Применение группового наложенного управления ресурсами операционной системы на примере потоков ввода-вывода

А. Г. Тормасов, А. Л. Кобец, С. А. Опекунов, В. В. Луковников
Московский физико-технический институт, Россия
e-mail: tor@sw.ru, kobets@sw.ru, osergey@sw.ru, vlukovnikov@sw.ru

This article addresses a mathematical model for the imposed group management of input/output data flows in modern operating systems. An experimental demonstration of the model effectiveness is provided for the case of virtual servers.

Введение

В современной IT-индустрии намечается тенденция роста интереса к системам виртуализации. Например, основные игроки на рынке производителей процессоров, Intel и AMD, внедряют технологии аппаратной виртуализации Virtualization Technology и Pacifica соответственно. Производители операционных систем, такие как Microsoft, Apple и Linux, включают в свое программное обеспечение поддержку виртуализации. Виртуализация порождает проблему эффективного распределения и управления ресурсами.

1. Модель группового наложенного управления

Рассмотрим проблему планирования и распределения ресурсов между задачами в операционной системе (ОС) применительно к группам задач, оперирующим дисковым вводом-выводом.

Сформулируем модель наложенного управления и исследуем ограничения, которые требуется наложить на управление уровня ОС, для обеспечения требуемого качества обслуживания (QoS).

Для любой i-й задачи в системе введем три вида функций потребления ресурсов ввода-вывода — желаемого, фактического и идеального потребления — как функции собственного (t^*), системного (t) и идеального (t^{**}) времени соответственно:

$$R_i(t^*),\ R_i^*(t),\ R_i^{**}(t^{**}). \tag{1}$$

Собственное время идет только, если задача потребляет ресурс, а если ресурс не потребляется, то время "замораживается". Системное время — время, в котором задача получала ресурс. Идеальное время — это время, исполняясь в котором достигается

требуемое качество обслуживания, т. е. наложенное планирование работает так, как требуется.

Любая функция потребления по определению может принимать только три значения: 1 — когда ресурс потребляется, 0 — когда задача простаивает, -1 — когда ресурс освобождается, т. е. справедливо

$$R_i(t^*) \in \{0, 1, -1\}. \tag{2}$$

Введем преобразования времен:

$$t = F_i(t^*); \tag{3}$$

$$t^{**} = S_i(t^*). \tag{4}$$

Критерий качества [1] можно сформулировать следующим образом: при каких предположениях о поведении $R_i(t^*)$ выполняется условие

$$\int_{T_1}^{T_2} \|R_i(t^*) - R_i^{**}(t^{**})\| dt < a(T_2 - T_1), \tag{5}$$

где T_1, T_2 — временные отрезки наложенного управления; a — допустимая погрешность с соответствующей размерностью.

Полоса пропускания диска является возобновляемым ресурсом [1], поэтому далее мы рассматриваем этот тип ресурсов.

Перейдем в собственное время задачи. Пусть $DR_i(F_i(t^*))$ — функция, описывающая неконтролируемые эффекты операционной системы. Функция желаемого потребления будет выглядеть как

$$R_i^*(F_i(t^*)) = \tilde{R}_i(F_i(t^*)) + DR_i(F_i(t^*)). \tag{6}$$

Здесь $\tilde{R}_i(F_i(t^*))$ — некоторая "реальная" функция потребления без учета неконтролируемых эффектов операционной системы. В случае идеального наложенного планирования, когда гарантия совпадает с лимитом, справедливо выражение

$$\int_{T_1}^{T_2} (R_i^{**}(S(t^*)) \frac{dF_i(t^*)}{dt^*} dt^* = b(T_2 - T_1) = b\Delta T. \tag{7}$$

Здесь b — доля возобновляемого ресурса, выделенная задаче, которая является постоянной на данном временном интервале.

В собственном времени задачи имеем

$$\int_{T_1}^{T_2} \tilde{R}_i(F_i(t^*)) \frac{dF_i(t^*)}{dt^*} dt^* > \int_{T_1}^{T_2} (DR_i(F_i(t^*)) \frac{dF_i(t^*)}{dt^*} dt^* + b\Delta T - a\Delta T. \tag{8}$$

Мы получили выражение, удовлетворяя которому на рассматриваемом отрезке времени, можно получить требуемое качество обслуживания наложенного управления для возобновляемых ресурсов, в частности для дисковой пропускной способности.

Перейдем к описанию модели группового наложенного управления [2]. Под группой задач будем считать конечное число задач в операционной системе, которые удовлетворяют следующим критериям:

— все задачи в группе на рассматриваемом интервале времени потребляют один и тот же возобновляемый ресурс;

— задачи можно объединить по какому-либо признаку.

Будем считать, что в любой момент времени ресурс может потреблять только одна из задач группы. Таким образом, для возобновляемых ресурсов

$$GR(t_{gr}) \subset \{0, 1\}, \tag{9}$$

где t_{gr} — собственное время группы задач.

Для групп задач справедливы аналогичные (6) и (7) равенства.

Перейдем в собственное время группы, в результате мы имеем следующий критерий качества для групп [3]:

$$\int_{T_1}^{T_2} \tilde{GR}(F^{gr}(t^{gr}))\frac{dF^{gr}(t^{gr})}{dt^{gr}}dt^{gr} > \int_{T_1}^{T_2} DG(F^{gr}(t^*))\frac{dF^{gr}(t^*)}{dt^{gr}}dt^{gr} + B\Delta T - A\Delta T. \tag{10}$$

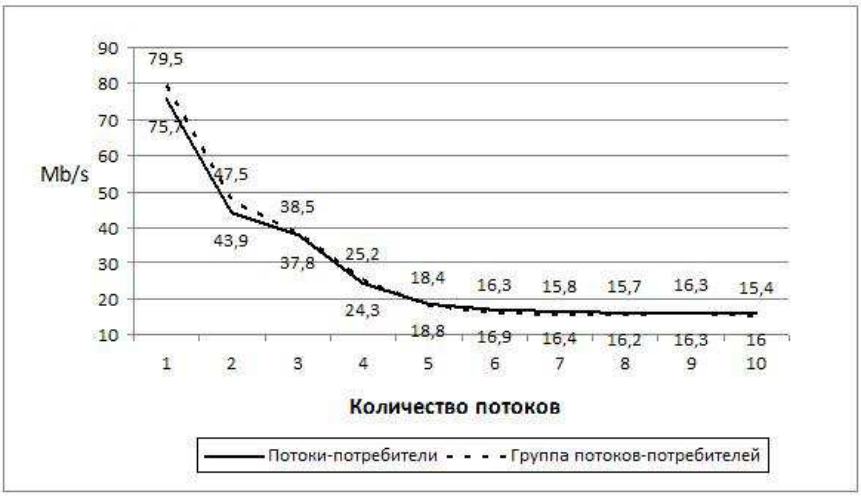
Накладывая ограничения на функцию преобразования времени $F^{gr}(t_{gr})$, путем изменения скорости потребления (уменьшая значение производных в формуле (10), замедляя собственное время группы, ограничивая доступ к диску для всех потоков группы) и анализируя поведение системы в режиме реального времени, мы можем достигать приемлемой точности выделения пропускной способности диска.

2. Практическая применимость модели

Рассмотрим распределение пропускной способности диска между группами потоков нескольких виртуальных серверов на примере системы Virtuozzo Linux (ядро 2.6.9-023stab033.7-smp, 4 Гб RAM, 170 Гб SCSI диск, 2 CPU). Запускается один виртуальный сервер, внутри него порождается дисковая активность чтения 64 Кб блоков (стандартная единица обмена кеша и диска) из трех файлов, суммарного размера 3 Гб. Относительный максимум полосы пропускания находится в зависимости от числа потоков, читающих диск. Перед каждым замером производится очищение дискового кеша. Затем рассматривается зависимость полосы пропускания записи от числа потоков-потребителей (см. рисунок, *а*). Для того чтобы принудить систему сбрасывать данные на диск, а не помещать их в кеш, мы изначально заполняем кеш, прогоняя дополнительные итерации записи (см. рисунок, *б*).

В обоих случаях мы пришли к приблизительно одному и тому же значению пропускной способности, что объясняется тем, что распределение запросов на уровне диска происходит одним из алгоритмов аппаратного планирования, который работает как одна очередь. Падение полосы пропускания с возрастанием числа потоков объясняется эффектом рандомизации запросов и фрагментацией диска. Большие величины при малом числе потоков записи обусловлены работой дискового кеша. Далее рассматриваются группы потоков с просыпающимися и засыпающими потоками. В основе эксперимента лежала следующая программа. Создается N процессов, внутри каждого процесса создается несколько потоков, которые по очереди читают 640 Кб (10 итераций по 64 Кб)

а



б

Зависимость полосы пропускания от числа потоков чтения (*а*) и записи (*б*) в группе

и засыпают. Эти потоки объединяются в группу. Этот эксперимент показывает работу группового времени, поскольку время группы в целом идет, а время отдельного потока — нет. Рассмотрим пропускную способность чтения или записи в зависимости от N на таких группах (см. рисунок). Большая величина значения пропускной способности при малом числе потоков записи обусловлена работой дискового кеша.

Далее в двух виртуальных серверах запускается по группе из 10 потоков, читающих и пишущих на диск. В каждой группе пять потоков пишут, а пять потоков читают, происходит чередование потоков, как в предыдущем примере. Как следствие, при паритетном запуске этих двух групп мы получаем пропускную способность суммы в 16 Мб/с, стандартный планировщик системы должен выдать им равное количество пропускной способности диска. Мы хотим достигнуть соотношения 11 : 5 для отношения потребления этими группами. Будем тормозить вторую группу таким образом, чтобы ее потребление было 5 Мб/с. Сделаем оценку задержки, которую нужно осуществлять при чтении или записи потоками второй группы. Если времена относятся обратно пропорционально полосе потребления группы, то новое собственное время должно быть

в 8/5 раз медленнее первоначального. Теперь, поскольку группа поедает 8 Мб/с, а за одно обращение мы пишем или считываем 64 Кб, группа должна 128 раз обратиться к диску, т. е. наша задержка при обращении должна быть

$$\frac{8/5 - 1}{128} \approx 0.0047 \text{ с.} \tag{11}$$

На практике для достижения требуемого соотношения потребовалась бóльшая задержка, поскольку при добавлении задержек уменьшается и число обращений, т. е. общая пропускная полоса уменьшается. Практическая величина задержки 0.0053 с. Таким образом, функция преобразования времени второй группы в данной задаче выглядит следующим образом:

$$F^{gr}(t^{gr}) = 1.6784\, t. \tag{12}$$

Мы показали на практике целесообразность использования построенной модели для обеспечения требуемого качества управления пропускной способностью дискового ввода-вывода.

3. Классификация потребителей

Рассмотрим варианты использования нашей модели с потребителями полосы пропускания диска. Основные потребители приведены ниже:

1) операция отложенной записи через кеш (Lazy write);

2) операция чтения через кеш;

3) операция “принудительного” обращения к диску. Данная операция характерна для ОС, в которых есть файл подкачки(Swap file для ОС Linux, Pagefile для ОС Windows);

4) операции обращения к Windows Registry;

5) операции чтения через сеть. В этом случае мы сталкиваемся с взаимодействием подсистемы ввода/вывода с сетевой подсистемой ОС;

6) журналируемые файловые системы;

7) низкоуровневое обращение к диску;

8) фильтры файловой системы. В данном случае речь идет о встраивании другого программного обеспечения в процесс обращения к диску;

9) принудительный сброс данных из кеша на диск.

Исходя из вариантов, предложенных выше, можно привести следующую классификацию запросов ввода-вывода, которые нужно обрабатывать специальным образом в рамках нашей модели (в скобках указаны соответствующие варианты):

— отложенная запись через кеш (1) [4];

— кешированное чтение (2) [4];

— принудительное чтение или запись (3, 4, 6–9) [4];

— внешнее ограничение на ввод/вывод (5, 8) [4].

4. Способы управления ресурсами для разных типов потребителей

Рассмотрим способы планирования ресурсами ввода/вывода для потребителей разного вида, приведем основные направления развития предложенной модели группового наложенного управления.

Наша модель позволяет вводить дополнительные группы потоков-потребителей, для того чтобы обеспечить более честное планирование. Предлагается ввести дополнительные группы:

— отложенная запись через кеш;

— кешированное чтение;

— принудительное чтение или запись;

— внешнее ограничение на ввод/вывод.

Мы хотим повысить справедливость модели в случае, когда несколько потребителей обращается к закешированным данным. Основная проблема работы с кешем в модели: какому именно потребителю приписывать использование ресурса? Один из вариантов ее решения — перейти к рассмотрению ресурса дискового кеша и выделить группы из потоков-потребителей полосы пропускания диска тех, кто обращается к одним и тем же данным в кеше.

Решение проблемы, описанной выше, упрощается при максимально возможном разделении потребителей по группам.

Одним из результатов исследования является тот факт, что при разделении потребителей по типу запроса мы приходим к тому, что полоса пропускания диска является общей для обоих типов и максимальная полоса пропускания — величина не постоянная.

Одним из способов оптимизации модели может быть построение комбинированного планировщика второго уровня, который управляет долями пропускных способностей для каждой из групп потребителей.

Заключение

Модели и методы группового наложенного управления ресурсами в виде потоков ввода-вывода позволяют существенно упростить разработку систем управления полосой пропускания диска в условиях работы систем виртуализации операционных систем или отдельно взятых приложений. Разработанная модель может быть распространена на другие возобновляемые ресурсы или использована в виде подсистемы при разработке систем резервного сохранения копий данных, миграции виртуальных серверов.

Список литературы

[1] Кобец А.Л, Луковников В.В., Пименов В.М., Соколов Е.В Оценка точности группового наложенного управления ресурсами операционной системы для дискового ввода/вывода // Вест. НГУ. Сер. Информационные технологи. 2007. Т. 5, вып. 1. С. 28–31.

[2] Луковников И.В. Математическая модель двухуровневого управления ресурсами в операционных системах с закрытыми кодами: дис. ... канд. физ.-мат. наук. М., 2006.

[3] Тормасов А.Г., Кобец А.Л., Луковников В.В. Модель управления группами потоков ввода/вывода с заданной точностью // Моделирование процессов обработки информации: Сб. науч. тр. / М.: Моск. физ.-тех. ин-т, 2007. С. 272–275.

[4] Rajeev N. File System Internals: A Developer's Guide, O'REILLY, 1997.

[5] Solomon D.A., Russinovich M.E. Inside Microsoft Windows 2000, Third Edition, Microsoft Press.

Поступила в редакцию 28 марта 2008 г.