# Моделирование аппаратной платформы мультипроцессора баз данных, оснащенного многоядерными сопроцессорами*

К.Ю. Беседин, П.С. Костенецкий

Южно-Уральский Государственный Университет

Гетерогенные вычислительные системы, оснащенные различными видами многоядерных сопроцессоров наряду с центральными процессорами играют важную роль в высокопроизводительных вычислениях. Применение таких систем для обработки баз данных является перспективным направлением исследований. Одним из подходов исследования гибридных вычислительных систем является их математическое моделирование, однако, существующие на сегодняшний день модели вычислительных систем не позволяют выполнить это моделирование в контексте приложений баз данных. Для решения этой проблемы предлагается разработать модель гибридных мультипроцессоров баз данных (DHM), являющуюся адаптацией модели DMM для моделирования обработки баз данных на гибридных вычислительных кластерах. В данной работе описаны две подмодели модели DHM - подмодель аппаратной среды и подмодель выполнения.

*Ключевые слова:* базы данных, гибридные системы, GPU, Intel Xeon Phi, моделирование

## 1. Введение

На сегодняшний день вычислительные кластеры, оснащенные различными видами многоядерных ускорителей показывают наиболее высокие показатели производительности при выполнении научных вычислений [2]. Параллельные системы управлениями баз данных также являются областью, где востребованы возможности, предоставляемые гибридными вычислительными системами. Одной из причин, затрудняющее использование этих возможностей для обработки баз данных, является наличие у гибридных вычислительных систем целого ряда особенностей и отличий от традиционных вычислительных систем, оснащенных только центральными процессорами [1]. Грамотное использование этих особенностей является одним из ключевых требований для разработки высокопроизводительных алгоритмов для гибридных вычислительных систем. Зачастую, переход к использованию гибридных вычислительных систем требует разработку новых подходов и решений. Математическое моделирование может значительно упростить поиск таких подходов и оценку их эффективности, но существующие на сегодняшний день математические модели гибридных вычислительных систем не предназначены для моделирования обработки баз данных.

Для решения данной проблемы предлагается разработать математическую модель мультипроцессоров баз данных (DHM). В данной работе описывается начальный этап исследования, посвященного ее разработке. Предложены две подмодели модели DHM — подмодель аппаратной платформы и подмодель выполнения. В разделе "Обзор работ по тематике исследования" представлен краткий обзор известных математических моделей гибридных вычислительных систем и баз данных. В разделе "Модель DHM" дано краткое описание разрабатываемой модели DHM и описаны подмодель аппаратной платформы и подмодель выполнения.

*Работа выполнена при поддержке гранта РФФИ № 16-37-00245 (2016-2017) "Моделирование параллельной обработки запросов с использованием сжатых колоночных индексов на кластерных вычислительных системах с многоядерными ускорителями"

## 2. Обзор работ по тематике исследования

За последние несколько лет появилось несколько математических моделей, предназначенных для моделирования работы гибридных вычислительных систем.

Модель Roofline [6] предназначена для оценки производительности и эффективности различных низкоуровневых оптимизаций кода для вычислительных систем, оснащенных многоядерными процессорами. Roofline может быть также использована для оценки производительности гибридных вычислительных систем при выполнении определенных условий [5]. Основными недостатками модели Roofline в контексте моделирования параллельной обработки баз данных является невозможность моделирования кластерных вычислительных систем и неприспособленность к моделирования обработки баз данных, заключающаяся в использовании производительности операций над числами с плавающей запятой как ключевого показателя производителности и невозможности моделирования операций ввода/вывода.

В работе [4] предложена модель для оценки производительности и энергопотребления кластерных вычислительных систем, оснащенных сопроцессорами Intel Xeon Phi, работающим в режиме offload или графическими ускорителями (GPU). В рамках модели предполагается, что исходная задача делится на несколько подзадач. Решение каждой подзадачи состоит из итеративного выполнения комбинации из этапа вычислений и этапа обмена данными. При этом, вычисления и обмен данными не могут производиться одновременно. Такой ход работы не характерен для обработки баз данных, что, наряду с невозможностью моделирования операций ввода/вывода, является основным недостатком данной модели в контексте моделирования обработки баз данных.

Модель PerDome [5] предназначена для оценки производительности гетерогенных вычислительных систем. Модель представляет собой расширение модели Roofline [6] для гетерогенных систем и обладает теми же недостатками, что и она.

Модель мультипроцессоров баз данных (Database Multiprocessor Model, DMM) [3] позволяет моделировать параллельную обработку баз данных с использованием вычислительного кластера. Основным недостатком даннной модели является невозможность моделирования обработки баз данных на гибридных вычислительных системах.

## 3. Модель DHM

В качестве альтернативы рассмотренным моделям предлагается разработать модель гибридных мультипроцессоров баз данных (DHM). Данная модель создается по образцу модели DMM, разработанной в ЮУрГУ. Модель DHM состоит из нескольких подмоделей, описывающие различные аспекты работы гибридных вычислительных кластеров. На сегодняшний день выделены три подмодели:

1. подмодель аппаратной платформы — описывает аппаратные компоненты, из которых состоит моделируемая вычислительная система;

2. подмодель выполнения — описывает правила, по которым взаимодействуют компоненты аппаратной платформы и алгоритмы их работы;

3. подмодель транзакций — будет описывать правила параллельного выполнения транзакций.

В данной работе рассматриваются подмодель аппаратной платформы и подмодель выполнения. Далее приведено описание предложенных подмоделей.

### 3.1. Подмодель аппаратной платформы

Подмодель аппаратной платформы описывает вычислительную систему в виде DHM-графа. DHM-граф представляет собой связный граф, вершины которого называются модулями и являются абстрактными представлениями различных технических устройств. Модули могут относиться к одному из трех типов: вычислительные модули, модули хранения информации и коммуникационные модули. DHM граф должен содержать как минимум один модуль каждого типа. Ребра графа соответствуют линиям связи между аппаратными компонентами.

*Вычислительный модуль* $P \in \mathfrak{P}$ представляет собой устройство, используемое для выполнения процесса обработки базы данных. Вычислительный модуль может представлять собой вычислительный узел кластера, оснащенный только центральными процессорами, вычислительный узел кластера оснащенный CPU и GPU, сопроцессор Intel Xeon Phi, работающий в режиме native и т.п. В общем случае, отдельным вычислительным модулем будем считать любое устройство, используемое для обработки запросов к базе данных и способное самостоятельно инициировать обмен данными с модулями хранения данных. В DHM-графе вычислительный модуль может быть соединен только с коммуникационным модулем, причем только с одним.

*Модуль хранения данных* $M \in \mathfrak{M}$ — это устройство, предназначенное для хранения объектов базы данных. Может быть соединено только с коммуникационным модулем, причем только с одним. В реальной вычислительной системе такому модулю может соответствовать дисковый или твердотельный накопитель сетевое хранилище данных и так далее.

*Коммуникационные модули* $N \in \mathfrak{N}$ служат для обмена данными между вычислительными модулями. Модули хранения и вычислительные модули могут быть соединены только с коммуникационными модулями. В реальных вычислительных системах эти модули могут соответствовать сетевым концентраторам и компьютерным шинам.

На рис. 1 изображен пример DHM-графа для вычислительного кластера, состоящего из $n$ узлов. Каждый узел оборудован центральным процессором, который моделируется вычислительным модулем $P_i^1$, многоядерными сопроцессорами, работающими в native-режиме, моделируемыми с помощью вычислительных модулей $P_j^2$. Сопроцессоры подсоединены к шине PCI-E, представленной в графе коммуникационными модулями $N_k^1$. С помощью коммуникационных модулей $N_m^2$ моделируются системные шины вычислительных узлов и их сетевые адаптеры. Каждый узел оснащен жестким диском, который моделируется модулем хранения данных $M_o$. Узлы кластера соединены между собой сетевым концентратором, моделируемым коммуникационным модулем $N$.

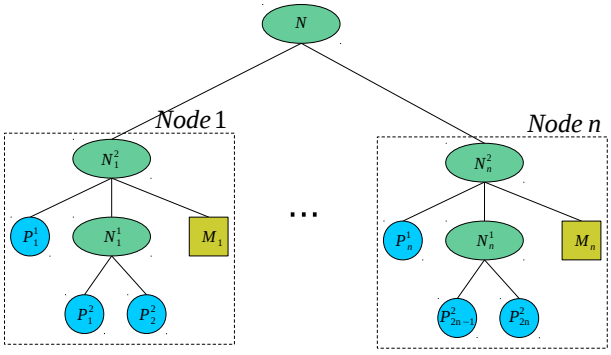


**Рис. 1.** Пример DHM-графа

### 3.2. Подмодель выполнения

Наименьшей атомарной единицей обработки данных в модели DHM является *пакет*. Все пакеты имеют одинаковый размер. У каждого пакета есть заголовок, в котором указаны

адреса отправителя и получателя. В реальной системе в качестве пакета может выступать один или несколько кортежей отношения, колонка или ее фрагмент и т.д. Операции обмена данными инициируются вычислительными модулями, которые обмениваются данными с модулями хранения информации. Вычислительный модуль может инициировать новые операции обмена данными, не дожидаясь завершения предыдущих операций. Вычислительный модуль может иметь не больше $s_r$ незавершенных операций чтения и не больше $s_w$ незавершенных операций записи.

Процесс обработки данных моделируется как цикл, состоящий из последовательности шагов, называемых *тактами*. Время такта фиксировано и не меняется во время работы модели. С каждым модулем ассоциируется *производительность* — число $h_m \in \mathbb{N}$, $1 \leq h_m < +\infty$, определяющее максимальное число пакетов, которое может быть обработано модулем в рамках одного такта. Каждый модуль обладает собственной *очередью пакетов* $Q$, в которую помещаются ожидающие обработки модулем пакеты. Размер этой очереди будем обозначать как $size(Q)$. Далее описаны действия, выполняемые модулями модели во время каждого такта.

*Вычислительный модуль*

На каждом такте модели, вычислительный модуль $P \in \mathfrak{P}$ обрабатывает находящиеся в его очереди $Q$ пакеты и инициирует некоторое количество операции чтения и записи пакетов. Количество и операций чтения и записи пакетов определяется моделируемым алгоритмом. Псевдокод алгоритма обработки пакетов вычислительным модулем представлен на рис. 2.

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

**Рис. 2.** Алгоритм обработки пакетов процессорным модулем

Пусть вычислительному модулю $P$ требуется прочитать пакет $E$ с модуля хранения $M \in \mathfrak{M}$. Каждому читаемому пакету назначается *стоимостный коэффициент* $j_E \in \mathbb{N}$, $1 \leq j_E < h_m$, означающий трудоемкость обработки пакета в контексте выполняемых над ним действий. Псевдокод алгоритма чтения пакета приведен на рис. 3.

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

**Рис. 3.** Алгоритм чтения пакета процессорным модулем

Если вычислительный модуль ранее инициировал $s_r$ еще не завершенных операций чтения, то он переводится в режим ожидания. Если количество незавершенных операций чтения меньше $s_r$, то в очередь модуля $M$ помещается пакет $E$ с адресом получателя $P$ и адресом отправителя $M$. Под $r(P)$ понимается количество незавершенных операций чтения вычислительного модуля $P$.

Пусть вычислительному модулю $P$ требуется записать пакет $E$ в модуль хранения $M \in \mathfrak{M}$. Если вычислительный модуль ранее инициировал $s_w$ еще не завершенных операций записи, то он переводится в режим ожидания. Если количество операций меньше $s_w$, то в очередь модуля $N_p$ помещается пакет $E$ с адресом получателя $M$ и адресом отправителя $P$. Псевдокод алгоритма записи пакета приведен на рис. 4, где $N_p$ — коммуникационный

модуль, соединенный с модулем $P$.

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

**Рис. 4.** Алгоритм записи пакета процессорным модулем

*Коммуникационный модуль*

На каждом такте модели коммуникационный модуль $N \in \mathfrak{N}$ осуществляет передачу пакетов по соединительной сети. Для передачи пакета коммуникационный модуль извлекает его из своей очереди, определяет модуль, которому следует передать пакет и помещает пакет в очередь выбранного модуля. В рамках модели предполагается, что передача будет производиться по пути с кратчайшей длиной. Если существует несколько таких путей, то случайным образом выбирается один из них.

Пусть модуль $X \in \mathfrak{P} \cup \mathfrak{M}$ — модуль-получатель, указанный в заголовке пакета. Если модуль $X$ смежен с модулем $N$, то пакет помещается в очередь модуля $X$. Если же модуль $X$ не смежен с модулем $N$, то пусть $(N, N'_1, N'_2, \ldots, N'_k, X)$ — модули, через которые проходит кратчайший путь из модуля $N$ в модуль $X$, где $N, N'_i \in \mathfrak{N}$, $N_{i+1}$ смежен с модулем $N_i$, а $N'_1$ смежен с модулем $N$. В этом случае, извлеченный пакет помещается в очередь модуля $N'_1$. Псевдокод алгоритма работы коммуникационного модуля приведен на рис. 5.

```
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
   else
      Поместить E в очередь N'_1
```

**Рис. 5.** Алгоритм работы коммуникационного модуля

*Модуль хранения данных*

На каждом такте модели модуль хранения данных производит чтение и запись пакетов, инициированные вычислительными модулями. Алгоритм работы модуля хранения данных приведен на рис. 6.

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

**Рис. 6.** Алгоритм работы модуля хранения данных

Здесь $N_m$ — коммуникационный модуль, смежный модулю хранения данных, $\beta(E)$ — отправитель пакета, $w(\beta(E))$ — количество незавершенных операций записи отправителя.

## 4. Заключение

Данная работа описывает еще не завершенное научное исследование, посвященное разработке модели гибридных мультипроцессоров баз данных DHM, предназначеной для мо-

делирования обработки баз данных на гибридных вычислительных кластерах. Разрабатываемая модель состоит из трех подмоделей, описывающих различные аспекты работы гибридной вычислительной системы. В рамках данной работы предложены две такие подмодели — подмодель аппаратной платформы, описывающая аппаратные компоненты, из которых состоит моделируемая система, и подмодель выполнения, описывающая правила, по которым компоненты аппаратной платформы взаимодействуют между собой. Дальнейшими направлениями работы будут:

- разработка подмодели транзакций;
- программная реализация модели в виде эмулятора баз данных;
- проверка адекватности предложенных подмоделей путем сравнения результатов работы эмулятора с результатами экспериментов на реальном оборудовании;
- использование модели для поиска оптимальных алгоритмов обработки баз данных на гибридных вычислительных кластерах.

## Литература

1. Besedin K.Y., Kostenetskiy P.S., Prikazchikov S.O. Increasing Efficiency of Data Transfer Between Main Memory and Intel Xeon Phi Coprocessor or NVIDIA GPUS with Data Compression. // Lecture Notes in Computer Science, Springer, 2015. – Vol. 9251. – P. 319–323.

2. Костенецкий П.С., Сафонов А.Ю. Суперкомпьютерный комплекс ЮУрГУ // Параллельные вычислительные технологии (ПаВТ'2016): труды международной научной конференции (28 марта–1 апреля 2016 г., г. Архангельск). Челябинск: Издательский центр ЮУрГУ, 2016. – С. 561–573

3. Костенецкий П.С., Соколинский Л.Б. Моделирование иерархических многопроцессорных систем баз данных // Программирование, 2013. – Т. 39. – № 1. – С. 3–22.

4. 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.

5. 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.

6. 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.

# Modeling the Hardware Platform of the Database Multiprocessor, Equipped With Manycore Coprocessors

Konstantin Y. Besedin, Pavel S. Kostenetskiy

South Ural State University, Chelyabinsk, Russia

Heterogeneous computational systems, equipped with manycore coprocessors or GPUs play very important role in high performance computing. Scientific community shows a growing interest in using such systems for database processing. One of important approaches for studying hybrid computational systems is mathematical modelling, but existing models of computational systems are not suited to be used for modelling database processing. To address this problem, we propose to develop the Heterogeneous Database Multiprocessor model (DHM) by extending the DMM model for modelling database processing on hybrid computational systems. This paper describes two submodels of DHM — hardware platform submodel and execution submodel.

*Keywords:* databases, hybrid systems, GPU, Intel Xeon Phi, modelling

## References

1. Besedin K.Y., Kostenetskiy P.S., Prikazchikov S.O. Increasing Efficiency of Data Transfer Between Main Memory and Intel Xeon Phi Coprocessor or NVIDIA GPUS with Data Compression. // Lecture Notes in Computer Science, Springer, 2015. – Vol. 9251. – P. 319–323.

2. Kostenetskiy P.S., Safonov A.Y. SUSU Supercomputer Resources // Proceedings of the 10th Annual International Scientific Conference on Parallel Computing Technologies (PCT 2016). Arkhangelsk, Russia, March 29–31, 2016. CEUR Workshop Proceedings, 2016. – Vol. 1576. – P. 561–573. ( Костенецкий П.С., Сафонов А.Ю. Суперкомпьютерный комплекс ЮУрГУ // Параллельные вычислительные технологии (ПаВТ'2016): труды международной научной конференции (28 марта–1 апреля 2016 г., г. Архангельск). Челябинск: Издательский центр ЮУрГУ, 2016. – С. 561–573 )

3. Kostenetskii P.S., Sokolinsky L.B. Simulation of Hierarchical Multiprocessor Database Systems // Programming and Computer Software, 2013. – Vol. 39. – No. 1. – P. 10–24. ( Костенецкий П.С., Соколинский Л.Б. Моделирование иерархических многопроцессорных систем баз данных // Программирование, 2013. – Т. 39. – № 1. – С. 3–22. )

4. 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.

5. 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.

6. 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.