个实施例重点说明的都是与其它实施例的不同之处,各个实施例之间相同或相似部分互相参见即可。对于实施例公开的装置而言,由于其与实施例公开的方法相对应,所以描述的比较简单,相关之处参见方法部分说明即可。

[0106] 专业人员还可以进一步意识到,结合本文中所公开的实施例描述的各示例的单元及算法步骤,能够以电子硬件、计算机软件或者二者的结合来实现,为了清楚地说明硬件和软件的可互换性,在上述说明中已经按照功能一般性地描述了各示例的组成及步骤。这些功能究竟以硬件还是软件方式来执行,取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能,但是这种实现不应认为超出本申请的范围。结合本文中所公开的实施例描述的方法或算法的步骤可以直接用硬件、处理器执行的软件模块,或者二者的结合来实施。软件模块可以置于随机存储器(RAM)、内存、只读存储器(ROM)、电可编程ROM、电可擦除可编程ROM、寄存器、硬盘、可移动磁盘、CD-ROM、或技术领域内所公知的任意其它形式的存储介质中。

[0107] 最后,还需要说明的是,在本文中,诸如第一和第二等之类的关系术语仅仅用来将一个实体或者操作与另一个实体或操作区分开来,而不一定要求或者暗示这些实体或操作之间存在任何这种实际的关系或者顺序。而且,术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含,从而使得包括一系列要素的过程、方法、物品或者设备不仅包括那些要素,而且还包括没有明确列出的其他要素,或者是还包括为这种过程、方法、物品或者设备所固有的要素。在没有更多限制的情况下,由语句“包括一个……”限定的要素,并不排除在包括所述要素的过程、方法、物品或者设备中还存在另外的相同要素。

[0108] 以上对本发明所提供的一种控制器虚拟化控制系统、方法、存储介质进行了详细介绍,本文中应用了具体个例对本发明的原理及实施方式进行了阐述,以上实施例的说明只是用于帮助理解本发明的方法及其核心思想;同时,对于本领域的一般技术人员,依据本

发明的思想,在具体实施方式及应用范围上均会有改变之处,综上所述,本说明书内容不应理解为对本发明的限制。

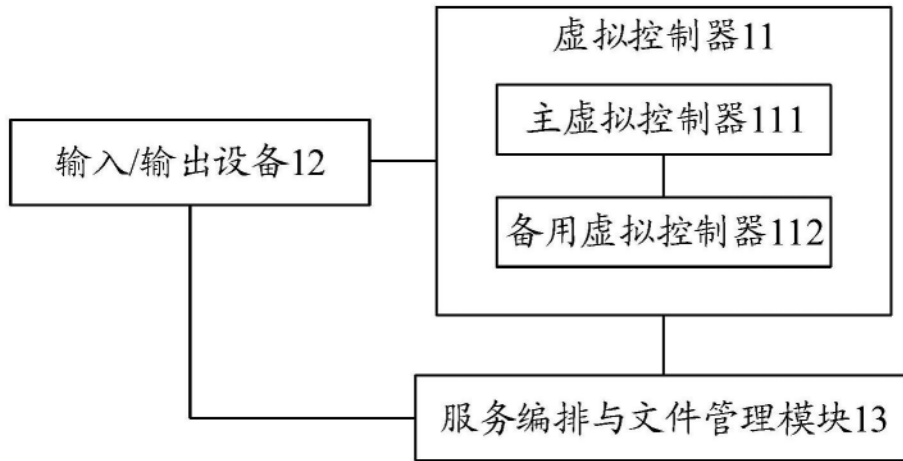


图1

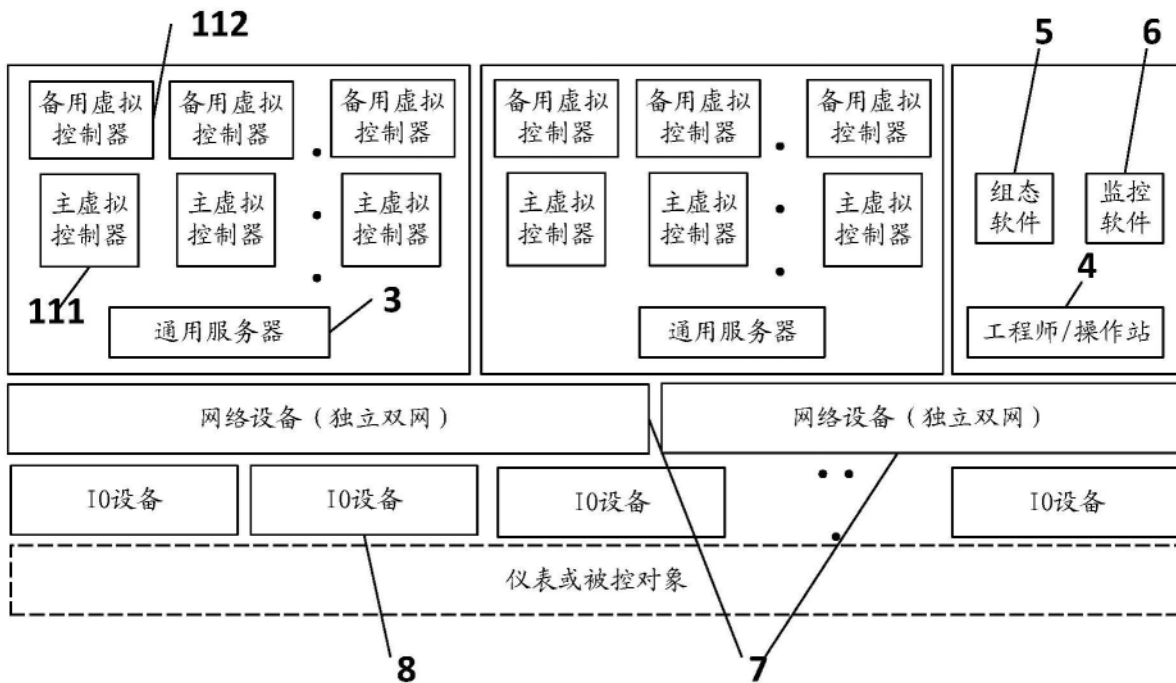


图2

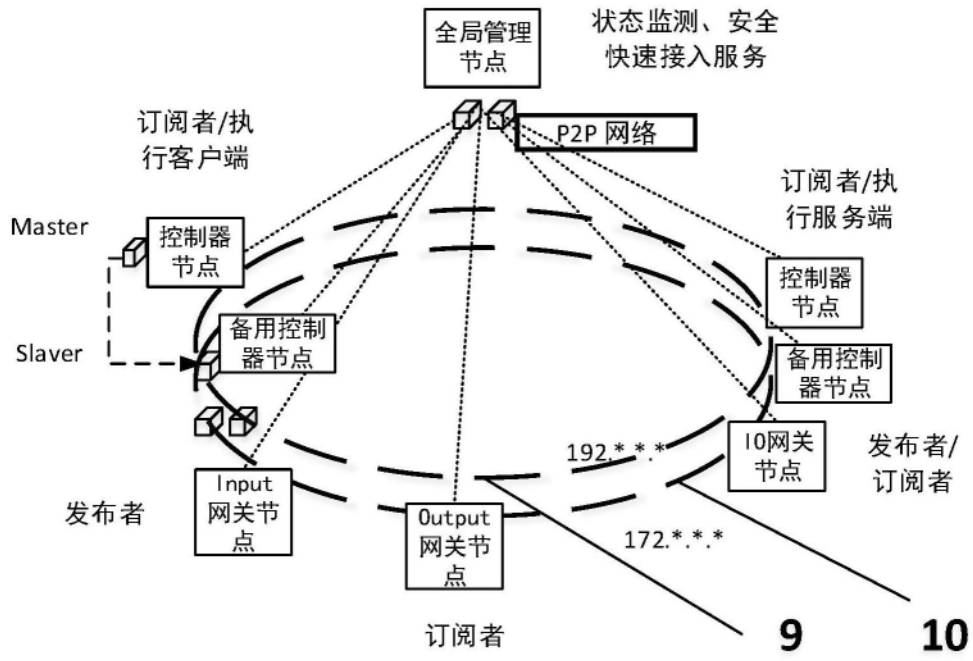


图3



图4

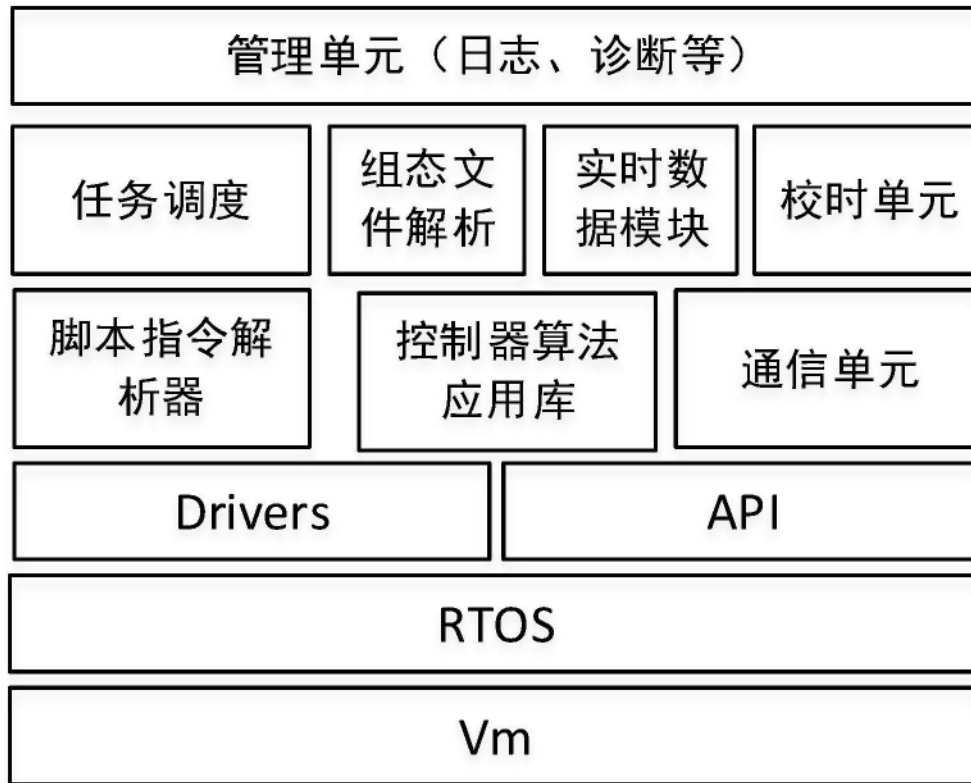


图5



图6

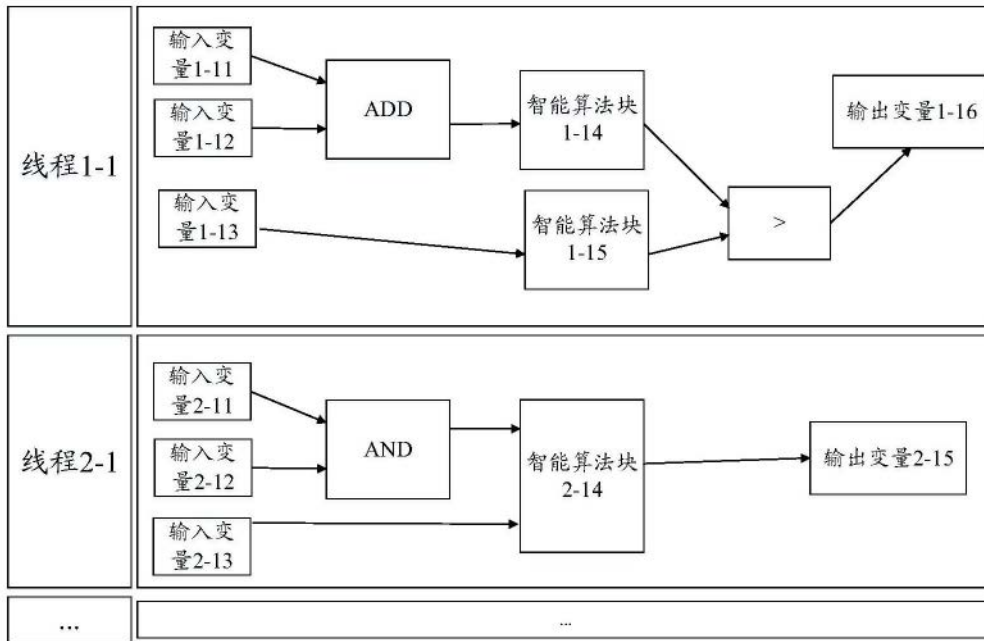


图7

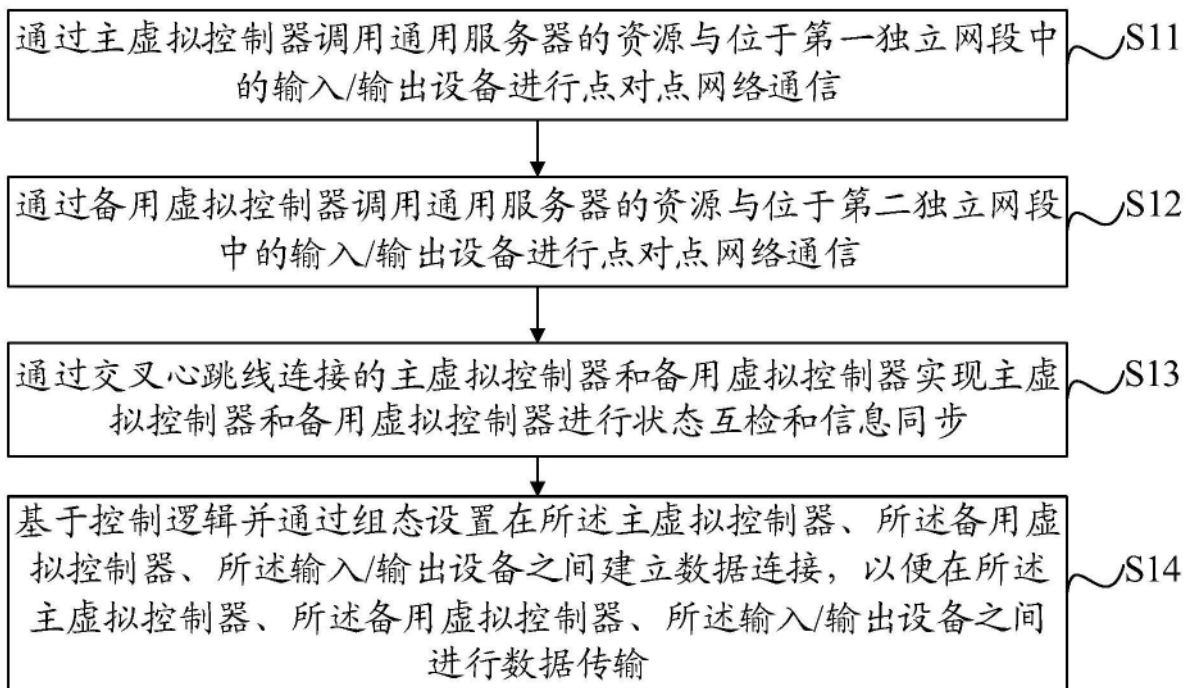


图8