Санкт-Петербургский государственный электротехнический университет им. В.И. Ульянова (Ленина)

Разработка алгоритмов координированного планирования ресурсов в ОС общего назначения

Выполнил: Валерия Евгеньевна Допира, гр. 5303

Руководитель: Кирилл Владимирович Кринкин, к.т.н, зав. кафедры МО ЭВМ

Санкт-Петербург, 2021

Актуальность

- Задача операционной системы (ОС) - эффективное и честное распределение ресурсов между пользователями и программами.
- Проблема нехватки ресурсов часто решается более качественным оборудованием, но ее можно было бы решить на программном уровне.
- Существующие подходы обеспечивают гарантии производительности QoS (quality-of-service) путем управления 1-им или 2-мя взаимодействующими ресурсами. Координации между этими общими ресурсами нет.
- Первый шаг к поддержке QoS в многоядерных системах - хотя бы основные вычислительные ресурсы и ресурсы памяти управляются эффективно скоординированным образом.
- ОС Tessellation использует разделение пространства-времени как центральный компонент многоядерной операционной системы. Она обеспечивает гарантии QoS за счет контроля обмена данными между разделами.

Цель и задачи

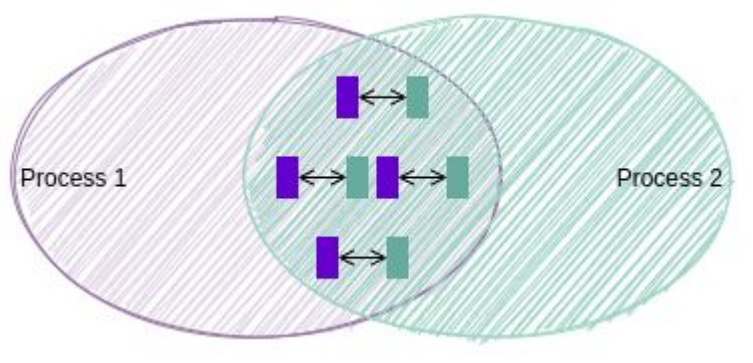
Целью работы является разработать алгоритм, при применении которого возникает выигрыш производительности операционной системы, и осуществляющий координированное планирование ресурсов.

Задачи:

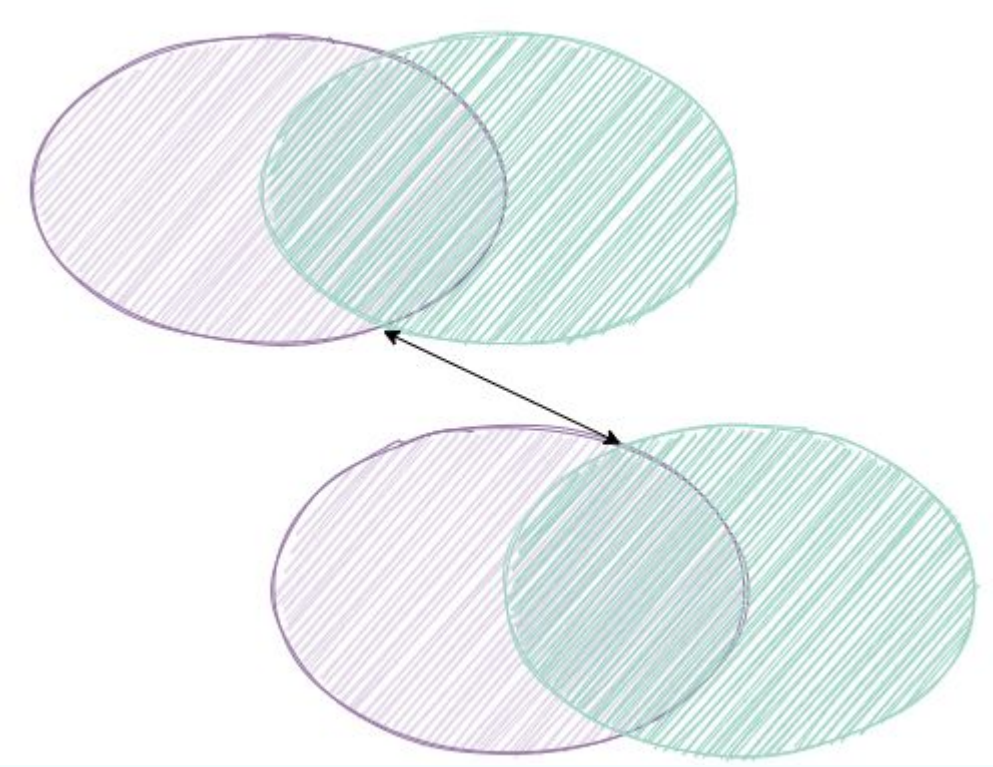
1. Сформулировать необходимость использования координированного планировании ресурсов.
2. Построить модель операционной системы.
3. Разработать алгоритм, позволяющий добиться выигрыша при координированном управлении ресурсами в ОС общего назначения.

Практическая значимость работы: разработанные механизмы позволят добиться выигрыша в производительности при координированном распределении ресурсов в ОС, особенно в мобильных телефонах.

Координированное планирование в случае конкуренции за ресурсы

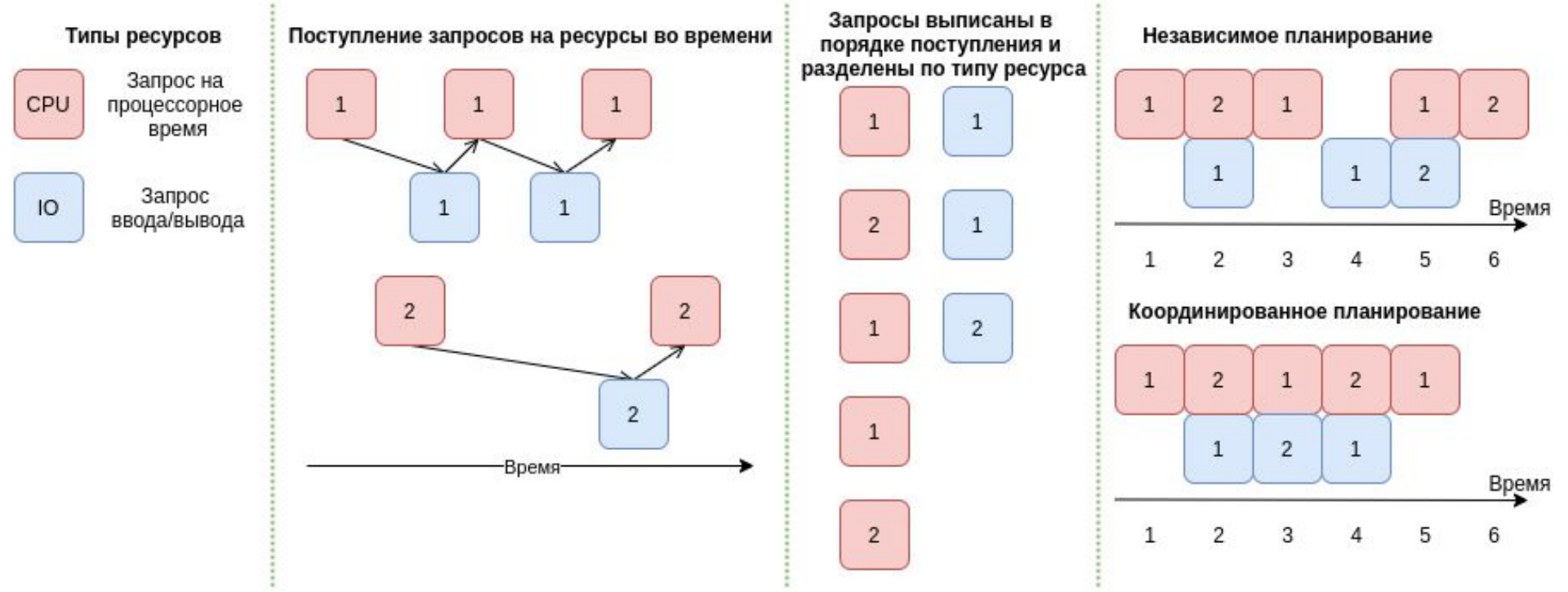


Независимое планирование



Координированное планирование

Координированное планирование в случае последовательного предоставления ресурсов



Модель координированного планирования ресурсами

Существующие ОС, в своем большинстве, планируют ресурсы независимо друг от друга, не координируя множество различных ресурсов. Они основываются на следующих принципах:

- Принятие краткосрочных решений.
- Планировщики оптимизируют неправильную целевую функцию.
- Планировщики не заботятся о количестве ресурсов.

Модель координированного управления ресурсами должна выполнять следующие задачи:

●Обеспечить конечному пользователю более эффективное использование ресурсов устройства.

●Повысить удовлетворенность владельцев устройств, то есть минимизировать время отклика ОС, выделять ресурсы приложениям переднего плана, с которым пользователи работают в данный момент.

Модель координированного планирования ресурсами

Рабочая нагрузка: направленный ациклический граф (DAG) W f =< T, E >, где T = {ti} - множество задач, а E = {ej,k} - множество ребер.

2 типа графов ожидания:

- представляющие зависимости распределения ресурсов. Например, IO, CPU, RAM, сетевые ресурсы.
- представляющие вычислительные / процессинговые потоки задач/процессов ОС.

Каждая вычислительная модель типа T: набор вычислительных узлов, на которых эта задача может быть выполнена (задача CPU, RAM, IO).

Ребро ej,k между задачами tj и tk - зависимость данных между ними.

Задача tk является дочерней и не может начать свое выполнение, пока не получит все необходимые входные данные от родительской задачи tj. succ(t) - набор дочерних задач и pred(t) возвращает ее родительские задачи.

Модель координированного планирования ресурсами

Если 2 процесса не имеют общих родителей или потомков, то можно создать новый псевдо процесс, который является родителем 2-х процессов, объединить в один DAG.

Вычислительная среда состоит из набора физических или виртуальных вычислительных ресурсов или узлов $N = \{ni\}$.

Ресурсы соединены множеством сетей: набор пропускных способностей: $B = \{bi\}$.

Расписание - это упорядоченное распределение задач по ресурсам
$S = \{(ti, nj)\}$.

Время выполнения задачи на узле: $ET(t, n)$.

Время передачи данных от родительской задачи ti к ее дочерней tj :
$TT(ti, t\ j) = ei, j/bk$

Модель координированного планирования ресурсами

Полное время выполнения задачи: $FT(t, n) = \max_{ti\_pred(t)}(TT(ti, t))+ET(t, n)$,

Фактическое время завершения задачи AFT(t): момент времени, когда задача будет завершена с учетом всех зависимостей и очередей в расписании.

Общее время выполнения: makespan = $\max_{ti\_T}$ (AFT(ti)).

Целью задачи планирования рабочего процесса является нахождение оптимального расписания S opt, которое минимизирует время выполнения рабочего процесса.

Модель координированного планирования ресурсами

Не все объекты в ОС связаны работой.

Ограничения:

- В ОС также есть мьютексы, которые ждут ресурсы. Они связаны правами доступа, а не объектами.
- Ациклические графы (DAG) не описывают циклические процессы в ОС.

Эти ограничения планируется устранить в дальнейшем развитии проекта, но они выходят за рамки данной работы.

Алгоритм

1. Построить граф DAG для процессов.
2. Построить граф B для ресурсов.
3. Сделать такой обход графа, чтобы параллельно выдавать максимально возможному количеству процессов ресурсы, которые они запрашивают.
4. Найти время выполнения задачи на узле и время передачи данных от родительской к дочерней.
5. Рассчитать полное время выполнения задачи (makespan). Выбрать процесс с минимальным временем выполнения.
6. Если завершился процесс, то он удаляется из графа DAG.
7. Вот тут должна быть проверка, что в DAG нет псевдо задач, связанных с удаленным процессом. У нас в модели создаются дополнительные вершины, чтобы граф оставался один. Если есть псевдо задачи, то их тоже удалить.
8. По истечении кванта времени, снова выбрать процессы, которым будут отдано максимально возможное количество ресурсов. Повторить нахождение оптимального расписания (алгоритм с шага 3).

Оценка качества обслуживания QoS

- Qv- Ценность (пропускная способность сети, пропускная способность памяти и батареи и т.д.).
- Qa - Время доступа (задержка, время выделения и т.д.)

  QoS = Qv* Qa

- Стоимость владения ресурсами

  AppCost*=iCost(Ri)*QoSv*QoSa=iCost(Ri)*QoS*

- Затраты на передачу класса ресурсов

  TransitionCostk1=iCost(Ri,QoSk,QoSl)

- Стоимость владения непрерывными ресурсами

  TotalAppCosttj,tk=i{l,..k}AppCosti*ti+TransitionCosti

Заключение

При эффективном управлении ресурсами нужно решать, что выгоднее:

- быстро обслужить наиболее важные запросы,
- предоставить процессам равные возможности,
- обслужить максимально возможное количество процессов, используя наибольшее число ресурсов.

Честный планировщик в Linux потоки хорошо выполняются, когда они независимы. Но непродуктивны при частой синхронизации для доступа к общим ресурсам.

Координированное планирование позволит обеспечить гарантии производительности QoS и удовлетворить требованиям к задержке и пропускной способности при совместном использовании различных ресурсов.

Предложена модель координированного управления ресурсами и алгоритм планирования. В дальнейшем будет реализован предложенный алгоритм и возможно его внедрит к себе один крупный производитель мобильных телефонов.

Апробация работы

- Публичный репозиторий apagescan доступен по ссылке: https://github.com/OSLL/apagescan
- Представлен на 26-ой Конференции Ассоциации Открытых Инноваций FRUCT, которая проходила с 23 по 24 апреля 2020 года в Ярославле, Россия. Видео выступления на конференции: https://youtu.be/nHdYpmBbnto
- Публикация: «Krinkin, K., Dopira, V., Kochneva, O., Petrov, S., Kopylov, M. «Android Memory Inspection Techniques and Tools.» 2020 26th Conference of Open Innovations Association (FRUCT). IEEE, 2020»
- Свидетельство о государственной регистрации программы для ЭВМ №2020665435. Программа для сбора, анализа и визуализации использования памяти процессами в RAM и zRAM (Apagescan), 26 ноября 2020 года.
- Также планируется опубликовать статью с описанием разработанного алгоритма.

Дополнительные слайды

apagescan