



(12)发明专利

(10)授权公告号 CN 104111897 B

(45)授权公告日 2017.06.13

(21)申请号 201310131430.2

US 2011265090 A1, 2011.10.27,

(22)申请日 2013.04.16

CN 102907055 A, 2013.01.30,

(65)同一申请的已公布的文献号

Stefan Lankes等. The Path to MetalSVM

申请公布号 CN 104111897 A

- Shared Virtual Memory for the SCC.

(43)申请公布日 2014.10.22

《Proceedings of the 4th Manycore

(73)专利权人 华为技术有限公司

Applications Research Community (MARC)

地址 518129 广东省深圳市龙岗区坂田华为总部办公楼

Symposium》.2011, 第2011年卷第1-8页.

审查员 吴海旋

(72)发明人 林擎天 史经浩 王卓立 朱望斌

(51)Int.Cl.

G06F 12/0806(2016.01)

权利要求书8页 说明书28页 附图7页

G06F 11/14(2006.01)

(56)对比文件

CN 1818874 A, 2006.08.16,

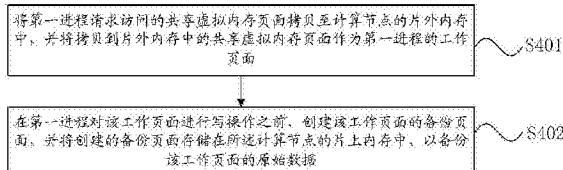
WO 2011090515 A2, 2011.07.28,

(54)发明名称

一种数据处理方法、装置及计算机系统

(57)摘要

本发明实施例提供一种计算节点上的数据共享方法及装置,包括:将第一进程请求访问的共享虚拟内存页面拷贝至计算节点的片外内存中,并将拷贝到片外内存中的共享虚拟内存页面作为第一进程的工作页面;在第一进程对该工作页面进行写操作之前,在计算节点的片上内存中,创建该工作页面的备份页面,以备份该工作页面的原始数据;本发明实施例通过利用计算节点的可编程片上内存,在对工作页面进行写操作之前,将页面数据在片上内存中备份,以保证多个进程在对共享虚拟内存页面进行操作时的数据一致性,同时尽可能少的访问片外内存,提高程序的速度。



1. 一种数据处理方法,其特征在于,包括:

将第一进程请求访问的共享虚拟内存页面拷贝至计算节点的片外内存中,并将拷贝到所述片外内存中的共享虚拟内存页面作为第一进程的工作页面;其中,所述共享虚拟内存页面为所述第一进程所属应用程序的共享虚拟内存中的虚拟内存页面,所述应用程序运行在所述计算节点上;

在所述第一进程对所述工作页面进行写操作之前,创建所述工作页面的备份页面,并将创建的所述备份页面存储在所述计算节点的片上内存中,以备份所述工作页面的原始数据;

其中,在将创建的所述备份页面存储在所述计算节点的片上内存之前,还包括:

判断所述片上内存的剩余空间是否小于第一阈值,如果所述片上内存的剩余空间小于第一阈值,则触发所述第一进程将所述第一进程的M个工作页面中被修改的内容同步更新到所述M个工作页面所对应的M个共享虚拟内存页面中,并将所述M个工作页面在所述片上内存中的备份页面所占用的空间释放;如果所述片上内存的剩余空间大于或等于第一阈值,则执行所述将创建的所述备份页面存储在所述计算节点的片上内存中的步骤,所述共享虚拟内存页面的个数为N,N为大于或等于1的正整数,所述第一进程的工作页面的个数为M,M为大于或等于1的正整数。

2. 根据权利要求1所述的方法,其特征在于,在创建所述工作页面的备份页面之前,还包括:在所述计算节点的片上内存中,预先给所述第一进程分配特定大小的片上存储区;

所述将创建的所述备份页面存储在所述计算节点的片上内存中,包括:

将创建的所述备份页面存储在为所述第一进程预先分配的片上存储区中。

3. 根据权利要求2所述的方法,其特征在于,所述判断所述片上内存的剩余空间是否小于第一阈值,如果所述片上内存的剩余空间小于第一阈值,则触发所述第一进程将所述第一进程的M个工作页面中被修改的内容同步更新到所述M个工作页面所对应的M个共享虚拟内存页面中,并将所述M个工作页面在所述片上内存中的备份页面所占用的空间释放;如果所述片上内存的剩余空间大于或等于第一阈值,则执行所述将创建的所述备份页面存储在所述计算节点的片上内存中的步骤,包括:

判断所述第一进程的片上存储区的剩余空间是否小于创建的所述备份页面的大小,或者小于第二阈值,如果所述第一进程的片上存储区的剩余空间小于创建的所述备份页面的大小,或者小于第二阈值,则触发所述第一进程将所述第一进程的M个工作页面中被修改的内容同步更新到所述M个工作页面所对应的M个共享虚拟内存页面中,并将所述M个工作页面在所述片上内存中的备份页面所占用的空间释放;如果所述第一进程的片上存储区的剩余空间大于或等于创建的所述备份页面的大小,或者小于第二阈值,则执行所述将创建的所述备份页面存储在所述计算节点的片上内存中的步骤。

4. 根据权利要求1所述的方法,其特征在于,所述应用程序的各个进程共享所述计算节点的片上内存;

所述第一阈值为创建的所述备份页面的大小。

5. 根据权利要求4所述的方法,其特征在于,还包括:如果所述计算节点的片上内存的剩余空间小于创建的所述备份页面的大小,则触发所述应用程序中,除所述第一进程之外的其它至少一个进程将所述其它至少一个进程的各个工作页面中被修改的内容同步更新

到所述各个工作页面所对应的各个共享虚拟内存页面中，并将所述各个工作页面在所述片上内存中的备份页面所占用的空间释放。

6. 根据权利要求1所述的方法，其特征在于，所述计算节点的片上内存包括多个独立的存储区域，所述应用程序的所有进程被划分为至少一个进程组，每个所述进程组中的各个进程共享所述多个独立的存储区域中的一个存储区域，以作为该进程组的片上公共缓存区；

所述将创建的所述备份页面存储在所述计算节点的片上内存中，包括：

将创建的所述备份页面存储在为所述第一进程所在进程组的片上公共缓存区中；

所述判断所述片上内存的剩余空间是否小于第一阈值，如果所述片上内存的剩余空间小于第一阈值，则触发所述第一进程将所述第一进程的M个工作页面中被修改的内容同步更新到所述M个工作页面所对应的M个共享虚拟内存页面中，并将所述M个工作页面在所述片上内存中的备份页面所占用的空间释放；如果所述片上内存的剩余空间大于或等于第一阈值，则执行所述将创建的所述备份页面存储在所述计算节点的片上内存中的步骤，包括：

判断所述第一进程所在的进程组的片上公共缓存区的剩余空间是否小于创建的所述备份页面的大小，或者小于第二阈值，如果所述第一进程所在的进程组的片上公共缓存区的剩余空间是否小于创建的所述备份页面的大小，或者小于所述第二阈值时，则触发所述第一进程将所述第一进程的M个工作页面中被修改的内容同步更新到所述M个工作页面所对应的M个共享虚拟内存页面中，并将所述M个工作页面在所述片上内存中的备份页面所占用的空间释放；如果所述第一进程所在的进程组的片上公共缓存区的剩余空间大于或等于创建的所述备份页面的大小，或者小于所述第二阈值时，则执行所述将创建的所述备份页面存储在所述计算节点的片上内存中的步骤。

7. 根据权利要求6所述的方法，其特征在于，还包括：如果所述第一进程所在的进程组的片上公共缓存区的剩余空间小于创建的所述备份页面的大小，或者小于所述第二阈值，则触发所述第一进程所在进程组中，除所述第一进程之外的其它进程将所述其它进程的各个工作页面中被修改的内容同步更新到所述各个工作页面所对应的各个共享虚拟内存页面中，并将所述各个工作页面在所述片上内存中的备份页面所占用的空间释放。

8. 根据权利要求1-7任一项所述的方法，其特征在于，所述将所述第一进程的M个工作页面中被修改的内容同步更新到所述M个工作页面所对应的M个共享虚拟内存页面中，包括：将所述第一进程的M个工作页面中的每一个工作页面与该工作页面在片上内存中的备份页面进行比较，生成用于记录两者的数据内容差异的日志文件，根据所述日志文件将所述第一进程对所述M个工作页面修改的内容更新到所述M个工作页面所对应的M个共享虚拟内存页面中。

9. 根据权利要求1-7任一项所述的方法，其特征在于，还包括：在创建所述工作页面的备份页面之前，在所述计算节点的片上内存中查找是否存在所述工作页面的备份页面；

所述创建所述工作页面的备份页面，包括：如果没有在所述计算节点的片上内存中查找到所述工作页面的备份页面，则创建所述工作页面的备份页面。

10. 根据权利要求9所述的方法，其特征在于，所述计算节点的片上内存中保存有备份页面信息表，其中，所述备份页面信息表包含有所述片上内存中所有备份页面的元数据信息，每一个备份页面的元数据信息包括：所述每一个备份页面的页号和版本号，其中，所述

每一个备份页面的页号和版本号分别与所述每一个备份页面所对应的工作页面的页号和版本号相同；

所述在所述计算节点的片上内存中查找是否存在所述工作页面的备份页面，包括：

在所述备份页面信息表中查找是否有页号和版本号分别与所述工作页面的页号以及版本号相同的元数据信息；

所述创建所述工作页面的备份页面，包括：如果没有在所述备份页面信息表中查找到页号和版本号分别与所述工作页面的页号以及版本号相同的元数据信息，则创建所述工作页面的备份页面。

11. 根据权利要求10所述的方法，其特征在于，在所述创建所述工作页面的备份页面之后，还包括：

将所述工作页面的页号以及版本号分别作为创建的所述备份页面的页号和版本号记录到所述备份页面信息表。

12. 根据权利要求9所述的方法，其特征在于，所述片上内存包括多个独立的存储区域，所述应用程序的所有进程被划分为至少一个进程组，每个所述进程组中的各个进程共享所述多个独立的存储区域中的一个存储区域，以作为该进程组的片上公共缓存区，且每个所述进程组有一个单独的备份页面信息表；每一个进程组的备份页面信息表包含有所述进程组中所有进程的所有备份页面的元数据信息，每一个备份页面的元数据信息包括：所述每一个备份页面的页号和版本号，其中，所述每一个备份页面的页号和版本号分别与所述每一个备份页面所对应的工作页面的页号和版本号相同；

所述在所述计算节点的片上内存中查找是否存在所述工作页面的备份页面，包括：

在所述第一进程所在进程组的备份页面信息表中查找是否有页号和版本号分别与所述工作页面的页号以及版本号相同的元数据信息；

所述创建所述工作页面的备份页面，包括：如果没有在所述第一进程所在进程组的备份页面信息表中查找到页号和版本号分别与所述工作页面的页号以及版本号相同的元数据信息，则创建所述工作页面的备份页面。

13. 根据权利要求12所述的方法，其特征在于，在所述创建所述工作页面的备份页面之后，还包括：

将所述工作页面的页号以及版本号分别作为创建的所述备份页面的页号和版本号记录到所述第一进程所在进程组的备份页面信息表。

14. 一种数据处理方法，其特征在于，包括：

将第一进程请求访问的共享虚拟内存页面拷贝至计算节点的片外内存中，并将拷贝到所述片外内存中的共享虚拟内存页面作为所述第一进程的工作页面；其中，所述共享虚拟内存页面为所述第一进程所属应用程序的共享虚拟内存中的虚拟内存页面，所述应用程序运行在所述计算节点上；

在所述第一进程对所述工作页面进行写操作之前，在所述计算节点的片上内存中查找是否存在所述工作页面的备份页面，如果查找到，则确定查找到的所述备份页面为所述工作页面的备份页面，其中，所述备份页面保存有所述工作页面中的原始数据；

如果没有查找到，则创建所述工作页面的备份页面，并将创建的所述备份页面存储在所述计算节点的片上内存中；

其中,在将创建的所述备份页面存储在所述计算节点的片上内存之前,还包括:

判断所述片上内存的剩余空间是否小于第一阈值,如果是,则触发所述第一进程将所述第一进程的M个工作页面中被修改的内容同步更新到所述M个工作页面所对应的M个共享虚拟内存页面中,并将所述M个工作页面在所述片上内存中的备份页面所占用的空间释放;如果否,则执行所述将创建的所述备份页面存储在所述计算节点的片上内存中的步骤,所述共享虚拟内存页面的个数为N,N为大于或等于1的正整数,所述第一进程的工作页面的个数为M,M为大于或等于1的正整数。

15.根据权利要求14所述的方法,其特征在于,所述计算节点的片上内存中保存有备份页面信息表,其中,所述备份页面信息表包含有所述片上内存中所有备份页面的元数据信息,每一个备份页面的元数据信息包括:所述每一个备份页面的页号和版本号,其中,所述每一个备份页面的页号和版本号分别与所述每一个备份页面所对应的工作页面的页号和版本号相同;

所述在所述计算节点的片上内存中查找是否存在所述工作页面的备份页面,如果查找到,则确定查找到的所述备份页面为所述工作页面的备份页面,包括:

在所述备份页面信息表中查找是否有页号和版本号分别与所述工作页面的页号以及版本号相同的元数据信息,如果查找到,则确定查找到的所述元数据信息所对应的备份页面为所述工作页面的备份页面。

16.根据权利要求14所述的方法,其特征在于,所述片上内存包括多个独立的存储区域,所述应用程序的所有进程被划分为至少一个进程组,每个所述进程组中的各个进程共享所述多个独立的存储区域中的一个存储区域,以作为该进程组的片上公共缓存区,且每个所述进程组有一个单独的备份页面信息表;每一个进程组的备份页面信息表包含有所述进程组中所有进程的所有备份页面的元数据信息,每一个备份页面的元数据信息包括:所述每一个备份页面的页号和版本号,其中,所述每一个备份页面的页号和版本号分别与所述每一个备份页面所对应的工作页面的页号和版本号相同;

所述在所述计算节点的片上内存中查找是否存在所述工作页面的备份页面,如果查找到,则确定查找到的所述备份页面为所述工作页面的备份页面,包括:

在所述第一进程所在进程组的备份页面信息表中查找是否有页号和版本号分别与所述工作页面的页号以及版本号相同的元数据信息,如果查找到,则确定查找到的所述元数据信息所对应的备份页面为所述工作页面的备份页面。

17.一种数据处理装置,其特征在于,包括:

拷贝单元,用于将第一进程请求访问的共享虚拟内存页面拷贝至计算节点的片外内存中,并将拷贝到所述片外内存中的共享虚拟内存页面作为所述第一进程的工作页面;其中,所述共享虚拟内存页面为所述第一进程所属应用程序的共享虚拟内存中的虚拟内存页面,所述应用程序运行在所述计算节点上;

备份单元,用于在所述第一进程对所述工作页面进行写操作之前,创建所述工作页面的备份页面,并将创建的所述备份页面存储在所述计算节点的片上内存中,以备份所述工作页面的原始数据;

判断单元,用于在所述备份单元将创建的所述备份页面存储在所述计算节点的片上内存之前,判断所述片上内存的剩余空间是否小于第一阈值;

触发单元,用于在所述判断单元判断出所述片上内存的剩余空间小于第一阈值时,触发所述第一进程将所述第一进程的M个工作页面中被修改的内容同步更新到所述M个工作页面所对应的M个共享虚拟内存页面中,并将所述M个工作页面在所述片上内存中的备份页面所占用的空间释放;

其中,所述共享虚拟内存页面的个数为N,N为大于或等于1的正整数;所述第一进程的工作页面的个数为M,M为大于或等于1的正整数。

18.根据权利要求17所述的数据处理装置,其特征在于,还包括:

内存分配单元,用于在所述计算节点的片上内存中,预先给所述第一进程所属的应用程序的各个进程分配特定大小的片上存储区;

所述备份单元,具体用于将创建的所述备份页面存储在所述内存分配单元为所述第一进程预先分配的片上存储区中。

19.根据权利要求18所述的数据处理装置,其特征在于,所述判断单元,具体用于:

判断所述第一进程的片上存储区的剩余空间是否小于创建的所述备份页面的大小,或者小于第二阈值;

所述触发单元,具体用于在所述判断单元判断出所述第一进程的片上存储区的剩余空间小于创建的所述备份页面的大小,或者小于第二阈值时,触发所述第一进程将所述第一进程的M个工作页面中被修改的内容同步更新到所述M个工作页面所对应的M个共享虚拟内存页面中,并将所述M个工作页面在所述片上内存中的备份页面所占用的空间释放。

20.根据权利要求17所述的数据处理装置,其特征在于,所述应用程序的各个进程共享所述计算节点的片上内存;

所述判断单元,具体用于:判断所述计算节点的片上内存的剩余空间是否小于创建的所述备份页面的大小;

所述触发单元,具体用于在所述判断单元判断出所述计算节点的片上内存的剩余空间小于创建的所述备份页面的大小时,触发所述第一进程将所述第一进程的M个工作页面中被修改的内容同步更新到所述M个工作页面所对应的M个共享虚拟内存页面中,并将所述M个工作页面在所述片上内存中的备份页面所占用的空间释放。

21.根据权利要求20所述的数据处理装置,其特征在于,所述触发单元,还用于:在所述判断单元判断出所述计算节点的片上内存的剩余空间小于创建的所述备份页面的大小时,触发所述应用程序中,除所述第一进程之外的其它至少一个进程将所述其它至少一个进程的各个工作页面中被修改的内容同步更新到所述各个工作页面所对应的各个共享虚拟内存页面中,并将所述各个工作页面在所述片上内存中的备份页面所占用的空间释放。

22.根据权利要求17所述的数据处理装置,其特征在于,所述计算节点的片上内存包括多个独立的存储区域,所述应用程序的所有进程被划分为至少一个进程组,每个所述进程组中的各个进程共享所述多个独立的存储区域中的一个存储区域,以作为该进程组的片上公共缓存区;

所述备份单元,具体用于将创建的所述备份页面存储在为所述第一进程所在进程组的片上公共缓存区中;

所述判断单元,具体用于:判断所述第一进程所在的进程组的片上公共缓存区的剩余空间是否小于创建的所述备份页面的大小,或者小于第二阈值;

所述触发单元，具体用于在所述判断单元判断出所述第一进程所在的进程组的片上公共缓存区的剩余空间小于创建的所述备份页面的大小，或者小于所述第二阈值时，触发所述第一进程将所述第一进程的M个工作页面中被修改的内容同步更新到所述M个工作页面所对应的M个共享虚拟内存页面中，并将所述M个工作页面在所述片上内存中的备份页面所占用的空间释放。

23. 根据权利要求22所述的数据处理装置，其特征在于，所述触发单元，还用于：如果所述判断单元判断出所述第一进程所在的进程组的片上公共缓存区的剩余空间是否小于创建的所述备份页面的大小，或者小于所述第二阈值时，触发所述第一进程所在进程组中，除所述第一进程之外的其它进程将所述其它进程的各个工作页面中被修改的内容同步更新到所述各个工作页面所对应的各个共享虚拟内存页面中，并将所述各个工作页面在所述片上内存中的备份页面所占用的空间释放。

24. 根据权利要求17-22任一项所述的数据处理装置，其特征在于，还包括：查询单元，用于在所述备份单元创建所述工作页面的备份页面之前，在所述计算节点的片上内存中查找是否存在所述工作页面的备份页面；

所述备份单元，具体用于在所述查询单元没有在所述计算节点的片上内存中查找到所述工作页面的备份页面时，创建所述工作页面的备份页面。

25. 根据权利要求24所述的数据处理装置，其特征在于，所述计算节点的片上内存中保存有备份页面信息表，其中，所述备份页面信息表包含有所述片上内存中所有备份页面的元数据信息，每一个备份页面的元数据信息包括：所述每一个备份页面的页号和版本号，其中，所述每一个备份页面的页号和版本号分别与所述每一个备份页面所对应的工作页面的页号和版本号相同；

所述查询单元，具体用于在所述备份页面信息表中查找是否有页号和版本号分别与所述工作页面的页号以及版本号相同的元数据信息；

所述备份单元，具体用于在所述查询单元没有在所述备份页面信息表中查找到页号和版本号分别与所述工作页面的页号以及版本号相同的元数据信息时，创建所述工作页面的备份页面。

26. 根据权利要求25所述的数据处理装置，其特征在于，还包括：

记录单元，用于将所述工作页面的页号以及版本号分别作为创建的所述备份页面的页号和版本号记录到所述备份页面信息表。

27. 根据权利要求24所述的数据处理装置，其特征在于，所述片上内存包括多个独立的存储区域，所述应用程序的所有进程被划分为至少一个进程组，每个所述进程组中的各个进程共享所述多个独立的存储区域中的一个存储区域，以作为该进程组的片上公共缓存区，且每个所述进程组有一个单独的备份页面信息表；每一个进程组的备份页面信息表包含有所述进程组中所有进程的所有备份页面的元数据信息，每一个备份页面的元数据信息包括：所述每一个备份页面的页号和版本号，其中，所述每一个备份页面的页号和版本号分别与所述每一个备份页面所对应的工作页面的页号和版本号相同；

所述查询单元，具体用于在所述第一进程所在进程组的备份页面信息表中查找是否有页号和版本号分别与所述工作页面的页号以及版本号相同的元数据信息；

所述备份单元，具体用于在所述查询单元没有在所述第一进程所在进程组的备份页面

信息表中查找到页号和版本号分别与所述工作页面的页号以及版本号相同的元数据信息时，创建所述工作页面的备份页面。

28. 根据权利要求27所述的数据处理装置，其特征在于，还包括：

记录单元，用于将所述工作页面的页号以及版本号分别作为创建的所述备份页面的页号和版本号记录到所述第一进程所在进程组的备份页面信息表。

29. 一种数据处理装置，其特征在于，包括：

拷贝单元，用于将第一进程请求访问的共享虚拟内存页面拷贝至计算节点的片外内存中，并将拷贝到所述片外内存中的共享虚拟内存页面作为所述第一进程的工作页面；其中，所述共享虚拟内存页面为所述第一进程所属应用程序的共享虚拟内存中的虚拟内存页面，所述应用程序运行在所述计算节点上；

查询单元，用于在所述第一进程对所述工作页面进行写操作之前，在所述计算节点的片上内存中查找是否存在所述工作页面的备份页面；

备份单元，用于在所述查询单元在所述计算节点的片上内存中查找到所述工作页面的备份页面时，确定查找到的所述备份页面为所述工作页面的备份页面，在所述查询单元没有在所述计算节点的片上内存中查找到所述工作页面的备份页面时，创建所述工作页面的备份页面，并将创建的所述备份页面存储在所述计算节点的片上内存中，其中，所述备份页面用于备份所述工作页面中的原始数据；

判断单元，用于在所述备份单元将创建的所述备份页面存储在所述计算节点的片上内存之前，判断所述片上内存的剩余空间是否小于第一阈值；

触发单元，用于在所述判断单元判断出所述片上内存的剩余空间小于第一阈值时，触发所述第一进程将所述第一进程的M个工作页面中被修改的内容同步更新到所述M个工作页面所对应的M个共享虚拟内存页面中，并将所述M个工作页面在所述片上内存中的备份页面所占用的空间释放；

其中，所述共享虚拟内存页面的个数为N，N为大于或等于1的正整数；所述第一进程的工作页面的个数为M，M为大于或等于1的正整数。

30. 根据权利要求29所述的数据处理装置，其特征在于，所述计算节点的片上内存中，保存有备份页面信息表，所述备份页面信息表包含有所述片上内存中所有备份页面的元数据信息，每一个备份页面的元数据信息包括：所述每一个备份页面的页号和版本号，其中，所述每一个备份页面的页号和版本号分别与所述每一个备份页面所对应的工作页面的页号和版本号相同；

所述查询单元，具体用于在所述备份页面信息表中查找是否有页号和版本号分别与所述工作页面的页号以及版本号相同的元数据信息；

所述备份单元，具体用于当所述查询单元查找到页号和版本号分别与所述工作页面的页号以及版本号相同的元数据信息时，将查找到的所述元数据信息所对应的备份页面作为所述工作页面的备份页面。

31. 根据权利要求29所述的数据处理装置，其特征在于，所述片上内存包括多个独立的存储区域，所述应用程序的所有进程被划分为至少一个进程组，每个所述进程组中的各个进程共享所述多个独立的存储区域中的一个存储区域，以作为该进程组的片上公共缓存区，且每个所述进程组有一个单独的备份页面信息表；每一个进程组的备份页面信息表包

含有所述进程组中所有进程的所有备份页面的元数据信息，每一个备份页面的元数据信息包括：所述每一个备份页面的页号和版本号，其中，所述每一个备份页面的页号和版本号分别与所述每一个备份页面所对应的工作页面的页号和版本号相同；

所述查询单元，具体用于在所述第一进程所在进程组的备份页面信息表中查找是否有页号和版本号分别与所述工作页面的页号以及版本号相同的元数据信息；

所述备份单元，具体用于当所述查询单元查找到页号和版本号分别与所述工作页面的页号以及版本号相同的元数据信息时，则将查找到的所述元数据信息所对应的备份页面作为所述工作页面的备份页面。

32. 一种计算机系统，其特征在于，包括：处理器、第一存储器、操作系统内核；其中，所述处理器用于运行应用程序，且所述处理器内部包含有第二存储器，所述第二存储器的数据存取速度大于所述第一存储器的数据存取速度；

所述操作系统内核，用于将所述应用程序的第一进程请求访问的共享虚拟内存页面拷贝至所述第一存储器中，并将拷贝到所述第一存储器中的共享虚拟内存页面作为所述第一进程的工作页面；在所述第一进程对所述工作页面进行写操作之前，创建所述工作页面的备份页面，并将创建的所述备份页面存储在所述第二存储器中，以备份所述工作页面的原始数据；其中，所述共享虚拟内存页面为所述应用程序的共享虚拟内存中的虚拟内存页面；

所述操作系统内核，还用于，在将创建的所述备份页面存储在所述第一存储器之前，判断所述第一存储器的剩余空间是否小于第一阈值，如果是，则触发所述第一进程将所述第一进程的M个工作页面中被修改的内容同步更新到所述M个工作页面所对应的M个共享虚拟内存页面中，并将所述M个工作页面在所述第一存储器中的备份页面所占用的空间释放，其中，所述共享虚拟内存页面的个数为N，N为大于或等于1的正整数，所述第一进程的工作页面的个数为M，M为大于或等于1的正整数。

33. 根据权利要求32所述的计算机系统，其特征在于，所述第二存储器中保存有备份页面信息表，其中，所述备份页面信息表包含有所述第二存储器中所有备份页面的元数据信息，每一个备份页面的元数据信息包括：所述每一个备份页面的页号和版本号，其中，所述每一个备份页面的页号和版本号分别与所述每一个备份页面所对应的工作页面的页号和版本号相同；

所述操作系统内核，具体用于，在所述第一进程对所述工作页面进行第一次写操作之前，在所述备份页面信息表中查找是否有页号和版本号分别与所述工作页面的页号以及版本号相同的元数据信息，如果未查找到，则创建所述工作页面的备份页面，并将创建的所述备份页面存储在所述第二存储器中，以备份所述工作页面的原始数据；如果查找到，则将查找到的所述元数据信息所对应的备份页面为所述工作页面的备份页面。

一种数据处理方法、装置及计算机系统

技术领域

[0001] 本发明涉及计算机技术领域,尤其涉及一种数据处理方法、装置及计算机系统。

背景技术

[0002] 在多核处理器中,由于各个处理器核都在共享内存方式下维护其高速缓冲存储器(Cache),因而常会产生缓存不一致的问题。这种情况发生在:不同处理器核的缓存存储了对应于同一物理内存地址但不同内容的数据,例如,在一个由A和B两个处理器核构成的多核处理器的共享内存系统中,每个处理器核都维护着独立的Cache资源。假设处理器核A和处理器核B从同一物理内存地址中读数据,即这两个处理器核读回的数据对应于同一个物理内存单元。如果之后处理器核A向这个地址写数据,则处理器核A的Cache会得到更新而处理器核B的Cache依旧保存的是旧的数据,这就会造成缓存内容不一致的问题。

[0003] 在传统多核处理器中,一般通过硬件缓存一致性协议来解决缓存一致性问题,常见的硬件缓存一致性协议有嗅探协议、基于目录结构的协议、基于令牌的协议等。然而随着众核芯片核数的增加,使用硬件缓存一致性的代价随着核数的增加而线性增长,甚至最终抵消掉核数增加带来的收益,其开销主要包括如下几个方面:

[0004] (1)通信代价:为了实现缓存一致性,需要通过缓存通信协议来进行状态更新,研究表明实现硬件缓存一致性协议的系统比非缓存一致性协议的系统片上通信流量增加20%。随着核数的增加,情况会更加糟糕;

[0005] (2)设计与验证困难:采用硬件实现成百上千个核之间的状态同步极为困难,设计复杂度使设计和验证成本急剧上升。

[0006] 虽然通过采用一些更加灵巧的设计能够减轻上述问题,但无法根本解决,因此抛弃硬件缓存一致性而采用软件缓存一致性成为了选择,比如Intel的SCC、Teraflops等众核研究芯片已经最终放弃了硬件缓存一致性实现。

[0007] DSM(Distributed Shared Memory)模型是一种用于实现软件缓存一致性的主流的内存模型,如图1所示,在这种内存模型中,一个应用程序的各进程间拥有一个相同的共享虚拟内存,每个进程分别将共享虚拟内存中的部分或者全部虚拟内存页映射到该进程维护的私有物理内存空间。各个进程在用户层面看到的是一个完整的共享虚拟内存空间,而感知不到共享虚拟内存空间中某块虚拟内存页包含的共享数据实际上是在其它进程维护的私有物理内存空间中。各个进程可以对共享虚拟内存进行任意数据操作,DSM底层通过片上网络(On-chip Network)或者所有进程都可以访问的系统共享物理内存各进程间进行数据的同步。一个应用程序的多个进程可以运行于一个处理器核上,也可以每个进程运行于一个单独处理器核上。

[0008] 区域一致性协议是一种主流的基于DSM的软件一致性协议,具有简单高效的优点。应用程序中通过Acquire(lock)/Release(lock)利用同一个锁lock进行保护的代码范围属于同一个区域(Scope),区域一致性协议只保证同一个区域中的共享变量是同步的,不同区域可以不同步。并且,区域一致性协议中一般采用Twin/Diff(备份/比较)机制来维护同一

个区域内部共享数据的一致性,现有Twin/Diff机制是基于多核平台的片外内存实现的,Twin页面是备份当前工作页面的页面,当cache空间不够时,Twin页面会存放在本地片外内存上,当进程完成对工作页面的写操作后,再将修改后的工作页面与Twin页面进行diff比较操作,并将比较结果发送给工作页面的home进程,使得home进程更新工作页面。

[0009] 在现有的Twin/Diff机制中,如果在Scope内部,程序访问了大量的页面,由于cache的大小限制,根据cache替换算法,后面访问的页面会将前面访问的页面(工作页面和Twin页面)从cache中移除,这样,当程序退出Scope时,对前面访问的页面进行Diff操作,就需要从片外内存中将工作页面和Twin页面重新加载到cache中,这样会造成很大的片外访存开销,同时也会增加数据访问的延时,影响程序的执行效率。

发明内容

[0010] 本发明实施例提供一种数据处理方法、装置及计算机系统,以保证对共享虚拟内存页面进行访问时的数据一致性,同时减少数据访问的延时,提高程序的执行效率。

[0011] 第一方面,本发明实施例提供一种数据处理方法,包括:

[0012] 将第一进程请求访问的共享虚拟内存页面拷贝至计算节点的片外内存中,并将拷贝到所述片外内存中的共享虚拟内存页面作为第一进程的工作页面;其中,所述共享虚拟内存页面为所述第一进程所属应用程序的共享虚拟内存中的虚拟内存页面,所述应用程序运行在所述计算节点上;

[0013] 在所述第一进程对所述工作页面进行写操作之前,创建所述工作页面的备份页面,并将创建的所述备份页面存储在所述计算节点的片上内存中,以备份所述工作页面的原始数据。

[0014] 在第一方面的第一种可能的实施方式中,所述共享虚拟内存页面的个数为N,N为大于或等于1的正整数;所述第一进程的工作页面的个数为M,M为大于或等于1的正整数;

[0015] 在将创建的所述备份页面存储在所述计算节点的片上内存之前,所述方法还包括:

[0016] 判断所述片上内存的剩余空间是否小于第一阈值,如果所述片上内存的剩余空间小于第一阈值,则触发所述第一进程将所述第一进程的M个工作页面中被修改的内容同步更新到所述M个工作页面所对应的M个共享虚拟内存页面中,并将所述M个工作页面在所述片上内存中的备份页面所占用的空间释放;如果所述片上内存的剩余空间大于或等于第一阈值,则执行所述将创建的所述备份页面存储在所述计算节点的片上内存中的步骤。

[0017] 结合第一方面的第一种可能的实施方式,在第二种可能的实施方式中,在创建所述工作页面的备份页面之前,还包括:在所述计算节点的片上内存中,预先给所述第一进程分配特定大小的片上存储区;

[0018] 所述将创建的所述备份页面存储在所述计算节点的片上内存中,包括:

[0019] 将创建的所述备份页面存储在为所述第一进程预先分配的片上存储区中。

[0020] 结合第一方面的第二种可能的实施方式,在第三种可能的实施方式中,所述判断所述片上内存的剩余空间是否小于第一阈值,如果所述片上内存的剩余空间小于第一阈值,则触发所述第一进程将所述第一进程的M个工作页面中被修改的内容同步更新到所述M个工作页面所对应的M个共享虚拟内存页面中,并将所述M个工作页面在所述片上内存中的

备份页面所占用的空间释放；如果所述片上内存的剩余空间大于或等于第一阈值，则执行所述将创建的所述备份页面存储在所述计算节点的片上内存中的步骤，包括：

[0021] 判断所述第一进程的片上存储区的剩余空间是否小于创建的所述备份页面的大小，或者小于第二阈值，如果所述第一进程的片上存储区的剩余空间小于创建的所述备份页面的大小，或者小于第二阈值，则触发所述第一进程将所述第一进程的M个工作页面中被修改的内容同步更新到所述M个工作页面所对应的M个共享虚拟内存页面中，并将所述M个工作页面在所述片上内存中的备份页面所占用的空间释放；如果所述第一进程的片上存储区的剩余空间大于或等于创建的所述备份页面的大小，或者小于第二阈值，则执行所述将创建的所述备份页面存储在所述计算节点的片上内存中的步骤。

[0022] 结合第一方面的第一种可能的实施方式，在第四种可能的实施方式中，所述第一阈值为创建的所述备份页面的大小。

[0023] 结合第一方面的第四种可能的实施方式，在第五种可能的实施方式中，该数据处理方法还包括：如果所述计算节点的片上内存的剩余空间小于创建的所述备份页面的大小，则触发所述应用程序中，除所述第一进程之外的其它至少一个进程将所述其它至少一个进程的各个工作页面中被修改的内容同步更新到所述各个工作页面所对应的各个共享虚拟内存页面中，并将所述各个工作页面在所述片上内存中的备份页面所占用的空间释放。

[0024] 结合第一方面的第一种可能的实施方式，在第六种可能的实施方式中，所述计算节点的片上内存包括多个独立的存储区域，所述应用程序的所有进程被划分为至少一个进程组，每个所述进程组中的各个进程共享所述多个独立的存储区域中的一个存储区域，以作为该进程组的片上公共缓存区；

[0025] 所述将创建的所述备份页面存储在所述计算节点的片上内存中，包括：

[0026] 将创建的所述备份页面存储在为所述第一进程所在进程组的片上公共缓存区中；

[0027] 所述判断所述片上内存的剩余空间是否小于第一阈值，如果所述片上内存的剩余空间小于第一阈值，则触发所述第一进程将所述第一进程的M个工作页面中被修改的内容同步更新到所述M个工作页面所对应的M个共享虚拟内存页面中，并将所述M个工作页面在所述片上内存中的备份页面所占用的空间释放；如果所述片上内存的剩余空间大于或等于第一阈值，则执行所述将创建的所述备份页面存储在所述计算节点的片上内存中的步骤，包括：

[0028] 判断所述第一进程所在的进程组的片上公共缓存区的剩余空间是否小于创建的所述备份页面的大小，或者小于所述第二阈值，如果所述第一进程所在的进程组的片上公共缓存区的剩余空间是否小于创建的所述备份页面的大小，或者小于所述第二阈值时，则触发所述第一进程将所述第一进程的M个工作页面中被修改的内容同步更新到所述M个工作页面所对应的M个共享虚拟内存页面中，并将所述M个工作页面在所述片上内存中的备份页面所占用的空间释放；如果所述第一进程所在的进程组的片上公共缓存区的剩余空间大于或等于创建的所述备份页面的大小，或者小于所述第二阈值时，则执行所述将创建的所述备份页面存储在所述计算节点的片上内存中的步骤。

[0029] 结合第一方面的第一、第二、第三、第四、第五或第六种可能的实施方式，在第七种可能的实施方式中，该数据处理方法还包括：在创建所述工作页面的备份页面之前，在所述

计算节点的片上内存中查找是否存在所述工作页面的备份页面；

[0030] 所述创建所述工作页面的备份页面，包括：如果没有在所述计算节点的片上内存中查找到所述工作页面的备份页面，则创建所述工作页面的备份页面。

[0031] 结合第一方面的第七种可能的实施方式，在第八种可能的实施方式中，所述计算节点的片上内存中保存有备份页面信息表，其中，所述备份页面信息表包含有所述片上内存中所有备份页面的元数据信息，每一个备份页面的元数据信息包括：所述每一个备份页面的页号和版本号，其中，所述每一个备份页面的页号和版本号分别与所述每一个备份页面所对应的工作页面的页号和版本号相同；

[0032] 所述在所述计算节点的片上内存中查找是否存在所述工作页面的备份页面，包括：

[0033] 在所述备份页面信息表中查找是否有页号和版本号分别与所述工作页面的页号以及版本号相同的元数据信息；

[0034] 所述创建所述工作页面的备份页面，包括：如果没有在所述备份页面信息表中查找到页号和版本号分别与所述工作页面的页号以及版本号相同的元数据信息，则创建所述工作页面的备份页面。

[0035] 第二方面，本发明实施例还提供另一种数据处理方法，包括：

[0036] 将第一进程请求访问的共享虚拟内存页面拷贝至计算节点的片外内存中，并将拷贝到所述片外内存中的共享虚拟内存页面作为所述第一进程的工作页面；其中，所述共享虚拟内存页面为所述第一进程所属应用程序的共享虚拟内存中的虚拟内存页面，所述应用程序运行在所述计算节点上；

[0037] 在所述第一进程对所述工作页面进行写操作之前，在所述计算节点的片上内存中查找是否存在所述工作页面的备份页面，如果查找到，则确定查找到的所述备份页面为所述工作页面的备份页面。

[0038] 在第二方面的第一种可能的实施方式中，所述数据处理方法还包括：如果没有查找到，则创建所述工作页面的备份页面，并将创建的所述备份页面存储在所述计算节点的片上内存中。

[0039] 结合第二方面的第一种可能的实施方式，在第二种可能的实施方式中，所述共享虚拟内存页面的个数为N，N为大于或等于1的正整数；所述第一进程的工作页面的个数为M，M为大于或等于1的正整数；

[0040] 在将创建的所述备份页面存储在所述计算节点的片上内存之前，所述方法还包括：

[0041] 判断所述片上内存的剩余空间是否小于第一阈值，如果是，则触发所述第一进程将所述第一进程的M个工作页面中被修改的内容同步更新到所述M个工作页面所对应的M个共享虚拟内存页面中，并将所述M个工作页面在所述片上内存中的备份页面所占用的空间释放；如果不是，则执行所述将创建的所述备份页面存储在所述计算节点的片上内存中的步骤。

[0042] 结合第二方面的第一种，或者第二种可能的实施方式，在第三种可能的实施方式中，所述计算节点的片上内存中保存有备份页面信息表，其中，所述备份页面信息表包含有所述片上内存中所有备份页面的元数据信息，每一个备份页面的元数据信息包括：所述每

一个备份页面的页号和版本号,其中,所述每一个备份页面的页号和版本号分别与所述每一个备份页面所对应的工作页面的页号和版本号相同;

[0043] 所述在所述计算节点的片上内存中查找是否存在所述工作页面的备份页面,如果查找到,则确定查找到的所述备份页面为所述工作页面的备份页面,包括:

[0044] 在所述备份页面信息表中查找是否有页号和版本号分别与所述工作页面的页号以及版本号相同的元数据信息,如果查找到,则确定查找到的所述元数据信息所对应的备份页面为所述工作页面的备份页面。

[0045] 结合第二方面的第一种,或者第二种可能的实施方式,在第四种可能的实施方式中,所述片上内存包括多个独立的存储区域,所述应用程序的所有进程被划分为至少一个进程组,每个所述进程组中的各个进程共享所述多个独立的存储区域中的一个存储区域,以作为该进程组的片上公共缓存区,且每个所述进程组有一个单独的备份页面信息表;每一个进程组的备份页面信息表包含有所述进程组中所有进程的所有备份页面的元数据信息,每一个备份页面的元数据信息包括:所述每一个备份页面的页号和版本号,其中,所述每一个备份页面的页号和版本号分别与所述每一个备份页面所对应的工作页面的页号和版本号相同;

[0046] 所述在所述计算节点的片上内存中查找是否存在所述工作页面的备份页面,如果查找到,则确定查找到的所述备份页面为所述工作页面的备份页面,包括:

[0047] 在所述第一进程所在进程组的备份页面信息表中查找是否有页号和版本号分别与所述工作页面的页号以及版本号相同的元数据信息,如果查找到,则确定查找到的所述元数据信息所对应的备份页面为所述工作页面的备份页面。

[0048] 第三方面,本发明实施例还提供一种数据处理装置,包括:

[0049] 拷贝单元,用于将第一进程请求访问的共享虚拟内存页面拷贝至计算节点的片外内存中,并将拷贝到所述片外内存中的共享虚拟内存页面作为所述第一进程的工作页面;其中,所述共享虚拟内存页面为所述第一进程所属应用程序的共享虚拟内存中的虚拟内存页面,所述应用程序运行在所述计算节点上;

[0050] 备份单元,用于在所述第一进程对所述工作页面进行写操作之前,创建所述工作页面的备份页面,并将创建的所述备份页面存储在所述计算节点的片上内存中,以备份所述工作页面的原始数据。

[0051] 在第三方面的第一种可能的实施方式中,所述共享虚拟内存页面的个数为N,N为大于或等于1的正整数;所述第一进程的工作页面的个数为M,M为大于或等于1的正整数;

[0052] 所述数据处理装置还包括:判断单元,用于在所述备份单元将创建的所述备份页面存储在所述计算节点的片上内存之前,判断所述片上内存的剩余空间是否小于第一阈值;

[0053] 触发单元,用于在所述判断单元判断出所述片上内存的剩余空间小于第一阈值时,触发所述第一进程将所述第一进程的M个工作页面中被修改的内容同步更新到所述M个工作页面所对应的M个共享虚拟内存页面中,并将所述M个工作页面在所述片上内存中的备份页面所占用的空间释放。

[0054] 结合第三方面第一种可能的实施方式,在第二种可能的实施方式中,所述数据处理装置还包括:

[0055] 内存分配单元,用于在所述计算节点的片上内存中,预先给所述第一进程所属的应用程序的各个进程分配特定大小的片上存储区;

[0056] 所述备份单元,具体用于将创建的所述备份页面存储在所述内存分配单元为所述第一进程预先分配的片上存储区中。

[0057] 结合第三方面第二种可能的实施方式,在第三种可能的实施方式中,所述判断单元,具体用于:

[0058] 判断所述第一进程的片上存储区的剩余空间是否小于创建的所述备份页面的大小,或者小于第二阈值;

[0059] 所述触发单元,具体用于在所述判断单元判断出所述第一进程的片上存储区的剩余空间小于创建的所述备份页面的大小,或者小于第二阈值时,触发所述第一进程将所述第一进程的M个工作页面中被修改的内容同步更新到所述M个工作页面所对应的M个共享虚拟内存页面中,并将所述M个工作页面在所述片上内存中的备份页面所占用的空间释放。

[0060] 结合第三方面第一种可能的实施方式,在第四种可能的实施方式中,所述应用程序的各个进程共享所述计算节点的片上内存;

[0061] 所述判断单元,具体用于:判断所述计算节点的片上内存的剩余空间是否小于创建的所述备份页面的大小;

[0062] 所述触发单元,具体用于在所述判断单元判断出所述计算节点的片上内存的剩余空间小于创建的所述备份页面的大小时,触发所述第一进程将所述第一进程的M个工作页面中被修改的内容同步更新到所述M个工作页面所对应的M个共享虚拟内存页面中,并将所述M个工作页面在所述片上内存中的备份页面所占用的空间释放。

[0063] 结合第三方面的第一、第二、第三或第四种可能的实施方式,在第五种可能的实施方式中,所述数据处理装置还包括:查询单元,用于在所述备份单元创建所述工作页面的备份页面之前,在所述计算节点的片上内存中查找是否存在所述工作页面的备份页面;

[0064] 所述备份单元,具体用于在所述查询单元没有在所述计算节点的片上内存中查找到所述工作页面的备份页面时,创建所述工作页面的备份页面。

[0065] 结合第三方面的第五种可能的实施方式,在第六种可能的实施方式中,所述计算节点的片上内存中保存有备份页面信息表,其中,所述备份页面信息表包含有所述片上内存中所有备份页面的元数据信息,每一个备份页面的元数据信息包括:所述每一个备份页面的页号和版本号,其中,所述每一个备份页面的页号和版本号分别与所述每一个备份页面所对应的工作页面的页号和版本号相同;

[0066] 所述查询单元,具体用于在所述备份页面信息表中查找是否有页号和版本号分别与所述工作页面的页号以及版本号相同的元数据信息;

[0067] 所述备份单元,具体用于在所述查询单元没有在所述备份页面信息表中查找到页号和版本号分别与所述工作页面的页号以及版本号相同的元数据信息时,创建所述工作页面的备份页面。

[0068] 第四方面,本发明实施例还提供另一种数据处理装置,包括:

[0069] 拷贝单元,用于将第一进程请求访问的共享虚拟内存页面拷贝至计算节点的片外内存中,并将拷贝到所述片外内存中的共享虚拟内存页面作为所述第一进程的工作页面;其中,所述共享虚拟内存页面为所述第一进程所属应用程序的共享虚拟内存中的虚拟内

存页面,所述应用程序运行在所述计算节点上;

[0070] 查询单元,用于在所述第一进程对所述工作页面进行写操作之前,在所述计算节点的片上内存中查找是否存在所述工作页面的备份页面;

[0071] 备份单元,用于在所述查询单元在所述计算节点的片上内存中查找到所述工作页面的备份页面时,确定查找到的所述备份页面为所述工作页面的备份页面,其中,所述备份页面用于备份所述工作页面中的原始数据。

[0072] 在第四方面的第一种可能的实施方式中,所述备份单元还用,在所述查询单元没有在所述计算节点的片上内存中查找到所述工作页面的备份页面时,创建所述工作页面的备份页面,并将创建的所述备份页面存储在所述计算节点的片上内存中。

[0073] 结合第四方面的第一种可能的实施方式,在第二种可能的实施方式中,所述共享虚拟内存页面的个数为N,N为大于或等于1的正整数;所述第一进程的工作页面的个数为M,M为大于或等于1的正整数;

[0074] 所述数据处理装置还包括:判断单元,用于在所述备份单元将创建的所述备份页面存储在所述计算节点的片上内存之前,判断所述片上内存的剩余空间是否小于第一阈值;

[0075] 触发单元,用于在所述判断单元判断出所述片上内存的剩余空间小于第一阈值时,触发所述第一进程将所述第一进程的M个工作页面中被修改的内容同步更新到所述M个工作页面所对应的M个共享虚拟内存页面中,并将所述M个工作页面在所述片上内存中的备份页面所占用的空间释放。

[0076] 结合第四方面的第一种、或者第二种可能的实施方式,在第三种可能的实施方式中,所述计算节点的片上内存中,保存有备份页面信息表,所述备份页面信息表包含有所述片上内存中所有备份页面的元数据信息,每一个备份页面的元数据信息包括:所述每一个备份页面的页号和版本号,其中,所述每一个备份页面的页号和版本号分别与所述每一个备份页面所对应的工作页面的页号和版本号相同;

[0077] 所述查询单元,具体用于在所述备份页面信息表中查找是否有页号和版本号分别与所述工作页面的页号以及版本号相同的元数据信息;

[0078] 所述备份单元,具体用于当所述查询单元查找到页号和版本号分别与所述工作页面的页号以及版本号相同的元数据信息时,将查找到的所述元数据信息所对应的备份页面作为所述工作页面的备份页面。

[0079] 结合第四方面的第一种、或者第二种可能的实施方式,在第四种可能的实施方式中,所述片上内存包括多个独立的存储区域,所述应用程序的所有进程被划分为至少一个进程组,每个所述进程组中的各个进程共享所述多个独立的存储区域中的一个存储区域,以作为该进程组的片上公共缓存区,且每个所述进程组有一个单独的备份页面信息表;每一个进程组的备份页面信息表包含有所述进程组中所有进程的所有备份页面的元数据信息,每一个备份页面的元数据信息包括:所述每一个备份页面的页号和版本号,其中,所述每一个备份页面的页号和版本号分别与所述每一个备份页面所对应的工作页面的页号和版本号相同;

[0080] 所述查询单元,具体用于在所述第一进程所在进程组的备份页面信息表中查找是否有页号和版本号分别与所述工作页面的页号以及版本号相同的元数据信息;

[0081] 所述备份单元，具体用于当所述查询单元查找到页号和版本号分别与所述工作页面的页号以及版本号相同的元数据信息时，则将查找到的所述元数据信息所对应的备份页面作为所述工作页面的备份页面。

[0082] 第五方面，本发明实施例提供一种计算机系统，包括：处理器、第一存储器、操作系统内核；其中，所述处理器用于运行应用程序，且所述处理器内部包含有第二存储器，所述第二存储器的数据存取速度大于所述第一存储器的数据存取速度；

[0083] 所述操作系统内核，用于将所述应用程序的第一进程请求访问的共享虚拟内存页面拷贝至所述第一存储器中，并将拷贝到所述第一存储器中的共享虚拟内存页面作为所述第一进程的工作页面；在所述第一进程对所述工作页面进行写操作之前，创建所述工作页面的备份页面，并将创建的所述备份页面存储在所述第二存储器中，以备份所述工作页面的原始数据；其中，所述共享虚拟内存页面为所述应用程序的共享虚拟内存中的虚拟内存页面。

[0084] 在第五方面的第一种可能的实施方式中，所述共享虚拟内存页面的个数为N，N为大于或等于1的正整数；所述第一进程的工作页面的个数为M，M为大于或等于1的正整数；

[0085] 所述操作系统内核，还用于，在将创建的所述备份页面存储在所述第一存储器之前，判断所述第一存储器的剩余空间是否小于第一阈值，如果是，则触发所述第一进程将所述第一进程的M个工作页面中被修改的内容同步更新到所述M个工作页面所对应的M个共享虚拟内存页面中，并将所述M个工作页面在所述第一存储器中的备份页面所占用的空间释放。

[0086] 结合第五方面，或者第五方面的第一种可能的实施方式，在第二种可能的实施方式中，所述第二存储器中保存有备份页面信息表，其中，所述备份页面信息表包含有所述第二存储器中所有备份页面的元数据信息，每一个备份页面的元数据信息包括：所述每一个备份页面的页号和版本号，其中，所述每一个备份页面的页号和版本号分别与所述每一个备份页面所对应的工作页面的页号和版本号相同；

[0087] 所述操作系统内核，具体用于，在所述第一进程对所述工作页面进行第一次写操作之前，在所述备份页面信息表中查找是否有页号和版本号分别与所述工作页面的页号以及版本号相同的元数据信息，如果未查找到，则创建所述工作页面的备份页面，并将创建的所述备份页面存储在所述第二存储器中，以备份所述工作页面的原始数据；如果查找到，则将查找到的所述元数据信息所对应的备份页面为所述工作页面的备份页面。

[0088] 由上述技术方案可知，本发明实施例将共享虚拟页面中的数据内容拷贝到片外内存中的工作页面，供进程进行读和写操作，同时利用计算节点的可编程片上内存，在对工作页面进行写操作之前，将页面数据在片上内存中备份，以保证多个进程在对共享虚拟内存页面进行操作时的数据一致性，由于备份页面存储在片上内存中，不会与工作页面竞争缓存空间，更多的工作页面可以存放在缓存中，而缓存和片上内存的访问速度都比较快(比访问片外内存快5~10倍)，从而可以提高程序运行的性能；

[0089] 进一步地，通过主动的比较机制，及时更新共享虚拟内存页面的内容，保证在进行比较操作时，工作页面基本上都还位于cache中，不需要进行片外内存访问，因此数据更新的速度很快。

附图说明

[0090] 为了更清楚地说明本发明的技术方案,下面将对实施例中所需要使用的附图作一简单地介绍,显而易见地,下面附图只是本发明的一些实施例的附图,对于本领域普通技术人员来讲,在不付出创造性劳动性的前提下,还可以根据这些附图获得同样能实现本发明技术方案的其它附图。

- [0091] 图1为本发明实施例提供的分布式共享内存模型示意图;
- [0092] 图2为本发明实施例提供的Twin/Diff流程示意图;
- [0093] 图3为本发明实施例提供的计算节点的逻辑架构图;
- [0094] 图4为本发明实施例提供的一种数据处理方法流程图;
- [0095] 图5为本发明实施例提供的共享虚拟内存空间示意图;
- [0096] 图6为本发明实施例提供的另一种数据处理方法流程图;
- [0097] 图7为本发明实施例提供的一种的数据处理方法示意图;
- [0098] 图8为本发明实施例提供的备份页面的元数据信息示意图;
- [0099] 图9为本发明实施例提供的另一种数据处理方法流程图;
- [0100] 图10为本发明实施例提供的一种数据处理装置示意图;
- [0101] 图11为本发明实施例提供的另一种数据处理装置示意图;
- [0102] 图12为本发明实施例提供的一种计算节点的示意图;
- [0103] 图13为本发明实施例提供的一种计算机系统的示意图。

具体实施方式

[0104] 为使本发明的目的、技术方案和优点更加清楚,下面将结合本发明实施例中的附图,对本发明的技术方案进行清楚、完整地描述。显然,下述的各个实施例都只是本发明一部分的实施例。基于本发明下述的各个实施例,本领域普通技术人员即使没有作出创造性劳动,也可以通过等效变换部分甚至全部的技术特征,而获得能够解决本发明技术问题,实现本发明技术效果的其它实施例,而这些变换而来的各个实施例显然并不脱离本发明所公开的范围。

[0105] 为使本领域一般技术人员更好的了解本发明实施例提供的技术方案,首先对本发明实施例技术方案的应用场景做一些简单的介绍,本发明实施例提供的技术方案可应用于在计算节点上,通过改进现有的Twin/Diff机制以实现软件缓存一致性,其中,计算节点是指具有处理器和存储器,部署有操作系统并支持片上内存的计算机或芯片,比如Intel的SCC(Single Chip Cloud Computer,单芯片云计算),计算节点的处理器包含有一个或多个处理器核,且各个处理器核都在共享内存方式下维护其高速缓冲存储器(Cache)。在其它的应用场景中,只要涉及软件缓存一致性,本发明实施例的技术方案均适用。在具体的实施过程中,本发明实施例技术方案的产品形态包括但不限于:操作系统,LIB库、使用第三方LIB库的应用系统以及部署有操作系统和/或LIB库的计算机。

[0106] 进一步地,为了使本领域一般技术人员更好的了解本发明实施例提供的技术方案,对现有技术中的Twin/Diff机制做一些简单的介绍,参见图2,现有技术中,Twin/Diff机制流程如下:

[0107] 步骤1,进程A通过执行Acquire (lock1),获得锁lock1,进入区域1;

[0108] 步骤2,如果进程A读一个共享虚拟内存页面P,且共享虚拟内存页面P在进程A的地址空间中不存在映射,则操作系统触发页错误 (Page Fault),在进程A的页错误处理函数中,从共享虚拟内存页面P的home进程获得共享虚拟内存页面P的数据,并从片外内存申请一个物理内存页面保存该数据,然后在进程A的地址空间中建立共享虚拟内存页面P与该物理内存页面的映射关系,并修改进程A对共享虚拟内存页面P的访问权限为只读;转步骤5;

[0109] 步骤3,如果进程A写一个共享虚拟内存页面P,且共享虚拟内存页面P在进程A的地址空间中不存在映射,则操作系统触发页错误 (Page Fault),并在进程A的页错误处理函数中,从共享虚拟内存页面P的home进程获得共享虚拟内存页面P的数据,从片外内存申请一个物理内存页面保存该数据,该物理内存页面即为进程A的工作页面,然后在进程A的地址空间中建立共享虚拟内存页面P与该物理内存页面的映射关系,并在片外内存中为共享虚拟内存页面P保存一个Twin页面(维护一份相同的数据),修改进程A对共享虚拟内存页面P的访问权限为可读写;转步骤5;

[0110] 步骤4,如果进程A写一个共享虚拟内存页面P,且共享虚拟内存页面P在进程A的地址空间中存在映射,且进程A对共享虚拟内存页面P的访问权限为只读,则操作系统触发页错误 (Page Fault),在进程A页错误处理函数中,在片外内存中用一个备份页面 (Twin页面) 来备份共享虚拟内存页面P中的原始数据,修改进程A对共享虚拟内存页面P的访问权限为可读写;转步骤5;

[0111] 步骤5,进程A正常读/写共享虚拟内存页面P;

[0112] 步骤6,进程A通过执行Release (lock1),触发比较 (Diff) 操作:比较共享虚拟内存页面P所映射的物理内存页面(工作页面)与Twin页面的数据内容,生成Diff文件,更新home进程所维护的共享虚拟内存页面P的数据。Diff操作完成后,进程A释放锁lock1,离开区域1。

[0113] 针对现有的Twin/Diff机制存在的问题,本发明提出了一种数据处理方法,首先,以图3为例介绍本发明实施例提供的数据处理方法所应用的计算节点的逻辑结构,该计算机节点具体可以是集成有操作系统内核的多核芯片、通用计算机、云计算、分布式系统中的计算节点、服务器等大型计算设备、也可以是手机、平板电脑等移动终端。如图3所示,计算节点的硬件层包括一个或多个处理器(一个处理器可能包含多个核),例如CPU,GPU,当然还可以包括片外存储器(片外内存、硬盘等)、输入/输出设备、网络接口等,其中,处理器内存集成有缓存(cache)和片上内存,处理器的各个核之中也集成有cache;在硬件层之上运行有操作系统内核(例如Linux Kernel)以及第三方库Libraries(例如显示管理器Surface Manager,媒体框架Media Framework等),第三方库可以被操作系统载入内存执行;除此之外,该计算节点还可以包括应用层,该应用层包括一个或多个应用程序等,一个应用程序包含有一个或多个进程,一个进程又包含有一个或多个线程,应用程序运行在该计算节点上,通过操作系统内核实现对硬件层的控制,以完成相应的功能。

[0114] 如图3所示,本发明实施例提供的数据处理方法可以由第三方库Libraries层以及应用层实施,也可以在操作系统或者超级管理程序(hypervisor)层面实现。应用程序通过执行acquire、release等函数,来访问虚拟共享内存页面,并触发硬件层执行相应操作,以保证计算节点上运行的多个进程或线程在对共享虚拟内存页面进行操作时的数据一致性。

[0115] 具体地,如图4所示,本发明实施例提供的数据处理方法包括:

[0116] S401,将第一进程请求访问的共享虚拟内存页面拷贝至计算节点的片外内存中,并将拷贝到所述片外内存中的共享虚拟内存页面作为第一进程的工作页面;其中,所述共享虚拟内存页面为所述第一进程所属应用程序的共享虚拟内存中的虚拟内存页面,且所述应用程序运行在所述计算节点上;

[0117] 图5所示的是第一进程P0所属的应用程序1的共享虚拟内存空间示意图,假设应用程序1包含4个进程:第一进程P0,以及第二进程P1~P3分别运行在计算节点的不同的处理器核上,它们有统一的共享虚拟内存用于存放共享数据,且每一个进程都单独维护该共享虚拟内存中的一块共享区域,每个共享区域包含一个或多个虚拟内存页面;如图5所示,虚拟地址空间中的第1块区域映射到P0的私有物理内存中,第2块区域映射到P1的私有物理内存中,第3块区域映射到P2的私有物理内存中,第4块区域映射到P3的私有物理内存中。各个进程在用户层面看到的是一个完整的共享虚拟内存,而感知不到共享虚拟内存中某块虚拟内存页包含的共享数据实际上是在其它进程维护的私有物理内存空间中。各个进程可以对共享虚拟内存进行任意数据操作,计算节点底层通过片上网络(On-chip Network)、互联网络(Interconnection Network)或者所有进程都可以访问的系统共享物理内存各进程间进行数据的同步。其中,本发明实施例所述的第二进程,是指应用程序1中,除第一进程之外的其它所有进程中的一个,仅用于与第一进程所区分,并不用于特指某一个进程。

[0118] 在本发明的各个实施例中,第一进程请求访问的共享虚拟内存页面,具体是指第一进程请求读或写的共享虚拟内存页面;具体地,如果第一进程P0读或写一个共享虚拟内存页面P,且共享虚拟内存页面P在第一进程P0的物理地址空间中不存在映射,则操作系统触发页错误(Page Fault),在第一进程P0的页错误处理函数中,从P的home进程获得共享虚拟内存页面P的数据,并从计算节点的片外内存申请一个物理内存页面作为第一进程的一个工作页面,同时建立共享虚拟内存页面P与该物理内存页面的映射关系,并将共享虚拟内存页面P的数据写入该物理内存页面;其中,工作页面是可供第一进程进行读和写操作的页面,第一进程在读或者写共享虚拟页面的时候,实际上是在读或者写该共享虚拟页面对应的工作页面,这样保证了多个进程在读写同一个共享虚拟页面时,不会造成冲突。另外需要说明的是,第一进程拷贝到片外内存中的共享虚拟页面可以为多个,第一进程的工作页面的个数也为多个,一般而言,两者个数相同,即每一个工作页面都对应于一个共享虚拟内存页面,可以理解的是,在另一种实施方式中,共享虚拟内存页面的和工作页面可以不一一对应,一个共享虚拟内存页面的数据可以存放到多个工作页面中,或者多个虚拟内存页面的数据可以存放到一个工作页面中。

[0119] 需要说明的是,在一个实施例中,应用程序1的共享虚拟内存中的不同区域,是由不同的进程或线程来单独维护的(即图5所描述的情形),在这种情形下,第一进程具体可以从共享虚拟内存页面P的home进程获得共享虚拟内存页面P的数据。在另一个实施例中,如果应用程序1的整个共享虚拟内存空间是由应用程序1的所有进程共同维护的,那么第一进程在读共享虚拟内存页面P的时候,就不需要从P的home进程获得页面P的数据,而是可以直接读取共享虚拟内存页面P的数据。还需要说明的是,本发明实施例所描述的片外内存,是指计算节点的CPU外部的存储器,比如计算节点的内存、硬盘等。

[0120] S402,在第一进程对该工作页面进行写操作之前,创建该工作页面的备份页面,并

将创建的备份页面存储在所述计算节点的片上内存中，以备份该工作页面的原始数据；

[0121] 具体地，可以在第一进程首次写该工作页面的时候，创建该工作页面的备份页面，以备份该工作页面的原始数据，可以理解的是，工作页面的原始数据，即为该工作页面在被进程进行读或写操作之前保存的数据；当在片上内存中成功创建了该工作页面的备份页面后，操作系统跳出第一进程的页错误处理函数，从而使得第一进程从产生page fault错误的地方重新执行，进而对该工作页面中的数据进行读或写操作。由于在片上内存中已经维护了一份该工作页面的原始数据，后续无论第一进程对该工作页面作何修改，均可以通过比较该工作页面的当前数据内容与该工作页面的备份页面中的原始数据内容，确定该工作页面中哪些部分的内容已经被进程修改，进而可以将被修改的这些内容同步更新到该工作页面所对应的共享虚拟内存页面中。

[0122] 本发明实施例所描述的片上内存，可以是计算节点的CPU内部的存储器，比如可编程片上内存（例如Intel SCC众核平台提供了可编程片上内存MPB（Message Passing Buffer））；进一步地，片上内存可以是计算节点的CPU内部，除缓存（如L1cache，L2cache）之外的另一片存储区域，如果该计算节点的CPU具有多个核，则各个核可以共享该片上内存。片上内存空间不是很大，但访问延时与计算节点的CPU的二级缓存L2cache类似，是一种不错的资源。

[0123] 需要说明的是，第一进程在读或写共享虚拟内存页面P之前，一般会通过执行Acquire（lock1），获得锁lock1，以进入页面P所在的共享虚拟内存区域1；同时，第一进程可通过执行Release（lock1），退出共享虚拟内存区域1，同时触发Diff操作：比较共享虚拟内存页面P所映射的物理内存页面（工作页面）与备份页面的数据内容，生成用于记录两者的数据内容差异的日志文件Diff，以使共享虚拟内存页面P的home进程更新共享虚拟内存页面P中的数据。Diff操作完成后，第一进程释放锁lock1，离开区域1。

[0124] 在一种更具体的实施方式中，第一进程所属应用程序1的每个进程，在该片上内存中分别独占一块片上存储区，以存放该进程的工作页面的备份页面。在这种情形下，在创建所述工作页面的备份页面之前，还包括：在所述计算节点的片上内存中，预先给应用程序1的各个进程分配特定大小的片上存储区；其中，为各个进程分配的片上存储区的大小，本领域技术人员可以根据片上内存的总容量，可用容量来确定，也可以根据经验值来设定，另外，各个进程的片上存储区大小可以相同，也可以不同。

[0125] 相应地，将创建的所述备份页面存储在所述计算节点的片上内存中，包括：将创建的所述备份页面存储在为第一进程预先分配的片上存储区中。

[0126] 最后需要说明的是，本发明实施例中以第一进程为例来阐述本发明技术方案，但不应理解为对本发明方案执行主体的限制，本领域技术人员可以理解的是，本发明实施例方案的执行实体可以为进程或者线程。该执行实体或者为线程，或者为进程，不可以在完整的实现方法中既代表线程，又代表进程。也就是说，本发明各实施例中所述的方法可以在线程对应的维度内实现，也可以在进程对应的维度内实现。

[0127] 本发明实施例通过以上技术方案，将进程请求读或写的共享虚拟内存页面先拷贝在计算节点片外内存中，作为可供进程进行读写操作的工作页面，同时利用计算节点的CPU的片上内存，在进程对工作页面进行写操作之前，将工作页面中的原始数据在片上内存中备份，以保证多个进程在对共享虚拟内存页面进行操作时的数据一致性，由于备份页面存

储在片上内存中，页面的访问速度可以得到保证，同时，备份页面与工作页面分开存储，使得备份页面不会与工作页面竞争缓存空间，更多的工作页面可以存放在缓存中，从而可以提高程序运行的性能。

[0128] 本发明实施例提供了另一种计算节点上的数据处理方法，如图6所示，该方法包括：

[0129] S601，将第一进程请求访问的N个共享虚拟内存页面拷贝至计算节点的片外内存中，并将拷贝到片外内存中的N共享虚拟内存页面作为第一进程的M个工作页面；其中，共享虚拟内存页面为第一进程所属应用程序的共享虚拟内存中的虚拟内存页面，M、N均为大于或等于1的正整数；

[0130] 一般而言，M=N，即每一个工作页面都唯一对应于一个共享虚拟内存页面，可以理解的是，在另一种实施方式中，共享虚拟内存页面的和工作页面可以不一一对应，一个共享虚拟内存页面的数据可以存放到多个工作页面中，或者多个虚拟内存页面的数据可以存放到一个工作页面中。

[0131] S602，在第一进程对M个工作页面的任一个进行写操作之前，创建该工作页面的备份页面；

[0132] S603，判断该计算节点的片上内存的剩余空间是否小于第一阈值，如果是，则触发第一进程将M个工作页面中被修改的内容同步更新到该M个工作页面所对应的各个共享虚拟内存页面中，并将该M个工作页面在片上内存中的备份页面所占用的空间释放。

[0133] 其中，第一进程将自身的M个工作页面中被修改的内容同步更新到各个工作页面所对应的各个共享虚拟内存页面中，并将各个工作页面在片上内存中的备份页面所占用的空间释放的过程，我们称之为“第一进程执行比较(Diff)操作”。在一个实施例中，进程A执行比较操作具体包括：将进程A的所有工作页面中的每一个工作页面与该工作页面在片上内存中的备份页面进行比较，比较两者的数据内容差异，即找出被进程A修改过，数据发生变化的部分，并生成用于记录两者的数据内容差异的日志文件，根据所述日志文件将进程A对每一个工作页面修改的内容更新到该工作页面所对应的共享虚拟页面中。需要说明的是，根据日志文件更新共享虚拟页面的操作，可以由进程A自身来完成，也可以由被更新的共享虚拟页面的home进程来完成，具体需要根据具体的应用场景来确定。

[0134] 具体地，在创建一个工作页面的备份页面后，如果片上内存的剩余空间小于第一阈值，会导致该工作页面的数据无法备份到片上内存中，从而触发第一进程执行比较操作，从而可以释放片上内存空间，并刷新共享虚拟内存空间中相应页面的数据；可以理解的是，本领域技术人员可将片上内存的总容量、可用存储空间等因素综合考虑后，来设定该第一阈值，也可根据经验值来设置，此处不对其做特别的限定。

[0135] 需要说明的是，本发明实施例通过计算节点的片上内存空间来备份工作页面，在不同的应用实例中，片上内存空间的划分可以采用不同的方式，相应地，根据片上内存空间的划分方式不同，触发第一进程执行比较操作的时机和方式也有所不同：

[0136] 在第一种可能的实施方式中，应用程序的每个进程在片上内存中有单独的片上存储区；即在所述计算节点的片上内存中，预先给应用程序的各个进程分配特定大小的片上存储区；其中，为各个进程在片上内存分配的存储空间大小可以相同，也可以不同。在这种情形下，步骤S603中的判断步骤，就具体包括：判断所述第一进程的片上存储区的剩余空间

是否小于步骤S602中创建的一个备份页面的大小,或者小于第二阈值。相应地,如果判断出第一进程的片上存储区的剩余空间小于一个备份页面的大小,或者小于第二阈值,则触发第一进程执行比较操作。

[0137] 在第二种可能的实施方式中,应用程序的所有进程共享整个片上内存,或者片上内存中的局部区域,即计算节点上的所有进程都可以将备份页面保存在片上内存中。在这种情形下,第一阈值可以为步骤S602中创建的备份页面的大小;即步骤S603中的判断步骤,就具体包括:判断所述计算节点的片上内存的剩余空间是否小于步骤S602中创建的一个备份页面的大小。相应地,如果判断出所述计算节点的片上内存的剩余空间小于步骤S602中创建的一个备份页面的大小,则触发进程执行比较操作。进一步地,触发进程执行比较操作有两种策略:

[0138] (1) 仅触发第一进程执行比较操作,应用程序的其它进程不受影响。

[0139] (2) 触发第一进程所属应用程序1的所有进程执行比较操作。

[0140] 在第三种可能的实施方式中,计算节点的片上内存空间不是全局共享的,而是局部共享的,即只有部分进程可以共享片上内存中的局部区域。这种片上内存的划分方式具体为:整个片上内存空间被划分为多块相互独立的存储区域,第一进程所属的应用程序1的所有进程被划分为一个或多个进程组,每个进程组中的进程共享多块独立的存储区域中的一个,以作为该进程组的片上公共缓存区;即每个进程组中的进程可以访问相同的片上内存空间,不同进程组访问不同的片上内存空间。比如片上内存1可以被core0~core3访问,片上内存2可以被core4~core7访问,片上内存3可以被core8~core11访问,片上内存4可以被core12~core15访问。一个进程组中的所有进程都可以将自身工作页面的备份页面保存在片上内存中的某块存储区域上。在这种片上内存划分方式下,将创建的所述备份页面存储在所述计算节点的片上内存中,具体包括:将创建的所述备份页面存储在为所述第一进程所在进程组的片上公共缓存区中;相应地,步骤S603中的判断步骤,就具体包括:判断第一进程所在的进程组的片上公共缓存区的剩余空间是否小于步骤S602中创建的一个备份页面的大小,或者小于第二阈值。相应地,如果判断出第一进程所在的进程组的片上公共缓存区的剩余空间小于步骤S602中创建的一个备份页面的大小,或者小于第二阈值,则会导致创建该工作页面的备份页面失败,进而触发进程执行比较操作,同样,触发进程执行比较操作有两种策略:

[0141] (1) 仅触发第一进程执行比较操作,第一进程所在的进程组中的其它进程不受影响。

[0142] (2) 在第一进程所在的进程组中广播失败信息,触发进程组中所有进程执行比较操作。

[0143] 需要说明的是,进程执行比较操作的具体流程与之前的描述相同,其它进程执行比较操作的流程,与第一进程执行比较操作流程类似,此处不再赘述,还需要说明的是,第二阈值可根据片上内存的总容量、可用存储空间等因素来设定,也可根据经验值来设定,此处不对其做特别的限定。

[0144] 本发明实施例中,假设计算节点的片上内存空间比L2cache空间要小,因此以片上内存空间无法容纳下一个备份页面为比较操作的触发条件;如果片上内存空间比L2cache大,则可以以备份页面所占空间与L2cache大小相等为比较操作的触发条件。另外,其它启

发式的触发条件判断也是可行的。如果计算节点的片上内存比较大,也可以考虑将工作页面保存在片上内存空间中。

[0145] 本发明实施例通过以上技术方案,将进程请求读或写的共享虚拟内存页面先拷贝在计算节点片外内存中,作为可供进程进行读写操作的工作页面,同时利用计算节点的CPU的片上内存,在进程对工作页面进行写操作之前,将工作页面中的原始数据在片上内存中备份,以保证多个进程在对共享虚拟内存页面进行操作时的数据一致性,由于备份页面存储在片上内存中,页面的访问速度可以得到保证,同时,备份页面与工作页面分开存储,使得备份页面不会与工作页面竞争缓存空间,更多的工作页面可以存放在缓存中,从而可以提高程序运行的性能;进一步地,通过主动触发的比较操作,及时更新共享虚拟内存页面的内容,保证在进行比较操作时,工作页面基本上都还位于cache中,不需要进行片外内存访问,因此比较操作的速度很快。

[0146] S604,将创建的所述备份页面存储在所述计算节点的片上内存中,以备份所述该工作页面的原始数据;

[0147] 其中,步骤S601、S602、S604的具体实施细节可以参照上述方法实施例中的步骤S401、S402,此处不再赘述。

[0148] 基于上述方法实施例,下面通过一个具体的应用实例来描述本发明技术方案,假设应用程序的各个进程各自维护共享虚拟内存中的一片区域,即是不同区域内的虚拟内存页面有不同的home进程,如图7所示,基于这种场景的方法流程如下:

[0149] (1) 第一进程A通过执行Acquire (lock1),获得锁lock1,进入区域1;

[0150] (2) 如果第一进程A读一个共享虚拟内存页面P,且共享虚拟内存页面P在进程A的地址空间中不存在映射,则操作系统触发页错误 (Page Fault),并在进程A的页错误处理函数中,从共享虚拟内存页面P的home进程获得共享虚拟内存页面P的数据,并从片外内存申请一个物理内存页面保存该数据,然后在进程A的地址空间中建立共享虚拟内存页面P与该物理内存页面的映射关系,并将该物理内存页面作为进程A的工作页面,同时修改进程A对共享虚拟内存页面P的访问权限为只读;转步骤5;

[0151] (3) 如果进程A写一个共享虚拟内存页面P,且共享虚拟内存页面P在进程A的地址空间中不存在映射,则操作系统触发页错误 (Page Fault),并在进程A的页错误处理函数中,从共享虚拟内存页面P的home进程获得共享虚拟内存页面P的数据,并从片外内存申请一个物理内存页面保存该数据,然后在进程A的地址空间中建立共享虚拟内存页面P与该物理内存页面的映射关系,将该物理内存页面作为进程A的工作页面,并在可编程片上内存中为该工作页面创建一个Twin页面(维护一份相同的数据,这个过程中如果创建Twin页面失败,会触发主动的Diff操作),修改进程A对共享虚拟内存页面P的访问权限为可读写;转步骤5;

[0152] (4) 如果进程A写一个共享虚拟内存页面P,且共享虚拟内存页面P在进程A的地址空间中存在映射,且进程A对共享虚拟内存页面P的访问权限为只读,则操作系统触发页错误 (Page Fault),并在进程A的页错误处理函数中,在可编程片上内存中为共享虚拟内存页面P保存一个Twin页面(维护一份相同的数据,这个过程中可能会触发积极主动的Diff操作),修改进程A对共享虚拟内存页面P的访问权限为可读写;转步骤5;

[0153] (5) 进程A正常读/写共享虚拟内存页面P;

[0154] (6) 进程A通过执行Release(lock1),触发Diff操作:对还未进行Diff操作的共享虚拟内存页面所映射的物理内存页面(工作页面)与Twin页面的数据内容进行比较,生成日志文件Diff,更新各共享虚拟内存页面所对应home进程维护的共享虚拟内存页面的数据。Diff操作完成后,进程A释放锁lock1,离开区域1。

[0155] 需要说明的是,其它共享虚拟内存页面处理方式同共享虚拟内存页面P。

[0156] 进一步地,由于计算节点的片上内存,一般都是全局共享的,也就是说可以被计算节点的所有core访问,因此,本发明实施例提出了备份页面资源池的方案。

[0157] 由于可编程片上内存是所有core都可以访问的,因此可编程片上内存中保存的备份页面也是一种共享资源,可以被不同core共享。比如多个进程(不管在相同core上或者不同core上)在某个时间段都对页面x的不同位置进行了写操作,按照上述实施例中的方案,每个进程都需要一个单独的备份页面,这样浪费了内存空间。由于片上内存是所有core都可以访问的,因此不同进程的同一个共享虚拟内存页面的同一个版本,在片上内存空间只需要保存一份备份页面即可。基于此,可以对本发明实施例的数据处理方法做进一步优化,在步骤创建工作页面的备份页面之前,可以先在计算节点的片上内存中查找是否存在该工作页面的备份页面,如果查找到,就可以跳过S602中创建备份页面的步骤,以及步骤S603、S604;如果没有查找到,才执行步骤S602中创建备份页面的步骤,以及步骤S603、S604。

[0158] 在一种具体的实现方式中,可以用一个数据结构,比如备份页面队列或备份页面信息表来记录片上内存中各个备份页面的元数据信息,其中,备份页面的元数据信息包括:备份页面的页号Twin Page Id和版本号Twin Page Version;在一个更优的实施例中,如图8所示,备份页面的元数据信息可以包括:Twin Page Id、Twin Page Version、Twin Page Usage和Twin Page Address。其中,各个字段的含义如下:

[0159] Twin Page Id:备份页面所对应的工作页面所映射的共享虚拟内存页面的页号,不同进程中同一个共享虚拟内存页面的页号相同;

[0160] Twin Page Version:备份页面的版本号,不同进程中同一个共享虚拟内存页面所对应的备份页面版本号可能不同;

[0161] 需要说明的是,备份页面的版本号与备份页面所对应的工作页面的版本号相同,而工作页面的版本号,是进程在创建该工作页面时,该工作页面所对应的共享虚拟内存页面的版本号。具体地,进程在创建一个工作页面时,即将一个共享虚拟内存页面拷贝至计算节点的片外内存中作为该进程的一个工作页面时,该工作页面的版本号即为该拷贝的共享虚拟内存页面的版本号;一般来说,一个共享虚拟内存页面的版本号的初始值为1,后续不同的进程均可以对该共享虚拟内存页面中的数据进行读和写操作,进程每次更新该共享虚拟内存页面中的内容之后,该共享虚拟内存页面的版本号就会递增。因此,同一个共享虚拟内存页面在不同进程中所对应的工作页面的版本号可能不同,进而导致其所对应的备份页面版本号也不同。

[0162] Twin Page Usage:备份页面的使用情况,记录当前使用该版本的备份页面的进程数目;

[0163] Twin Page Address:备份页面在可编程片上内存中的地址,进程可以根据该地址访问相应版本的备份页面。

[0164] 在这种情形下,具体可以采用如下方式,在计算节点的片上内存中查找是否存在

某个工作页面的备份页面:在创建一个工作页面的备份页面之前,根据该工作页面的页号以及版本号,在用于记录备份页面资源池中各个备份页面的元数据信息的备份页面信息表中查找是否有页号和版本号分别与该工作页面的页号以及版本号相同的元数据信息;

[0165] 相应地,在步骤S602中,创建备份页面的步骤具体包括:如果没有在所述备份页面信息表中查找到页号和版本号分别与所述工作页面的页号以及版本号相同的元数据信息,则创建该工作页面的备份页面;然后继续执行后续步骤S603、S604;进一步地,如果查找到页号和版本号分别与该工作页面的页号以及版本号相同的元数据信息,则确定查找到的元数据信息所对应的备份页面为该工作页面的备份页面,并跳过S602中创建备份页面的步骤,以及步骤S603、S604。

[0166] 进一步地,在所述创建该工作页面的备份页面之后,还可以将该工作页面的页号以及版本号分别作为创建的备份页面的页号和版本号记录到备份页面信息表。

[0167] 下面通过具体的实例对上述方案进行说明:

[0168] (1) 初始阶段,共享虚拟内存页面x在Home进程的版本号为1。

[0169] (2) 某个时刻,进程A、进程B分别进行Acquire (lock1), Acquire (lock2) 操作,从Home进程获得了共享虚拟内存页面x的拷贝,并且分别进行了写操作。假设进程A先写,由于在备份页面信息表中找不到共享虚拟内存页面x所对应的版本号为1的备份页面x,则在片上内存中创建版本号为1的备份页面x,并设置页面x的Twin Page Usage为1。当进程B写共享虚拟内存页面x时,由于可以在备份页面信息表中找到版本号为1的备份页面x,因此不用再创建同样的备份页面x,只需要将备份页面x的Twin Page Usage修改为2。

[0170] (3) 后续某个时刻,进程A通过Release操作触发Diff操作,或者通过上述主动的Diff操作,更新Home进程的共享虚拟内存页面x,Home进程的共享虚拟内存页面x的版本号顺序递增,变为2,同时,备份页面信息表中备份页面x的Twin Page Usage修改为1。

[0171] (4) 进程C进行Acquire (lock3) 操作,从Home进程获得了共享虚拟内存页面x的拷贝,并且进行了写操作。进程C获得的共享虚拟内存页面x的拷贝的版本号为2,而备份页面信息表中能够查找到的备份页面x的版本号为1,两者不匹配,所以会在片上内存中创建版本号为2的备份页面x,并设置其TwinPage Usage为1。

[0172] 进一步地,在另一个实施例中,计算节点的片上内存空间可以不是全局共享的,而是局部共享的,也就是说整个片上内存空间被划分为多块相互独立的存储区域;相应地,第一进程所属的应用程序的所有进程被划分为至少一个进程组,每个所述进程组中的各个进程共享所述多个独立的存储区域中的一个存储区域,以作为该进程组的片上公共缓存区,且每个进程组维护有一个单独的备份页面信息表,每一个进程组的备份页面信息表包含有该进程组中所有进程的所有备份页面的元数据信息。其中,备份页面的元数据信息的具体定义前面已经做了说明,此处不再赘述。在这种片上内存局部共享的情形下,在计算节点的片上内存中查找是否存在某个工作页面的备份页面,具体包括:

[0173] 根据工作页面的页号以及版本号,在第一进程所在进程组的备份页面信息表中查找是否有页号和版本号分别与该工作页面的页号以及版本号相同的元数据信息;

[0174] 相应地,在步骤S602中,创建备份页面的步骤具体包括:如果没有在第一进程所在进程组的备份页面信息表中查找到页号和版本号分别与该工作页面的页号以及版本号相同的元数据信息,则创建该工作页面的备份页面;然后继续执行后续步骤S603、S604;如果

查找到页号和版本号分别与该工作页面的页号以及版本号相同的元数据信息，则确定查找到的元数据信息所对应的备份页面为该工作页面的备份页面，并跳过S602中创建备份页面的步骤，以及步骤S603、S604。

[0175] 进一步地，在创建该工作页面的备份页面之后，可以将该工作页面的页号以及版本号分别作为创建的备份页面的页号和版本号记录到第一进程所在进程组的备份页面信息表。

[0176] 本发明实施例提出的基于备份页面共享的优化方案，可在前述方法实施例的基础上，进一步减少备份页面占用的片上内存空间，节约系统资源。

[0177] 如图9所示，本发明实施例提供另一种数据处理方法，包括：

[0178] S901，将第一进程请求访问的共享虚拟内存页面拷贝至计算节点的片外内存中，并将拷贝到片外内存中的共享虚拟内存页面作为第一进程的工作页面；其中，共享虚拟内存页面为第一进程所属应用程序的共享虚拟内存中的虚拟内存页面，该应用程序运行在该计算节点上；

[0179] S802，在第一进程对该工作页面进行写操作之前，在该计算节点的片上内存中查找是否存在该工作页面的备份页面，如果查找到，则确定查找到的备份页面为所述工作页面的备份页面，其中，该备份页面保存有该工作页面中的原始数据。

[0180] 可选地，该数据处理方法还可以包括：

[0181] S803，如果没有查找到，则创建该工作页面的备份页面，并将创建的该备份页面存储在该计算节点的片上内存中，其中，该备份页面用于备份该工作页面中的原始数据。

[0182] 进一步地，如果第一进程请求访问的共享虚拟内存页面的个数为N；第一进程的工作页面的个数为M，则该数据处理方法还包括：

[0183] 在将创建的该备份页面存储在计算节点的片上内存之前，判断片上内存的剩余空间是否小于第一阈值，如果是，则触发第一进程将自身的M个工作页面中被修改的内容同步更新到该M个工作页面所对应的M个共享虚拟内存页面中，并将该M个工作页面在片上内存中的备份页面所占用的空间释放；如果否，则执行S803中将备份页面存储在片上内存中的步骤。

[0184] 在一种具体的实现方式中，可以在用保存在片上内存中的一个数据结构，比如备份页面队列或备份页面信息表来记录片上内存中各个备份页面的元数据信息，其中，备份页面的元数据信息的具体定义前面已经做了说明，此处不再赘述。

[0185] 在这种情形下，具体可以采用如下方式查找是否存在某个工作页面的备份页面：在创建一个工作页面的备份页面之前，根据该工作页面的页号以及版本号，在用于记录备份页面资源池中各个备份页面的元数据信息的备份页面信息表中查找是否有页号和版本号分别与该工作页面的页号以及版本号相同的元数据信息。相应地，如果查找到，则确定查找到的元数据信息所对应的备份页面为该工作页面的备份页面；如果没有查找到，则创建该工作页面的备份页面，并将创建的备份页面存储在计算节点的片上内存中。

[0186] 进一步地，在所述创建该工作页面的备份页面之后，还可以将该工作页面的页号以及版本号分别作为创建的备份页面的页号和版本号记录到备份页面信息表。

[0187] 进一步地，在另一个实施例中，计算节点的片上内存空间可以不是全局共享的，而是局部共享的，也就是说整个片上内存空间被划分为多块相互独立的存储区域；相应地，第

一进程所属的应用程序的所有进程被划分为至少一个进程组,每个所述进程组中的各个进程共享所述多个独立的存储区域中的一个存储区域,以作为该进程组的片上公共缓存区,且每个进程组维护有一个单独的备份页面信息表,每一个进程组的备份页面信息表包含有该进程组中所有进程的所有备份页面的元数据信息。其中,备份页面的元数据信息的具体定义前面已经做了说明,此处不再赘述。在这种片上内存局部共享的情形下,在计算节点的片上内存中查找是否存在某个工作页面的备份页面,具体包括:

[0188] 根据工作页面的页号以及版本号,在第一进程所在进程组的备份页面信息表中查找是否有页号和版本号分别与该工作页面的页号以及版本号相同的元数据信息;

[0189] 相应地,如果查找到,则直接将查找到的元数据信息所对应的备份页面作为该工作页面的备份页面。如果没有查找到,则创建该工作页面的备份页面,并将创建的备份页面存储在计算节点的片上内存中。

[0190] 进一步地,在创建该工作页面的备份页面之后,可以将该工作页面的页号以及版本号分别作为创建的备份页面的页号和版本号记录到第一进程所在进程组的备份页面信息表。

[0191] 本发明实施例提出的基于备份页面共享的数据处理方法,可在前述方法实施例的基础上,进一步减少备份页面占用的片上内存空间,节约系统资源。

[0192] 本发明实施例还提供了一种数据处理装置,该数据处理装置可以以library库形式实现,也可以在操作系统或者超级管理程序(hypervisor)层面实现。如图10所示,该数据处理装置90包括:拷贝单元910,用于将第一进程请求访问的共享虚拟内存页面拷贝至计算节点的片外内存中,并将拷贝到所述片外内存中的共享虚拟内存页面作为第一进程的工作页面;其中,第一进程请求访问的共享虚拟内存页面,具体为第一进程所属应用程序的共享虚拟内存中,第一进程请求读或者写的一个或多个虚拟内存页面;

[0193] 具体地,如果第一进程P0读或者写一个共享虚拟内存页面P,则数据处理装置会触发页错误(Page Fault),在第一进程P0的页错误处理函数中,拷贝单元910获得共享虚拟内存页面P的数据,从计算节点的片外内存申请一个物理内存页面作为第一进程的一个工作页面,同时建立共享虚拟内存页面P与该物理内存页面(工作页面)的映射关系,并将共享虚拟内存页面P的数据写入该物理内存页面;其中,工作页面是可供第一进程进行读和写操作的页面,第一进程在读或者写共享虚拟页面的时候,实际上是在读或者写该共享虚拟页面对应的工作页面,这样保证了多个进程在读写同一个共享虚拟页面时,不会造成冲突。每一个工作页面都对应于一个共享虚拟内存页面,如果第一进程拷贝到片外内存中的共享虚拟页面为多个,那么第一进程的工作页面的个数也为多个。需要说明的是,在一个实施例中,第一进程所属应用程序的共享虚拟内存中的不同区域,是由不同的进程或线程来单独维护的(即图5所描述的情形),在这种情形下,拷贝单元910具体可以从共享虚拟内存页面P的home进程获得共享虚拟内存页面P的数据。在另一个实施例中,如果应用程序的共享虚拟内存空间是由应用程序的所有进程共同维护的,那么拷贝单元910在读或写共享虚拟内存页面P的时候,就不需要从P的home进程获得页面P的数据,而是可以直接读取共享虚拟内存页面P的数据。

[0194] 备份单元920,用于在第一进程对该工作页面进行写操作之前,创建该工作页面的备份页面,并将创建的备份页面存储在该计算节点的片上内存中,以备份该工作页面的原

始数据；

[0195] 如果备份单元920成功创建了一个工作页面的备份页面，并将该备份页面存放到在片上内存之后，会自动跳出第一进程的页错误处理函数，从而触发第一进程从产生page fault错误的地方重新执行，进而对该工作页面中的数据进行读或写操作。由于在片上内存中已经维护了一份该工作页面的原始数据，后续无论第一进程对该工作页面作何修改，均可以通过比较该工作页面的当前数据内容与该工作页面的备份页面中的原始数据内容，确定该工作页面中哪些部分的内容已经被进程修改，进而可以将被修改的这些内容同步更新到该工作页面所对应的共享虚拟内存页面中。

[0196] 具体地，本发明实施例所描述的片外内存，是指计算节点的CPU外部的存储器，比如计算节点的内存、硬盘等；本发明实施例所描述的片上内存，可以是计算节点的CPU内部的存储器，比如可编程片上内存（例如Intel SCC众核平台提供了可编程片上内存MPB（Message Passing Buffer））；进一步地，片上内存可以是计算节点的CPU内部，除缓存（如L1cache,L2cache）之外的另一片存储区域，如果该计算节点的CPU具有多个核，则各个核可以共享该片上内存。片上内存空间不是很大，但访问延时与计算节点的CPU的二级缓存L2cache类似，是一种不错的资源。

[0197] 进一步地，在另一个实施例中，假设拷贝单元910拷贝了N个共享虚拟内存页面到计算节点的片外内存中，第一进程的工作页面为M个，M,N均为大于1的正整数，则数据处理装置90还包括：

[0198] 判断单元930，用于在备份单元920将创建的备份页面存储在所述计算节点的片上内存之前，判断所述片上内存的剩余空间是否小于第一阈值；

[0199] 触发单元940，用于在判断单元930判断出片上内存的剩余空间小于第一阈值时，触发第一进程将第一进程的M个工作页面中被修改的内容同步更新到该M个工作页面所对应的M个共享虚拟内存页面中，并将该M个工作页面的备份页面所占用的空间释放；当所述判断单元判断出所述片上内存的剩余空间不小于第一阈值时，触发备份单元920将创建的备份页面存储到该计算节点的片上内存中。

[0200] 其中，第一进程将自身的M个工作页面中被修改的内容同步更新到各个工作页面所对应的各个共享虚拟内存页面中，并将各个工作页面在片上内存中的备份页面所占用的空间释放的过程，我们称之为“第一进程执行比较(Diff)操作”。在一个实施例中，进程A执行比较操作具体包括：将进程A的所有工作页面中的每一个工作页面与该工作页面在片上内存中的备份页面进行比较，比较两者的数据内容差异，即找出被进程A修改过，数据发生变化的部分，并生成用于记录两者的数据内容差异的日志文件，根据所述日志文件将进程A对每一个工作页面修改的内容更新到该工作页面所对应的共享虚拟页面中。需要说明的是，根据日志文件更新共享虚拟页面的操作，可以由进程A自身来完成，也可以由被更新的共享虚拟页面的home进程来完成，具体需要根据具体的应用场景来确定。

[0201] 具体地，在备份单元920创建一个工作页面的备份页面后，如果片上内存的剩余空间小于第一阈值，会导致该工作页面的数据无法备份到片上内存中，从而使得触发单元940触发第一进程执行比较操作，从而释放片上内存空间，并刷新共享虚拟内存空间中相应页面的数据；可以理解的是，本领域技术人员可将片上内存的总容量、可用存储空间等因素综合考虑后，来设定该第一阈值，也可根据经验值来设置，此处不对其做特别的限定。

[0202] 需要说明的是,本发明实施例中,备份单元920通过计算节点的片上内存空间来存储工作页面的备份页面,在不同的应用实例中,片上内存空间的划分可以采用不同的方式,相应地,根据片上内存空间的划分方式不同,判断单元930和触发单元940的工作方式也有所不同:

[0203] 在第一种可能的实施方式中,第一进程所属的应用程序1的每个进程在片上内存中有单独的片上存储区;即数据处理装置90还包括:内存分配单元970,用于在计算节点的片上内存中,预先给应用程序1的各个进程分配特定大小的片上存储区;在这种情形下,备份单元920,具体用于将创建的备份页面存储在所述内存分配单元为所述第一进程预先分配的片上存储区中。进一步地,判断单元930,具体用于:判断第一进程的片上存储区的剩余空间是否小于备份单元920当前要存储的备份页面的大小,或者小于第二阈值。相应地,如果判断单元930判断出第一进程的片上存储区的剩余空间小于备份单元920当前要存储的备份页面的大小,或者小于第二阈值,则触发单元940触发第一进程执行比较操作。

[0204] 在第二种可能的实施方式中,应用程序1的所有进程共享整个片上内存,或者片上内存中的局部区域,即计算节点上的所有进程都可以将备份页面保存在片上内存中。在这种情形下,判断单元930,具体用于:判断该计算节点的片上内存的剩余空间是否小于备份单元920当前创建并要存储的备份页面的大小,或者小于第二阈值。相应地,如果判断单元930判断出该计算节点的片上内存的剩余空间小于备份单元920当前创建并要存储的备份页面的大小,则触发单元940触发进程执行比较操作。进一步地,触发单元940触发进程执行比较操作有两种策略:

[0205] (1) 仅触发第一进程执行比较操作,应用程序的其它进程不受影响。

[0206] (2) 触发第一进程所属应用程序1的所有进程执行比较操作。

[0207] 在第三种可能的实施方式中,计算节点的片上内存空间不是全局共享的,而是局部共享的,即只有部分进程可以共享片上内存中的局部区域。这种片上内存的划分方式具体为:整个片上内存空间被划分为多块相互独立的存储区域,应用程序1的所有进程被划分为一个或多个进程组,每个进程组中的进程共享多块独立的存储区域中的一个,以作为该进程组的片上公共缓存区;即每个进程组中的进程可以访问相同的片上内存空间,不同进程组访问不同的片上内存空间。比如片上内存1可以被core0~core3访问,片上内存2可以被core4~core7访问,片上内存3可以被core8~core11访问,片上内存4可以被core12~core15访问。一个进程组中的所有进程都可以将自身工作页面的备份页面保存在片上内存中的某块存储区域上。在这种片上内存划分方式下,备份单元920,具体用于将创建的所述备份页面存储在为所述第一进程所在进程组的片上公共缓存区中;判断单元930,具体用于:判断第一进程所在的进程组的片上公共缓存区的剩余空间是否小于备份单元920当前创建并要存储的备份页面的大小,或者小于第二阈值。相应地,如果判断单元930判断出第一进程所在的进程组的片上公共缓存区的剩余空间小于备份单元920当前创建并要存储的备份页面的大小,或者小于第二阈值,则触发单元940触发进程执行比较操作。进一步地,触发单元940触发进程执行比较操作有两种策略:

[0208] (1) 仅触发第一进程执行比较操作,第一进程所在的进程组中的其它进程不受影响。

[0209] (2) 在第一进程所在的进程组中广播失败信息,触发该进程组中所有进程执行主

动的比较操作。

[0210] 需要说明的是,进程执行比较操作的具体流程与之前的描述相同,其它进程执行比较操作流程,与第一进程执行比较操作流程类似,此处不再赘述,还需要说明的是,第二阈值可根据片上内存的总容量、可用存储空间等因素来设定,也可根据经验值来设定,此处不对其做特别的限定。

[0211] 进一步地,由于计算节点的片上内存,一般都是全局共享的,也就是说可以被计算节点的所有core访问,因此,本发明实施例提出了备份页面资源池的方案。

[0212] 由于片上内存是所有core都可以访问的,因此片上内存中保存的备份页面也是一种共享资源,可以被不同core共享。比如多个进程(不管在相同core上或者不同core上)在某个时间段都对页面x的不同位置进行了写操作,按照上述实施例中的方案,每个进程都需要一个单独的备份页面,这样浪费了内存空间。由于片上内存是所有core都可以访问的,因此不同进程的同一个共享虚拟内存页面的同一个版本,在片上内存空间只需要保存一份备份页面即可。基于此,本发明实施例提出了备份页面资源池的概念,将片上内存中维护的所有备份页面集合叫作备份页面资源池。相应地,数据处理装置90还包括:

[0213] 查询单元950,用于在备份单元920创建工作页面的备份页面之前,在计算节点的片上内存中查找是否存在该工作页面的备份页面;相应地,当查询单元950没有在所述计算节点的片上内存中查找到该工作页面的备份页面时,备份单元920才会执行创建该工作页面的备份页面的步骤。

[0214] 具体地,在一个实施例中,可以在计算节点的片上内存中,用一个数据结构,比如备份页面队列或备份页面信息表来记录片上内存中各个备份页面的元数据信息,其中,备份页面的元数据信息包括:备份页面的页号Twin Page Id和版本号Twin Page Version;在一个更优的实施例中,如图8所示,备份页面的元数据信息可以包括:Twin Page Id、Twin Page Version、Twin Page Usage和Twin Page Address。其中,备份页面的元数据信息的具体定义前面已经做了说明,此处不再赘述。可以理解的是,由于计算节点的片上内存资源宝贵,为了减少对片上内存的占用,也可以将备份页面信息表保存在计算节点的片外内存中。

[0215] 基于备份页面信息表,本发明实施例提供的据处理装置90中,查询单元950,具体用于在备份页面信息表中查找是否有页号和版本号分别与该工作页面的页号以及版本号相同的元数据信息;

[0216] 备份单元920,具体用于当查询单元950没有查找到页号和版本号分别与该工作页面的页号以及版本号相同的元数据信息时,创建该工作页面的备份页面;当查询单元950查找到页号和版本号分别与该工作页面的页号以及版本号相同的元数据信息时,则将查找到的元数据信息所对应的备份页面作为该工作页面的备份页面;。

[0217] 进一步地,在一个实施例中,数据处理装置90还包括:记录单元960,用于在备份单元920创建该工作页面的备份页面之后,将该工作页面的页号及版本号分别作为创建的备份页面的页号和版本号记录到所述备份页面信息表。

[0218] 在另一个实施例中,如果计算节点的片上内存空间可以不是全局共享的,而是局部共享的,也就是说整个片上内存空间被划分为多块相互独立的存储区域;相应地,第一进程所属的应用程序1的所有进程被划分为至少一个进程组,每个所述进程组中的各个进程共享所述多个独立的存储区域中的一个存储区域,以作为该进程组的片上公共缓存区,且

每个进程组维护有一个单独的备份页面信息表,每一个进程组的备份页面信息表包含有该进程组中所有进程的所有备份页面的元数据信息。其中,备份页面的元数据信息的具体定义前面已经做了说明,此处不再赘述。在这种片上内存局部共享的情形下,数据处理装置90的查询单元950,具体用于在所述第一进程所在进程组的备份页面信息表中查找是否有页号和版本号分别与所述工作页面的页号以及版本号相同的元数据信息;

[0219] 备份单元920,具体用于当查询单元950没有查找到页号和版本号分别与所述工作页面的页号以及版本号相同的元数据信息时,创建所述工作页面的备份页面;进一步地,当查询单元950查找到页号和版本号分别与所述工作页面的页号以及版本号相同的元数据信息时,则将查找到的所述元数据信息所对应的备份页面作为所述工作页面的备份页面。

[0220] 相应地,数据处理装置90的记录单元960,具体用于在备份单元920创建该工作页面的备份页面之后,将该工作页面的页号以及版本号分别作为创建的备份页面的页号和版本号记录到第一进程所在进程组的备份页面信息表。

[0221] 如图11所示,本发明实施例提供另一种数据处理装置11,包括:

[0222] 拷贝单元110,用于将第一进程请求访问的共享虚拟内存页面拷贝至计算节点的片外内存中,并将拷贝到片外内存中的共享虚拟内存页面作为第一进程的工作页面;其中,共享虚拟内存页面为第一进程所属应用程序的共享虚拟内存中的虚拟内存页面,且该应用程序运行在该计算节点上;

[0223] 查询单元120,用于在第一进程对该工作页面进行写操作之前,在该计算节点的片上内存中查找是否存在所述工作页面的备份页面;

[0224] 备份单元130,用于在查询单元120在该计算节点的片上内存中查找到该工作页面的备份页面时,确定查找到的备份页面为该工作页面的备份页面;进一步地,当查询单元120没有在片上内存中查找到该工作页面的备份页面时,备份单元130创建该工作页面的备份页面,并将创建的所述备份页面存储在所述计算节点的片上内存中,其中,该工作页面的备份页面用于备份该工作页面的原始数据。

[0225] 优选地,在一个实施例中,如果第一进程请求访问的共享虚拟内存页面的个数为N,第一进程的工作页面的个数为M,M,N均为大于或等于1的正整数;

[0226] 数据处理装置11还包括:判断单元140,用于在备份单元130将创建的所述备份页面存储在所述计算节点的片上内存之前,判断所述片上内存的剩余空间是否小于第一阈值;

[0227] 触发单元150,用于在判断单元140判断出所述片上内存的剩余空间小于第一阈值时,触发所述第一进程将所述第一进程的M个工作页面中被修改的内容同步更新到所述M个工作页面所对应的M个共享虚拟内存页面中,并将所述M个工作页面在所述片上内存中的备份页面所占用的空间释放。

[0228] 在一种具体的实现方式中,可以计算节点的片上内存中,用一个数据结构,比如备份页面队列或备份页面信息表来记录片上内存中各个备份页面的元数据信息,其中,备份页面的元数据信息包括:备份页面的页号Twin Page Id和版本号Twin Page Version;在一个更优的实施例中,如图8所示,备份页面的元数据信息可以包括:Twin Page Id、Twin Page Version、Twin Page Usage和Twin Page Address。其中,各个字段的含义如下:

[0229] Twin Page Id:备份页面所对应的工作页面所映射的共享虚拟内存页面的页号,

不同进程中同一个共享虚拟内存页面的页号相同；

[0230] Twin Page Version: 备份页面的版本号, 不同进程中同一个共享虚拟内存页面所对应的备份页面版本号可能不同；

[0231] 需要说明的是, 备份页面的版本号与备份页面所对应的工作页面的版本号相同, 而工作页面的版本号, 是备份单元130在创建该工作页面时, 该工作页面所对应的共享虚拟内存页面的版本号。具体地, 备份单元130在创建一个工作页面时, 即将一个共享虚拟内存页面拷贝至计算节点的片外内存中作为该进程的一个工作页面时, 该工作页面的版本号即为该拷贝的共享虚拟内存页面的版本号; 一般来说, 一个共享虚拟内存页面的版本号的初始值为1, 后续不同的进程均可以对该共享虚拟内存页面中的数据进行读和写操作, 进程每次更新该共享虚拟内存页面中的内容之后, 该共享虚拟内存页面的版本号就会递增。因此, 同一个共享虚拟内存页面在不同进程中所对应的工作页面的版本号可能不同, 进而导致其所对应的备份页面版本号也不同。

[0232] Twin Page Usage: 备份页面的使用情况, 记录当前使用该版本的备份页面的进程数目；

[0233] Twin Page Address: 备份页面在可编程片上内存中的地址, 进程可以根据该地址访问相应版本的备份页面。

[0234] 可以理解的是, 由于计算节点的片上内存资源宝贵, 为了减少对片上内存的占用, 也可以将备份页面信息表保存在计算节点的片外内存中; 基于所述备份页面信息表, 本发明实施例提供的数据处理装置11中, 查询单元120, 具体用于在所述备份页面信息表中查找是否有页号和版本号分别与所述工作页面的页号以及版本号相同的元数据信息；

[0235] 备份单元130, 具体用于当查询单元120查找到页号和版本号分别与该工作页面的页号以及版本号相同的元数据信息时, 则将查找到的元数据信息所对应的备份页面作为该工作页面的备份页面。

[0236] 进一步地, 在一个实施例中, 数据处理装置11还包括: 记录单元160, 用于在备份单元130创建工作页面的备份页面之后, 将该工作页面的页号及版本号分别作为创建的备份页面的页号和版本号记录到备份页面信息表。

[0237] 在另一个实施例中, 如果计算节点的片上内存空间可以不是全局共享的, 而是局部共享的, 也就是说整个片上内存空间被划分为多块相互独立的存储区域; 相应地, 第一进程所属的应用程序1的所有进程被划分为至少一个进程组, 每个所述进程组中的各个进程共享所述多个独立的存储区域中的一个存储区域, 以作为该进程组的片上公共缓存区, 且每个进程组维护有一个单独的备份页面信息表, 每一个进程组的备份页面信息表包含有该进程组中所有进程的所有备份页面的元数据信息。其中, 备份页面的元数据信息的具体定义前面已经做了说明, 此处不再赘述。在这种片上内存局部共享的情形下, 数据处理装置11的查询单元120, 具体用于在第一进程所在进程组的备份页面信息表中查找是否有页号和版本号分别与该工作页面的页号以及版本号相同的元数据信息；

[0238] 备份单元130, 具体用于当查询单元120查找到页号和版本号分别与该工作页面的页号以及版本号相同的元数据信息时, 则将查找到的元数据信息所对应的备份页面作为该工作页面的备份页面。

[0239] 相应地, 数据处理装置11的记录单元160, 具体用于在备份单元130创建该工作页

面的备份页面之后,将该工作页面的页号以及版本号分别作为创建的备份页面的页号和版本号记录到第一进程所在进程组的备份页面信息表。

[0240] 需要说明的是,图10和11所示数据处理装置中,其各个模块的具体实施过程以及各个模块之间的信息交互等内容,由于与本发明方法实施例基于同一发明构思,可以参见方法实施例,在此不一一赘述。

[0241] 需要说明的是,在实际应用中,本发明实施例的数据处理装置90和11,可以为计算节点,换言之,即具有处理器和存储器的设备(其示意性的架构可参考附图3),其产品形态可以是通用计算机、云计算机、分布式系统中的计算节点、嵌入式平台、服务器等等,也可以是操作系统、LIB库等软件系统,本发明对此不作限定。

[0242] 本发明实施例通过以上技术方案,将进程请求读或写的共享虚拟内存页面先拷贝在计算节点片外内存中,作为可供进程进行读写操作的工作页面,同时利用计算节点的CPU的片上内存,在进程对工作页面进行写操作之前,将工作页面中的原始数据在片上内存中备份,以保证多个进程在对共享虚拟内存页面进行操作时的数据一致性,由于备份页面存储在片上内存中,页面的访问速度可以得到保证,同时,备份页面与工作页面分开存储,使得备份页面不会与工作页面竞争缓存空间,更多的工作页面可以存放在缓存中,从而可以提高程序运行的性能;进一步地,通过主动触发的比较操作,及时更新共享虚拟内存页面的内容,保证在进行比较操作时,工作页面基本上都还位于cache中,不需要进行片外内存访问,因此Diff操作的速度很快;进一步地,通过备份页面资源池方案,可以使得进程间充分共享备份页面,进一步减少备份页面占用的片上内存空间,节约系统资源。

[0243] 图12示出了本发明实施例提供的一种计算节点的示意图,如图12所示,计算节点100包括:至少一个处理器1001、存储器1002和总线,其中处理器1001内部包含有片上内存1003。处理器1001和存储器1002通过总线连接并完成相互间的通信。该总线可以是工业标准体系结构(Industry Standard Architecture,简称为ISA)总线、外部设备互连(Peripheral Component,简称为PCI)总线或扩展工业标准体系结构(Extended Industry StandardArchitecture,简称为EISA)总线等。该总线可以分为地址总线、数据总线、控制总线等。为便于表示,图12中仅用一条粗线表示,但并不表示仅有一根总线或一种类型的总线。其中:

[0244] 存储器1002用于存储数据和可执行程序代码,该程序代码包括计算机操作指令。存储器1002可能包含高速RAM存储器,也可能还包括非易失性存储器(non-volatile memory),例如至少一个磁盘存储器。

[0245] 在一个实施例中,存储器1002存储有应用程序1对应的可执行代码,处理器1001通过读取存储器1002中存储的可执行程序代码来运行与所述可执行程序代码对应的程序,其中,应用程序1包括一个或多个进程,且应用程序1的各进程间拥有一个相同的共享虚拟内存,每个进程分别将共享虚拟内存中的部分或者全部虚拟内存页面映射到该进程维护的私有物理内存空间;进一步地,存储器1002还存储有操作系统对应可执行代码,处理器1001通过读取并执行存储器1002中的操作系统对应的可执行代码,以用于:

[0246] 将应用程序1的第一进程请求访问的共享虚拟内存页面拷贝至计算节点的存储器1002中,并将拷贝到存储器1002中的共享虚拟内存页面作为应用程序1的第一进程的工作页面;其中,第一进程请求访问的共享虚拟内存页面为应用程序1的共享虚拟内存中,第一

进程请求读或写的虚拟内存页面；

[0247] 在第一进程对该工作页面进行写操作之前，创建该工作页面的备份页面，并将创建的备份页面存储在片上内存1003中，以备份该工作页面的原始数据。

[0248] 其中，上述流程的具体细节可参照上述方法及装置实施例，此处不再赘述。

[0249] 需要说明的是，处理器1001可以是中央处理器(Central Processing Unit，简称为CPU)，或者是特定集成电路(Application Specific Integrated Circuit，简称为ASIC)，或者是被配置成实施本发明实施例的一个或多个集成电路，并且，处理器1001可以包括一个或者多个核，各个核共享处理器1001的片上内存1003。

[0250] 另外上述处理器1001除了执行上述方法流程之外，还可用于执行可执行代码，以实现本发明方法实施例中的其他步骤，在此不再赘述。

[0251] 本发明实施例通过以上技术方案，将进程请求读或写的共享虚拟内存页面先拷贝在计算节点片外内存中，作为可供进程进行读写操作的工作页面，同时利用计算节点的CPU的片上内存，在进程对工作页面进行写操作之前，将工作页面中的原始数据在片上内存中备份，以保证多个进程在对共享虚拟内存页面进行操作时的数据一致性，由于备份页面存储在片上内存中，页面的访问速度可以得到保证，同时，备份页面与工作页面分开存储，使得备份页面不会与工作页面竞争缓存空间，更多的工作页面可以存放在缓存中，从而可以提高程序运行的性能；进一步地，通过主动触发的比较操作，及时更新共享虚拟内存页面的内容，保证在进行比较操作时，工作页面基本上都还位于cache中，不需要进行片外内存访问，因此Diff操作的速度很快；进一步地，通过备份页面资源池方案，可以使得进程间充分共享备份页面，进一步减少备份页面占用的片上内存空间，节约系统资源。

[0252] 图13示出了本发明实施例提供的一种计算机系统的示意图，如图13所示，该计算机系统110包括：处理器1101、第一存储器1102、操作系统内核1103；其中，处理器1101内部包含有第二存储器1104和至少一个处理器核1105，处理器核用于运行应用程序，第二存储器1104是处理器1101的各个处理器核共用的片上存储，其数据存取速度大于第一存储器1102的数据存取速度；

[0253] 其中，操作系统内核1103，用于将所述应用程序的第一进程请求访问共享虚拟内存页面拷贝至第一存储器1102中，并将拷贝到第一存储器1102中的共享虚拟内存页面作为第一进程的工作页面；在第一进程对该工作页面进行写操作之前，创建该工作页面的备份页面，并将创建的备份页面存储在第二存储器1104中，以备份该工作页面的原始数据；其中，共享虚拟内存页面为第一进程所属应用程序的共享虚拟内存中的虚拟内存页面。

[0254] 在一个实施例中，共享虚拟内存页面的个数为N，N为大于或等于1的正整数；所述第一进程的工作页面的个数为M，M为大于或等于1的正整数；操作系统内核1103，还用于，在将创建的备份页面存储在所述第一存储器之前，判断第一存储器1102的剩余空间是否小于第一阈值，如果是，则触发第一进程将自身的M个工作页面中被修改的内容同步更新到该M个工作页面所对应的M个共享虚拟内存页面中，并将该M个工作页面在第一存储器1102中的备份页面所占用的空间释放；如果否，则执行将创建的所述备份页面存储在第二存储器1104中的步骤。

[0255] 进一步地，在另一个实施例中，第二存储器1104中保存有备份页面信息表，其中，所述备份页面信息表包含有第二存储器1104中所有备份页面的元数据信息，每一个备份页

面的元数据信息包括:所述每一个备份页面的页号和版本号,其中,所述每一个备份页面的页号和版本号分别与所述每一个备份页面所对应的工作页面的页号和版本号相同;

[0256] 在这种情形下,操作系统内核1103在创建工作页面的备份页面之前,会在该备份页面信息表中查找是否有页号和版本号分别与该工作页面的页号以及版本号相同的元数据信息,如果查找到,则将查找到的元数据信息所对应的备份页面为该工作页面的备份页面。

[0257] 可以理解的是,为了减少对第二存储器1104存储空间的占用,也可以将备份页面信息表保存在第一存储器1102中;在另一个实施例中,处理器1101中还包括有缓存(cache)1106,用于缓存各个处理器核的临时数据。另外,本发明实施例的操作系统管理装置1103的具体操作步骤,可以参见前述各个方法实施例,此处不再赘述。

[0258] 本发明实施例通过以上技术方案,将进程请求读或写的共享虚拟内存页面先拷贝在计算节点片外内存中,作为可供进程进行读写操作的工作页面,同时利用计算节点的CPU的片上内存,在进程对工作页面进行写操作之前,将工作页面中的原始数据在片上内存中备份,以保证多个进程在对共享虚拟内存页面进行操作时的数据一致性,由于备份页面存储在片上内存中,页面的访问速度可以得到保证,同时,备份页面与工作页面分开存储,使得备份页面不会与工作页面竞争缓存空间,更多的工作页面可以存放在缓存中,从而可以提高程序运行的性能;进一步地,通过主动触发的比较操作,及时更新共享虚拟内存页面的内容,保证在进行比较操作时,工作页面基本上都还位于cache中,不需要进行片外内存访问,因此比较操作的速度很快;进一步地,通过备份页面共享的方式,可以使得进程间充分共享备份页面,进一步减少备份页面占用的片上内存空间,节约系统资源。

[0259] 在本申请所提供的几个实施例中,应该理解到,所揭露的装置和方法,可以通过其它的方式实现。例如,以上所描述的装置实施例仅仅是示意性的。

[0260] 所述作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元显示的部件可以是或者也可以不是物理单元,即可以位于一个地方,或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

[0261] 另外,在本发明各个实施例提供的网络设备中的各功能单元可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现,也可以采用软件功能单元的形式实现。

[0262] 所述集成的单元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取存储介质中。基于这样的理解,本发明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全部或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计算机设备(可以是个人计算机,服务器,或者网络设备等)执行本发明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(ROM, Read-Only Memory)、随机存取存储器(RAM, Random Access Memory)、磁碟或者光盘等各种可以存储程序代码的介质。

[0263] 最后应说明的是:以上实施例仅用以说明本发明的技术方案,而非对其限制;尽管参照前述实施例对本发明进行了详细的说明,本领域的普通技术人员应当理解:其依然可

以对前述各实施例所记载的技术方案进行修改,或者对其中部分技术特征进行等同替换;而这些修改或者替换,并不使相应技术方案的本质脱离本发明各实施例技术方案的精神和范围。

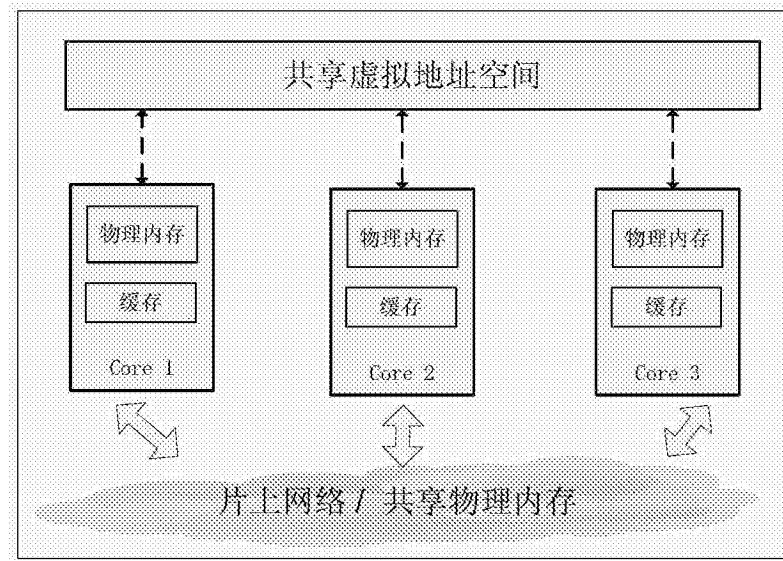


图1

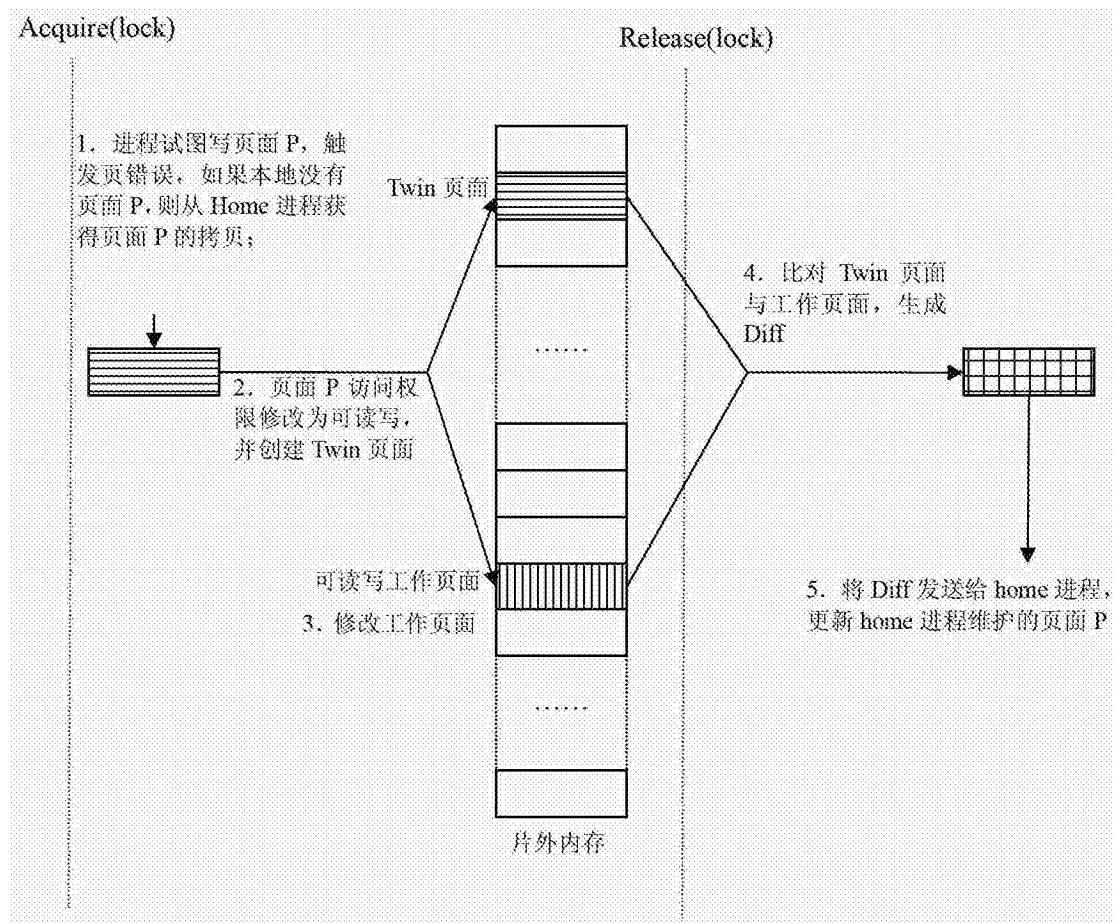


图2

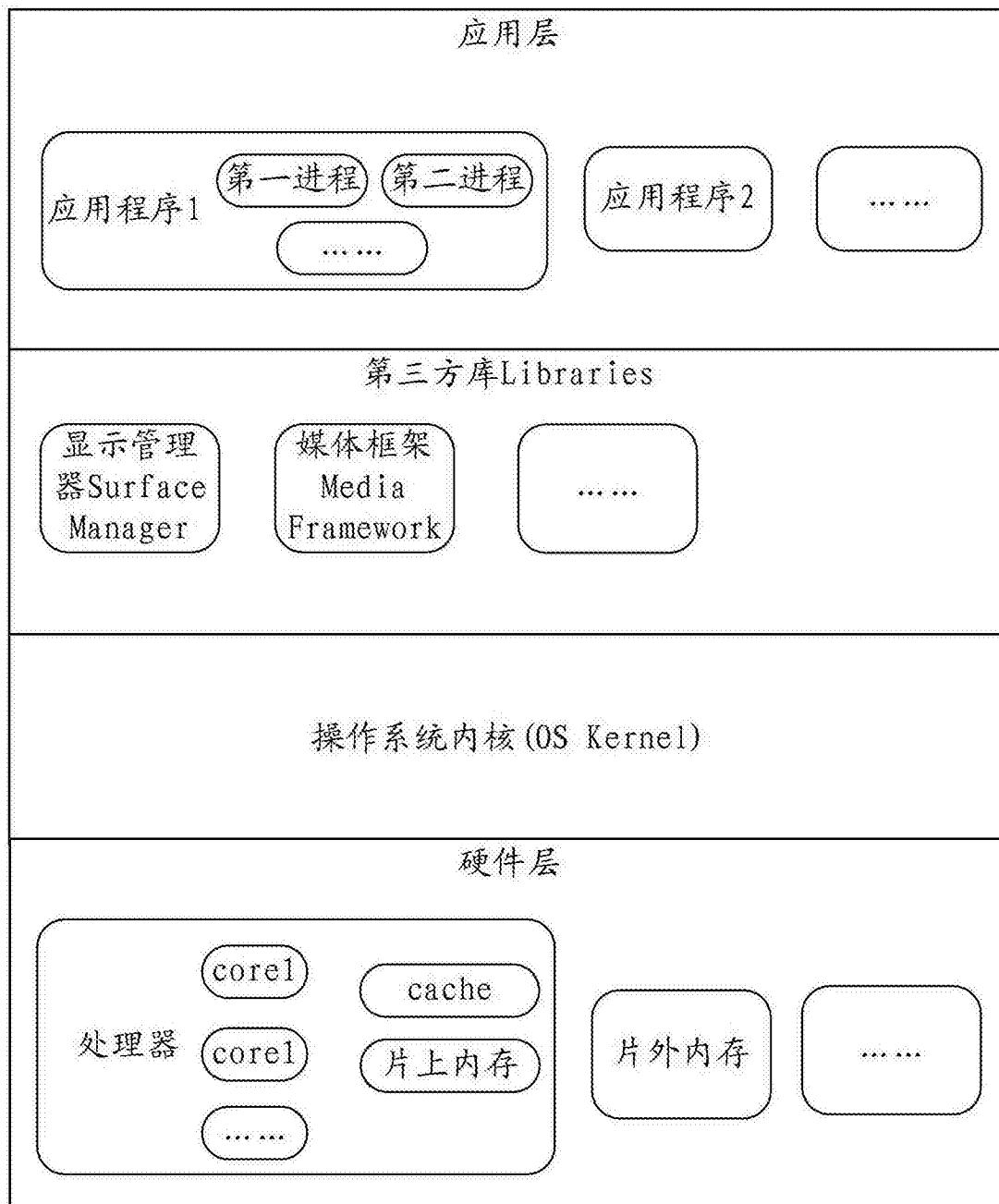


图3

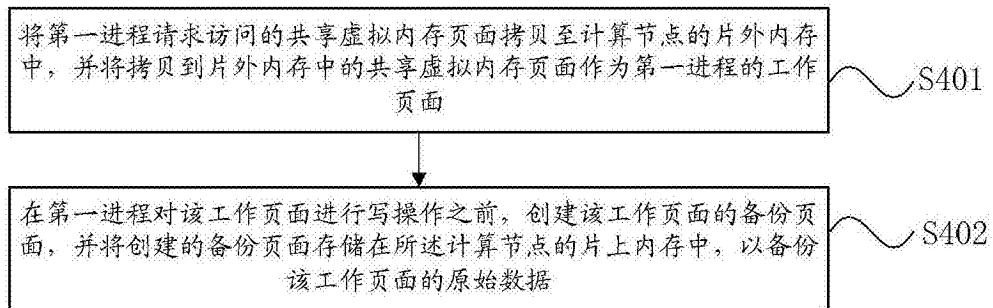


图4

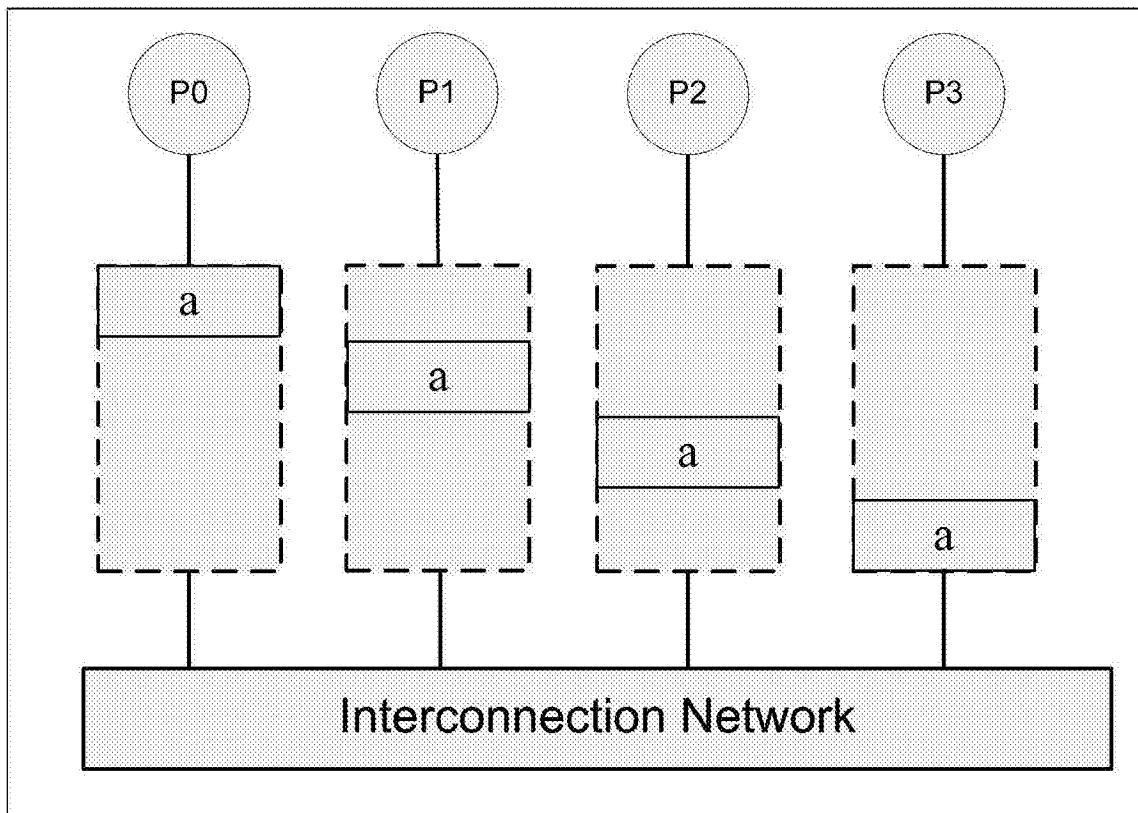


图5

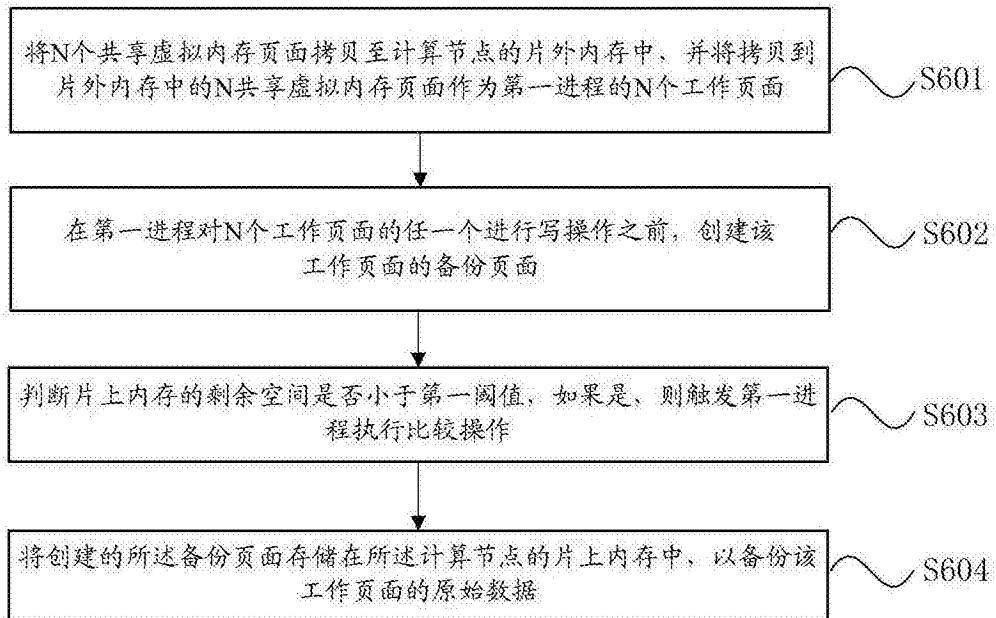


图6

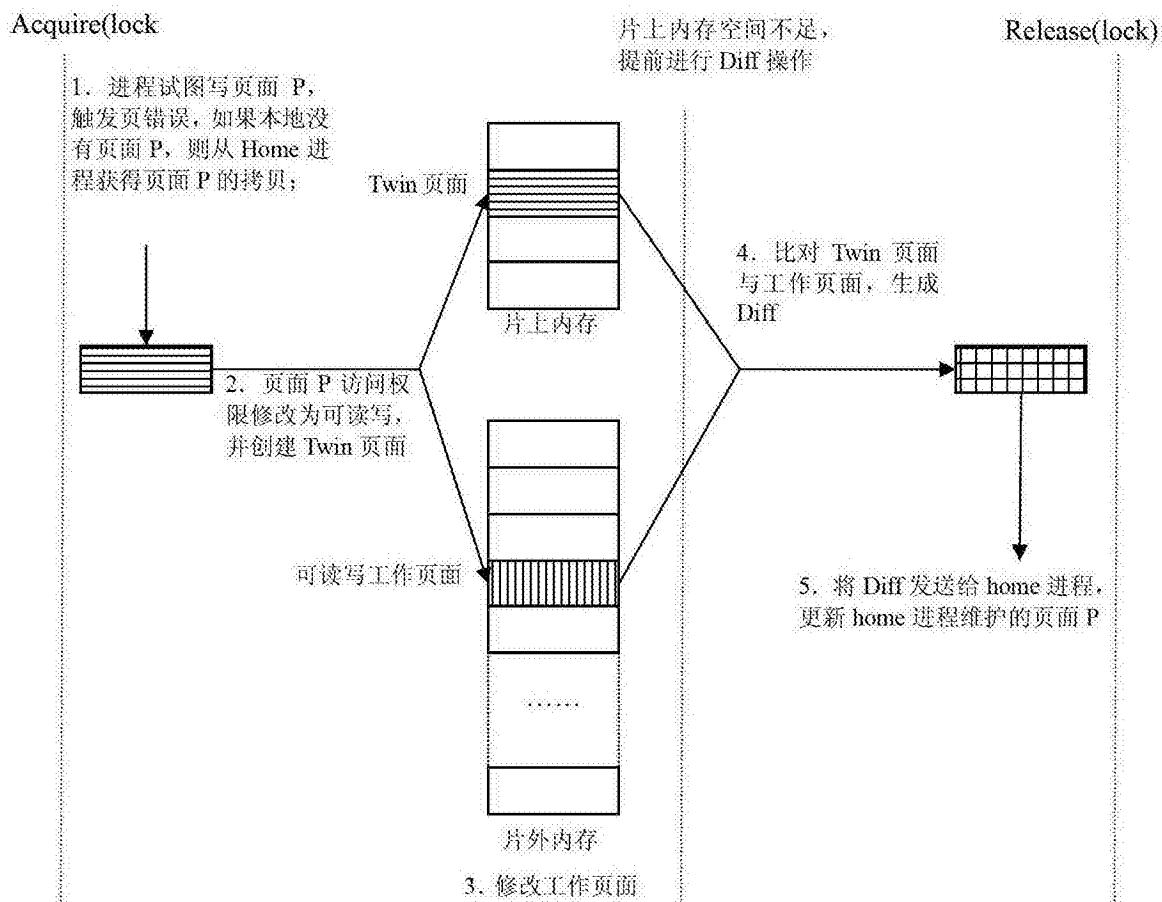


图7

Twin Page Id	Twin Page Version	Twin Page Usage	Twin Page Address
--------------	-------------------	-----------------	-------------------

图8

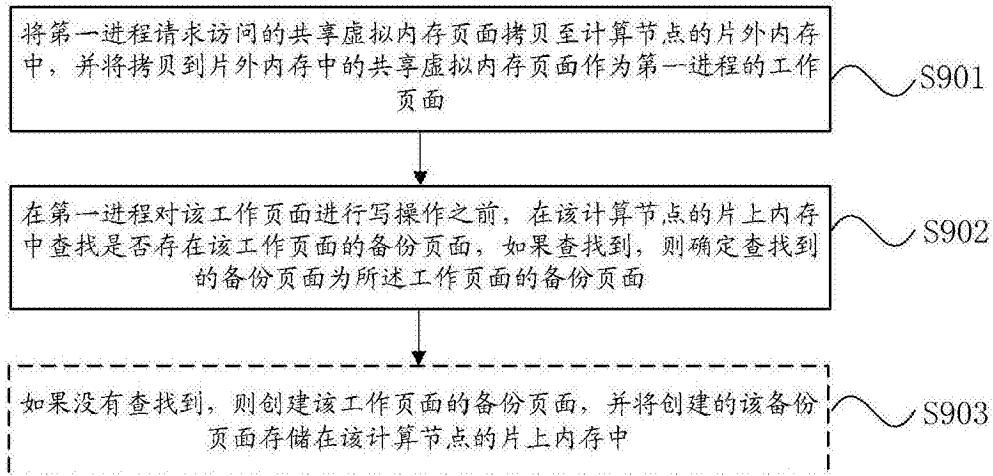


图9

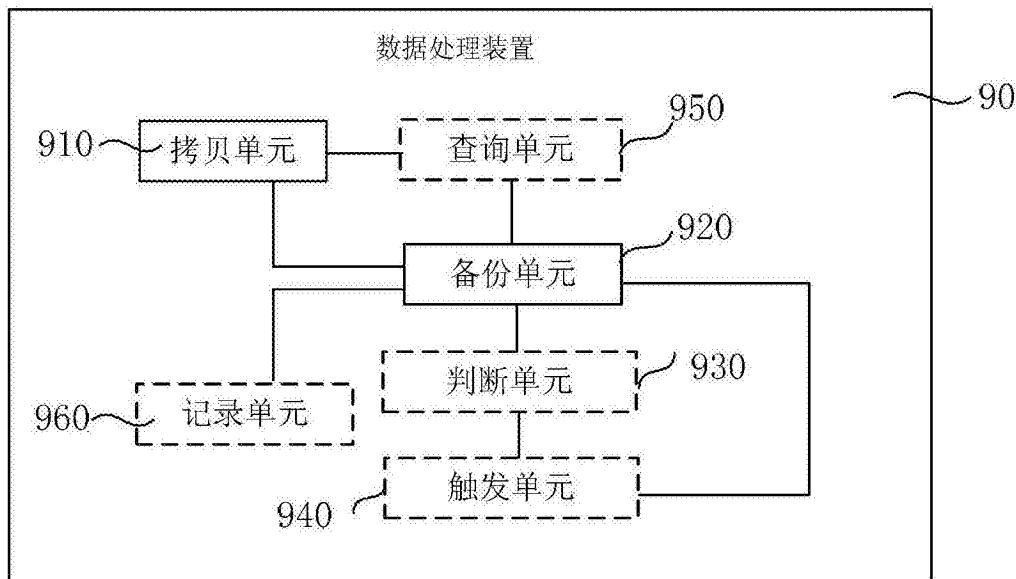


图10

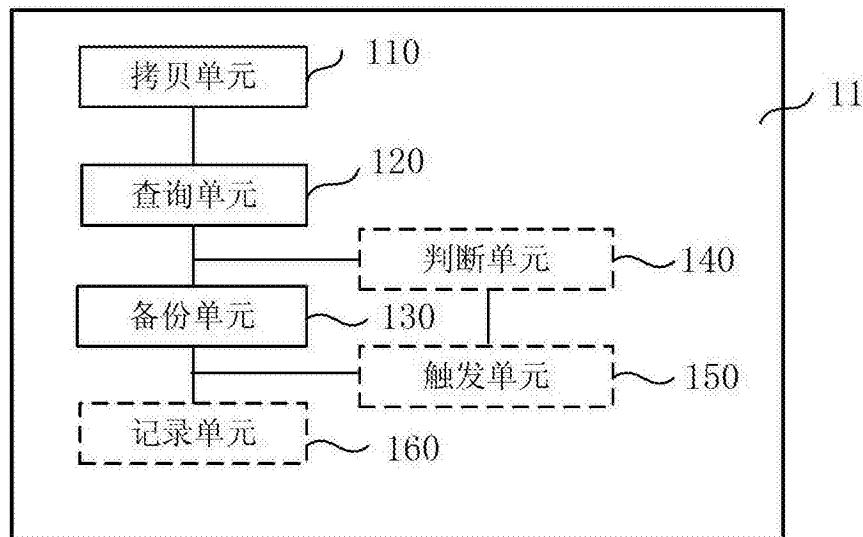


图11

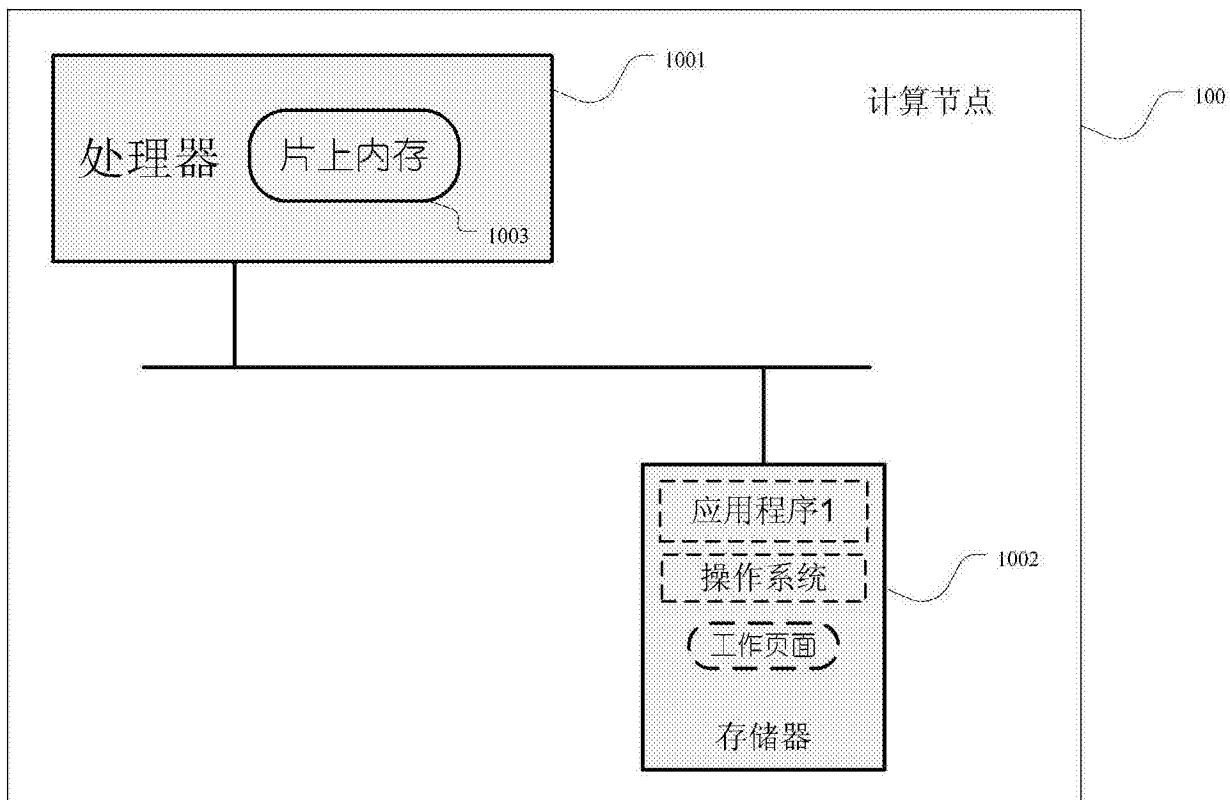


图12

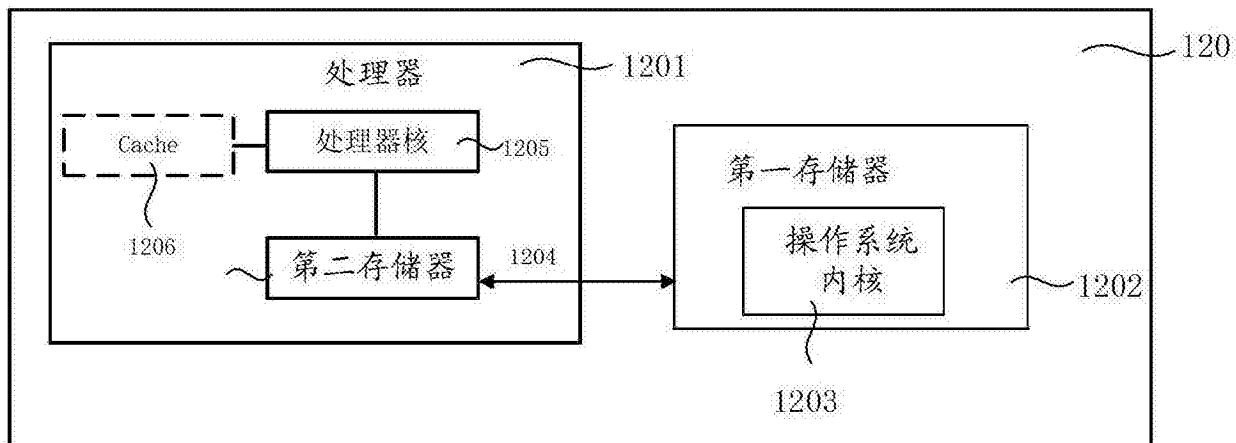


图13