



(12)发明专利

(10)授权公告号 CN 104063330 B

(45)授权公告日 2017.04.26

(21)申请号 201410290563.9

(22)申请日 2014.06.25

(65)同一申请的已公布的文献号  
申请公布号 CN 104063330 A

(43)申请公布日 2014.09.24

(73)专利权人 华为技术有限公司  
地址 518129 广东省深圳市龙岗区坂田华为总部办公楼

(72)发明人 曾华荣 朱春节 王学友 王芳

(74)专利代理机构 北京同立钧成知识产权代理有限公司 11205

代理人 刘芳

(51)Int.Cl.  
G06F 12/0862(2016.01)

(56)对比文件

EP 1149342 B1,2007.04.04,  
CN 103383666 A,2013.11.06,

审查员 李江

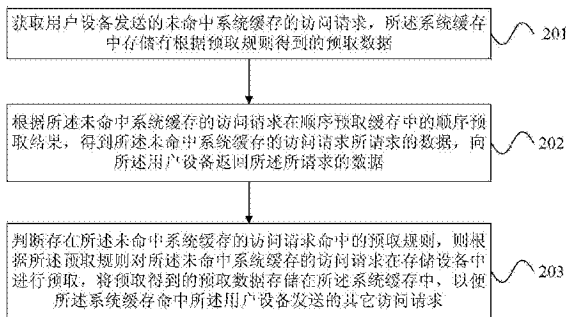
权利要求书6页 说明书11页 附图5页

(54)发明名称

数据预取方法及装置

(57)摘要

本发明实施例提供一种数据预取方法及装置,该方法包括获取用户设备发送的未命中系统缓存的访问请求,所述系统缓存中存储有根据预取规则得到的预取数据;根据所述未命中系统缓存的访问请求在顺序预取缓存中的顺序预取结果,得到所述未命中系统缓存的访问请求所请求的数据,向所述用户设备返回所述所请求的数据;判断存在所述未命中系统缓存的访问请求命中的预取规则,则根据所述预取规则对所述未命中系统缓存的访问请求在存储设备中进行预取,将预取得到的预取数据存储在上述系统缓存中,以便所述系统缓存命中所述用户设备发送的其它访问请求。本实施例可以提高缓存的命中率,提高存储系统的性能。



1. 一种数据预取方法,其特征在于,包括:

获取用户设备发送的未命中系统缓存的访问请求,所述系统缓存中存储有根据预取规则得到的预取数据;

根据所述未命中系统缓存的访问请求在顺序预取缓存中的顺序预取结果,得到所述未命中系统缓存的访问请求所请求的数据,向所述用户设备返回所述所请求的数据;

判断存在所述未命中系统缓存的访问请求命中的预取规则,则根据所述预取规则对所述未命中系统缓存的访问请求在存储设备中进行预取,将预取得到的预取数据存储在上述系统缓存中,以便所述系统缓存命中所述用户设备发送的其它访问请求。

2. 根据权利要求1所述的方法,其特征在于,所述判断是否存在所述未命中系统缓存的访问请求命中的预取规则之前,还包括:

获取预设数量的访问请求,得到访问请求序列,对所述访问请求序列进行分段,得到多个子访问请求序列;

对所述多个子访问请求序列进行简化,得到多个子简化访问请求序列,所述子访问请求序列的数量与所述子简化访问请求序列的数量相等;

根据所述子简化访问请求序列,确定预取规则子集,所述预取规则子集中包括多个预取规则。

3. 根据权利要求2所述的方法,其特征在于,所述对所述多个子访问请求序列进行简化,得到多个子简化访问请求序列,包括:

根据所述多个子访问请求序列,得到每个一次请求项的频度,确定大于第一预设频度的多个频繁一次请求项,每个一次请求项对应一个访问请求;

根据各所述频繁一次请求项在上述多个子访问请求序列中的位置,对所述多个子访问请求序列进行简化,得到多个子简化访问请求序列。

4. 根据权利要求2或3所述的方法,其特征在于,所述根据所述子简化访问请求序列,确定预取规则子集,包括:

根据所述多个子简化访问请求序列,得到每个二次请求项的频度,确定大于第二预设频度的多个频繁二次请求项,每个二次请求项对应两个访问请求;

根据各所述频繁二次请求项,确定预取规则子集。

5. 根据权利要求4所述的方法,其特征在于,所述根据所述子简化访问请求序列,确定预取规则子集之后,还包括:

将所述预取规则子集添加到预取规则集中,以更新所述预取规则集;

判断所述预取规则集中的预取规则子集是否超过预设数量;

若是,根据所述预取规则集中的各所述预取规则子集的使用率,删除所述预取规则集中所述使用率满足预设条件的预取规则子集。

6. 根据权利要求1至3任一项所述的方法,其特征在于,所述根据所述未命中系统缓存的访问请求在顺序预取缓存中的顺序预取结果,得到所述未命中系统缓存的访问请求所请求的数据,包括:

判断所述未命中系统缓存的访问请求是否命中顺序预取缓存中的顺序预取队列,所述顺序预取缓存包括多个顺序预取队列;

若是,则根据所述未命中系统缓存的访问请求命中当前预取窗口的结果,得到所述未

命中系统缓存的访问请求所请求的数据；

若否，则从所述存储设备中获取与所述未命中系统缓存的访问请求所请求的数据。

7. 根据权利要求4所述的方法，其特征在于，所述根据所述未命中系统缓存的访问请求在顺序预取缓存中的顺序预取结果，得到所述未命中系统缓存的访问请求所请求的数据，包括：

判断所述未命中系统缓存的访问请求是否命中顺序预取缓存中的顺序预取队列，所述顺序预取缓存包括多个顺序预取队列；

若是，则根据所述未命中系统缓存的访问请求命中当前预取窗口的结果，得到所述未命中系统缓存的访问请求所请求的数据；

若否，则从所述存储设备中获取与所述未命中系统缓存的访问请求所请求的数据。

8. 根据权利要求5所述的方法，其特征在于，所述根据所述未命中系统缓存的访问请求在顺序预取缓存中的顺序预取结果，得到所述未命中系统缓存的访问请求所请求的数据，包括：

判断所述未命中系统缓存的访问请求是否命中顺序预取缓存中的顺序预取队列，所述顺序预取缓存包括多个顺序预取队列；

若是，则根据所述未命中系统缓存的访问请求命中当前预取窗口的结果，得到所述未命中系统缓存的访问请求所请求的数据；

若否，则从所述存储设备中获取与所述未命中系统缓存的访问请求所请求的数据。

9. 根据权利要求6所述的方法，其特征在于，所述根据所述未命中系统缓存的访问请求命中当前预取窗口的结果，得到所述未命中系统缓存的访问请求所请求的数据，包括：

判断所述未命中系统缓存的访问请求是否命中所述预取窗口内的数据；

若是，则从所述预取窗口内得到所述未命中系统缓存的访问请求所请求的数据，并根据所述未命中系统缓存的访问请求所请求的数据从所述存储设备中预取数据到预取窗口中，使所述预取窗口的长度增加，将长度增加的所述预取窗口内的预取数据更新到所述预取窗口对应的顺序预取队列中；

若否，则从所述顺序预取缓存中的顺序预取队列中，得到所述未命中系统缓存的访问请求所请求的数据。

10. 根据权利要求7所述的方法，其特征在于，所述根据所述未命中系统缓存的访问请求命中当前预取窗口的结果，得到所述未命中系统缓存的访问请求所请求的数据，包括：

判断所述未命中系统缓存的访问请求是否命中所述预取窗口内的数据；

若是，则从所述预取窗口内得到所述未命中系统缓存的访问请求所请求的数据，并根据所述未命中系统缓存的访问请求所请求的数据从所述存储设备中预取数据到预取窗口中，使所述预取窗口的长度增加，将长度增加的所述预取窗口内的预取数据更新到所述预取窗口对应的顺序预取队列中；

若否，则从所述顺序预取缓存中的顺序预取队列中，得到所述未命中系统缓存的访问请求所请求的数据。

11. 根据权利要求8所述的方法，其特征在于，所述根据所述未命中系统缓存的访问请求命中当前预取窗口的结果，得到所述未命中系统缓存的访问请求所请求的数据，包括：

判断所述未命中系统缓存的访问请求是否命中所述预取窗口内的数据；

若是,则从所述预取窗口内得到所述未命中系统缓存的访问请求所请求的数据,并根据所述未命中系统缓存的访问请求所请求的数据从所述存储设备中预取数据到预取窗口中,使所述预取窗口的长度增加,将长度增加的所述预取窗口内的预取数据更新到所述预取窗口对应的顺序预取队列中;

若否,则从所述顺序预取缓存中的顺序预取队列中,得到所述未命中系统缓存的访问请求所请求的数据。

12. 根据权利要求6所述的方法,其特征在于,所述从所述存储设备中获取与所述未命中系统缓存的访问请求所请求的数据之后,还包括:

根据最少使用算法LRU选择需要淘汰的顺序预取队列,将所述需要淘汰的顺序预取队列中命中过访问请求的数据缓存到系统缓存中,将所述需要淘汰的顺序预取队列中未命中过访问请求的数据丢弃;

将所述未命中系统缓存的访问请求所请求的数据与预取窗口内的数据放入所述需要淘汰的顺序预取队列中,得到新的顺序预取队列。

13. 根据权利要求7所述的方法,其特征在于,所述从所述存储设备中获取与所述未命中系统缓存的访问请求所请求的数据之后,还包括:

根据最少使用算法LRU选择需要淘汰的顺序预取队列,将所述需要淘汰的顺序预取队列中命中过访问请求的数据缓存到系统缓存中,将所述需要淘汰的顺序预取队列中未命中过访问请求的数据丢弃;

将所述未命中系统缓存的访问请求所请求的数据与预取窗口内的数据放入所述需要淘汰的顺序预取队列中,得到新的顺序预取队列。

14. 根据权利要求8所述的方法,其特征在于,所述从所述存储设备中获取与所述未命中系统缓存的访问请求所请求的数据之后,还包括:

根据最少使用算法LRU选择需要淘汰的顺序预取队列,将所述需要淘汰的顺序预取队列中命中过访问请求的数据缓存到系统缓存中,将所述需要淘汰的顺序预取队列中未命中过访问请求的数据丢弃;

将所述未命中系统缓存的访问请求所请求的数据与预取窗口内的数据放入所述需要淘汰的顺序预取队列中,得到新的顺序预取队列。

15. 一种数据预取装置,其特征在于,包括:

获取模块,用于获取用户设备发送的未命中系统缓存的访问请求,所述系统缓存中存储有根据预取规则得到的预取数据;

处理模块,用于根据所述未命中系统缓存的访问请求在顺序预取缓存中的顺序预取结果,得到所述未命中系统缓存的访问请求所请求的数据,向所述用户设备返回所述所请求的数据;

判断模块,用于判断存在所述未命中系统缓存的访问请求命中的预取规则,则根据所述预取规则对所述未命中系统缓存的访问请求在存储设备中进行预取,将预取得到的预取数据存储到所述系统缓存中,以便所述系统缓存命中所述用户设备发送的其它访问请求。

16. 根据权利要求15所述的装置,其特征在于,还包括:

分段模块,用于在所述判断模块判断是否存在所述未命中系统缓存的访问请求命中的预取规则之前,获取预设数量的访问请求,得到访问请求序列,对所述访问请求序列进行分

段,得到多个子访问请求序列;

序列简化模块,用于对所述多个子访问请求序列进行简化,得到多个子简化访问请求序列,所述子访问请求序列的数量与所述子简化访问请求序列的数量相等;

规则生成模块,用于根据所述子简化访问请求序列,确定预取规则子集,所述预取规则子集中包括多个预取规则。

17. 根据权利要求16所述的装置,其特征在于,所述序列简化模块具体用于:

根据所述多个子访问请求序列,得到每个一次请求项的频度,确定大于第一预设频度的多个频繁一次请求项,每个一次请求项对应一个访问请求;

根据各所述频繁一次请求项在所述多个子访问请求序列中的位置,对所述多个子访问请求序列进行简化,得到多个子简化访问请求序列。

18. 根据权利要求16或17所述的装置,其特征在于,所述规则生成模块具体用于:

根据所述多个子简化访问请求序列,得到每个二次请求项的频度,确定大于第二预设频度的多个频繁二次请求项,每个二次请求项对应两个访问请求;

根据各所述频繁二次请求项,确定预取规则子集。

19. 根据权利要求18所述的装置,其特征在于,还包括规则更新模块,用于在所述规则生成模块根据所述子简化访问请求序列,确定预取规则子集之后,

将所述预取规则子集添加到预取规则集中,以更新所述预取规则集;

判断所述预取规则集中的预取规则子集是否超过预设数量;

若是,根据所述预取规则集中的各所述预取规则子集的使用率,删除所述预取规则集中所述使用率满足预设条件的预取规则子集。

20. 根据权利要求15至17任一项所述的装置,其特征在于,所述处理模块具体用于:

判断所述未命中系统缓存的访问请求是否命中顺序预取缓存中的顺序预取队列,所述顺序预取缓存包括多个顺序预取队列;

若是,则根据所述未命中系统缓存的访问请求命中当前预取窗口的结果,得到所述未命中系统缓存的访问请求所请求的数据;

若否,则从所述存储设备中获取与所述未命中系统缓存的访问请求所请求的数据。

21. 根据权利要求18所述的装置,其特征在于,所述处理模块具体用于:

判断所述未命中系统缓存的访问请求是否命中顺序预取缓存中的顺序预取队列,所述顺序预取缓存包括多个顺序预取队列;

若是,则根据所述未命中系统缓存的访问请求命中当前预取窗口的结果,得到所述未命中系统缓存的访问请求所请求的数据;

若否,则从所述存储设备中获取与所述未命中系统缓存的访问请求所请求的数据。

22. 根据权利要求19所述的装置,其特征在于,所述处理模块具体用于:

判断所述未命中系统缓存的访问请求是否命中顺序预取缓存中的顺序预取队列,所述顺序预取缓存包括多个顺序预取队列;

若是,则根据所述未命中系统缓存的访问请求命中当前预取窗口的结果,得到所述未命中系统缓存的访问请求所请求的数据;

若否,则从所述存储设备中获取与所述未命中系统缓存的访问请求所请求的数据。

23. 根据权利要求20所述的装置,其特征在于,所述处理模块还具体用于:

判断所述未命中系统缓存的访问请求是否命中所述预取窗口内的数据；

若是，则从所述预取窗口内得到所述未命中系统缓存的访问请求所请求的数据，并根据所述未命中系统缓存的访问请求所请求的数据从所述存储设备中预取数据到预取窗口中，使所述预取窗口的长度增加，将长度增加的所述预取窗口内的预取数据更新到所述预取窗口对应的顺序预取队列中；

若否，则从所述顺序预取缓存中的顺序预取队列中，得到所述未命中系统缓存的访问请求所请求的数据。

24. 根据权利要求21所述的装置，其特征在于，所述处理模块还具体用于：

判断所述未命中系统缓存的访问请求是否命中所述预取窗口内的数据；

若是，则从所述预取窗口内得到所述未命中系统缓存的访问请求所请求的数据，并根据所述未命中系统缓存的访问请求所请求的数据从所述存储设备中预取数据到预取窗口中，使所述预取窗口的长度增加，将长度增加的所述预取窗口内的预取数据更新到所述预取窗口对应的顺序预取队列中；

若否，则从所述顺序预取缓存中的顺序预取队列中，得到所述未命中系统缓存的访问请求所请求的数据。

25. 根据权利要求22所述的装置，其特征在于，所述处理模块还具体用于：

判断所述未命中系统缓存的访问请求是否命中所述预取窗口内的数据；

若是，则从所述预取窗口内得到所述未命中系统缓存的访问请求所请求的数据，并根据所述未命中系统缓存的访问请求所请求的数据从所述存储设备中预取数据到预取窗口中，使所述预取窗口的长度增加，将长度增加的所述预取窗口内的预取数据更新到所述预取窗口对应的顺序预取队列中；

若否，则从所述顺序预取缓存中的顺序预取队列中，得到所述未命中系统缓存的访问请求所请求的数据。

26. 根据权利要求20所述的装置，其特征在于，还包括：队列更新模块，用于在所述处理模块从所述存储设备中获取与所述未命中系统缓存的访问请求所请求的数据之后，

根据最少使用算法LRU选择需要淘汰的顺序预取队列，将所述需要淘汰的顺序预取队列中命中过访问请求的数据缓存到系统缓存中，将所述需要淘汰的顺序预取队列中未命中过访问请求的数据丢弃；

将所述未命中系统缓存的访问请求所请求的数据与预取窗口内的数据放入所述需要淘汰的顺序预取队列中，得到新的顺序预取队列。

27. 根据权利要求21所述的装置，其特征在于，还包括：队列更新模块，用于在所述处理模块从所述存储设备中获取与所述未命中系统缓存的访问请求所请求的数据之后，

根据最少使用算法LRU选择需要淘汰的顺序预取队列，将所述需要淘汰的顺序预取队列中命中过访问请求的数据缓存到系统缓存中，将所述需要淘汰的顺序预取队列中未命中过访问请求的数据丢弃；

将所述未命中系统缓存的访问请求所请求的数据与预取窗口内的数据放入所述需要淘汰的顺序预取队列中，得到新的顺序预取队列。

28. 根据权利要求22所述的装置，其特征在于，还包括：队列更新模块，用于在所述处理模块从所述存储设备中获取与所述未命中系统缓存的访问请求所请求的数据之后，

根据最少使用算法LRU选择需要淘汰的顺序预取队列,将所述需要淘汰的顺序预取队列中命中过访问请求的数据缓存到系统缓存中,将所述需要淘汰的顺序预取队列中未命中过访问请求的数据丢弃;

将所述未命中系统缓存的访问请求所请求的数据与预取窗口内的数据放入所述需要淘汰的顺序预取队列中,得到新的顺序预取队列。

## 数据预取方法及装置

### 技术领域

[0001] 本发明实施例涉及存储技术,尤其涉及一种数据预取方法及装置。

### 背景技术

[0002] 随着计算机技术的迅速发展,计算机硬件的处理速度提高非常快。然而,存储设备的性能提升却始终跟不上计算机硬件的处理速度,并且两者的距离在逐渐增大。为了提高存储设备的性能,在读取数据时,基于预取的方式,先将请求的数据读取到缓存中,从而有效减少响应时间,提高命中率。

[0003] 现有技术中,最常用的预取方式为顺序预取方式,适用于顺序性比较好的负载。在顺序预取方式中,假设所有的访问请求行为都是顺序的,当一个访问请求到达时,如果该访问请求满足顺序性判定条件,则在缓存中预取当前访问请求之后的数据。

[0004] 然而,由于顺序预取是在假设的前提下去做的,当负载中的顺序性访问特征比较弱时,顺序预取不仅不能帮助提高存储系统的性能,还会导致存储系统会比正常情况下多预取更多的无效数据到缓存,降低存储系统的性能。

### 发明内容

[0005] 本发明实施例提供一种数据预取方法及装置,以提高存储系统的性能。

[0006] 第一方面,本发明实施例提供一种数据预取方法,包括:

[0007] 获取用户设备发送的未命中系统缓存的访问请求,所述系统缓存中存储有根据预取规则得到的预取数据;

[0008] 根据所述未命中系统缓存的访问请求在顺序预取缓存中的顺序预取结果,得到所述未命中系统缓存的访问请求所请求的数据,向所述用户设备返回所述所请求的数据;

[0009] 判断存在所述未命中系统缓存的访问请求命中的预取规则,则根据所述预取规则对所述未命中系统缓存的访问请求在存储设备中进行预取,将预取得到的预取数据存储在所述系统缓存中,以便所述系统缓存命中所述用户设备发送的其它访问请求。

[0010] 结合第一方面,在第一方面的第一种可能的实现方式中,所述判断是否存在所述未命中系统缓存的访问请求命中的预取规则之前,还包括:

[0011] 获取预设数量的访问请求,得到访问请求序列,对所述访问请求序列进行分段,得到多个子访问请求序列;

[0012] 对所述多个子访问请求序列进行简化,得到多个子简化访问请求序列,所述子访问请求序列的数量与所述子简化访问请求序列的数量相等;

[0013] 根据所述子简化访问请求序列,确定预取规则子集,所述预取规则子集中包括多个预取规则。

[0014] 结合第一方面的第一种可能的实现方式,在第一方面的第二种可能的实现方式中,所述对所述多个子访问请求序列进行简化,得到多个子简化访问请求序列,包括:

[0015] 根据所述多个子访问请求序列,得到每个一次请求项的频度,确定大于第一预设



频度的多个频繁一次请求项,每个一次请求项对应一个访问请求;

[0016] 根据各所述频繁一次请求项在所述多个子访问请求序列中的位置,对所述多个子访问请求序列进行简化,得到多个子简化访问请求序列。

[0017] 结合第一方面的第一种或第二种可能的实现方式,在第一方面的第三种可能的实现方式中,所述根据所述子简化访问请求序列,确定预取规则子集,包括:

[0018] 根据所述多个子简化访问序列,得到每个二次请求项的频度,确定大于第二预设频度的多个频繁二次请求项,每个二次请求项对应两个访问请求;

[0019] 根据各所述频繁二次请求项,确定预取规则子集。

[0020] 结合第一方面的第三种可能的实现方式,在第一方面的第四种可能的实现方式中,所述根据所述子简化访问请求序列,确定预取规则子集之后,还包括:

[0021] 将所述预取规则子集添加到预取规则集中,以更新所述预取规则集;

[0022] 判断所述预取规则集中的预取规则子集是否超过预设数量;

[0023] 若是,根据所述预取规则集中的各所述预取规则子集的使用率,删除所述预取规则集中所述使用率满足预设条件的预取规则子集。

[0024] 结合第一方面、第一方面的第一种至第四种任一种可能的实现方式,在第一方面的第五种可能的实现方式中,所述根据所述未命中系统缓存的访问请求在顺序预取缓存中的顺序预取结果,得到所述未命中系统缓存的访问请求所请求的数据,包括:

[0025] 判断所述未命中系统缓存的访问请求是否命中顺序预取缓存中的顺序预取队列,所述顺序预取缓存包括多个顺序预取队列;

[0026] 若是,则根据所述未命中系统缓存的访问请求命中当前预取窗口的结果,得到所述未命中系统缓存的访问请求所请求的数据;

[0027] 若否,则从所述存储设备中获取与所述未命中系统缓存的访问请求所请求的数据。

[0028] 结合第一方面的第五种可能的实现方式中,在第一方面的第六种可能的实现方式中,所述根据所述未命中系统缓存的访问请求命中当前预取窗口的结果,得到所述未命中系统缓存的访问请求所请求的数据,包括:

[0029] 判断所述未命中系统缓存的访问请求是否命中所述预取窗口内的数据;

[0030] 若是,则从所述预取窗口内得到所述未命中系统缓存的访问请求所请求的数据,并根据所述未命中系统缓存的访问请求所请求的数据从所述存储设备中预取数据到预取窗口中,使所述预取窗口的长度增加,将长度增加的所述预取窗口内的预取数据更新到所述预取窗口对应的顺序预取队列中;

[0031] 若否,则从所述顺序预取缓存中的顺序预取队列中,得到所述未命中系统缓存的访问请求所请求的数据。

[0032] 结合第一方面的第五种可能的实现方式中,在第一方面的第七种可能的实现方式中,所述从所述存储设备中获取与所述未命中系统缓存的访问请求所请求的数据之后,还包括:

[0033] 根据最少使用算法LRU选择需要淘汰的顺序预取队列,将所述需要淘汰的顺序预取队列中命中过访问请求的数据缓存到系统缓存中,将所述需要淘汰的顺序预取队列中未命中过访问请求的数据丢弃;

[0034] 将所述未命中系统缓存的访问请求所请求的数据与预取窗口内的数据放入所述需要淘汰的顺序预取队列中,得到新的顺序预取队列。

[0035] 第二方面,本发明实施例提供一种数据预取装置,包括:

[0036] 获取模块,用于获取用户设备发送的未命中系统缓存的访问请求,所述系统缓存中存储有根据预取规则得到的预取数据;

[0037] 处理模块,用于根据所述未命中系统缓存的访问请求在顺序预取缓存中的顺序预取结果,得到所述未命中系统缓存的访问请求所请求的数据,向所述用户设备返回所述所请求的数据;

[0038] 判断模块,用于判断存在所述未命中系统缓存的访问请求命中的预取规则,则根据所述预取规则对所述未命中系统缓存的访问请求在存储设备中进行预取,将预取得到的预取数据存储在该系统缓存中,以便所述系统缓存命中所述用户设备发送的其它访问请求。

[0039] 结合第二方面,在第二方面的第一种可能的实现方式中,还包括:

[0040] 分段模块,用于在所述判断模块判断是否存在所述未命中系统缓存的访问请求命中的预取规则之前,获取预设数量的访问请求,得到访问请求序列,对所述访问请求序列进行分段,得到多个子访问请求序列;

[0041] 序列简化模块,用于对所述多个子访问请求序列进行简化,得到多个子简化访问请求序列,所述子访问请求序列的数量与所述子简化访问请求序列的数量相等;

[0042] 规则生成模块,用于根据所述子简化访问请求序列,确定预取规则子集,所述预取规则子集中包括多个预取规则。

[0043] 结合第二方面的第一种可能的实现方式,在第二方面的第二种可能的实现方式中,所述序列简化模块具体用于:

[0044] 根据所述多个子访问请求序列,得到每个一次请求项的频度,确定大于第一预设频度的多个频繁一次请求项,每个一次请求项对应一个访问请求;

[0045] 根据各所述频繁一次请求项在所述多个子访问请求序列中的位置,对所述多个子访问请求序列进行简化,得到多个子简化访问请求序列。

[0046] 结合第二方面的第一种或第二种可能的实现方式,在第二方面的第三种可能的实现方式中,所述规则生成模块具体用于:

[0047] 根据所述多个子简化访问序列,得到每个二次请求项的频度,确定大于第二预设频度的多个频繁二次请求项,每个二次请求项对应两个访问请求;

[0048] 根据各所述频繁二次请求项,确定预取规则子集。

[0049] 结合第二方面的第三种可能的实现方式,在第二方面的第四种可能的实现方式中,还包括规则更新模块,用于在所述规则生成模块根据所述子简化访问请求序列,确定预取规则子集之后,

[0050] 将所述预取规则子集添加到预取规则集中,以更新所述预取规则集;

[0051] 判断所述预取规则集中的预取规则子集是否超过预设数量;

[0052] 若是,根据所述预取规则集中的各所述预取规则子集的使用率,删除所述预取规则集中所述使用率满足预设条件的预取规则子集。

[0053] 结合第二方面、第二方面的第一种至第四种任一种可能的实现方式,在第二方面

的第五种可能的实现方式中,所述处理模块具体用于:

[0054] 判断所述未命中系统缓存的访问请求是否命中顺序预取缓存中的顺序预取队列,所述顺序预取缓存包括多个顺序预取队列;

[0055] 若是,则根据所述未命中系统缓存的访问请求命中当前预取窗口的结果,得到所述未命中系统缓存的访问请求所请求的数据;

[0056] 若否,则从所述存储设备中获取与所述未命中系统缓存的访问请求所请求的数据。

[0057] 结合第二方面的第五种可能的实现方式中,在第二方面的第六种可能的实现方式中,所述处理模块还具体用于:

[0058] 判断所述未命中系统缓存的访问请求是否命中所述预取窗口内的数据;

[0059] 若是,则从所述预取窗口内得到所述未命中系统缓存的访问请求所请求的数据,并根据所述未命中系统缓存的访问请求所请求的数据从所述存储设备中预取数据到预取窗口中,使所述预取窗口的长度增加,将长度增加的所述预取窗口内的预取数据更新到所述预取窗口对应的顺序预取队列中;

[0060] 若否,则从所述顺序预取缓存中的顺序预取队列中,得到所述未命中系统缓存的访问请求所请求的数据。

[0061] 结合第二方面的第五种可能的实现方式中,在第二方面的第七种可能的实现方式中,还包括:队列更新模块,用于在所述处理模块从所述存储设备中获取与所述未命中系统缓存的访问请求所请求的数据之后,

[0062] 根据最少使用算法LRU选择需要淘汰的顺序预取队列,将所述需要淘汰的顺序预取队列中命中过访问请求的数据缓存到系统缓存中,将所述需要淘汰的顺序预取队列中未命中过访问请求的数据丢弃;

[0063] 将所述未命中系统缓存的访问请求所请求的数据与预取窗口内的数据放入所述需要淘汰的顺序预取队列中,得到新的顺序预取队列。

[0064] 本发明实施例提供的数据预取方法及装置,该方法通过获取用户设备发送的未命中系统缓存的访问请求,系统缓存中存储有根据预取规则得到的预取数据;根据未命中系统缓存的访问请求在顺序预取缓存中的顺序预取结果,得到未命中系统缓存的访问请求所请求的数据,向用户设备返回所请求的数据;判断存在未命中系统缓存的访问请求命中的预取规则,则根据预取规则对未命中系统缓存的访问请求在存储设备中进行预取,将预取得到的预取数据存储存储在系统缓存中,以便系统缓存命中用户设备发送的其它访问请求,将顺序预取与预取规则结合起来,由于预取规则主要从数据块之间的相互关系入手,得到的预取数据之间的关联关系可以是连续的,也可以是非连续的,而顺序预取对于连续的预取数据具有更好的发掘能力,两者的结合可以提高缓存的命中率。

## 附图说明

[0065] 为了更清楚地说明本发明实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作一简单地介绍,显而易见地,下面描述中的附图是本发明的一些实施例,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其他的附图。

- [0066] 图1为本发明数据预取方法应用场景示意图；  
[0067] 图2为本发明数据预取方法实施例一的流程示意图；  
[0068] 图3为本发明数据预取方法实施例二的流程示意图；  
[0069] 图4为本发明实施例数据挖掘算法挖掘过程示意图；  
[0070] 图5为本发明预取规则子集删除示意图；  
[0071] 图6为本发明数据预取方法实施例三的流程示意图；  
[0072] 图7为本发明数据预取装置实施例一的结构示意图；  
[0073] 图8为本发明数据预取装置实施例二的结构示意图。

### 具体实施方式

[0074] 为使本发明实施例的目的、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，对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例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作出创造性劳动前提下所获得的所有其他实施例，都属于本发明保护的范围。

[0075] 图1为本发明数据预取方法应用场景示意图。如图1所示，上层用户在使用oracle、DB2等数据库或xfs、zfs等文件系统时，在下层会产生一系列的访问请求，该访问请求将会到达底层的存储设备。图1中粗箭头代表在缓存系统中引入输入/输出(Input/Output,简称IO) 轨迹学习算法之后的IO路径，当上层应用产生访问请求后，首先会查找系统缓存，如果在系统缓存中命中，则把该访问请求的数据返回给用户，同时轨迹学习模块获取该访问请求的元数据信息，供数据挖掘算法使用。如果系统缓存不命中，则直接把该访问请求发送给轨迹学习模块，然后轨迹学习模块会根据当前访问请求给出预取建议，发到底层的存储设备，然后存储设备将该访问请求的数据返回给系统缓存或轨迹学习模块的内部缓存，存储设备同时将该访问请求对应的数据返回给上层用户。

[0076] 轨迹学习模块涉及到的缓存包括两部分，一部分是系统缓存，另一部分是轨迹学习模块维护的一块缓存，由于这块缓存是顺序预取专用的，所以把这块缓存命名为顺序预取缓存。当顺序预取缓存放满时，它的一部分数据需要淘汰到系统缓存中，另一部分数据被直接丢弃。

[0077] 在系统缓存中主要存储三部分数据，一部分为顺序预取缓存放满时，顺序预取缓存将一部分数据淘汰到系统缓存中，另一部分为数据挖掘算法得到的预取规则，通过预取规则从存储设备中获取的预取数据，又一部分为正常运行过程中系统缓存中存储的数据。

[0078] 下面采用具体的实施例，对本发明数据预取方法进行详细说明。

[0079] 图2为本发明数据预取方法实施例一的流程示意图。本实施例可由任意的数据预取装置实现，该装置可通过软件和/或硬件实现。该方法包括：

[0080] 步骤201、获取用户设备发送的未命中系统缓存的访问请求，所述系统缓存中存储有根据预取规则得到的预取数据；

[0081] 步骤202、根据所述未命中系统缓存的访问请求在顺序预取缓存中的顺序预取结果，得到所述未命中系统缓存的访问请求所请求的数据，向所述用户设备返回所述所请求的数据；

[0082] 步骤203、判断存在所述未命中系统缓存的访问请求命中的预取规则，则根据所述

预取规则对所述未命中系统缓存的访问请求在存储设备中进行预取,将预取得到的预取数据存储在所述系统缓存中,以便所述系统缓存命中所述用户设备发送的其它访问请求。

[0083] 在步骤201中,用户设备发送的访问请求,先到系统缓存中,以确定系统缓存中是否存储有与该访问请求对应的数据,若系统缓存中没有存储有该访问请求对应的数据,则该访问请求未命中系统缓存,该访问请求则为未命中系统缓存的访问请求。此时,执行步骤202。

[0084] 在步骤202中,对该未命中系统缓存的访问请求在顺序预取缓存中执行顺序预取流程,在顺序预取流程中,通过顺序预取算法得到未命中系统缓存的访问请求所请求的数据,向用户设备返回所请求的数据。

[0085] 在步骤203中,为了后续能够提高系统缓存的命中率,需要进一步判断是否存在未命中系统缓存的访问请求命中的预取规则,本实施例的预取规则主要从数据块之间的相互关系入手,得到的预取数据之间的关联关系可以是连续的,也可以是非连续的。当命中预取规则时,根据预取规则对未命中系统缓存的访问请求在存储设备中进行预取,将预取得到的预取数据存储在系统缓存中。例如,针对任意的一个未命中系统缓存的访问请求,当该未命中系统缓存的访问请求针对10时,查找预取规则,当预取规则对应的数据为10、14、15时,则该未命中系统缓存的访问请求命中该预取规则,根据该预取规则,将10、14、15写入系统缓存中。在后续的其他访问请求中,当访问请求针对14时,将直接从系统缓存中获取14。

[0086] 本发明实施例提供的预取数据方法,通过获取用户设备发送的未命中系统缓存的访问请求,系统缓存中存储有根据预取规则得到的预取数据;根据未命中系统缓存的访问请求在顺序预取缓存中的顺序预取结果,得到未命中系统缓存的访问请求所请求的数据,向用户设备返回所请求的数据;判断存在未命中系统缓存的访问请求命中的预取规则,则根据预取规则对未命中系统缓存的访问请求在存储设备中进行预取,将预取得到的预取数据存储在系统缓存中,以便系统缓存命中用户设备发送的其它访问请求,将顺序预取与预取规则结合起来,由于预取规则主要从数据块之间的相互关系入手,得到的预取数据之间的关联关系可以是连续的,也可以是非连续的,而顺序预取对于连续的预取数据具有更好的发掘能力,两者的结合可以提高缓存的命中率。

[0087] 下面采用详细的实施例,详细说明如何获取预取规则,以及如何通过预取顺序缓存得到未命中系统缓存的访问请求所请求的数据。

[0088] 图3为本发明数据预取方法实施例二的流程示意图。本实施例主要说明图2实施例中,如何获取预取规则的方法。为了形象说明本实施例中预取规则的获取方法,并结合图4。图4为本发明实施例数据挖掘算法挖掘过程示意图。图3中所示的方法包括:

[0089] 步骤301、获取预设数量的访问请求,得到访问请求序列,对所述访问请求序列进行分段,得到多个子访问请求序列。

[0090] 获取预取数量的访问请求,该访问请求为用户设备访问数据库等的访问请求,得到访问请求序列。例如,预设数量为24,得到的访问请求序列具体为针对请求项“abcdabcefagbchabijcaklc”的访问请求序列。对访问请求进行分段,得到多个子访问请求序列,具体为<abcd>、<abcef>、<agbch>、<abijc>、<aklc>。

[0091] 步骤302、对所述多个子访问请求序列进行简化,得到多个子简化访问请求序列,所述子访问请求序列的数量与所述子简化访问请求序列的数量相等。

[0092] 具体地,根据多个子访问请求序列,得到每个一次请求项的频度,确定大于第一预设频度的多个频繁一次请求项,每个一次请求项对应一个访问请求;根据各频繁一次请求项在多个子访问请求序列中的位置,对多个子访问请求序列进行简化,得到多个子简化访问请求序列。

[0093] 在具体实现过程中,并请结合图4,根据多个子访问请求序列,得到每个一次请求项的频段,其中,每个一次请求项对应一个访问请求。每个一次请求项的频段例如为:<a>,5;<b>,4;<c>5;<e>2……;当第一预设频度为3时,确定大于第一预设频度的多个频繁一次请求项,具体为<a>;<b>,<c>。然后,根据多个频繁一次请求项在多个子访问请求序列中的位置,对多个子访问请求序列进行简化,得到多个子简化访问请求序列,例如,对<abcd>简化得到<abc>,对<abcef>简化得到<abc>、对<agbch>简化得到<abc>、对<abijc>简化得到<abc>、对<aklc>简化得到<ac>。

[0094] 步骤303、根据所述子简化访问请求序列,确定预取规则子集,所述预取规则子集中包括多个预取规则。

[0095] 具体地,根据多个子简化访问序列,得到每个二次请求项的频度,确定大于第二预设频度的多个频繁二次请求项,每个二次请求项对应两个访问请求;

[0096] 根据各频繁二次请求项,确定预取规则子集。

[0097] 在具体实现过程中,根据子简化请求序列<abc>、<abc>、<abc>、<abc>、<ac>,得到每个二次请求项的频度,具体为<ab>,4;<ac>,5;<bc>,4。当第二预设频度为3时,得到多个频繁二次请求项,即为<ab>,4;<ac>,5;<bc>,4。然后根据各频繁二次请求项,确定预取规则子集内包括四个预取规则,分别为预取规则1、预取规则2、预取规则3。预取规则1针对的数据为a、b;预取规则2针对的数据为a、c;预取规则3针对的数据为b、c。

[0098] 本发明实施例在生成多个频繁一次请求项之后,精简原始访问请求序列的样本空间,从而减少后续遍历样本空间的时间,在生成多个频繁二次请求项之后即结束,不再继续迭代,不仅减少内存资源开销和算法的执行时间,而且不会丢失数据块之间的关联信息,同时利用散列和二叉排序树等技术,可以加速频繁一次请求项和频繁二次请求项的生成,使算法能满足实时挖掘的需要。

[0099] 可选地,在根据子简化访问请求序列,确定预取规则子集之后,还包括:

[0100] 将所述预取规则子集添加到预取规则集中,以更新所述预取规则集;

[0101] 判断所述预取规则集中的预取规则子集是否超过预设数量;

[0102] 若是,根据所述预取规则集中的各所述预取规则子集的使用率,删除所述预取规则集中所述使用率满足预设条件的预取规则子集。

[0103] 具体地,由于预取规则集中的预取规则子集的数量是有限的,因此,在判断预取规则集中的预取规则子集是否超过预设数量,若超过预设数量,则根据所述预取规则集中的各所述预取规则子集的使用率,删除所述预取规则集中所述使用率满足预设条件的预取规则子集。

[0104] 进一步地,预取规则子集的使用率与预取规则子集的命中率具有对应关系,预取规则子集的命中是指任一访问请求命中预取规则子集中的预取规则。具体地,针对任意的一个访问请求,当该访问请求针对10时,查找预取规则子集中的各预取规则,当预取规则1对应的数据为10、14、15时,则该访问请求命中该预取规则1,根据该预取规则1,将10、14、15

写入缓存中,此时该预取规则对应的预取规则子集的使用率加1,在后续的申请请求中,很可能直接访问请求14,此时,可直接从缓存中获取14,而不需要从磁盘中获取。

[0105] 图5为本发明预取规则子集删除示意图。如图5所示,第一步S1中,预取规则集为空,当生成预取规则子集1时,将预取规则子集1添加到预取规则集中,在第二步S2中,预取规则子集2生成时,添加到预取规则集中。在第三步S3(中间省略了其它预取规则子集的添加过程)中,新生成的预取规则子集n添加到预取规则集中,n为整数,此时,预取规则集中的预取规则子集达到了预设数量。在第四步S4中,当预取规则子集添加到预取规则集中时,预取规则子集超出了预设数量,此时,删除使用率满足预设条件的预取规则子集。

[0106] 以图5实施例为例,在图5实施例的S4中,预取规则子集2的使用率最低,因此,删除预取规则子集2,此时,将新生成的预取规则子集添加到预取规则集中。本领域技术人员可以理解,如果预设条件为使用率小于预设使用率的条件或使用率在所有使用率中排在预设排名之后的条件,则满足预设条件的预取规则子集可能为多个,删除的预取规则子集也为多个。

[0107] 本发明实施例提供的数据预取方法,通过将预取规则子集添加到预取规则集中,以更新预取规则集;判断预取规则集中的预取规则子集超过预设数量,根据预取规则集中的各预取规则子集的使用率,删除预取规则集中使用率满足预设条件的预取规则子集,避免了根据所有的访问请求生成预取规则,而是生成了多个预取规则子集,并不断地对预取规则子集进行自适应更新,不仅减少了时间复杂度,提高了空间利用率,还提高缓存的命中率。

[0108] 图6为本发明数据预取方法实施例三的流程示意图。本实施例对图2实施例中的预取流程进行详细说明。具体如下:

[0109] 步骤601、判断未命中系统缓存的申请请求是否命中顺序预取缓存中的顺序预取队列,顺序预取缓存包括多个顺序预取队列,若是执行步骤605,若否,执行步骤602;

[0110] 步骤602、从存储设备中获取与未命中系统缓存的申请请求所请求的数据;

[0111] 步骤603、根据LRU选择需要淘汰的顺序预取队列,将需要淘汰的顺序预取队列中命中过申请请求的数据缓存到系统缓存中,将需要淘汰的顺序预取队列中未命中过申请请求的数据丢弃;

[0112] 步骤604、将未命中系统缓存的申请请求所请求的数据与预取窗口内的数据放入需要淘汰的顺序预取队列中,得到新的顺序预取队列。

[0113] 步骤605、判断未命中系统缓存的申请请求是否命中预取窗口内的数据,若是,执行步骤606,若否,执行步骤607;

[0114] 步骤606、从预取窗口内得到未命中系统缓存的申请请求所请求的数据,并根据未命中系统缓存的申请请求所请求的数据从存储设备中预取数据到预取窗口中,使预取窗口的长度增加,将长度增加的预取窗口内的预取数据更新到预取窗口对应的顺序预取队列中;

[0115] 步骤607、从顺序预取缓存中的顺序预取队列中,得到未命中系统缓存的申请请求所请求的数据。

[0116] 在具体实现过程中,顺序预取缓存中包括多个顺序预取队列,首先判断未命中系统缓存的申请请求是否命中顺序预取缓存中的顺序预取队列,当未命中时,执行步骤602至

步骤604。

[0117] 在步骤602中,向存储设备发送未命中系统缓存的访问请求,从存储设备中获取未命中系统缓存的访问请求所请求的数据。

[0118] 在步骤603中,根据最少使用算法(Least Recently Used,简称LRU)选择需要淘汰的顺序预取队列。在具体实现过程中,根据LRU确定最少使用的顺序预取队列,即命中率最低的顺序预取队列,该命中率最低的顺序预取队列即为需要淘汰的顺序预取队列,然后将需要淘汰的顺序预取队列中命中过访问请求的数据缓存到系统缓存中,将需要淘汰的顺序预取队列中未命中过访问请求的数据丢弃。

[0119] 在步骤604中,将未命中系统缓存的访问请求所请求的数据与预取窗口内的数据放入需要淘汰的顺序预取队列中,得到新的顺序预取队列。本实施例实现了对顺序预取缓存的更新。

[0120] 步骤605至步骤607主要说明了如何根据未命中系统缓存的访问请求命中当前预取窗口的结果,得到未命中系统缓存的访问请求所请求的数据。下面进行详细说明。

[0121] 在步骤605中,预取窗口中的数据如果被未命中系统缓存的访问请求命中,则执行步骤606,否则执行步骤607。

[0122] 例如,预取窗口中的数据为4、5、6,且4、5、6均被未命中系统缓存的访问请求命中,则执行步骤606,从预取窗口内得到未命中系统缓存的访问请求所请求的数据4、5、6,然后根据未命中系统缓存的访问请求所请求的数据从存储设备中预取数据到预取窗口中,使预取窗口的长度增加,将长度增加的预取窗口内的预取数据更新到预取窗口对应的顺序预取队列中。

[0123] 在具体实现过程中,由于4、5、6表现出很好的顺序性,则从存储设备中预取7、8、9、10、11、12到预取窗口中,使预取窗口的长度通过翻倍的方式增加。再将预取窗口滑动到新的预取窗口对应的顺序预取队列的最后一个数据之后,将7、8、9、10、11、12添加到顺序预取队列的最后一个数据之后,以更新该顺序预取队列,当更新后的顺序预取队列的长度超过预设长度时,如果该更新的顺序预取队列头部的数据被命中过,则将该数据缓存到系统缓存中,如果没有被命中过,则直接丢弃。

[0124] 在步骤607,如果未命中系统缓存的访问请求命中顺序预取缓存的顺序预取队列,则直接从顺序预取缓存中的顺序预取队列中,得到未命中系统缓存的访问请求所请求的数据。

[0125] 本实施例提供的顺序预取流程,通过预取窗口长度的增加,对顺序预取队列的更新,保证了顺序预取流程能够及时提供数据预取,提高顺序预取缓存的命中率。

[0126] 综上,本发明提供的数据预取方法,首先能够有效提高缓存的命中率,在数据缓存大小占总数据量5%的情况下,对一些负载的命中率可以达到80%,相比只有LRU算法的情况可以提高60%,相比单纯的顺序预取可以提高20%;其次,本发明的预取规则能够充分利用底层块存储设备的带宽;最后,本发明实施例将顺序预取算法与预取规则结合使用,使本实施例提供的数据流程方法更通用,在时间和空间上开销更小。

[0127] 进一步地,对于磁盘而言,由于其自身特性,所以一般应用程序在访问存储设备时会适应磁盘的特性,所以目前不少负载的顺序性还是不错的,而对于固态硬盘(solid-state disk或solid-state drive,简称SSD)而言,它的非顺序访问和顺序访问的差别就没



那么大,所以在以SSD为介质的存储系统中,顺序预取的效果就没那么好,影响存储系统的性能,而本实施例提供的数据预取方法对存储系统的性能影响不大。

[0128] 图7为本发明数据预取装置实施例一的结构示意图。如图7所示,本实施例提供的数据预取装置70包括:

[0129] 获取模块701,用于获取用户设备发送的未命中系统缓存的访问请求,所述系统缓存中存储有根据预取规则得到的预取数据;

[0130] 处理模块702,用于根据所述未命中系统缓存的访问请求在顺序预取缓存中的顺序预取结果,得到所述未命中系统缓存的访问请求所请求的数据,向所述用户设备返回所述所请求的数据;

[0131] 判断模块703,用于判断存在所述未命中系统缓存的访问请求命中的预取规则,则根据所述预取规则对所述未命中系统缓存的访问请求在存储设备中进行预取,将预取得到的预取数据存储在该所述系统缓存中,以便所述系统缓存命中所述用户设备发送的其它访问请求。

[0132] 图8为本发明数据预取装置实施例二的结构示意图。如图8所示,本实施例在图7实施例的基础上实现,具体如下:

[0133] 可选地,所述装置还包括:

[0134] 分段模块704,用于在所述判断模块判断是否存在所述未命中系统缓存的访问请求命中的预取规则之前,获取预设数量的访问请求,得到访问请求序列,对所述访问请求序列进行分段,得到多个子访问请求序列;

[0135] 序列简化模块705,用于对所述多个子访问请求序列进行简化,得到多个子简化访问请求序列,所述子访问请求序列的数量与所述子简化访问请求序列的数量相等;

[0136] 规则生成模块706,用于根据所述子简化访问请求序列,确定预取规则子集,所述预取规则子集中包括多个预取规则。

[0137] 可选地,所述序列简化模块705具体用于:

[0138] 根据所述多个子访问请求序列,得到每个一次请求项的频度,确定大于第一预设频度的多个频繁一次请求项,每个一次请求项对应一个访问请求;

[0139] 根据各所述频繁一次请求项在所述多个子访问请求序列中的位置,对所述多个子访问请求序列进行简化,得到多个子简化访问请求序列。

[0140] 可选地,所述规则生成模块706具体用于:

[0141] 根据所述多个子简化访问序列,得到每个二次请求项的频度,确定大于第二预设频度的多个频繁二次请求项,每个二次请求项对应两个访问请求;

[0142] 根据各所述频繁二次请求项,确定预取规则子集。

[0143] 可选地,所述装置还包括:

[0144] 规则更新模块707,用于在所述规则生成模块根据所述子简化访问请求序列,确定预取规则子集之后,

[0145] 将所述预取规则子集添加到预取规则集中,以更新所述预取规则集;

[0146] 判断所述预取规则集中的预取规则子集是否超过预设数量;

[0147] 若是,根据所述预取规则集中的各所述预取规则子集的使用率,删除所述预取规则集中所述使用率满足预设条件的预取规则子集。

- [0148] 可选地,所述处理模块702具体用于:
- [0149] 判断所述未命中系统缓存的访问请求是否命中顺序预取缓存中的顺序预取队列,所述顺序预取缓存包括多个顺序预取队列;
- [0150] 若是,则根据所述未命中系统缓存的访问请求命中当前预取窗口的结果,得到所述未命中系统缓存的访问请求所请求的数据;
- [0151] 若否,则从所述存储设备中获取与所述未命中系统缓存的访问请求所请求的数据。
- [0152] 可选地,所述处理模块702还具体用于:
- [0153] 判断所述未命中系统缓存的访问请求是否命中所述预取窗口内的数据;
- [0154] 若是,则从所述预取窗口内得到所述未命中系统缓存的访问请求所请求的数据,并根据所述未命中系统缓存的访问请求所请求的数据从所述存储设备中预取数据到预取窗口中,使所述预取窗口的长度增加,将长度增加的所述预取窗口内的预取数据更新到所述预取窗口对应的顺序预取队列中;
- [0155] 若否,则从所述顺序预取缓存中的顺序预取队列中,得到所述未命中系统缓存的访问请求所请求的数据。
- [0156] 可选地,还包括:队列更新模块708,用于在所述处理模块从所述存储设备中获取与所述未命中系统缓存的访问请求所请求的数据之后,
- [0157] 根据最少使用算法LRU选择需要淘汰的顺序预取队列,将所述需要淘汰的顺序预取队列中命中过访问请求的数据缓存到系统缓存中,将所述需要淘汰的顺序预取队列中未命中过访问请求的数据丢弃;
- [0158] 将所述未命中系统缓存的访问请求所请求的数据与预取窗口内的数据放入所述需要淘汰的顺序预取队列中,得到新的顺序预取队列。
- [0159] 本实施例的装置,可以用于执行上述方法实施例的技术方案,其实现原理和技术效果类似,此处不再赘述。
- [0160] 在本申请所提供的几个实施例中,应该理解到,所揭露的系统,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,例如,所述模块的划分,仅仅为一种逻辑功能划分,实际实现时可以有另外的划分方式,例如多个模块或组件可以结合或者可以集成到另一个系统,或一些特征可以忽略,或不执行。另一点,所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口,装置或模块的间接耦合或通信连接,可以是电性,机械或其它的形式。
- [0161] 本领域普通技术人员可以理解:实现上述各方法实施例的全部或部分步骤可以通过程序指令相关的硬件来完成。前述的程序可以存储于一计算机可读取存储介质中。该程序在执行时,执行包括上述各方法实施例的步骤;而前述的存储介质包括:ROM、RAM、磁碟或者光盘等各种可以存储程序代码的介质。
- [0162] 最后应说明的是:以上各实施例仅用以说明本发明的技术方案,而非对其限制;尽管参照前述各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,或者对其中部分或者全部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范围。

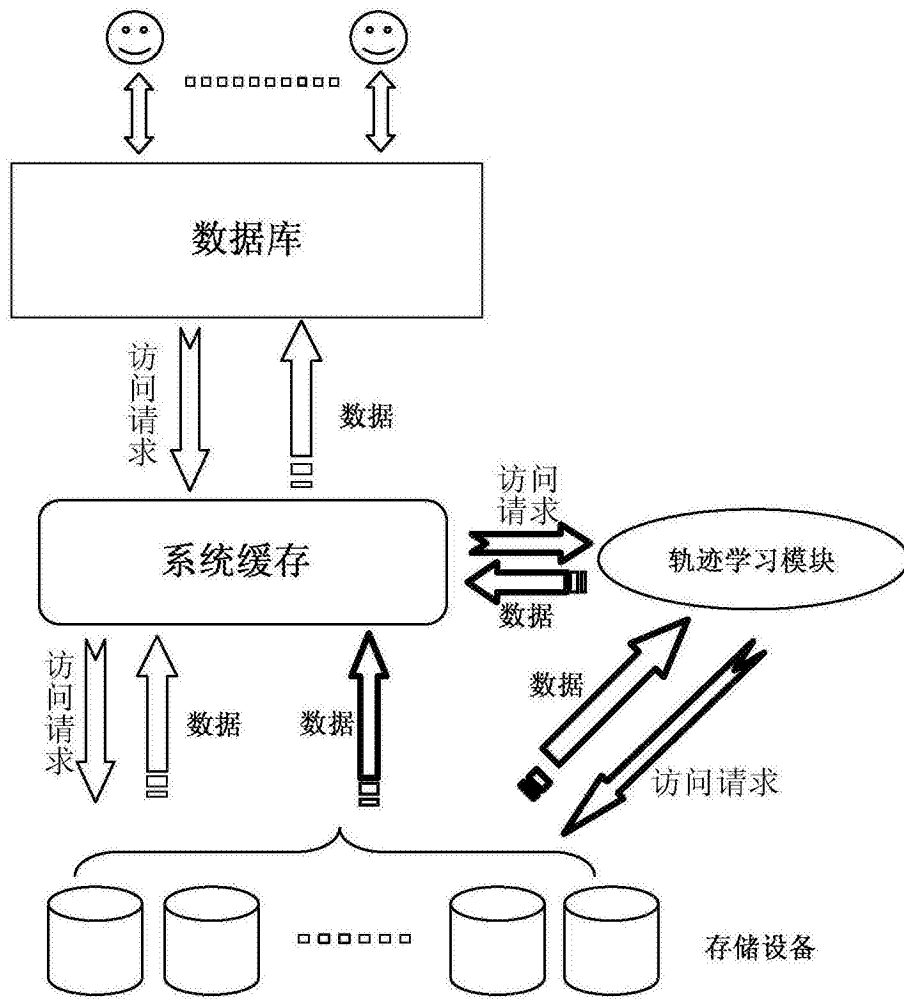


图1

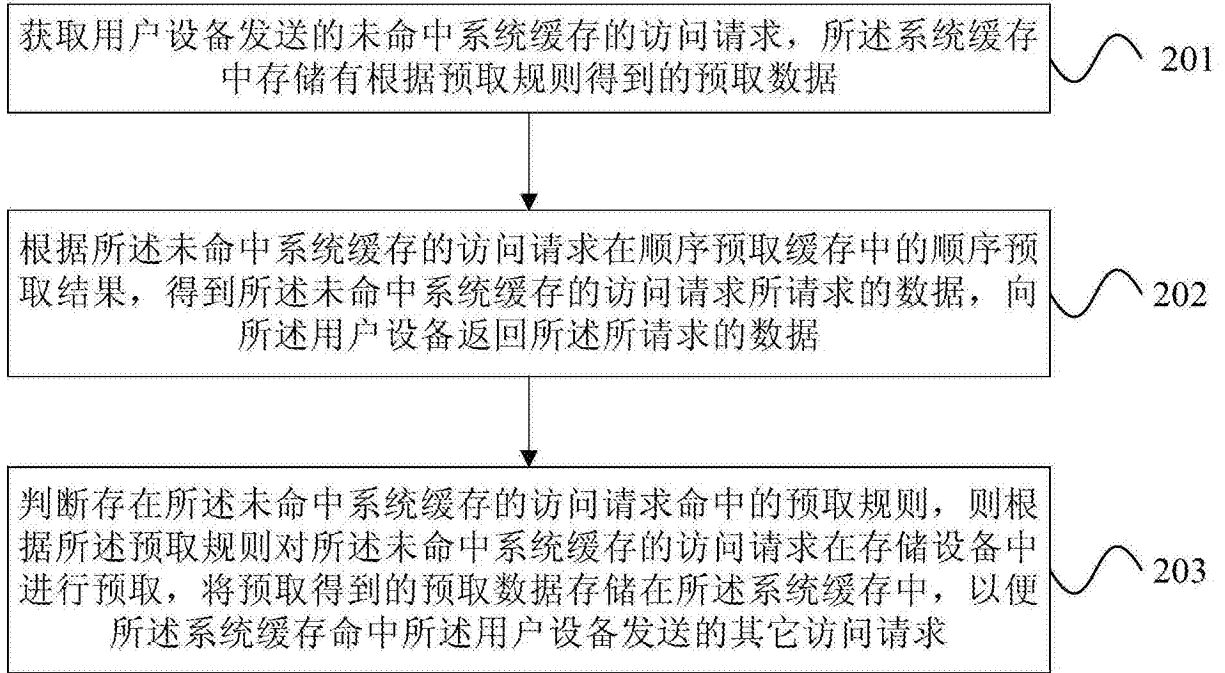


图2

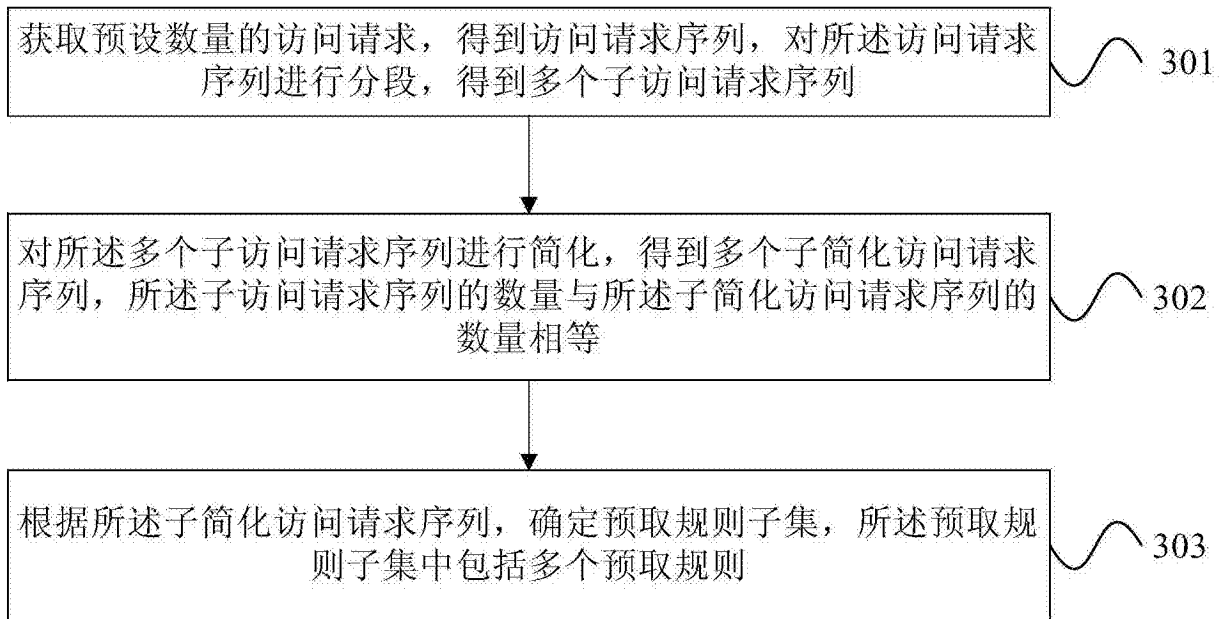


图3

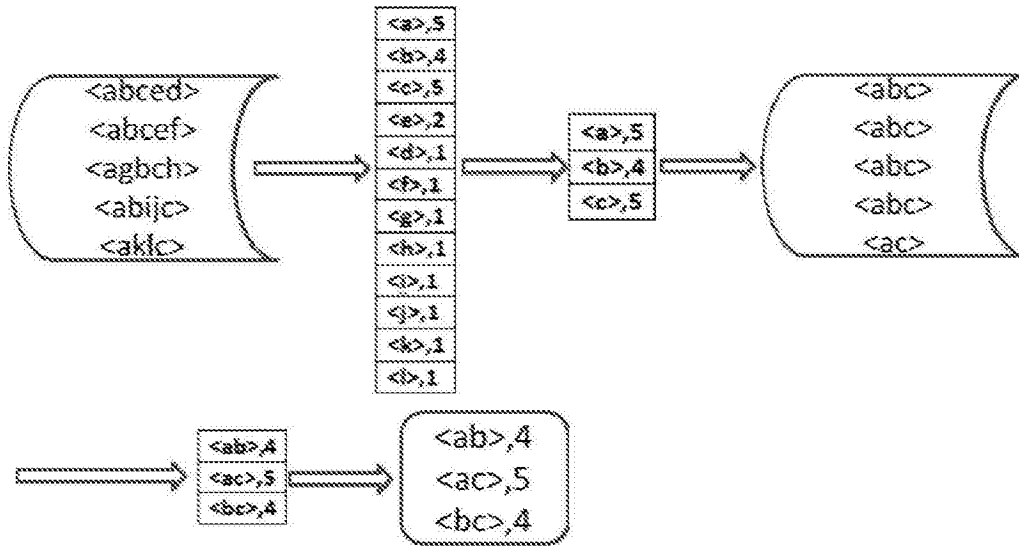


图4

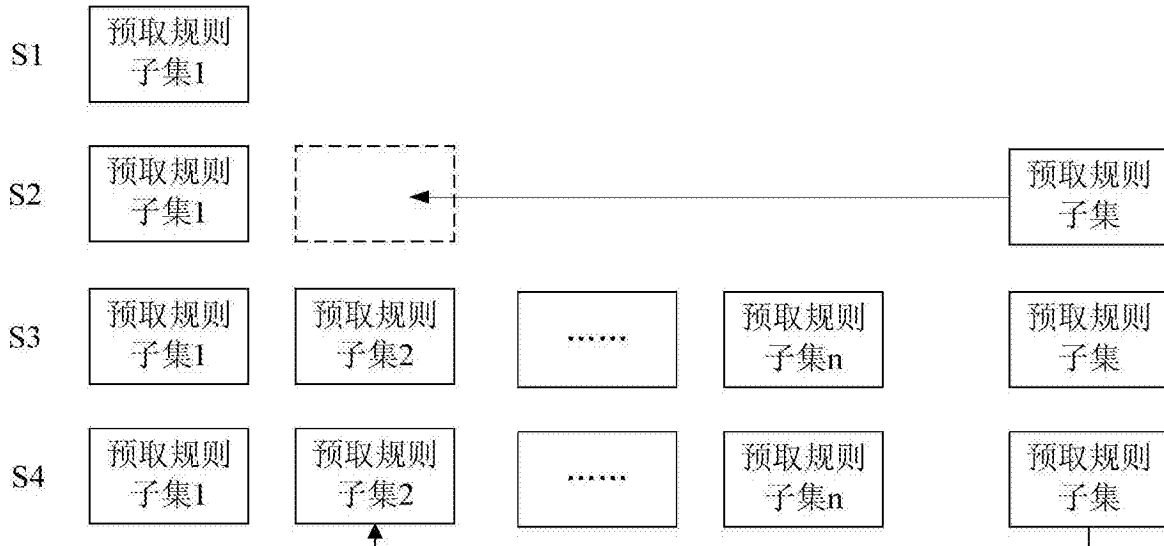


图5

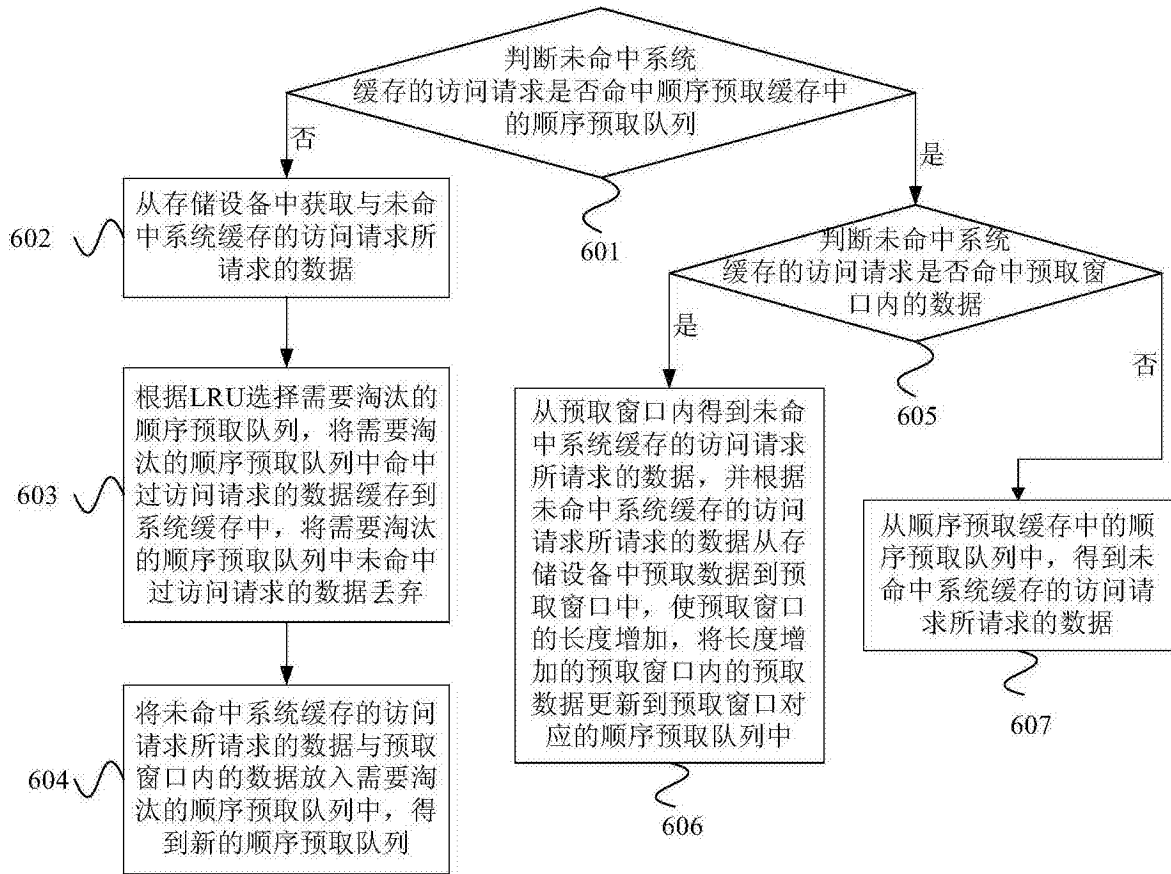


图6

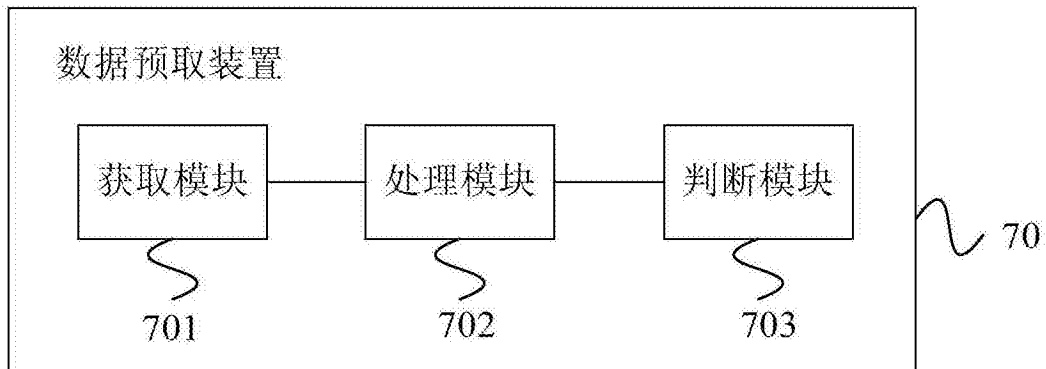


图7

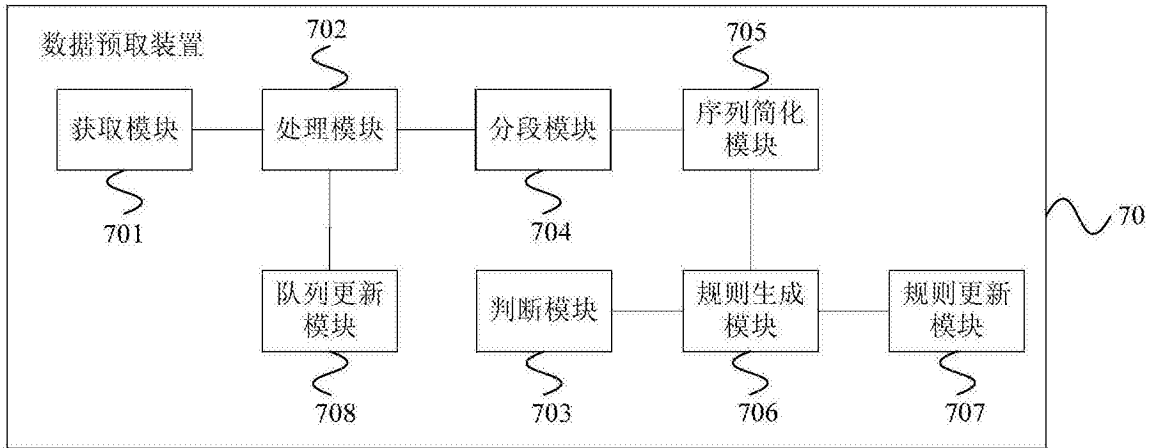


图8