一种流媒体系统中文件存储分配算法①

李　洋　李　俊　(中国科学技术大学 信息科学与技术学院 安徽 合肥 230027)

摘　要： 随着集群搭建的高性能计算中心快速发展，海量数据处理中 CPU 和内存的速度已经有了质的飞跃，但是 I/O 部分的速度仍是制约整个系统性能的瓶颈。本论文目的在于通过深入研究流媒体系统的特性，来提取出一种高效、均衡的文件存储分配策略，增大服务并行度并且减小服务延时。首先对媒体文件进行特征评估，然后通过存储调度器进行磁盘间的均衡性平摊，进而应用更细粒度的分划,做到每个磁盘内部的存储分布优化。仿真结果验证了算法的有效性，这样的分配策略可使盘间、盘内在数据组织上更为合理，有效的提高了并行服务能力，提供了更加迅捷的用户体验。

关键词： 流媒体系统；存储分配；均衡性平摊；并行服务

File Storage Distribution Algorithm in Streaming Media System

LI Yang, LI Jun (School of Information Science & Technology, University of Science and Technology of China, Hefei 230027, China)

Abstract: With the development of high performance center build on server cluster, the speed of CPU and Memory have gained a big leap in massive data processing, but the unmatched I/O speed is still a bottleneck of whole system. This paper digs into streaming media system's specific characters deeply in order to find out a more efficient, balanced way of file storage distribution, increasing degree of parallelism while reducing service time delay. First, give an evaluation of each media file based on target features, then distribute all files onto different disks through storage scheduler, finally there is a fine grit division to guarantee storage inside one disk is highly optimized. The simulation results verify the validity of the algorithm, this strategy of storage allocation not only improves data organization among disks, but also works well in one disk, which brings a better concurrent service and user experience.

Keywords: streaming media system; storage distribution; balanced divide; concurrent service

1　引言

流媒体应用具有数据处理量大、带宽要求高、存储容量多、实时性强等一系列特点[1,2]，目前服务提供者为了优化服务质量，一般会选择采用由多台服务器来搭建集群的方式来提供诸如 VOD(Video-On-Demand) 和 TVOD(TV-On-Demand)这样的视频点播服务。但是实际环境中遇到的情况是：一方面摩尔定律预测了 CPU 的速度每 18 个月增加一倍，吉尔德定律预测了主干网的速度每 6 个月增加一倍，同时内存的速度也随时间增长很快；另一方面外部存储设备如磁带、硬盘等主要的增长方向在存储容量而并非读写速度上，这就导致了 I/O 部分成为制约整个系统的瓶颈。目前，解决这一问题的较为有效方式就是增加 I/O 的并行性[3,4]，具体来讲就是由位于计算设备和存储设备之间的调度器统一规划，通过合理的策略将文件的存储和访问任务平摊到相互独立的各个磁盘上来减少延时[5]。人们针对此问题做了大量的研究工作，已经有很多好的方法如镜像法、条带法、基于点播率的动态调整法和启发式文件分类分配法等被提出[6-9]。但是现阶段提出的方法存在

① 基金项目:国家高技术研究发展计划(863)重大专项(2008AA01A317);安徽省高校自然科研项目(KJ2008A106)
收稿时间:2010-03-30;收到修改稿时间:2010-04-27

着一定的缺陷：只侧重于从均衡划分存储容量的角度来考虑而并未涉及均衡划分文件热度，导致出现虽然各磁盘占用容量大体一致，但系统集中访问某一磁盘上热门影片的情况，实际上没有起到预期的并行处理效果。本文目的在于将文件热度的因素也纳入到存储分配的策略中来，以达到文件在磁盘间分配的最终效果在容量和热度上都处于一个比较均衡的状态，使磁盘组能更为有效的为整个点播系统服务。仿真结果表明，在经过本论文所提出算法的改进后，存储系统表现出了更加可观的并行性和良好的用户体验。

2 文件的盘间存储分配模型

集群在处理批量的流化任务时，一般要通过存储调度器把 I/O 任务分布到磁盘阵列的各个硬盘上，目标是尽可能的达到一种“平衡”的工作状态。已经被提出并实现的方案主要有轮转分配法，轮转分配法，容量平均分配法等几种，但这些方法都有局限和不足，因此本文提出评估因子配比法。

2.1 模型建立

用来搭建流媒体服务器的集群在架构上可分为三个子部分。其中，计算子系统由刀片服务器组成，存储子系统由磁盘阵列所划分出的 LUN 组成，网络连接子系统 N 由光纤和光交换机组成。处于其中的调度器往往是一台高性能服务器。整个系统通常采用如图 1 所示的拓扑结构互连在一起。

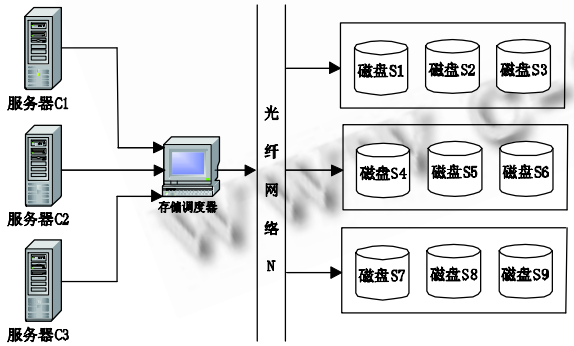

图 1 集群拓扑结构图示

2.2 决策变量和参数

该模型中将用到以下参数：

M：总服务器数

N：总磁盘数

C_i：第 i 个服务器的服务能力 (i=1~M)

D_j：第 j 块磁盘的磁盘容量 (j=1~N)

d_j：第 j 块磁盘目前可用容量

H_j：第 j 块磁盘目前存储文件总的热度

F_{mn}：存储在第 m 块磁盘上的第 n 个媒体文件

S_{mn}：第 m 块磁盘上的第 n 个文件的文件大小

H_{mn}：第 m 块磁盘上的第 n 个文件的文件热度

D_{avg}：当前总的磁盘平均可用空间

H_{avg}：当前总的磁盘平均存储热度

根据这些参数，如果文件 F_{ju}~F_{jv} 分布在 D_j 上，可以得到如下的结论

$$D_j = d_j + \sum_{k=u}^{v} S_{jk} \tag{1}$$

$$H_j = \sum_{k=u}^{v} H_{jk} \tag{2}$$

$$D_{avg} = \frac{\sum_{j=1}^{N} D_j - \sum_m \sum_n S_{mn}}{N} \tag{3}$$

$$H_{avg} = \frac{\sum_m \sum_n H_{mn}}{N} \tag{4}$$

2.3 确立约束条件

在此模型中必须达到的五点约束条件是：

① 批量流化任务中总的媒体文件大小之和必须小于或等于所有磁盘容量之和，则有

$$\sum_m \sum_n S_{mn} \le \sum_j D_j \tag{5}$$

② 单个硬盘上存储所有影片(F_u~F_v)大小之和必须小于其固有存储容量，则有

$$\sum_{n=u}^{v} S_{jn} \le D_j \tag{6}$$

③ 所有媒体文件的热度值和影片大小都必须是非负数值，则有

$$S_{mn} \ge 0, H_{mn} \ge 0 \tag{7}$$

上面三点是显而易见的，另外两点就是由文件热度和磁盘存储量的均衡要求所限制的。

④ 考虑所有的磁盘，其中出现的最大存储热度值与最小存储热度值之差不能超过规定的阈值 H_{thres}，则有

$$|H_{\max} - H_{\min}| \le H_{thres} \tag{8}$$

⑤ 考虑所有的磁盘，其中出现的最大存储占用空

间与最小存储占用之差不能超过规定的阈值 D_{thres}，则有

$$|S_{\max} - S_{\min}| \leq D_{thres} \quad (9)$$

3 评估因子的概念

因为磁盘间文件存储划分要兼顾文件热度和磁盘占用容量尽可能平均，实际上属于一个多目标优化问题。对于求解这类问题，通常的做法有以下几种：

1)根据有效解的定义，求解出许多有效解

2)多目标问题转化成单目标优化问题求解

3)问题转化成多层单目标优化问题进行求解

4)非统一模型法。

5)直接法

本文使用方法 2 类别当中的线性加权和法。即根据实际问题给每个目标参数一个相应的权重，然后将其加权和作为合并后的单目标。现在模型中主要考察的参数有两个：热度 H_{mn} 和大小 S_{mn}。并且依据实际情况中获取的经验，热度分配的不均匀对系统的 I/O 延时影响更为直接，相对来说容量分配的不均匀并不严重影响速度，只需要防止系统出现极端的存储容量失衡情况。由上述分析可得出加权和计算函数：

$$f(H_{mn},S_{mn}) = Fac_H \times H_{mn} + Fac_S / S_{mn} \quad (10)$$

各参数取值为 $Fac_H = \frac{\sqrt{5}-1}{2}, Fac_S = 1 - Fac_H$，$H_{mn} = 0 \sim 10$ S_{mn};是影片大小和 1GB 比值的归一化结果。这个函数值定义为一部影片的评估因子，通过公式(10)的转换，热度和大小的多目标优化问题就变成了评估因子的单目标优化问题。

假设现在有 $D_1 \sim D_N$ 是容量一致的 N 块磁盘，那么根据公式(10)计算出它们的评估因子分别为 $f_1, f_2, \cdots, f_N$，而所有磁盘评估因子和的算术平均值为 $\overline{f} = \sum_{i=1}^{N} f_i / N$。结合对模型中约束条件的讨论，可以进一步得出目标函数：

$$f_{target}(N) = \sum_{i=1}^{N} \left(f_i - \overline{f} \right)^2 \quad (11)$$

最优化目标就是在存储分配后能使整个系统的 $f_{target}(N)$ 达到最小值即 $f_{target}(N)_{\min}$。

4 算法流程

4.1 模型的输入和输出

输入：总磁盘数 N，每磁盘容量 D（假设同构性系统中磁盘大小一致），以及由 k 个媒体文件构成的文件组 Fgrp，形式上设定为给定了相应热度和大小的一个文件列表{(P1,Q1), (P2,Q2),…, (Pk,Qk)}。

输出：Fgrp 最终在磁盘间的分布情况，包括各个磁盘上最终的占用空间和文件热度之和。

目标：使文件组以较为均衡的热度和大小被散布到存储介质上，同时通过多次随机点播实验来统计出平均响应时间以验证本算法确实增加了系统的并行服务能力。

4.2 算法的详细步骤

其步骤可以使用伪代码表示如下

1) 初始化所有磁盘上的文件存储量和热度和为零，各磁盘上文件分配列表均为空表。

2) 按照文件的大小将文件列表进行升序排列，同时计算出文件组的评估因子平均值 favg。

3) 从文件列表起始位置进行热度匹配工作。

```
设定当前等待分配的磁盘编号 Dcurr=1，
while (还有文件没有被分配到磁盘上去)
{
从头选择第一个还没有被分配的文件 file1;
file1 评估因子=fbeg;
与 file1 匹配的 file2 因子 fend=2xfavg-fbeg;
    fmin=Max;
  for(file=列表尾;file 非列表头;file 未分配)
  {
    if(abs(fend-ffile) < fmin)
        fmin= abs(fend-ffile);
  }
    选择 fmin 对应的 file 为 file2;
    file1 和 file2 都保存到 Dcurr 上，同时更新 Dcurr 的文件存储量和文件热度值数据;
    Dcurr =(Dcurr+1)%N;
}
```

5 仿真结果及分析

选用 12 部测试影片和 3 个磁盘组成的阵列作为实验环境，利用 C 语言和 Matlab 进行编程仿真。实验的初始和结果数据如表 1、表 2、图 2、图 3 所示。

表 1 初始文件组特征值

12 个用作测试的影片文件的热度/大小												
热度(0~10)	2.6	7.5	3.4	6.1	9.9	1.1	9.2	8.8	1.4	8.4	5.6	4.4
大小(/1GB)	0.53	0.76	0.95	1.00	1.23	1.27	1.56	1.89	2.07	2.21	2.35	2.40
评估因子值	2.33	5.14	2.50	4.15	6.43	0.98	5.93	5.64	1.05	5.36	3.62	2.88

表 2 多个存储算法分配后各磁盘热度和/大小和数

	Disk1	Disk2	Disk3
轮询算法	31.8/5.3	26.3/6.23	10.3/6.69
随机算法(1)	8.5/4.82	32/8.01	27.9/5.39
随机算法(2)	23/6.65	15.5/6.61	29.9/4.96
容量平均算法	17.3/5.76	31.8/6.23	19.3/6.23
评估因子配比算法	22.3/6.04	21.2/5.54	24.9/6.64

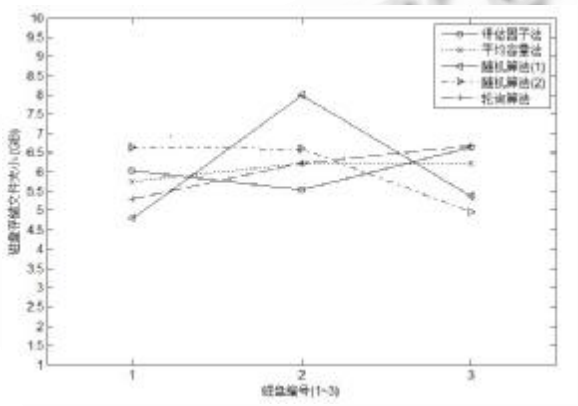

图 2 各种算法下文件容量分配的情况

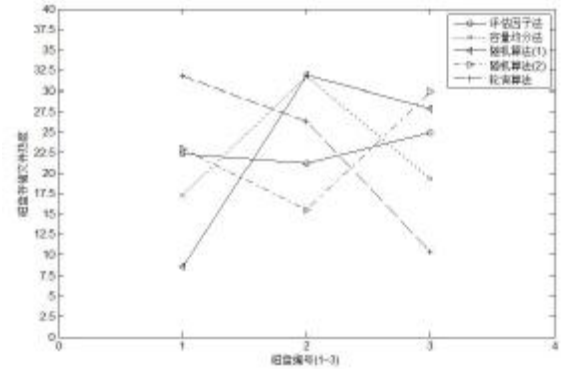

图 3 各种算法下文件热度分配的情况

由表 2、图 2 和图 3 得出的数据和曲线可以分析得知，轮询算法在设计思路和实现上最简单，但是其性能得不到保证，无论是对文件热度还是文件大小的划分上都比较失衡；随机算法的表现具有很大的不确定性，在两轮实验中得到的曲线对应比较可以看出其性能上的波动幅度较大，表现好的情况接近最优的算法，表现差的情况也趋近于最劣值；而容量平均算法对容量分配时表现的最好，但是它在热度的分配上偏离均匀状态很远，甚至不如随机算法在某些情况下的性能；评估因子算法虽然在容量分配上略差于容量平均算法，但是它兼顾了容量和热度的均摊划分，整体性能表现出色，在文件大小和热度两条分配曲线上都很平稳。

在利用上述的轮询、随机、容量均分和评估因子四种算法进行对文件组的存储分配后，通过对磁盘上流媒体文件的大量随机点播实验后统计出的平均响应时间可以大致分析出本文提出的算法给并行服务带来了怎样的性能改变。具体实验方法是通过程序脚本设定一个随机的点播序列向服务器提出请求，序列中每部影片被点播的次数和它的热度值成正比关系，然后记录每次从请求发出到请求得到回应过程中的时延。因为每种分配算法都能够事先确定文件与磁盘的对应存储关系，所以很容易的统计出每个磁盘提供服务的累计时间延迟，此数值再对其所承担的点播请求数做算术平均，就可以近似得到在统计意义上的该磁盘对每点播请求的响应延时。那么，比较不同算法之下整体曲线的相对位置高低就可以做出性能优劣的判断；同时，对于每种算法来说，观察自身曲线上三个点的呈平缓或陡峭分布就可以得出是否将 I/O 负载分配均衡，对于改善并行性是否有积极的意义。

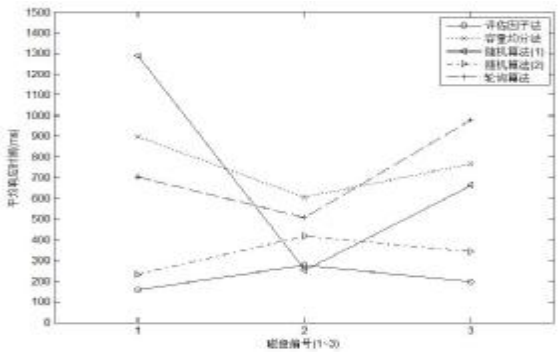

图 4　各种算法下磁盘平均响应延时

由图 4 可以看出，各条响应延时曲线走向是和几种算法在容量分配和热度分配中的两条曲线走向综合起来的情况大体趋于一致的。轮询算法中整体响应时延大，同时曲线在各磁盘间的变化很不稳定，是 I/O 任务量没有被很好的分发的具体表现。随机算法在第一轮实验中表现的非常差，整体响应延时甚至达到了最高峰值，同时曲线在各磁盘间走向十分陡峭，说明负载是一种极不均衡的状态；在第二轮实验中表现的非常好，整体响应值已经非常接近最低谷值，并且曲线在各磁盘间走向比较平缓，说明负载已经被较好的分布到了各个磁盘上。由此也可以验证前面所讨论的随机算法的固有特点，本身带有非常大的不确定性，没有很好的利用到文件组中不同文件的特征值来制定更为合理的存储方案，很难应用在实际环境中。而容量均分法虽然在整体响应时间上表现介于最优和最差中间，曲线在磁盘间走向也不是非常的陡峭，但是可以看出它的性能只能说是中等类别的，甚至不如随机算法在第二轮的表现。根本原因在于，这种算法只追求容量上的绝对平均，而有可能导致热度高的片子集中存储在一个磁盘上，而热度低的片子集中在另外一个磁盘上。这样“冷热”不均的存储会导致接受用户点播时，热门影片所在的磁盘频繁收到访问请求延时很大，而另外一方面冷门影片所在磁盘基本没有访问量延时较小但丧失了其剩余的服务能力。评估因子法因为在算法设计上就兼顾了热度和容量两方面的因素，采用互补配对的方式存储，因而性能提升比较突出。可以从图 4 中看到，整体响应时间上该算法最小，并且曲线在磁盘间走向十分平稳，I/O 负载在三个磁盘上分布的比较均衡，并行服务度高。

6　结论

本文通过研究流媒体系统中文件的特征，发现目前基于集群的并行服务模型中存在 I/O 速度瓶颈的问题，并针对此问题提出了一种利用评估因子的存储分配算法。该方法通过评估媒体文件的特征值来兼顾对文件热度和文件大小的同时均衡处理，仿真实验证明此方法有效的提高了系统的并发能力和响应速度。但目前尚未考虑到的一些情况如巨型媒体文件在磁盘间的切分和存储工作，求解多目标问题时所采用的线性加权和法可尝试其他加权参数，以及在大量磁盘、大量文件、极多请求下可能出现的一些极端情况的有效处理等。这些问题留待今后的进一步研究。

参考文献

1 Long X, Li ZZ, Gao XP. A new method to improve the I/O efficiency on network of workstations. Journal of Computer Research & Development, 2000,37(6):650－656.

2 Abawajy JH. Performance analysis of parallel I/O scheduling approaches on cluster computing systems. Proceedings of the 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid(CCGRID’03). 1992:724－729.

3 Gemmell J, Christodoulakis S. Principles of delay sensitive multimedia data storage and retrieval. ACM Trans Office Inform System, 1992,10(1):51－59.

4 Jadav D, Choudhary A. Designing and implementing high-performance media-on-demand servers. IEEE Parallel & Distributed Technology, 1995,3(2):29－39.

5 谢长生,顾鹏,刘朝斌. 流媒体服务器存储子系统高性能 I/O 接口的研究与实现. 计算机工程与科学, 2004,26(4):4－7.

6 刘衡竹,陈灿旭,陈福接. 视频服务器中视频流存储分配算法的研究. 计算机学报, 1998,21(4):289－295.

7 曾碧卿,陈志刚,陈恒法.一种分布式并行 I/O 中新型动态数据调度算法.微电子学与计算机, 2007, 24(7):119－121.

8 曾碧卿,陈志刚,刘安丰,曾志文. 一种集群计算系统中并行 I/O 文件存储分配策略. 小型微型计算机系统, 2005,26(5):873－876.

9 黄伟,谢冬青. 视频数据的存储调度优化策略. 电脑与信息技术, 2006,14(2):55－59.