

一种请求调度方法和设备

本申请要求于 2023 年 10 月 24 日提交国家知识产权局、申请号为 202311386911.8、申请名称为“一种请求调度方法和设备”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

技术领域

本发明涉及计算机技术领域，特别是一种请求调度方法和设备。

背景技术

大规模语言模型 (Large Language Model, LLM) 在人工智能 (Artificial Intelligence, AI) 界掀起了新的浪潮。使用键值缓存 (KV Cache) 技术的生成式预训练转换器 (Generative Pre-trained Transformer, GPT) 的推理过程包括两个阶段，即首轮上下文填充 (Prefill) 阶段和多轮的解码 (Decode) 阶段。大规模语言模型服务部署中，服务系统虽然采用了动态的调度方式，但是 LLM 服务部署中的服务系统的吞吐量仍旧较低。

发明内容

有鉴于此，本发明实施例提供了一种请求调度方法和设备，能够提升 LLM 服务部署中的服务系统的吞吐量。

第一方面，本发明实施例提供了一种请求调度方法，应用于请求调度系统，所述方法包括：

第一设备接收来自第一用户的第一请求，所述第一请求为第一服务的请求，所述第一请求包括所述第一服务的标识；

所述第一设备将所述第一请求调度至第二设备；

所述第二设备向所述第一设备发送第二请求，所述第二请求包括所述第一服务的标识以及所述第一服务对应的第一输出信息，所述第二请求与所述第一请求相关联；

当所述第一输出信息不满足设定的条件时，所述第一设备将所述第二请求调度至第三设备，所述第三设备与所述第二设备不同；

所述第三设备向所述第一设备发送第三请求，所述第三请求包括所述第一服务的标识以及所述第一服务对应的第二输出信息，所述第三请求与所述第二请求相关联；

当所述第二输出信息不满足所述设定的条件时，所述第一设备将所述第三请求调度至所述第三设备。

本发明实施例同时在请求调度系统中，采用独立的分离部署方式，部署第二设备和第三设备，通过第一设备对请求的调度，使所述第二设备和所述第三设备分别执行各自阶段的推理过程，同时，两个设备之间可以进行信息传递，使得推理过程可以正常进行，从而能够提升服务系统的吞吐量。

结合第一方面，在第一方面的某些实现方式中，所述方法还包括：

所述第一设备接收来自第二用户的第四请求，所述第四请求为第二服务的请求，所述第四请求包括所述第二服务的标识；

所述第一设备将所述第四请求调度至所述第二设备；

所述第二设备向所述第一设备发送第五请求，所述第五请求包括所述第二服务的标识以及所述第二服务对应的第三输出信息，所述第五请求与所述第四请求相关联；

当所述第三输出信息不满足所述设定的条件时，所述第一设备将所述第五请求调度至第三设备；

所述第三设备向所述第一设备发送第六请求，所述第六请求包括所述第二服务的标识以及所述第二服务对应的第四输出信息，所述第六请求与所述第五请求相关联；

当所述第四输出信息不满足所述设定的条件时，所述第一设备将所述第六请求调度至所述第三设备。

本发明实施例中，第二用户和第一用户不同，第一设备能够把来自不同用户的请求调度到所述第二设备和所述第三设备进行处理，不需要等所述第二设备和/或所述第三设备处理完一个用户的请求再处理另一个用户的请求，从而能够提升服务系统的吞吐量。

结合第一方面，在第一方面的某些实现方式中，所述方法还包括：

当所述第二输出信息满足所述设定的条件时，所述第一设备向所述第一用户发送所述第二输出信息。

本发明实施例中，通过第一设备判断服务的输出信息是否满足设定的条件，若不满足设定的条件，继续对服务进行调度和推理计算；若满足设定的条件，对服务进行早停，并向第一用户发送第二输出信息。第一设备对服务进行早停后，可以通知第三设备释放该服务对应的资源；第三设备向第一设备上报第三设备的当前资源状态，若当前资源状态存在空闲的计算服务资源，第一设备向第三设备继续调度服务以插入空闲的计算服务资源，从而实现动态插入。

结合第一方面，在第一方面的某些实现方式中，所述请求调度系统包括所述第一设备、至少一个所述第二设备和至少一个所述第三设备。本发明实施例中，请求调度系统中第二设备和第三设备的数量较多时，采用独立的分离部署方式，部署第二设备和第三设备，第二设备配置小批次或单批次，第三设备配置大批次。第二设备资源和第三设备资源可以通过系统负载动态调整资源比例，以获得最大的吞吐量，从而大幅度提高吞吐量的目的。

第二方面，本发明实施例提供了一种请求调度方法，应用于第一设备，所述方法包括：

接收来自第一用户的第一请求，所述第一请求为第一服务的请求，所述第一请求包括所述第一服务的标识；

将所述第一请求调度至第二设备；

接收所述第二设备发送的第二请求，所述第二请求包括所述第一服务的标识以及所述第一服务对应的第一输出信息，所述第二请求与所述第一请求相关联；

当所述第一输出信息不满足设定的条件时，将所述第二请求调度至第三设备，所述第三设备与所述第二设备不同；

接收所述第三设备发送的第三请求，所述第三请求包括所述第一服务的标识以及所述第一服务对应的第二输出信息，所述第三请求与所述第二请求相关联；

当所述第二输出信息不满足所述设定的条件时，将所述第三请求调度至所述第三设备。

本发明实施例将来自用户的新服务的请求调度至第二设备，即 Prefill 推理执行容器进行 Prefill 阶段的推理计算；将经过 Prefill 阶段的旧服务的请求调度至第三设备，即 Decode 推理执行容器进行 Decode 阶段的推理计算。本发明通过分离部署的方式，综合考虑了 Prefill 阶段算力瓶颈和 Decode 阶段访存瓶颈的显著差异，能够提升服务系统的吞吐量。

结合第二方面，在第二方面的某些实现方式中，所述方法还包括：

接收来自第二用户的第四请求，所述第四请求为第二服务的请求，所述第四请求包括所述第二服务的标识；

将所述第四请求调度至所述第二设备；

接收所述第二设备发送的第五请求，所述第五请求包括所述第二服务的标识以及所述第二服务对应的第三输出信息，所述第五请求与所述第四请求相关联；

当所述第三输出信息不满足所述设定的条件时，将所述第五请求调度至第三设备；

接收所述第三设备发送的第六请求，所述第六请求包括所述第二服务的标识以及所述第二服务对应的第四输出信息，所述第六请求与所述第五请求相关联；

当所述第四输出信息不满足所述设定的条件时，将所述第六请求调度至所述第三设备。

本发明实施例中，第二用户和第一用户不同，第一设备能够把来自不同用户的请求调度到第二设备和第三设备进行处理，不需要等第二设备和/或第三设备处理完一个用户的请求再处理另一个用户的请求，从而能够提升服务系统的吞吐量。

结合第二方面，在第二方面的某些实现方式中，所述方法还包括：

当所述第二输出信息满足所述设定的条件时，向所述第一用户发送所述第二输出信息。

本发明实施例中，通过第一设备判断服务的输出信息是否满足设定的条件，若不满足设定的条件，继续对服务进行调度和推理计算；若满足设定的条件，对服务进行早停，并向第一用户发送第二输出信息。第一设备对服务进行早停后，可以通知第三设备释放该服务对应的资源；第三设备向第一设备上报第三设备的当前资源状态，若当前资源状态存在空闲的计算服务资源，第一设备向第三设备继续调度服务以插入空闲的计算服务资源，从而实现动态插入。

结合第二方面，在第二方面的某些实现方式中，所述标识包括服务 ID。

结合第二方面，在第二方面的某些实现方式中，所述设定的条件包括以下至少一个：

输出信息的长度大于或者等于第一阈值，或

所述输出信息包括终止符。

第三方面，本发明实施例提供了一种请求调度方法，应用于第二设备，所述方法包括：

接收第一设备发送的来自第一用户的第一请求，所述第一请求为第一服务的请求，所述第一请求包括所述第一服务的标识；

向所述第一设备发送第二请求，所述第二请求包括所述第一服务的标识以及所述第一服务对应的第一输出信息，所述第二请求与所述第一请求相关联。

本发明实施例将来自用户的新服务的请求调度至第二设备，即 Prefill 推理执行容器进行 Prefill 阶段的推理计算；将经过 Prefill 阶段的旧服务的请求调度至第三设备，即 Decode 推理执行容器进行 Decode 阶段的推理计算。本发明通过分离部署的方式，综合考虑了 Prefill 阶段算力瓶颈和 Decode 阶段访存瓶颈的显著差异，从而提升服务系统的吞吐量。

结合第三方面，在第三方面的某些实现方式中，所述方法还包括：

接收所述第一设备发送的来自第二用户的第四请求，所述第四请求为第二服务的请求，所述第四请求包括所述第二服务的标识；

向所述第一设备发送第五请求，所述第五请求包括所述第二服务的标识以及所述第二服务对应的第三输出信息，所述第五请求与所述第四请求相关联。

本发明实施例中，第二用户和第一用户不同，第一设备能够把来自不同用户的请求调度到第二设备和第三设备进行处理，不需要等第二设备和/或第三设备处理完一个用户的请求再处理另一个用户的请求，从而能够提升服务系统的吞吐量。

结合第三方面，在第三方面的某些实现方式中，所述向所述第一设备发送第二请求之前，还包括：

根据第一计算资源状态和第一条件对所述第一请求和与所述第一请求同时到达所述第二设备的至少一个第七请求进行组批次得到第一批次，所述第七请求为来自第一用户或第二用户或其他用户的请求；

对所述第一批次进行第一推理计算，得到所述第二请求和第一缓存信息。

结合第三方面，在第三方面的某些实现方式中，所述方法还包括：

将所述第一缓存信息发送至所述第三设备。

结合第三方面，在第三方面的某些实现方式中，所述缓存信息包括键值缓存 KV 信息。

结合第三方面，在第三方面的某些实现方式中，所述根据第一计算资源状态和第一条件对所述第一请求和与所述第一请求同时到达所述第二设备的至少一个第七请求进行组批次得到第一批次，包括：

根据所述第一请求，得到与所述第一请求对应的输入信息及其第一长度；

根据所述第一计算资源状态、所述第一条件和所述第一长度，对所述第一请求和所述至少一个第七请求进行组批次得到第一批次。其中，所述第七请求对应的输入信息的长度和第一长度相似，有利于在第一推理计算过程中减少空闲资源的浪费。

结合第三方面，在第三方面的某些实现方式中，所述第一计算资源状态包括所述第二设备中的空闲资源的数量。

结合第三方面，在第三方面的某些实现方式中，所述第一条件包括所述第二设备最多能组的批次的数量。

结合第三方面，在第三方面的某些实现方式中，所述第一处理方式当前批次的所有请求均推理计算结束后，释放当前批次所占用的计算服务资源并将下一批次插入空闲计算服务资源进行第一推理计算。

本发明实施例中，对于 Prefill 阶段的推理计算，按照单批次进行推理计算，即当前批次的所有请求均推理计算结束后，再释放当前批次所占用的计算服务资源并将下一批次插入空闲计算服务资源，开始对下一批次进行第一推理计算。

第四方面，本发明实施例提供了一种请求调度方法，应用于第三设备，所述方法包括：

接收第一设备发送的来自第二设备的第二请求，所述第二请求包括第一服务的标识以及所述第一服务对应的第一输出信息，所述第二请求与所述第一请求相关联，所述第一请求为来自第一用户的所述第一服务的请求；

向所述第一设备发送第三请求，所述第三请求包括所述第一服务的标识以及所述第一服务对应的第二输出信息，所述第三请求与所述第二请求相关联。

本发明实施例将来自用户的新服务的请求调度至第二设备，即 Prefill 推理执行容器进行 Prefill 阶段的推理计算；将经过 Prefill 阶段的旧服务的请求调度至第三设备，即 Decode 推理执行容器进行 Decode 阶段的推理计算。本发明通过分离部署的方式，综合考虑了 Prefill 阶段算力瓶颈和 Decode 阶段访存瓶颈的显著差异，从而提升服务系统的吞吐量。

结合第四方面，在第四方面的某些实现方式中，所述方法还包括：

接收所述第一设备发送的第五请求，所述第五请求包括第二服务的标识以及所述第二服务对应的第三输出信息，所述第五请求与第四请求相关联，所述第四请求为来自第二用户的所述第二服务的请求；

向所述第一设备发送第六请求，所述第六请求包括所述第二服务的标识以及所述第二服务对应的第四输出信息，所述第六请求与所述第五请求相关联。

本发明实施例中，第二用户和第一用户不同，第一设备能够把来自不同用户的请求调度到第二设备和第三设备进行处理，不需要等第二设备和/或第三设备处理完一个用户的请求再处理另一个用户的请求，从而能够提升服务系统的吞吐量。

结合第四方面，在第四方面的某些实现方式中，所述向所述第一设备发送第三请求之前，还包括：

根据第二计算资源状态和第二条件对所述第二请求和与所述第二请求同时到达所述第三设备的至少一个第八请求进行组批次得到第二批次，所述第八请求来自第二设备或第三设备；

接收所述第二设备发送的第一缓存信息；

根据所述第一缓存信息，对所述第二批次进行第二推理计算，得到所述第三请求和第二缓存信息。

第五方面，本发明实施例提供了一种请求调度系统，所述系统包括：

第一设备，用于接收来自第一用户的第一请求，所述第一请求为第一服务的请求，所述第一请求包括所述第一服务的标识；

所述第一设备还用于将所述第一请求调度至第二设备；

所述第二设备，用于向所述第一设备发送第二请求，所述第二请求包括所述第一服务的标识以及所述第一服务对应的第一输出信息，所述第二请求与所述第一请求相关联；

当所述第一输出信息不满足设定的条件时，所述第一设备还用于将所述第二请求调度至第三设备，所述第三设备与所述第二设备不同；

所述第三设备，用于向所述第一设备发送第三请求，所述第三请求包括所述第一服务的标识以及所述第一服务对应的第二输出信息，所述第三请求与所述第二请求相关联；

当所述第二输出信息不满足设定的条件时，所述第一设备还用于将所述第三请求调度至所述第三设备。

第六方面，本发明实施例提供了一种设备，包括处理器和存储器，其中，所述存储器用于存储程序，所述程序包括程序指令，当所述处理器运行所述程序指令时，使所述设备执行如上述所述的方法的步骤。

第七方面，本发明实施例提供了一种可读存储介质，所述可读存储介质存储有程序，所述程序包括程序指令，当所述程序请求被设备运行时使所述设备执行如上述所述的方法。

第八方面，本发明实施例提供了一种程序产品，所述程序产品包含指令，当所述程序产品在设备或任一至少一种处理器上运行时，使得所述设备执行如上述方法中的功能/步骤。

本发明实施例提供的请求调度方法和设备的技术方案中，第一设备接收第一用户的第一请求，第一请求为第一服务的请求，第一请求包括第一服务的标识；第一设备将第一请求调度至第二设备；第二设备向第一设备发送第二请求，第二请求包括第一服务的标识和第一输出信息，第二请求与第一请求相关联；当第一输出信息不满足设定的条件时，第一设备将第二请求调度至第三设备，第三设备与第二设备不同；第三设备向第一设备发送第三请求，第三请求包括第一服务的标识和第二输出信息，第三请求与第二请求相关联；当第二输出信息不满足设定的条件时，第一设备将第三请求调度至第三设备，能够提升大规模语言模型服务部署中的服务系统的吞吐量。

附图说明

图1为本发明实施例提供的一种设备的结构示意图；

图2为本发明实施例的设备100的软件结构框图；

图3为本发明实施例提供的又一种请求调度系统的架构图；

图4为本发明实施例提供的一种请求调度方法的信令交互图；

图5为本发明实施例中Prefill推理和Decode推理的示意图；
图6为本发明实施例提供的又一种请求调度方法的信令交互图；
图7为本发明实施例提供的一种请求调度方法的流程图；
图8为本发明实施例提供的又一种请求调度方法的流程图；
图9为本发明实施例提供的一种第一设备的结构示意图；
图10为本发明实施例提供的一种第二设备的结构示意图；
图11为本发明实施例提供的一种第三设备的结构示意图。

具体实施方式

为了更好的理解本发明的技术方案，下面结合附图对本发明实施例进行详细描述。

应当明确，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它实施例，都属于本发明保护的范围。

在本发明实施例中使用的术语是仅仅出于描述特定实施例的目的，而非旨在限制本发明。在本发明实施例和所附权利要求书中所使用的单数形式的“一种”、“所述”和“该”也旨在包括多数形式，除非上下文清楚地表示其他含义。

应当理解，本文中使用的术语“和/或”仅仅是一种描述关联对象的关联关系，表示可以存在三种关系，例如，甲和/或乙，可以表示：单独存在甲，同时存在甲和乙，单独存在乙这三种情况。另外，本文中字符“/”，一般表示前后关联对象是一种“或”的关系。

图1示出了设备100的结构示意图。

设备100可以包括处理器110，外部存储器接口120，内部存储器121，通用串行总线(universal serial bus, USB)接口130，充电管理模块140，电源管理模块141，电池142，天线1，天线2，移动通信模块150，无线通信模块160，音频模块170，扬声器170A，受话器170B，麦克风170C，耳机接口170D，传感器模块180，按键190，马达191，指示器192，摄像头193，显示屏194，以及用户标识模块(subscriber identification module, SIM)卡接口195等。其中传感器模块180可以包括压力传感器180A，陀螺仪传感器180B，气压传感器180C，磁传感器180D，加速度传感器180E，距离传感器180F，接近光传感器180G，指纹传感器180H，温度传感器180J，触摸传感器180K，环境光传感器180L，骨传导传感器180M等。

可以理解的是，本发明实施例示意的结构并不构成对设备100的具体限定。在本申请另一些实施例中，设备100可以包括比图示更多或更少的部件，或者组合某些部件，或者拆分某些部件，或者不同的部件布置。图示的部件可以以硬件，软件或软件和硬件的组合实现。

处理器110可以包括一个或多个处理单元，例如：处理器110可以包括应用处理器(application processor, AP)，调制解调处理器，图形处理器(graphics processing unit, GPU)，图像信号处理器(image signal processor, ISP)，控制器，视频编解码器，数字信号处理器(digital signal processor, DSP)，基带处理器，和/或神经网络处理器(neural-network processing unit, NPU)等。其中，不同的处理单元可以是独立的器件，也可以集成在一个或多个处理器中。

控制器可以根据指令操作码和时序信号，产生操作控制信号，完成取指令和执行指令的控制。

处理器110中还可以设置存储器，用于存储指令和数据。在一些实施例中，处理器110中的存储器为高速缓冲存储器。该存储器可以保存处理器110刚用过或循环使用的指令或数据。如果处理器110需要再次使用该指令或数据，可从所述存储器中直接调用。避免了重复存取，减少了处理器110的等待时间，因而提高了系统的效率。

在一些实施例中，处理器110可以包括一个或多个接口。接口可以包括集成电路(inter-integrated circuit, I2C)接口，集成电路内置音频(inter-integrated circuit sound, I2S)接口，脉冲编码调制(pulse code modulation, PCM)接口，通用异步收发传输器(universal asynchronous receiver/transmitter, UART)接口，移动产业处理器接口(mobile industry processor interface, MIPI)，通用输入输出(general-purpose input/output, GPIO)接口，用户标识模块(subscriber identity module, SIM)接口，和/或通用串行总线(universal serial bus, USB)接口等。

I2C接口是一种双向同步串行总线，包括一根串行数据线(serial data line, SDA)和一根串行时钟

线(derail clock line, SCL)。在一些实施例中,处理器 110 可以包含多组 I2C 总线。处理器 110 可以通过不同的 I2C 总线接口分别耦合触摸传感器 180K,充电器,闪光灯,摄像头 193 等。例如:处理器 110 可以通过 I2C 接口耦合触摸传感器 180K,使处理器 110 与触摸传感器 180K 通过 I2C 总线接口通信,实现设备 100 的触摸功能。

I2S 接口可以用于音频通信。在一些实施例中,处理器 110 可以包含多组 I2S 总线。处理器 110 可以通过 I2S 总线与音频模块 170 耦合,实现处理器 110 与音频模块 170 之间的通信。在一些实施例中,音频模块 170 可以通过 I2S 接口向无线通信模块 160 传递音频信号,实现通过蓝牙耳机接听电话的功能。

PCM 接口也可以用于音频通信,将模拟信号抽样,量化和编码。在一些实施例中,音频模块 170 与无线通信模块 160 可以通过 PCM 总线接口耦合。在一些实施例中,音频模块 170 也可以通过 PCM 接口向无线通信模块 160 传递音频信号,实现通过蓝牙耳机接听电话的功能。所述 I2S 接口和所述 PCM 接口都可以用于音频通信。

UART 接口是一种通用串行数据总线,用于异步通信。该总线可以为双向通信总线。它将要传输的数据在串行通信与并行通信之间转换。在一些实施例中,UART 接口通常被用于连接处理器 110 与无线通信模块 160。例如:处理器 110 通过 UART 接口与无线通信模块 160 中的蓝牙模块通信,实现蓝牙功能。在一些实施例中,音频模块 170 可以通过 UART 接口向无线通信模块 160 传递音频信号,实现通过蓝牙耳机播放音乐的功能。

MIPI 接口可以被用于连接处理器 110 与显示屏 194,摄像头 193 等外围器件。MIPI 接口包括摄像头串行接口(camera serial interface, CSI),显示屏串行接口(display serial interface, DSI)等。在一些实施例中,处理器 110 和摄像头 193 通过 CSI 接口通信,实现设备 100 的拍摄功能。处理器 110 和显示屏 194 通过 DSI 接口通信,实现设备 100 的显示功能。

GPIO 接口可以通过软件配置。GPIO 接口可以被配置为控制信号,也可被配置为数据信号。在一些实施例中,GPIO 接口可以用于连接处理器 110 与摄像头 193,显示屏 194,无线通信模块 160,音频模块 170,传感器模块 180 等。GPIO 接口还可以被配置为 I2C 接口,I2S 接口,UART 接口,MIPI 接口等。

USB 接口 130 是符合 USB 标准规范的接口,具体可以是 Mini USB 接口, Micro USB 接口, USB Type C 接口等。USB 接口 130 可以用于连接充电器为设备 100 充电,也可以用于设备 100 与外围设备之间传输数据。也可以用于连接耳机,通过耳机播放音频。该接口还可以用于连接其他设备,例如 AR 设备等。

可以理解的是,本发明实施例示意的各模块间的接口连接关系,只是示意性说明,并不构成对设备 100 的结构限定。在本申请另一些实施例中,设备 100 也可以采用上述实施例中不同的接口连接方式,或多种接口连接方式的组合。

充电管理模块 140 用于从充电器接收充电输入。其中,充电器可以是无线充电器,也可以是有线充电器。在一些有线充电的实施例中,充电管理模块 140 可以通过 USB 接口 130 接收有线充电器的充电输入。在一些无线充电的实施例中,充电管理模块 140 可以通过设备 100 的无线充电线圈接收无线充电输入。充电管理模块 140 为电池 142 充电的同时,还可以通过电源管理模块 141 为设备供电。

电源管理模块 141 用于连接电池 142,充电管理模块 140 与处理器 110。电源管理模块 141 接收电池 142 和/或充电管理模块 140 的输入,为处理器 110,内部存储器 121,显示屏 194,摄像头 193,和无线通信模块 160 等供电。电源管理模块 141 还可以用于监测电池容量,电池循环次数,电池健康状态(漏电,阻抗)等参数。在其他一些实施例中,电源管理模块 141 也可以设置于处理器 110 中。在另一些实施例中,电源管理模块 141 和充电管理模块 140 也可以设置于同一个器件中。

设备 100 的无线通信功能可以通过天线 1,天线 2,移动通信模块 150,无线通信模块 160,调制解调处理器以及基带处理器等实现。

天线 1 和天线 2 用于发射和接收电磁波信号。设备 100 中的每个天线可用于覆盖单个或多个通信频段。不同的天线还可以复用,以提高天线的利用率。例如:可以将天线 1 复用为无线局域网的分集天线。在另外一些实施例中,天线可以和调谐开关结合使用。

移动通信模块 150 可以提供应用在设备 100 上的包括 2G/3G/4G/5G 等无线通信的解决方案。移动通信模块 150 可以包括至少一个滤波器,开关,功率放大器,低噪声放大器(low noise amplifier, LNA)等。移动通信模块 150 可以由天线 1 接收电磁波,并对接收的电磁波进行滤波,放大等处理,传送至调制解调处理器进行解调。移动通信模块 150 还可以对经调制解调处理器调制后的信号放大,经天线 1 转为电磁波辐射出去。在一些实施例中,移动通信模块 150 的至少部分功能模块可以被设置于处理器 110

中。在一些实施例中，移动通信模块 150 的至少部分功能模块可以与处理器 110 的至少部分模块被设置在同一个器件中。

调制解调处理器可以包括调制器和解调器。其中，调制器用于将待发送的低频基带信号调制成中高频信号。解调器用于将接收的电磁波信号解调为低频基带信号。随后解调器将解调得到的低频基带信号传送至基带处理器处理。低频基带信号经基带处理器处理后，被传递给应用处理器。应用处理器通过音频设备(不限于扬声器 170A，受话器 170B 等)输出声音信号，或通过显示屏 194 显示图像或视频。在一些实施例中，调制解调处理器可以是独立的器件。在另一些实施例中，调制解调处理器可以独立于处理器 110，与移动通信模块 150 或其他功能模块设置在同一个器件中。

无线通信模块 160 可以提供应用在设备 100 上的包括无线局域网(wireless local area networks, WLAN)(如无线保真(wireless fidelity, Wi-Fi)网络)，蓝牙(bluetooth, BT)，全球导航卫星系统(global navigation satellite system, GNSS)，调频(frequency modulation, FM)，近距离无线通信技术(near field communication, NFC)，红外技术(infrared, IR)等无线通信的解决方案。无线通信模块 160 可以是集成至少一个通信处理模块的一个或多个器件。无线通信模块 160 经由天线 2 接收电磁波，将电磁波信号调频以及滤波处理，将处理后的信号发送到处理器 110。无线通信模块 160 还可以从处理器 110 接收待发送的信号，对其进行调频，放大，经天线 2 转为电磁波辐射出去。

在一些实施例中，设备 100 的天线 1 和移动通信模块 150 耦合，天线 2 和无线通信模块 160 耦合，使得设备 100 可以通过无线通信技术与网络以及其他设备通信。所述无线通信技术可以包括全球移动通讯系统(global system for mobile communications, GSM)，通用分组无线服务(general packet radio service, GPRS)，码分多址接入(code division multiple access, CDMA)，宽带码分多址(wideband code division multiple access, WCDMA)，时分码分多址(time-division code division multiple access, TD-SCDMA)，长期演进(long term evolution, LTE)，BT, GNSS, WLAN, NFC, FM, 和/或 IR 技术等。所述 GNSS 可以包括全球卫星定位系统(global positioning system, GPS)，全球导航卫星系统(global navigation satellite system, GLONASS)，北斗卫星导航系统(beidou navigation satellite system, BDS)，准天顶卫星系统(quasi-zenith satellite system, QZSS)和/或星基增强系统(satellite based augmentation systems, SBAS)。

设备 100 通过 GPU，显示屏 194，以及应用处理器等实现显示功能。GPU 为图像处理的微处理器，连接显示屏 194 和应用处理器。GPU 用于执行数学和几何计算，用于图形渲染。处理器 110 可包括一个或多个 GPU，其执行程序指令以生成或改变显示信息。

显示屏 194 用于显示图像，视频等。显示屏 194 包括显示面板。显示面板可以采用液晶显示屏(liquid crystal display, LCD)，有机发光二极管(organic light-emitting diode, OLED)，有源矩阵有机发光二极管或主动矩阵有机发光二极管(active-matrix organic light emitting diode 的, AMOLED)，柔性发光二极管(flex light-emitting diode, FLED)，Miniled, MicroLed, Micro-oLed, 量子点发光二极管(quantum dot light emitting diodes, QLED)等。在一些实施例中，设备 100 可以包括 1 个或 N 个显示屏 194，N 为大于 1 的正整数。

设备 100 可以通过 ISP，摄像头 193，视频编解码器，GPU，显示屏 194 以及应用处理器等实现拍摄功能。

ISP 用于处理摄像头 193 反馈的数据。例如，拍照时，打开快门，光线通过镜头被传递到摄像头感光元件上，光信号转换为电信号，摄像头感光元件将所述电信号传递给 ISP 处理，转化为肉眼可见的图像。ISP 还可以对图像的噪点，亮度，肤色进行算法优化。ISP 还可以对拍摄场景的曝光，色温等参数优化。在一些实施例中，ISP 可以设置在摄像头 193 中。

摄像头 193 用于捕获静态图像或视频。物体通过镜头生成光学图像投射到感光元件。感光元件可以是电荷耦合器件(charge coupled device, CCD)或互补金属氧化物半导体(complementary metal-oxide-semiconductor, CMOS)光电晶体管。感光元件把光信号转换为电信号，之后将电信号传递给 ISP 转换成数字图像信号。ISP 将数字图像信号输出到 DSP 加工处理。DSP 将数字图像信号转换成标准的 RGB, YUV 等格式的图像信号。在一些实施例中，设备 100 可以包括 1 个或 N 个摄像头 193，N 为大于 1 的正整数。

数字信号处理器用于处理数字信号，除了可以处理数字图像信号，还可以处理其他数字信号。例如，当设备 100 在频点选择时，数字信号处理器用于对频点能量进行傅里叶变换等。

视频编解码器用于对数字视频压缩或解压缩。设备 100 可以支持一种或多种视频编解码器。这样，

设备 100 可以播放或录制多种编码格式的视频,例如:动态图像专家组(moving picture experts group, MPEG)1, MPEG2, MPEG3, MPEG4 等。

NPU 为神经网络(neural-network, NN)计算处理器,通过借鉴生物神经网络结构,例如借鉴人脑神经元之间传递模式,对输入信息快速处理,还可以不断的自学习。通过 NPU 可以实现设备 100 的智能认知等应用,例如:图像识别,人脸识别,语音识别,文本理解等。

外部存储器接口 120 可以用于连接外部存储卡,例如 Micro SD 卡,实现扩展设备 100 的存储能力。外部存储卡通过外部存储器接口 120 与处理器 110 通信,实现数据存储功能。例如将音乐,视频等文件保存在外部存储卡中。

内部存储器 121 可以用于存储计算机可执行程序代码,所述可执行程序代码包括指令。内部存储器 121 可以包括存储程序区和存储数据区。其中,存储程序区可存储操作系统,至少一个功能所需的应用程序(比如声音播放功能,图像播放功能等)等。存储数据区可存储设备 100 使用过程中所创建的数据(比如音频数据,电话本等)等。此外,内部存储器 121 可以包括高速随机存取存储器,还可以包括非易失性存储器,例如至少一个磁盘存储器件,闪存器件,通用闪存存储器(universal flash storage, UFS)等。处理器 110 通过运行存储在内部存储器 121 的指令,和/或存储在设置于处理器中的存储器的指令,执行设备 100 的各种功能应用以及数据处理。

设备 100 可以通过音频模块 170,扬声器 170A,受话器 170B,麦克风 170C,耳机接口 170D,以及应用处理器等实现音频功能。例如音乐播放,录音等。

音频模块 170 用于将数字音频信息转换成模拟音频信号输出,也用于将模拟音频输入转换为数字音频信号。音频模块 170 还可以用于对音频信号编码和解码。在一些实施例中,音频模块 170 可以设置于处理器 110 中,或将音频模块 170 的部分功能模块设置于处理器 110 中。

扬声器 170A,也称“喇叭”,用于将音频电信号转换为声音信号。设备 100 可以通过扬声器 170A 收听音乐,或收听免提通话。

受话器 170B,也称“听筒”,用于将音频电信号转换成声音信号。当设备 100 接听电话或语音信息时,可以通过将受话器 170B 靠近人耳接听语音。

麦克风 170C,也称“话筒”,“传声器”,用于将声音信号转换为电信号。当拨打电话或发送语音信息时,用户可以通过人嘴靠近麦克风 170C 发声,将声音信号输入到麦克风 170C。设备 100 可以设置至少一个麦克风 170C。在另一些实施例中,设备 100 可以设置两个麦克风 170C,除了采集声音信号,还可以实现降噪功能。在另一些实施例中,设备 100 还可以设置三个,四个或更多麦克风 170C,实现采集声音信号,降噪,还可以识别声音来源,实现定向录音功能等。

耳机接口 170D 用于连接有线耳机。耳机接口 170D 可以是 USB 接口 130,也可以是 3.5mm 的开放移动电子设备平台(open mobile terminal platform, OMTp)标准接口,美国蜂窝电信工业协会(cellular telecommunications industry association of the USA, CTIA)标准接口。

压力传感器 180A 用于感受压力信号,可以将压力信号转换成电信号。在一些实施例中,压力传感器 180A 可以设置于显示屏 194。压力传感器 180A 的种类很多,如电阻式压力传感器,电感式压力传感器,电容式压力传感器等。电容式压力传感器可以是包括至少两个具有导电材料的平行板。当有力作用于压力传感器 180A,电极之间的电容改变。设备 100 根据电容的变化确定压力的强度。当有触摸操作作用于显示屏 194,设备 100 根据压力传感器 180A 检测所述触摸操作强度。设备 100 也可以根据压力传感器 180A 的检测信号计算触摸的位置。在一些实施例中,作用于相同触摸位置,但不同触摸操作强度的触摸操作,可以对应不同的操作指令。例如:当有触摸操作强度小于第一压力阈值的触摸操作作用于短消息应用图标时,执行查看短消息的指令。当有触摸操作强度大于或等于第一压力阈值的触摸操作作用于短消息应用图标时,执行新建短消息的指令。

陀螺仪传感器 180B 可以用于确定设备 100 的运动姿态。在一些实施例中,可以通过陀螺仪传感器 180B 确定设备 100 围绕三个轴(即,x,y 和 z 轴)的角速度。陀螺仪传感器 180B 可以用于拍摄防抖。示例性的,当按下快门,陀螺仪传感器 180B 检测设备 100 抖动的角度,根据角度计算出镜头模组需要补偿的距离,让镜头通过反向运动抵消设备 100 的抖动,实现防抖。陀螺仪传感器 180B 还可以用于导航,体感游戏场景。

气压传感器 180C 用于测量气压。在一些实施例中,设备 100 通过气压传感器 180C 测得的气压值计算海拔高度,辅助定位和导航。

磁传感器 180D 包括霍尔传感器。设备 100 可以利用磁传感器 180D 检测翻盖皮套的开合。在一些实施例中,当设备 100 是翻盖机时,设备 100 可以根据磁传感器 180D 检测翻盖的开合。进而根据检测到的皮套的开合状态或翻盖的开合状态,设置翻盖自动解锁等特性。

加速度传感器 180E 可检测设备 100 在各个方向上(一般为三轴)加速度的大小。当设备 100 静止时可检测出重力的大小及方向。还可以用于识别设备姿态,应用于横竖屏切换,计步器等应用。

距离传感器 180F,用于测量距离。设备 100 可以通过红外或激光测量距离。在一些实施例中,拍摄场景,设备 100 可以利用距离传感器 180F 测距以实现快速对焦。

接近光传感器 180G 可以包括例如发光二极管(LED)和光检测器,例如光电二极管。发光二极管可以是红外发光二极管。设备 100 通过发光二极管向外发射红外光。设备 100 使用光电二极管检测来自附近物体的红外反射光。当检测到充分的反射光时,可以确定设备 100 附近有物体。当检测到不充分的反射光时,设备 100 可以确定设备 100 附近没有物体。设备 100 可以利用接近光传感器 180G 检测用户手持设备 100 贴近耳朵通话,以便自动熄灭屏幕达到省电的目的。接近光传感器 180G 也可用于皮套模式,口袋模式自动解锁与锁屏。

环境光传感器 180L 用于感知环境光亮度。设备 100 可以根据感知的环境光亮度自适应调节显示屏 194 亮度。环境光传感器 180L 也可用于拍照时自动调节白平衡。环境光传感器 180L 还可以与接近光传感器 180G 配合,检测设备 100 是否在口袋里,以防误触。

指纹传感器 180H 用于采集指纹。设备 100 可以利用采集的指纹特性实现指纹解锁,访问应用锁,指纹拍照,指纹接听来电等。

温度传感器 180J 用于检测温度。在一些实施例中,设备 100 利用温度传感器 180J 检测的温度,执行温度处理策略。例如,当温度传感器 180J 上报的温度超过阈值,设备 100 执行降低位于温度传感器 180J 附近的处理器的性能,以便降低功耗实施热保护。在另一些实施例中,当温度低于另一阈值时,设备 100 对电池 142 加热,以避免低温导致设备 100 异常关机。在其他一些实施例中,当温度低于又一阈值时,设备 100 对电池 142 的输出电压执行升压,以避免低温导致的异常关机。

触摸传感器 180K,也称“触控器件”。触摸传感器 180K 可以设置于显示屏 194,由触摸传感器 180K 与显示屏 194 组成触摸屏,也称“触控屏”。触摸传感器 180K 用于检测作用于其上或附近的触摸操作。触摸传感器可以将检测到的触摸操作传递给应用处理器,以确定触摸事件类型。可以通过显示屏 194 提供与触摸操作相关的视觉输出。在另一些实施例中,触摸传感器 180K 也可以设置于设备 100 的表面,与显示屏 194 所处的位置不同。

骨传导传感器 180M 可以获取振动信号。在一些实施例中,骨传导传感器 180M 可以获取人体声部振动骨块的振动信号。骨传导传感器 180M 也可以接触人体脉搏,接收血压跳动信号。在一些实施例中,骨传导传感器 180M 也可以设置于耳机中,结合成骨传导耳机。音频模块 170 可以基于所述骨传导传感器 180M 获取的声部振动骨块的振动信号,解析出语音信号,实现语音功能。应用处理器可以基于所述骨传导传感器 180M 获取的血压跳动信号解析心率信息,实现心率检测功能。

按键 190 包括开机键,音量键等。按键 190 可以是机械按键。也可以是触摸式按键。设备 100 可以接收按键输入,产生与设备 100 的用户设置以及功能控制有关的键信号输入。

马达 191 可以产生振动提示。马达 191 可以用于来电振动提示,也可以用于触摸振动反馈。例如,作用于不同应用(例如拍照,音频播放等)的触摸操作,可以对应不同的振动反馈效果。作用于显示屏 194 不同区域的触摸操作,马达 191 也可对应不同的振动反馈效果。不同的应用场景(例如:时间提醒,接收信息,闹钟,游戏等)也可以对应不同的振动反馈效果。触摸振动反馈效果还可以支持自定义。

指示器 192 可以是指示灯,可以用于指示充电状态,电量变化,也可以用于指示消息,未接来电,通知等。

SIM 卡接口 195 用于连接 SIM 卡。SIM 卡可以通过插入 SIM 卡接口 195,或从 SIM 卡接口 195 拔出,实现和设备 100 的接触和分离。设备 100 可以支持 1 个或 N 个 SIM 卡接口,N 为大于 1 的正整数。SIM 卡接口 195 可以支持 Nano SIM 卡, Micro SIM 卡, SIM 卡等。同一个 SIM 卡接口 195 可以同时插入多张卡。所述多张卡的类型可以相同,也可以不同。SIM 卡接口 195 也可以兼容不同类型的 SIM 卡。SIM 卡接口 195 也可以兼容外部存储卡。设备 100 通过 SIM 卡和网络交互,实现通话以及数据通信等功能。在一些实施例中,设备 100 采用 eSIM,即:嵌入式 SIM 卡。eSIM 卡可以嵌在设备 100 中,不能和设备 100 分离。

设备 100 的软件系统可以采用分层架构,事件驱动架构,微核架构,微服务架构,或云架构。本发

明实施例以分层架构的 Android 系统为例，示例性说明设备 100 的软件结构。

图 2 是本发明实施例的设备 100 的软件结构框图。

分层架构将软件分成若干个层，每一层都有清晰的角色和分工。层与层之间通过软件接口通信。在一些实施例中，将 Android 系统分为四层，从上至下分别为应用程序层，应用程序框架层，安卓运行时(Android runtime)和系统库，以及内核层。

应用程序层可以包括一系列应用程序包。

如图 2 所示，应用程序包可以包括相机，图库，日历，通话，地图，导航，WLAN，蓝牙，音乐，视频，短信息等应用程序。

应用程序框架层为应用程序层的应用程序提供应用编程接口(application programming interface, API)和编程框架。应用程序框架层包括一些预先定义的函数。

如图 2 所示，应用程序框架层可以包括窗口管理器，内容提供器，视图系统，电话管理器，资源管理器，通知管理等。

窗口管理器用于管理窗口程序。窗口管理器可以获取显示屏大小，判断是否有状态栏，锁定屏幕，截取屏幕等。

内容提供器用来存放和获取数据，并使这些数据可以被应用程序访问。所述数据可以包括视频，图像，音频，拨打和接听的电话，浏览历史和书签，电话簿等。

视图系统包括可视控件，例如显示文字的控件，显示图片的控件等。视图系统可用于构建应用程序。显示界面可以由一个或多个视图组成的。例如，包括短信通知图标的显示界面，可以包括显示文字的视图以及显示图片的视图。

电话管理器用于提供设备 100 的通信功能。例如通话状态的管理(包括接通，挂断等)。

资源管理器为应用程序提供各种资源，比如本地化字符串，图标，图片，布局文件，视频文件等等。

通知管理器使应用程序可以在状态栏中显示通知信息，可以用于传达告知类型的消息，可以短暂停留后自动消失，无需用户交互。比如通知管理器被用于告知下载完成，消息提醒等。通知管理器还可以是以图表或者滚动条文本形式出现在系统顶部状态栏的通知，例如后台运行的应用程序的通知，还可以是对话窗口形式出现在屏幕上的通知。例如在状态栏提示文本信息，发出提示音，设备振动，指示灯闪烁等。

Android Runtime 包括核心库和虚拟机。Android runtime 负责安卓系统的调度和管理。

核心库包含两部分：一部分是 java 语言需要调用的功能函数，另一部分是安卓的核心库。

应用程序层和应用程序框架层运行在虚拟机中。虚拟机将应用程序层和应用程序框架层的 java 文件执行为二进制文件。虚拟机用于执行对象生命周期的管理，堆栈管理，线程管理，安全和异常的管理，以及垃圾回收等功能。

系统库可以包括多个功能模块。例如：表面管理器(surface manager)，媒体库(Media Libraries)，三维图形处理库(例如：OpenGL ES)，2D 图形引擎(例如：SGL)等。

表面管理器用于对显示子系统进行管理，并且为多个应用程序提供了 2D 和 3D 图层的融合。

媒体库支持多种常用的音频，视频格式回放和录制，以及静态图像文件等。媒体库可以支持多种音频视频编码格式，例如：MPEG4，H.264，MP3，AAC，AMR，JPG，PNG 等。

三维图形处理库用于实现三维图形绘图，图像渲染，合成，和图层处理等。

2D 图形引擎是 2D 绘图的绘图引擎。

内核层是硬件和软件之间的层。内核层至少包含显示驱动，摄像头驱动，音频驱动，传感器驱动。

下面结合捕获拍照场景，示例性说明设备 100 软件以及硬件的工作流程。

当触摸传感器 180K 接收到触摸操作，相应的硬件中断被发给内核层。内核层将触摸操作加工成原始输入事件(包括触摸坐标，触摸操作的时间戳等信息)。原始输入事件被存储在内核层。应用程序框架层从内核层获取原始输入事件，识别该输入事件所对应的控件。以该触摸操作是触摸单击操作，该单击操作所对应的控件为相机应用图标的控件为例，相机应用调用应用框架层的接口，启动相机应用，进而通过调用内核层启动摄像头驱动，通过摄像头 193 捕获静态图像或视频。

首先，对本发明实施例涉及的请求调度方法的相关技术进行解释，以便于本领域技术人员理解。

转换器/变换器(Transformer)：一种特殊的深度学习领域的结构，包括多头自注意力和前向反馈两个部分，这一机制可以按输入数据各部分重要性的不同而分配不同的权重。该模型主要用于自然语言处

理 (NLP) 与计算机视觉 (CV) 领域。

标记/令牌 (Token): 中文自然语言处理中, 一个 token 对应着文本里的一个字或一个词。输入文本会首先通过分词器和词表映射处理, 获取对应的 token 序列后, 再送入模型进一步处理。

Prefill 推理: 通常输入信息长度很大, 时延上占决定性因素, 存在计算瓶颈。

Decode 推理: 输入信息长度固定为 1, 时延开销较小, 存在访存瓶颈。

吞吐量 (Throughout): 系统在单位时间内处理请求的数量, 本发明所指吞吐量计算公式为:

$$TPS = \frac{\text{总请求数量}}{\text{芯片数量} * \text{处理时长}}$$

大规模语言模型在 AI 界掀起了新的浪潮。以生成式模型为代表的 GPT 类模型的推理过程很有特点, 给定一组 S 长度的输入 token 序列, 模型会输出一个长度为 N 的输出 token 序列, 其实该过程存在 N 次推理过程。即 GPT 类模型一次推理只输出一个 token, 输出的 token 会与之前的输入 token 拼接到一起 (S+1、S+2、...、S+N-1), 作为下一次推理的输入, 直到遇到终止符或最大输出长度。

这样的处理方式会引入一个问题: 随着迭代次数的增多, 每次推理的计算量会随之增长; 并且, 由于每次迭代都包含上一次迭代的输入部分, 因此存在大量的冗余计算。因此, 技术人员通过 KV cache 技术将首轮迭代后的每轮推理的计算量维持基本恒定, 来节约算力和降低推理时延。

使用 KV cache 技术的 GPT 推理过程分为第一阶段, 即首轮上下文填充 (Prefill) 阶段和第二阶段, 即多轮的解码 (Decode) 阶段。

Prefill 阶段, 输入数据维度为 [B, S], B 为组批次的维度, 即机器学习里常见的 batch (批次)。模型中间层数据处理和传输维度为 [B, S, H], H 为隐藏层维度。输出数据维度是 [B, 1], 即批次个 token, 若只有一个批次就只有一个 token, 仅执行推理一次, 推理的同时会将中间层, 即转换器 (Transformer) 层的键值缓存 KV 信息缓存到 KV cache 中, 以备 Decode 阶段取用。

Decode 阶段, 输入数据维度为 [B, 1]。模型中间层数据处理和传输维度为 [B, 1, H]。输出数据维度是 [B, 1]。Decode 阶段会执行 N 次, 直到遇到停止条件停止执行。同样的, 推理过程先从 KV cache 中获取前一轮存储的键值缓存 KV 信息, 同时会将新生成的键值缓存 KV 信息缓存到 KV cache 进行更新。

表 1 GPT 大模型计算、访存关系

S=4K/16K	Prefill 阶段	Decode 阶段
计算 (TFLOPS)	572/2816	0.1396/0.1719
访存 (TB)	0.1853/0.3867	0.1343/0.1504
计算/访存比 (FLOPS/Byte)	3087/7281	1.04/1.14

通过统计计算, 可以获得在 B=1, L=88, S=4K/16K 时推理过程的计算量和访存 (访问内存) 关系, 如表 1 所示, L 为网络层中 Transformer 的个数。显而易见, 在 Prefill 阶段计算是决定性因素 (高时延, 计算瓶颈), 而 Decode 阶段访存是决定性因素 (低时延, 访存瓶颈)。当存在大量用户的服务请求时, 将会采用将不同服务请求组批次进行推理; 而两种推理阶段的显著差异, 将导致两个问题: 1、由于通常采用补齐长度的方式, 将一个批次内短的输入语句补齐到长的输入语句长度, 然后进行推理, 因此短的输入语句在 Prefill 阶段存在大量无效计算; 2、短的输出语句将等待一个批次内最长的输出语句推理结束统一进行输出, 同样存在大量无效计算。这两种问题将导致整个系统的吞吐量很低。

基于上述技术问题, 本发明实施例提供一种请求调度系统。本发明主要解决针对大规模语言模型服务部署中的服务系统的吞吐量提升问题, 其中大规模语言模型针对 transformer 为结构的模型。

当前主要存在问题: 大模型推理过程存在两种推理状态, 即 Prefill 阶段和 Decode 阶段。服务部署未考虑到两个阶段在算力、访存等方面的显著差异和自身特点; 笼统的将两个阶段部署于同一计算节点中, 当大批量服务同时请求服务时, 若提供单服务请求独享同一计算节点, 将导致服务部署成本巨大; 组批次推理 (即: 将不同的服务进行组批次, 放在同一计算节点中) 将存在大量的冗余计算和时延开销, 导致服务系统整体吞吐量低下。

图 3 为本发明实施例提供的一种请求调度系统的架构图。

请求调度系统包括第一设备、至少一个第二设备、至少一个第三设备。第二设备和第三设备均与第一设备连接。其中, 连接包括有线连接或无线连接。

其中, 本发明实施例提供的设备的硬件结构和软件结构可以参见图 1 和图 2 中关于设备 100 的相关

描述。

示例性的，第一设备包括服务器。

示例性的，第二设备和第三设备包括具备计算能力的电子设备。电子设备包括计算卡。示例性的，计算卡包括 GPU 或 NPU。第二设备用于进行 Prefill 推理，即 Prefill 阶段的推理；第三设备用于进行 Decode 推理，即 Decode 阶段的推理。

以请求调度系统包括第一设备、第二设备和第三设备为例，如图 3，请求调度系统包括第一设备 210、第二设备 220、第三设备 230。

第一设备 210 包括大规模语言模型 (LLM) 服务模块 211 和服务调度模块 212。

LLM 服务模块 211 用于对接外部用户服务入口。

服务调度模块 212 用于调度新服务进入第二设备 220，调度未达到停止条件的旧服务进入第三设备 230。

第二设备 220 包括第一推理组件包 221、第一 AI 引擎 222 和第一计算资源模块 223。

第一推理组件包 221、第一 AI 引擎 222 用于大规模语言模型的部署和调用资源。

第一计算资源模块 223 包括第一批次调度模块 2231 和第一执行引擎 2232。

第一批次调度模块 2231 用于对新服务进行动态组批次或动态插入。其中，新服务为未经过 Prefill 阶段的服务。

第一执行引擎 2232 用于对第一批次调度模块 2231 输出的服务请求进行推理计算。

第三设备 230 包括第二推理组件包 231、第二 AI 引擎 232 和第二计算资源模块 233。

第二推理组件包 231、第二 AI 引擎 232 用于大规模语言模型的部署和调用资源。

第二计算资源模块 233 包括第二批次调度模块 2331 和第二执行引擎 2332。

第二批次调度模块 2331 用于对未达到停止条件的旧服务进行动态组批次、动态插入或动态早停。其中，旧服务为已经过 Prefill 阶段的服务。

第二执行引擎 2332 用于对第二批次调度模块 2331 输出的服务请求进行推理计算。

如图 3 所示，第二设备 220 还包括第一存储模块 241，第三设备 230 还包括第二存储模块 242。第二设备 220 和第三设备 230 有线连接或无线连接。

在一些可能的实施例中，第一存储模块 241 用于缓存第二设备 220 在推理过程中生成的第一缓存信息，并将已缓存的第一缓存信息传输至第二存储模块 242；第二执行引擎 2232 用于在推理过程中从第二存储模块 242 获取第一缓存信息。

在一些可能的实施例中，第一执行引擎 2232 将第二设备 220 在推理过程中生成的第一缓存信息不发送至第一存储模块 241，而是直接发送至第二执行引擎 2232，以供第二执行引擎 2232 在推理过程中使用。

第二存储模块 242 还用于缓存第三设备 230 在推理过程中生成的缓存信息；第二执行引擎 2232 用于在推理过程中从第二存储模块 242 获取上一轮推理过程生成的缓存信息。

示例性的，缓存信息为键值缓存 KV 信息。

多个服务的请求由服务调度模块 212 调度至第二设备 220 或第三设备 230，通过第二设备 220 或第三设备 230 计算，获得当前迭代轮的输出信息和对应的缓存信息，输出信息会返回至服务调度模块 212。当不满足停止条件时，服务调度模块 212 将本服务调度至空闲的第三设备 230，进行下一轮推理；推理迭代后会继续返回给服务调度模块 212。当满足停止条件时，服务调度模块 212 将结束本服务的调度，给出对外的业务响应。

本发明实施例同时在请求调度系统中，采用独立的分离部署方式，部署第二设备 220，即 Prefill 推理执行容器和第三设备 230，即 Decode 推理执行容器两种执行容器，通过第一设备对请求的调度，使第二设备和第三设备分别执行各自阶段的推理过程；同时，两个设备间可以进行通信传递，使得推理过程可以正常进行，从而能够提升服务系统的吞吐量。

两种执行容器均拥有一个批次调度模块，批次调度模块进行两种功能的实现：自动分批和自动早停。自动分批用于将多个服务请求的输入信息进行动态的组批次，包括组批次和动态插入两种方式；自动早停用于将已完成的服务提前停止，释放对应资源并通知系统当前资源可被下一服务插入和占用。

通过分离部署的方式，综合考虑了 Prefill 阶段算力瓶颈和 Decode 阶段访存瓶颈的显著差异，能够提升服务系统的吞吐量。

请求调度系统中第二设备和第三设备的数量较多时，采用独立的分离部署方式，部署第二设备和第

三设备，第二设备配置小批次或单批次，第三设备配置大批次。第二设备资源和第三设备资源可以通过系统负载动态调整资源比例，以获得最大的吞吐量。本发明可大幅提升服务系统的吞吐量。

基于图 3 所示的系统架构，本发明实施例提供一种请求调度方法。图 4 为本发明实施例提供的一种请求调度方法的信令交互图。如图 4 所示，该方法包括：

步骤 302、LLM 服务模块接收来自第一用户的第一请求。在步骤 302 之前，即请求调度系统响应服务前，已经通过推理组件包和 AI 引擎将 LLM 部署至第二设备 220 和第三设备 230 内的计算资源模块中，计算资源模块与存储模块均已准备就绪。

示例性的，第一请求为第一服务的请求。

示例性的，第一请求包括第一服务的标识。

示例性的，标识包括服务 ID。

示例性的，第一请求不包括第二设备和第三设备返回的请求，即第一请求为新服务的请求。

该步骤中，如图 3 所示，LLM 服务模块接收来自第一用户的第一请求。

步骤 304、LLM 服务模块将第一请求发送至服务调度模块。

该步骤中，如图 3 所示，LLM 服务模块将第一请求发送至服务调度模块。

步骤 306、服务调度模块将第一请求发送至第一推理组件包。

该步骤中，如图 3 所示，服务调度模块将新到达的服务的请求调度至 Prefill 推理执行容器，即第二设备，同时建立服务池用于存储由于资源受限而等待的服务的请求。

步骤 308、第一推理组件包将第一请求发送至第一 AI 引擎。

该步骤中，如图 3 所示，第一推理组件包将第一请求发送至第一 AI 引擎。

步骤 310、第一 AI 引擎将第一请求发送至第一计算资源模块。

该步骤中，如图 3 所示，第一 AI 引擎将第一请求发送至第一计算资源模块。

步骤 312、第一计算资源模块对第一请求进行组批次得到第一批次，对第一批次进行第一推理计算，得到第二请求和第一缓存信息。

该步骤中，如图 3 所示，第一计算资源模块的第一批次调度模块根据第一计算资源状态和第一条件对第一请求和与第一请求同时到达第二设备的至少一个第七请求进行组批次得到第一批次；第一计算资源模块的第一执行引擎对第一批次进行第一推理计算，得到第二请求和第一缓存信息。第七请求为来自第一用户或第二用户或其他用户的请求。

示例性的，第一请求还包括输入信息；

示例性的，输入信息包括输入 Token 序列。

示例性的，第一计算资源状态包括第二设备中的空闲资源的数量。

示例性的，第一条件包括第二设备最多能组的批次的数量。

示例性的，第二请求包括第一服务的标识以及第一服务对应的第一输出信息，第二请求与第一请求相关联。

示例性的，第一输出信息包括输出 Token 序列。第一输出信息为第一输出 Token 序列。

示例性的，缓存信息包括键值缓存 KV 信息。第一缓存信息为第一键值缓存 KV 信息。

在一些可能的实施例中，步骤 312 具体包括：第一计算资源模块的第一批次调度模块根据第一请求，得到与第一请求对应的输入信息及其第一长度；根据第一计算资源状态、第一条件和第一长度，对第一请求和与第一请求同时到达第二设备的至少一个第七请求进行组批次得到第一批次；第一计算资源模块的第一执行引擎采用单批次处理方式对第一批次进行第一推理计算，得到第二请求和第一服务的第一缓存信息。其中，第七请求对应的输入信息的长度和第一长度相似，有利于在第一推理计算过程中减少空闲资源的浪费。

示例性的，第一批次包括多个请求。

示例性的，第一批次调度模块将同一时刻到达的多个请求组成一个批次。

示例性的，第一推理计算为 Prefill 阶段的推理计算，即 Prefill 计算。

示例性的，第一处理方式为当前批次的所有请求均推理计算结束后，释放当前批次所占用的计算服务资源并将下一批次插入空闲计算服务资源进行第一推理计算。本发明实施例中，对于 Prefill 阶段的推理计算，按照单批次进行推理计算，即当前批次的所有请求均推理计算结束后，再释放当前批次所占用的计算服务资源并将下一批次插入空闲计算服务资源，开始对下一批次进行第一推理计算。

示例性的，第二设备在释放当前批次所占用的计算服务资源的同时，还释放当前批次所占用的存储资源。

示例性的，第二设备存在空闲的计算服务资源时，向第一设备通知第二设备存在空闲的计算服务资源，第一设备根据该通知向第二设备调度新服务。

示例性的，第二设备不存在空闲的计算服务资源时，向第一设备通知第二设备不存在空闲的计算服务资源，第一设备根据该通知暂停向第二设备调度新服务。

步骤 314、第一计算资源模块将第一缓存信息发送至第二计算资源模块。

该步骤中，如图 3 所示，第一计算资源模块的第一执行引擎将第一缓存信息发送至第二计算资源模块。

可选地，如图 3 所示，第一计算资源模块可以将第一缓存信息发送至第一存储模块，第一存储模块将第一缓存信息发送至第二存储模块，然后第二存储模块将第一缓存信息发送至第二执行引擎模块。其中，第二存储模块可以是自动将第一缓存信息发送至第二执行引擎模块，也可以是根据第二执行引擎模块发送的消息将第一缓存信息发送至第二执行引擎模块。

可选地，如图 3 所示，第一计算资源模块可以将第一缓存信息直接发送至第二计算资源模块。

步骤 316、第一计算资源模块将第二请求发送至服务调度模块，继续执行 318 或 320。

该步骤中，如图 3 和图 4 所示，第一计算资源模块将第二请求发送至服务调度模块。

步骤 318、当第一输出信息满足设定的条件时，服务调度模块将第一输出信息发送至 LLM 服务模块。LLM 服务模块向第一用户发送第一输出信息，流程结束。

本发明实施例中，通过第一设备判断服务的输出信息是否满足设定的条件，若不满足设定的条件，继续对服务进行调度和推理计算；若满足设定的条件，对服务进行早停，并向第一用户发送第二输出信息。

步骤 320、当第一输出信息不满足设定的条件时，服务调度模块将第二请求调度至第二推理组件包。

该步骤中，如图 3 所示，服务调度模块将第二请求发送至第二推理组件包。

示例性的，第二请求为第二设备返回的第一服务的请求。

示例性的，设定的条件包括以下至少一个：

输出信息的长度大于或者等于第一阈值，或

输出信息包括终止符。

步骤 322、第二推理组件包将第二请求发送至第二 AI 引擎。

该步骤中，如图 3 所示，第二推理组件包将第二请求发送至第二 AI 引擎。

步骤 324、第二 AI 引擎将第二请求发送至第二计算资源模块。

该步骤中，如图 3 所示，第二 AI 引擎将第二请求发送至第二计算资源模块。

步骤 326、第二计算资源模块对第二请求进行组批次得到第二批次。

该步骤中，如图 3 所示，第二计算资源模块的第二批次调度模块根据第二计算资源状态和第二条件对第二请求进行组批次得到第二批次。

在一些可能的实施例中，步骤 326 具体包括：第二批次调度模块根据第二计算资源状态、第二条件，对第二请求和与第二请求同时到达第三设备的至少一个第八请求进行组批次得到第二批次，第八请求来自第二设备或第三设备。

示例性的，第二计算资源状态包括第三设备中的空闲资源的数量。

示例性的，第二条件包括第三设备最多能组的批次的数量。

示例性的，第二批次包括多个请求。

示例性的，第二批次调度模块将同一时刻到达的至少一个请求组成一个批次。Decode 阶段，每个请求的输入长度均为 1，只要当前批次有空闲位置，就可以将这个长度为 1 的 token 放到批次里，进行推理。

步骤 328、第二计算资源模块根据第一缓存信息，对第二批次进行第二推理计算，得到第三请求和第二缓存信息。

该步骤中，如图 3 所示，第二计算资源模块的第二执行引擎根据第一缓存信息，对第二批次进行第二推理计算，得到第三请求和第二缓存信息。

在一些可能的实施例中，步骤 328 具体包括：第二计算资源模块的第二执行引擎根据第一缓存信息，采用第二处理方式对第二批次进行第二推理计算，得到第三请求和第一服务的第二缓存信息。

示例性的，第三请求包括第一服务的标识以及第一服务对应的第二输出信息，第三请求与第二请求相关联。

示例性的，第二缓存信息为第三设备对第三请求进行第二推理计算时需要的缓存信息。

示例性的，第二推理计算为 Decode 阶段的推理计算，即 Decode 计算。

示例性的，第二处理方式是在当前批次推理过程中若存在输出信息满足设定的条件的目标请求，释放所述目标请求的计算服务资源并将下一批次中的请求插入空闲的计算服务资源进行第二推理计算。本发明实施例中，对于 Decode 阶段的推理计算，按照多批次进行推理计算，即在当前批次推理过程中，若有服务提前结束推理，即存在输出信息满足设定的条件的目标请求，可以提前释放该目标请求的计算服务资源，并在当前批次未结束之前，将下一批次中的请求插入空闲的计算服务资源中以进行第二推理计算。因此，相比于第一处理方式需要等待当前批次中的请求全部推理结束才释放当前批次的资源，第二处理方式无需等待当前批次中的请求全部推理结束才释放当前批次的资源，不会存在空闲资源的浪费，从而更加提升系统的吞吐量。

示例性的，第三设备在释放目标请求所占用的计算服务资源的同时，还释放目标请求所占用的存储资源。

示例性的，第三设备存在空闲的计算服务资源时，向第一设备通知第三设备存在空闲的计算服务资源，第一设备根据该通知向第二设备调度旧服务。

示例性的，第三设备不存在空闲的计算服务资源时，向第一设备通知第三设备不存在空闲的计算服务资源，第一设备根据该通知暂停向第三设备调度旧服务。

步骤 330、第二计算资源模块将第二缓存信息发送至第二存储模块。

该步骤中，如图 3 所示，第二计算资源模块的第二执行引擎将第二缓存信息发送至第二存储模块。

步骤 332、第二计算资源模块将第三请求发送至服务调度模块，并继续执行步骤 334 或步骤 336。

该步骤中，如图 3 所示，第二计算资源模块的第二执行引擎将第三请求发送至服务调度模块。

示例性的，第三请求为第三设备返回的第一服务的请求。

步骤 334、当第二输出信息满足设定的条件时，服务调度模块将第二输出信息发送至 LLM 服务模块。

LLM 服务模块向第一用户发送第二输出信息，流程结束。

本发明实施例中，通过第一设备判断服务的输出信息是否满足设定的条件，若不满足设定的条件，继续对服务进行调度和推理计算；若满足设定的条件，对服务进行早停，并向第一用户发送第二输出信息。第一设备对服务进行早停后，可以通知第三设备释放该服务对应的资源；第三设备向第一设备上报第三设备的当前资源状态，若当前资源状态存在空闲的计算服务资源，第一设备向第三设备继续调度服务以插入空闲的计算服务资源，从而实现动态插入。

该步骤中，如图 3 所示，当第二输出信息满足设定的条件时，服务调度模块将第二输出信息发送至 LLM 服务模块。LLM 服务模块向第一用户发送第二输出信息，流程结束。

步骤 336、当第二输出信息不满足设定的条件时，服务调度模块将第三请求调度至第二推理组件包。

该步骤中，如图 3 所示，当第二输出信息不满足设定的条件时，服务调度模块将第三请求发送至第二推理组件包，使得第三设备对第三请求继续进行下一轮的 Decode 计算，在下一轮的 Decode 计算中使用上一轮 Decode 计算生成的第二缓存信息。第二执行引擎将下一轮 Decode 计算生成的缓存信息发送至第二存储模块进行存储，以供第二执行引擎执行再下一轮 Decode 计算时使用。

以请求调度服务系统包括 4 个第二设备和 1 个第三设备为例，图 5 为本发明实施例中 Prefill 推理和 Decode 推理的示意图。如图 5 所示，4 个第二设备分别是设备 0、设备 1、设备 2 和设备 3；一个第三设备为设备 x。图 5 展示了以 Prefill 推理采用第一处理方式、Decode 推理采用第二处理方式的推理过程，即：左侧 Prefill 推理中一行代表在一组计算资源中处理一个服务的请求，一共有 4 组 Prefill 计算资源、4 个服务的请求；右侧 Decode 推理中 4 行代表在一组计算资源中处理一个批次为 4 的服务的请求，一共有 1 组计算资源、4 个服务的请求。图 5 中，每一个方格代表一个计算时延。左侧 Prefill 计算和右侧 Decode 计算过程并行进行，不存在时序关系。

图 5 所示的 Prefill 推理和 Decode 推理的流程如下：

1) req1、req2、req3 和 req4 这 4 个服务的输入 Token 序列的长度分别为 3、2、4、3，因此当 4 个服务是同时到达设备 0、设备 1、设备 2 和设备 3 时，req2 会提前运算完成，其次为 req1、req4、req3。

2) Prefill 计算中，提前完成的 req2 将提前进入 Decode 计算。在图 5 的右侧展示了在系统启动过

程的示意；设备 x 要组成 1 个批次，1 个批次包括 4 个请求，则需要 req1、req2、req3 和 req4 这 4 个服务均进入 Decode 计算。如图 5 所示，右侧 Decode 推理中，req2 首先到达，其次是 req1、req4 到达，最后是 req3 到达。因此，req1、req4 将产生 1 个单位的空闲气泡（bubble），而 req3 将产生两个单位的空闲 bubble。这里的 bubble 在系统稳定后将不会再次存在。

3) Decode 计算中，req1~4 的输出 token 数量分别为 5、7、6、4，因此 req4 虽然在 Prefill 计算中是第二个进入 Decode 计算的，但它却是最早完成 Decode 计算的，因此 req4 的输出信息是所在批次中最早满足设定的条件的。完成 Decode 计算的 req4，将提前停止运算（早停机制）同时释放相应资源占用；其次完成计算的请求分别为 req1、req2、req3。

4) 在 req1~4 的 Decode 计算过程，req5~8 已到达 Prefill 推理执行容器进行计算并且在 req1~4 计算结束之前已完成计算。因此，在 req4、req1、req2、req3 相继停止计算过程中，服务调度模块会相继自动调度 req6、req5、req7、req8 分别设备 x，设备 x 相继将 req6、req5、req7、req8 分别插入 req4、req1、req2、req3 之前所占用的资源位置，动态插入后进行后续计算。

5) 系统稳定后，将重复进行 3)、4) 步骤，动态插入组批和动态的停止并给出服务响应。

需要说明的是，服务调度模块将第一服务的第三请求调度至第三设备的时候，只会将第三请求调度至存储有该第一服务上一轮 Decode 推理时生成的缓存信息的第三设备中。因此，Decode 阶段，Decode 部分是在当前第三设备迭代的服务，该服务的请求不会转发到其它第三设备。

综上，本发明通过执行引擎将推理计算后输出的请求发送至服务调度模块，服务调度模块判断出请求满足设定的条件，从而实现服务的自动早停；通过批次调度模块对请求进行组批次得到批次，从而实现组批次；在执行引擎存在空闲的计算服务资源时，通过批次调度模块自动将下一个批次中的请求发送至执行引擎实现动态插入。

本发明实施例将来自用户的新服务的请求调度至第二设备，即 Prefill 推理执行容器进行 Prefill 阶段的推理计算；将经过 Prefill 阶段的旧服务的请求调度至第三设备，即 Decode 推理执行容器进行 Decode 阶段的推理计算。本发明通过分离部署的方式，综合考虑了 Prefill 阶段算力瓶颈和 Decode 阶段访存瓶颈的显著差异，从而提升服务系统的吞吐量。

可选地，如图 6 所示，该方法还包括：步骤 338-步骤 372。

步骤 338、LLM 服务模块接收来自第二用户的第四请求。在步骤 338 之前，即请求调度系统响应服务前，已经通过推理组件包和 AI 引擎将 LLM 部署至第二设备和第三设备内的计算资源模块中，计算资源模块与存储模块均已准备就绪。

示例性的，第二用户不同于第一用户。，第一设备能够调度来自不同用户的请求。

示例性的，第四请求为第二服务的请求。

示例性的，第四请求包括第二服务的标识。

示例性的，标识包括服务 ID。

示例性的，第四请求不包括第二设备和第三设备返回的请求，即第四请求为新服务的请求。

该步骤中，如图 3 所示，LLM 服务模块接收来自第二用户的第四请求。

步骤 340、LLM 服务模块将第四请求发送至服务调度模块。

该步骤中，如图 3 所示，LLM 服务模块将第四请求发送至服务调度模块。

步骤 342、服务调度模块将第四请求发送至第一推理组件包。

该步骤中，如图 3 所示，服务调度模块将新到达的服务的请求调度至 Prefill 推理执行容器，即第二设备，同时建立服务池用于存储由于资源受限而等待的服务的请求。

步骤 344、第一推理组件包将第四请求发送至第一 AI 引擎。

该步骤中，如图 3 所示，第一推理组件包将第四请求发送至第一 AI 引擎。

步骤 346、第一 AI 引擎将第四请求发送至第一计算资源模块。

该步骤中，如图 3 所示，第一 AI 引擎将第四请求发送至第一计算资源模块。

步骤 348、第一计算资源模块对第四请求进行组批次得到第三批次，对第三批次进行第一推理计算，得到第五请求和第三缓存信息。

该步骤中，如图 3 所示，第一计算资源模块的第一批次调度模块根据第一计算资源状态和第一条件对第四请求进行组批次得到第三批次；第一计算资源模块的第一执行引擎对第三批次进行第一推理计算，得到第五请求和第三缓存信息。

示例性的，第四请求还包括输入信息：

示例性的，输入信息包括输入 Token 序列。

示例性的，第一计算资源状态包括第二设备中的空闲资源的数量。

示例性的，第一条件包括第二设备最多能组的批次的数量。

示例性的，第五请求包括第二服务的标识以及第二服务对应的第三输出信息，第五请求与第四请求相关联。

示例性的，第三输出信息包括输出 Token 序列。第三输出信息为第三输出 Token 序列。

示例性的，缓存信息包括键值缓存 KV 信息。第三缓存信息为第三键值缓存 KV 信息。

在一些可能的实施例中，步骤 348 具体包括：第一计算资源模块的第一批次调度模块根据第四请求，得到与第四请求对应的输入信息及其第二长度；根据第一计算资源状态、第一条件和第二长度，对第四请求进行组批次得到第三批次；第一计算资源模块的第一执行引擎采用第一处理方式对第三批次进行第一推理计算，得到第五请求和第二服务的第三缓存信息。

示例性的，第三批次包括多个请求。

示例性的，第一批次调度模块将同一时刻到达的多个请求组成一个批次。

示例性的，第一推理计算为 Prefill 阶段的推理计算，即 Prefill 计算。

示例性的，第一处理方式为当前批次的所有请求均推理计算结束后，释放当前批次所占用的计算服务资源并将下一批次插入空闲计算服务资源进行第一推理计算。

步骤 350、第一计算资源模块将第三缓存信息发送至第二计算资源模块。

该步骤中，如图 3 所示，第一计算资源模块的第一执行引擎将第三缓存信息发送至第二计算资源模块。

可选地，如图 3 所示，第一计算资源模块可以将第三缓存信息发送至第一存储模块，第一存储模块将第三缓存信息发送至第二存储模块，然后第二存储模块将第三缓存信息发送至第二执行引擎模块。其中，第二存储模块可以是自动将第三缓存信息发送至第二执行引擎模块，也可以是根据第二执行引擎模块发送的消息将第三缓存信息发送至第二执行引擎模块。

可选地，如图 3 所示，第一计算资源模块可以将第三缓存信息直接发送至第二计算资源模块。

步骤 352、第一计算资源模块将第五请求发送至服务调度模块，继续执行 354 或 356。

该步骤中，如图 3 和图 4 所示，第一计算资源模块将第五请求发送至服务调度模块。

步骤 354、当第三输出信息满足设定的条件时，服务调度模块将第三输出信息发送至 LLM 服务模块。LLM 服务模块向第二用户发送第三输出信息，流程结束。

步骤 356、当第三输出信息不满足设定的条件时，服务调度模块将第五请求调度至第二推理组件包。

该步骤中，如图 3 所示，服务调度模块将第五请求发送至第二推理组件包。

示例性的，第五请求为第二设备返回的第二服务的请求。

示例性的，设定的条件包括以下至少一个：

输出信息的长度大于或者等于第一阈值，或

输出信息包括终止符。

步骤 358、第二推理组件包将第五请求发送至第二 AI 引擎。

该步骤中，如图 3 所示，第二推理组件包将第五请求发送至第二 AI 引擎。

步骤 360、第二 AI 引擎将第五请求发送至第二计算资源模块。

该步骤中，如图 3 所示，第二 AI 引擎将第五请求发送至第二计算资源模块。

步骤 362、第二计算资源模块对第五请求进行组批次得到第四批次。

该步骤中，如图 3 所示，第二计算资源模块的第二批次调度模块根据第二计算资源状态和第二条件对第五请求进行组批次得到第四批次。

在一些可能的实施例中，步骤 362 具体包括：第二批次调度模块根据第二计算资源状态、第二条件，对第五请求进行组批次得到第四批次。

示例性的，第二计算资源状态包括第三设备中的空闲资源的数量。

示例性的，第二条件包括第三设备最多能组的批次的数量。

示例性的，第四批次包括多个请求。

示例性的，第二批次调度模块将同一时刻到达的至少一个请求组成一个批次。Decode 阶段，每个请

求的输入长度均为 1，只要当前批次有空闲位置，就可以将这个长度为 1 的 token 放到批次里，进行推理。

步骤 364、第二计算资源模块根据第三缓存信息，对第四批次进行第二推理计算，得到第六请求和第四缓存信息。

该步骤中，如图 3 所示，第二计算资源模块的第二执行引擎根据第三缓存信息，对第四批次进行第二推理计算，得到第六请求和第四缓存信息。

在一些可能的实施例中，步骤 364 具体包括：第二计算资源模块的第二执行引擎根据第三缓存信息，采用第二处理方式对第四批次进行第二推理计算，得到第六请求和第二服务的第四缓存信息。

示例性的，第六请求包括第二服务的标识以及第二服务对应的第四输出信息，第六请求与第五请求相关联。

示例性的，第四缓存信息为第三设备对第六请求进行推理计算时需要的缓存信息。

示例性的，第二推理计算为 Decode 阶段的推理计算，即 Decode 计算。

示例性的，第二处理方式为在当前批次推理过程中若存在输出信息满足设定的条件的目标请求，释放所述目标请求的计算服务资源并将下一批次中的请求插入空闲的计算服务资源进行第二推理计算。

步骤 366、第二计算资源模块将第四缓存信息发送至第二存储模块。

该步骤中，如图 3 所示，第二计算资源模块的第二执行引擎将第四缓存信息发送至第二存储模块。

步骤 368、第二计算资源模块将第六请求发送至服务调度模块，并继续执行步骤 370 或步骤 372。

该步骤中，如图 3 所示，第二计算资源模块的第二执行引擎将第六请求发送至服务调度模块。

示例性的，第六请求为第三设备返回的第二服务的请求。

步骤 370、当第四输出信息满足设定的条件时，服务调度模块将第四输出信息发送至 LLM 服务模块。LLM 服务模块向第二用户发送第四输出信息，流程结束。

该步骤中，如图 3 所示，当第四输出信息满足设定的条件时，服务调度模块将第四输出信息发送至 LLM 服务模块。LLM 服务模块向第二用户发送第四输出信息，流程结束。

步骤 372、当第四输出信息不满足设定的条件时，服务调度模块将第六请求调度至第二推理组件包。

基于图 3 所示的系统架构，本发明实施例提供一种请求调度方法。图 7 为本发明实施例提供了一种请求调度方法的流程图。如图 7 所示，该方法包括：

步骤 402、第一设备接收来自第一用户的第一请求，第一请求为第一服务的请求，第一请求包括第一服务的标识。

示例性的，标识包括服务 ID。

示例性的，第一请求不包括第二设备和第三设备返回的请求，即第一请求为新服务的请求。

步骤 404、第一设备将第一请求调度至第二设备。

第一设备将新到达的服务的请求调度至 Prefill 推理执行容器，即第二设备，同时建立服务池用于存储由于资源受限而等待的服务的请求。

步骤 406、第二设备向第一设备发送第二请求，第二请求包括第一服务的标识以及第一服务对应的第一输出信息，第二请求与第一请求相关联。

示例性的，第一输出信息包括输出 Token 序列。第一输出信息为第一输出 Token 序列。

步骤 408、当第一输出信息不满足设定的条件时，第一设备将第二请求调度至第三设备，第三设备与第二设备不同。

示例性的，第二请求为第二设备返回的第一服务的请求。

示例性的，设定的条件包括以下至少一个：

输出信息的长度大于或者等于第一阈值，或

输出信息包括终止符。

步骤 410、第三设备向第一设备发送第三请求，第三请求包括第一服务的标识以及第一服务对应的第二输出信息，第三请求与第二请求相关联。

步骤 412、当第二输出信息不满足设定的条件时，第一设备将第三请求调度至第三设备。

示例性的，第三请求为第三设备返回的第一服务的请求。

第一设备将第三请求发送至第三设备，使得第三设备对第三请求继续进行下一轮的 Decode 计算，在下一轮的 Decode 计算中使用上一轮 Decode 计算生成的第二缓存信息。第三设备将下一轮 Decode 计算生成的缓存信息进行存储，以供再下一轮 Decode 计算时使用。

可选地，如图 7 所示，步骤 404 和步骤 406 之间还包括：步骤 4052-步骤 4054。

步骤 4052、第二设备根据第一计算资源状态和第一条件对第一请求进行组批次得到第一批次。

示例性的，第一请求还包括输入信息。

示例性的，输入信息包括输入 Token 序列。

示例性的，第一计算资源状态包括第二设备中的空闲资源的数量。

示例性的，第一条件包括第二设备最多能组的批次的数量。

在一些可能的实施例中，步骤 4052 具体包括：第二设备根据第一请求，得到与第一请求对应的输入信息及其第一长度；根据第一计算资源状态、第一条件和第一长度，对第一请求和与第一请求同时到达所述第二设备的至少一个第七请求进行组批次得到第一批次。

第七请求为来自第一用户或第二用户或其他用户的请求。

第七请求对应的输入信息的长度和第一长度相似，有利于在第一推理计算过程中减少空闲资源的浪费。

步骤 4054、第二设备对第一批次进行第一推理计算，得到第二请求和第一缓存信息。

示例性的，缓存信息包括键值缓存 KV 信息。第一缓存信息为第一键值缓存 KV 信息。

在一些可能的实施例中，步骤 4052 具体包括：第二设备采用第一处理方式对第一批次进行第一推理计算。

示例性的，第一推理计算为 Prefill 阶段的推理计算，即 Prefill 计算。

示例性的，第一处理方式为当前批次的所有请求均推理计算结束后，释放当前批次所占用的计算服务资源并将下一批次插入空闲计算服务资源进行第一推理计算。

本发明实施例中，步骤 4052 之后，还包括：第二设备将第一缓存信息发送至第三设备。

可选地，步骤 406 之后，还包括：当第一输出信息满足设定的条件时，第一设备向第一用户发送第一输出信息。

可选地，如图 7 所示，步骤 408 和步骤 410 之间还包括：步骤 4092-步骤 4094。

步骤 4092、第三设备根据第二计算资源状态和第二条件对第二请求进行组批次得到第二批次。

示例性的，第二计算资源状态包括第三设备中的空闲资源的数量。

示例性的，第二条件包括第三设备最多能组的批次的数量。

在一些可能的实施例中，步骤 4092 具体包括：根据第二计算资源状态和第二条件对所述第二请求和与所述第二请求同时到达所述第三设备的至少一个第八请求进行组批次得到第二批次，所述第八请求来自所述第二设备或第三设备。

步骤 4094、第三设备根据第一缓存信息，对第二批次进行第二推理计算，得到第三请求和第二缓存信息。

示例性的，第二推理计算为 Decode 阶段的推理计算，即 Decode 计算。Decode 阶段，每个请求的输入长度均为 1，只要当前批次有空闲位置，就可以将这个长度为 1 的 token 放到批次里，进行推理。

在一些可能的实施例中，步骤 4094 具体包括：第三设备根据第一缓存信息，采用第二处理方式对第二批次进行第二推理计算，得到第三请求和第二缓存信息。

示例性的，第二处理方式为在当前批次推理过程中若存在输出信息满足设定的条件的目标请求，释放目标请求的计算服务资源并将下一批次中的请求插入空闲的计算服务资源进行第二推理计算。

本发明实施例中，步骤 4094 之后，还包括：第三设备存储第二缓存信息。

示例性的，第二缓存信息为第三设备对第三请求进行第二推理计算时需要的缓存信息。

可选地，如图 7 所示，步骤 412 之后还包括：步骤 414-步骤 424。

步骤 414、第一设备接收来自第二用户的第四请求，第四请求为第二服务的请求，第四请求包括第二服务的标识。

步骤 416、第一设备将第四请求调度至第二设备。

步骤 418、第二设备向第一设备发送第五请求，第五请求包括第二服务的标识以及第二服务对应的第三输出信息，第五请求与第四请求相关联。

步骤 420、当第三输出信息不满足设定的条件时，第一设备将第五请求调度至第三设备。

步骤 422、第三设备向第一设备发送第六请求，第六请求包括第二服务的标识以及第二服务对应的第四输出信息，第六请求与第五请求相关联。

步骤 424、当第四输出信息不满足设定的条件时，第一设备将第六请求调度至第三设备。

可选地，步骤 416 和步骤 418 之间还包括：步骤 4172-步骤 4174。

步骤 4172、第二设备根据第一计算资源状态和第一条件对第四请求进行组批次得到第三批。

示例性的，第四请求还包括输入信息。

在一些可能的实施例中，步骤 4172 具体包括：第二设备根据第四请求，得到与第四请求对应的输入信息及其第二长度；根据第一计算资源状态、第一条件和第二长度，对第四请求进行组批次得到第三批。

步骤 4174、第二设备对第三批进行第一推理计算，得到第五请求和第三缓存信息。

在一些可能的实施例中，步骤 4172 具体包括：第二设备采用第一处理方式对第三批进行第一推理计算。

示例性的，第一处理方式为当前批次的所有请求均推理计算结束后，释放当前批次所占用的计算服务资源并将下一批次插入空闲计算服务资源进行第一推理计算。

本发明实施例中，步骤 4172 之后，还包括：第二设备将第三缓存信息发送至第三设备。

可选地，步骤 418 之后，还包括：当第三输出信息满足设定的条件时，第一设备向第一用户发送第三输出信息。

可选地，步骤 420 和步骤 422 之间还包括：步骤 4212-步骤 4214。

步骤 4212、第三设备根据第二计算资源状态和第二条件对第五请求进行组批次得到第四批。

步骤 4214、第三设备根据第三缓存信息，对第四批次进行第二推理计算，得到第六请求和第四缓存信息。

在一些可能的实施例中，步骤 4214 具体包括：第三设备根据第三缓存信息，采用第二处理方式对第四批次进行第二推理计算，得到第六请求和第四缓存信息。

示例性的，第二处理方式为在当前批次推理过程中若存在输出信息满足设定的条件的目标请求，释放目标请求的计算服务资源并将下一批次中的请求插入空闲的计算服务资源进行第二推理计算。

本发明实施例中，步骤 4094 之后，还包括：第三设备存储第四缓存信息。

本发明公开了一种请求调度方法和设备，所述方法包括：第一设备接收第一用户的第一请求，第一请求为第一服务的请求，第一请求包括第一服务的标识；第一设备将第一请求调度至第二设备；第二设备向第一设备发送第二请求，第二请求包括第一服务的标识和第一输出信息，第二请求与第一请求相关联；当第一输出信息不满足设定的条件时，第一设备将第二请求调度至第三设备，第三设备与第二设备不同；第三设备向第一设备发送第三请求，第三请求包括第一服务的标识和第二输出信息，第三请求与第二请求相关联；当第二输出信息不满足设定的条件时，第一设备将第三请求调度至第三设备，能够提升大规模语言模型服务部署中的服务系统的吞吐量。

基于图 3 所示的系统架构，本发明实施例提供一种请求调度方法，应用于第一设备。图 8 为本发明实施例提供的又一种请求调度方法的流程图。如图 8 所示，该方法包括：

步骤 502、第一设备接收来自第一用户的第一请求，第一请求为第一服务的请求，第一请求包括第一服务的标识。

步骤 504、将第一请求调度至第二设备。

步骤 506、接收第二设备发送的第二请求，第二请求包括第一服务的标识以及第一服务对应的第一输出信息，第二请求与第一请求相关联。

步骤 508、当第一输出信息不满足设定的条件时，将第二请求调度至第三设备，第三设备与第二设备不同。

步骤 510、接收第三设备发送的第三请求，第三请求包括第一服务的标识以及第一服务对应的第二输出信息，第三请求与第二请求相关联。

步骤 512、当第二输出信息不满足设定的条件时，将第三请求调度至第三设备。

本发明实施例提供的请求调度方法的技术方案中，所述方法包括：第一设备接收至少一个第一请求，第一请求为服务的请求，将至少一个第一请求调度至第二设备，第一请求不包括第二设备和第三设备返回的请求；接收第二设备返回的至少一个第二请求，将至少一个第二请求调度至第三设备；接收第三设备返回的至少一个第三请求，根据第三请求，判断第三请求是否满足第三条件，若判断出第三请求不满足第三条件，将第三请求调度至第三设备，能够提升大规模语言模型服务部署中的服务系统的吞吐量。

图 9 为本发明实施例提供的一种第一设备的结构示意图，应理解，第一设备 600 能够执行上述请求调度方法中第一设备的各个步骤，为了避免重复，此处不再详述。第一设备 600 包括：第一收发单元 601。

第一收发单元 601，用于接收来自第一用户的第一请求，所述第一请求为第一服务的请求，所述第一请求包括所述第一服务的标识；将所述第一请求调度至第二设备；接收所述第二设备发送的第二请求，所述第二请求包括所述第一服务的标识以及所述第一服务对应的第一输出信息，所述第二请求与所述第一请求相关联；当所述第一输出信息不满足设定的条件时，将所述第二请求调度至第三设备，所述第三设备与所述第二设备不同；接收所述第三设备发送的第三请求，所述第三请求包括所述第一服务的标识以及所述第一服务对应的第二输出信息，所述第三请求与所述第二请求相关联；当所述第二输出信息不满足所述设定的条件时，将所述第三请求调度至所述第三设备。

可选地，第一收发单元 601 还用于接收来自第二用户的第四请求，所述第四请求为第二服务的请求，所述第四请求包括所述第二服务的标识；将所述第四请求调度至所述第二设备；接收所述第二设备发送的第五请求，所述第五请求包括所述第二服务的标识以及所述第二服务对应的第三输出信息，所述第五请求与所述第四请求相关联；当所述第三输出信息不满足所述设定的条件时，将所述第五请求调度至第三设备；接收所述第三设备发送的第六请求，所述第六请求包括所述第二服务的标识以及所述第二服务对应的第四输出信息，所述第六请求与所述第五请求相关联；当所述第四输出信息不满足所述设定的条件时，将所述第六请求调度至所述第三设备。

可选地，第一收发单元 601 还用于当所述第二输出信息满足所述设定的条件时，向所述第一用户发送所述第二输出信息。

可选地，所述标识包括服务 ID。

可选地，所述设定的条件包括以下至少一个：

输出信息的长度大于或者等于第一阈值，或

所述输出信息包括终止符。

图 10 为本发明实施例提供的一种第二设备的结构示意图，应理解，第二设备 700 能够执行上述请求调度方法中第二设备的各个步骤，为了避免重复，此处不再详述。第二设备 700 包括：第二收发单元 701 和第二处理单元 702。

第二收发单元 701，用于接收第一设备发送的来自第一用户的第一请求，所述第一请求为第一服务的请求，所述第一请求包括所述第一服务的标识；向所述第一设备发送第二请求，所述第二请求包括所述第一服务的标识以及所述第一服务对应的第一输出信息，所述第二请求与所述第一请求相关联。

可选地，第二收发单元 701 还用于接收所述第一设备发送的来自第二用户的第四请求，所述第四请求为第二服务的请求，所述第四请求包括所述第二服务的标识；向所述第一设备发送第五请求，所述第五请求包括所述第二服务的标识以及所述第二服务对应的第三输出信息，所述第五请求与所述第四请求相关联。

可选地，第二收发单元 701 向所述第一设备发送第二请求之前，第二处理单元 702，用于根据第一计算资源状态和第一条件对所述第一请求和与所述第一请求同时到达所述第二设备的至少一个第七请求进行组批次得到第一批次，所述第七请求为来自第一用户或第二用户或其他用户的请求；对所述第一批次进行第一推理计算，得到所述第二请求和第一缓存信息。

可选地，第二收发单元 701 还用于将所述第一缓存信息发送至所述第三设备。

可选地，所述缓存信息包括键值缓存 KV 信息。

可选地，第二处理单元 702 具体用于根据所述第一请求，得到与所述第一请求对应的输入信息及其第一长度；根据所述第一计算资源状态、所述第一条件和所述第一长度，对所述第一请求和与所述第一请求同时到达所述第二设备的至少一个第七请求进行组批次得到第一批次。

图 11 为本发明实施例提供的一种第三设备的结构示意图，应理解，第三设备 800 能够执行上述请求调度方法中第三设备的各个步骤，为了避免重复，此处不再详述。第三设备 800 包括：第三收发单元 801 和第三处理单元 802。

第三收发单元 801，用于接收第一设备发送的来自第二设备的第二请求，所述第二请求包括第一服务的标识以及所述第一服务对应的第一输出信息，所述第二请求与所述第一请求相关联，所述第一请求为来自第一用户的所述第一服务的请求；向所述第一设备发送第三请求，所述第三请求包括所述第一服务的标识以及所述第一服务对应的第二输出信息，所述第三请求与所述第二请求相关联。

可选地，第三收发单元 801 还用于接收所述第一设备发送的来自第二设备的第五请求，所述第五请求包括第二服务的标识以及所述第二服务对应的第三输出信息，所述第五请求与第四请求相关联，所述第四请求为来自第二用户的所述第二服务的请求；向所述第一设备发送第六请求，所述第六请求包括所述第二服务的标识以及所述第二服务对应的第四输出信息，所述第六请求与所述第五请求相关联。

可选地，第三收发单元 801 向所述第一设备发送第三请求之前，第三处理单元 802，用于根据第二计算资源状态和第二条件对所述第二请求和与所述第二请求同时到达所述第三设备的至少一个第八请求进行组批得到第二批次，所述第八请求来自第二设备或第三设备；

第三收发单元 801 还用于接收所述第二设备发送的第一缓存信息；

第三处理单元 802 还用于根据所述第一缓存信息，对所述第二批次进行第二推理计算，得到所述第三请求和第二缓存信息。

应理解，这里的第一设备 600、第二设备 700 和第三设备 800 以功能单元的形式体现。这里的术语“单元”可以通过软件和/或硬件形式实现，对此不作具体限定。例如，“单元”可以是实现上述功能的软件程序、硬件电路或二者结合。所述硬件电路可能包括应用特有集成电路（application specific integrated circuit, ASIC）、电子电路、用于执行一个或多个软件或固件程序的处理器（例如共享处理器、专有处理器或组处理器等）和存储器、合并逻辑电路和/或其它支持所描述的功能的合适组件。

因此，在本发明的实施例中描述的各示例的单元，能够以电子硬件、或者计算机软件和电子硬件的结合来实现。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发明的范围。

本申请实施例提供一种设备，该设备可以是终端设备也可以是内置于所述终端设备的电路设备。该设备可以用于执行上述方法实施例中的功能/步骤。

本申请实施例提供一种可读存储介质，该可读存储介质中存储有指令，当该指令在设备或处理器上运行时，使得设备执行如上述方法实施例中的功能/步骤。

本申请实施例还提供了一种包含指令的程序产品，当该程序产品在设备或任一至少一种处理器上运行时，使得设备执行如上述方法实施例中的功能/步骤。

本申请实施例中，“至少一个”是指一个或者多个，“多个”是指两个或两个以上。“和/或”，描述关联对象的关联关系，表示可以存在三种关系，例如，A 和/或 B，可以表示单独存在 A、同时存在 A 和 B、单独存在 B 的情况。其中 A，B 可以是单数或者复数。字符“/”一般表示前后关联对象是一种“或”的关系。“以下至少一项”及其类似表达，是指的这些项中的任意组合，包括单项或复数项的任意组合。例如，a，b 和 c 中的至少一项可以表示：a，b，c，a-b，a-c，b-c，或 a-b-c，其中 a，b，c 可以是单个，也可以是多个。

本领域普通技术人员可以意识到，本文中公开的实施例中描述的各单元及算法步骤，能够以电子硬件、计算机软件和电子硬件的结合来实现。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。

所属领域的技术人员可以清楚地了解到，为描述的方便和简洁，上述描述的系统、装置和单元的具体工作过程，可以参考前述方法实施例中的对应过程，在此不再赘述。

在本申请所提供的几个实施例中，任一功能如果以软件功能单元的形式实现并作为独立的产品销售或使用时，可以存储在一个计算机可读存储介质中。基于这样的理解，本申请的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的部分可以以软件产品的形式体现出来，该计算机软件产品存储在一个存储介质中，包括若干指令用以使得一台设备（可以是个人计算机，服务器，或者网络设备等）执行本申请各个实施例所述方法的全部或部分步骤。而前述的存储介质包括：U 盘、移动硬盘、只读存储器（read-only memory，ROM）、随机存取存储器（random access memory，RAM）、磁碟或者光盘等各种可以存储程序代码的介质。

以上所述，仅为本申请的具体实施方式，任何熟悉本技术领域的技术人员在本申请揭露的技术范围内，可轻易想到变化或替换，都应涵盖在本申请的保护范围之内。本申请的保护范围应以所述权利要求的保护范围为准。

权 利 要 求 书

1. 一种请求调度方法，其特征在于，应用于请求调度系统，所述方法包括：

第一设备接收来自第一用户的第一请求，所述第一请求为第一服务的请求，所述第一请求包括所述第一服务的标识；

所述第一设备将所述第一请求调度至第二设备；

所述第二设备向所述第一设备发送第二请求，所述第二请求包括所述第一服务的标识以及所述第一服务对应的第一输出信息，所述第二请求与所述第一请求相关联；

当所述第一输出信息不满足设定的条件时，所述第一设备将所述第二请求调度至第三设备，所述第三设备与所述第二设备不同；

所述第三设备向所述第一设备发送第三请求，所述第三请求包括所述第一服务的标识以及所述第一服务对应的第二输出信息，所述第三请求与所述第二请求相关联；

当所述第二输出信息不满足所述设定的条件时，所述第一设备将所述第三请求调度至所述第三设备。

2. 根据权利要求1所述的方法，其特征在于，所述方法还包括：

所述第一设备接收来自第二用户的第四请求，所述第四请求为第二服务的请求，所述第四请求包括所述第二服务的标识；

所述第一设备将所述第四请求调度至所述第二设备；

所述第二设备向所述第一设备发送第五请求，所述第五请求包括所述第二服务的标识以及所述第二服务对应的第三输出信息，所述第五请求与所述第四请求相关联；

当所述第三输出信息不满足所述设定的条件时，所述第一设备将所述第五请求调度至第三设备；

所述第三设备向所述第一设备发送第六请求，所述第六请求包括所述第二服务的标识以及所述第二服务对应的第四输出信息，所述第六请求与所述第五请求相关联；

当所述第四输出信息不满足所述设定的条件时，所述第一设备将所述第六请求调度至所述第三设备。

3. 根据权利要求1或2所述的方法，其特征在于，所述方法还包括：

当所述第二输出信息满足所述设定的条件时，所述第一设备向所述第一用户发送所述第二输出信息。

4. 根据权利要求1-3中任一项所述的方法，其特征在于，所述请求调度系统包括所述第一设备、至少一个所述第二设备和至少一个所述第三设备。

5. 一种请求调度方法，其特征在于，应用于第一设备，所述方法包括：

接收来自第一用户的第一请求，所述第一请求为第一服务的请求，所述第一请求包括所述第一服务的标识；

将所述第一请求调度至第二设备；

接收所述第二设备发送的第二请求，所述第二请求包括所述第一服务的标识以及所述第一服务对应的第一输出信息，所述第二请求与所述第一请求相关联；

当所述第一输出信息不满足设定的条件时，将所述第二请求调度至第三设备，所述第三设备与所述第二设备不同；

接收所述第三设备发送的第三请求，所述第三请求包括所述第一服务的标识以及所述第一服务对应的第二输出信息，所述第三请求与所述第二请求相关联；

当所述第二输出信息不满足所述设定的条件时，将所述第三请求调度至所述第三设备。

6. 根据权利要求5所述的方法，其特征在于，所述方法还包括：

接收来自第二用户的第四请求，所述第四请求为第二服务的请求，所述第四请求包括所述第二服务的标识；

将所述第四请求调度至所述第二设备；

接收所述第二设备发送的第五请求，所述第五请求包括所述第二服务的标识以及所述第二服务对应的第三输出信息，所述第五请求与所述第四请求相关联；

当所述第三输出信息不满足所述设定的条件时，将所述第五请求调度至第三设备；

接收所述第三设备发送的第六请求，所述第六请求包括所述第二服务的标识以及所述第二服务对应的第四输出信息，所述第六请求与所述第五请求相关联；

当所述第四输出信息不满足所述设定的条件时，将所述第六请求调度至所述第三设备。

7. 根据权利要求 5 所述的方法, 其特征在于, 所述方法还包括:
当所述第二输出信息满足所述设定的条件时, 向所述第一用户发送所述第二输出信息。
8. 根据权利要求 5-7 中任一项所述的方法, 其特征在于, 所述标识包括服务 ID。
9. 根据权利要求 5-8 中任一项所述的方法, 其特征在于, 所述设定的条件包括以下至少一个:
输出信息的长度大于或者等于第一阈值, 或
所述输出信息包括终止符。
10. 一种请求调度方法, 其特征在于, 应用于第二设备, 所述方法包括:
接收第一设备发送的来自第一用户的第一请求, 所述第一请求为第一服务的请求, 所述第一请求包括所述第一服务的标识;
向所述第一设备发送第二请求, 所述第二请求包括所述第一服务的标识以及所述第一服务对应的第一输出信息, 所述第二请求与所述第一请求相关联。
11. 根据权利要求 10 所述的方法, 其特征在于, 所述方法还包括:
接收所述第一设备发送的来自第二用户的第四请求, 所述第四请求为第二服务的请求, 所述第四请求包括所述第二服务的标识;
向所述第一设备发送第五请求, 所述第五请求包括所述第二服务的标识以及所述第二服务对应的第三输出信息, 所述第五请求与所述第四请求相关联。
12. 根据权利要求 10 所述的方法, 其特征在于, 所述向所述第一设备发送第二请求之前, 还包括:
根据第一计算资源状态和第一条件对所述第一请求和与所述第一请求同时到达所述第二设备的至少一个第七请求进行组批次得到第一批次, 所述第七请求为来自第一用户或第二用户或其他用户的请求;
对所述第一批次进行第一推理计算, 得到所述第二请求和第一缓存信息。
13. 根据权利要求 12 所述的方法, 其特征在于, 所述方法还包括:
将所述第一缓存信息发送至所述第三设备。
14. 根据权利要求 12 或 13 所述的方法, 其特征在于, 所述缓存信息包括键值缓存 KV 信息。
15. 根据权利要求 12-14 中任一项所述的方法, 其特征在于, 所述根据第一计算资源状态和第一条件对所述第一请求和与所述第一请求同时到达所述第二设备的至少一个第七请求进行组批次得到第一批次, 包括:
根据所述第一请求, 得到与所述第一请求对应的输入信息及其第一长度;
根据所述第一计算资源状态、所述第一条件和所述第一长度, 对所述第一请求和所述至少一个第七请求进行组批次得到第一批次。
16. 一种请求调度方法, 其特征在于, 应用于第三设备, 所述方法包括:
接收第一设备发送的来自第二设备的第二请求, 所述第二请求包括第一服务的标识以及所述第一服务对应的第一输出信息, 所述第二请求与所述第一请求相关联, 所述第一请求为来自第一用户的所述第一服务的请求;
向所述第一设备发送第三请求, 所述第三请求包括所述第一服务的标识以及所述第一服务对应的第二输出信息, 所述第三请求与所述第二请求相关联。
17. 根据权利要求 16 所述的方法, 其特征在于, 所述方法还包括:
接收所述第一设备发送的来自第二设备的第五请求, 所述第五请求包括第二服务的标识以及所述第二服务对应的第三输出信息, 所述第五请求与第四请求相关联, 所述第四请求为来自第二用户的所述第二服务的请求;
向所述第一设备发送第六请求, 所述第六请求包括所述第二服务的标识以及所述第二服务对应的第四输出信息, 所述第六请求与所述第五请求相关联。
18. 根据权利要求 16 或 17 所述的方法, 其特征在于, 所述向所述第一设备发送第三请求之前, 还包括:
根据第二计算资源状态和第二条件对所述第二请求和与所述第二请求同时到达所述第三设备的至少一个第八请求进行组批次得到第二批次, 所述第八请求来自第二设备或第三设备;
接收所述第二设备发送的第一缓存信息;
根据所述第一缓存信息, 对所述第二批次进行第二推理计算, 得到所述第三请求和第二缓存信息。
19. 一种设备, 其特征在于, 包括处理器和存储器, 其中, 所述存储器用于存储计算机程序, 所述

计算机程序包括程序指令，当所述处理器运行所述程序指令时，使所述设备执行如权利要求 5-9 任一项所述方法的步骤，如权利要求 10-15 任一项所述方法的步骤，或如权利要求 16-18 任一项所述方法的步骤。

20. 一种可读存储介质，其特征在于，所述可读存储介质存储有程序，所述程序包括程序指令，当所述程序请求被设备运行时使所述设备执行如权利要求 5-9 任一项所述的方法，如权利要求 10-15 任一项所述的方法，或如权利要求 16-18 任一项所述的方法。