

智慧配电网辅助规划平台分布式计算 数据通信实现方法

朱佳明¹ 王浩国¹ 方焯阳¹ 杨金涛¹ 杨松坤² 吴倩²

(1. 国网浙江省电力有限公司杭州市钱塘区供电公司 2. 天地电研(北京)科技有限公司)

摘要: 大数据的出现为智慧配电网的规划和管理注入了新的活力,对配电网中海量异构的信息进行充分整合、分析和挖掘,可解决智慧化配电网规划业务带来的各种问题和挑战;与此同时,这也需要强大的算力作支撑。随着分布式技术和计算技术日益成为加速大数据分析最有效的解决方案,研究如何在集群环境中配置分布式计算的数据通信实现方法,使得各任务得以可靠、灵活、快速地执行是本文的研究内容。本文介绍了RabbitMQ通信技术,并基于RabbitMQ中间件开展了通信架构设计、通信模式选择和消息格式设计,然后阐述了系统的可伸缩性和健壮性的实现途径。通过对实际电网规划数据进行计算用时分析,验证了所提分布式快速计算架构和数据通信架构设计的可行性和实用性。

关键词: 配电网规划; 分布式计算; 通信架构; 消息格式

0 引言

配电网规划软件平台对于实现智慧配电网智能辅助规划十分重要。配电网规划要求获取电网的现状态数据和规划态数据。现状态数据包括电网模型信息、一次设备的电气参数、连接关系以及二次设备的配置和台账数据。规划状态数据是根据多元负荷预测和分布式电源历史运行数据合理预测得出的未来年份配电网数据。在配电网规划业务中,除了大量计算所需的数据之外,还需要大量的地理接线图数据和拓扑图数据。这些数据来自不同系统,需要解决系统之间的数据通信问题。另外,配电网的规模庞大,规划离不开各种辅助计算,如拓扑计算,潮流计算等,计算过程需要处理大量的节点,耗费时间长;如果计算速度慢,将无法满足规划工作的需求。

可见,辅助规划平台的设计要想达到大范围工程化实用的目的,需要进行分布式架构设计并解决大规模节点快速计算的问题。目前主流的分布式设计架构^[1-4]通常基于微服务架构、容器化部署、消息队列和事件驱动等技术,将多种技术组合在一起,能够构建高效、可伸缩和可靠的分布式计算服务系统,以满足大规模系统或平台的应用需求^[5-10]。

为此,本文提出一种基于RabbitMQ消息中间件的分布式计算架构,以有效支撑智慧配电网辅助规划平台业务应用的信息交互和快速计算。本文开展的工作为采用RabbitMQ消息中间件作为平台通信架构基础,以解决跨机器和跨系统之间的异构系统数据交互的问题。该架构具备可伸缩、可扩展和负载均衡性,即使在单点故障情况下,仍能支持基于数据级任务分

解的分布式计算。数据交互和模块协同均通过消息传递实现；另外，为满足规划业务的具体需求，还设计了一种可扩展的通信数据格式以支持多类型的消息传输。

1 数据通信实现

1.1 RabbitMQ 技术

RabbitMQ 是一款强大的面向消息的中间件，适用于复杂的全局拓扑和消息路由。独立于传输协议，可在不同网络和传输层协议上运行；其套接字能够感知路由和网络拓扑，通过交换机和绑定定义消息路由，可实现消息的精确控制^[11-13]。

其构成实体中，Exchange（交换机）用于接收消息并将其路由到一个或多个队列中。Queue（队列）用于存储消息，接收任务或消息。Consumer（消费者）是订阅队列的应用程序，用于从队列中接收和处理消息。Producer（生产者）是发布消息到 Exchange 的应用程序。RabbitMQ 可通过使用不同类型的队列和交换机来实现不同的 Request/Response（请求 / 响应）模式。

分布式快速计算需要高吞吐量的消息传递和队列管理以支持任务分发、并行计算和结果汇总等操作。因此，本文使用 AMQP 套接字用于应用程序之间的通信，选择 WebSockets 套接字用于 Web 应用程序与服务器之间的通信。

1.2 通信架构设计

根据智慧配电网规划系统的业务功能需求，辅助规划平台由客户端、服务实例、执行进程、管理模块 4 个部分组成^[14-15]，与辅助规划平台进行数据交互的系统为外部数据源。平台内部及外部数据源均通过 RabbitMQ 消息中间件进行数据交换。

信息交互图如图 1 所示，客户端主要功能包括后台服务的启停管理、计算参数设置、计算请求发

起、计算结果展示。外部数据源为电网资源业务中台和数据中台，提供电网模型数据、年份断面数据、台账数据等。管理模块包含负载均衡中间件和实例心跳监测模块，负责任务和数据的全局负载均衡、服务实例的心跳管理、后台服务管理等；还负责计算任务切分、计算结果回收等。服务实例是基于 Redis 内存数据库的数据平台，支持符合公共信息模型（CIM 2.0）文件导入，可向外部数据源请求所需时间断面的电网数据、接收外部数据源推送的动态更新准实时数据。执行进程是包含若干算法模块的程序，如分布式电源出力概率性分析、源网荷储一体化平衡计算等。

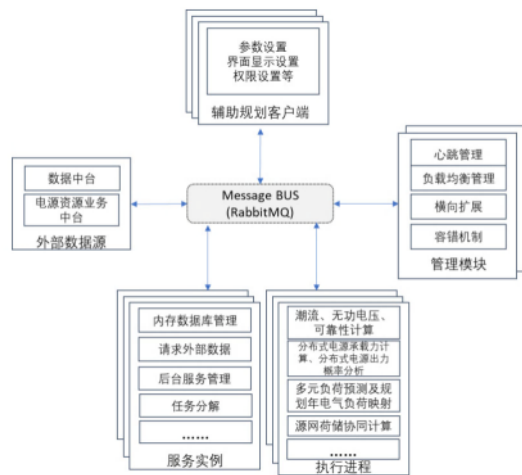


图 1 辅助规划系统信息交互图

实际部署时，可配置多台辅助规划服务器，每台服务器分别部署 1 个管理模块（Master）、若干个服务实例、若干个执行进程、若干个辅助规划客户端，分布式快速计算通信架构设计如图 2 所示。

图 2 中，智慧配电网辅助规划平台服务器集群的初始化、计算任务请求的数据流向以及配合关系如下：

- 1) 客户端通过 Exchange 向服务实例发起计算请求，启动服务实例，并可获取可用服务实例信息。
- 2) 服务实例在启动后首先必须向 Master 管理模

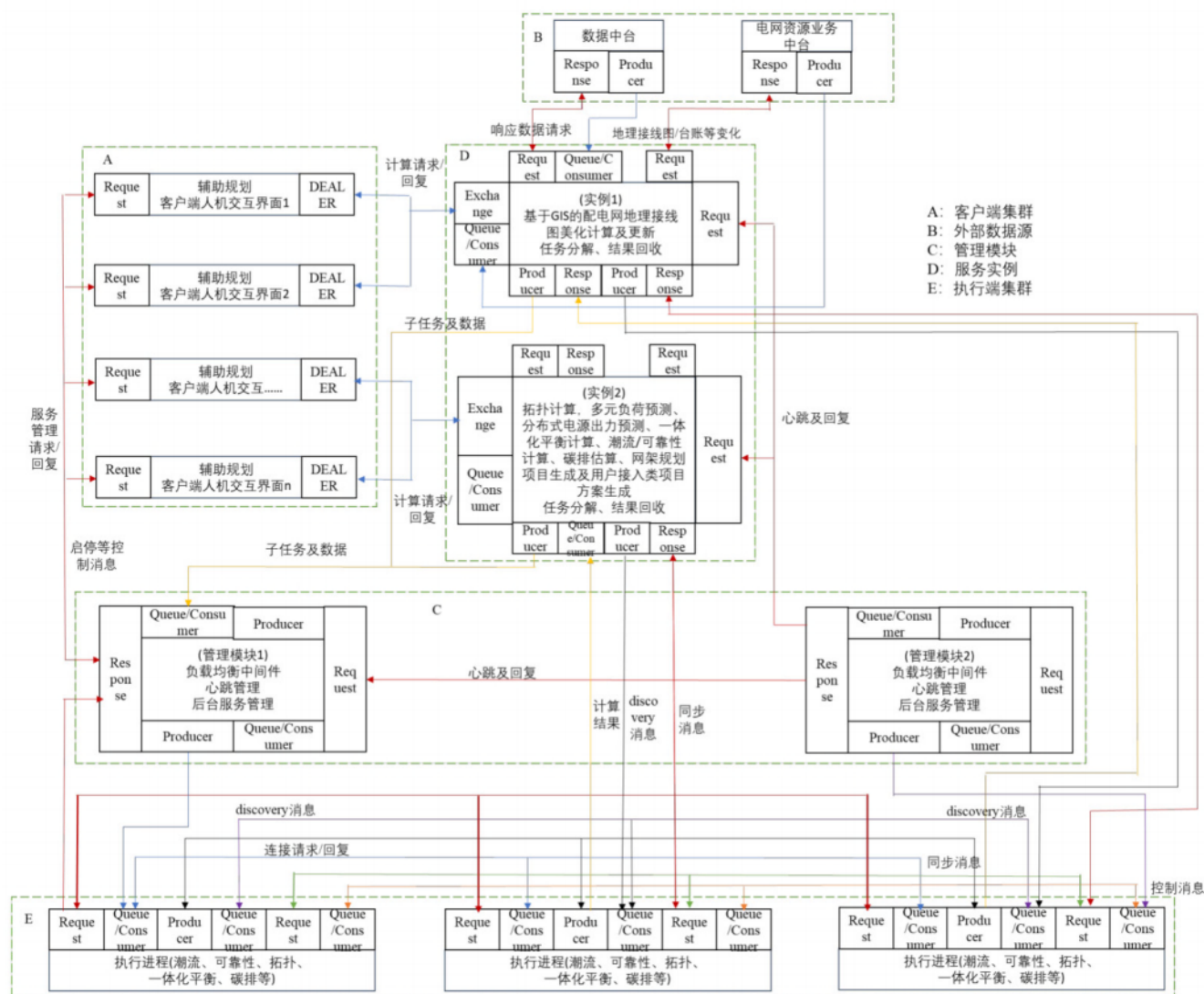


图 2 分布式通信架构

块“注册”自己的信息，包括所在机器的 IP、绑定的端口号等。注册后，所有实例和 Master 管理模块进行心跳互测。服务实例注册成功后再通过 Response-Request 从电网资源业务中台或数据中台主动请求电网数据，通过 Producer-Queue/Consumer 接收地理接线图 / 台账变化更新数据，初始化电网模型数据（CIM）、配网地理接线图形数据（GIS）及各类设备台账数据后，服务实例还将任务切分生成子任务及计算所需数据集以完成任务分解，随后子任务和数据

集被推送至并列运行的管理模块。

3) Master 管理模块通过 Producer-Queue/Consumer 将子任务均衡派发给各个执行进程，并监视各执行进程的心跳信息。

4) 执行端启动后先从 Master 管理模块获取服务实例绑定信息，再连接至服务实例完成初始化；然后按照自身获得的子任务业务类型调用相应算法模块进行计算，计算完毕后通过 Producer-Queue/Consumer 返回给对应的服务实例。

5) 服务实例在子任务结果收集完毕后进行结果组装，并将结果写入数据库后通知客户端计算完毕。

6) 客户端可以通过 Master 管理模块对后台服务程序进行管控，通过 Response-Request 向管理模块发送启、停信号，管理模块判断服务实例状态之后可对实例进行启停操作。

1.3 通信模式选择

根据规划业务的需求使用不同的队列和交换机组合来实现以下各类消息的通信模式：

1) 设备发现消息（discovery/hello 消息）是简单的一对一通信，用于建立基本的连接，选取交换机类型为直接交换机。方法是创建一个简单的队列（discovery/hello_queue），发送 discovery/hello 消息的应用程序将消息发布到 discovery_queue 队列，接收 discovery/hello 消息的应用程序订阅并从此队列中接收消息。

2) 控制消息使用自定义消息格式来定义，包含指令或控制信息的数据，需要特定的路由规则，因此选取交换机类型为主题交换机。发送控制消息的应用程序将消息发布到特定的主题交换机或队列，以便接收方可以按需订阅并处理这些消息。

3) 连接请求 / 回复消息是建立通信连接的一部分，需要精确的路由，选取交换机类型为直接交换机或队列。发送连接请求的应用程序将连接请求消息发布到一个直接交换机或队列，服务器应用程序订阅并接收连接请求消息，然后发送连接回复消息以响应连

接请求。

4) 同步消息使用请求 / 响应模式中，客户端发送请求消息，服务器接收并处理请求并发送响应消息，需要两者之间匹配以实现同步通信，选择直接交换机或主题交换机类型，使用专门的队列或交换机来处理同步请求和响应。

5) 心跳及回复用于维持连接的活跃状态，需要广播给所有连接设备，因此选用的交换机类型为扇出交换机。心跳消息定期发送到一个专用的队列或交换机，以保持连接的活跃状态；接收端定期检查心跳队列并响应心跳消息。

6) 服务管理请求 / 回复用于管理和监控系统状态，使用自定义消息格式，包括管理命令和操作，使用特定的路由规则，因此选用的交换机类型为主题交换机。发送管理请求的应用程序将请求消息发布到管理队列，接收方接收并处理请求，然后发送回复消息以响应管理操作。

1.4 消息格式设计

在配电网辅助规划平台中，所有功能模块之间的配合与数据交互都是基于消息的，合理的通信消息格式设计非常重要。图 3 为通信消息通用格式，由消息帧头、消息体、附加信息 3 部分组成。

消息帧头用于确保消息的正确处理和路由。其中，帧类型用于标识消息帧的类型，表示这是一个方法帧、请求帧或心跳帧等。规划平台中各类高级应用或计算大部分属于方法帧，例如各类计算及结果返

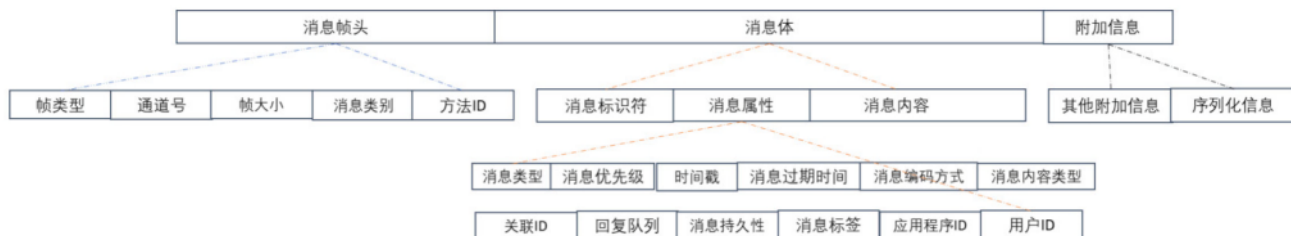


图 3 通信消息格式

回、出错处理、启停控制、数据更新等。通道号表示消息帧所属的通道号，用于多通道复用。帧大小表示消息帧的长度，以字节为单位。消息类别用于标识消息的类别，如连接、通道、交换机、队列等。方法 ID 用于标识执行的具体方法，与消息类别相关联。

消息体包含需要传递的实际数据内容。其中，消息标识符是消息 ID，用于消息的唯一性检查。消息属性提供了消息的类型、优先级、时间戳、编码方式等。消息的附加信息通常包含在消息属性中：消息类型代表消息的类型或用途；消息优先级允许接收端根据消息的重要性或紧急性来优先处理消息；时间戳是消息创建时的时间；消息过期时间指定消息的有效期限；消息编码方式指定消息内容的编码方式，本文采用 UTF-8 编码方式；消息内容类型指示消息内容的类型，本文使用 JSON 或 XML 方式；关联 ID 用于关联请求和响应消息，在请求消息和响应消息之间传递相同的关联 ID；回复队列指示接收端可以将响应消息发送到的队列，用于实现请求 - 响应模式；消息持久性指示消息是否持久化，本文选用持久化方式；用户 ID 指示发送消息的用户或应用程序的标识符。消息内容是消息的主要数据内容，本文采用文本、JSON 或 XML 格式，消息的实际有效载荷就存在于消息内容中，用于传递应用程序所需的信息。

附加信息并不是必需的，可根据应用程序需求按需配置。

discovery/hello 消息是 AMQP 协议握手的一部分，心跳消息、连接请求 / 回复消息、同步消息均由 RabbitMQ 自动处理，无需用户干预。需要用户自行定义消息格式的是控制消息和服务管理请求 / 回复消息，本文所设计消息格式如下：

(1) 控制消息包含指令或控制信息，使用 JSON 对象描述，包含命令类型和相关参数。

定义其以十六进制表示的帧头信息为：[0x01]

[0x03] [0x01] [0x0018] [0x03E8] [0x000A]，帧起始标识 0x01 代表消息帧的起始；帧类型 0x03 表示这是一个内容帧（Content Frame）；通道号 0x01 指示消息帧所属的通道号；帧大小 0x0018 表示整个消息帧的大小为 24 字节；消息类别 0x03E8 表示这是一个控制消息类别；方法 ID 0x000A 表示这是控制消息类别中的具体方法。

消息体格式为：

```
{
  "消息标识符": "message_12345",
  "目标设备": "device_1",
  "消息属性": {
    "命令类型": "balance_calc",
    "参数 1": "value1",
    "参数 2": "value2"
  },
  .....
},
"消息内容": "源网荷储一体化平衡计算"
```

(2) 服务管理 / 请求消息包含管理命令和相关参数，服务管理回复消息包含执行结果。

以管理模块查看某执行端的服务管理请求消息为例进行说明，定义其以十六进制表示的帧头信息为：

[0x1] [0x15] [0x11] [0x0028] [0x04F8] [0x001A]，帧类型 0x15 表示这是一个方法帧；消息类别 0x04F8 表示这是一个服务请求消息类别；方法 ID 0x001A 表示这是服务请求消息类别中的方法，1 表示启动服务，2 表示停止服务。

消息体按以下方式构建：

```
{
  "命令": "process.start",
  "服务名称": "XXX",
  "参数项": {
```

```
" process_name": "XXXX", // 进程名称
" process_id " : "XXXX", // 进程 ID
"operation": start, // 操作类型
}
}
```

其余的控制消息和服务管理请求/回复类消息均需按照协议规范和操作流程要求构建消息帧头和消息体，以确保消息格式正确、协议符合要求。需要注意的是，消息格式不是一成不变的，可根据需求进行扩展和修改。

1.5 可伸缩性和健壮性设计

各功能模块间相互配合，通过消息中间件不同套接字的配合使整个分布式系统具备可伸缩性和健壮性。

1) 架构可伸缩性设计

部署前，用户可根据需求配置 Master 管理模块、服务实例个数，执行进程和客户端可任意部署；部署后，Master 管理模块、服务实例、执行进程均可动态增减。

Master 管理模块使用 AMQP 套接字和消息队列实现服务实例的任务分发和负载均衡。通过配置心跳检测，定期向服务实例发送心跳请求，如果服务实例未能响应，则将其标记为断开连接。当有一个或几个实例发生异常，管理模块会通知所有执行进程实例发生变化的消息，执行进程随后刷新连接，从而达到系统运行时计算服务实例动态增删、执行进程动态连接的目的。

另外，发布-订阅模式使用扇出交换机，将消息广播给多个绑定的队列，使多个消费者订阅相同的消息流。复杂的消息路由和筛选使用主题交换机，生产者根据消息主题发布消息，消费者使用通配符订阅感兴趣的，使不同消费者接收特定类型的消息。使用负载均衡器确保系统能够处理大量消息流和连接流，以应对不断增长的负载。

2) 架构健壮性设计

由于采用异步的方式处理消息，可能出现任务丢失、结果未按时回收的情况，故每个执行进程在收到 Master 管理模块派发的任务后，首先通过“同步消息”与管理模块同步。如果管理模块探测到某个执行进程在规定的时间内 3 次均未同步成功，则删除此执行进程，并重启另一个进程，将未派出的子任务重新派发给新的执行进程。如果多个执行进程在规定的时间内 3 次均未与某个管理模块同步成功，则认为此管理模块故障，自动同步至其他管理模块，向同步成功的管理模块重新注册，接收并回应其发出的心跳请求。还配置有消息重试策略以处理由于临时问题而失败的消息，将无法处理的消息发送到死信队列，以便稍后进行分析和处理。

另外，实施限流策略防止消息发送者过快地向消息代理（RabbitMQ 服务器）发送消息，导致系统不稳定。通过建立 RabbitMQ 服务器集群以确保消息代理的高可用性，如果一个节点失败，其他节点仍可继续处理消息，避免单点故障。最后，使用备份还原机制还原消息代理的状态，并在灾难恢复时提供支持。

2 应用实例

以某县级市项目为例验证实际运用的结果。

(1) 数据交互性能测试

配置 16 台 PC 机及千兆交换机测试大数据架构的通信性能，操作系统为 Win 10，CPU 型号为 Intel (R) Xeon (R) CPU E5-2643 v4 @ 3.40GHz；内存为 DDR4 128G，内存带宽为 85.2GB/s；硬盘型号为 INTEL SSDSC2KB480GB，容量为 5T，硬盘带宽为 SATA 3.2 - 6.0GB/s。RabbitMQ 消息中间件版本为 3.12.8，内存数据库为 Redis。该县级市配电网包含 87 个供电网格、14876 个节点。大数据块情况下的数据交互性能测试结果如图 4 所示。

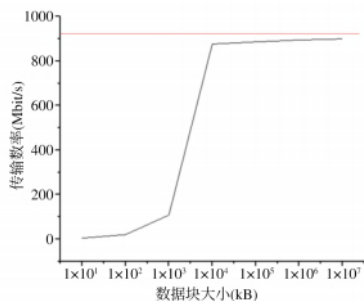


图4 大数据块通信速率测试

由图4可知，当规划平台进行大数据块通信时，随着数据块的增大其通信速率快速上升，当数据块大小超过 8.7×10^3 kB 时，通信速率接近局域网通信上限。

图5显示了大量小数据频繁交互时架构的通信性能，其中单个数据块大小为 1000kB。图5中通信时间随着交互次的增加而线性增加，说明在频繁交互时仍能保持较好的稳定性。

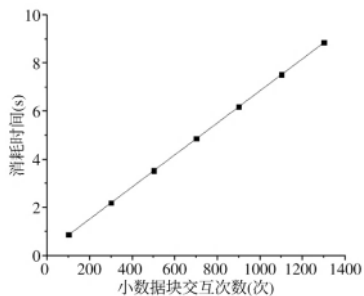


图5 小数据块频繁交互测试

(2) 功能模块工程实测

本文以碳排计算、潮流计算和源网荷储各要素时序仿真计算为例，测试复杂度不同的任务在分布式环境下的通信和计算速度效果。

由于全断面数据获取由配电网辅助规划平台触发，因此，测试时包括了请求消息通信时间和数据中台、电网资源业务中台准备数据时间。系统全断面数据初始化可在 3s 内完成，动态更新时间为毫秒级，可以满足规划业务的需求。

图6显示了不同复杂度算法和在分布式集群规模下的通信用时与计算用时分布图。

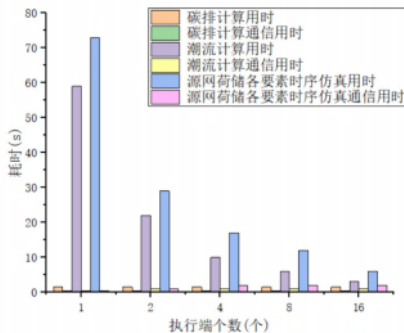


图6 不同算法不同规模的用时分布

随着分布式集群规模的不断增大，碳排计算功能模块总计算时间维持在 2s 以内。潮流计算、源网荷储各要素时序仿真计算的计算时间随着执行端规模的增大显著减少，由串行的 1.2h 降至 10s 以下，潮流计算的计算时间由串行的近 60s 降至 8s。

通过分析不同算法不同规模的用时分布可知，碳排计算算法简单，单个任务计算时间短，多执行端并行计算时加速效果不明显；全网潮流计算、源网荷储各要素时序仿真计算的算法较为复杂，串行计算时时间开销大。采用分布式并行计算后，可并行化计算项包括源、网、荷、储各部分的计算，不可并行化计算项包括全系统运行模拟、计算任务请求、子任务分发、结果回收、通知客户端的通信代价，以及任务分解、数据准备、结果组装、结果写库等操作。例如，全系统运行模拟将电源、电网、负荷和储能系统集成在一起仿真，模拟整个配电系统的运行状态，需要协调各子系统的计算结果。不可并行化计算耗时约 4s，核心算法并行化计算时间约 6s，总耗时约 10s。

当计算规模增大到原来的 10 倍以上，计算规模达到省级电网，通信耗时陡增，这时将千兆交换机换成万兆交换机即可缩短通信时间。因为对于路由器和交换机之类的网络设备，将千兆带宽提高到万兆以后，总背板带宽的提高不仅提升了每个端口的交换速度，也会带动整个网络速度的提高。

综合上述测试结果可知，使用较小的通信代价即可换取较大的计算时间缩减，加速后的计算效果满足

规划业务需求。

3 结束语

本文立足于工程实践提出了一种基于 RabbitMQ 消息中间件的分布式通信架构，并且设计了满足各模块间交互的通用消息格式，解决了配电网辅助规划平台异构数据获取问题和快速计算问题，满足了分布式计算平台的数据通信及模块间相互配合的需要，为配电网规划平台分布式计算提供了算力支持。

下一步将关注根据负载变化自适应地进行计算资源的灵活调度，例如开展计算负载错峰调度。更进一步地，将数据挖掘算法和资源调度结合起来，构建负载随着时间变化的模型，并基于此提前预测和调度，以减少负载急剧变化时资源分配不合理造成的延迟攀升，或者根据目标性能和当前集群资源，自动给出资源的最优调度方案。

参考文献

- [1] Qasaimeh M, Denolf K, Lo J, et al. Comparing energy efficiency of CPU, GPU and FPGA implementations for vision kernels[C]. 2019 IEEE International Conference on Embedded Software and Systems, 2019 : 1-8.
- [2] Yuan Y, Salmi M F, Huai Y, et al. Spark-GPU : An accelerated in-memory data processing engine on clusters[C]. 2016 IEEE International Conference on Big Dat, 2016.
- [3] Reuther, Albert, et al. Scheduler technologies in support of high performance data analysis[C]. 2016 IEEE High Performance Extreme Computing Conference, 2016.
- [4] Yu, Jia, Jinxuan Wu, Mohamed Sarwat. Geospark : A cluster computing framework for processing large-scale spatial data[C]. Proceedings of the 23rd SIGSPATIAL international conference on advances in geographic information systems, 2015.
- [5] 蔡乾乾, 孙勇, 阙华坤, 等. 计量自动化系统分布式计算和大数据分析技术研究 [J]. 工业控制计算机, 2023, 36 (10) : 140-142.
- [6] 唐元博. 分布式计算引擎架构设计及在城市数据分析中的应用 [D]. 南京: 东南大学, 2021.
- [7] 赵云翔, 面向任务的通用分布式计算系统的设计与实现 [D]. 石家庄: 河北科技大学, 2023.
- [8] 张生顺. 针对 Ray 框架的分布式计算调度仿真平台 [D]. 成都: 电子科技大学, 2023.
- [9] 赵云翔. 面向任务的通用分布式计算系统的设计与实现 [D]. 石家庄: 河北科技大学, 2023.
- [10] 赖丹晖, 罗伟峰. 基于分布式计算的电网运行稳定性评估方法 [J]. 信息与电脑 (理论版), 2022, 34 (21) : 30-32, 36.
- [11] 潘莹丽, 刘飞, 刘展, 等. 基于大规模数据尾期望回归的分布式计算方法 [J]. 统计与决策, 2022, 38 (12) : 11-16.
- [12] 周玲玲. 基于节点分组方法的编码分布式计算方案 [J]. 桂林: 广西师范大学, 2022.
- [13] 颜帅. 基于 LBS 数据的出行目的识别和轨迹预测研究 [D]. 南京: 东南大学, 2020.
- [14] 焦润海, 王景兵, 林碧英. 配电网高级应用分布式计算平台通信架构设计 [J]. 电力系统自动化, 2013, 37 (22) 66-72.
- [15] 王军亮, 开源平台下网上阅卷系统的分布式计算架构设计 [D]. 贵阳: 贵州大学, 2011.

(收稿日期: 2023-12-25)