(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 112600882 B
(45)授权公告日 2022.03.08

(21)申请号 202011389606.0
(22)申请日 2020.12.01
(65)同一申请的已公布的文献号
申请公布号 CN 112600882 A
(43)申请公布日 2021.04.02
(73)专利权人 上海交通大学
地址 200240 上海市闵行区东川路800号
(72)发明人 李健 庄树隽 管海兵
(74)专利代理机构 上海旭诚知识产权代理有限公司 31220
代理人 郑立
(51)Int.Cl.
H04L 43/10(2022.01)
H04L 67/06(2022.01)
H04L 67/1074(2022.01)
H04L 67/63(2022.01)
H04L 69/30(2022.01)

(56)对比文件
CN 106663021 A,2017.05.10
CN 111314429 A,2020.06.19
CN 110865953 A,2020.03.06
CN 106537340 A,2017.03.22
WO 0114959 A2,2001.03.01

审查员 丛文

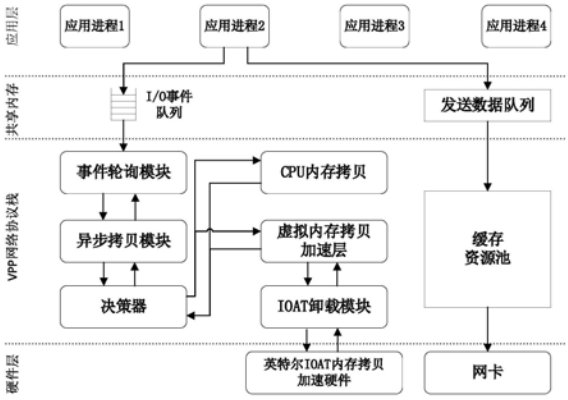
权利要求书1页 说明书4页 附图3页

(54)发明名称

一种基于共享内存通信模式的硬件加速方法

(57)摘要

本发明公开了一种基于共享内存通信模式的硬件加速方法,涉及网络协议领域,使用VPP作为用户态网络协议栈,NGINX应用用作网页服务器,通过共享内存的模式进行通信,其中协议栈包括异步拷贝模块,决策器和虚拟内存拷贝加速层。本发明能够在保证用户态协议栈和应用低耦合的情况下,减少通信中拷贝操作对于性能的影响,在大包请求较多的场景下仍能达到良好的性能。



CN 112600882 B

1.一种基于共享内存通信模式的硬件加速方法,使用VPP作为用户态网络协议栈,NGINX为网络应用,通过共享内存的模式进行通信,其特征在于,包括以下三个阶段:

提交阶段:分配发送缓存区来存储即将从共享内存中拷贝来的包,然后进行虚拟地址和物理地址间的翻译,并将拷贝的参数打包成一个拷贝请求,所述拷贝请求被下放到决策器上,由所述决策器决定交给CPU还是IOAT加速硬件,在所述决策器决定将所述拷贝请求交给所述IOAT加速硬件之后,再将卸载到所述IOAT加速硬件的所述拷贝请求放入等待缓存区进行暂存;

轮询阶段:从所述IOAT加速硬件描述队列中周期性地获取完成的拷贝请求;

后拷贝阶段:在完成了共享内存和协议栈之间的拷贝过程之后,网络传输层协议设置重传计时器,然后将拷贝完成的包送到下一个VPP协议处理节点。

2.如权利要求1所述的基于共享内存通信模式的硬件加速方法,其特征在于,所述决策器根据所述拷贝请求的数据大小决定将请求交给CPU还是所述IOAT加速硬件。

3.如权利要求2所述的基于共享内存通信模式的硬件加速方法,其特征在于,所述协议栈通过调用虚拟卸载函数接口实现将所述拷贝请求先卸载到虚拟加速器上,然后再由所述虚拟加速器将所述拷贝请求卸载到硬件加速器上,从而实现了协议栈逻辑和硬件驱动逻辑的分离。

4.如权利要求3所述的基于共享内存通信模式的硬件加速方法,其特征在于,所述虚拟加速器有面向多种错误情况的容错机制。

5.如权利要求4所述的基于共享内存通信模式的硬件加速方法,其特征在于,在拷贝请求过多超过了硬件加速器内缓存队列的长度时,所述虚拟加速器将多余的拷贝请求暂时交给CPU处理。

6.如权利要求4所述的基于共享内存通信模式的硬件加速方法,其特征在于,对于永久不可用的请求,所述虚拟加速器优先寻找另外的硬件加速器,并将未完成的拷贝请求一次性发送给新的硬件加速器,以此代替错误的硬件加速器。

7.如权利要求4所述的基于共享内存通信模式的硬件加速方法,其特征在于,对于永久不可用的请求,在找不到可用的其他硬件加速器时,所述虚拟加速器将拷贝请求交给CPU处理。

一种基于共享内存通信模式的硬件加速方法

技术领域

[0001] 本发明涉及网络协议领域,尤其涉及一种基于共享内存通信模式的硬件加速方法。

背景技术

[0002] 网络协议栈是计算机网络协议套件的一个具体软件实现。它负责将上层网络应用发送的数据打包成网络包并从网卡发出,并确保网络包在整个网络链路传输的稳定性和正确性。目前终端机器上的网络协议栈大都参照OSI七层模型在内核态实现。但这种传统的内核态网络协议栈自身存在一些低效性问题,比如频繁的上下文切换以及全局锁的竞争。随着近些年来网络流量的飞速增长,这些低效性的问题导致网络协议栈成了传输中主要的性能瓶颈。

[0003] 为了应对这些低效性的问题,研究者们开始寻找替代性的方法。RDMA(Remote Direct Memory Access)远程直接数据存取技术是其中一种代替方案,但是限制在于需要网卡支持RDMA。另外一种就是直接在用户态实现网络协议,即用户态网络协议栈,这种方案的好处在于避免了内核态和用户态频繁的切换开销,而且由于用户态开发的便利性,大大缩短了网络新特性的开发与部署的时间周期。

[0004] 用户态网络协议栈在实现之后,需要和上层的网络应用进行通信。目前有两种通信的模式,一种为LibOS模式,将协议栈以库的形式嵌入应用进程中,然后和应用通过函数调用的形式进行通信,另外一种是将网络协议栈启动为一个单独的进程,然后利用一块共享内存与应用进行异步通信。

[0005] LibOS模式优点是采用函数调用这种低开销的通信模式和RTC(Run-To-Completion)的线程模型,降低了通信成本,取得了较好的性能。但其缺点是首先是函数接口与应用紧耦合,开发和部署需要与应用同步,降低了新网络特性的开发和部署速度。其次,它可能造成一些安全隐患,比如一些恶意的应用能够攻击协议栈。最后,这种方法会与应用进程共用核计算资源,不能灵活分配。

[0006] 共享内存模式优点是低耦合性,开发与部署新网络特性的周期短,可以灵活地调度计算资源,支持协议栈的透明升级等高级功能。此外,共享内存作为中间层隔离了应用的恶意攻击。但其缺点是应用和协议栈的通信需要通过两次拷贝操作,在大包请求较多时,这两次拷贝操作会占用过多的CPU资源,严重影响协议栈吞吐量和延迟方面的性能。

[0007] 综上所述,两种用户态协议栈和应用的通信方式都有着各自的缺点,LibOS模式无法满足开发者快速开发以及支持高级功能的需求,而共享内存模式无法满足协议栈高性能的需求。

[0008] 因此,本领域的技术人员致力于开发一种基于共享内存通信模式的硬件加速方法。

发明内容

[0009] 有鉴于现有技术的上述缺陷，本发明所要解决的技术问题是如何让通信模式既保有和应用低耦合带来的一系列优势，又可以达到通信的高性能。

[0010] 为实现上述目的，本发明提供了一种基于共享内存通信模式的硬件加速方法，使用VPP作为用户态网络协议栈，NGINX应用用作网页服务器，通过共享内存的模式进行通信，所述协议栈包括异步拷贝模块，决策器和虚拟内存拷贝加速层。

[0011] 进一步地，利用英特尔的内存拷贝专用加速硬件IOAT，将共享内存和协议栈之间的拷贝由CPU卸载到所述IOAT专用硬件上。

[0012] 进一步地，所述异步拷贝模块的功能实现分三个步骤：

[0013] 步骤1、分配足够数量的发送缓存区来存储即将从共享内存中拷贝来的包，然后进行虚拟地址和物理地址间的翻译，并将拷贝的参数打包成一个拷贝请求，拷贝请求被下放到决策器上，由决策器决定交给CPU还是所述IOAT加速硬件，最后将卸载到IOAT的拷贝请求放入等待缓存区进行暂存；

[0014] 步骤2、从所述IOAT硬件描述队列中周期性地获取完成的拷贝请求；

[0015] 步骤3、拷贝过程完成后，网络传输层协议需要进行一些协议相关的处理，最后将拷贝完成的包送到下一个VPP协议处理节点。

[0016] 进一步地，所述决策器根据拷贝请求的数据大小决定将请求交给CPU还是IOAT硬件。

[0017] 进一步地，所述虚拟内存拷贝加速层将硬件卸载逻辑和协议栈的逻辑进行了隔离。

[0018] 进一步地，所述协议栈通过调用虚拟卸载接口实现拷贝请求卸载到硬件加速器上。

[0019] 进一步地，所述虚拟拷贝加速器有面向多种错误情况的容错机制。

[0020] 进一步地，在拷贝请求过多超过了硬件加速器队列的长度时，所述虚拟拷贝加速器将多余的拷贝请求暂时交给CPU处理。

[0021] 进一步地，对于永久不可用的请求，所述虚拟拷贝加速器优先寻找另外的硬件加速器，并将未完成的请求一次性发送给新的硬件加速器，以此代替错误的硬件加速器。

[0022] 进一步地，对于永久不可用的请求，在找不到可用的其他硬件加速器时，所述虚拟拷贝加速器将拷贝请求交给CPU处理。

[0023] 技术效果：

[0024] 能够在保证用户态协议栈和应用低耦合的情况下，减少通信中拷贝操作对于性能的影响，在大包请求较多的场景下仍达到良好的性能。

[0025] 以下将结合附图对本发明的构思、具体结构及产生的技术效果作进一步说明，以充分地了解本发明的目的、特征和效果。

附图说明

[0026] 图1是本发明的一个较佳实施例的总体流程框图；

[0027] 图2是本发明的一个较佳实施例的协议栈中各个操作所占用CPU的对比图；

[0028] 图3是本发明的一个较佳实施例的异步拷贝模块框架图；

[0029] 图4是本发明的一个较佳实施例的IOAT硬件和CPU的拷贝速度比较图；

[0030] 图5是本发明的一个较佳实施例的虚拟内存拷贝加速器层的架构图。

具体实施方式

[0031] 以下参考说明书附图介绍本发明的多个优选实施例，使其技术内容更加清楚和便于理解。本发明可以通过许多不同形式的实施例来得以体现，本发明的保护范围并非仅限于文中提到的实施例。

[0032] 首先使用VPP作为用户态网络协议栈，NGINX应用用作网页服务器，让它们通过共享内存的模式进行通信。图1为本发明的总体架构图，异步拷贝模块为VPP协议栈中嵌入的异步拷贝模块，决策器根据拷贝请求的数据大小决定将请求交给CPU还是IOAT硬件，本发明为了保证高通用性和可用性，通过虚拟拷贝加速层对硬件卸载逻辑和协议栈的逻辑进行了隔离。

[0033] 如图2所示，在这个场景下我们定量地分析出拷贝操作所占的CPU比例，实验表明：

[0034] (1)在客户端请求文件大小为4KB/8KB时，内存拷贝操作大概占20％-30％的CPU资源，但当文件大小超过32KB后，内存拷贝成了协议栈中主要的性能瓶颈，大概占用60％的CPU资源。

[0035] (2)在请求的文件大小逐步增大的过程图中，内存拷贝所占用的CPU资源线性增加。

[0036] (3)随着请求文件逐渐增大，内存拷贝开销增大，VPP用户态协议栈的吞吐量性能也从比内核态网络协议栈快40％退化成比内核态网络协议栈慢40％。

[0037] 如图3所示，提交阶段：分配足够大的发送缓存区用来存储即将从共享内存中拷贝来的包，然后进行虚拟地址和物理地址间的翻译，并将拷贝的参数打包成一个拷贝请求。拷贝请求被下放到决策器上，由决策器决定交给CPU还是IOAT加速硬件。最后将卸载到IOAT的拷贝请求放入等待缓存区进行暂存。轮询阶段：从IOAT硬件描述队列中周期性地获取完成的拷贝请求。后拷贝阶段：拷贝过程完成后，网络传输层协议需要进行一些协议相关的处理(比如设置重传计时器)，最后将拷贝完成的包送到下一个VPP协议处理节点。

[0038] 如图4所示，我们发现在1KB时IOAT拷贝的速度与CPU相接近。在请求文件小于1KB时，CPU拷贝速度较快，而在文件大于1KB时，IOAT拷贝速度较快。所以当拷贝数据大于1KB时，我们将请求卸载到IOAT硬件上处理，反之，我们将拷贝请求交给CPU处理。

[0039] 如图5所示，协议栈通过调用虚拟卸载接口实现拷贝请求卸载先卸载到虚拟加速器上，然后再由虚拟加速器卸载请求到到硬件加速器上，实现了协议栈逻辑和硬件驱动逻辑的分离。开发者可以在不了解上层协议栈逻辑的情况下，实现这些接口并将不同机器上或者未来的加速硬件绑定到VPP协议栈上，让本发明支持更多的硬件加速器。

[0040] 本发明在虚拟拷贝加速器这个模块实现了一个容错机制，面向多种错误情况：暂时不可用的情况，如拷贝请求过多超过了硬件加速器队列的长度，容错机制将多余的拷贝请求暂时交给CPU处理。永久不可用的请求，如IOAT硬件发生错误，或者由于拷贝地址非法而引起的通道错误，容错机制会优先寻找另外的硬件加速器，并将未完成的请求一次性发送给新的硬件加速器，以此代替错误的硬件加速器。若找不到可用的其他硬件加速器，则容错机制会将拷贝请求交给CPU处理。

[0041] 以上详细描述了本发明的较佳具体实施例。应当理解，本领域的普通技术无需创造性劳动就可以根据本发明的构思作出诸多修改和变化。因此，凡本技术领域中技术人员依本发明的构思在现有技术的基础上通过逻辑分析、推理或者有限的实验可以得到的技术方案，皆应在由权利要求书所确定的保护范围内。

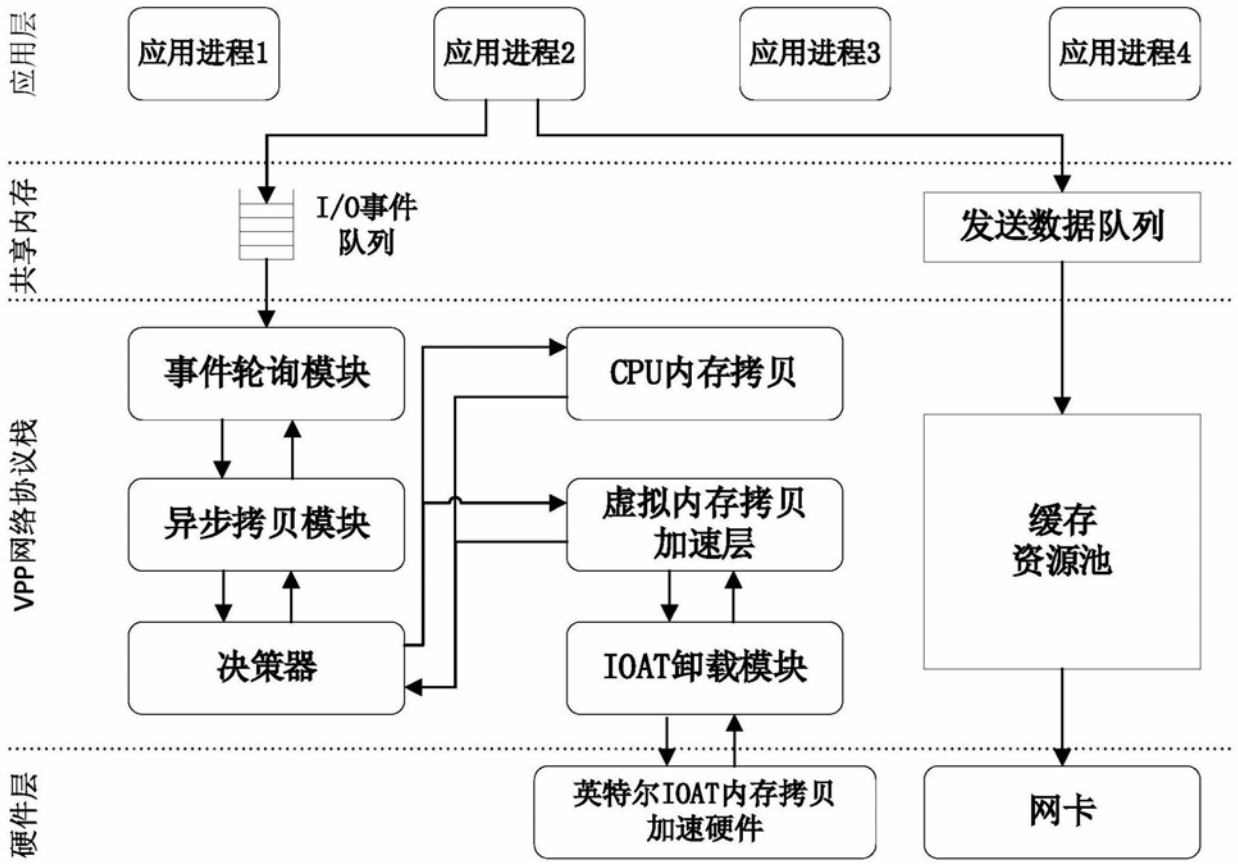

应用层
应用进程1
应用进程2
应用进程3
应用进程4
共享内存
I/O事件
队列
发送数据队列
VPP网络协议栈
事件轮询模块
CPU内存拷贝
异步拷贝模块
虚拟内存拷贝
加速层
缓存
资源池
决策器
IOAT卸载模块
硬件层
英特尔IOAT内存拷贝
加速硬件
网卡

图1

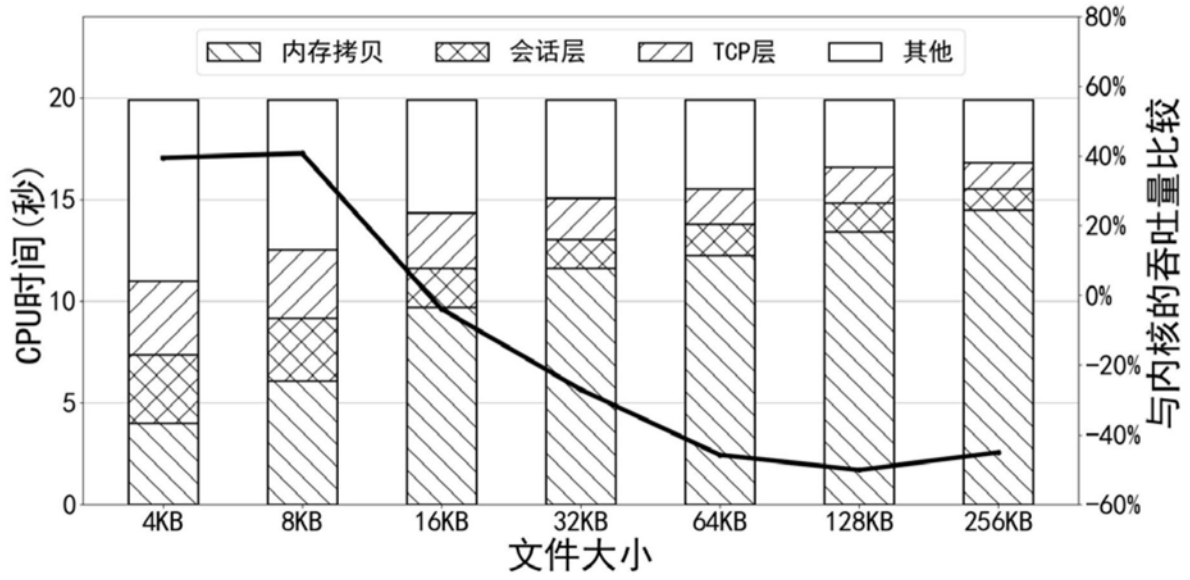

内存拷贝
会话层
TCP层
其他
CPU时间(秒)
与内核的吞吐量比较
文件大小
4KB
8KB
16KB
32KB
64KB
128KB
256KB
0
5
10
15
20
-60%
-40%
-20%
0%
20%
40%
60%
80%

图2

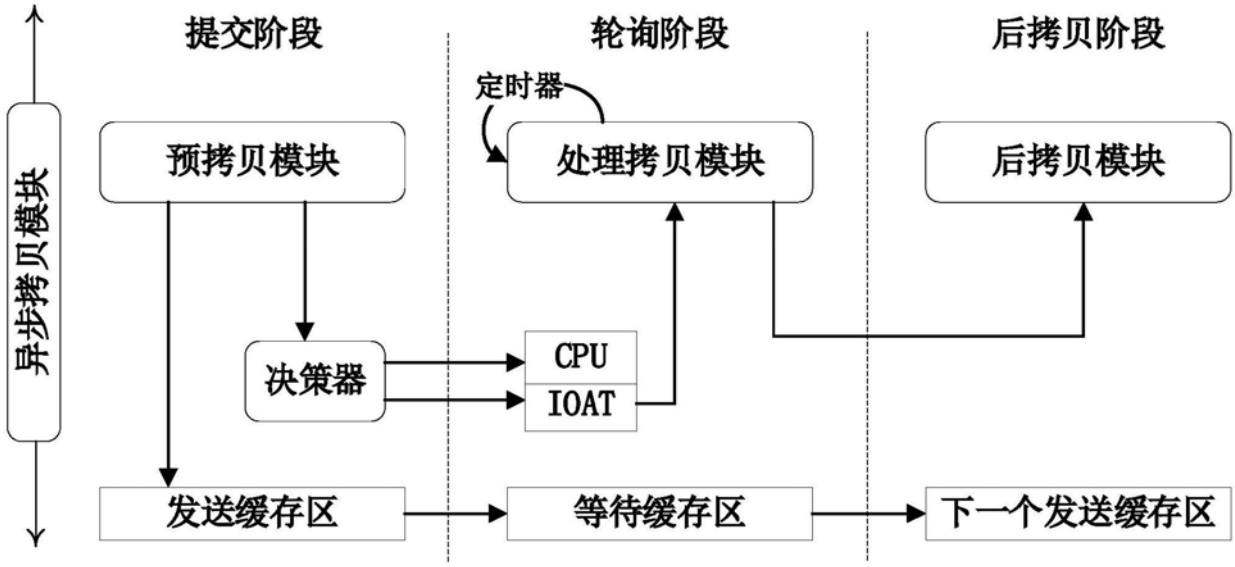

提交阶段
轮询阶段
后拷贝阶段
异步拷贝模块
定时器
预拷贝模块
处理拷贝模块
后拷贝模块
决策器
CPU
IOAT
发送缓存区
等待缓存区
下一个发送缓存区

图3

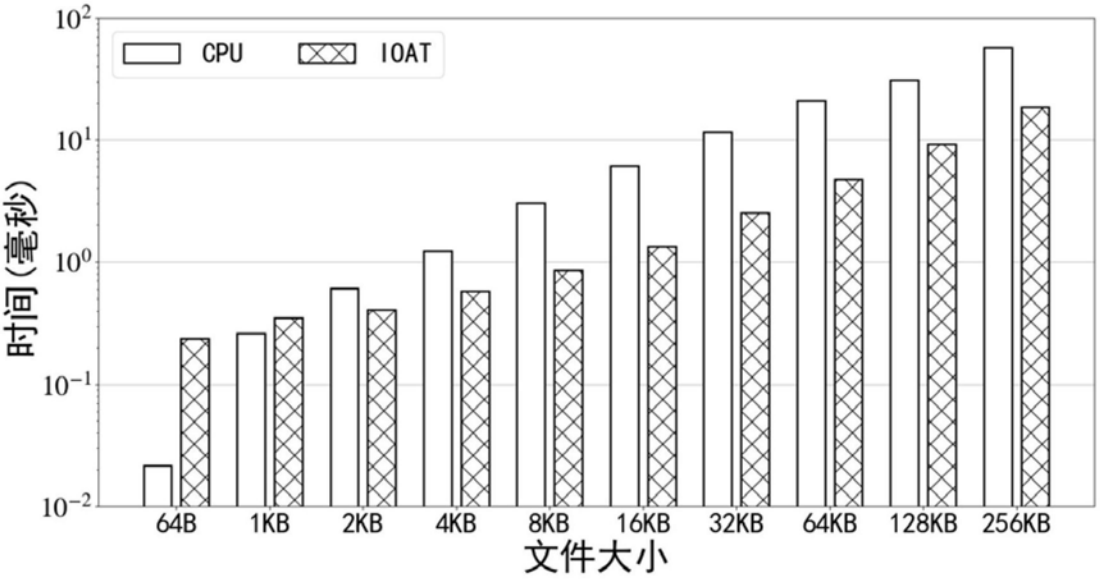

CPU
IOAT
时间(毫秒)
10^2
10^1
10^0
10^{-1}
10^{-2}
64B
1KB
2KB
4KB
8KB
16KB
32KB
64KB
128KB
256KB
文件大小

图4

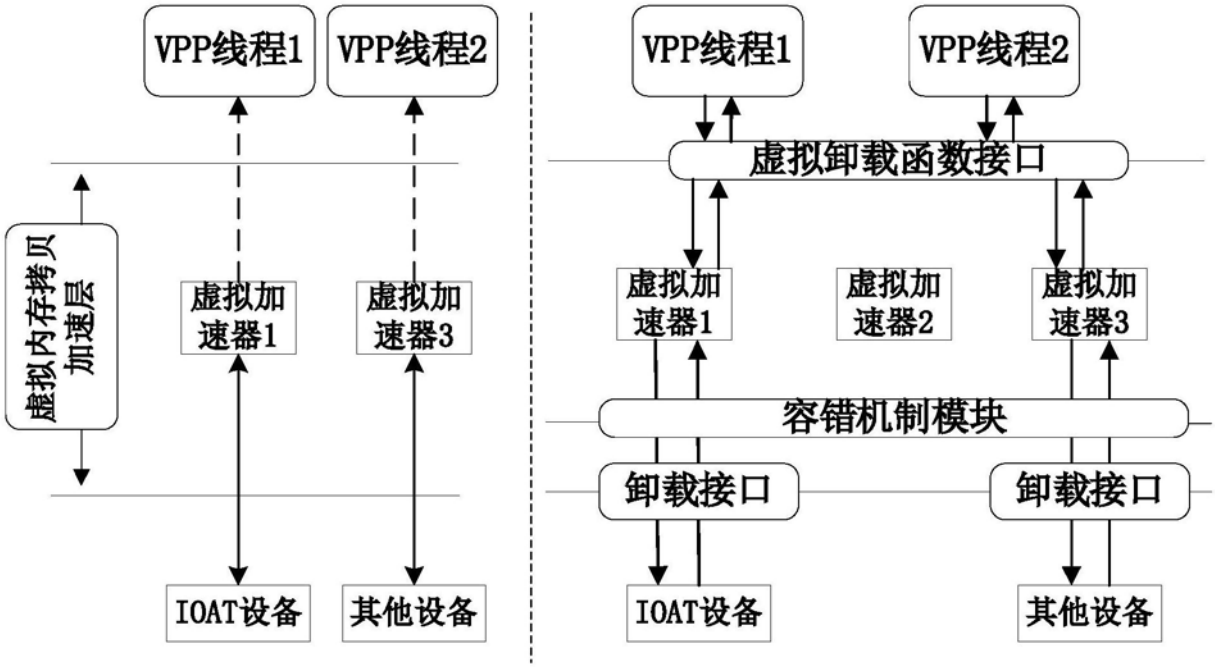

VPP线程1
VPP线程2
虚拟内存拷贝加速层
虚拟加速器1
虚拟加速器3
IOAT设备
其他设备
VPP线程1
VPP线程2
虚拟卸载函数接口
虚拟加速器1
虚拟加速器2
虚拟加速器3
容错机制模块
卸载接口
卸载接口
IOAT设备
其他设备

图5