На правах рукописи

КОБЕЦ Алексей Леонидович Кобец

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ НАЛОЖЕННОГО УПРАВЛЕНИЯ РЕСУРСАМИ ВИДА "ПОТОКИ ВВОДА-ВЫВОДА" В ОПЕРАЦИОННЫХ СИСТЕМАХ

Специальность 05.13.18 – математическое моделирование, численные методы и комплексы программ

Автореферат диссертации на соискание
ученой степени кандидата физико-математических наук

Москва - 2007

Работа выполнена на кафедре информатики Московского физико-технического института (государственного университета)

Научный руководитель: кандидат физико-математических наук
Тормасов Александр Геннадиевич

Официальные оппоненты: доктор технических наук
профессор
Семенихин Сергей Владимирович

кандидат технических наук
старший научный сотрудник
Шилов Валерий Владимирович

Ведущая организация: Институт Автоматизации Проектирования РАН

Защита состоится «9» ноября 2007 года в 12[30] часов на заседании диссертационного совета К 212.156.02 при Московском физико-техническом институте (Государственном университете) по адресу: 141700, г.Долгопрудный, Московской обл., Институтский пер. д.9., ауд. 903 КПМ.

С диссертацией можно ознакомиться в библиотеке МФТИ

Автореферат разослан «8» октября 2007 г.

Ученый секретарь диссертационного совета
К 212.156.02 Федько О.С.

Общая характеристика работы

Актуальность темы

Производительность современных операционных систем существенно зависит от производительности операций ввода-вывода, но производительность дискового оборудования растёт меньшими темпами. Например, производительность RISC-процессора увеличивается приблизительно на 50% в год, а скорость доступа к диску лишь на 10% [данные с сайта http://www.sisoftware.net]. На уровне программного обеспечения основными факторами, обостряющими данную проблему, являются:

- Увеличивающаяся «плотность» программного обеспечения на физическом компьютере, что является следствием стремительного распространения средств виртуализации программного обеспечения и операционных систем;
- Развитие средств мультимедиа, которое порождает необходимость хранения больших объёмов данных на диске и более надёжного и быстрого доступа к этим данным;
- Распространение услуг предоставления хостинга, т.е. размещения пользовательских данных в интернете и большое количество потребителей, обычно расположенных на одном компьютере.

Намечается тенденция роста интереса к системам виртуализации в IT индустрии. 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 своё программное обеспечение поддержку виртуализации. Существует ряд компаний или открытых проектов, которые разрабатывают исключительно средства виртуализации операционных систем (VMware, Xen, SWsoft). Существуют также средства дисковой виртуализации, например, Loopback Device в семействе операционных систем Unix или Virtual Disk в Microsoft Windows или VMware Workstation.

Данная тенденция позволяет утверждать, что со временем приложения будут выставлять всё более жёсткие требования к работе с диском, поэтому требуется корректное планирование дисковой пропускной способности – иначе диск может стать узким местом в производительности всей системы.

В диссертационной работе рассматривается математическая задача планирования и распределения ресурсов между задачами в операционной системе, применительно к группам задач, оперирующим дисковым вводом-выводом. Задача поставлена с учётом следующих требований:

- Пропускная способность распределяется согласно соглашению об уровне сервиса (Service Level Agreement);

- Возможность построения механизмов планирования ресурса пропускной способности диска «над» аналогичными механизмами планирования операционной системы;
- Ограничения, наложенные на механизмы управления дисковой активностью ОС, не должны приводить к существенному ухудшению производительности операционной системы в целом.

Отметим, что задачи такого уровня актуальны для средств встроенной виртуализации ОС.

Цель работы, задачи исследования

Цель диссертационной работы – разработка математической модели и методов наложенного управления ресурсами вида потоки дискового ввода-вывода в современных ОС. Будем называть наложенным управлением управление, которое удовлетворяет следующим требованиям:

- Управление осуществляется на основе стандартных механизмов управления ресурсами ОС.
- Интервалы, на которых осуществляется такое управление, превышают интервалы, на которых работают механизмы управления самой ОС.

Задачи исследования:

- Разработка математической модели группового наложенного управления ресурсами вида потоки ввода-вывода в современных ОС.
- Исследование ограничений, которые должны выполняться в ОС, для обеспечения требуемого качества обслуживания.
- Исследование и объяснение, в рамках данной модели, существующих механизмов контроля ресурсов дискового ввода-вывода в современных ОС.

Объект исследования – алгоритмы планирования подсистемы ввода-вывода в ОС.

Предмет исследования – механизмы управления ресурсами ввода-вывода в современных ОС.

В рамках выполнения данного исследования автором был проведён ряд экспериментов, подтверждающих эффективную работоспособность данной модели.

Методы исследования

В процессе научных исследований по разработке математической модели наложенного управления ресурсами дискового ввода-вывода в современных системах использовались аналитические методы теории цифровой фильтрации, методы теории операционных систем, системного программирования, а так же методы, используемые в современных системах виртуализации.

Предложенная модель была реализована как часть программного комплекса Virtuozzo. Был проведён ряд экспериментов с использованием данного комплекса.

Научная новизна

Научная новизна работы заключается в том, что автором предложена математическая модель группового наложенного управления ресурсами вида потоки ввода-вывода при предоставлении различного вида сервиса в современных ОС, а так же системах, основанных на виртуализации среды конечного пользователя.

В отличие от ранее существовавших моделей виртуализации разработанная в ходе данного диссертационного исследования математическая модель более полно учитывает особенности, налагаемые собственными механизмами управления ОС. Это позволяет существенно повысить утилизацию ресурсов системы и повысить её надёжность, а также достигнуть требуемого качества обслуживания. Вводиться понятие группы потребителей, собственное время группы, а также функция потребления группы потребителей.

Разработанная математическая модель группового наложенного управления ресурсами вида потоки ввода-вывода современных ОС является новым вкладом в развитие теории ОС, системного программирования и технологий виртуализации операционных систем.

Практическая значимость

Разработанная математическая модель может быть использована при создании новых комплексов программ, предназначенных для решения задач виртуализации с целью достижения максимальной утилизации ресурсов ввода-вывода, повышения надёжности и безопасности системы, а также предоставления высокого качества обслуживания.

Также разработанная модель и методы могут быть использованы в качестве самостоятельных решений различных задач, возникающих при представлении различного рода сервиса в современных операционных системах.

Например, математическая модель группового наложенного управления ресурсами дискового ввода-вывода позволяет решить ряд

технических проблем, связанных с обеспечением заданного распределения ресурсов дискового ввода-вывода, которое имеет большое практическое значение для бесперебойной работы консолидированных файлообменных сервисов и сервисов баз данных, используемых государственными, коммерческими, а также образовательными организациями.

Апробация и реализация результатов работы

По выполненным диссертационным исследованиям опубликовано 10 работ, в том числе три [1,4,5] – в ведущих научных журналах, рекомендованных ВАК РФ. В опубликованных работах автору принадлежит более 40% материала, связанного с изложением основ математической модели наложенного управления ресурсами (включая группы потоков ввода-вывода) с использованием виртуальных сред.

Результаты диссертационного исследования докладывались, обсуждались и получили одобрение специалистов на научных конференциях и семинарах: XI Всероссийской научно-практической конференции «Научное творчество молодёжи», посвящённой 50-летию СО РАН 2007 г. Анжеро-Судженск, Всероссийской научно-практической конференции, посвященной 60-летию ТОИПКРО, 2006 г. Томск, Московской международной конференции по вычислительной молекулярной биологии(МССМВ'07), 2007 г., семинары кафедры информатики МФТИ (2001-2007гг), семинар в ВЦ РАН под руководством члена-корреспондента РАН Флёрова Ю. А. (2006г).

Также результаты докладывались и получили одобрение специалистов на научных семинарах, проводимых кафедрой информатики МФТИ.

Результаты работы реализованы при создании программного комплекса Virtuozzo, разработанного в компании SWsoft. В настоящее время этот программный комплекс занимает лидирующее положение на рынке предоставления услуг виртуализации операционных систем.

Положения, выносимые на защиту

На защиту выносятся следующие основные положения:

1) Математическая модель наложенного управления ресурсами вида потоки ввода-вывода в операционных системах.
2) Математическая модель и метод группового наложенного управления ресурсами потоков ввода-вывода в условиях функционирования виртуальных выделенных серверов с целью ограничения потребления пропускной способности дискового ввода-вывода.

Структура и объем диссертации

Диссертация состоит из введения, трех глав, заключения, списка используемых источников и трех приложений. Работа изложена на 111 страницах, список используемых источников содержит 107 наименований в алфавитном порядке.

Содержание работы

Во **введении** обосновывается актуальность темы, дается исторический обзор исследований, посвященных решаемым в диссертации задачам, формулируются цели исследования и основные положения, которые выносятся на защиту, обосновывается научная и практическая значимость выполненного исследования.

В **главе 1** содержится обзор существующих алгоритмов реализации управлением дискового ввода-вывода. Алгоритмы разбиты на две группы:

- *Аппаратные*, которые реализованы в самих устройствах дисков, о которых операционной системе зачастую не известно;
- *Системные*, которые являются частью операционной системы или построены «над» ней.

В качестве *аппаратных* алгоритмов рассмотрены FCFS, SSTF, Elevator, EDF, LSF, P-SCAN, FD-EDF, SCAN-EDF, SSDEO, SSDEV и др. Из *системных* алгоритмов рассмотрены Rate-Monotonic, WFQ, FairSched, P-SFQ, пропорциональное планирование на базе вектора ошибки, WRR, Lottery Scheduling и др. Для всех рассмотренных алгоритмов показаны преимущества и недостатки. Рассмотрено сравнение алгоритмов планирования на основе приоритетов и пропорциональности, приведена таблица. Далее рассмотрено комбинированное планирование, как отдельный класс планировщиков. Приведены преимущества использования комбинированных планировщиков, а также способы их реализации. Отдельно рассмотрен фильтр Калмана, как инструмент для достижения заданной точности через определённый круг итераций, показана его математическая реализация, приведены примеры продуктов, в которых он используется.

Глава 2 посвящается постановке задачи путём составления модели наложенного управления и исследования ограничений, которые требуется наложить, для обеспечения требуемым качеством обслуживания(QoS). Вводиться понятие наложенного управления. Рассматриваются особенности построения модели, её ограничения и способы реализации. В дальнейшем речь идёт в основном о ресурсах дискового ввода-вывода.

Далее, для любой *i-ой* задачи в системе, вводиться три вида функций потребления ресурсов ввода-вывода - желаемого, фактического и идеального

потребления, как функции собственного(t^*), системного(t) и идеального(t^{**}) времени соответственно:

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

Собственное время – это время, которое идёт только, если задача потребляет ресурс, 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 3 значения, это 1 – когда ресурс потребляется, 0 – когда задача простаивает, -1 – когда ресурс освобождается, т.е. справедливо:

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

Далее, для случаев потребления нескольких(K) ресурсов, по аналогии с функциями потребления, вводятся векторы желаемого, фактического и идеального потребления. Вектор желаемого потребления:

$$R_i(t^*) = (R_i^1(t^*), R_i^2(t^*), ..., R_i^K(t^*)) \quad (3)$$

Рассматриваются основные свойства вектора потребления:

- Координаты вектора потребления могут иметь ненулевую ковариацию (зависеть друг от друга). Например, копирование файла через сеть приводит не только к использованию дисковой пропускной способности, но и сетевой.
- Координаты так же удовлетворяют (2)
- Ковариационная матрица векторов потребления разных процессов может быть ненулевой. Например, увеличение потребления CPU одним процессом может привести к уменьшению потребления другим.
- Поскольку все ресурсы операционной системы связаны, иногда сложно определить – какой параметр изменился, например, при возникновении ошибки страницы (Page Fault) [cc], можно считать, что либо процессорное время используется, либо пропускная способность жёсткого диска. В таких ситуациях решение принимается согласно принятой модели планирования.

Вектор фактического и идеального потребления строиться таким же образом из соответствующих функций потребления и обладает такими же свойствами.

Далее утверждается, что справедливо неравенство

$$R_i^*(t) \neq R_i(t) \quad (4)$$

обусловленное следующими факторами:

a) Внешним воздействием. Работа процесса с разными компонентами системы может осуществляться с разной скоростью, например, доступ к оперативной памяти происходит гораздо быстрее, чем доступ к диску, поэтому, если задача работает с памятью, которая была выгружена в файл подкачки, «торможение в реальном времени» неизбежно.

b) Эффект CPU boosts, приоритет процесса при завершении операции ввода-вывода будет поднят стандартным планировщиком Windows 2003 Server

c) Swapping – операции работы с файлом подкачки памяти.

d) Влияние аппаратного уровня. Например, многие современные портативные компьютеры могут менять тактовую частоту процессора, что в свою очередь меняет и «скорость» работы задачи. Данный эффект наблюдается при работе виртуальных серверов VMware на сервере с динамическим изменением частоты [cc], например, время внутри виртуального сервера бежит то быстрее то медленнее.

e) Starvation. Эффект, при котором, процесс получает ресурс после очень долгого его ожидания.

Далее отмечается, что требуемое качество управления на заданном промежутке времени может быть достигнуто при реальных предположениях о поведении функций потребления. Вводиться преобразование времени:

$$t = F_i(t^*) \tag{5}$$

Это преобразование «растягивает» временную ось собственного времени процесса, t^* (такое «растяжение» происходит, поскольку, например, в системе существуют другие задачи и рассматриваемой задаче ресурс может быть не предоставлен в «желаемый» им момент времени).

Вводиться функция преобразования идеального времени в собственное:

$$t^{**} = S_i(t^*) \tag{6}$$

Далее приводиться иллюстрация модели на примере потребления дисковой пропускной способности. На Рис. 1 показана функция желаемого потребления:

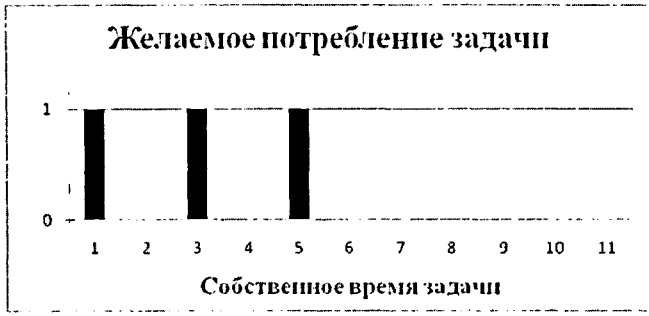

Рис 1. Желаемого потребление

Далее приводится пример функции фактического потребления

Рис 2. Фактическое потребление

Иллюстрация идеального потребления показана на Рис. 3.

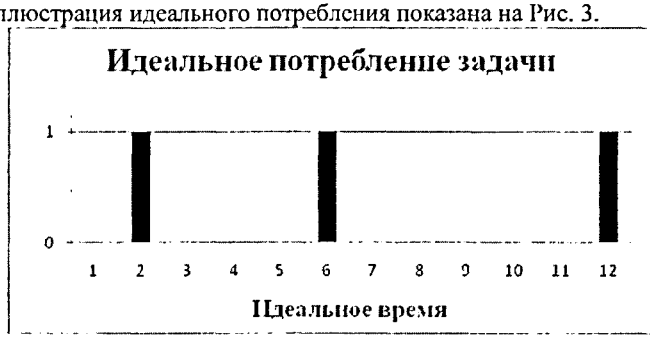

Рис 3. Идеальное потребление

Из иллюстраций видно, что в собственном времени задача потребляла пропускную способность диска в момент времени 1, а фактически она потребляла в момент времени 2.

Критерием качества считается интегральное отклонение функции идеального потребления от функции фактического потребления. Математически критерий качества можно сформулировать следующим образом: при каких предположениях о поведении $R_i(t^*)$ выполняется условие

$$\int_{T1}^{T2} \| R_i^*(t) - R_i^{**}(t^{**}(t)) \| dt < a(T_2 - T_1) \qquad (7)$$

T1 – начало временного отрезка наложенного управления

T2 – конец временного отрезка наложенного управления

a – допустимая погрешность с соответствующей размерностью,

Под $t^{**}(t)$ подразумевается зависимость идеального времени от фактического времени.

Подынтегральное выражение это не что иное как мера вектора, для удобства мерой вектора вводиться сумма квадратов его координат. Учитывая это и то, что для функций потребления справедливо утверждение:

$$R^2 = |R| \qquad (8)$$

Перейдем к собственному времени задачи и перепишем неравенство (7) в виде:

$$\int_{T1}^{T2} \{|R_i^*(F_i(t^*))| + |R_i^{**}(S_i(t^*))| - 2R_i^{**}(S_i(t^*))R_i^*(F_i(t^*))\} \frac{dF_i}{dt^*} dt^* < a(T_2 - T_1) \qquad (9)$$

Будем классифицировать ресурсы, как предложено в работе Луковникова И.В. «Математическая модель двухуровневого управления ресурсами в операционных системах с закрытыми кодами»:

- *Возобновляемые* – это ресурсы, для которых операционная система может безболезненно запретить потребление для определённой задачи и передать освобожденный ресурс нуждающимся задачам. Примером таких ресурсов может послужить пропускная способность диска (если мы не даем процессу единицу пропускной способности, то она может быть с успехом использована другими задачами в системе), пропускная способность сети, CPU и так далее. Отметим так же, что для возобновляемых ресурсов перераспределение, как правило, не приводит к существенному изменению поведения задачи или потерям данных. Очевидно, что для возобновляемых ресурсов функция желаемого потребления не может принимать отрицательного значения. Использование, например, памяти не удовлетворяет данному критерию, поскольку, операционная система

может освобождать память с намного позднее запрещения её потребления задаче.

- *Невозобновляемые* – это ресурсы, для которых можно запретить потребление задачей ресурса, но нет возможности освободить уже используемые ресурсы. Например, дисковое пространство – операционная система может запретить дальнейшее потребление (например, через механизм квотирования дискового пространства на основе различных критериев – это реализовано почти во всех современных операционных системах), но освобождение занятых ресурсов возможно, только если сама задача или операционная система примет решение о необходимости освобождения части данных. В этом случае возможна потеря данных или замедление работы системы.

- *Частично возобновляемые* – это ресурсы, для которых можно запретить потребление задачей данного ресурса и через некоторое время освободить ресурсы для использования другими процессами. Например, физическая память задачи. Обычно за освобождение и выделение таких ресурсов отвечает некая подсистема (в случае памяти – подсистема управления памятью ОС). Можно запретить процессу потреблять память, и, при необходимости, подсистема управления памятью ОС освободит физическую память для других процессов за счет выпихивания памяти данного процесса в файл подкачки.

Далее переходим к рассмотрению возобновляемых ресурсов, т.к. дисковая пропускная способность является возобновляемой. Рассматриваются существующие способы обеспечения качества такому типу ресурсов.

Далее делается допущение. Пусть $DR_i(F_i(t^*))$ – функция, описывающая неконтролируемые эффекты операционной системы. Тогда функция желаемого потребления будет выглядеть

$$R_i^*(F_i(t^*)) = \hat{R}_i^q(F_i(t^*)) + DR_i(F_i(t^*)) \qquad (10)$$

$\hat{R}_i^q(F_i(t^*))$ - это некоторая «реальная» функция потребления, без учётов неконтролируемых эффектов операционной системы. В случае идеального наложенного планирования, когда гарантия совпадает с лимитом справедливо выражение

$$\int_{T1}^{T2} (R_i^{**}(S(t^*)) \frac{dF_i(t^*)}{dt^*} dt^* = b(T_2 - T_1) = b\Delta T \qquad (11)$$

Где

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

Далее допускается, что без учёта эффекта ОС задача потребляет возобновляемый ресурс точно так, как заявлено в требованиях по качеству обслуживания, т.е. справедливо равенство:

$$R_i^{\%}(F_i(t^*)) = R^{**}(S_i(t^*)) \tag{12}$$

Учитывая равенства (10), (11), (12), после преобразований получаем:

$$\int_{T1}^{T2} R_i^{\%}(F_i(t^*))\frac{dF_i(t^*)}{dt^*}dt^* > \int_{T1}^{T2} DR_i(F_i(t^*))\frac{dF_i(t^*)}{dt^*}dt^* + b\Delta T - a\Delta T \tag{13}$$

Таким образом, 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. В отличие от CPU, особенностью планирования ввода-вывода является падение пропускной полосы при увеличении числа потребителей

Далее кратко рассматриваются невозобновляемые и частично возобновляемые ресурсы. Анализируется вид неравенства (9) применительно к таким ресурсам.

Переходим к описанию модели наложенного группового управления.

Иногда требуется объединить несколько задач в группу и предоставлять ресурс группе в целом. Под *группой задач*, мы 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, например, все задачи в группе могут представлять процессы, запущенные под определённым пользователем в системе или процессы, которые выполняются в одной виртуальной среде

Будем считать, что в любой момент времени ресурс может потреблять только одна из задач группы. Группа задач характеризуется функцией желаемого потребления *GR*, которая равна 1, если хотя бы одна из задач группы потребляет ресурс, равна 0, если никакая из задач группы ресурс не потребляет. Таким образом, для возобновляемых ресурсов:

$$GR(t_{gr}) \subset \{0,1\} \tag{14}$$

Где t_{gr} - это собственное время группы задач

Группа задач имеет собственное время, t_{gr}, которое в общем случае не совпадает с собственным временем задач в группе (совпадением может быть, если в группе только 1 задача или только 1 задача потребляет ресурс, а остальные бездействуют).

Стоит отметить, что под задачей, обычно подразумевается поток, поэтому можно говорить *группа потоков*.

Рассмотрим группу, состоящую из 2-х потоков, один из которых потребляет ресурс в моменты системного времени(кванты) 2, 7 и 11, а второй только в 9. Преобразование системного времени в групповое время, а затем в собственное время первого потока представлено на Рис. 4.

Рис 4. Преобразование времени

Значение функции потребления GR совпадает со значением функции потребления i-го потока в группе, когда i-й поток потребляет ресурс, т.е. $R_i = 1$. Очевидно, т.к. речь идёт об одном ресурсе, $GR = R_i$, в тот момент времени, когда i-й поток потребляет ресурс. Таким образом, если мы имеем N потоков в группе, которые потребляют один ресурс, то поскольку в каждый момент времени ресурс потребляется одним потоком:

$$GR(t_{gr}) = \sum_{i=1}^{N} R_i(P(t_{gr})) \tag{15}$$

$P(t_{gr})$ – функция преобразования времени группы в собственное время потока, t_{gr} – собственное время группы потока.

$$t^* = P(t_{gr}) \tag{16}$$

Функция преобразования собственного времени группы в реальное (системное) время выглядит следующим образом:

$$t = F^{gr}(t_{gr}) \tag{17}$$

Данные функции «растягивают» временную ось собственного времени группы до собственного времени задачи (см. Рис 4).

В связи с эффектами внешней среды, введём, функцию фактического потребления $GR^*(t)$, которая зависит от реального времени, для которой справедливо неравенство

$$GR^*(t) \neq GR(t) \tag{18}$$

Пусть $GR_i^{**}(t^{gr*})$ – это функция идеального потребления, т.е. та же самая функция, что и функция желаемого потребления, но в идеальном времени группы t^{gr*}. Идеальное время группы t^{gr*} - это время при котором идеально выполняется наложенное нами ограничение (например, гарантия доли дисковой пропускной способности для группы в целом). Отметим, что ограничения наложены на отдельно взятую задачу из группы могут и не выполняться. Функция преобразования собственного времени группы в в идеальное время:

$$t^{gr*} = S^{gr}(t^{gr}) \tag{19}$$

Мы будем считать, что оно зависит только от собственного времени группы, не зависит от собственных времён задач группы.

Особенности построения векторов фактического, желаемого и идеального потребления ничем не отличаются от модели на основе задачи.

Далее, с учётом, свойства (15) и, считая критерием качества, интегральное отклонение фактического потребления от идеального мы можем вывести математический критерий качества:

$$\int_{T1}^{T2} \| GR^{*}(t) - GR^{**}(t^{gr*}(t)) \| dt < A(T_2 - T_1) \qquad (20)$$

Где

T1 – начало временного отрезка наложенного управления

T2 – конец временного отрезка наложенного управления

A – допустимая погрешность с соответствующей размерностью,

Под $t^{gr*}(t)$ подразумевается зависимость идеального времени от фактического времени. Используем ту же меру векторов потребления для групп потоков. Перейдём от векторов потребления к отдельно взятым аргументам. Учитывая свойство нашей меры, имеем:

$$\int_{T1}^{T2} (GR^{*}(t) - GR^{**}(t^{gr*}(t)))^2 dt < A(T_2 - T_1) \qquad (21)$$

Поскольку, функция потребления удовлетворяет свойству (15), то, как следствие, для функций потребления группы справедливо свойство (8). Учитывая это свойство, и, переходя к собственному времени группы, имеем:

$$\int_{T1}^{T2} \{ |GR^{*}(F^{gr}(t^{gr}))| + |GR^{**}(S^{gr}(t^{gr}))| - 2GR^{**}(S^{gr}(t^{gr}))GR^{*}(F^{gr}(t^{gr}))\} \frac{dF^{gr}}{dt^{gr}} dt^{gr} < A(T_2 - T_1) \qquad (12)$$

Теперь, представим функцию фактического потребления как сумму реальной функции потребления и влияния вносимого эффектом ОС.

$$GR^{*}(F^{gr}(t^{gr})) = \tilde{GR}(F^{gr}(t^{gr})) + DG(F^{gr}(t^{gr})) \qquad (13)$$

Учитывая, что каждая задача в группе удовлетворяет выражению (11), мы можем утверждать, что группа удовлетворяет

$$\int_{T1}^{T2} (GR^{**}(S^{gr}(t^{gr})) \frac{dF^{gr}(t^{gr})}{dt^{gr}} dt^{gr} = B\Delta T \qquad (14)$$

Где

B – доля возобновляемого ресурса выделенная группе, которая является постоянной на данном временном интервале

Поскольку, мы рассуждаем о возобновляемом ресурсе, то мы можем сделать аналогичные преобразования. В результате мы имеем:

$$\int_{T1}^{T2} \tilde{GR}(F^{gr}(t^{gr}))\frac{dF^{gr}(t^{gr})}{dt^{gr}}dt^{gr} > \int_{T1}^{T2} DG(F^{gr}(t^{*}))\frac{dF^{gr}(t^{*})}{dt^{gr}}dt^{gr} + B\Delta T - A\Delta T \quad (15)$$

Таким образом, накладывая ограничения на функцию преобразования времени $F^{gr}(t_{gr})$, путём изменения скорости потребления (уменьшая значение производных в формуле (1.33), замедляя собственное время группы ограничивая доступ к диску для всех потоков группы) и анализируя поведение системы в режиме реального времени, мы можем достигать приемлемую точность выделения пропускной способности диска.

Далее приводиться пример поведения двух виртуальных серверов, которые конкурируют за дисковую пропускную способность. Показывается способ внедрения данной модели в данный пример для реализации корректного планирования дисковой пропускной способности. Для этого мы можем сколь угодно много растягивать собственное время каждой из групп, ставя задержки на выход из функций обращения к диску(например, в функции NtReadFile()).

В **главе 3** рассмотрен способ практического применения модели, для распределения пропускной способностью диска в случае возникновения конкуренции между несколькими группами потоков нескольких виртуальных серверов на примере системы Virtuozzo Linux (ядро 2.6.9-023stab033.7-smp, 4Гб RAM, 170Гб SCSI диск, 2 CPU). Производиться ряд экспериментов показывающих эффективность данной модели.

Запускается один виртуальный сервер, внутри него порождается дисковая активность чтения 64Кб блоков из 3х файлов, суммарного размера 3Гб. Находиться относительный максимум полосы пропускания в зависимости от количества потоков читающих диск. Величина 64Кб выбрана не случайно, поскольку это стандартная единица обмена между дисковым кешом и диском. Перед каждым замером производиться перезагрузка системы, для очищения дискового кеша. Данная зависимость показана сплошной линией на Рис. 6.

Далее рассматривается зависимость полосы пропускания записи от количества потоков записи. В данном случае, для того чтобы принудить систему сбрасывать данные на диск, а не помещать их в кеш, мы изначально заполнем кеш, путем прогона дополнительных итераций чтения и записи. Зависимость показана сплошной линией на Рис. 7.

В обоих случаях мы пришли к приблизительно одному и тому же значению пропускной способности, что объясняется тем, что распределение

запросов, в конце концов, происходит одним из алгоритмов аппаратного планирования, подробно рассмотренных в первой главе. Падение полосы пропускания с возрастанием количества потоков объясняется эффектом рэндомизации запросов, т.е. при чтении или записи большим количеством потоков, головка диска вынуждена совершать гораздо больше «прыжков». Так же большое влияние на пропускную способность влияет фрагментация диска, чем хаотичнее разбросаны данные, тем дольше их поиск. Использование процессора не превышало 10% в обоих случаях, что говорит об отсутсвии CPU голодания. Большая величина значения при малом количестве потоков записи обусловлена работой дискового кеша.

Далее рассматривается группы потоков с просыпающими и засыпающими потоками. В основе эксперимента лежала следующая программа. Создается *N* процессов, внутри каждого процесса создаётся несколько потоков, которые по очереди читают 640Кб(10 итераций по 64Кб) и засыпают. Таким образом, мы имеем *N* потоков в единицу времени, которые чередуются через определённые периоды. Объединим эти потоки в группу. Данный эксперимент показывает работу группового времени, поскольку время группы в целом идёт, а время отдельного потока - нет. Схематически, функцию потребления такой группы можно представить, как показано на Рис. 5. Здесь группа потребляет в течение квантов времени от 0 до 15, первый поток в кванты от 0 до 1 и от 10 до 11, второй от 1 до 2 и от 11 до 12, остальные потоки для простоты иллюстрации опущены.

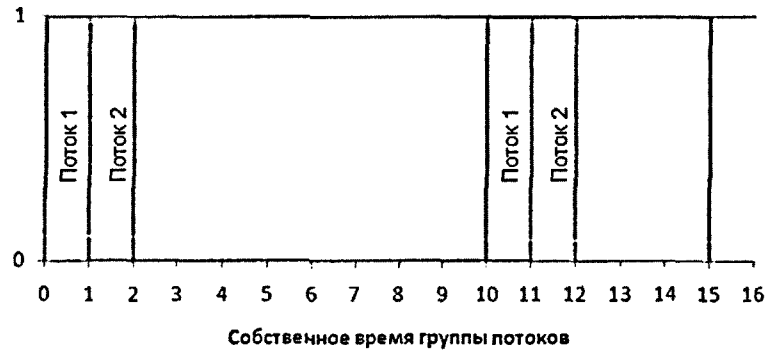

Рис 5. Пример функции потребления группы

Далее рассматривается зависимость пропускной способности чтения или записи в зависимости от *N* на таких группах. Результат, получается предсказуемым, поскольку мы снова приходим к некоторой постоянной пропускной способности обусловленной аппаратным планировщиком.

Зависимость изображена пунктирной линией, см. Рис. 6-7, для чтения и записи соответственно.

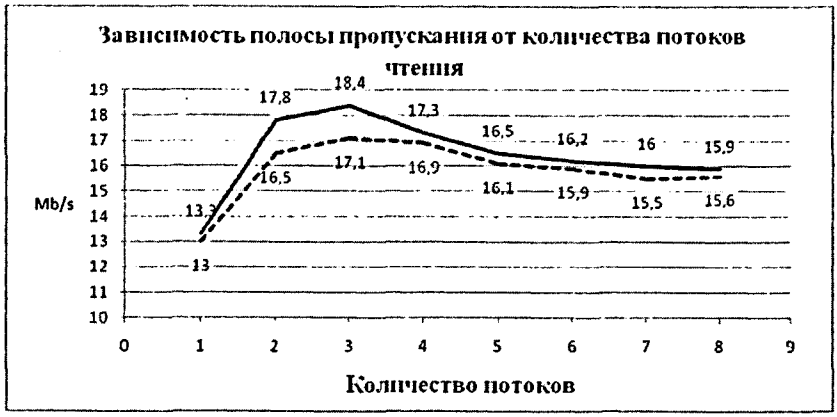

Рис 6. Полоса пропускания от количества потоков чтения в группе

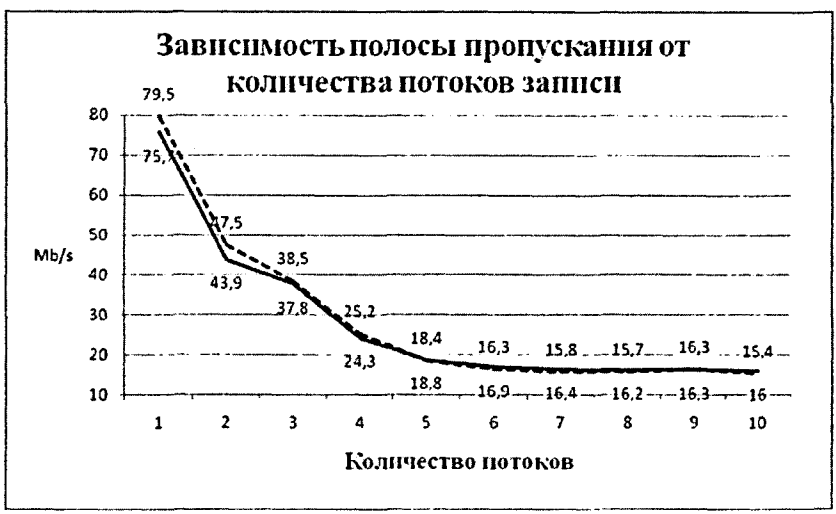

Рис 7. Полоса пропускания от количества потоков записи в группе

Большая величина значения пропускной способности при малом количестве потоков записи обусловлена работой дискового кеша.

Далее рассматривается способ распределения пропускной способности диска между двумя виртуальными серверами с требуемым соотношением(33% и 66%) доли пропускной способности между ними.

Запускается два виртуальных сервера, в каждом запускается группа из 10 потоков, читающая и пишущая на диск. В каждой группе 5 потоков пишут, а 5 потоков читают, происходит чередование потоков как в предыдущем примере. Как следствие, при паритетном запуске этих двух групп му получаем пропускную способность суммы в 16 Мб/сек, стандартный планировщик системы должен выдать им равное количество пропускной способности диска. Пусть, теперь мы хотим достигнуть соотношения 11 к 5 для отношения потребления этими группами пропускных способностей. Для этих целей мы будем тормозить вторую группу таким образом, чтобы её потребление было 5 Мб/сек. Сделаем оценку задержки, которую нужно осуществлять при чтении или записи потокам 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 8Мб в секунду, а за одно обращение мы пишем или считываем 64Кб, таким образом, группа 128 раз должна сделать обращение к диску, т.е. наша задержка при обращении должна быть

$$\frac{(8/5-1)}{128} \approx 0{,}0047 \qquad (16)$$

секунд.

На практике, для достижения требуемого соотношения потребовалась большая задержка, поскольку при добавлении задержек уменьшается и количество обращений, т.е. общая пропускная полоса уменьшается. Практическая величина задержки 0,0053 сек.

Таким образом, функция преобразования времени второй группы в данной задаче выглядит, следующим образом:

$$F^{gr}(t^{gr}) = 1{,}6784 * t \qquad (17)$$

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

В **заключении** приведены основные результаты исследования.

В **приложениях** приведены основные характеристики комплекса Virtuozzo и описание ряда других проектов в области виртуализации ресурсов, ряд экспериментов на данных комплексах, иллюстрирующих эффективность разработанной в диссертации модели.

Основные результаты и выводы диссертации

1. Разработана математическая модель группового наложенного управления ресурсами вида потоки ввода-вывода в ОС.

2. Исследованы и явно выражены ограничения, которые нужно наложить на системный планировщик, чтобы обеспечить требуемое качество управления дисковым групповым вводом-выводом.
3. Разработана математическая модель и метод группового управления дисковой пропускной способностью в условиях функционирования виртуальных серверов с целью ограничения сверху потребления дисковой пропускной способности.
4. Разработанные модели реализованы в виде комплекса программ, обеспечивающего требуемое качество обслуживания. Проведен ряд экспериментов на реальных программных комплексах. Результаты экспериментов полностью подтвердили справедливость предложенных аналитических моделей решаемым задачам.

Список публикаций по теме диссертации

1. *Луковников И., Коротаев К., Кобец А.* Проблемы управления распределяемыми ресурсами ОС // *Информационные технологии.* - М. 2006. - №10 С. 71-78.
2. *Луковников И., Коротаев К., Кобец А,* Проблемы управления распределяемыми ресурсами ОС// Приоритетные направления модернизации общего образования. Материалы Всероссийской научно-практической конференции, посвященной 60-летию ТОИПКРО. –Т. 1. – Томск: ТОИПКРО, 2006. – С. 180—181.
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. инст., 2007. – С. 272—275.
4. *Кобец А.Л, Луковников В.В., Пименов В.М., Соколов Е.В.*, Оценка точности группового наложенного управления ресурсами операционной системы для дискового ввода/вывода //"Вестник НГУ. Серия: Информационные технологии". 2007, Т. 5, вып. 1, С. 28-31
5. *Пименов В.М., Соколов Е.В., Кобец А.Л.,* Способы увеличения производительности алгоритмов для отказоустойчивых систем хранения данных// "Вестник НГУ. Серия: Информационные технологии". 2007, Т. 5, вып. 1, С. 32-39
6. *Kobets A., Votyakov K., Lukovnikov V.*, Optimization of resources distribution for high performance computation// Moscow Conference on Computational Molecular Biology (MCCMB'07), http://mccmb.genebee.msu.su/2007/, - 2c.
7. *Кудрин М.Ю., Гилимьянов Р.Ф., Кобец А.Л., Луковников В.В,* Сравнение цепочечной модели с существующими распределёнными объектными моделями// Научное творчество молодежи. Часть IV. Материалы XI Всероссийской научно-практической конференции. – Кемерово: Кемеровский гос. универ-т, - С. 7-8.
8. *Петров В.А., Луковников В.В., Кобец А.Л,* Влияние пропускной способности соединений на длительность поиска в регулярных распределённых системах// Научное творчество молодежи. Часть IV. Материалы XI Всероссийской научно-практической конференции. – Кемерово: Кемеровский гос. универ-т, - С. 10-11.
9. *Тормасов А.Г., Кобец А.Л., Луковников В.В,* Модель оценки точности группового наложенного управления ресурсами операционной системы на примере дискового ввода/вывода// Научное творчество молодежи. Часть IV. Материалы XI Всероссийской научно-практической конференции. – Кемерово: Кемеровский гос. универ-т, - С. 13-14.
10. *Votyakov K., Kobets A.*, Semi-macroscopic model for computation of electrostatic effects in membrane proton pump – *bactriorhodopsin*// Moscow Conference on Computational Molecular Biology (MCCMB'07), http://mccmb.genebee.msu.su/2007/, - 2c.

Кобец Алексей Леонидович

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ НАЛОЖЕННОГО УПРАВЛЕНИЯ РЕСУРСАМИ ВИДА "ПОТОКИ ВВОДА-ВЫВОДА" В ОПЕРАЦИОННЫХ СИСТЕМАХ

Автореферат

Подписано в печать 01.10.2007. Формат 60x90/16.
Усл печ л 1.0. Тираж 80 экз. Заказ No 470.
Московский физико-технический институт
(государственный университет)

Печать на аппарате Rex-Rotary Copy Printer 1280. НИЧ МФТИ.

141700, г Долгопрудный Московской обл, Институтский пер, 9,
тел.: (095) 4088430, факс (095) 5766582

№ 19928

2087A
19928