

分级存储中自适应选择迁移策略的研究

胡 娜 李战怀 张 晓

(西北工业大学计算机学院 西安 710129)

摘 要 在分级存储中,业务数据的访问模式多样且多变,使得为其选择合适的数据迁移策略非常困难,为此引入迁移策略优先级的概念:系统内的迁移策略按优先级高低排序,每次启动数据迁移时,都选择当前优先级最高的策略作为迁移所用策略,并且在数据迁移进行时以及完成后对策略的优先级进行动态调整,从而达到系统自适应选择最优迁移策略的目的。原型实验证明,该方法确实能够根据优先级自适应地选择出最佳策略,从而降低数据迁移对前台应用的影响。

关键词 分级存储,迁移策略,优先级,自适应选择

中图分类号 TP302 **文献标识码** A

Adaptive Method to Select Policy in HSM

HU Na LI Zhan-huai ZHANG Xiao

(School of Computer Science and Technology, Northwestern Polytechnical University, Xi'an 710129, China)

Abstract In the hierarchical storage management (HSM), it is difficult to choose a data migration policy which meets precisely the diverse and the changes in business data access patterns, so that the concept of policy priority could help to quickly choose the most appropriate policy for the HSM system. The value of a policy priority is dynamically adjusted before and after a migration which is driven by the policy, and then, the HSM system chooses the policy that has highest priority value as the policy for the next migration. After a few adjustments like this, the system will obtain the optimal policy. The prototype experiments show that the method can indeed adaptively choose the best policy by dynamically adjusting the priority, therefore, the impact on the business applications is greatly reduced.

Keywords HSM, Migration policy, Priority, Adaptively choose

1 引言

分级存储管理(Hierarchical Storage Management, HSM)是为解决企业业务数据急剧膨胀问题而提出的存储理念。HSM 是指将数据按照其重要性、可用性、访问频率等特点存放到不同性能级别的存储设备(如 Cache、RAID、磁带库等),并且根据应用变化,可以实现这些数据在不同存储设备层次之间自动迁移^[1]。

数据的自动迁移由一定的迁移策略来实现。数据迁移策略是分级存储管理的核心研究内容,它直接决定了数据分级存储系统设计的优劣。若分级存储系统采用的迁移策略不能切合业务数据的访问模式,则易造成迁移抖动^[2],从而严重影响系统性能。但由于迁移策略与特定业务数据的访问模式(Access pattern)息息相关^[3],因此很难制定出高效能且具有高通用性的策略^[4]。针对这个问题,本文提出一种基于文件的自适应的策略选择方法,即通过动态调整策略优先级并按优先级排序策略,挑选出最适合当前业务访问模式的数据迁移策略。

2 相关工作

制定数据迁移策略首先是分析业务数据的访问模式,然后对数据对象的价值和活跃度进行排序。在满足一定条件时,把价值较高或者访问频率较高的数据存放在高性能的在线存储设备中,以提高系统整体的可靠性和性能。

数据迁移策略一般由数据分类规则 R 、迁移计划 Mp (migrate plan)和触发条件 C 部分组成,因而迁移策略 $Policy$ 可以形式化表示为:

$$Policy = \langle R, Mp, C \rangle$$

数据分类 R 就是根据数据所附带的辅助信息(如 meta-data 等)以及业务应用对数据的处理规律来评判数据价值的高低,价值低的数据即为迁移的目标数据。一般而言,数据分类主要从数据的静态特性和动态特性两方面进行,即根据这两方面的特征来制定数据分类规则 R 。

数据的静态特性主要有数据大小、数据用户数和数据的内容。动态特性分为(a)宏观访问规律,如文件系统中对文件大小访问的分布关系^[5]; (b)数据创建时间,如数据被创建初期其访问价值比较高,其后则很少被访问; (c)数据访问频度,

到稿日期:2011-04-03 返修日期:2011-08-01 本文受国家 863 重大项目(2009AA01A404)资助。

胡 娜(1985-),女,硕士生,主要研究方向为海量数据存储、数据容灾,E-mail:huna052654@163.com;李战怀(1961-),男,博士,博士生导师,主要研究方向为数据管理、数据存储。

数据访问次数越多,意味着其使用价值也越高;(d)数据间的关联度,即数据之间可能存在一定的业务相关性。

数据分类完成后,需要确定将哪些类(what)的数据迁移到哪一个(where)^[6]具体存储设备上,即制定所谓的迁移计划。一般处理是价值高的数据存储到高端设备上,价值低的数据存储到相对低端的设备上。策略触发条件就是确定策略何时(when)^[6]执行,策略可以根据业务数据所在设备的空间使用程度或者时间周期等因素被触发执行,还可以利用人为干预的方法来调整策略的执行。

目前有关分级存储数据迁移策略的研究重点是如何制定出切合业务数据特征的策略。如文献[9]提出了一种基于日志文件系统 LFS 层内容感知的块级数据迁移策略方法;文献[3]的大量实验验证了基于 space-time(时间和空间)的策略模式是最有效的;文献[8]提出了一种文件价值(FV)评定模型,引入文件访问频度、文件相关性等因素来综合评定文件数据的价值;针对策略查询效率问题,文献[12]提出了一种 Policy Cache(策略缓存)的方法来快速获得数据迁移时所需要的策略。这些方法都试图去解决数据迁移策略制定这个难题,但它们也只能将适宜的策略锁定到一定范围内,如何在这个策略集合内选出最佳策略,则只能由数据管理员一一去尝试。

如何为各式各样的应用环境制定适宜的数据迁移策略,一直是一个难以解决的问题。良好的数据迁移策略只能在业务数据访问模式比较固定并且数据管理员对这种模式很清楚的情况下才能确定下来。一旦数据访问模式呈不确定动态变化或者管理员无法确定数据访问模式,就很难制定有效的迁移策略。

3 基于带优先级的迁移策略

针对策略选择难这一问题,本文在迁移策略中引入策略优先级概念,即根据迁移策略对业务影响的大小来评定迁移策略的优劣。

3.1 带优先级的数据迁移策略

引入优先级 P (priority)的数据迁移策略表示如下:

$$Policy = \langle R, M_p, C, P \rangle$$

策略优先级是评判数据迁移策略符合业务数据特征程度的基准。策略越符合数据访问模式,则其优先级数值越高,否则优先级数值越低。

3.1.1 策略触发条件

分级存储系统根据策略的优先级选择下一次数据迁移的策略。为了不改变数据迁移的执行时机,这里将策略拆分成 $\langle R, M_p, P \rangle$ 和 $\langle C \rangle$ 两部分。当 $\langle C \rangle$ 集合中某一条件 C_i 满足时,就在 $\langle R, M_p, P \rangle$ 集合中选取优先级最高的一条,与 C_i 组合成迁移策略,从而启动一次数据迁移。

3.1.2 迁移事件

以基于文件的分级存储系统为例,定义一次数据迁移移动作为迁移事件(MigEvent),在该事件中被迁移的文件集合 $\{f_1, f_2, \dots, f_n\}$ 设为 F_{set} , 迁移事件发生时间设为 T_{ex} , 迁移所用到的策略为 $Policy$, 故迁移事件可以表示为:

$$MigEvent = \{Policy, F_{set}, T_{ex}\}$$

3.2 基于优先级的数据迁移策略调度

针对一次迁移事件,系统按一定规则为每条迁移策略计

算其优先级,然后按照优先顺序对策略进行排序,选择优先级高的作为后续数据迁移策略。

3.2.1 策略按优先级分组

假设系统策略的优先级的动态变化取值是 $[0, \theta]$ 之间的整数(θ 为一正整数常量)。为快速查找出优先级最高的策略,将策略按照优先级数值划分为 $\theta+1$ 组。选择策略时,总是从最大值 θ 所对应的策略组开始。若该组为空,则查找上一组,直到找到某一组存有策略,并取该组的第一条策略。如图 1 中第 θ 组中有两条策略 $Policy-7$ 和 $Policy-5$,且 $Policy-7$ 为该组第一条策略,故取其作为下一次迁移数据的策略。

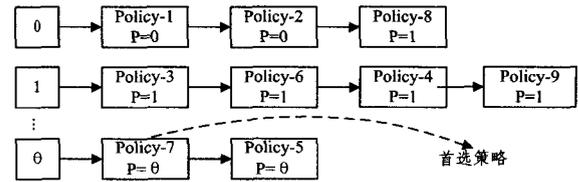


图 1 策略按优先级分组

3.2.2 策略优先级初始化

系统初始化时无法区分策略优劣,故各个策略均赋予相同的优先级,如图 2 所示。初始时,令所有策略优先级值为 1,且各个策略不分先后在第 1 策略组内随机排序。

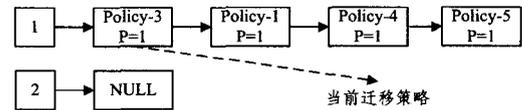


图 2 初始化时策略调度

在系统正常运行状态下,若有新的策略加入,这些策略往往是由用户根据实际情况定制的,因而其可能更符合业务数据的访问模式,故总是将其插入当前最高优先级策略组的队头位置,从而保证优先选取这些策略。

3.2.3 策略优先级调整

每一次迁移事件完成之后,相关的策略优先级都会重新调整。调整后的策略按照新的优先级数值插入到对应的优先级策略组队尾,以保证同组靠后的策略也能得到公平的调度,同时也保证了被降级的策略不会在下次再被选中。

如图 3 所示, $Policy-3$ 在迁移事件完成之后优先级由 5 降为 4,故将其从第 5 策略组移出,并插入至第 4 策略组末尾处,下一次预选的迁移策略就是 $Policy-1$,而不再是 $Policy-3$ 。

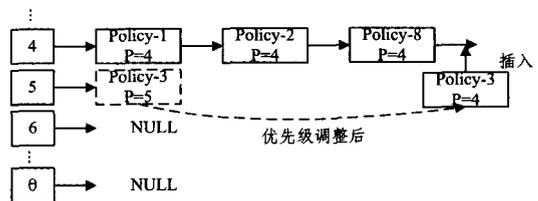


图 3 优先级调整后策略插入

4 策略优先级动态调整方法

由于数据分级存储的主要目的是将那些数据存储到价格相对低廉而速度也相对慢的设备上去,因此只有这些被评判为低价值的文件数据再次被访问的可能性很低,才不会对前

台应用造成过大的性能影响。故评价数据迁移策略优劣主要考察在使用该策略进行数据迁移时:

- ① 迁移过程中是否与上层数据请求存在冲突;
- ② 迁移之后是否由于业务需要而马上被迁回。

因此,系统以迁移事件为单位对策略优先级进行调整,其总体思想是:统计数据迁移过程中,等待迁移的文件集合由于被前台应用访问而取消迁移的文件数目,以及迁移完成后的一段时间内又被应用迁回的文件数目,然后根据这些统计数据重新调整策略的优先级。

4.1 迁移时文件状态的转换

为避免一次迁移事件中文件频繁地迁出、迁回,若迁移时某个文件被前台应用访问的次数过高,则要取消这个文件的迁移。对于一次迁移事件,需要统计其候选文件集合 $F_{set} = \{f_1, f_2, \dots, f_n\}$ 中的每一个文件在迁移过程中被前台应用访问的次数。设迁移队列中文件 f_i 的访问次数为 C_{f_i} 。在 f_i 的迁移过程中,若前台进程访问一次 f_i ,则 C_{f_i} 加 1;

另一方面,如果业务数据所在的源卷空间使用率较高,需要紧急将一些文件数据迁出,以释放更多的空间来存储价值更高的数据,此时即使文件被上层访问,也需要把该文件数据迁移。定义源卷使用率 U_{sf} 为源卷文件系统已使用的空间 ($Space_{used}$) 与源卷文件系统总空间 ($Space_{total}$) 的比值,即

$$U_{sf} = \frac{Space_{used}}{Space_{total}} \times 100\%$$

候选迁移文件 f_i 被取消迁移的条件是:① C_{f_i} 达到一定阈值 (threshold-value); 且② 源卷空间使用率低于限度。

使用常数 ϵ 表示候选文件被上层访问次数的阈值,易知 ϵ 取值应该大于等于 1;

使用常量 μ ($0 < \mu \leq 100\%$) 来表示空间使用率阈值。当空间使用率处于 $[0, \mu]$ 范围内时,可以对文件迁出进行宽松管理,即只有文件 f_i 的 $C_{f_i} \geq \epsilon$ 时才将 f_i 迁出。

设定迁移事件 $MigEvent$ 所涉及的各个文件在迁移过程中的状态变化,如图 4 所示。

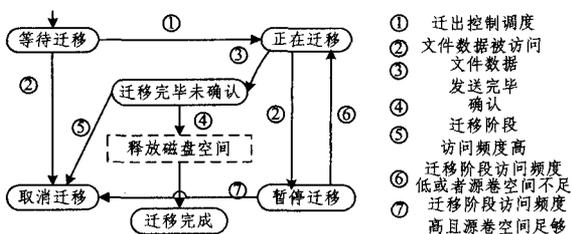


图 4 迁移文件状态转化图

被迁移的文件总是处于图 4 所示的状态转换图中的某一种状态:等待迁移、正在迁移、暂停迁移、迁移完成但未确认、迁移完成、取消迁移 6 个状态。启动迁移事件后,要迁移的目标文件标识都进入等待迁移状态,之后由迁出控制调度某一个文件的数据开始迁移;若正在迁移的文件被上层应用访问,则将其转入暂停状态;若有应用访问正在等待迁移的文件,则直接将该文件从迁移队列中取消。

当文件由于被上层应用访问而进入暂停迁移状态时,需要对其进行如下处理:

- (1) 若 $C_f \geq \epsilon$ 且 $U_{sf} \leq \mu$, 则取消迁移文件 f ;
- (2) 若 $C_f \geq \epsilon$ 且 $U_{sf} > \mu$, 则继续迁移文件 f ;

(3) 若 $C_f < \epsilon$, 则对 f 监控一段时间 $T_{monitor}$, 然后根据监控结束后 C_f 的值决定是否取消文件 f 的迁移。

同样,当文件数据迁移完毕后也要对文件监控一段时间 $T_{monitor}$ 并考察 C_f 的变化。监控结束之后,若文件 $C_f \geq \epsilon$, 则取消该文件迁移, 否则确认该文件迁移完成。数据迁移确认后,则释放文件数据块所对应的磁盘块 (block)。至此一个文件的数据迁移才算正式完成。

从图 4 可以看出,当取 $\epsilon = 1$ 时,若文件 f 被访问,其 C_f 就已经等于 1, 故不需要在其暂停迁移状态下考察监控 $T_{monitor}$ 时间,只要 $U_{sf} \leq \mu$, 即可以取消其迁移。

4.2 数据迁移时策略的优先级调整

迁移事件完成之后,根据统计的各个文件的 C_f 总数来调整事件对应策略的优先级。优先级调整依照以下两个原则进行:

① 计算历史迁移事件 (包括本次迁移事件) 中上层访问候选文件总次数的平均值, 并以此值为参考。若本次迁移事件文件访问总数大于该平均值, 则说明使用的策略对系统影响相对较大; 若小于该平均值, 则说明使用的策略对系统影响相对较小。

② 数据迁移是为了使源卷上能存储更多的高价值数据。迁移事件启动时,为了释放源卷空间而将一些访问频度较高的文件迁移至其他设备, 这样会导致文件数据频繁迁出、迁回, 势必对系统性能造成较大影响。因此,对于有因空间紧张而强制 (即 $C_f \geq \epsilon$ 且 $U_{sf} > \mu$) 迁出的文件,需要对 Policy 的优先级加大调整力度。

根据原则①, 设迁移事件 $MigEvent-m$ 的 F_{set} 中各文件的 C_f 总和为 C , 即有

$$C = \sum_{i=1}^n C_{f_i} \quad (1)$$

式中, \bar{C} 取值为系统中历次迁移事件集合 (包括当前迁移事件 $MigEvent-m$) C 的平均值。若系统中所有的迁移事件集合为 $\{MigEvent-1, \dots, MigEvent-m\}$, 则有

$$\bar{C} = \left(\sum_{i=1}^m C_i \right) / m \quad (2)$$

当 $MigEvent-m$ 为系统首次迁移事件时,有 $\bar{C} = C$ 。

根据原则②, 因条件 $\{C_f \geq \epsilon$ 且 $U_{sf} > \mu\}$ 而迁移 (即所谓强制迁移) 的文件需要和其他文件区分对待。设 $\{f_1, f_2, \dots, f_k\}$ 为迁移事件中被强制迁出的文件, $\{f_{k+1}, f_{k+2}, \dots, f_n\}$ 为 F_{set} 中其他类型文件。对于被强制迁出的文件,其 C_f 值调整如式 (3) 所示, 其中常量 β ($0 \leq \beta \leq 1$) 为空间影响因子, 其值越大说明空间影响越大:

$$C_{f_i} = (1 + \beta\mu) \cdot C_{f_i} \quad (f_i \in \{f_1, f_2, \dots, f_k\}) \quad (3)$$

从而 C 值计算亦需做相应调整:

$$C = (1 + \beta\mu) \cdot \left(\sum_{i=1}^k C_{f_i} \right) + \sum_{j=k+1}^n C_{f_j} \quad (4)$$

式中, $f_i \in \{f_1, f_2, \dots, f_k\}$, $f_j \in \{f_{k+1}, f_{k+2}, \dots, f_n\}$ 。

最后,依照式 (3) 和式 (4) 重新计算 \bar{C} 。

确定 C 和 \bar{C} 之后,就可计算策略的新优先级。如上所述,策略的优先级总是在范围 $[0, \theta]$ 内,对策略优先级进行调整之后也要保证调整的优先值仍落在此范围内。考虑双曲正切函数 $\tanh(x)$ 在 $[-\infty, +\infty]$ 内单调递增, 且 $\lim_{x \rightarrow \infty} \tanh(x) =$

1. 为此,选取 $\tanh(x)$ 来限定调整后数值的变动范围。

按式(5)计算策略新优先级 P_{new} :

$$P_{new} = [1 + \alpha_1 \cdot \tanh(\bar{C} - C)] \cdot P \quad (0 < \alpha_1 \leq 1) \quad (5)$$

式中,常量 α_1 称为迁出优先级调整因子,取值越大,策略优先级调整梯度越大。调整后,若 $P_{new} > \theta$,则令 $P_{new} = \theta$ 。

从式(5)可以看出:

a) 若 $C < \bar{C}$, 有 $(\bar{C} - C) > 0$, 即 $\tanh(\bar{C} - C) > 0$, 故 $P_{new} > P$, 即策略优先级提升;

b) 若 $C \geq \bar{C}$, 有 $(\bar{C} - C) \leq 0$, 即 $\tanh(\bar{C} - C) \leq 0$, 故 $P_{new} \leq P$, 即策略优先级降低或者不变。

把当前迁移事件中的 C 与历史平均访问次数 \bar{C} 相比,能充分体现当前策略的优劣。若 \bar{C} 相对 C 较低,则说明使用当前策略迁移数据引起业务系统较大波动,即该策略不适用当前数据访问模式,从而需要降低其优先级;若 \bar{C} 相对 C 较大,则说明使用当前策略迁移数据没有对前台应用造成相对较大的影响,即该策略比较符合当前业务数据访问模式,从而需要提升其优先级。

4.3 数据迁移后策略优先级的调整

根据分级存储的设计目标,若迁出的文件在短时间内被请求回迁,说明迁移策略不适用当前数据访问模式,需要对策略进行降级;反之,若迁出的文件在很长一段时间内没有被访问,则说明此次迁移是成功的,应该提高相关迁移事件对应的策略优先级。

一次迁移事件完成之后,系统跟踪各个文件迁出后的动向,统计在 T_{after} 时间内请求迁回的文件数目,并依此调整策略优先级。具体处理如下:

对于当前迁移事件 $MigEvent-m$, 统计 T_{after} 时间内其对应的候选文件集合 F_{set} 中被迁回的文件数目 C_{back} , 并记录系统中历史平均迁回文件数目 \bar{C}_{back} 。对于系统历史迁移事件集合(包括 $MigEvent-m$) $\{MigEvent-1, \dots, MigEvent-m\}$, 有

$$\bar{C}_{back} = (\sum_{i=1}^m C_{back-i}) / m \quad (6)$$

然后对该事件所对应策略的优先级做如下调整:

$$P_{new} = [1 + \alpha_2 \cdot \tanh(\bar{C}_{back} - C_{back})] \cdot P \quad (0 < \alpha_2 \leq 1) \quad (7)$$

常量 α_2 称为迁回优先级调整因子,其值越大则说明文件回迁对策略优先级影响越大。调整后,若 $P_{new} > \theta$,则令 $P_{new} = \theta$ 。

从式(7)可以看出:

a) 当 $C_{back} < \bar{C}_{back}$, 有 $\tanh(\bar{C}_{back} - C_{back}) > 0$, $P_{new} > P$, 即策略优先级被提升;

b) 当 $C_{back} \geq \bar{C}_{back}$, 有 $\tanh(\bar{C}_{back} - C_{back}) \leq 0$, $P_{new} \leq P$, 即策略优先级被降低或者不变。

C_{back} 低于 \bar{C}_{back} , 说明相对迁移历史而言,迁移事件 $MigEvent-m$ 所用的策略比较适合当前数据访问模式,故其优先级应该提升;反之, C_{back} 高于 \bar{C}_{back} , 说明所用策略迁移的数据在短期内又被迁回,对系统造成较大的性能影响,因而需要降低该策略的优先级。

5 实验实现与分析

5.1 原型系统

为验证自适应迁移策略选择方法的有效性,开发了一个

基于文件的分级存储管理原型系统 APHSM (Adaptive-policy HSM),其体系结构如图 5 所示。

APHSM 主要分为 HSM-Server, Client 和 Target 3 个部分。HSM-Server 模块主要负责管理数据迁移策略、系统存储资源、通知 Client 端启动数据迁移等; Client 是业务数据系统,当 HSM-Server 对其进行数据迁移启动时, Client 按照指示执行数据迁移操作; Target 即为数据迁移的目的存储设备,主要分为本地 Target (Local-Target, 如 SATA 磁盘组) 和网络 Target (Net-Target, 如远程磁带备份中心)。

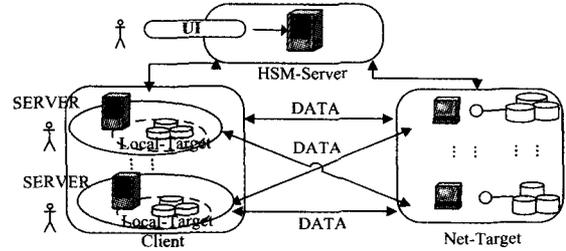


图 5 APHSM 体系结构图

HSM-Server 选择当前优先级最高的策略定义迁移事件,然后将该事件发往对应的 Client。Client 启动这个迁移事件,迁移完后将迁移期间的统计数字提交给 HSM-Server,后者计算新的策略优先级,同时对完成的迁移事件进行统计监控,以便再次调整策略优先级。

5.2 实验测试

自适应选择符合业务数据访问模式的迁移策略,在性能上应该体现为一个对上层 IO 响应时间影响从大到小的动态调整过程。即初始时,各个策略优先级一致,选择的策略可能不符合业务数据特征,需要经过数次优先级调整,才能找到最适合业务数据特征的迁移策略。为验证自适应选择算法的有效性,在原型系统 APHSM 上设计如下实验。

假定应用环境为数据采集和在线分析系统。该系统不断地有大大小小的文件被写入,数据分析进程优先处理先来的文件,处理过的文件基本上不再访问,即较旧的数据可以被迁移到次级存储设备上。该数据在线采集和分析系统由两个用户级线程 A 和 B 来模拟实现,其中线程 A 不断地创建新文件,线程 B 模拟数据分析,即读出线程 A 产生的文件数据进行简单处理。实验测试过程中,以线程 A 读源卷文件数据的 IO 响应时间(ms)为目标测试数据。

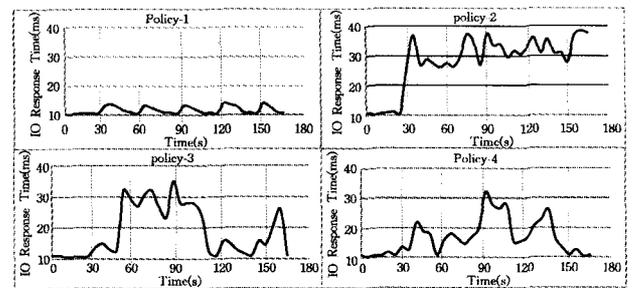


图 6 单条策略迁移数据对文件 IO 响应的影响

源卷初始情况为空,设 $\mu = 80\%$, $\epsilon = 2$, $\theta = 16$, $T_{monitor} = T_{after} = 0.5s$, $\alpha_1 = \alpha_2 = \beta = 1$, 触发条件是每隔 $t = 30s$ 启动一次迁移事件。

Client 使用两颗 Dual Intel Nehalem 5502 双核存储专用处

理器,4GB DDR3(1333MHz)内存,千兆网卡,采用 Redhat Linux 2.6.27_x86_64 系统;一级存储为 100GB FC 磁盘存储,本地二级存储为 SATA 磁盘 300GB,均采用 ext3 文件系统。HSM_Server 通过 TCP/IP 协议传输数据,采用 MySQL 存储文件系统元数据和策略相关信息。

系统中设定的策略如下:

Policy-1:按创建时间分类文件数据,迁出较旧的数据;

Policy-2:按创建时间分类文件数据,迁出较新的数据;

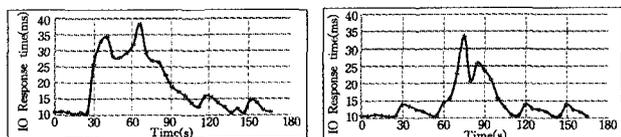
Policy-3:按文件数据大小分类,迁出较大的文件;

Policy-4:按文件数据大小分类,迁出较小的文件。

若只使用其中一条策略,各个策略对应的文件 IO 响应时间如图 6 所示。

若 4 条策略同时存在系统且 *Policy-1* 和 *Policy-3* 为系统预定义策略时,*Policy-2* 和 *Policy-4* 为用户在中途自定义的迁移策略。自适应选择迁移策略过程中对文件 IO 响应的影响如图 7(a)所示。

若 *Policy-2* 和 *Policy-4* 为系统预定义策略,*Policy-1* 和 *Policy-3* 标记为用户在中途自定义的策略。自适应选择迁移策略过程中对文件 IO 响应的影响如图 7(b)所示。



(a) 自适应选择过程对文件 IO 响应的影响 (b) 自适应选择过程对文件 IO 响应的影响

图 7

5.3 实验分析

根据应用环境的假设,*Policy-1* 应该是最符合业务应用模式的迁移策略。从图 6 可以看出,使用 *Policy-1* 迁移数据时对系统影响最小;使用 *Policy-3* 与 *Policy-4* 对系统的影响相当;使用 *Policy-2* 迁移数据对系统影响最大,这是因为其每次都把最新产生的文件数据迁移到次级设备上,稍后线程 A 请求这些数据时,系统总是要先迁回这些数据后才能做出响应,这大大增加了 IO 响应时间。

如 3.2.2 节所述,在初始时刻,用户定义的迁移策略总是优先被选中。图 7(a)中优先选择用户自定义策略,而最佳策略 *Policy-1* 为系统预定义,故调整了 3 次才使得对 IO 文件带宽影响最小;图 7(b)中最先选择的就用户自定义的 *Policy-1*,即一开始即得到最佳策略。但根据式(5)和式(7),第一次优先级调整不变,而调整后的策略放入了同组的队末,故中间有两次调整,并最终选定 *Policy-1* 作为最佳策略。由图 7(a)和图 7(b)可以看出,系统最初选择的策略对文件 IO 带宽影响较大,经过数次调整之后影响趋于最小。

以上实验证明,自适应策略选择算法确实能够有效地帮助系统自动挑选出最适合业务数据特征的迁移策略。

结束语 本文针对现有分级存储系统难以定制确切符合业务数据特征的迁移策略问题,提出了一种自适应选择迁移策略的方法。在文件级迁移基础上,对策略的优先级进行动

态调整,使得分级存储系统可以在众多策略中自动挑选出最佳数据迁移策略。原型实验也验证了本方法的有效性。但是本方法也存在不足,即它是基于分级存储系统中已有迁移策略上的,若系统中没有最符合的策略,则只能选出当中局部最佳的迁移策略。如何感知业务系统处理数据的规律,并根据这些规律系统自动定义出全局最佳的迁移策略,是下一步研究工作的重点。

参考文献

- [1] SNIA tutorial. DMF-ILM and Tiered Storage[EB/OL]. <http://www.snia.org>, 2007
- [2] Yin L, Uttamchandani S, Katz R. SmartMig: Risk-modulated proactive data migration for maximizing storage system utility [C]//Proceeding of NASA/IEEE Conference on Mass Storage Systems and Technologies. USA; IEEE Computer Society, 2006: 136-149
- [3] Kohl J T, Staelin C, Stonebraker M. HighLight: a file system for tertiary storage[C]//Proceedings of Twelfth IEEE Symposium on Mass Storage Systems. NW USA; IEEE Computer Society, 1993;157-161
- [4] Zhao X N, Li Z H, Zeng L J. A Hierarchical Storage Strategy Based on Block-level Data Valuation[C]// Proceedings of the 2008 Fourth International Conference on Networked Computing and Advanced Information Management. NW USA; IEEE Computer Society, 2008;36-41
- [5] 舒继武. 分级存储与管理 [J]. 中国教育网络, 2007(7)
- [6] Bhadkamkar M, Guerra J, Useche L, et al. BORG: Block-reOrganization for Self-optimizing Storage Systems[A]//Proceeding of USENIX FAST[C]. 2009;1-12
- [7] Massiglia P. Exploiting Multi-tier File Storage Effectively[M]. SNIA Tutorial. SNIA, 2009
- [8] 吕帅, 刘光明, 徐凯, 等. 海量信息分级存储数据迁移策略研究 [J]. 计算机工程与技术, 2009, 31(A1): 163-167
- [9] Watson R W. High performance storage system scalability: architecture, implementation and experience[C]// Proceedings of 22nd IEEE/13th NASA Mass Storage Systems and Technologies Goddard Conference. USA; IEEE Computer Society, 2005: 145-159
- [10] 吕帅. 基于对象的分级存储系统数据迁移技术研究[D]. 长沙: 国防科技大学, 2009
- [11] Strange S. Analysis of Long-term UNIX File Access Patterns for Application to Automatic File Migration Strategies[R]. UCB/CSD Report 92-700. Berkeley, CA, 1992
- [12] Verma A, et al. An architecture for lifecycle management in very large file systems[C]// Proceedings of the 22nd IEEE/13th NASA Goddard Conference on Mass Storage Systems and Technologies. Monterey USA; IEEE Computer Society. 2005; 160-168
- [13] Yalamanchili C, Vijayasankar K, Zadok E. DHIS: discriminating hierarchical storage[C]// Proceedings of SYSTOR 2009. NY USA; ACM, 2009;1-12