Южно-Уральский государственный университет

Научный семинар по информационным технологиям под руководством профессора Л.Б. Соколинского

Моделирование аппаратной платформы гибридных вычислительных кластеров в контексте обработки баз данных

Докладчик: Беседин К.Ю
Научный руководитель: к.ф.-м.н., доц. Костенецкий П.С.

Челябинск, 2017

Работа выполнена при финансовой поддержке гранта РФФИ № 16-37-00245 мол_а (2016-2017) "Моделирование параллельной обработки запросов с использованием сжатых колоночных индексов на кластерных вычислительных системах с многоядерными ускорителями. Руководитель — К.Ю. Беседин"

Актуальность исследования

- Под гетерогенным вычислительным кластером понимается вычислительный кластер, состоящий из вычислительных устройств, построенных на базе различных программно-аппаратных платформ.
- Согласно рейтингу TOP500[1], на сегодняшний день гетерогенные вычислительные кластера обладают наиболее высокими показателями производительности
- Особенности гетерогенных вычислительных систем требуют разработки новых подходов и решений
- Математические модели позволяют серьезно упростить создание таких решений
- Существующие математические модели не позволяют производить моделирование в контексте приложений баз данных

[1] https://www.top500.org/lists/2016/11/

Существующие модели

№	Название	Журнал/конференция	Год
1	Roofline	Williams S., Waterman A., Patterson D.A. Roofline: an insightful visual performance model for multicore architectures. // Commun. ACM, 2009. – Vol. 52. – No. 4. – P. 65–76.	2009
2	PerDome	Tang L., Hu X.S., Barrett R.F. PerDome: a performance model for heterogeneous computing systems. // Proceedings of the Symposium on High Performance Computing, part of the 2015 Spring Simulation Multiconference, SpringSim '15 April 12–15 2015, Alexandria, VA. – USA: SCS/ACM, 2015. – P. 225–232.	2015
3	Модель Лоусона	Lawson G., Sundriyal V., Sosonkina M., Shen Y. Modeling performance and energy for applications offloaded to Intel Xeon Phi. // Proceedings of the 2nd International Workshop on Hardware-Software Co-Design for High Performance Computing, Co-HPC 2015 November 15 2015, Austin, Texas. – USA: ACM, 2015. – P. 7:1–7:8.	2015
4	Модель мультипроцессоров баз данных (DMM)	Kostenetskiy P.S., Sokolinsky L.B. Analysis of Hierarchical Multiprocessor Database Systems. // Proceedings of the 2007 International Conference on High Performance Computing, Networking and Communication Systems (HPCNCS–07), July 9–12 2007, Orlando, FL. – USA: ISRST, 2007. – P. 245–251.	2007

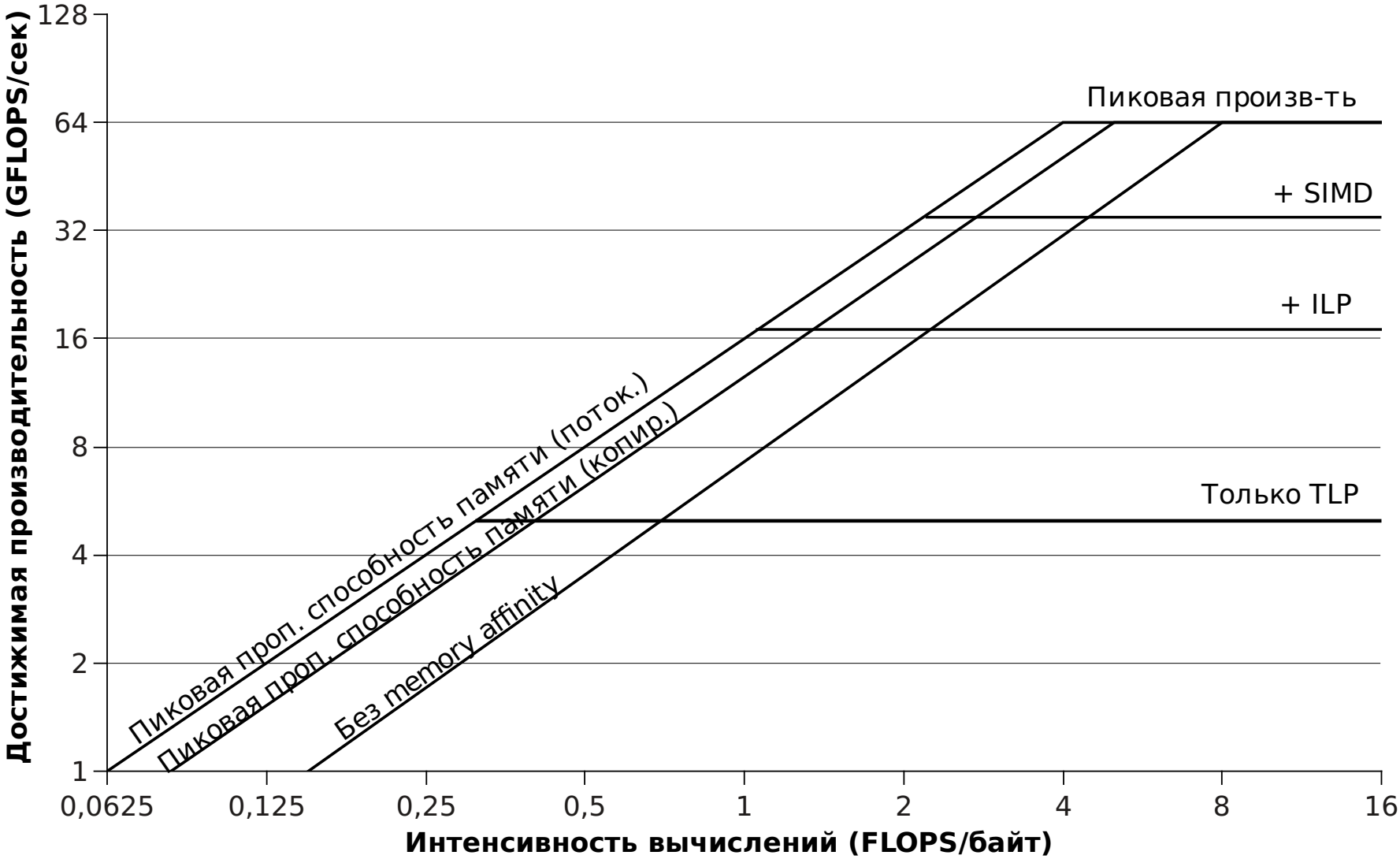
Roofline

- Предназначена для определения узких мест в реализации параллельных алгоритмов и оценки эффективности возможных оптимизаций
- Производительность определяется "интенсивностью вычислений" — отношению количества операций над числами с плавающей запятой к числу байт, считанных/записанных из/в оперативную память
- Производительность ограничена пропускной способностью памяти и производительностью операций над плавающей запятой

Roofline

- Интенсивность вычислений: $I = \frac{Q}{M}$, где Q — количество операций над данными, M — количество обменов с памятью
- Наибольшая достижимая производительность: $P = min \begin{cases} \pi \\ \beta * I \end{cases}$, где π — пиковая производительность на числах с плавающей запятой, β — пиковая пропускная способность памяти
- Зависимость максимально достижимой производительности от интенсивности вычислений изображается на графике
- На графике также могут быть отражены эффекты от различных оптимизаций
- График строится индивидуально для каждой конкретной вычислительной системы

Roofline



Пример графика Roofline

PerDome

- Основана на модели Roofline
- Предназначена для оценки производительности гибридных вычислительных систем, оснащенных GPU
- Как и в Roofline, производительность определяется "интенсивностью вычислений" — отношению количества операций над числами с плавающей запятой к числу байт, считанных/записанных из/в оперативную память, а производительность ограничена производительностью операций над числами с плавающей запятой и пропускной способностью памяти

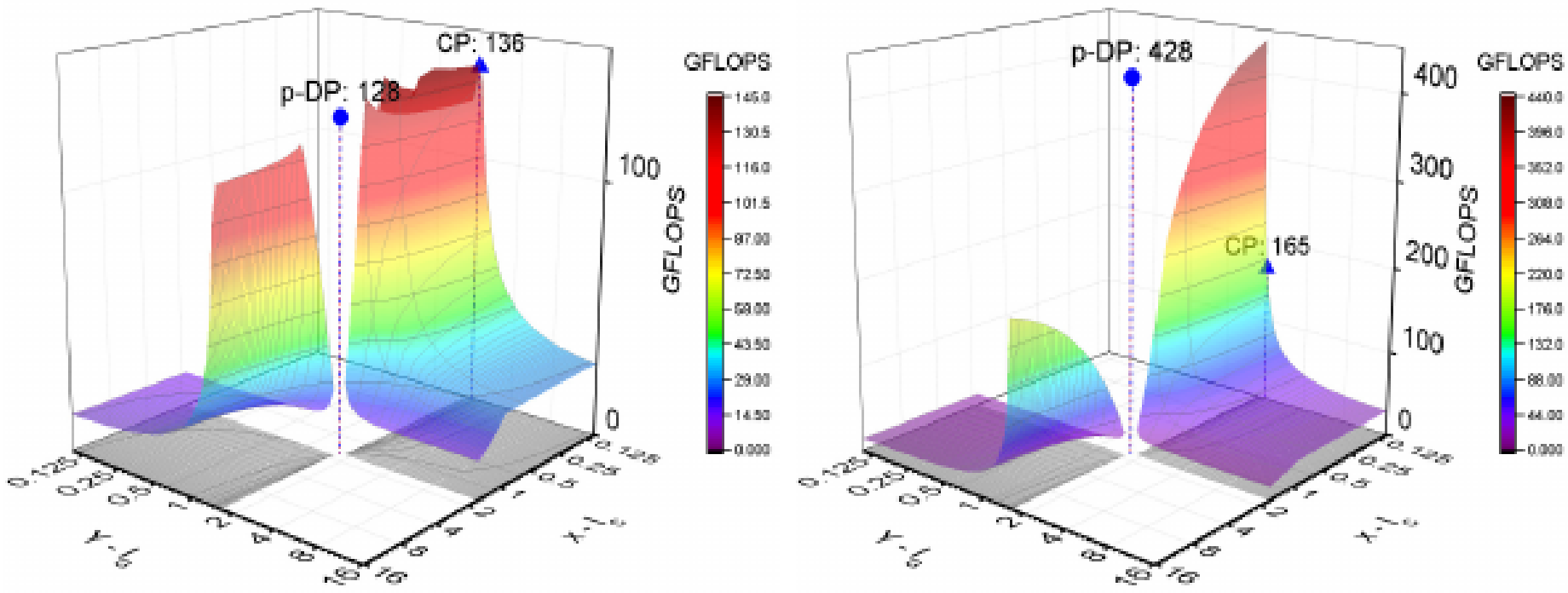
PerDome

- Предполагается, что CPU и GPU одновременно участвуют в обработке данных
- Если $\{I, I_C, I_G\}$ — интенсивности вычислений для всего приложения, кода на CPU и кода на GPU соответственно, а $\{t_{f-C}, t_{b-C}\}, \{t_{f-G}, t_{b-G}\}$ — время на одну операцию с плавающей запятой/перемещение одного байта для CPU и GPU соответственно, то производительность можно посчитать по формуле:

$$P = \max\{t_{f-C}\frac{I_C(I - I_G)}{I(I_C - I_G}, t_{b-C}\frac{I - I_G}{I(I_C - I_G}, t_{f-G}\frac{I_G(I - I_C)}{I(I_G - I_C}, t_{b-G}\frac{I - I_C}{I(I_G - I_C}\}^{-1}$$

- Зависимость производительности от интенсивности вычислений также можно изобразить в виде графика

PerDome



Пример графика PerDome

Модель Лоусона

- Предназначена для оценки производительности и энергопотребления гибридных вычислительных кластеров, оснащенных сопроцессорами Intel Xeon Phi, работающими в режиме offload
- Позволяет моделировать систему из нескольких узлов
- Основное внимание уделяется вычислениям внутри узла

Модель Лоусона

- Предполагается, что исходная задача делится на подзадачи, распределяемые по устройствам
- Процесс решения задачи моделируется как цикл, каждая итерация которого состоит из вычислительной фазы и фазы обмена данными, которые не могут выполняться одновременно
- CPU и Intel Xeon Phi не могут работать параллельно друг с другом

Модель Лоусона

- Время вычислительной фазы рассчитывается по формуле

$$T_{\text{comp}} = T_{\text{host}} + T_{\text{acc}} + T_{\text{pci}},$$

где T_{host} — время вычислений на хосте, T_{acc} — время вычислений на Xeon Phi, T_{pci} — время обмена данными по шине PCI

- Время коммуникационной фазы расчитывается по формуле

$$T_{\text{comm}} = t_l + M_{\text{comm}} * \tau_{\text{comm}},$$

где: t_l — задержки при передаче данных по сети, M_{comm} — количество данных, участвующих в обмене, τ_{comm} — время, требуемое для передачи единицы данных. В случае с обменом данными внутри одного узла, t_l считается равным нулю.

DMM

- Предназначена для моделирования обработки баз данных на параллельных вычислительных кластерах
- Не позволяет моделировать гетерогенные вычислительные системы
- моделирует только системы с иерархической структурой соединительной сети

Модель DHM

Предложена модель DHM (Database Heterogeneous Multiprocessor):

- Находится в процессе разработки
- Основана на модели DMM и является ее расширением
- Позволяет моделировать гетерогенные вычислительные системы со сложной конфигурацией соединительной сети

Подмодели модели DHM

- Модель аппаратной платформы — описывает вычислительную систему в виде *DHM-графа*, вершины которого соответствуют аппаратным компонентам системы, а ребра — каналам связи между ними
- Модель выполнения — описывает ход моделирования и правила по которым модули аппаратной платформы взаимодействуют друг с другом
- Модель транзакций — описывает механизмы работы параллельных транзакций в моделируемой системе

На данном этапе исследований предложены модель аппаратной платформы и модель выполнения.

Модель аппаратной платформы

- Вычислительная система описывается в виде *DHM-графа*
- Вершины графа — *модули* — соответствуют реальным устройствам
- Ребра соответствуют линиям связи между устройствами
- Три вида модулей — вычислительный модуль, коммуникационный модуль, модуль хранения данных

Модель аппаратной платформы

Вычислительный модуль $P \in \mathfrak{P}$:

- Устройство, используемое для выполнения процесса обработки базы данных
- Может соответствовать узлу вычислительного кластера, одному или нескольким процессорам, центральному процессору и GPU, сопроцессору Intel Xeon Phi, работающему в native-режиме и так далее
- Должен быть соединен с одним и только одним коммуникационным модулем
- В графе должен быть хотя бы один вычислительный модуль

Модель аппаратной платформы

Коммуникационный модуль $N \in \mathfrak{N}$:

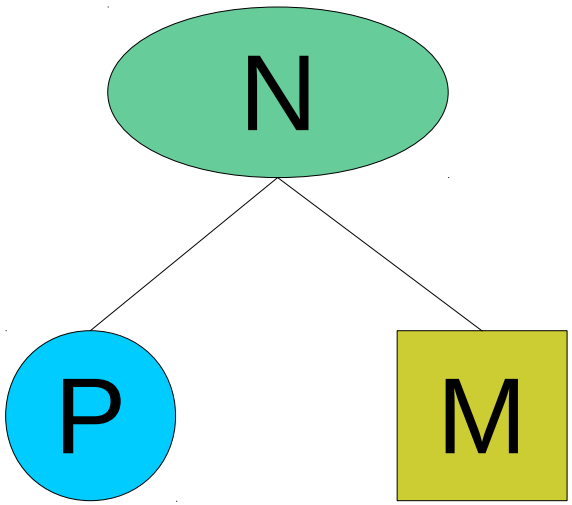
- Служат для обмена данными внутри вычислительной системы
- Может соответствовать сетевому коммутатору или внутренней компьютерной шине
- В графе должен присутствовать хотя бы один коммуникационный модуль

Модель аппаратной платформы

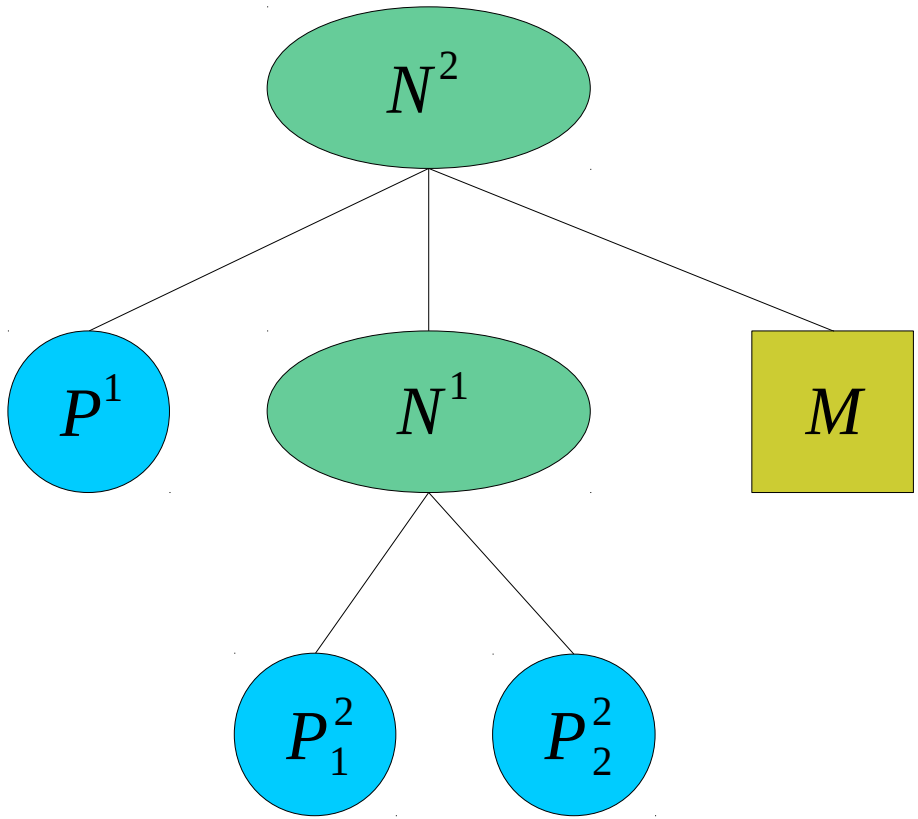
Модуль хранения данных $M \in \mathfrak{M}$:

- Имитирует устройство для хранения объектов базы данных
- Может соответствовать дисковому или твердотельному накопителю, сетевому хранилищу или модулю оперативной памяти
- Должен быть соединен с одним и только одним коммуникационным модулем
- В графе должен быть хотя бы один вычислительный модуль

Модель аппаратной платформы

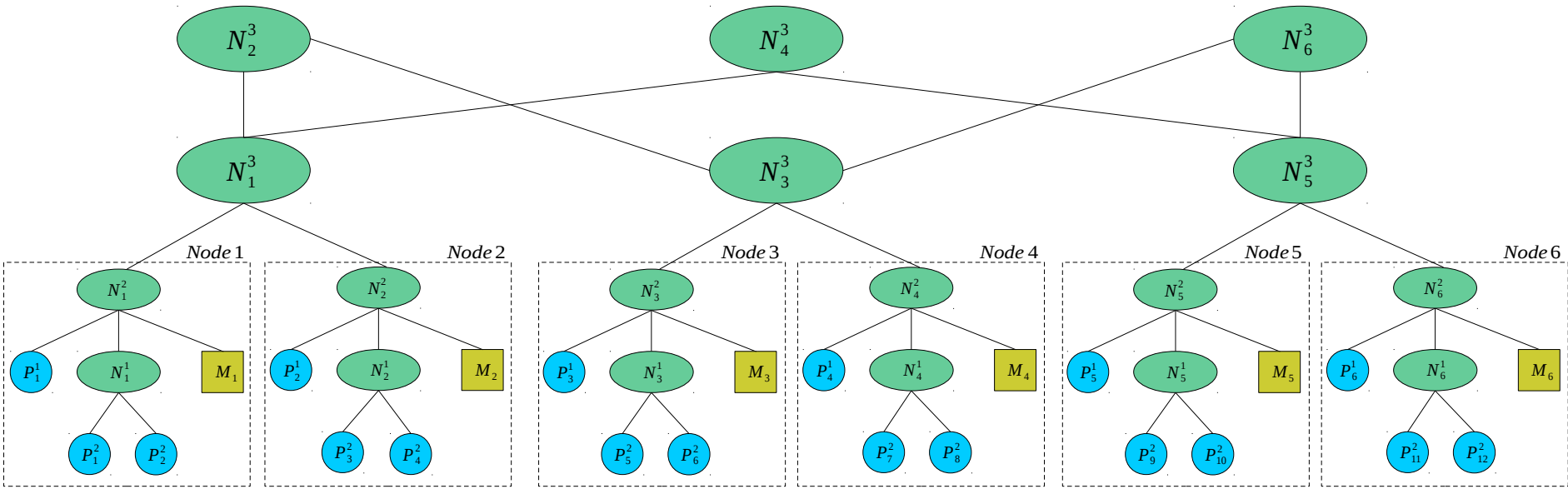


Минимально возможный DHM-граф



Пример DHM-графа для одного вычислительного узла

Модель аппаратной платформы



Пример DHM-графа для кластера со сложной структурой сети

Модель выполнения

- Объединяет модель операционной среды и стоимостную модели DMM
- Описывает ход работы модели, алгоритмы работы модулей и правила их взаимодействия

Модель выполнения

- Минимальный объем времени — такт
- Минимальная единица обмена информацией — пакет
- Операции обмена данными инициируются вычислительными модулями
- С каждым модулем ассоциируется очередь пакетов
- Допускается асинхронный обмен данными, т.е. вычислительный модуль может инициировать новые операции чтения/записи пакетов, не дожидаясь завершения уже инициированных операций
- С каждым модулем ассоциируется число $h_m \in \mathfrak{A}, A = \mathfrak{P} \cup \mathfrak{N} \cup \mathfrak{M}$ — производительность

Модель выполнения

Вычислительный модуль:

- Может инициировать операции чтения и записи пакетов
- На каждом такте обрабатывает пакеты из своей очереди
- Каждому читаемому пакету назначает число j_E — трудоемкость обработки пакета

```
begin
    for i = 0; i < h_m and not Empty(Q); i += j_E do
        E = Front(Q)
        Process(E)
    end
end
```

Алгоритм обработки пакетов процессорным модулем

Модель выполнения

```
begin
    if r(P) < s_r then
        Поместить пакет E с адресом получателя P в очередь модуля
         M
        r(P)++
    end
    else
        Wait
    end
end
```

Алгоритм чтения пакета процессорным модулем

Модель выполнения

```
begin
    if w(P) < s_w then
        Поместить пакет E с адресом получателя M в очередь модуля
         N_p
        w(P)++
    end
    else
        Wait
    end
end
```

Алгоритм записи пакета процессорным модулем

Модель выполнения

Коммуникационный модуль:

- Осуществляет передачу пакетов по сети
- Каждый пакет передается по кратчайшему пути (т.е. по пути с наименьшим числом модулей)

```
begin
    for I = 0; I < h_m and not Empty(Q); ++I do
        E = Front(Q)
        if X смежен с N then
            Поместить E в очередь X
        end
        else
            Поместить E в очередь N'_1
        end
    end
end
```

Модель выполнения

Модуль хранения данных:

- Осуществляет чтение и запись пакетов, инициированные вычислительными модулями

```
begin
    for I = 0; I < h_m and not Empty(Q); ++I do
        E = Front(Q)
        if α(E) = M then
            --(w(β(E)))
        end
        else
            Поместить E в очередь N_p
        end
    end
end
```

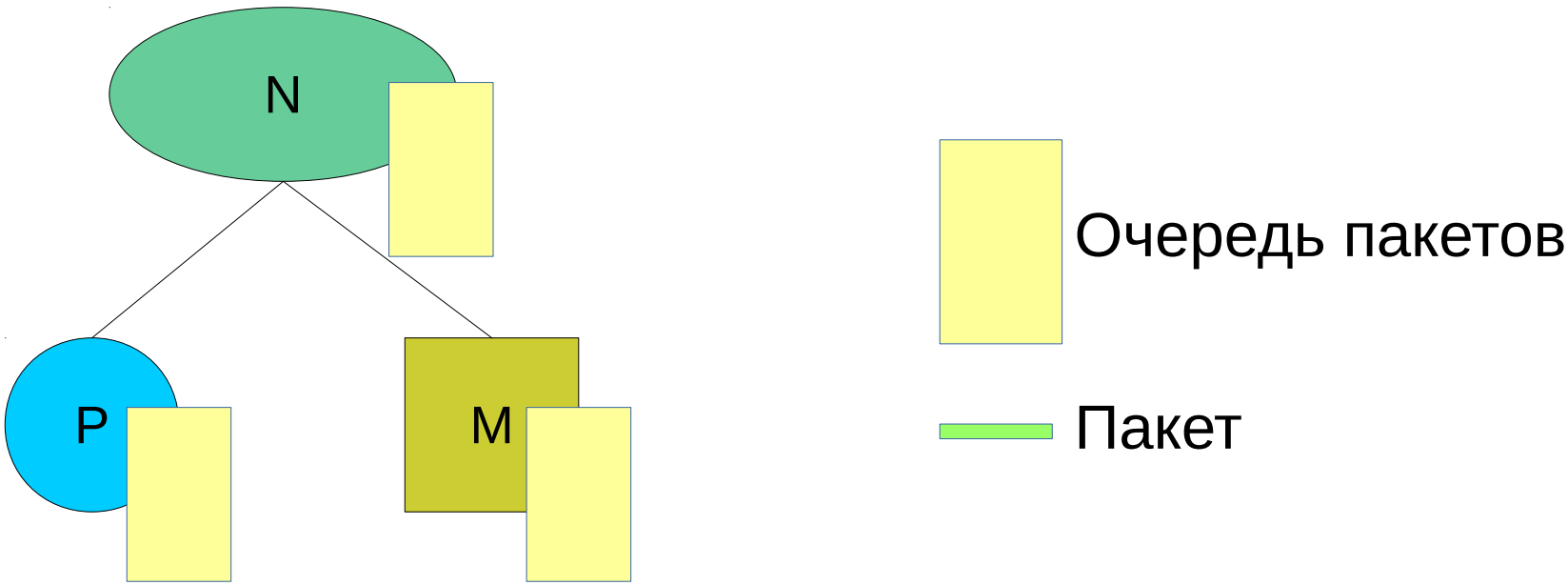
Алгоритм работы модуля хранения данных

Модель выполнения

Процесс обработки данных моделируется как цикл, состоящий из тактов модули. На каждом такте выполняются следующие действия:

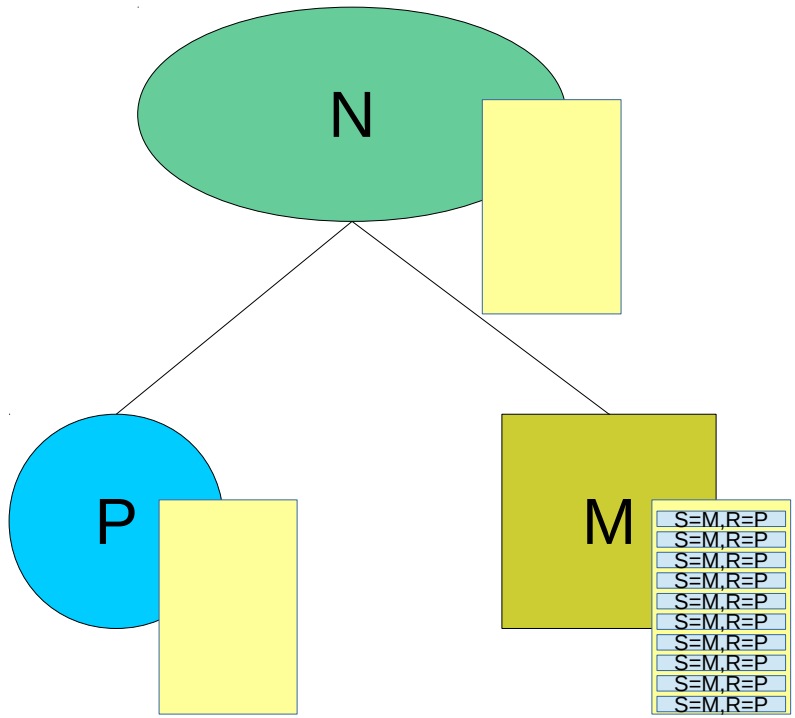
1. Каждый коммуникационный модуль M передает не более чем h_M пакетов, находящихся в очереди модуля
2. Каждый модуль хранения данных D обрабатывает не более чем h_D пакетов из своей очереди
3. Каждый вычислительный модуль P обрабатывает не более чем h_P пакетов из своей очереди и инициирует не более одной операции обмена данными с модулями хранения данных

Пример



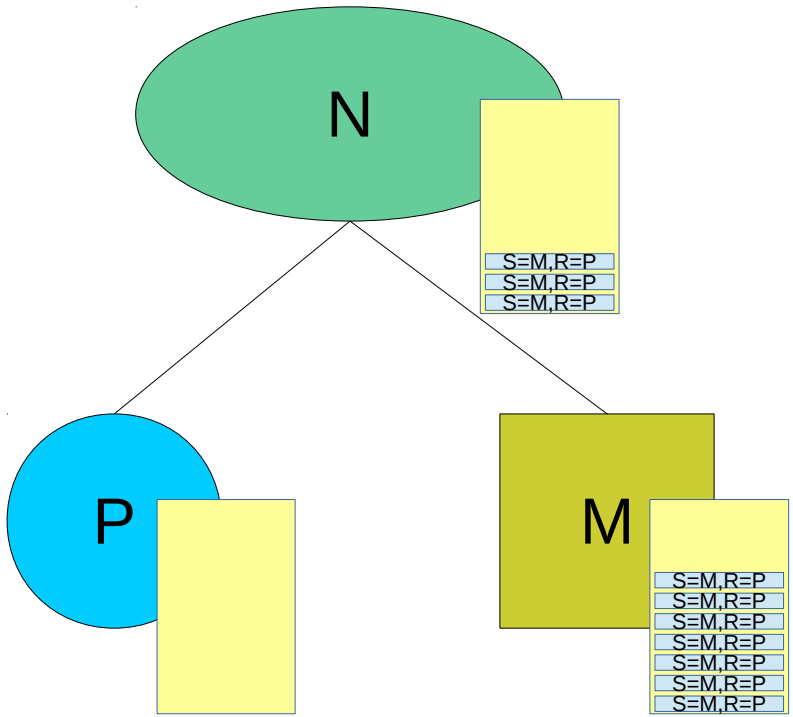
Пример: выполнение выборки. Модулируемая система состоит из трех модулей: вычислительного (M), коммуникационного (N) и модуля хранения информации (M), $h_P = h_N = h_M = 3$. Сканируемое отношение состоит из 10 блоков. Для передачи каждого блока требуется один пакет.

Пример



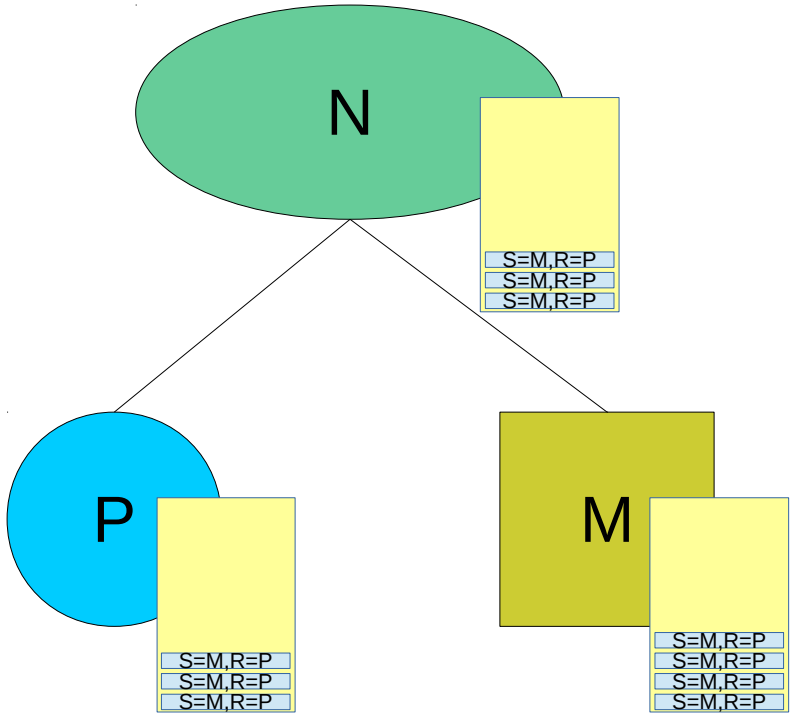
Модуль P инициирует 10 операций чтения пакетов из модуля хранения информации M. В очередь модуля M помещаются 10 пакетов с M в качестве отправителя и P в качестве получателя. Каждому пакету назначается коэффициент трудоемкости 1.

Пример



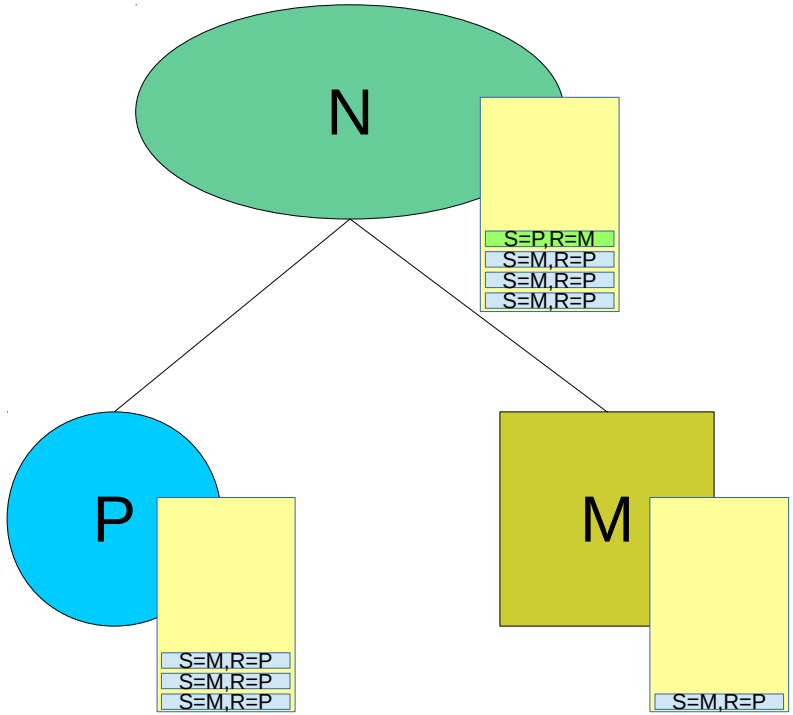
Модуль M помещает h_M пакетов в очередь модуля N.

Пример



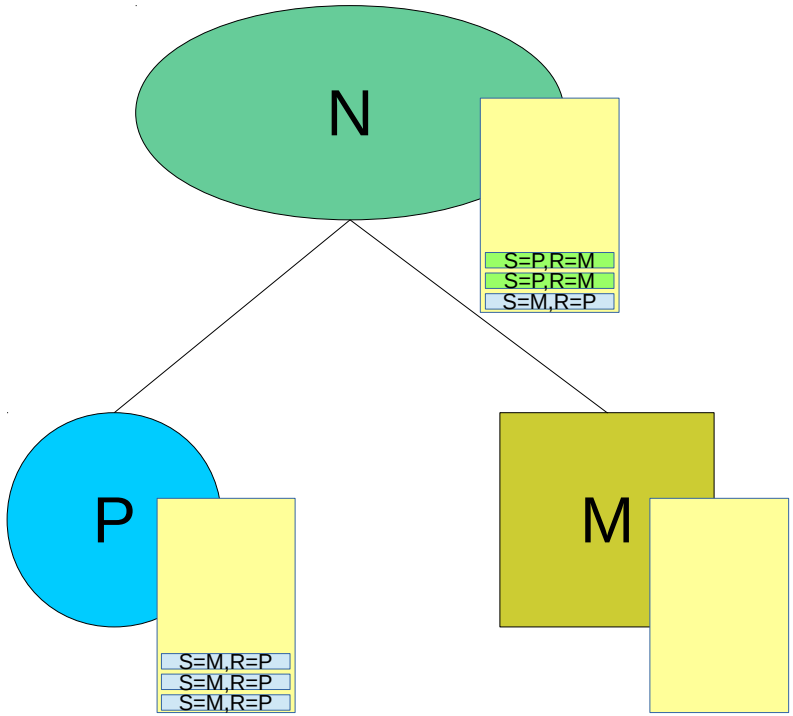
Модуль M помещает следующие h_M пакетов в очередь модуля N, модуль N передает h_N полученных на предыдущем такте пакетов модулю P.

Пример



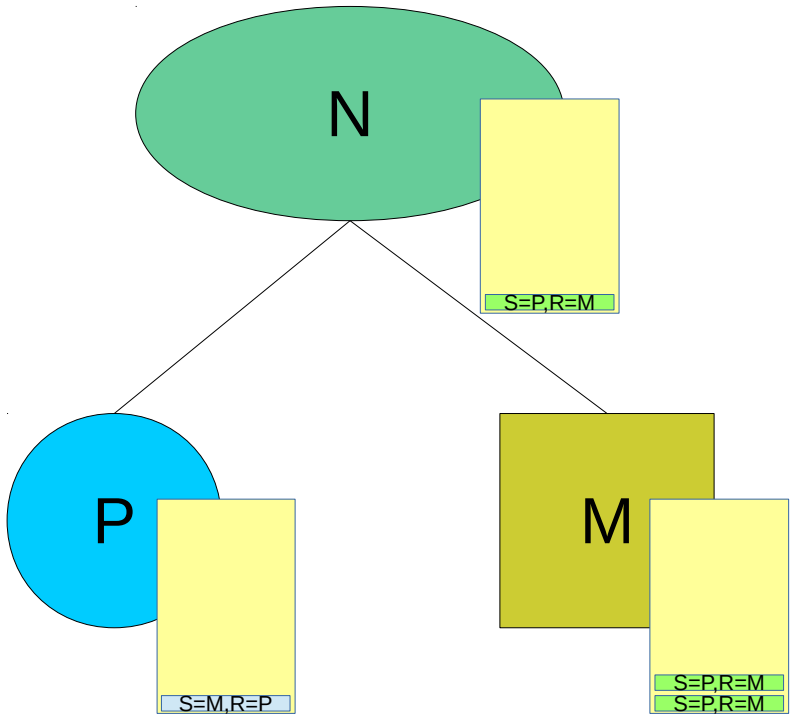
Модуль M помещает следующие h_M пакетов в очередь модуля N, модуль N передает h_N полученных на предыдущем такте пакетов модулю P. Модуль P смог обработать все полученные на предыдущем такте пакеты и инициировал операцию записи одного пакета в модуль хранения данных M. Записанный пакет с P в качестве отправителя (S=P) и M в качестве получателя (R=M) был помещен в очередь модуля N.

Пример



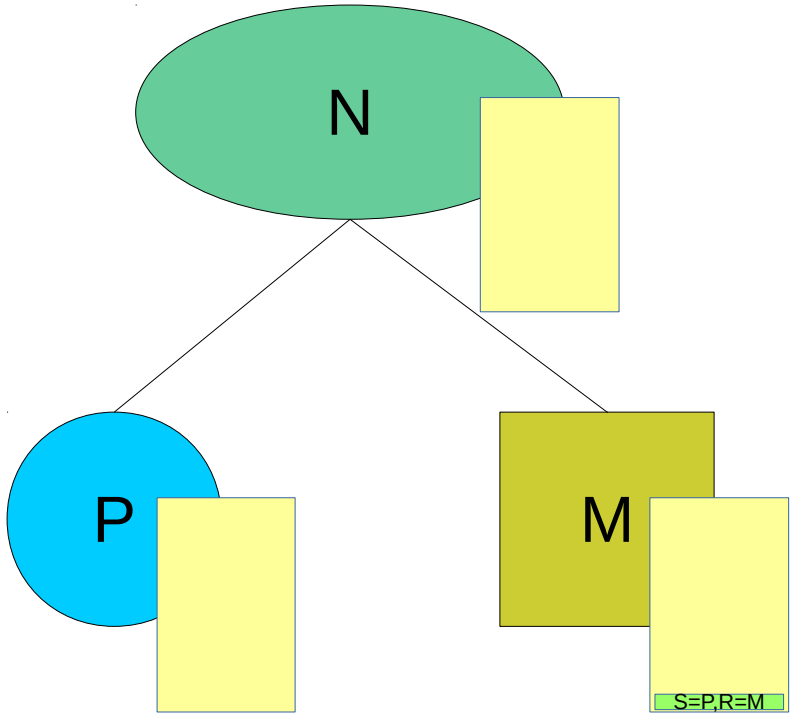
Модуль M помещает пакет в очередь модуля N, модуль N передает h_N полученных ранее пакетов модулю P. Так как количество пакетов в очереди превышало h_N, то один пакет остался в очереди. Модуль P смог обработать все полученные на предыдущем такте пакеты и инициировал операцию записи одного пакета в модуль хранения данных M. Записанный пакет с P в качестве отправителя (S=P) и M в качестве получателя (R=M) был помещен в очередь модуля N.

Пример



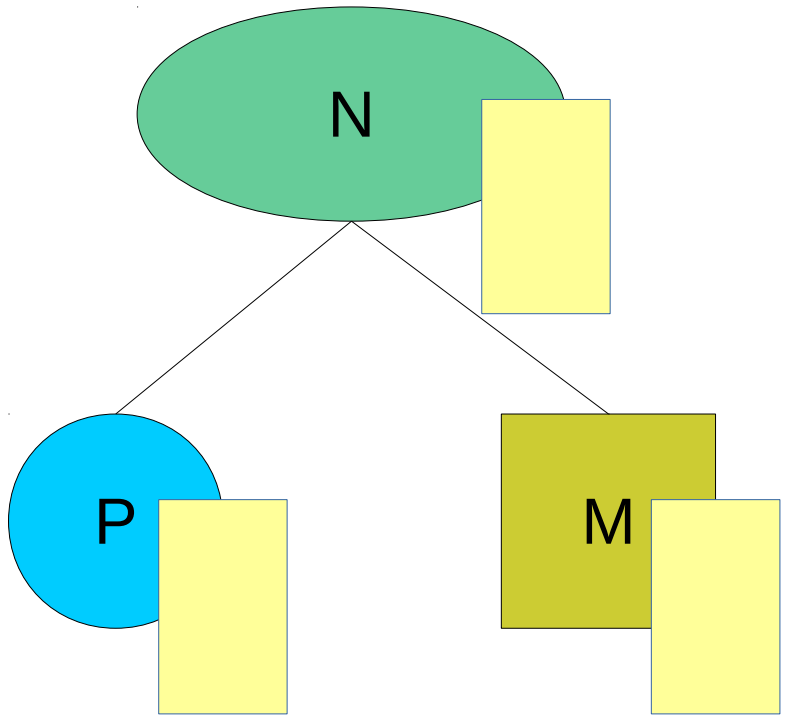
Модуль N передает h_N полученных на предыдущем такте пакетов модулям P и M. Модуль P смог обработать все полученные на предыдущем такте пакеты и инициировал операцию записи одного пакета в модуль хранения данных M. Записанный пакет с P в качестве отправителя (S=P) и M в качестве получателя (R=M) был помещен в очередь модуля N.

Пример



Модуль P обрабатывает последний пакет из своей очереди. Модуль N передает пакет из своей очереди модулю M. Модуль M обрабатывает полученные на предыдущем такте пакеты.

Пример



Модуль *M* обрабатывает последний пакет из своей очереди.
Выполнение выборки завершено.

Направления дальнейших исследований

- Разработка модели транзакций
- Программная реализация модели в виде эмулятора баз данных
- Проверка адекватности модели DHM путем сравнения результатов работы эмулятора с результатами экспериментов на реальном оборудовании
- Использование модели для поиска оптимальных алгоритмов обработки баз данных на гетерогенных вычислительных кластерах

Вопросы

Вопрос: Чем модель выполнения в DHM отличается от модели выполнения в DMM?

DMM	DHM
Модель операционной среды + стоимостная модель	Модель выполнения
Модуль обрабатывает все пакеты из своей очереди	Количество обрабатываемых модулем пакетов ограничено коэффициентом трудоемкости
Длительность такта вычисляется исходя из коэффициентов трудоемкости модулей	Длительность такта фиксирована
Трудоемкость обработки пакета процессорным модулем всегда равна единице	Трудоемкость обработки пакета вычислительным модулем зависит от характеристик модуля и моделируемого алгоритма

Вопросы

Вопрос: Где формула для подсчета стоимости?
Ответ: Отдельной формулы подсчета стоимости нет. Стоимости обработки смеси транзакций соответствует количеству затраченных на это тактов.

Вопросы

Вопрос: Как в терминах DHM описать MapReduce?
Ответ: Вычислительные модули играют роль рабочих узлов (worker).

1. На шаге *Map* вычислительные модули читают пакеты со входными данными из модулей хранения информации, обрабатывают их и записывают промежуточные результаты в модули хранения данных (шаг *Shuffle*)
2. На шаге *Reduce* вычислительные модули считывают назначенные им результаты шага *Map*, обрабатывают их и записывают конечный результат в модули хранения данных

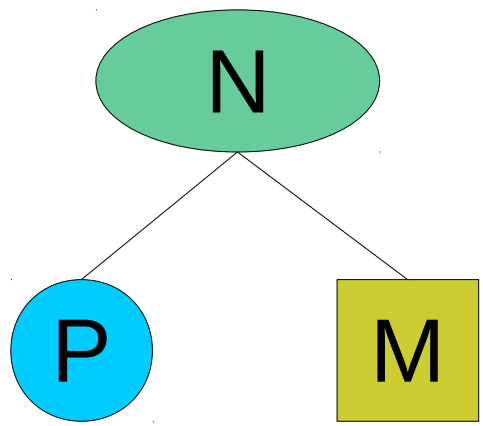
Вопросы

Вопрос: Как в терминах DHM описать фреймворк «Мастер-рабочие»?
Ответ: Вычислительные модули играют роль рабочих процессов. Главный процесс не моделируется. Предполагается, что назначение задач рабочим процессам занимает пренебрежительно малое время по сравнению с самим выполнением заданий. Вычислительные модули выполняют чтение, обработку и запись пакетов в соответствии с моделируемым алгоритмом.

Вопросы

Вопрос: Как в терминах DHM описать использование колоночного сопроцессора из диссертации Е.В. Ивановой?

Ответ: Узлы-исполнители и узел-координатор моделируются с помощью подграфа, изображенного ниже. С помощью вычислительного модуля P моделируются вычислительные ядра узлов, с помощью модуля хранения данных M — оперативная память узлов. При выполнении запросов узлы исполнители считывают пакеты с фрагментами колоночного индекса из памяти, обрабатывают их и записывают пакеты с фрагментами ТПВ в модуль хранения данных, соответствующий памяти узла-координатора.



Представление узла сопроцессора БД в DHM

Вопросы

Вопрос: Кластер на KNL – это гибридная система или нет?
Ответ: Нет, т.к. все вычислительные модули в таком кластере обладают одинаковыми характеристиками.