



(12)发明专利

(10)授权公告号 CN 104636209 B

(45)授权公告日 2018.08.24

(21)申请号 201510082522.5

(22)申请日 2015.02.15

(65)同一申请的已公布的文献号
申请公布号 CN 104636209 A

(43)申请公布日 2015.05.20

(73)专利权人 大连云动力科技有限公司
地址 116033 辽宁省大连市甘井子区虹港
路23号

(72)发明人 李漾 张婷

(74)专利代理机构 大连东方专利代理有限责任
公司 21212

代理人 李馨 李洪福

(51)Int.Cl.
G06F 9/50(2006.01)

(56)对比文件

- CN 104035807 A, 2014.09.10,
- CN 103345514 A, 2013.10.09,
- CN 103455579 A, 2013.12.18,
- CN 103678579 A, 2014.03.26,
- CN 103942315 A, 2014.07.23,

审查员 吴黄飞

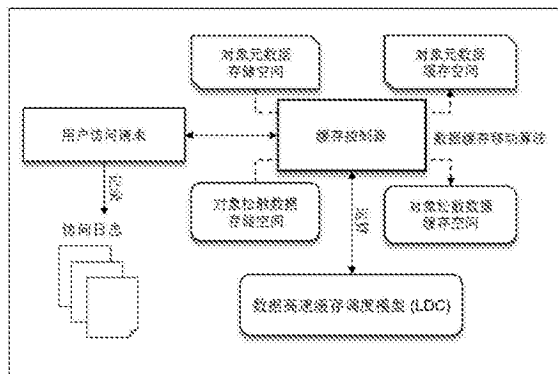
权利要求书3页 说明书8页 附图2页

(54)发明名称

基于大数据和云存储系统定向性能优化的资源调度系统和方法

(57)摘要

本发明公开了一种基于大数据和云存储系统定向性能优化的资源调度系统,具有:对象元数据存储单元MSS、对象元数据缓存单元MCS、对象松散数据存储单元OSS、对象松散数据缓存单元OCS;日志存储单元,存储对象松散数据存储单元OSS中的松散数据的操作日志。系统还具有:缓存控制单元,通过分析所述的操作日志,得出当前时刻或一段时间内访问量超过阈值的数据对象m;根据缓存调度模型计算所述数据对象由对象松散数据存储单元OSS转移至对象松散数据缓存单元OCS后的应用性能优化度;若优化度超过缓存门槛,则将该数据对象m转移至对象松散数据缓存单元OCS;若经缓存后优化度低于所述的缓存门槛;则不转移数据对象m。



1. 一种基于大数据和云存储系统定向性能优化的资源调度系统,包括:

对象元数据存储单元MSS、对象元数据缓存单元MCS、对象松散数据存储单元OSS、对象松散数据缓存单元OCS;日志存储单元,存储对象松散数据存储单元OSS中的松散数据的操作日志;

其特征在于还包括:

缓存控制单元,通过分析所述的操作日志和元数据,得出当前时刻访问量超过阈值的数据对象 m ;根据缓存调度模型计算所述数据对象由对象松散数据存储单元OSS转移至对象松散数据缓存单元OCS后的应用性能优化度;

若优化度超过缓存门槛,则将该数据对象 m 转移至对象松散数据缓存单元OSS;若经缓存后优化度低于所述的缓存门槛;则不转移数据对象 m ;所述的缓存门槛为全部松散数据对象的总访问量与全部松散数据对象数量。

2. 根据权利要求1所述的基于大数据和云存储系统定向性能优化的资源调度系统,其特征还在于所述的缓存调度模型为:

$$\max \sum_{m \in M} S_m x_m$$

S_m 表示所述的数据对象 m 置于缓存空间的得分, x_m 为决策变量, S_m 至少包含用户使用优化度函数 $f(m)$;

对于每个数据对象 $m \in M$, x_m 是一个二元变量,表示是否应将 m 放入缓存空间内;如果 $x_m = 1$,则放入缓存空间; $x_m = 0$,则不放入缓存空间;

$f(m) =$ 对象松散数据存储单元OSS数据读取相应时间 (m) - 对象松散数据缓存单元OCS读取相应时间 (m) ,单位为秒;

其中,对象松散数据存储单元OSS数据读取相应时间 $(m) = PA^1 \cdot z_m + PA^2$;对象松散数据缓存单元OCS的读取相应时间 $(m) = PA^3 \cdot z_m + PA^4$; z_m 为所述数据对象 m 占用空间大小, $m \in M$; M 为松散数据对象集合, PA^1 和 PA^2 分别为针对不同大小区间的对象松散数据的读取的时间系数和修正值; PA^3 和 PA^4 分别为针对不同大小区间对象松散缓存数据的读取的时间系数和修正值。

3. 根据权利要求2所述的基于大数据和云存储系统定向性能优化的资源调度系统,其特征还在于所述 S_m 还包括:数据对象 m 所产生的预期收益函数 $E(m)$ 和移动数据对象 m 的成本函数 $C(m)$;

$$S_m = \alpha f(m) + \beta E(m) - \gamma C(m); E(m) = V_m P_r + V_m z_m P_t;$$

P_t 为数据传输流量每单位价格, P_r 为数据请求次数每单位价格; V_m 为数据对象 m 的所占空间大小, $m \in M$; z_m 为数据对象 m 的所占空间大小, $m \in M$;

$$C(m) = \tau z_m + \psi \frac{R_m}{V_m} + \theta U_c + \iota U_r + \kappa U_n;$$

其中, R_m 为数据对象 m 被修改的次数; U_c 为当前CPU负载比率; U_r 为当前内存使用量; U_n 为当前网络负载比率; $\alpha, \beta, \gamma, \tau, \psi, \theta, \iota, \kappa$ 是模型中的常量,反映系统中不同因素对最终得分的影响程度。

4. 根据权利要求1所述的基于大数据和云存储系统定向性能优化的资源调度系统,其

特征还在于：

当需要计算的对象的总数量超过设定阈值时，所述的缓存控制单元在所述的缓存调度模型中引入松弛变量，求得原始NP完全问题的最优上限或下限；引入松弛变量后的缓存调度模型为：

$$\text{最大化} \left[\sum_{m \in M} s_m x_m + \lambda \left(d - \sum_{m \in M} z_m x_m \right) \right]$$

其中， $x_m \in (0, 1)$ ， $m \in M$ ，模型的约束变量 d 为所述对象松散数据缓存空间总容量， λ 为模型中的常量且 $\lambda \geq 0$ 。

5. 根据权利要求4所述的基于大数据和云存储系统定向性能优化的资源调度系统，其特征还在于：在求解过程中，所述的缓存控制单元引入以下拉格朗日对偶函数来求解规划问题：

$$\text{最小化}_{\lambda \geq 0} \left\{ \text{最大化} \left[\sum_{m \in M} s_m x_m + \lambda \left(d - \sum_{m \in M} z_m x_m \right) \right] \right\}$$

其中， $x_m \in (0, 1)$ ， $m \in M$ ，模型的约束变量 d 为所述对象松散数据缓存空间总容量， λ 为模型中的常量。

6. 一种基于大数据和云存储系统定向性能优化的资源调度方法，应用于对象元存储单元MCS、对象元数据缓存单元OCS、对象松散数据存储单元MSS、对象松散数据缓存单元OSS的大数据和存储系统中，具有如下步骤：

一分析对象松散数据存储单元MSS中的松散数据的操作日志和元数据，得到当前时刻之前一段时间内数据对象 m 的访问次数；

一根据缓存调度模型计算所述数据对象 m 转移至所述对象松散数据存储单元OSS后的应用系能优化度，该应用系能优化度包含了减少数据请求的响应时间及加快数据传输速率；

一若优化度超过缓存门槛，则将该数据对象 m 转移至对象松散数据缓存单元OSS；若经缓存后优化度低于所述的缓存门槛；则不转移数据对象 m 。

7. 根据权利要求6所述的基于大数据和云存储系统定向性能优化的资源调度方法，其特征还在于所述的缓存调度模型为：

$$\max \sum_{m \in M} S_m x_m$$

S_m 表示所述的数据对象 m 置于缓存空间的得分， x_m 为决策变量， S_m 至少包含用户使用优化度函数 $f(m)$ ；

对于每个数据对象 $m \in M$ ， x_m 是一个二元变量，表示是否应将 m 放入缓存空间内；如果 $x_m = 1$ ，则放入缓存空间； $x_m = 0$ ，则不放入缓存空间；

$f(m) =$ 对象松散数据存储单元MSS数据读取相应时间 (m) - 对象松散数据缓存单元OSS读取相应时间 (m) ；

其中，对象松散数据存储单元MSS数据读取相应时间 $(m) = PA^1 \cdot z_m + PA^2$ ；对象松散数据

缓存单元OSS读取相应时间 $(m) = PA^3 \cdot z_m + PA^4$; z_m 为所述数据对象 m 占用空间大小, $m \in M$; M 为数据对象集合; PA^1 和 PA^2 分别为针对不同大小区间的对象松散数据的读取的时间系数和修正值; PA^3 和 PA^4 分别为针对不同大小区间对象松散缓存数据的读取的时间系数和修正值。

8. 根据权利要求7所述的基于大数据和云存储系统定向性能优化的资源调度方法, 其特征还在于:

所述 S_m 还包括: 数据对象 m 所产生的预期收益函数 $E(m)$ 和移动数据对象 m 的成本函数 $C(m)$;

$$S_m = \alpha f(m) + \beta E(m) - \gamma C(m); E(m) = V_m P_r + V_m z_m P_t;$$

P_t 为数据传输流量每单位价格, P_r 为数据请求次数每单位价格; V_m 为数据对象 m 的所占空间大小, $m \in M$; z_m 为数据对象 m 的所占空间大小, $m \in M$;

$$C(m) = \tau z_m + \psi \frac{R_m}{V_m} + \theta U_c + \iota U_r + \kappa U_n;$$

其中, R_m 为数据对象 m 被修改的次数; U_c 为当前CPU负载比率; U_r 为当前内存使用量; U_n 为当前网络负载比率; $\alpha, \beta, \gamma, \tau, \psi, \theta, \iota, \kappa$ 是模型中的常量, 反映系统中不同因素对最终得分的影响程度。

9. 根据权利要求6所述的基于大数据和云存储系统定向性能优化的资源调度方法, 其特征还在于:

当需要计算的对象的总数量超过设定阈值时, 所述的缓存控制单元在所述的缓存调度模型中引入松弛变量, 求得原始NP完全问题的最优上限或下限: 引入松弛变量后的缓存调度模型为:

$$\text{最大化} \left[\sum_{m \in M} s_m x_m + \lambda \left(d - \sum_{m \in M} z_m x_m \right) \right]$$

其中, $x_m \in (0, 1)$, $m \in M$, 模型的约束变量 d 为所述对象松散数据缓存空间总容量, λ 为模型中的常量且 $\lambda \geq 0$ 。

10. 根据权利要求9所述的基于大数据和云存储系统定向性能优化的资源调度方法, 其特征还在于: 在求解过程中, 所述的缓存控制单元引入以下拉格朗日对偶函数来求解规划问题:

$$\text{最小化}_{\lambda \geq 0} \left\{ \text{最大化} \left[\sum_{m \in M} s_m x_m + \lambda \left(d - \sum_{m \in M} z_m x_m \right) \right] \right\}$$

其中, $x_m \in (0, 1)$, $m \in M$, 模型的约束变量 d 为所述对象松散数据缓存空间总容量, λ 为模型中的常量。

基于大数据和云存储系统定向性能优化的资源调度系统和 方法

技术领域

[0001] 本发明涉及一种资源优化调度系统和方法,尤其涉及基于大数据和云存储系统定向性能优化的资源调度系统和方法。涉及专利分类号G06计算;推算;计数G06F电数字数据处理G06F9/00程序控制装置,例如,控制器G06F9/06应用存入的程序的,即应用处理设备的内部存储来接收程序并保持程序的G06F9/46多道程序装置G06F9/50资源分配,例如,中央处理单元。

背景技术

[0002] 大数据时代,移动互联、社交网络、物联网、云服务等应用的迅速普及,全世界范围不同的领域内如各大商业网站、政府、医疗、金融、媒体等,都产生了的大量数据,这些数据的容量不断呈指数级增长。这对数据中心提出了革命性的要求,存储基础架构成为核心之一。

[0003] 通过对之前已公开的专利所述“一种大数据存储系统”(申请号:201410053769,公开号:103812939A)的部署和使用发现,在存储的大量数据中,有一些数据在一段时间被很集中的访问,其他数据很少被访问。当大量这种类型的“高峰时间”的现象同时发生时,许多存储设备,网络,磁盘I/O和CPU都存在访问瓶颈等问题。类似问题也出现在一些大型网站及互联网企业中,如淘宝(taobao.com)和Facebook(脸书)为了解决“高峰时间”问题,均采用缓存系统,比如Memcache来提升网页访问性能。据报道,Facebook使用超过800个Memcache服务器提供的内存超过28TB服务每秒超过200,000UDP请求。

[0004] 我们之后部署了简单的缓存服务器来试图解决这样的“高峰时间”的问题。但发现,尽管数据存取性能得到了提高,整个系统的I/O(包含磁盘I/O,网络I/O)几乎加倍。这在很大程度上是由于我们的存储服务器和新引入的缓存服务器之间的频繁的数据交换。

[0005] 因为缓存单元或设备较一般存储设备而言,同等容量下,成本更大,并且,大数据和云存储系统与传统的单用户文件系统相比,采用不同的计价模型,并有着不同的用户使用习惯和方式。

[0006] 因此,我们需要一个更有效的机制,把最“理想”的对象在最适当的时间放入缓存单元内,并在合适的时间删除它们。这样的机制可以1有效地使用可用的高速缓存单元,从而2提高了数据访问的总体性能,同时也3减少磁盘IO和CPU瓶颈。此外,4通过为用户提供额外的和优越的性能作为一个分离的服务,并相应地对其收费,服务供应商可以产生更大的收入。

[0007] 尽管一些大型云存储平台提供商如亚马逊(Amazon S3)、谷歌(Google),阿里云(Aliyun)已经在这方面取得了巨大成功,但目前大数据及云存储领域内仍然缺乏合理可用的资源调度及优化方法来将可用的缓存单元或设备用于提升系统性能及提高企业收入。

发明内容

[0008] 本发明针对以上问题的提出,而研制的一种基于大数据和云存储系统定向性能优化的资源调度系统,具有:

[0009] 对象元数据存储单元MSS、对象元数据缓存单元MCS、对象松散数据存储单元OSS、对象松散数据缓存单元OCS;日志存储单元,存储对象松散数据存储单元OSS中的松散数据的操作日志。

[0010] 存储单元指普通类型的存储空间或设备,缓存单元相对于本发明/一般所指的存储单元,读取和存储速度更高,带宽更大,存储于缓存空间中的数据访问速度大大高于存储于普通存储空间。

[0011] 系统还具有:缓存控制单元,通过分析所述的操作日志,得出当前时刻或一段时间内访问量超过阈值的数据对象 m ;根据缓存调度模型计算所述数据对象由对象松散数据存储单元OSS转移至对象松散数据缓存单元OCS后的应用性能优化度;

[0012] 若优化度超过缓存门槛,则将该数据对象 m 转移至对象松散数据缓存单元OCS;若经缓存后优化度低于所述的缓存门槛;则不转移数据对象 m 。在本发明中,该缓存门槛设置为全部松散数据对象的总访问量与全部松散数据对象数量。

[0013] 作为优选的实施方式,所述的缓存调度模型为:

$$[0014] \quad \max \sum_{m \in M} S_m x_m$$

[0015] S_m 表示所述的数据对象 m 置于缓存空间的得分, x_m 为决策变量, S_m 至少包含用户使用优化度函数 $f(m)$;对于每个数据对象 $m \in M$, x_m 是一个二元变量,表示是否应将 m 放入缓存空间内;如果 $x_m = 1$,则放入缓存空间; $x_m = 0$,则不放入缓存空间; $f(m) =$ 对象松散数据存储单元OSS数据读取相应时间(m)-对象松散数据缓存单元OCS读取相应时间(m);其中,对象松散数据存储单元MSS数据读取相应时间(m)= $PA1 \cdot z_m + PA2$;对象松散数据缓存单元OCS读取相应时间(m)= $PA3 \cdot z_m + PA4$; z_m 为所述数据对象 m 占用空间大小, $m \in M$; M 为松散数据对象集合。

[0016] 通过采用所述的缓存调度模型,对松散数据访问的历史情况进行统计,有效的对数据对象 m 未来的访问情况进行预测,相对于现有的访问机制(注入化简映射等手段可实现),本发明所采用的缓存调度机制调度的数据更为精确。

[0017] 更进一步的,为了能够协调用户满意度和大数据运营商收入,作为优选的实施方式,所述 S_m 还包括:数据对象 m 所产生的预期收益函数 $E(m)$ 和移动数据对象 m 的成本函数 $C(m)$;

$$[0018] \quad S_m = \alpha f(m) + \beta E(m) - \gamma C(m); E(m) = V_m P_r + V_m z_m P_t;$$

[0019] P_t 数据传输流量每单位价格 P_r 数据请求次数每单位价格; V_m 数据对象 m 的所占空间大小, $m \in M$; z_m 数据对象 m 的所占空间大小, $m \in M$;

$$[0020] \quad C(m) = \tau z_m + \psi \frac{R_m}{V_m} + \theta U_c + \iota U_r + \kappa U_n;$$

[0021] 其中, R_m 为数据对象 m 被修改的次数; U_c 当前CPU负载比率; U_r 当前内存使用量; U_n 当前网络负载比率; $\alpha, \beta, \gamma, \tau, \psi, \theta, \iota, \kappa$ 是模型中的常量,反映系统中不同因素对最终得分的影响程度。

[0022] $\alpha, \beta, \gamma, \tau, \psi, \theta, \iota, \kappa$ 是模型中的常量,它们反映出系统中不同因素对最终得分的影响程度。这样,云存储系统的供应商便可根据各自的需求定义适当的常量值,从而建立不同的缓存机制模型。

[0023] 比如:以盈利为首要目的的云存储供应商为了实现利润最大化,可将 α 设为0, β 设为1。相反的,对于不以盈利为目的的个体云存储服务供应者,唯一重要的因素是使用的性能如何,因此可将 α 设为1, β 设为0。其他常量 $\gamma, \tau, \psi, \theta, \iota, \kappa$,用于控制物理资源(如:CPU,内存,网络)的使用强度。例如: ψ 越高意味着缓存高频率更新数据对象的几率越低; τ 越大意味着缓存较大文件的几率越低,也可将 $\gamma, \tau, \psi, \theta, \iota, \kappa$ 均设为0则忽略计算成本函数。在本专利所述资源调度方法中,将 $\gamma, \tau, \psi, \theta, \iota, \kappa$ 均设为1。

[0024] 例如:有些公司的硬件环境所使用的服务器网络按流量计费,那么对于他们来说网络 U_n 权重很大,所以可将 κ 值设大。有些公司硬件环境内存属于稀缺资源,所以看重内存的负载,所以可将 U_r 的权重 ι 设置大。有些公司硬件环境CPU属于稀缺资源,所以看重内存的负载,所以可将 U_c 的权重 θ 设置大。有些公司硬件环境磁盘的IO属于稀缺资源, $\frac{R_m}{V_m}$ 大的话意味着会有频繁读写操作,所以可将 ψ 设置大。

[0025] 更进一步的,考虑到实际使用缓存调度模型运算时,可能出现计算对象的总数巨大(总数量可以通过分析访问日志得到。总数量大的,计算时间会慢。评判标准依据不同环境决定,主要取决于可以承受的计算时间。例如如果当前环境(公司/企业)使用模型发现计算时间超过可承受的时间,例如10分钟,便可引入松弛变量)超出运算能力,造成获得0-1整数规划的最优解耗费大量时间的情况。故为了能够在系统的多项式时间内求解模型,作为优选的实施方式,所述的缓存控制单元在所述的缓存调度模型中引入松弛变量,求得原始NP完全问题的最优上限或下限:

[0026] 引入松弛变量后的缓存调度模型为:

$$[0027] \quad \text{最大化} \left[\sum_{m \in M} s_m x_m + \lambda \left(d - \sum_{m \in M} z_m x_m \right) \right] (\lambda \geq 0)$$

[0028] 其中, $x_m \in (0, 1), m \in M$ 。

[0029] 更进一步的,上述公式的最小上限应为最接近初始规划问题最优解的值。这便需要引入以下拉格朗日对偶函数来求解规划问题:

$$[0030] \quad \text{最小化}_{\lambda \geq 0} \left\{ \text{最大化} \left[\sum_{m \in M} s_m x_m + \lambda \left(d - \sum_{m \in M} z_m x_m \right) \right] \right\}$$

[0031] 其中, $x_m \in (0, 1), m \in M$ 。

[0032] 一种基于大数据和云存储系统定向性能优化的资源调度方法,应用于对象元存储单元MSS、对象元数据缓存单元MCS、对象松散数据存储单元OSS、对象松散数据缓存单元OCS的大数据和存储系统中,具有如下步骤:

[0033] 一分析对象松散数据存储单元OSS中的松散数据的操作日志,得到当前时刻之前一段时间内数据对象 m 的访问次数;

[0034] 一根据缓存调度模型计算所述数据对象 m 转移至所述对象松散数据存储单元OCS后的应用系能优化度；

[0035] 一若优化度超过缓存门槛,则将该数据对象 m 转移至对象松散数据缓存单元OCS;若经缓存后优化度低于所述的缓存门槛;则不转移数据对象 m 。

[0036] 所述的缓存调度模型为:

$$[0037] \quad \max \sum_{m \in M} S_m x_m$$

[0038] S_m 表示所述的数据对象 m 置于缓存空间的得分, x_m 为决策变量, S_m 至少包含用户使用优化度函数 $f(m)$;

[0039] 对于每个数据对象 $m \in M$, x_m 是一个二元变量,表示是否应将 m 放入缓存空间内;如果 $x_m=1$,则放入缓存空间; $x_m=0$,则不放入缓存空间;

[0040] $f(m)$ = 对象松散数据存储单元MSS数据读取相应时间(m) - 对象松散数据缓存单元OCS读取相应时间(m);

[0041] 其中,对象松散数据存储单元MSS数据读取相应时间(m) = $PA1 * z_m + PA2$;对象松散数据缓存单元OCS读取相应时间(m) = $PA3 * z_m + PA4$; z_m 为所述数据对象 m 占用空间大小, $m \in M$; M 为数据对象集合。

[0042] 所述 S_m 还包括:数据对象 m 所产生的预期收益函数 $E(m)$ 和移动数据对象 m 的成本函数 $C(m)$;

[0043] $S_m = \alpha f(m) + \beta E(m) - \gamma C(m)$; $E(m) = V_m P_r + V_m z_m P_t$;

[0044] P_t 数据传输流量每单位价格 P_r 数据请求次数每单位价格; V_m 数据对象 m 的所占空间大小, $m \in M$; z_m 数据对象 m 的所占空间大小, $m \in M$;

$$[0045] \quad C(m) = \tau z_m + \psi \frac{R_m}{V_m} + \theta U_c + \iota U_r + \kappa U_n;$$

[0046] 其中, R_m 为数据对象 m 被修改的次数; U_c 当前CPU负载比率; U_r 当前内存使用量; U_n 当前网络负载比率; $\alpha, \beta, \gamma, \tau, \psi, \theta, \iota, \kappa$ 是模型中的常量,反映系统中不同因素对最终得分的影响程度。

[0047] 当需要计算的对象的总数量超过设定阈值时,所述的缓存控制单元在所述的缓存调度模型中引入松弛变量,求得原始NP完全问题的最优上限或下限:引入松弛变量后的缓存调度模型为:

$$[0048] \quad \text{最大化} \left[\sum_{m \in M} s_m x_m + \lambda \left(d - \sum_{m \in M} z_m x_m \right) \right] (\lambda \geq 0)$$

[0049] 其中, $x_m \in (0, 1)$, $m \in M$ 。

[0050] 在求解过程中,所述的缓存控制单元引入以下拉格朗日对偶函数来求解规划问题;

$$[0051] \quad \text{最小化}_{\lambda \geq 0} \left\{ \text{最大化} \left[\sum_{m \in M} s_m x_m + \lambda \left(d - \sum_{m \in M} z_m x_m \right) \right] \right\}$$

[0052] 其中, $x_m \in (0, 1)$, $m \in M$ 。

附图说明

[0053] 为了更清楚的说明本发明的实施例或现有技术的技术方案,下面将对实施例或现有技术描述中所需要使用的附图做一简单地介绍,显而易见地,下面描述中的附图仅仅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的前提下,还可以根据这些附图获得其他的附图。

[0054] 图1为对象数据元数据存储空间和空间示意图;

[0055] 图2为本发明的系统架构示意图

[0056] 图3为本发明的时序流程图

[0057] 图4为本发明实施例中映射化简示意图

具体实施方式

[0058] 为使本发明的实施例的目的、技术方案和优点更加清楚,下面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的描述:

[0059] 如图1-4所示:一种基于大数据和云存储系统定向性能优化的资源调度方法或者系统,主要应用于包含如下单元的大数据和/或云存储系统中。

[0060] 对象元数据存储单元MSS、

[0061] 对象元数据缓存单元MCS、

[0062] 对象松散数据存储单元OSS、

[0063] 对象松散数据缓存单元OCS;

[0064] 日志存储单元,存储对象松散数据存储单元OSS中的松散数据的操作日志。

[0065] 存储单元指普通类型的存储空间或设备,缓存单元相对于本发明/一般所指的存储单元,读取和存储速度更高,带宽更大,存储于缓存空间中的数据访问速度大大高于存储于普通存储空间。

[0066] 还具有:缓存控制单元/步骤,通过分析所述的操作日志,得出当前时刻或一段时间内访问量超过阈值的数据对象 m ;根据缓存调度模型计算所述数据对象由对象松散数据存储单元OSS转移至对象松散数据缓存单元OCS后的应用性能优化度。

[0067] 分析操作日志,主要通过映射和化简的方式进行,在本实施例中,可按如下方法实现(为方便描述,采用伪代码形式)

[0068] 日志处理流程

输入数据 *日志记录文件*

[0069] 据:

[0070] 映射操作Map(String key,String value)

[0071] 1.//value:代表每一行日志记录,例如“数据下载2014-11-02 23:27:01容器1/对

象1”

[0072] 2.set n_obj=0,n_meta=0,n_oc=0,n_mc=0//初始化参数
 [0073] 3.emitKey=parseEmitKey (value) //获取容器及对象命名空间从value参数中
 [0074] 4.if (is GetObject request)
 [0075] 5.set n_obj=1,n_meta=1//如果是数据下载操作,更新参数n_obj,n_meta
 [0076] 6.else if (is PutObject request)
 [0077] 7.set n_oc=1,n_mc=1//如果是数据上传操作,更新参数n_oc,n_mc
 [0078] 8.……//更新其他数据操作类型及参数
 [0079] 9.end if
 [0080] 10.将所有参数值写入emitValue
 [0081] 11.emit (emitKey,emitValue) //映射操作完成,将赋值参数返回
 [0082] 化简操作Reduce (String key,iterator values)
 [0083] 1.//输入key为Map操作所传递的emitKey
 [0084] 2.//输入values为Map操作所传递的所有emitKey对应的参数值 (emitValue)
 [0085] 1.set n_obj=0,n_meta=0,n_oc=0,n_mc=0//初始化参数
 [0086] 2.foreach (value in values) //遍历每一个值
 [0087] 3.//parse and update request counters
 [0088] 4.n_obj=n_obj+parse_n_obj (value)
 [0089] 5.n_meta=n_meta+parse_n_meta (value) //遍历每种数据操作类型,并更新参数
 值 (更新所有的变量,例如对象松散,元数据下载量,修改量)
 [0090] 6.……
 [0091] 7.end for
 [0092] 8.将所有参数值写入emitValue
 [0093] 9.emit (key,emitValue) ///化简操作完成,将赋值参数返回
 [0094] 其中,N_obj为对象分散数据下载次数;N_meta对象元数据下载次数N_oc对象分散
 数据修改次数;N_mc对象元数据修改次数。
 [0095] 缓存调度模型为:

$$[0096] \quad \max \sum_{m \in M} S_m x_m$$

[0097] S_m 表示所述的数据对象 m 置于缓存空间的得分, x_m 为决策变量, S_m 至少包含用户使用优化度函数 $f(m)$;对于每个数据对象 $m \in M$, x_m 是一个二元变量,表示是否应将 m 放入缓存空间内;如果 $x_m=1$,则放入缓存空间; $x_m=0$,则不放入缓存空间; $f(m) =$ 对象松散数据存储单元OSS数据读取相应时间 (m) -对象松散数据缓存单元OCS读取相应时间 (m) ;其中,对象松散数据存储单元MSS数据读取相应时间 $(m) = PA1 \cdot z_m + PA2$;对象松散数据缓存单元OCS读取相应时间 $(m) = PA3 \cdot z_m + PA4$; z_m 为所述数据对象 m 占用空间大小, $m \in M$; M 为松散数据对象集合。

[0098] 在本实施例中,所述 S_m 还包括:数据对象 m 所产生的预期收益函数 $E(m)$ 和移动数据对象 m 的成本函数 $C(m)$;

[0099] $S_m = \alpha f(m) + \beta E(m) - \gamma C(m)$; $E(m) = V_m P_r + V_m z_m P_t$;

[0100] P_t 数据传输流量每单位价格; P_r 数据请求次数每单位价格; V_m 数据对象m的所占空间大小, $m \in M$; z_m 数据对象m的所占空间大小, $m \in M$;

[0101] $C(m) = \tau z_m + \psi \frac{R_m}{V_m} + \theta U_c + \iota U_r + \kappa U_n$;

[0102] 其中, R_m 为数据对象m被修改的次数; U_c 当前CPU负载比率; U_r 当前内存使用量; U_n 当前网络负载比率; $\alpha, \beta, \gamma, \tau, \psi, \theta, \iota, \kappa$ 是模型中的常量,反映系统中不同因素对最终得分的影响程度。

[0103] $\alpha, \beta, \gamma, \tau, \psi, \theta, \iota, \kappa$ 是模型中的常量,在本实施例中 $\gamma, \tau, \psi, \theta, \iota, \kappa$ 均为1。

[0104] 使用模型的演算过程如下(采用伪代码形式):

[0105] LDC演算流程:

[0106] 输入:

[0107] C全局缓存哈希表<key,

: ObjectCache>

t_s 日志开始时间

:

[0108] t_e 日志结束时间

:

M 缓存空间最大允许存放对象数

AX: 量

[0109] 1.开始

[0110] 2.通过日志处理流程处理 t_s 和 t_e 时间段内的日志记录文件

[0111] 3.创建有新的哈希表C'

[0112] 4.foreach(log in logs)//遍历每一条访问记录

[0113] 5.设置对应key从处理后的访问日志记录中,并更新C'

[0114] 6.end for

[0115] 7.合并C'和C形成新的C,并更新所有对应的对象访问记录

[0116] 8.从C中移除低于缓存阈值(threshold)的对象

[0117] 9.if(size(C)>MAX),then//如果C中对象总数量大于MAX

[0118] 10.将C转化成排序数组A,最多不超过MAX

[0119] 11.else

[0120] 12.将C转化成排序数组A

[0121] 13.C=使用A作为输入并演算LDC模型解决对象松散数据调度问题

[0122] 14.C=使用A作为输入并演算LDC模型解决对象数据元数据调度问题

[0123] 15.结束

[0124] 数据移动流程

[0125] 输入:

[0126] C: 全局缓存哈希表<key, ObjectCache>

- [0127] 1.开始
- [0128] 2.初始化参数latestCacheFiles[]
- [0129] 3.foreach(key in C)//遍历C中每个key
- [0130] 4.c=C(key)
- [0131] 5.meta=fetchObjectMetadata(key)///获取对象元数据
- [0132] 6.c.setMetadata(meta)//设置缓存元数据
- [0133] 7.if(c.toObjCache())//判断是否要缓存对象松散数据
- [0134] 8.latest_etag=meta.getEtag()//获取etag,用于表示数据校验值(checksum)
- [0135] 9.cached_etag=c.getEtag()
- [0136] 10.if(cached_etag==latest_etag)//判断etag是否一致
- [0137] 11.latestCacheFiles.add(c.getCacheFile())//更新latestCacheFiles变量
- [0138] 12.else
- [0139] 13.file_id=randomUUID()//生成新的随机字符串作为文件标识
- [0140] 14.etag=copyfile(meta.getFile(),new File(file_id))//移动对象松散数据到缓存空间
- [0141] 15.c.setCachedFile(file_id)
- [0142] 16.c.setEtag(etag)
- [0143] 17.latestCacheFiles.add(c.getCachedFile())//更新latestCacheFiles变量
- [0144] 18.updateCachedHashTable(C,c)//更新全局缓存哈希表C
- [0145] 19.end if
- [0146] 20.end if
- [0147] 21.end for
- [0148] 22.allCacheFiles[]=listAllFiles()//获取目前缓存空间内现有对象列表
- [0149] 23.foreach(file in allCacheFiles)
- [0150] 24.if(!latestCacheFiles.contains(file))//若latestCacheFiles不含有对应file
- [0151] 25.删除file
- [0152] 26.end if
- [0153] 27.end for
- [0154] 28.结束

[0155] 以上所述,仅为本发明较佳的具体实施方式,但本发明的保护范围并不局限于此,任何熟悉本技术领域的技术人员在本发明揭露的技术范围内,根据本发明的技术方案及其发明构思加以等同替换或改变,都应涵盖在本发明的保护范围之内。

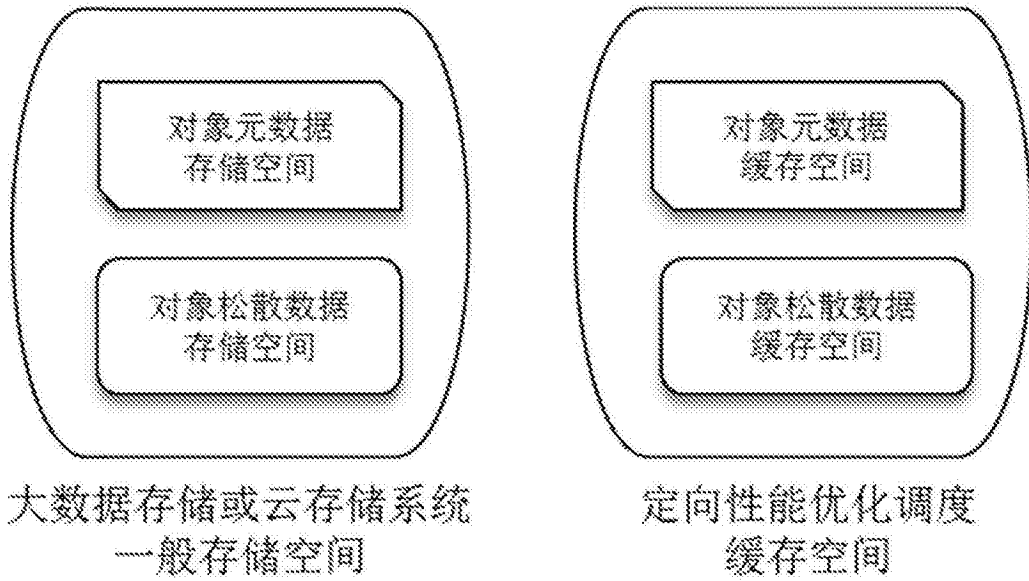


图1

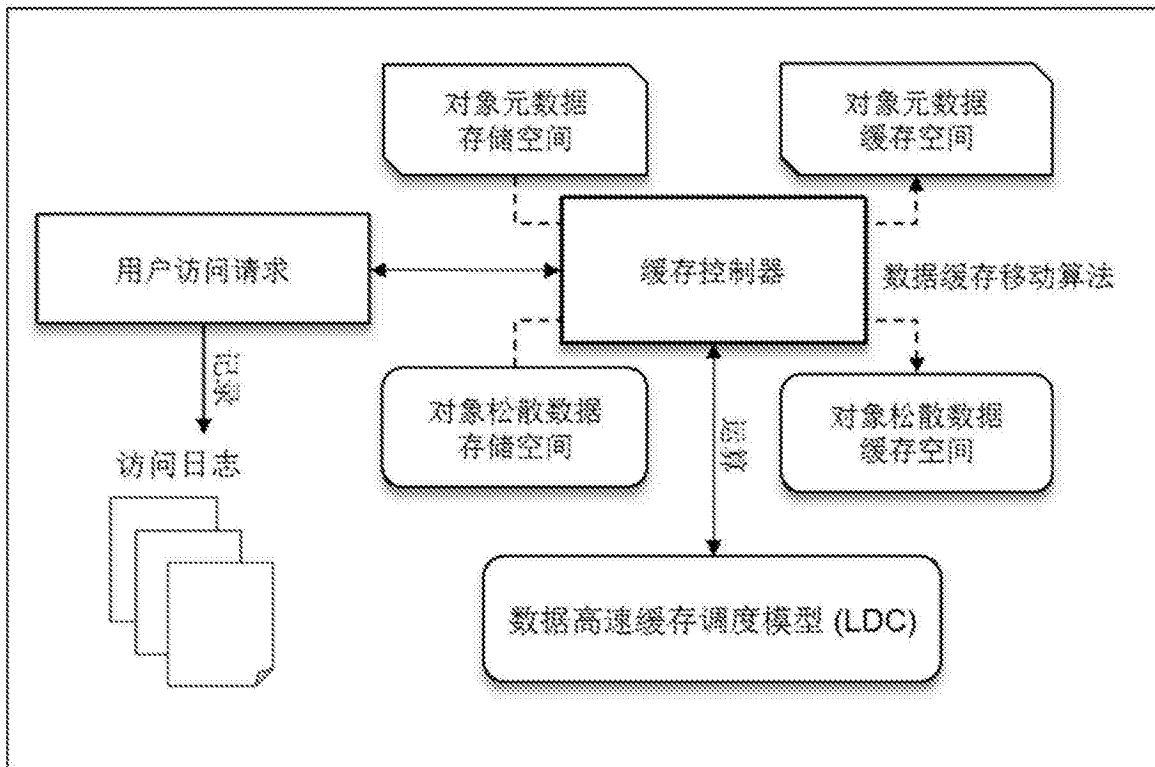


图2

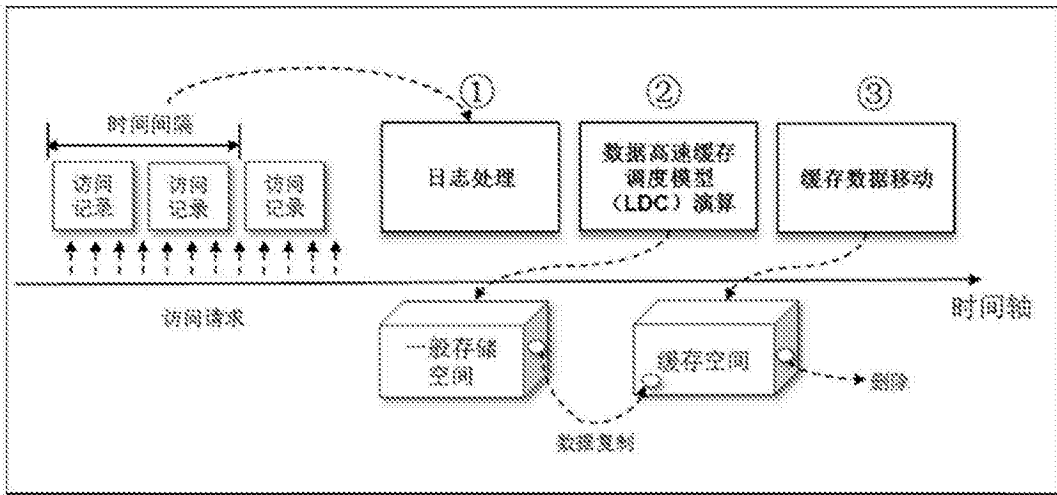


图3

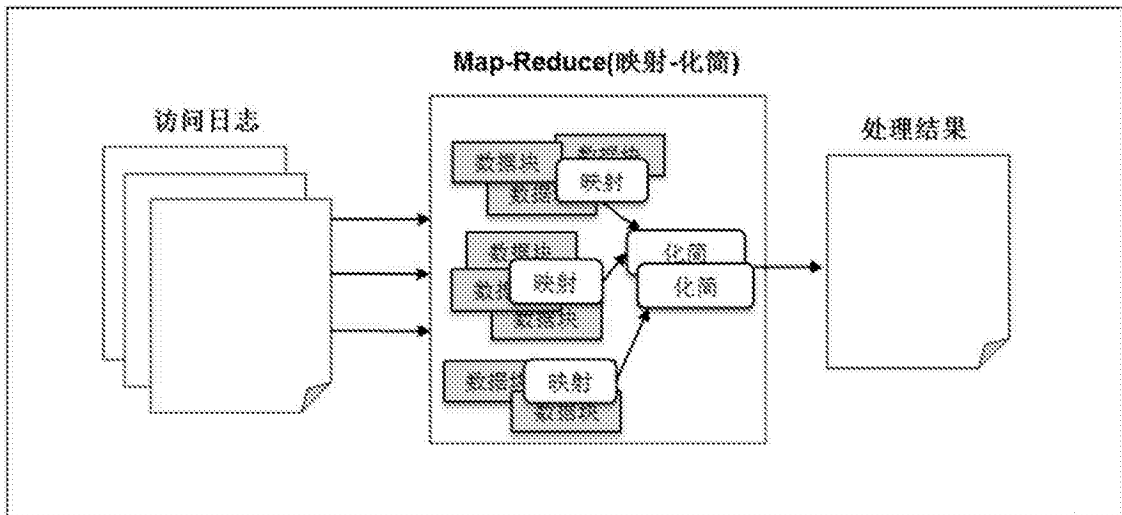


图4