



(12)发明专利

(10)授权公告号 CN 104407920 B

(45)授权公告日 2018.02.09

(21)申请号 201410816254.0

G06F 9/54(2006.01)

(22)申请日 2014.12.23

G06F 9/445(2018.01)

(65)同一申请的已公布的文献号

申请公布号 CN 104407920 A

(56)对比文件

CN 103780660 A,2014.05.07,

US 2011249010 A1,2011.10.13,

US 2013328890 A1,2013.12.12,

CN 101901042 A,2010.12.01,

CN 103902387 A,2014.07.02,

(43)申请公布日 2015.03.11

(73)专利权人 浪潮(北京)电子信息产业有限公司

地址 100085 北京市海淀区上地信息路2号  
2-1号C栋1层

审查员 宋秀之

(72)发明人 刘续东

(74)专利代理机构 北京安信方达知识产权代理有限公司 11262

代理人 王康 李丹

(51)Int.Cl.

G06F 9/46(2006.01)

权利要求书2页 说明书3页 附图3页

(54)发明名称

一种基于进程间通信的数据处理方法及系统

(57)摘要

本发明提供一种基于进程间通信的数据处理方法及系统,包括以下步骤:若图形处理器数据监听程序首次执行时,创建一个数据采集进程并由所述数据采集进程对所述图形处理器进行初始化并获取所述图形处理器的个数;所述数据采集进程为每个图形处理器分别创建一个数据采集线程并为每个数据采集线程分别传递一个图形处理器索引;所述数据采集进程为每个图形处理器分别创建一个服务线程,所述服务线程监听到图形处理器利用率数据请求后,从所述数据采集线程采集的图像处理器利用率数据中获取对应的图像处理器利用率数据,本发明缩短了获取GPU利用率数据的时间,从而提高了效率。

步骤201: 若图形处理器数据监听程序首次执行时,创建一个数据采集进程并由所述数据采集进程对所述图形处理器进行初始化并获取所述图形处理器的个数

步骤202: 所述数据采集进程为每个图形处理器分别创建一个数据采集线程并为每个数据采集线程分别传递一个图形处理器索引

步骤203: 所述数据采集进程为每个图形处理器分别创建一个服务线程,所述服务线程监听到图形处理器利用率数据请求后,从所述数据采集线程采集的图像处理器利用率数据中获取对应的图像处理器利用率数据

1. 一种基于进程间通信的数据处理方法,其特征在于,包括以下步骤:

若图形处理器数据监听程序首次执行时,创建一个数据采集进程并由所述数据采集进程对所述图形处理器进行初始化并获取所述图形处理器的个数;

所述数据采集进程为每个图形处理器分别创建一个数据采集线程并为每个数据采集线程分别传递一个图形处理器索引;

所述数据采集进程为每个图形处理器分别创建一个服务线程,所述服务线程监听到图形处理器利用率数据请求后,从所述数据采集线程采集的图像处理器利用率数据中获取对应的图像处理器利用率数据。

2. 根据权利要求1所述的方法,其特征在于:若图形处理器数据监听程序首次执行时,创建一个在后台运行的数据采集进程并将创建的数据采集进程的进程号记录在文件中,同时,所述数据采集进程对所述图形处理器进行初始化并获取所述图形处理器的个数。

3. 根据权利要求1所述的方法,其特征在于:所述数据采集线程根据获取的图像处理器索引,获取所述图像处理器利用率数据;其中,所述数据采集线程实时地或周期性获取所述图像处理器利用率数据并进行更新。

4. 根据权利要求1所述的方法,其特征在于:所述数据采集进程为每个图形处理器分别创建一个数据采集线程后,为每个图形处理器分别创建一个服务线程,对图形处理器利用率数据请求进行监听;

所述服务线程在监听到图形处理器利用率数据请求后,从所述数据采集线程采集的图像处理器利用率数据中获取对应的图像处理器利用率数据。

5. 根据权利要求4所述的方法,其特征在于:所述服务线程创建一个进程间通信套接字UNIX socket,对图形处理器利用率数据请求进行监听。

6. 根据权利要求1所述的方法,其特征在于:所述图像处理器利用率数据请求中携带图像处理器索引信息。

7. 根据权利要求1所述的方法,其特征在于:判断图形处理器数据监听程序是否为首次执行的方式为:检查记录所述数据采集进程对应的进程号的文件,若所述文件存在并且在/proc/路径下存在以所述进程号命名的目录,则所述图形处理器数据监听程序为非首次执行,否则为首次执行。

8. 根据权利要求1所述的方法,其特征在于,若图形处理器数据监听程序为非首次执行时,服务线程监听到图形处理器利用率数据请求后,从数据采集线程采集的图像处理器利用率数据中获取对应的图像处理器利用率数据。

9. 一种基于进程间通信的数据处理系统,其特征在于,包括数据采集进程管理模块、数据管理模块、请求处理模块;其中,所述数据采集进程管理模块通过所述数据管理模块与所述请求处理模块相连;

数据采集进程管理模块,用于图形处理器数据监听程序首次执行时,创建一个数据采集进程并由所述数据采集进程对所述图形处理器进行初始化并获取所述图形处理器的个数;还用于为每个图形处理器分别创建一个数据采集线程并为每个数据采集线程分别传递一个图形处理器索引;还用于为每个图形处理器分别创建一个服务线程;

数据管理模块,用于数据采集线程根据获取的图像处理器索引,获取所述图像处理器利用率数据;

请求处理模块,用于服务线程监听到图形处理器利用率数据请求后,从所述数据管理模块采集的图像处理器利用率数据中获取对应的图像处理器利用率数据。

10. 根据权利要求9所述的系统,其特征在于,还包括输出管理模块;其中,所述输出管理模块与所述请求处理模块相连;

所述输出管理模块,用于从所述请求处理模块中获取的图像处理器利用率数据后,进行打印输出。

## 一种基于进程间通信的数据处理方法及系统

### 技术领域

[0001] 本发明属于数据通信领域,尤其涉及一种基于进程间通信的数据处理方法及系统。

### 背景技术

[0002] 由于在获取GPU(Graphic Processing Unit,图形处理器)利用率数据时,需要首先对GPU进程初始化,而该初始化操作非常耗时;传统的获取GPU利用率数据的方法,在每次获取数据时,都需要进行一次初始化,造成了效率的降低。

### 发明内容

[0003] 本发明提供一种基于进程间通信的数据处理方法及系统,以解决上述问题。

[0004] 本发明还提供一种基于进程间通信的数据处理方法,包括以下步骤:

[0005] 若图形处理器数据监听程序首次执行时,创建一个数据采集进程并由所述数据采集进程对所述图形处理器进行初始化并获取所述图形处理器的个数;

[0006] 所述数据采集进程为每个图形处理器分别创建一个数据采集线程并为每个数据采集线程分别传递一个图形处理器索引;

[0007] 所述数据采集进程为每个图形处理器分别创建一个服务线程,所述服务线程监听到图形处理器利用率数据请求后,从所述数据采集线程采集的图像处理器利用率数据中获取对应的图像处理器利用率数据。

[0008] 本发明还提供一种基于进程间通信的数据处理系统,包括数据采集进程管理模块、数据管理模块、请求处理模块;其中,所述数据采集进程管理模块通过所述数据管理模块与所述请求处理模块相连;

[0009] 数据采集进程管理模块,用于图形处理器数据监听程序首次执行时,创建一个数据采集进程并由所述数据采集进程对所述图形处理器进行初始化并获取所述图形处理器的个数;还用于为每个图形处理器分别创建一个数据采集线程并为每个数据采集线程分别传递一个图形处理器索引;还用于为每个图形处理器分别创建一个服务线程;

[0010] 数据管理模块,用于数据采集线程根据获取的图像处理器索引,获取所述图像处理器利用率数据;

[0011] 请求处理模块,用于服务线程监听到图形处理器利用率数据请求后,从所述数据管理模块采集的图像处理器利用率数据中获取对应的图像处理器利用率数据。

[0012] 相较于先前技术,根据本发明提供一种基于进程间通信的数据处理方法及系统,缩短了获取GPU利用率数据的时间,从而提高了效率。

### 附图说明

[0013] 此处所说明的附图用来提供对本发明的进一步理解,构成本申请的一部分,本发明的示意性实施例及其说明用于解释本发明,并不构成对本发明的不当限定。在附图中:

- [0014] 图1所示为本发明实施例1的基于进程间通信的数据处理方法流程图；  
[0015] 图2所示为本发明实施例2的基于进程间通信的数据处理方法流程图；  
[0016] 图3所示为本发明实施例3的基于进程间通信的数据处理系统结构图；  
[0017] 图4所示为本发明实施例4的基于进程间通信的数据处理系统结构图。

### 具体实施方式

[0018] 下文中将参考附图并结合实施例来详细说明本发明。需要说明的是，在不冲突的情况下，本申请中的实施例及实施例中的特征可以相互组合。

[0019] 图1所示为本发明实施例1的基于进程间通信的数据处理方法流程图，包括以下步骤：

[0020] 步骤101：若GPU数据监听程序首次执行时，创建一个在后台运行的数据采集进程并将创建的数据采集进程的进程号记录在文件中，同时，所述数据采集进程对GPU进行初始化并获取所述GPU的个数；若GPU数据监听程序为非首次执行，则执行步骤104；

[0021] GPU数据监听程序每次执行时首先判断是否为首次执行，判断GPU数据监听程序是否为首次执行的方式为：检查记录所述数据采集进程对应的进程号的文件，若所述文件存在并且在/proc/路径下存在以所述进程号命名的目录，则所述GPU数据监听程序为非首次执行，否则为首次执行。

[0022] 步骤102：所述数据采集进程获取所述图形处理器的个数后，为每个图形处理器分别创建一个数据采集线程并为每个数据采集线程分别传递一个图形处理器索引；

[0023] 所述数据采集线程根据获取的图像处理器索引，获取所述图像处理器利用率数据。

[0024] 所述数据采集线程实时地或周期性获取所述图像处理器利用率数据并进行更新。

[0025] 周期可以根据实际情况进行灵活设定，例如：0.5秒。

[0026] 步骤103：所述数据采集进程为每个图形处理器分别创建一个数据采集线程后，为每个图形处理器分别创建一个服务线程，对图形处理器利用率数据请求进行监听；

[0027] 所述服务线程创建一个进程间通信套接字UNIX socket，对图形处理器利用率数据请求进行监听。

[0028] 步骤104：所述服务线程在监听到图形处理器利用率数据请求后，从所述数据采集线程采集的图像处理器利用率数据中获取对应的图像处理器利用率数据并发送给请求方；其中，所述图像处理器利用率数据请求中携带图像处理器索引信息。

[0029] 备注：

[0030] 1、GPU数据监听程序以后每次执行时，从所述数据采集进程处获取当前GPU利用率数据；

[0031] 2、GPU数据监听程序以后每次执行时，与所述服务线程通信，向所述服务线程发送获取GPU利用率数据请求并将返回的结果打印输出。

[0032] 图2所示为本发明实施例2的基于进程间通信的数据处理方法流程图，包括以下步骤：

[0033] 步骤201：若图形处理器数据监听程序首次执行时，创建一个数据采集进程并由所述数据采集进程对所述图形处理器进行初始化并获取所述图形处理器的个数；

[0034] 步骤202:所述数据采集进程为每个图形处理器分别创建一个数据采集线程并为每个数据采集线程分别传递一个图形处理器索引;

[0035] 步骤203:所述数据采集进程为每个图形处理器分别创建一个服务线程,所述服务线程监听到图形处理器利用率数据请求后,从所述数据采集线程采集的图像处理器利用率数据中获取对应的图像处理器利用率数据。

[0036] 图3所示为本发明实施例3的基于进程间通信的数据处理系统结构图,包括数据采集进程管理模块、数据管理模块、请求处理模块;其中,所述数据采集进程管理模块通过所述数据管理模块与所述请求处理模块相连;

[0037] 数据采集进程管理模块,用于图形处理器数据监听程序首次执行时,创建一个数据采集进程并由所述数据采集进程对所述图形处理器进行初始化并获取所述图形处理器的个数;还用于为每个图形处理器分别创建一个数据采集线程并为每个数据采集线程分别传递一个图形处理器索引;还用于为每个图形处理器分别创建一个服务线程;

[0038] 数据管理模块,用于数据采集线程根据获取的图像处理器索引,获取所述图像处理器利用率数据;

[0039] 请求处理模块,用于服务线程监听到图形处理器利用率数据请求后,从所述数据管理模块采集的图像处理器利用率数据中获取对应的图像处理器利用率数据。

[0040] 图4所示为本发明实施例4的基于进程间通信的数据处理系统结构图,在图3的基础上增加了“输出管理模块”;其中,所述输出管理模块与所述请求处理模块相连;

[0041] 所述输出管理模块,用于从所述请求处理模块中获取的图像处理器利用率数据后,进行打印输出。

[0042] 相较于先前技术,根据本发明提供一种基于进程间通信的数据处理方法及系统,缩短了获取GPU利用率数据的时间,从而提高了效率。

[0043] 以上所述仅为本发明的优选实施例而已,并不用于限制本发明,对于本领域的技术人员来说,本发明可以有各种更改和变化。凡在本发明的精神和原则之内,所作的任何修改、等同替换、改进等,均应包含在本发明的保护范围之内。

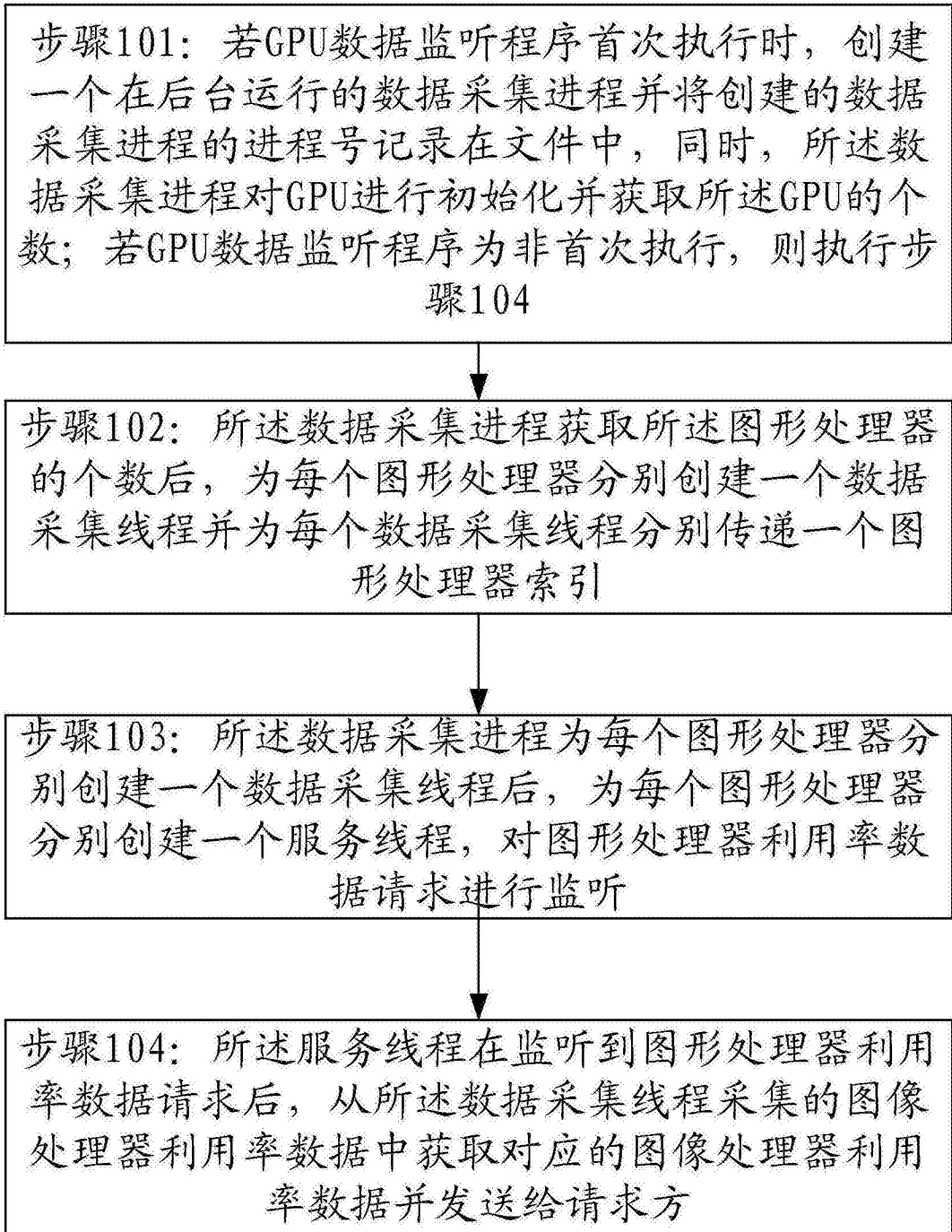


图1

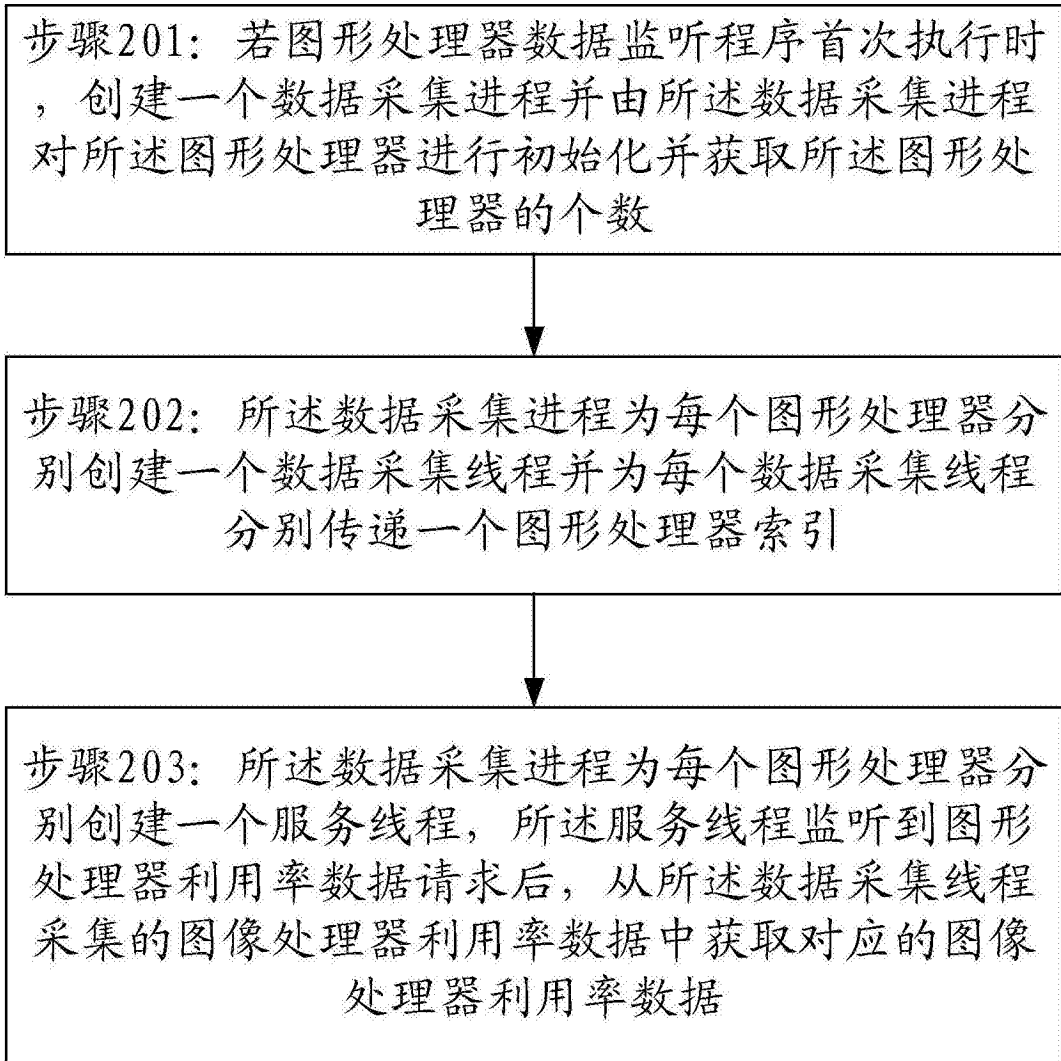


图2

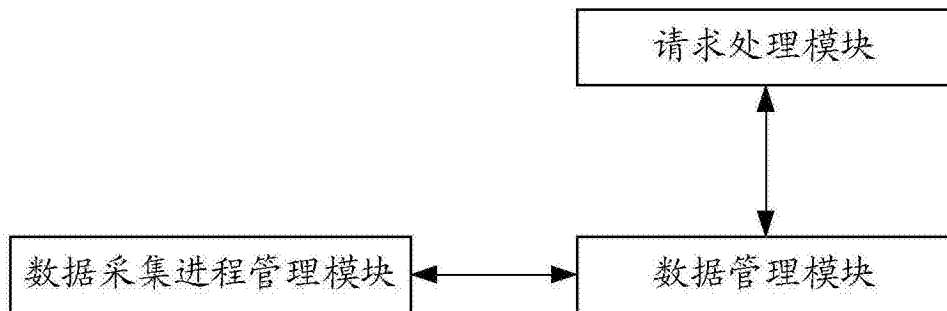


图3



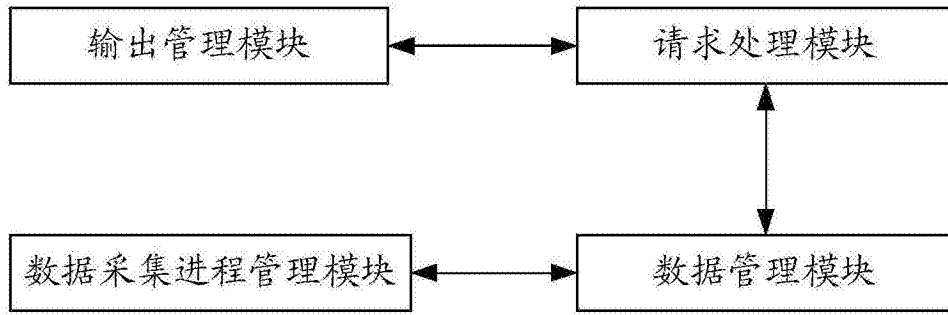


图4