Разработка модели колоночного сопроцессора баз данных*

С.О. Приказчиков, П.С. Костенецкий

Южно-Уральский государственный университет

В работе предлагается математическая модель, позволяющая исследовать эффективность различных аппаратных конфигураций кластерных вычислительных систем, построенных на базе многоядерных сопроцессоров, при обработке баз данных с использованием подхода распределенных колоночных индексов.

Ключевые слова: распределенные колоночные индексы, колоночный сопроцессор, параллельные базы данных.

1. Введение

Объем данных, генерируемых человечеством, постоянно увеличивается [3]. В то же время, существует множество задач, для решения которых необходима оперативная обработка накопленных сверхбольших баз данных [6]. Для их обработки необходимы новые методы и гибридные аппаратных архитектуры, содержащие многоядерные сопроцессоры и графические ускорители [1, 2].

В работах [7, 8] был предложен подход для выполнения запросов к сверхбольшим базам данных на основе распределенных колоночных индексов и доменно-интервальной фрагментации с использованием многоядерных сопроцессоров. Он позволяет существенно повысить эффективность обработки запросов в параллельных СУБД.

В настоящее время уже существуют модели параллельных вычислений: PRAM [13], BSP [15]. Но они не учитывают специфику параллельных систем баз данных. Также существуют модели, которые учитывают данную специфику: DMM [11], HDM [14], но они не поддерживают распределенные колоночные индексы.

В соответствии с этим актуальной является задача моделирования систем баз данных [10], работающих на кластерных системах, оснащенных многоядерными сопроцессорами, и использующих распределенные колоночные индексы.

2. Модель колоночного сопроцессора

Модель колоночного сопроцессора включает в себя три подмодели: модель аппаратной платформы, модель операционной среды и стоимостную модуль.

2.1. Модель аппаратной платформы

Множество модулей многопроцессорной системы разбивается на три непересекающихся подмножества:

$$\mathbb{M} = \mathbb{C} \cup \mathbb{N} \cup \mathbb{E}, \mathbb{C} \cap \mathbb{N} = \emptyset, \mathbb{N} \cap \mathbb{E} = \emptyset, \mathbb{C} \cap \mathbb{E} = \emptyset,$$

где: $\mathbb{C}$ – множество сопроцессорных модулей, $\mathbb{N}$ – множество модулей сетевых концентраторов, $\mathbb{E}$ – множество коммуникационных линий между двумя устройствами. Модуль сопроцессора-координатора обозначим $C_1 \in \mathbb{C}$, остальные сопроцессорные модули $C_n \in \mathbb{C}$, где $n = \{2, 3, \dots\}$, будем называть сопроцессорными модулями-исполнителями.

Передача данных между сопроцессорами по шине PCI-Express является узким местом [4, 9] и многократно замедляет обработку данных. Поэтому целесообразно хранить и обрабатывать распределенные колоночные индексы [8] непосредственно на сопроцессорных модулях. В связи с этим модули дисковых устройств и основной оперативной памяти не моделируются.

* Работа выполнена при финансовой поддержке гранта РФФИ (проект 16-37-00245-мол_а).

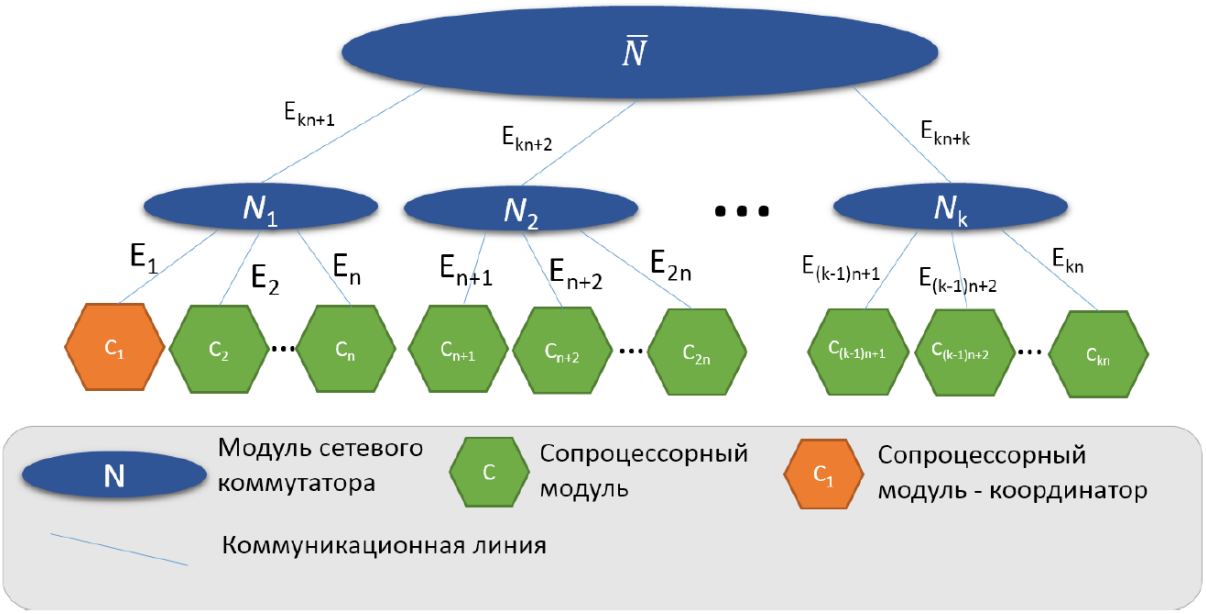

Рис. 1. Пример DCH дерева

DCH-деревом (Database Coprocessor's Hardware) будем называть взвешенное дерево, состоящее из множества модулей многопроцессорной системы, в котором имеется выделенная вершина $\overline{N} \in \mathbb{N}$, называемая корневым сетевым концентратором. На структуру *DCH*-дерева накладываются следующие ограничения:

1) корнем *DCH*-дерева может быть только модуль сетевого коммутатора;
2) каждый *i*-й модуль сетевого коммутатора имеет вес w_i.
3) листьями DCH дерева могут быть только сопроцессорные модули, каждый из которых имеет коэффициент производительности p^i;
4) сопроцессорные модули не могут иметь дочерних узлов;
5) любые два модуля соединяются посредством соединительного канала, представляющего собой ребро дерева, имеющее вес w_j,

где: *j*-номер модуля.

Пример *DCH*-дерева приведен на рис. 1. Данный пример соответствует вычислительному кластеру с k вычислительными узлами, каждый из которых оснащен n сопроцессорами.

2.2. Модель операционной среды

Минимальной единицей данных в предлагаемой модели будем считать кортеж таблицы предварительных вычислений (ТПВ) [5].

Время обработки данных на сопроцессорном модуле вычисляется по следующей формуле:

$$t_{\text{обр.}} = \frac{f_{\text{тр}}(|I_j^i|,|I_k^i|)}{p^i}, \quad (1)$$

где: $f_{\text{тр}}$– функция трудоемкости для конкретной операции с фрагментами распределенных колоночных индексов I_j^i, I_k^i , находящихся на *i*-ом сопроцессорном модуле;

p^i – производительность i-го сопроцессорного модуля (количество операций, которое способен обработать сопроцессорный модуль за единицу времени).

Будем считать, что фрагменты колоночных индексов заранее распределены по сопроцессорам, как это предложено в работе [3].

Пакет представляет собой сжатую часть таблицы предвычислений, предназначенную для отдельного сопроцессорного модуля.

Время работы модели разбивается на такты. Тактом будем называть последовательность действий, состоящую из следующих этапов:

1) обработка i-го соединения индексов;

2) разделение ТПВ на части, необходимые для передачи на каждый сопроцессор;
3) пересылка ТПВ, подготовленной в п. 2, по соответствующим сопроцессорам;
4) формирование из полученных частей ТПВ колоночных индексов.

Количество тактов определяется числом реляционных операций соединения (п. 2.1), необходимых для выполнения запроса и равняется n-1, где n – число отношений, участвующих в реляционных операциях соединения. На последнем такте ТПВ на части не делится, а отправляется на сопроцессорный модуль-координатор.

Время формирования колоночных индексов из ТПВ равняется:

$$t_{\text{созд.инд.}} = \frac{f_{\text{сложн.сорт.}}(|\text{ТПВ}|)}{p^i} \cdot N, \quad (2)$$

где: $f_{\text{сложн.сорт.}}$– функция сложности сортировки, используемой для формирования колоночных индексов; N – количество атрибутов ТПВ; p^i – производительность i-го сопроцессорного модуля (количество операций, которое способен обработать сопроцессорный модуль за единицу времени).

Количество кортежей, получаемых на i-ом сопроцессорном модуле-исполнителе в результате выполнения операции соединения на *j*-том такте, вычисляется по формуле:

$$n = \frac{\varepsilon_j}{N}, \quad (3)$$

где: ε_j – количество кортежей, получаемых всеми сопроцессорными модулями на j-ом такте;
N – количество сопроцессорных модулей-исполнителей.

Передача ТПВ между сопроцессорами моделируется как передача пакетов. Пакет представляет собой кортеж ТПВ. Количество пакетов вычисляется по формуле:

$$m = \frac{|\text{ТПВ}_i|}{N}, \quad (4)$$

где: ТПВ_i – часть ТПВ, получаемая на *i*-ом сопроцессорном модуле-исполнителе;
N – количество сопроцессорных модулей-исполнителей.

Время третьего этапа каждого такта вычисляется по алгоритму, изображенному на рис. 2.

```
функция ПолучитьВремяПередачи():
  общееВремяОтправки = 0
  пока есть пакеты на одном из модулей:
     для каждого модуля:
         модуль.ОтправитьДальше()
     общееВремяОтправки = общееВремяОтправки + 1
  вернуть общееВремяОтправки
```

Рис. 2. Алгоритм пересылки пакетов

Метод ОтправитьДальше() при наличии пакетов на модуле отправляет:

– для соединительного канала – на родительский и дочерний модуль по $\frac{w_j}{q}$ пакетов,

– для модуля сетевого коммутатора – на родительский и каждый дочерний модуль по $\frac{w_j}{q}$ пакетов,

– для сопроцессорного модуля-исполнителя – все пакеты на родительскую коммуникационную линию,

где: w_j – весовой коэффициент модуля *DCH* дерева, с которого происходит отправка;

q – количество атрибутов ТПВ, полученной на предыдущем этапе.

2.3. Стоимостная модель

Время работы моделируемой системы баз данных равняется сумме времени тактов модели:

$$t_{\text{общее}} = \sum_1^n t_i, \quad (5)$$

где: t_i – время тактов;
n – количество тактов.

Время такта состоит из времени выполнения запроса над фрагментами колоночного индекса, отправкой частей ТПВ на каждый сопроцессорный модуль, формирования распределенных колоночных индексов из полученной ТПВ на каждом сопроцессорном модуле.

2.4. Моделируемый запрос

Пусть даны реляционные отношения:

$\mathbf{R}_1$, состоящее из атрибутов $A_1^1, A_2^1, \dots, A_{k_1}^1$, обозначим $\{A_1^1, A_2^1, \dots, A_{k_1}^1\}$ как A^1;

$\mathbf{R}_2$, состоящее из атрибутов $A_1^2, A_2^2, \dots, A_{k_2}^2$, обозначим $\{A_1^2, A_2^2, \dots, A_{k_2}^2\}$ как A^2;

…

$\mathbf{R}_n$, состоящее из атрибутов $A_1^n, A_2^n, \dots, A_{k_n}^n$, обозначим $\{A_1^n, A_2^n, \dots, A_{k_n}^n\}$ как A^n.

Обозначим $\mathbb{A}$ набор всех атрибутов всех отношений: $A^1 \cup A^2 \cup \dots \cup A^n = \mathbb{A}$.

Необходимо выполнить реляционный запрос P:

$$P = \pi_{\mathbb{B}}\left(\sigma_{f(\mathbb{C})}(\boldsymbol{R}_1 \bowtie \cdots \bowtie \boldsymbol{R}_n)\right),$$

где: $\mathbb{B} \subset \mathbb{A}$ – набор атрибутов, из которых должно состоять результирующее реляционное отношение;

$f: \mathbb{C} \to \{0,1\}$ – булева функция, задающая условия выборки;

$\mathbb{C} \subset \mathbb{A}$ – множество атрибутов, участвующих в условии выборки.

3. Вычислительные эксперименты

Было проведено две серии вычислительных экспериментов: на первой производилась настройка модели путем задания весовых коэффициентов, на второй – произведена проверка адекватности модели.

Для настройки эмулятора были проведены эксперименты на суперкомпьютере «Торнадо ЮУрГУ» [11], основные характеристики которого приведены в таблице 1.

В ходе вычислительных экспериментов второй серии выполнялось соединение двух отношений *R* и *S*. Размер отношения *R* составляет 630 000 кортежей, а размер отношения *S* – 63 000 000 кортежей. Соединение производилось с использованием распределенных колоночных индексов.

Таблица 1. Характеристики суперкомпьютера «Торнадо ЮУрГУ»

Характеристика	Значение
Число вычислительных узлов/процессоров/сопроцессоров	480/960/384
Тип процессора	Intel Xeon X5680 (Gulftown, 6 ядер по 3.33 GHz) — 960 шт.
Тип сопроцессора	Intel Xeon Phi SE10X (61 ядро по 1.1 GHz) — 384 шт.
Оперативная память	16.9 TB
Тип системной сети	InfiniBand QDR (40 Gbit/s)
Операционная система	Linux CentOS 6.2

На рис. 3 приведены результаты экспериментов по выполнению естественного соединения отношений R и S на суперкомпьютере «Торнадо ЮУрГУ».

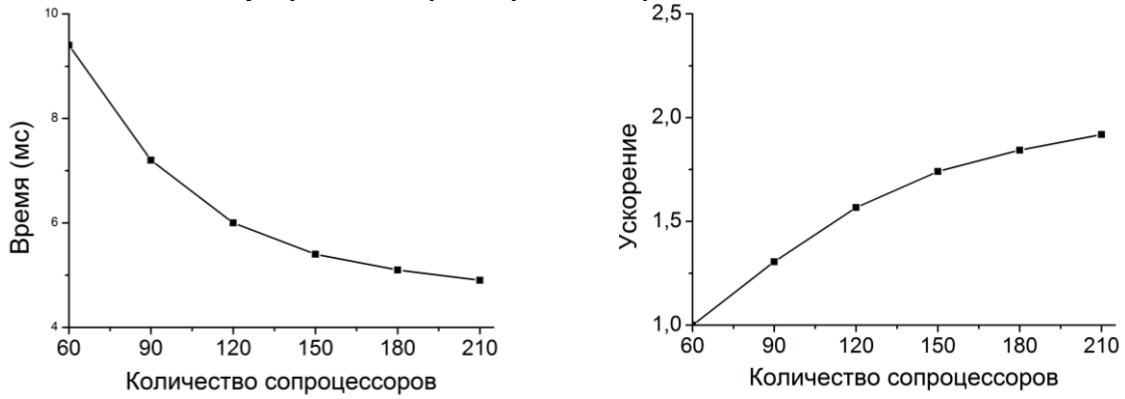

Рис. 3. Выполнение запроса на суперкомпьютере «Торнадо ЮУрГУ»

Эксперименты выполнялись с помощью колоночного сопроцессора. Эксперименты проводились с различным количеством узлов, оснащенных сопроцессорами Intel Xeon Phi: 60, 90, 120, 150, 180, 210 узлов.

Далее было произведено моделирование данного соединения с помощью разработанного эмулятора на *DCH*-дереве, описывающем архитектуру суперкомпьютера «Торнадо-ЮУрГУ». Результаты данного эксперимента приведены на рис. 4.

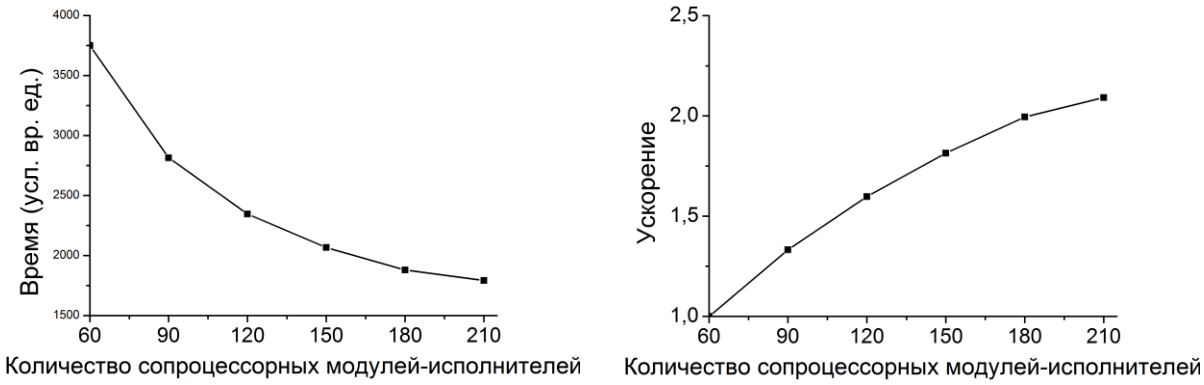

Рис. 4. Выполнение запроса на эмуляторе

Сравнение графиков на рис. 3 и рис. 4 показывает, что эмулятор адекватно моделирует выполнение запроса. Это подтверждает адекватность модели *DCH*.

Заключение

В статье рассмотрены вопросы моделирования аппаратной платформы колоночного сопроцессора баз данных. Выполнена разработка модели колоночного сопроцессора баз данных. Разработанная модель реализована в виде программного эмулятора, и с использованием разработанного эмулятора выполнены вычислительные эксперименты на проверку адекватности модели путем сравнения с прототипом колоночного сопроцессора.

Дальнейшим направлением исследований будет расширение модели для поддержки моделирования сжатия при обработке баз данных с использованием распределенных колоночных индексов.

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 // 13th International Conference on Parallel Computing Technologies, PaCT 2015, Petrozavodsk, Russian Federation, 31 August 2015 - 4 September 2015, Proceedings. Lecture Notes in Computer Science, Springer, 2015. Vol. 9251. P. 319–323.
2. Besedin K.Y., Kostenetskiy P.S., Prikazchikov S.O. Using Data Compression for Increasing Efficiency of Data Transfer Between Main Memory and Intel Xeon Phi Coprocessor or NVidia GPU in Parallel DBMS // 4th International Young Scientist Conference on Computational Science, Proceedings. – Procedia Computer Science, 2015. Vol. 66. P. 635–641.
3. Ivanova E.V., Prikazchikov S.O., Sokolinsky L.B. Join Execution Using Fragmented Columnar Indices on GPU and MIC // Proceedings of the 1st Ural Workshop on Parallel, Distributed, and Cloud Computing for Young Scientists. CEUR Workshop Proceedings, 2015. P. 1–10.
4. Беседин К.Ю., Костенецкий П.С. Применение сжатия информации при использовании многоядерных ускорителей для обработки баз данных / Параллельные вычислительные технологии (ПаВТ'2015): труды международной научной конференции (30 марта – 3 апреля 2015 г., г. Екатеринбург). Челябинск: Издательский центр ЮУрГУ, 2015. С. 43–53.
5. Иванова Е.В., Соколинский Л.Б. Декомпозиция операций пересечения и соединения на основе доменно-интервальной фрагментации колоночных индексов // Вестник Южно-

Уральского государственного университета. Серия: Вычислительная математика и информатика. 2015. Т. 4. – № 1. С. 44–56.
6. Иванова Е.В., Соколинский Л.Б. Использование распределенных колоночных индексов для выполнения запросов к сверхбольшим базам данных // Параллельные вычислительные технологии (ПаВТ'2014): труды международной научной конференции (1–3 апреля 2014 г., г. Ростов-на-Дону). Челябинск: Издательский центр ЮУрГУ, 2014. С. 270–275.
7. Иванова Е.В. Использование распределенных колоночных хеш-индексов для обработки запросов к сверхбольшим базам данных // Научный сервис в сети Интернет: многообразие суперкомпьютерных миров: Труды Международной суперкомпьютерной конференции (22-27 сентября 2014 г., Новороссийск). М.: Изд-во МГУ, 2014. С. 102–104.
8. Иванова Е.В. Исследование эффективности использования фрагментированных колоночных индексов при выполнении операции естественного соединения с использованием многоядерных ускорителей // Параллельные вычислительные технологии (ПаВТ'2015): труды международной научной конференции (30 марта – 3 апреля 2015 г., г. Екатеринбург). Челябинск: Издательский центр ЮУрГУ, 2015. С. 393–398.
9. Костенецкий П.С., Беседин К.Ю. Исследование эффективности различных методов сжатия при передаче данных из основной памяти в память сопроцессора Intel Xeon Phi // Вычислительные методы и программирование. 2014. Т. 15. № 4. С. 593–601.
10. Костенецкий П.С. Моделирование операции соединения, выполняемой на мультипроцессоре баз данных, оснащенном многоядерными сопроцессорами // Параллельные вычислительные технологии (ПаВТ’2015): труды международной научной конференции (30 марта – 3 апреля 2015 г., г. Екатеринбург). Челябинск: Издательский центр ЮУрГУ, 2015.
С. 420–425.
11. Костенецкий П.С. Моделирование параллельных систем баз данных для вычислительных кластеров // Научный сервис в сети Интернет: масштабируемость, параллельность, эффективность: Труды всероссийской научной конференции (21-26 сентября 2009 г., г. Новороссийск). – М.: Изд-во МГУ, 2009. – С. 300–304.
12. 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.
13. McColl W.F. General purpose parallel computing. Lectures on Parallel Computation. Lectures on parallel computation. USA: Cambridge University Press, 1993. 496 p.
14. Осипова А.М., Костенецкий П.С. Моделирование мультипроцессоров систем баз данных с многоядерными ускорителями // Научный сервис в сети Интернет: все грани параллелизма: Труды Международной суперкомпьютерной конференции (23-28 сентября 2013 г., Новороссийск). М.: Издательство МГУ, 2013. С. 194.
15. Valiant L.G. A bridging model for parallel computation // Communication of the ACM. USA: ACM, 1990. Vol. 33. No. 8. P. 103–111.

The model of column coprocessor development

S.O. Prikazchikov, P.S. Kostenetskiy

South-Ural State University

The paper proposes a mathematic model that allowing exploration of effectiveness of different hardware cluster computing systems configurations based on multi-core coprocessors while processing databases using approach of distributed columnar indices.

Keywords: distributed column indices, column coprocessor, parallel databases.

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 // 13th International Conference on Parallel Computing Technologies, PaCT 2015, Petrozavodsk, Russian Federation, 31 August 2015 - 4 September 2015, Proceedings. Lecture Notes in Computer Science, Springer, 2015. Vol. 9251. P. 319–323.
2. Besedin K.Y., Kostenetskiy P.S., Prikazchikov S.O. Using Data Compression for Increasing Efficiency of Data Transfer Between Main Memory and Intel Xeon Phi Coprocessor or NVidia GPU in Parallel DBMS // 4th International Young Scientist Conference on Computational Science, Proceedings. – Procedia Computer Science, 2015. Vol. 66. P. 635–641.
3. Ivanova E.V., Prikazchikov S.O., Sokolinsky L.B. Join Execution Using Fragmented Columnar Indices on GPU and MIC // Proceedings of the 1st Ural Workshop on Parallel, Distributed, and Cloud Computing for Young Scientists. CEUR Workshop Proceedings, 2015. P. 1–10.
4. Besedin K.Y., Kostenetskiy P.S. Primenenie szhatiya informatsii pri ispol'zovanii mnogoyadernykh uskoriteley dlya obrabotki baz dannykh [Using data compression for database processing when using multicore coprocessors] / Parallel'nye vychislitel'nye tekhnologii (PaVT'2015): trudy mezhdunarodnoy nauchnoy konferentsii (30 marta – 3 aprelya 2015 g., g. Ekaterinburg) [Parallel Computational Technologies (PCT'2015): Proceedings of the International Scientific Conference (Ekaterinburg, Russia, March, 30 – April, 3, 2015)]. Chelyabinsk, Publishing of the South Ural State University, 2015. P. 43–53.
5. Ivanova E.V., Sokolinskiy L.B. Dekompozitsiya operatsiy peresecheniya i soedineniya na osnove domenno-interval'noy fragmentatsii kolonochnykh indeksov [Join and intersection decomposition based on fragmented column indices] // Vestnik Yuzhno-Ural'skogo gosudarstvennogo universiteta. Seriya: Vychislitel'naya matematika i informatika [Bulletin of South Ural State University. Series: Computational mathemetic and informatic]. 2015. Vol. 4. – № 1. P. 44–56.
6. Ivanova E.V., Sokolinskiy L.B. Ispol'zovanie raspredelennykh kolonochnykh indeksov dlya vypolneniya zaprosov k sverkhbol'shim bazam dannykh [Using distributed column indexes for query execution over very large databases] // Parallel'nye vychislitel'nye tekhnologii (PaVT'2014): trudy mezhdunarodnoy nauchnoy konferentsii (1–3 aprelya 2014 g., g. Rostov-na-Donu) [Parallel Computational Technologies (PCT'2014): Proceedings of the International Scientific Conference (Rostov-on-Don, Russia, April, 1–3, 2014)]. Chelyabinsk, Publishing of the South Ural State University, 2014. P. 270–275.
7. Ivanova E.V. Ispol'zovanie raspredelennykh kolonochnykh khesh-indeksov dlya obrabotki zaprosov k sverkhbol'shim bazam dannykh [Using of distributed column hash-indices for query execution over very large databases] // Nauchnyy servis v seti Internet: mnogoobrazie superkomp'yuternykh mirov: Trudy Mezhdunarodnoy superkomp'yuternoy konferentsii (22–27 sentyabrya 2014 g., Novorossiysk) [Scientific service on the Internet: the variety of supercomputing worlds: Proceedings of the International Scientific Conference (Novorossiysk, Russia, September 22–27, 2014]. Moscow, Publishing of MSU, 2014. P. 102–104.

8. Ivanova E.V. Issledovanie effektivnosti ispol'zovaniya fragmentirovannykh kolonochnykh indeksov pri vypolnenii operatsii estestvennogo soedineniya s ispol'zovaniem mnogoyadernykh uskoriteley [Analysis of efficience of using of fragmented column indices when operation of natural join using multicore coprocessors] // Parallel'nye vychislitel'nye tekhnologii (PaVT'2015): trudy mezhdunarodnoy nauchnoy konferentsii (30 marta – 3 aprelya 2015 g., g. Ekaterinburg) [Parallel Computational Technologies (PCT'2015): Proceedings of the International Scientific Conference (Rostov-on-Don, Russia, March, 30– April, 3, 2015)]. Chelyabinsk, Publishing of the South Ural State University, 2015. P. 393–398.
9. Kostenetskiy P.S., Besedin K.Y. Issledovanie effektivnosti razlichnykh metodov szhatiya pri peredache dannykh iz osnovnoy pamyati v pamyat' soprotsessora Intel Xeon Phi [Analysis of various methods of compression with Data Transfer Between Main Memory and Intel Xeon Phi Coprocessor] // Vychislitel'nye metody i programmirovanie [Computational methods and programming]. 2014. Vol. 15. № 4. P. 593–601.
10. Kostenetskiy P.S. Modelirovanie operatsii soedineniya, vypolnyaemoy na mul'tiprotsessore baz dannykh, osnashchennom mnogoyadernymi soprotsessorami [Modelling of join operation which is executed on database multiprocessor which have multicore coprocessors]// Parallel'nye vychislitel'nye tekhnologii (PaVT'2015): trudy mezhdunarodnoy nauchnoy konferentsii (30 marta – 3 aprelya 2015 g., g. Ekaterinburg) [Parallel Computational Technologies (PCT'2015): Proceedings of the International Scientific Conference (Ekaterinburg, Russia, March, 30 – April, 3, 2015)]. Chelyabinsk, Publishing of the South Ural State University, 2015.P. 420–425.
11. Kostenetskiy P.S. Modelirovanie parallel'nyh sistem baz dannyh dlja vychislitel'nyh klasterov [Modeling of parallel databases systems for computational clusters] // Nauchnyj servis v seti Internet: masshtabiruemost', parallel'nost', jeffektivnost': Trudy vserossijskoj nauchnoj konferencii (21–26 sentyabrya 2009 g., Novorossiysk) [Scientific service on the Internet: scalability, parallelism, efficiency: Proceedings of the Russian Scientific Conference (Novorossiysk, Russia, September 21–26, 2009]. Moscow, Publishing of MSU, 2009. P. 300–304.
12. 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. V. 1576. P. 561-573.
13. McColl W.F. General purpose parallel computing. Lectures on Parallel Computation. Lectures on parallel computation. USA: Cambridge University Press, 1993. 496 p.
14. Osipova A.M., Kosteneckij P.S. Modelirovanie mul'tiprocessorov sistem baz dan-nyh s mnogojadernymi uskoriteljami // Nauchnyj servis v seti Internet: vse grani parallelizma: Trudy Mezhdunarodnoj superkomp'juternoj konferencii (23-28 sentjabrja 2013 g., Novorossijsk) [Scientific service on the Internet: all sides of parallelism: Proceedings of the international Supercomputer Conference]. Moscow, Publishing of MSU, 2013. P. 194.
15. Valiant L.G. A bridging model for parallel computation // Communication of the ACM. USA: ACM, 1990. Vol. 33. No. 8. P. 103–111.