СРАВНЕНИЕ МЕТОДОВ ОРГАНИЗАЦИИ ДИСКОВОГО ПРОСТРАНСТВА ФАЙЛОВЫХ СЕРВЕРОВ

М. С. Косяков (Санкт-Петербург)

1. Введение

Рост объема цифровых неструктурированных данных привел к появлению систем централизованного хранения, поиска и доставки документов по сети, таких как корпоративные цифровые библиотеки, приложения News-On-Demand, системы коллективной подготовки и обработки информации и т. д. При этом средства Web-серверов обеспечивают процессы публикации и распространения документов, сбора и хранения организующей информации, необходимой для их поиска [1]. Для хранения и воспроизведения запрашиваемых данных в таких серверах используется дисковая подсистема ввода/вывода. Однако в связи с тем, что обращения к дисковой подсистеме ввода/вывода выполняются на порядок медленнее, чем к другим более быстрым системам памяти, ее производительность становиться доминирующим фактором в общей производительности подобных систем. Для увеличения производительности используются различные методы организации дискового пространства и алгоритмы дискового планирования, позволяющие распараллеливать операции ввода/вывода [2–6].

В [2–6] рассмотрен метод поблочного чередования данных и предложены способы определения оптимального размера блока, построенные на основе моделей в виде замкнутых сетей массового обслуживания с фиксированным уровнем мультипрограммирования. Однако подобный подход оказывается непригодным для современных систем управления данными с большим числом обслуживаемых клиентов и меняющейся во времени нагрузки. В [5] рассматривается аналитическая модель для оценки пропускной способности массивов дисков RAID 1/0. В работе [6] предложен метод определения оптимального размера блока в условиях пуассоновского входящего потока и при предположении равномерного распределения нагрузки по набору дисков. Однако он не позволяет получать значение среднего времени пребывания запросов в подсистеме ввода/вывода при заданном размере блока и, следовательно, не может быть использован для оценки производительности системы и при контроле нагрузки для обеспечения требуемого качества обслуживания клиентов.

В связи с этим в данной работе проводится исследование методов размещения данных в адресном пространстве дисковой подсистемы ввода/вывода серверов хранения и доставки документов по сети. Предлагаются имитационные модели нахождения среднего времени пребывания запроса в системе для метода поблочного чередования и метода экстентного размещения данных. С их помощью проводится сравнение производительности подсистем ввода/вывода, реализующих тот или иной метод, при различных типах входящего потока и в условиях неравномерного распределения нагрузки между дисками. Кроме того, рассматривается влияние размера буфера на производительность системы в зависимости от используемого метода размещения данных.

2. Параметры и характеристики моделирования

При использовании метода поблочного чередования файлы разбиваются на непрерывные блоки фиксированного размера, которые затем распределяются по набору дисков в циклическом порядке. В этом случае функции чередования данных могут возлагаться на менеджера логических томов LVM, объединяющего адресные пространства каждого диска в единый том, воспринимаемый файловой системой как один большой логический диск. Также большинство современных файловых систем имеют собствен-

ные возможности чередования данных на уровне логического блока (кластера) [7]. Подобный подход позволяет балансировать нагрузку между дисками и реализовывать требуемый уровень отказоустойчивости, осуществляя избирательную репликацию только нужных данных (метаданных). Кроме того в этом случае возможна реализация механизмов динамического перераспределения данных между дисками в зависимости от изменений нагрузки [6]. В связи с этим в данной работе рассматривается чередование данных, осуществляемое именно средствами файловой системы на уровне логического блока.

В случае экстентного метода размещения данных каждый запрашиваемый файл образует непрерывный экстент и целиком располагается на одном диске. Однако в данном случае в отличие от [8] запрещается доступ к отдельному блоку внутри экстента: считывание экстента не может быть прервано любой другой операцией ввода/вывода. Такое условие может быть выполнено путем применения «связанных» команд стандарта SCSI или при использовании технологии NASD, предложенной в [9] для реализации в сетевых устройствах хранения, которая обеспечивает высокоуровневый объект – ориентированный интерфейс с файловой системой. Кроме того, это условие автоматически выполняется в случае непрерывного размещения запрашиваемого экстента в адресном пространстве диска и реализации дисциплины обслуживания LOOK.

Значения структурных и функциональных параметров моделирования приведены в таблице 1, а значения нагрузочных параметров – в таблице 2 [10, 11].

Таблица 1

Структурные и функциональные параметры моделирования.

Тип параметров	Значения параметров
Количество дисков IBM UltraStar 36LZX	D = 5
Количество цилиндров	15109
Скорость считывания данных	23173 байт/мс
Время полного оборота диска	6 мс
Зависимость времени позиционирования считывающей головки (мс) от расстояния перемещения в цилиндрах.	$seek(d) = \begin{cases} 1{,}6 + 0{,}036\sqrt{d}; d < 3022 \\ 1{,}3634 + 0{,}737 * 10^{-3} d; d \geq 3022 \end{cases}$ d – расстояние в цилиндрах, пройденное считывающей головкой
Дисциплина обслуживания очереди запросов к диску	FCFS, LOOK

Таблица 2

Нагрузочные параметры моделирования.

Тип входящего потока	Простейший, эрланговский и гиперэкспоненциальный поток с интенсивностью λ и коэфф. вариации ν
Длина запрашиваемого блока данных	Гамма-распределение, ξ = 24 Кб, $x_{1/2}$ = 2Кб

В качестве модельной характеристики в работе рассматривается среднее значение времени пребывания запроса в системе M($T_П$).

3. Описание моделей

Согласно методу поблочного чередования запрос, пришедший в дисковую подсистему, разбивается на подзапросы, размер которых кратен размеру логического блока. Количество подзапросов, соответствующих одному запросу, может варьироваться от 1 до D. При этом каждый подзапрос обслуживается отдельным диском.

Ниже приведена последовательность основных блоков языка GPSS, отражающих основные особенности предлагаемой имитационной модели для метода поблочного чередования данных:

```
            GENERATE   Request      // Генерируем запросы по заданному закону
                                    распределения //
            ASSIGN     Request      // Определяем параметры запроса: размер,
                                    количество подзапросов//
            SPLIT      Subrequest   // Генерируем подзапросы //
            ASSIGN     Subrequest   // Определяем параметры подзапроса: размер,
                                    номер диска, номер цилиндра, время обслу-
                                    живания //
            TRANSFER   DiskX        // Отправляем подзапросы на обслуживание к
                                    соответствующим дискам //
Disk1:                              // Описание модели первого диска //
            QUEUE                   // Ставим подзапрос в очередь на
                                    обслуживание //
            SEIZE                   // Если диск свободен, то выбираем из очереди
                                    следующий подзапрос в соответствии с дис-
                                    циплиной обслуживания//
            ADVANCE                 // Обслуживаем подзапрос //
            RELEASE                 // Освобождаем диск //
            TRANSFER   Out          // Подзапрос обслужен //

// Аналогично первому диску организованы остальные //
Out:        TEST   Subrequest, Notlast  // Проверяем является ли подзапрос
                                        последним подзапросом запроса //
            TABULATE   FullTime     // Если последний, то запрос обслужен //
Notlast:    TERMINATE               // Уничтожаем подзапросы //
```

Для метода экстентного размещения данных структура имитационной модели аналогична случаю метода поблочного чередования данных за тем исключением, что в ней отсутствуют блоки, описывающие процессы разделения запросов на подзапросы либо объединения подзапросов в запросы.

4. Сравнение производительности подсистем ввода/вывода при различных методах размещения данных

С помощью имитационных моделей было проведено сравнение производительности подсистем ввода/вывода при различных потоках запросов и в предположении равномерного распределения нагрузки по набору дисков. При этом для метода поблоч-

ного чередования размер блока полагался равным 8 Кб. Результаты представлены на рис. 1.

Как видно, экстентный метод размещения данных обеспечивает значительно большую производительность подсистемы ввода/вывода в предположении равномерного распределения нагрузки по набору дисков. В то же время указанный метод является более чувствительным к типу входящего потока.

Согласно проведенному анализу основная причина увеличения времени пребывания запросов для метода поблочного чередования заключается в том, что подзапросы обрабатываются каждым диском независимо и назначаются на обслуживание в разные моменты времени. В тоже время для этого случая время пребывания запроса определяется временем пребывания подзапроса, обслуженного последним.

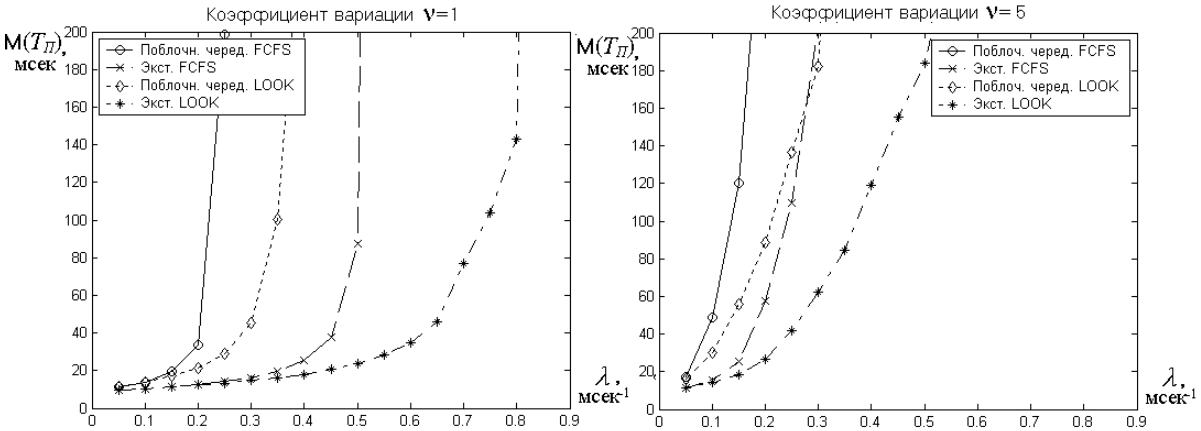

Рис. 1. Зависимости среднего времени пребывания запросов в системе M($T_П$) от интенсивности их поступления λ

Проведено исследование влияния неравномерного распределения нагрузки по набору дисков на производительность дисковых подсистем с различными дисциплинами обслуживания и методами размещения данных. В этом случае полагалось, что на первом диске располагаются самые популярные файлы, на втором – вторые по популярности и т. д. Частота запрашиваемых файлов описывалась случайной величиной, имеющей Ципф-подобное распределение [12]. Т. е. для дисковой подсистемы, содержащей *N* файлов, вероятность поступления запроса к i-му файлу определяется выражением $P_N(i) = \frac{\Omega}{i^{\alpha}}$, где $\Omega = \left(\sum_{i=1}^{N} \frac{1}{i^{\alpha}} \right)^{-1}$ - нормирующая константа. Согласно [12] для Web-серверов коэффициент α = 0,77. Количество файлов, хранящихся в подсистеме ввода/вывода полагалось равным 1500000, что соответствует 36 Гб данных при среднем размере файла 24 Кб. При этом, каждые 10 файлов обладали одинаковой популярностью. Результаты представлены на рис. 2 и в таблице 3.

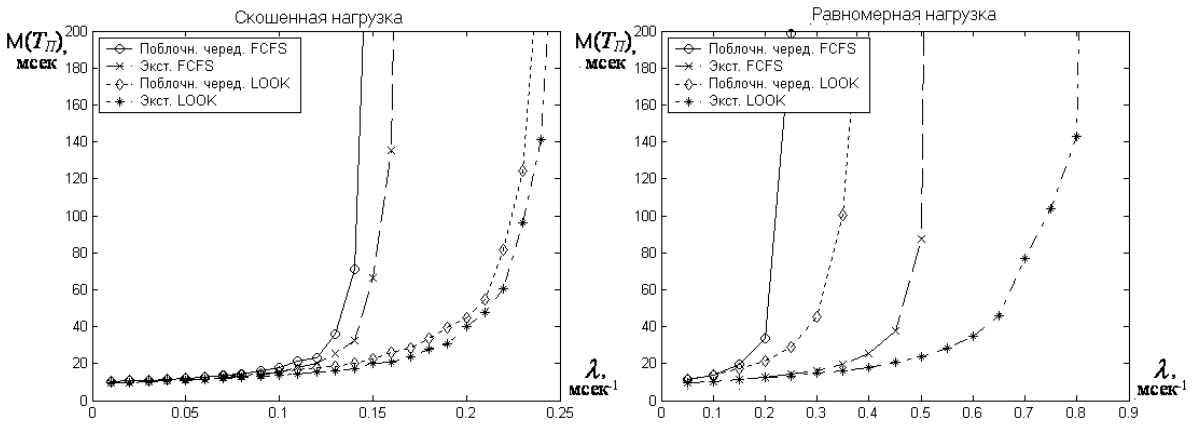

Рис. 2. Зависимости среднего времени пребывания запросов в системе M($T_П$) от интенсивности их поступления λ

Как видно экстентный метод размещения данных является более чувствительным к неравномерности распределения нагрузки по дискам. Так, его использование совместно с дисциплиной FCFS приводит к значительному уменьшению производительности подсистемы ввода/вывода даже по сравнению с методом поблочного чередования данных и дисциплиной обслуживания очереди запросов к диску LOOK.

Таблица 3

Значения предельной интенсивности $\lambda_{ПРЕД}$, сек.$^{-1}$,

при превышении которой значение M($T_П$) становиться большим 1 сек

Размещ./ДО / Нагрузка	Побл.черед./FCFS	Экст./FCFS	Побл.черед./LOOK	Экст./LOOK
Скошенная	157	169	258	271
Равномерная	264	541	429	853

В связи с этим в работе также было проведено исследование влияния размера буфера на предельную интенсивность $\lambda_{ПРЕД}$ для различных методов размещения данных. При этом рассматривался алгоритм замещения страниц Perfect-LFU. Т. е. предполагалось, что в буфере содержатся самые популярные файлы, а размер буфера определялся количеством хранимых файлов. Оставшиеся файлы распределялись между дисками аналогично предыдущему случаю. Результаты представлены на рис. 3.

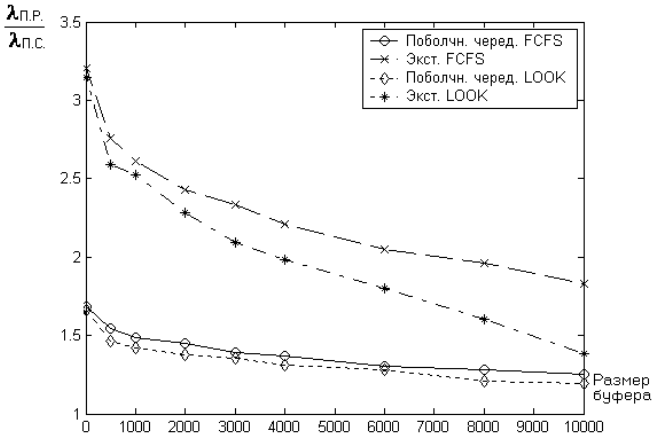

Рис. 3. Зависимости отношения предельной интенсивности системы с равномерно распределенной нагрузкой $\lambda_{П.Р.}$ к предельной интенсивности системы со скошенной нагрузкой $\lambda_{П.С.}$ от размера буфера

Как видно, системы с методом поблочного чередования менее чувствительны к размеру буфера. Для случая экстентного метода размещения данных и дисциплины LOOK наблюдается значительный рост производительности системы при увеличении размера буфера.

5. Выводы

В работе рассмотрены методы поблочного чередования и экстентного размещения данных в адресном пространстве дисковой подсистемы ввода/вывода серверов хранения и доставки документов по сети. Для данных методов построены имитационные модели, позволяющие оценивать среднее время пребывания запроса в дисковой подсистеме при реализации таких дисциплин обслуживания очереди запросов к диску, как FCFS и LOOK.

С помощью предлагаемых моделей было показано, что при условии равномерного распределения нагрузки между дисками экстентный метод размещения данных обеспечивает более чем двукратное увеличение производительности по сравнению с методом поблочного чередования вне зависимости от используемой дисциплины обслуживания. В то же время, указанный метод является более чувствительным к типу входящего потока и неравномерному распределению нагрузки по набору дисков. В связи с этим в работе также было проведено исследование влияния размера буфера на производительность системы. Показано, что для дисковых подсистем с экстентным методом размещения данных применение буфера дает больший прирост производительности, чем для систем с методом поблочного чередования данных.

Требование размещения файла в последовательно распложенных блоках одного диска приводит к необходимости эффективного поиска экстентов заданного размера и, как следствие, к усложнению архитектуры подобных файловых систем. Таким образом, в отличие от метода поблочного чередования данных, характерного для файловых систем общего назначения и предназначенных для широкого спектра задач, метод экстентного размещения предпочтителен для систем, ориентированных в основном на чтение данных, в которых вопросы управления дисковым пространством не стоят так остро. Однако более высокая производительность экстентного метода размещения данных совместно с дисциплиной обслуживания очереди запросов к диску LOOK позволяет автору сделать вывод о перспективности применения данного подхода в серверах хранения и доставки документов по сети.

Литература

1. **Н. Сергеева, Л. Павлов**. Корпоративные цифровые библиотеки. Открытые системы, № 3, 1997.
2. **P.M. Chen, E.K. Lee.** Striping in a RAID Level 5 Array, ACM SIGMETRICS Conference, pages 136–145, 1995.
3. **P.M. Chen, D.A. Patterson.** Maximizing Performance in a Striped Disk Array, Proceedings of the 17th International Symposium on Computer Architecture (SIGARCH), pages 322–331, 1990.
4. **E.K. Lee, R.H. Katz.** An Analytic Performance Model of Disk Arrays. Proceedings of the International Conference on Measurement and Modeling of Computer Systems (ACM SIGMETRICS), pages 98–109, 1993.
5. **M. Uysal, G.A. Alvarez, A. Merchant.** A Modular Analytical Throughput Model for Modern Disk Arrays, Ninth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS-2001), August 15–18, 2001.
6. **P. Scheuermann, G. Weikum, P. Zabback.** Data Partitioning and Load Balancing in Parallel Disk Systems, The VLDB Journal, Vol. 7, pages 48–66, February 1998.
7. **F. Schmuck, R. Haskin.** GPFS: A Shared-Disk File System for Large Computer Clusters, Proceedings of the Conference on File and Storage Technologies (FAST'02), pages 231–244, 2002.
8. **L. McVoy, S. Klieman.** Extent-like Performance from a UNIX File System. Proceedings of Summer USENIX Conference, Annaheim, CA, pages 137–144, June 1990.
9. **G.A. Gibson et al.** NASD Scalable Storage Systems, Proceedings of USENIX 1999, Linux Workshop, Monterey, CA, June 9–11, 1999.
10. Hard disk drive specification. Ultrastar 36LZX, Models: DDYS-T36950, DDYS-T18350, DDYS-T09170. Revision 2.1. IBM Corp. 9 June 2000.
11. **S. Saroiu et al.** An Analysis of Internet Content Delivery Systems. Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI 2002), December 2002.
12. **L. Breslau et al.** Web Caching and Zipf-like Distribution: Evidence and Implications. Proceedings of IEEE Infocom '99, pages 126–134, New York, NY, March, 1999.