



(12) 发明专利申请

(10) 申请公布号 CN 114492052 A

(43) 申请公布日 2022. 05. 13

(21) 申请号 202210110770.6

(22) 申请日 2022.01.29

(71) 申请人 上海交通大学

地址 200240 上海市闵行区东川路800号

(72) 发明人 曹培睿 赵世振 林圣凯 韩昕驰

刘运卓

(74) 专利代理机构 上海汉声知识产权代理有限公司

公司 31236

专利代理师 胡晶

(51) Int. Cl.

G06F 30/20 (2020.01)

G06F 9/455 (2006.01)

G06F 111/02 (2020.01)

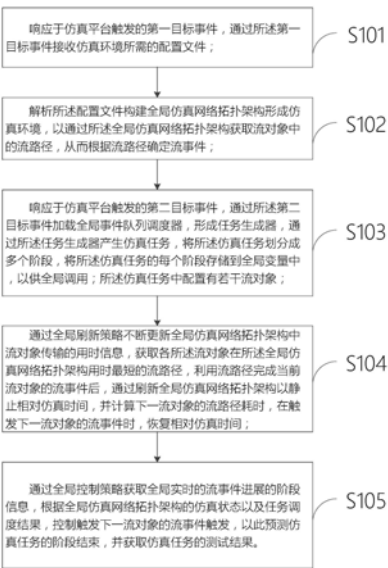
权利要求书2页 说明书11页 附图2页

(54) 发明名称

一种全局流级别网络仿真方法、系统及装置

(57) 摘要

本发明公开了一种全局流级别网络仿真方法、系统及装置,包括:构建全局仿真网络拓扑架构形成仿真环境后,通过任务生成器产生仿真任务,将仿真任务划分成多个阶段,将仿真任务的每个阶段存储到全局变量中,以供全局调用;通过全局刷新策略不断更新全局仿真网络拓扑架构中流对象传输的用时信息,确定全局各流对象用时最短的流事件,在当前流对象完成流事件后,通过刷新计算其他流对象的流事件完成时间预期值;通过全局控制策略获取全局阶段信息,检测某一阶段所有流对象是否全部完成流事件,以此预测仿真任务的阶段结束,并获取仿真任务的测试结果。本发明提供的网络仿真方案,在缩短资源消耗和计算时间的同时,确保了仿真的正确性。



1. 一种全局流级别网络仿真方法,其特征在于,包括:

响应于仿真平台触发的第一目标事件,通过所述第一目标事件接收仿真环境所需的配置文件;

解析所述配置文件构建全局仿真网络拓扑架构形成仿真环境,以通过所述全局仿真网络拓扑架构获取流对象中的流路径,从而根据流路径确定流事件;

响应于仿真平台触发的第二目标事件,通过所述第二目标事件加载全局事件队列调度器,形成任务生成器,通过所述任务生成器产生仿真任务,将所述仿真任务划分成多个阶段,将所述仿真任务的每个阶段存储到全局变量中,以供全局调用;所述仿真任务中配置有若干流对象;

通过全局刷新策略不断更新全局仿真网络拓扑架构中流对象传输的用时信息,获取各所述流对象在所述全局仿真网络拓扑架构用时最短的流路径,利用流路径完成当前流对象的流事件后,通过刷新全局仿真网络拓扑架构以静止相对仿真时间,并计算下一流对象的流路径耗时,在触发下一流对象的流事件时,恢复相对仿真时间;

通过全局控制策略获取全局实时的流事件进展的阶段信息,根据全局仿真网络拓扑架构的仿真状态以及任务调度结果,控制触发下一流对象的流事件触发,以此预测仿真任务的阶段结束,并获取仿真任务的测试结果。

2. 如权利要求1所述的全局流级别网络仿真方法,其特征在于,所述配置文件中设置有网络拓扑信息和图信息,通过所述网络拓扑信息和所述图信息构建仿真环境所需的全局仿真网络拓扑架构,并存储后供全局调用;

其中,所述图信息中配置有仿真节点、节点间的链接关系、节点权重,每个所述仿真节点配置有唯一的ID号,且每个所述仿真节点关联有与其连接的其他仿真节点的ID号,并反映在所述图信息中。

3. 如权利要求1所述的全局流级别网络仿真方法,其特征在于,所述全局刷新策略中,分析每个所述流对象关联的流信息中根据流路径可能触发的流事件,根据各所述流路径的完成时间预期值,确定待激活的流事件后,并标记为活跃状态,在执行全局刷新时,将已完成的流事件的标记改为不活跃后,同时遍历标记为活跃状态的所有流事件。

4. 如权利要求1所述的全局流级别网络仿真方法,其特征在于,所述全局控制策略中,分布式控制所述仿真任务的多个阶段,并通过不同的通信策略、通信对象以及流任务的数量,全局记录实时流进展的阶段信息。

5. 如权利要求1所述的全局流级别网络仿真方法,其特征在于,所述任务生成器采用事件基类的初始化函数控制仿真进程的相对时间和绝对时间,并设定激活状态变量进行加速时调用。

6. 如权利要求1所述的全局流级别网络仿真方法,其特征在于,所述任务生成器产生仿真任务后,对仿真任务的进程运行采用堆排序,并根据事件基类中重载运算逻辑来重新排序。

7. 如权利要求6所述的全局流级别网络仿真方法,其特征在于,各所述仿真任务通过所述事件基类衍生有三个作用于流对象的事件子类:分别为任务开始事件、传输事件、完成事件;

通过所述任务开始事件预设仿真任务参数,产生多条流对象,在检测所述仿真平台是

否触发第二目标事件的同时,接收当前所述仿真任务的修改配置;

通过所述传输事件控制仿真进程推进到某一时刻的瞬间,产生真正作用于网络的流对象;

通过所述完成事件在执行完所述流对象的流事件后,删除全局仿真网络拓扑架构中的这条流对象。

8.一种全局流级别网络仿真系统,采用权利要求1-7任意一项所述的方法,其特征在于,包括:

文件接收模块,配置为响应于仿真平台触发的第一目标事件,通过所述第一目标事件接收仿真环境所需的配置文件;

拓扑构建模块,配置为解析所述配置文件构建全局仿真网络拓扑架构形成仿真环境,以通过所述全局仿真网络拓扑架构获取流对象中的流路径,从而根据流路径确定流事件;

任务配置模块,配置为响应于仿真平台触发的第二目标事件,通过所述第二目标事件加载全局事件队列调度器,形成任务生成器,通过所述任务生成器产生仿真任务,将所述仿真任务划分成多个阶段,将所述仿真任务的每个阶段存储到全局变量中,以供全局调用;所述仿真任务中配置有若干流对象;

全局刷新模块,配置为通过全局刷新策略不断更新全局仿真网络拓扑架构中流对象传输的用时信息,获取各所述流对象在所述全局仿真网络拓扑架构用时最短的流路径,利用流路径完成当前流对象的流事件后,通过刷新全局仿真网络拓扑架构以静止相对仿真时间,并计算下一流对象的流路径耗时,在触发下一流对象的流事件时,恢复相对仿真时间;

全局控制模块,配置为通过全局控制策略获取全局实时的流事件进展的阶段信息,根据全局仿真网络拓扑架构的仿真状态以及任务调度结果,控制触发下一流对象的流事件触发,以此预测仿真任务的阶段结束,并获取仿真任务的测试结果。

9.一种全局流级别网络仿真装置,其特征在于,配置有权利要求8中所述的全局流级别网络仿真系统。

10.一种电子设备,其特征在于,包括:

一个或多个处理器;

存储器,用于存储一个或多个程序;

当所述一个或多个程序被所述一个或多个处理器执行时,使得所述一个或多个处理器实现如权利要求1-7中任一所述的方法。

11.一种计算机可读介质,其上存储有计算机程序,其特征在于,所述计算机程序被处理器执行时实现如权利要求1-7中任一所述的方法。

一种全局流级别网络仿真方法、系统及装置

技术领域

[0001] 本发明涉及仿真技术领域,尤其涉及一种全局流级别网络仿真方法、系统及装置。

背景技术

[0002] 数据中心网络、万维网通信、无线通信技术的研究与发展在很大程度上依赖于仿真技术,包括链路仿真、系统仿真以及网络仿真,其中网络仿真的仿真规模较大,通常有千万节点级别的仿真。现有的网络架构主要为服务器、交换器以及链路,为应对未来越来越多的流量以及越来越复杂的应用需求,在部署或升级真实的网络架构之前,往往需要先进行仿真验证分析。

[0003] 本申请发明人研究发现,现有网络仿真技术存在如下缺陷:

[0004] (1) 采用包级别的离散事件网络仿真,以NS3、OMNeT++为代表,适合进行层级协议的精细粒度仿真,但由于依赖于包级别以及事件驱动框架,进行大规模多节点多流仿真的时候,消耗过多的资源以及时间,即使开启分布式仿真功能,也需要消耗数天甚至数月的时间。基于此,现有衍生出机器学习来替代一部分相似网络结构的仿真,比如,MimicNet (Sigcomm 2021),只让OMNeT++仿真一个指定单位的网络结构,将剩余单位的网络结构放到Mimic近似模块中去计算,从而实现一定程度的速度提升,但是该方法对网络结构的奇异性和流量的突发性没法精确近似。现有,一些方法简化了一部分协议栈,比如,Netbench,但依然是受限于包级别和事件驱动的框架。

[0005] (2) 采用内核隔离虚拟化网络仿真器,以Mininet、IMUNES、CORE为例,仅适合做可实时控制每个节点的小规模仿真,但是单台机器资源有限,无法仿真大规模网络。

[0006] (3) 采用结合真实交换机的模拟器,比如可以实现让1台交换机虚拟仿真成8节点胖树的结构 TurboNet,但是它依旧没法做真正意义上大规模仿真。

发明内容

[0007] 为解决上述问题,本申请实施例通过提供一种全局流级别网络仿真方法、系统及装置,实现了在缩减资源消耗和计算时间的同时,保证了仿真方案测试的正确性。

[0008] 第一方面,本申请实施例提供了一种全局流级别网络仿真方法,包括:

[0009] 响应于仿真平台触发的第一目标事件,通过所述第一目标事件接收仿真环境所需的配置文件;

[0010] 解析所述配置文件构建全局仿真网络拓扑架构形成仿真环境,以通过所述全局仿真网络拓扑架构获取流对象中的流路径,从而根据流路径确定流事件;

[0011] 响应于仿真平台触发的第二目标事件,通过所述第二目标事件加载全局事件队列调度器,形成任务生成器,通过所述任务生成器产生仿真任务,将所述仿真任务划分成多个阶段,将所述仿真任务的每个阶段存储到全局变量中,以供全局调用;所述仿真任务中配置有若干流对象;

[0012] 通过全局刷新策略不断更新全局仿真网络拓扑架构中流对象传输的用时信息,获

取各所述流对象在所述全局仿真网络拓扑架构用时最短的流路径,利用流路径完成当前流对象的流事件后,通过刷新全局仿真网络拓扑架构以静止相对仿真时间,并计算下一流对象的流路径耗时,在触发下一流对象的流事件时,恢复相对仿真时间;

[0013] 通过全局控制策略获取全局实时的流事件进展的阶段信息,根据全局仿真网络拓扑架构的仿真状态以及任务调度结果,控制触发下一流对象的流事件触发,以此预测仿真任务的阶段结束,并获取仿真任务的测试结果。

[0014] 进一步地,所述配置文件中设置有网络拓扑信息和图信息,通过所述网络拓扑信息和所述图信息构建仿真环境所需的全局仿真网络拓扑架构,并存储后供全局调用;

[0015] 其中,所述图信息中配置有仿真节点、节点间的链接关系、节点权重,每个所述仿真节点配置有唯一的ID号,且每个所述仿真节点关联有与其连接的其他仿真节点的ID号,并反映在所述图信息中。

[0016] 进一步地,所述全局刷新策略中,分析每个所述流对象关联的流信息中根据流路径可能触发的流事件,根据各所述流路径的完成时间预期值,确定待激活的流事件后,并标记为活跃状态,在执行全局刷新时,将已完成的流事件的标记改为不活跃后,同时遍历标记为活跃状态的所有流事件。

[0017] 进一步地,所述全局控制策略中,分布式控制所述仿真任务的多个阶段,并通过不同的通信策略、通信对象以及流任务的数量,全局记录实时流进展的阶段信息。

[0018] 进一步地,所述任务生成器采用事件基类的初始化函数控制仿真进程的相对时间和绝对时间,并设定激活状态变量进行加速时调用。

[0019] 进一步地,所述任务生成器产生仿真任务后,对仿真任务的进程运行采用堆排序,并根据事件基类中重载运算逻辑来重新排序。

[0020] 进一步地,各所述仿真任务通过所述事件基类衍生有三个作用于流对象的事件子类:分别为任务开始事件、传输事件、完成事件;

[0021] 通过所述任务开始事件预设仿真任务参数,产生多条流对象,在检测所述仿真平台是否触发第二目标事件的同时,接收当前所述仿真任务的修改配置;

[0022] 通过所述传输事件控制仿真进程推进到某一时刻的瞬间,产生真正作用于网络的流对象;

[0023] 通过所述完成事件在执行完所述流对象的流事件后,删除全局仿真网络拓扑架构中的这条流对象。

[0024] 第二方面,本申请实施例提供了一种全局流级别网络仿真系统,采用第一方面中任意一项所述的方法,包括:

[0025] 文件接收模块,配置为响应于仿真平台触发的第一目标事件,通过所述第一目标事件接收仿真环境所需的配置文件;

[0026] 拓扑构建模块,配置为解析所述配置文件构建全局仿真网络拓扑架构形成仿真环境,以通过所述全局仿真网络拓扑架构获取流对象中的流路径,从而根据流路径确定流事件;

[0027] 任务配置模块,配置为响应于仿真平台触发的第二目标事件,通过所述第二目标事件加载全局事件队列调度器,形成任务生成器,通过所述任务生成器产生仿真任务,将所述仿真任务划分成多个阶段,将所述仿真任务的每个阶段存储到全局变量中,以供全局调

用;所述仿真任务中配置有若干流对象;

[0028] 全局刷新模块,配置为通过全局刷新策略不断更新全局仿真网络拓扑架构中流对象传输的用时信息,获取各所述流对象在所述全局仿真网络拓扑架构用时最短的流路径,利用流路径完成当前流对象的流事件后,通过刷新全局仿真网络拓扑架构以静止相对仿真时间,并计算下一流对象的流路径耗时,在触发下一流对象的流事件时,恢复相对仿真时间;

[0029] 全局控制模块,配置为通过全局控制策略获取全局实时的流事件进展的阶段信息,根据全局仿真网络拓扑架构的仿真状态以及任务调度结果,控制触发下一流对象的流事件触发,以此预测仿真任务的阶段结束,并获取仿真任务的测试结果。

[0030] 第三方面,本申请实施例提供了一种全局流级别网络仿真装置,配置有第二方面中所述的全局流级别网络仿真系统。

[0031] 第四方面,本申请实施例提供了一种电子设备,包括:

[0032] 一个或多个处理器;

[0033] 存储器,用于存储一个或多个程序;

[0034] 当所述一个或多个程序被所述一个或多个处理器执行时,使得所述一个或多个处理器实现如第一中任一所述的方法。

[0035] 第五方面,本申请实施例提供了一种计算机可读介质,其上存储有计算机程序,所述计算机程序被处理器执行时实现如第一方面中任一所述的方法。

[0036] 本申请实施例中提供的技术方案,至少具有如下技术效果:

[0037] 1、由于预先搭建仿真平台,在仿真平台上接收仿真需求的配置文件,从而构建出满足需求的全局仿真网络拓扑架构,从而可以知道,利用该仿真平台可以搭建任意仿真需求的仿真环境,仿真平台具有仿真通用性的有益效果。

[0038] 2、由于采用了流级别的仿真环境,用户只需要对配置仿真任务后,获取仿真任务中的流对象,每个仿真任务通过若干流对象触发的流事件完成,因此只要控制流事件即可控制整个仿真进程,从而限定仿真任务的可靠性。

[0039] 3、由于采用通过全局刷新策略确定流事件的触发开始和完成,本方案中不需要考虑仿真任务的所有流对象中可能发生的所有流事件,利用流路径关联的流事件可能完成时间的预期值,本方案中为了加快仿真运行,因此通过流事件完成用时最短的确定流路径,从而确定待激活的流事件,在确定该流事件前先将流事件推进权利事件队列中,根据上一流事件完成触发全局刷新,然后将下一流事件触发开始事件,可以看出,基于事件驱动的仿真体系,需要在全局维护一个唯一的全局事件队列,根据全局事件队列控制所有流事件的发生和结束,同时,流事件的发生也随之推进全局仿真时间(相对仿真时间),以此保证了仿真时间可控性,利用刷新操作,不断增加和删除事件以此不断的更新流信息。

[0040] 4、由于采用了给流对象关联的流事件添加激活标志变量,在全局刷新时只需要遍历有活跃标记的流事件,激活相应的流事件后,控制流事件的发生和完成处理,有效减少了事件遍历和修改次数。

[0041] 5、由于采用了先根据流路径分析流对象可能触发的流事件,然后针对该些流事件先利用各流路径所需完成时间的预期值确定一条流路径,从而确定流对象关联的流事件,也就是说,在全局刷新后,获取流对象的流量剩余大小以及计算的完成时间(预期值)变量,

以此减少一部分不必要的流事件完成时间的计算,以此达到在新流事件产生和旧流事件完成时,实时更新以保障仿真进展中流传输的有效性和及时性。

附图说明

[0042] 图1为本申请实施例一中的全局流级别网络仿真方法流程图;

[0043] 图2为本申请实施例一中的全局流级别网络仿真框架图;

[0044] 图3为本申请实施例二中的全局流级别网络仿真系统模块图。

具体实施方式

[0045] 为了更好的理解上述技术方案,下面将结合说明书附图以及具体的实施方式对上述技术方案进行详细的说明。

[0046] 实施例一

[0047] 参考附图1-2所示,本申请实施例提供了一种全局流级别网络仿真方法,包括:

[0048] 步骤S101,响应于仿真平台触发的第一目标事件,通过所述第一目标事件接收仿真环境所需的配置文件。

[0049] 本实施例中的仿真平台可以理解作为一种底层仿真框架,通过可视化界面操作获取仿真环境所需的配置文件,操作人员无需具备高深的技术背景,只要具备可视化界面的操作技能即可,为广大仿真需求的用户减少了开发工作量,利用本实施例提供的仿真平台可实现任意仿真环境的配置,极大地减少了大量开发资源和时间,并且本申请给出的技术方案历经无数次实践校验,可以达到超快速的网络仿真速度效果,也可以为后续相关领域的快速开发提供一定的技术参考。

[0050] 进一步说明,本实施例里的仿真平台采用Python语言编写的底层逻辑框架,能够区分绝对仿真时间和相对仿真时间。其中,绝对仿真时间可以理解为现实时间,而相对仿真时间可以理解为仿真环境中所需的相关进程时间,并且在仿真环境中,当执行进程计算中需要用到时间变量时,使用相对仿真时间,当不考虑时间变量时,那么理解为仿真环境的相关进程,在静止的相对仿真时间内执行相应的动作。

[0051] 步骤S102,解析所述配置文件构建全局仿真网络拓扑架构形成仿真环境,以通过所述全局仿真网络拓扑架构获取流对象中的流路径,从而根据流路径确定流事件。

[0052] 本实施例中的所述配置文件中设置有网络拓扑信息和图信息,通过所述网络拓扑信息和所述图信息构建仿真环境所需的全局仿真网络拓扑架构,并存储后供全局调用;其中,所述图信息中配置有仿真节点、节点间的链接关系、节点权重,每个所述仿真节点配置有唯一的ID号,且每个所述仿真节点关联有与其连接的其他仿真节点的ID号,并反映在所述图信息中。本实施例中,通过节点权值获取网络实时的链路占用情况;通过所述网络拓扑信息实时存储网络实际在流的流信息。

[0053] 全局仿真网络拓扑架构包括若干仿真节点,各仿真节点可以是服务器、电/光交换机,也就是说,通过服务器、电交换机或光交换机组成所需的大规模仿真网络。现有大规模网络可以有纯电交换机组成的静态A1集群、光电混合A1集群、或者其他自定义网络集群。基于预先配置的仿真平台中构建全局仿真网络拓扑架构,并且将全局仿真网络拓扑架构图通过可视化界面展示给操作人员。进一步说明,所有仿真节点可以认为仿真环境中的设备,并

且,本实施例中给出的所有代码信息可以理解为Python语言的底层仿真框架。在配置文件中通过device.py表示网络中的所有仿真节点,不管服务器还是交换机,都代表仿真节点,并且每一个仿真节点中配置有唯一的ID号作为标识编号,并且每个仿真节点中存储有与其连接的其他仿真节点的编号,并反映在图信息中。

[0054] 进一步说明,基于底层仿真框架采用Python语言编写,本实施例中的配置文件通过infra_base.py表示网络拓扑信息,通过graph.py表示图信息,通过infra_base.py、graph.py构建仿真环境中的物理网络拓扑架构,并且将构建好的全局仿真网络拓扑架构存储到模拟器Simulator中,以供全局调用。其中,模拟器Simulator用于创建包含应用程序在真实环境中的变量和配置的模拟环境,本实施例中的模拟器Simulator创造的软件环境也就是本实施例中所说的仿真环境,通过将全局仿真网络拓扑架构存储到模拟器Simulator中,因此运行模拟器Simulator,当得到仿真交互时的执行结果与预期结果一致的判断结果时,即为仿真测试的最终需求,并不需要考虑底层硬件配置,只需要一台终端主机或终端服务器即可完成仿真测试,大大的减少了仿真环境的运行成本。

[0055] 本实施例中的图信息中通过vertex表示全局仿真网络拓扑架构中的仿真节点,例如几何形状中一个顶点;通过edge表示节点间的链接关系,也就是仿真节点间的线或边,通过仿真节点间的拓扑连线数量表示节点权重,在图信息中,可以通过节点权重计算网络实时的链路占用情况,并且本实施例中的网络拓扑信息实时存储有网络实际在流的流信息。

[0056] 本实施例中的全局仿真网络拓扑架构用于传输流对象,由于流对象中既定有传输起点和终点,那么可以生成传输流对象的若干条流路径。本实施例中仿真进程采用串行方式执行任务中的各个流对象的传输动作,考虑到本实施例中的仿真目的为速度需求,因此本实施例通过找出一条用时最短的流路径来确定后续的传输动作的开始,即为流事件。可以看出本实施例通过流量传输的用时长短来确定流路径,触发传输流对象的流事件,并且在确定流路径后,执行当前流对象的流开始事件,在流对象到达终点后,执行流完成事件。

[0057] 进一步说明,流对象可以定义为有向的传输对象,也就是说本实施例中的全局仿真网络拓扑架构给出有向图,通过流量表示流对象的大小/尺寸,流事件开始到完成执行的是流对象传输的过程。本实施例中定义在仿真节点间的传输对象为流对象,其关联的流信息包括确定的起点(源点)和终点,触发流对象开始传输的动作称为流开始事件,那么流对象从起点到终点完成的动作称为流完成事件,本实施例中通过流开始事件到流完成事件触发的时间获取当前流对象的用时长短。

[0058] 进一步说明,针对流对象中的相关信息称为流信息,流信息包括起点、终端、流大小(流量)、流路径(在全局仿真网络拓扑架构中的路由信息)以及流时间(流事件在流路径中完成的时间)。在控制触发流对象进行传输的动作中产生流开始事件,在产生流开始事件之前,到上一个流对象执行完流完成事件之间,不考虑耗费时间,相当于将相对仿真时间设置为静止状态,在该仿真时间静止的时间段,获取当前流对象在全局仿真网络拓扑架构所有的流路径,若不考虑时间要求,那么根据网络拓扑图中的若干流路径可以形成若干流事件,但是本实施例中为了实现超快速流级别的网络仿真,优选地,选择用时最短的一条流路径执行当前流对象的流事件。

[0059] 进一步说明,获取流对象的流路径中,通过routing/ecmp/bfs_ecmp_routing_static.py提供整个网络节点之间的所有最短路径,由于全局网络中每一个仿真节点配置

有一个唯一的ID号,因此仿真节点之间的连接关系可以采用3元组[src,dst,连接数]表达,由于全局仿真网络拓扑架构存储在模拟器的`Simulator.infra_base`中,可以通过`Simulator.get_infrastructure()`获取。本实施例中在执行全局刷新策略时,调用所有仿真节点间的所有最短路径。根据此方法可以根据不同的配置需求,选择相应需求的路径。但是当网络规模太大时,全局搜索所有仿真节点间的所有最短路径可能耗时会过长,本实施例中进一步阐述结合广度对优先搜索算法的优化方法。

[0060] 步骤S103,响应于仿真平台触发的第二目标事件,通过所述第二目标事件加载全局事件队列调度器,形成任务生成器,通过所述任务生成器产生仿真任务,将所述仿真任务划分成多个阶段,将所述仿真任务的每个阶段存储到全局变量中,以供全局调用;所述仿真任务中配置有若干流对象序列,按序列顺序执行流对象。

[0061] 其中,基于底层仿真框架采用Python语言编写,本实施例中的任务生成器可以采用`task_generator.py`表示,根据仿真配置需求选择相应的通信策略,比如在`communication_strategy`文件中配置有butterfly蝴蝶优化算法、ring算法、all-to-all算法在内的多种通信算法,以此设计产生相应的多条流对象。本实施例基于仿真平台的底层仿真框架规则下,可以添加新的通信策略。

[0062] 基于各仿真任务可以配置若干流对象,每个流对象关联有相应的流信息,流信息可以包括源仿真节点的id号、目的仿真节点的id号、流事件的开始时间、流事件触发后实时的流量剩余大小、当前流事件所属仿真任务、当前流事件所属阶段信息、预期流事件的完成时间、流事件发生的流路径、流事件的实际完成时间等信息。尤其考虑到,本实施例中为了加快仿真进程的运行速度,调用了仿真节点之间所有可能的最短路径信息,但是在不同的仿真需求中,可以根据需求选择不同的流路径,比如,纯点交网络架构下的静态路由和ECMP路由、光电混合网络架构下的无拥塞路由。

[0063] 本实施例中加载全局事件队列调度器,可以理解为操作人员预先通过可视化界面(窗口)设定仿真任务,当然仿真任务也可以在构建仿真环境的配置文件中设置,对此本实施例不做限定,本实施例中的全局事件队列调度器利用相关的调度算法确定当前需要执行的任务,可以先获取所有的仿真任务,再将仿真任务挂起,若要执行某一仿真任务,通过选择就绪态中的某一仿真任务激活后,执行该仿真任务。进一步说明,在加载全局事件队列调度器时,需要根据当前全局仿真网络拓扑架的网络状态和新的仿真任务,决定这个新的仿真任务是否能够现在被执行,并将当前的调度信息返回给主框架,在执行全局控制策略时,利用调度信息进行相应的判断。例如,纯电网络下的locality亲和性调度、光电混合网络下的无拥塞调度。本实施例中的仿真平台基于底层仿真框架,在遵循主事件驱动规则的前提下,可以添加新的调度策略。

[0064] 本实施例中通过加载全局事件队列调度器后,形成任务生成器,也就产生任务开始事件。本实施例中的任务生成器产生仿真任务后,对仿真任务的进程运行采用堆排序,并根据事件基类中重载运算逻辑来重新排序。

[0065] 本实施例中各所述仿真任务通过所述事件基类衍生有三个作用于流对象的事件子类:分别为任务开始事件、传输事件、完成事件。通过所述任务开始事件预设仿真任务参数,产生多条流对象,在检测所述仿真平台是否触发第二目标事件的同时,接收当前所述仿真任务的修改配置。通过所述传输事件控制仿真进程推进到某一时刻的瞬间,产生真正作

用于网络的流对象。通过所述完成事件在执行完所述流对象的流事件后,删除全局仿真网络拓扑架构中的这条流对象。

[0066] 进一步说明,基于底层仿真框架采用Python语言编写,事件基类采用Event表示,衍生的三个事件子类:任务开始事件Task start event、传输事件Transmit event、完成事件Completion event,该三个事件子类各司其职,但是又与全局刷新策略以及全局控制策略紧密相连,遵循高内聚低耦合的设计理念。任务开始事件Task start event中,通过代码TaskStartEvent类中的rapidnetsim/core/event/task_start_event.py表示,根据预设仿真任务参数,产生多条流对象,在检测仿真平台是否触发第二目标事件的同时,接收当前仿真任务的修改配置,以此达到灵活动态调整的能力。其中,检测仿真平台是否触发第二目标事件可以理解为检测当前任务队列中是否增加新的仿真任务。接收当前仿真任务的修改配置可以理解为接收配置文件中修改任务的类型。因此可以看出,本实施例中给出的获取仿真任务的放置为获取配置文件。可以是仿真环境构建完成后执行仿真任务时获取配置文件,也可以是预先构建仿真环境的配置文件中预先设置仿真任务。传输事件Transmit event中,通过代码Flow Transmit Event类中的rapidnetsim/core/event/flow_transmit_event.py表示,其目的为,当相对仿真时间推进到某一时刻的瞬间,产生真正作用于网络的流实体。完成事件Completion event中,通过代码Flow Completion Event类中的rapid net sim/core/event/flow completion event.py表示,在根据全局刷新策略获取到流对象确定的流事件,以及根据全局控制策略控制完成流对象的流事件后,删除全局仿真网络拓扑架构中的这条流对象。

[0067] 步骤S104,通过全局刷新策略不断更新全局仿真网络拓扑架构中流对象传输的用时信息,获取各所述流对象在所述全局仿真网络拓扑架构用时最短的流路径,利用流路径完成当前流对象的流事件后,通过刷新全局仿真网络拓扑架构以静止相对仿真时间,并计算下一流对象的流路径耗时,在触发下一流对象的流事件时,恢复相对仿真时间。

[0068] 本实施例中所述全局刷新策略中,分析每个所述流对象关联的流信息中根据流路径可能触发的流事件,根据各所述流路径的完成时间预期值,确定待激活的流事件后,并标记为活跃状态,在执行全局刷新时,将已完成的流事件的标记改为不活跃后,同时遍历标记为活跃状态的所有流事件。

[0069] 进一步说明,基于底层仿真框架采用Python语言编写,全局刷新策略采用代码Refresh flow info中的rapidnetsim/core/network_refresh.py表示,在仿真环境的主线程循环执行仿真任务的流事件过程中,不断刷新整个网络的流信息,检测并更新最早需要完成的流对象。本实施例中循环执行仿真任务的流事件过程中,每一个流事件代表不同的流对象,由于仿真任务中可以配置有若干个流对象,那么在控制流对象传输触发流事件时,需要了解当前网络中的流信息。本实施例中通过代码refresh completion event()检测并获取到最早完成的流信息,并且刷新全局的记录标志量。最早完成的流信息可以理解为正常当前网络中仿真任务中的流对象,获取各个流对象的传输用时,当用时最短时,那么可以确定最早完成的流路径。本实施例中通过代码caculate and modify finish_time()表示根据下一流对象的流大小计算执行流事件完成所需的时间,以此计算获取全局各流对象用时最短的流事件。

[0070] 进一步说明,本实施例中的任务生成器采用事件基类的初始化函数控制仿真进程

的相对时间和绝对时间,并设定激活状态变量进行加速时调用。可以理解为任务生成器的生命周期相当于当前获取的仿真任务所执行的仿真程序的生命周期,从任务生成器产生的第一个仿真任务,并触发执行第一个流事件开始,到所有的仿真任务及其流事件完成,或者直接接收相关的仿真结束指令结束仿真程序,可以看出该过程相当于仿真进程的大脑和心脏,不断的更新流事件,然后执行流事件。本实施例中将构建的全局仿真网络拓扑架构存储到模拟器的主核心Simulator.py中,也就是说,该主核心中存储有仿真环境的基础配置信息、全局仿真网络拓扑架构、调度新、日志信息等全局调用的信息,并且维护这全局仿真时间,也就是相对仿真时间。本实施例中采用代码core_run控制与Event事件的时间关系。通过Event事件类的初始化函数控制仿真器的相对仿真时间和绝对时间的关系,并设定激活状态变量提供给程序加速用。通过core_run进行堆排序时,会根据Event基类中重载_It_运算逻辑来进行重新排序。

[0071] 本实施例中每个流对象关联的流信息根据流路径可以触发不同的流事件,将每个可能发生的流事件配置活跃状态标记以及动态变化的完成时间预期值(由于完成时间预期值为动态变量,因此可以设置-1来表示完成时间预期值还未被计算)。可以看出,在仿真进程的主事件驱动逻辑中,预先获取流对象的各个流路径,计算流对象在流路径上的传输用时长短,本实施例中选出流对象用时最短的流路径,并将关联的流事件配置为活跃状态,因此可以确定流对象需要的流事件,再次利用全局事件队列调度器执行流事件调度时,可以直接跳过流对象中标记为不活跃的流事件,以此省去再次筛选流事件以及避免重复性错误。

[0072] 本实施例中通过对全局仿真网络拓扑架构进行刷新时,可以理解为遍历全局记录的所有可能的流事件(每一个流路径都可能触发流事件),在根据流路径计算流事件完成时间预期值之前,检测流对象中的流路径是否已计算出流事件的完成时间预期值,若没有计算,则先计算后更新到流对象关联的流信息对应的流路径的关联信息中,因此可以知道,根据流路径以及流事件完成时间预期值中确定当前流对象中活跃标记的流事件,因此只要遍历有活跃标记的流路径关联的流事件,以此省去在计算完成时间中遍历路径产生的资源和时间消耗。遍历完所有活跃标记的流事件的完成时间期望值后,由于刷新之前的流事件已完成,那么将前一次刷新记录的全局最小完成期望值的流信息中活跃标记的流事件设定为不活跃,并将当前流事件中最小完成时间期望值的流信息中流事件记录在全局,并添加到全局事件驱动队列中,确保全局事件驱动队列中有且仅有一个可被执行的流完成事件,以此达到在缩减资源消耗和计算时间的同时,确保仿真进程的正确性,从而以此达到全局驱动和网络刷新的优化。

[0073] 步骤S105,通过全局控制策略获取全局实时的流事件进展的阶段信息,根据全局仿真网络拓扑架构的仿真状态以及任务调度结果,控制触发下一流对象的流事件触发,以此预测仿真任务的阶段结束,并获取仿真任务的测试结果。

[0074] 所述全局控制策略中,分布式控制所述仿真任务的多个阶段,并通过不同的通信策略、通信对象以及流任务的数量,全局记录实时流进展的阶段信息。

[0075] 基于底层仿真框架采用Python语言编写,全局控制策略采用代码Stage controller中的rapidnetsim/core/stage_controller.py表示,其有效控制分布式仿真任务的多个阶段。由于不同的通信策略会产生轮数不同,通信对象不同,数量不同的流任务,

本实施例中全局记录了实时流进展的阶段信息,并在阶段控制模块中结合网络占用状态和GPU分配状态,以及调度器的调度结果,综合控制下一阶段将要产生的新的流对象以及准确判断阶段结束,与每个仿真任务的运行结果,比如,仿真环境在仿真任务结束后的网络状态。

[0076] 相比现有包级别仿真进程,本实施例中缩减了流事件的刷新数量(本实施例中预先选择流路径后确定流事件),以此减轻全局事件队列,尤其对成千上万节点的分布式训练任务产生的流对象来说,依然需要花费昂贵的计算资源和时间来进行处理,通过将仿真任务划分成多个阶段,给每个阶段的流对象所触发的流事件配置完成时间预期值以及活跃状态标记,本实施例中只需要根据完成时间预期值对可能的流事件预先进行活跃标记,遍历所有筛选出活跃标记的流事件,以此大幅度减少全局网络刷新的遍历次数和计算开销。

[0077] 当全局仿真网络拓扑架构中的网络规模较大时,导致全局循环搜索所有最短路径耗时较长,本实施例中采用最短路径以优化搜索时间。

[0078] 将全局仿真网络拓扑架构的每个仿真节点配置有唯一的ID号,仿真节点间的连接关系采用三元组表示,网络拓扑存储在Simulator._infra_base中,并通过Simulator.get_infrastructure()获取。基于全局仿真网络拓扑架构的有向性,在搜索最短路径前,利用广度优先搜索的思想,计算出最短步数矩阵。将行表示起点仿真节点的ID号,列表示目的仿真节点的ID号,设定最短步数矩阵的每个元素初始化为无穷大,每个仿真节点做一遍广度优先搜索,更新最短步数矩阵。广度优先搜索思想中,从某个起点仿真节点开始推入BFS队列,先记录层数以及推出一个当前仿真节点,并记为临时仿真节点,同时记录已经访问过的节点,遍历当前仿真节点所连接的其他仿真节点,若没有被访问过,则将当前仿真节点所对应的临时仿真节点和相连的其他仿真节点对应应在最短步数矩阵处的元素值置为1,并将当前仿真节点到相连的其他节点对应应在最短步数矩阵处的元素值置为层数+1,将相连的其他仿真节点和层数+1推入BFS队列,循环操作,直到BFS队列为空停止。

[0079] 从步骤S101-S105中可以看出,本实施例中基于事件驱动的仿真体系,需要在全局维护一个唯一的全局事件队列,根据全局事件队列控制所有流事件的发生和结束,同时,流事件的发生也随之推进全局仿真时间(相对仿真时间),以此保证了仿真时间可控性,利用刷新操作,不断增加和删除事件以此不断的更新流信息。若直接遍历这个队列,那么将会带来很高的资源和时间开销,现有开发思想中采用并行模式或者多线程模式或者多实例加速方式,但是均无法解决仿真时间过久问题的原因。考虑到需要给仿真进程加速,本实施例中先根据流路径分析流对象可能触发的流事件,然后针对该些流事件先利用各流路径所需完成时间的预期值确定一条流路径,从而确定流对象关联的流事件,也就是说,在全局刷新后,获取流对象的流量剩余大小以及计算的完成时间(预期值)变量,以此减少一部分不必要的流事件完成时间的计算,以此达到在新流事件产生和旧流事件完成时,实时更新以保障仿真进展中流传输的有效性和及时性。同时,本实施例中给流对象关联的流事件添加激活标志变量,在全局刷新时只需要遍历有活跃标记的流事件,激活相应的流事件后,控制流事件的发生和完成处理,有效减少了事件遍历和修改次数。

[0080] 实施例二

[0081] 参考附图3所示,本申请实施例提供了一种全局流级别网络仿真系统,采用实施例一中任意一项所述的方法,包括:

[0082] 文件接收模块101,配置为响应于仿真平台触发的第一目标事件,通过所述第一目标事件接收仿真环境所需的配置文件。

[0083] 拓扑构建模块102,配置为解析所述配置文件构建全局仿真网络拓扑架构形成仿真环境,以通过所述全局仿真网络拓扑架构获取流对象中的流路径,从而根据流路径确定流事件。

[0084] 任务配置模块103,配置为响应于仿真平台触发的第二目标事件,通过所述第二目标事件加载全局事件队列调度器,形成任务生成器,通过所述任务生成器产生仿真任务,将所述仿真任务划分成多个阶段,将所述仿真任务的每个阶段存储到全局变量中,以供全局调用;所述仿真任务中配置有若干流对象。

[0085] 全局刷新模块104,配置为通过全局刷新策略不断更新全局仿真网络拓扑架构中流对象传输的用时信息,获取各所述流对象在所述全局仿真网络拓扑架构用时最短的流路径,利用流路径完成当前流对象的流事件后,通过刷新全局仿真网络拓扑架构以静止相对仿真时间,并计算下一流对象的流路径,在触发下一流对象的流事件时,恢复相对仿真时间。

[0086] 全局控制模块105,配置为通过全局控制策略获取全局实时的流事件进展的阶段信息,根据全局仿真网络拓扑架构的仿真状态以及任务调度结果,控制触发下一流对象的流事件触发,以此预测仿真任务的阶段结束,并获取仿真任务的测试结果。

[0087] 实施例三

[0088] 本申请实施例提供了一种全局流级别网络仿真装置,配置有权利要求11中所述的全局流级别网络仿真系统。本实施例中的装置可以为一种具有仿真服务的服务器或终端设备。

[0089] 实施例四

[0090] 本申请实施例提供了一种电子设备,包括:

[0091] 一个或多个处理器;

[0092] 存储器,用于存储一个或多个程序;

[0093] 当所述一个或多个程序被所述一个或多个处理器执行时,使得所述一个或多个处理器实现实施例一中任一所述的方法。

[0094] 本申请实施例还一种计算机可读介质,其上存储有计算机程序,所述计算机程序被处理器执行时实现如实施例一中任一所述的方法。

[0095] 步骤S101,响应于仿真平台触发的第一目标事件,通过所述第一目标事件接收仿真环境所需的配置文件;步骤S101,解析所述配置文件构建全局仿真网络拓扑架构形成仿真环境,以通过所述全局仿真网络拓扑架构获取流对象中的流路径,从而根据流路径确定流事件;步骤S101,响应于仿真平台触发的第二目标事件,通过所述第二目标事件加载全局事件队列调度器,形成任务生成器,通过所述任务生成器产生仿真任务,将所述仿真任务划分成多个阶段,将所述仿真任务的每个阶段存储到全局变量中,以供全局调用;所述仿真任务中配置有若干流对象;步骤S101,通过全局刷新策略不断更新全局仿真网络拓扑架构中流对象传输的用时信息,获取各所述流对象在所述全局仿真网络拓扑架构用时最短的流路径,利用流路径完成当前流对象的流事件后,通过刷新全局仿真网络拓扑架构以静止相对仿真时间,并计算下一流对象的流路径,在触发下一流对象的流事件时,恢复相对仿真时

间;步骤S101,通过全局控制策略获取全局实时的流事件进展的阶段信息,根据全局仿真网络拓扑架构的仿真状态以及任务调度结果,控制触发下一流对象的流事件触发,以此预测仿真任务的阶段结束,并获取仿真任务的测试结果。

[0096] 本领域内的技术人员应明白,本发明的实施例可提供为方法、系统、或计算机程序产品。因此,本发明可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实施例的形式。而且,本发明可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质(包括但不限于磁盘存储器、CD-ROM、光学存储器等)上实施的计算机程序产品的形式。

[0097] 本发明是参照根据本发明实施例的方法、设备(系统)、和计算机程序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器,使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的装置。

[0098] 这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中,使得存储在该计算机可读存储器中的指令产生包括指令装置的制造品,该指令装置实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能。

[0099] 这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上,使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理,从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的步骤。

[0100] 尽管已描述了本发明的优选实施例,但本领域内的技术人员一旦得知了基本创造性概念,则可对这些实施例作出另外的变更和修改。所以,所附权利要求意欲解释为包括优选实施例以及落入本发明范围的所有变更和修改。

[0101] 显然,本领域的技术人员可以对本发明进行各种改动和变型而不脱离本发明的精神和范围。这样,倘若本发明的这些修改和变型属于本发明权利要求及其等同技术的范围之内,则本发明也意图包含这些改动和变型在内。

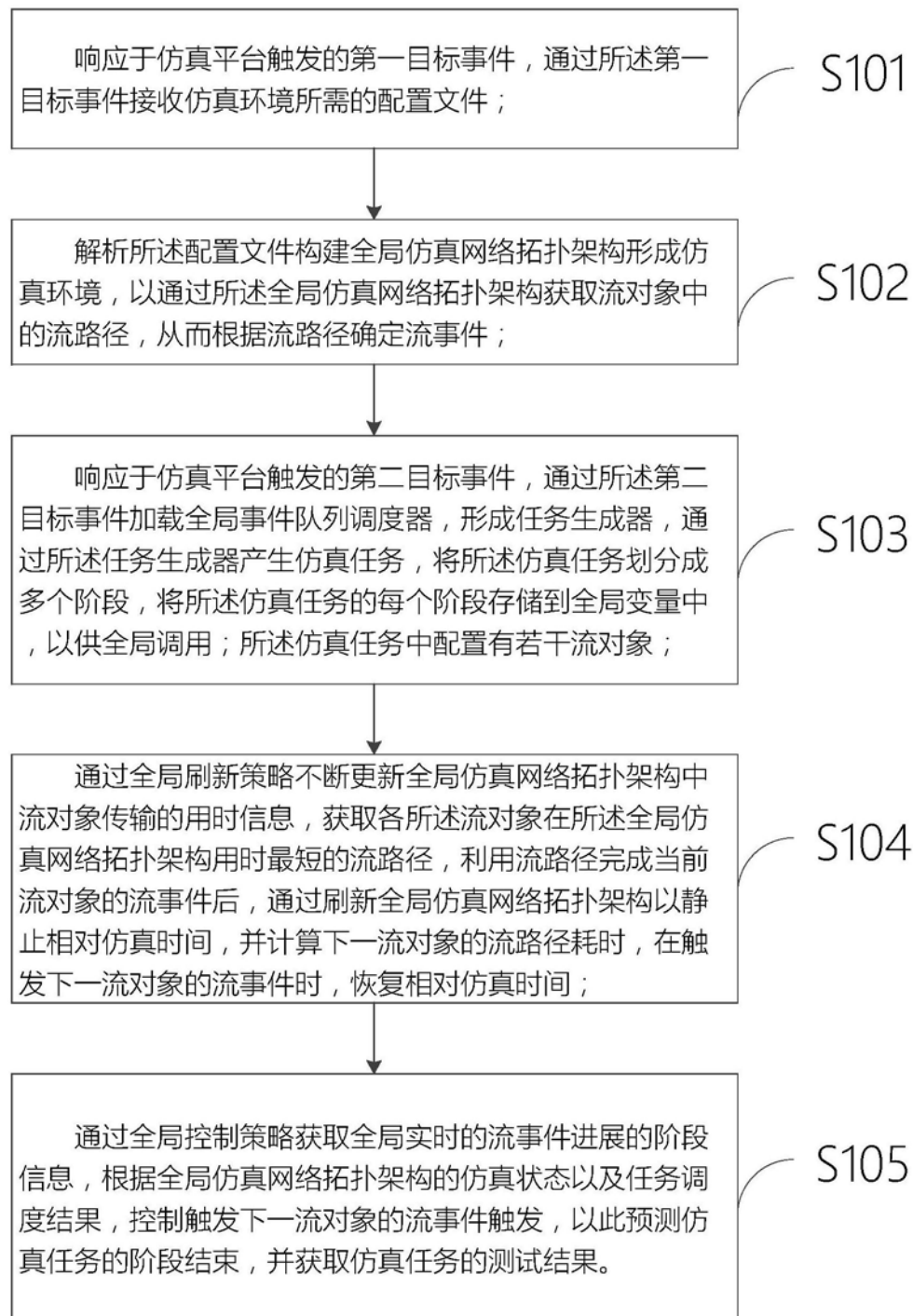


图1

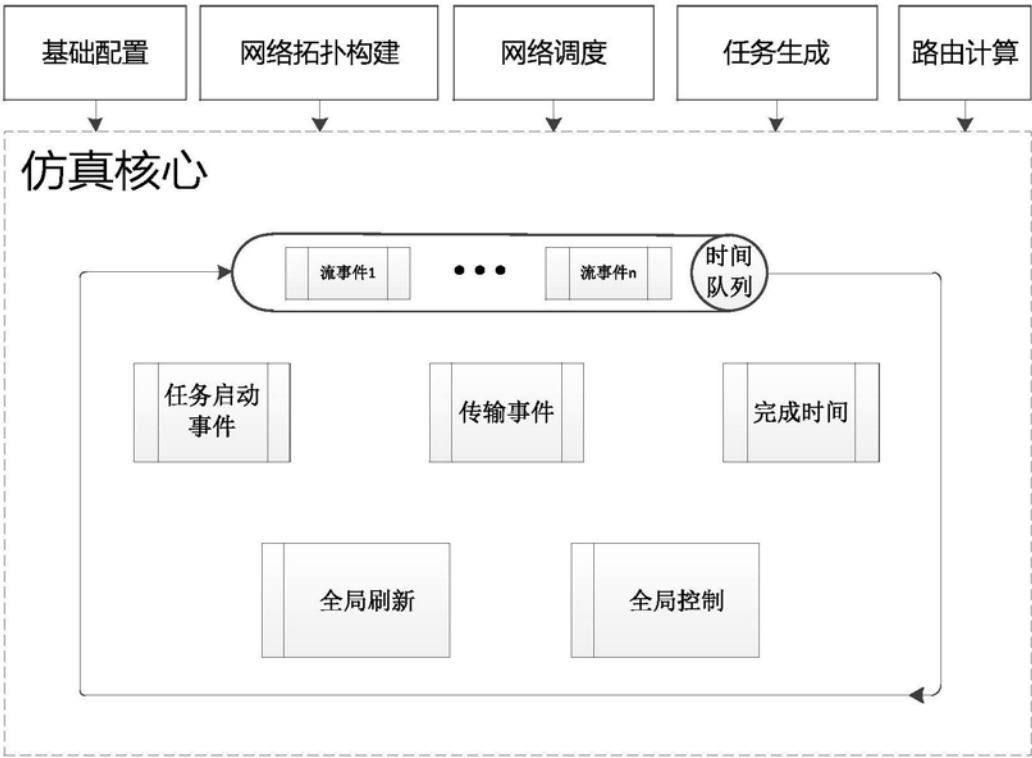


图2

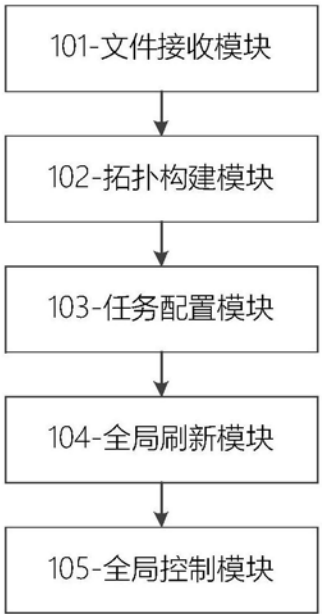


图3