Методы параллельной обработки сверхбольших баз данных с использованием распределенных колоночных индексов

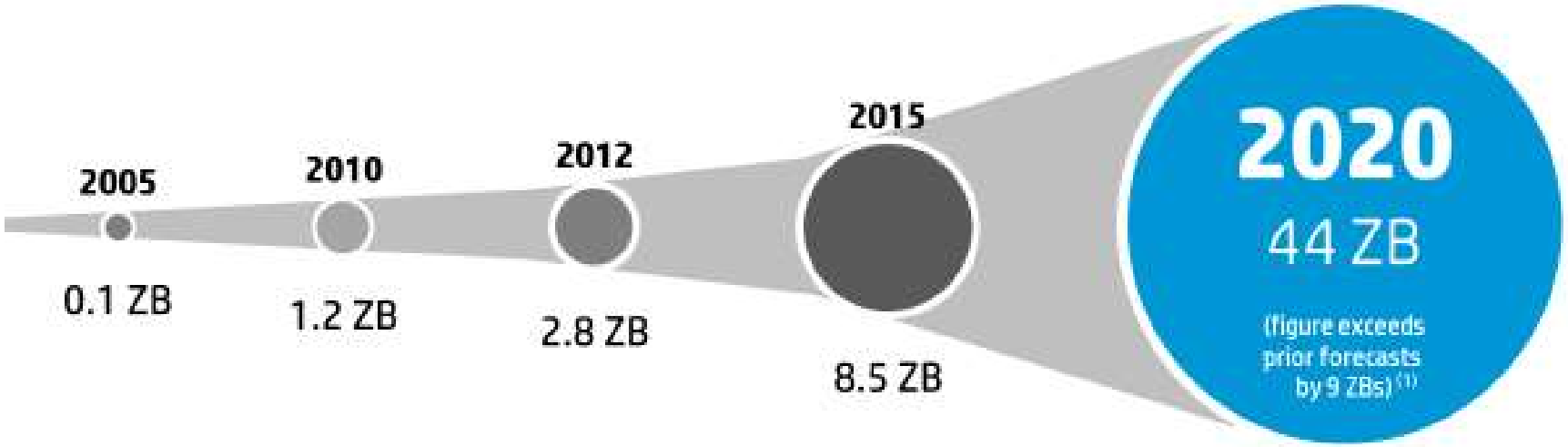
05.13.11 - математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей

Диссертация на соискание ученой степени кандидата физико-математических наук

Елена Владимировна Иванова

Научный руководитель:
СОКОЛИНСКИЙ Леонид Борисович,
доктор физ.-мат. наук, профессор

Проблема больших данных



- Объем хранимой информации удваивается каждые два года
- из всего объема существующих данных потенциально полезны 22%, из которых менее 5% были подвергнуты анализу

Тенденции в развитии технологий обработки больших данных

- Для обработки больших данных необходимо использовать СУБД
  (Майкл Стоунбрейкер)
- Кластерные вычислительные системы с большой суммарной оперативной памятью
- Базы данных в оперативной памяти
- Многоядерные ускорители
- Колоночное представление данных со сжатием

Цель диссертационной работы

Разработка и исследование эффективных методов параллельной обработки сверхбольших баз данных с использованием колоночного представления информации, ориентированных на кластерные вычислительные системы, оснащенные многоядерными ускорителями, и допускающих интеграцию с реляционными СУБД

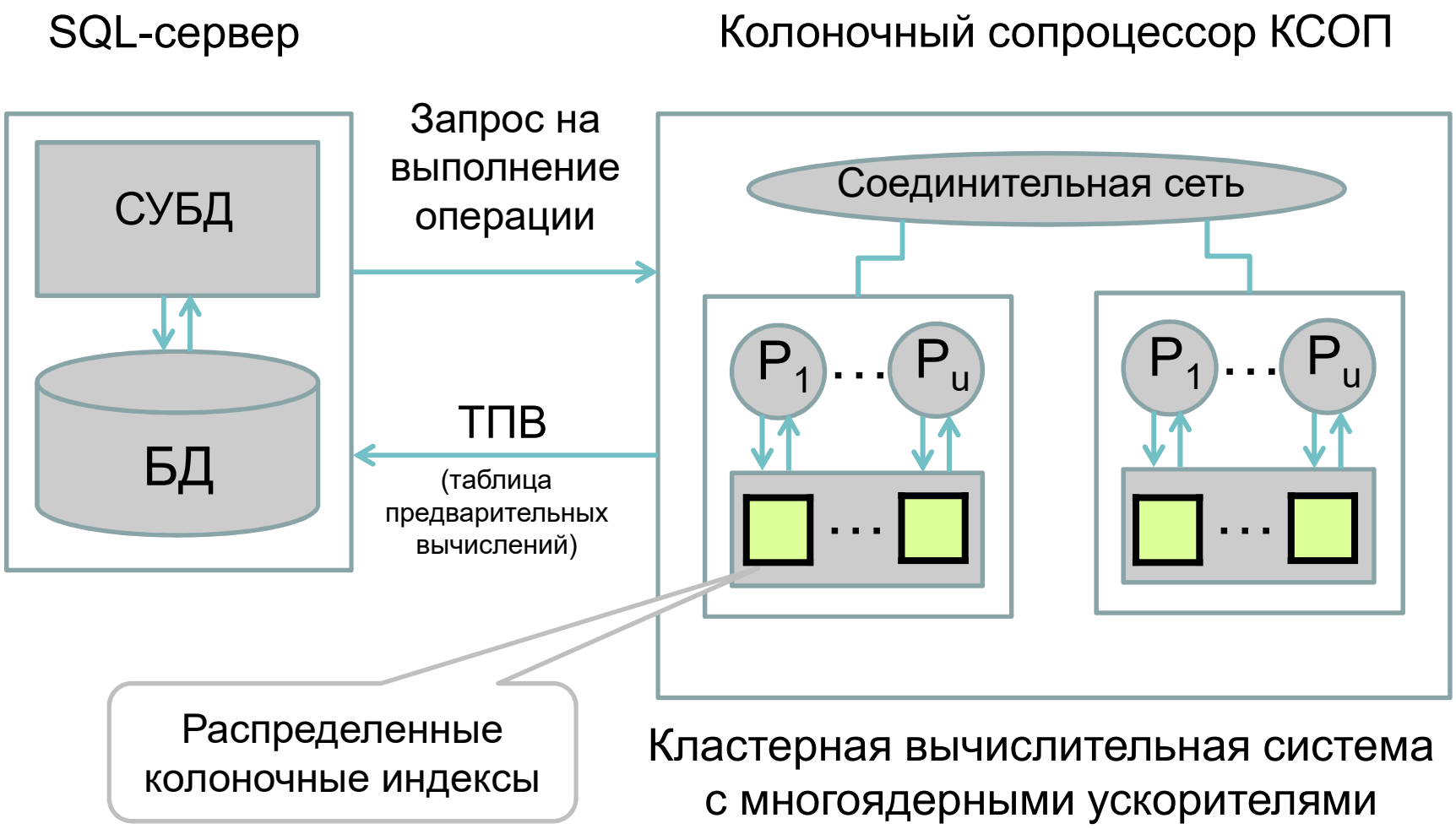
Основные задачи

1. Разработать колоночные индексы и методы их фрагментации
2. Разработать методы декомпозиции реляционных операций для распределенных колоночных индексов
3. Реализовать предложенные подходы и методы в виде колоночного сопроцессора КСОП для кластерных вычислительных систем
4. Провести вычислительные эксперименты с использованием КСОП

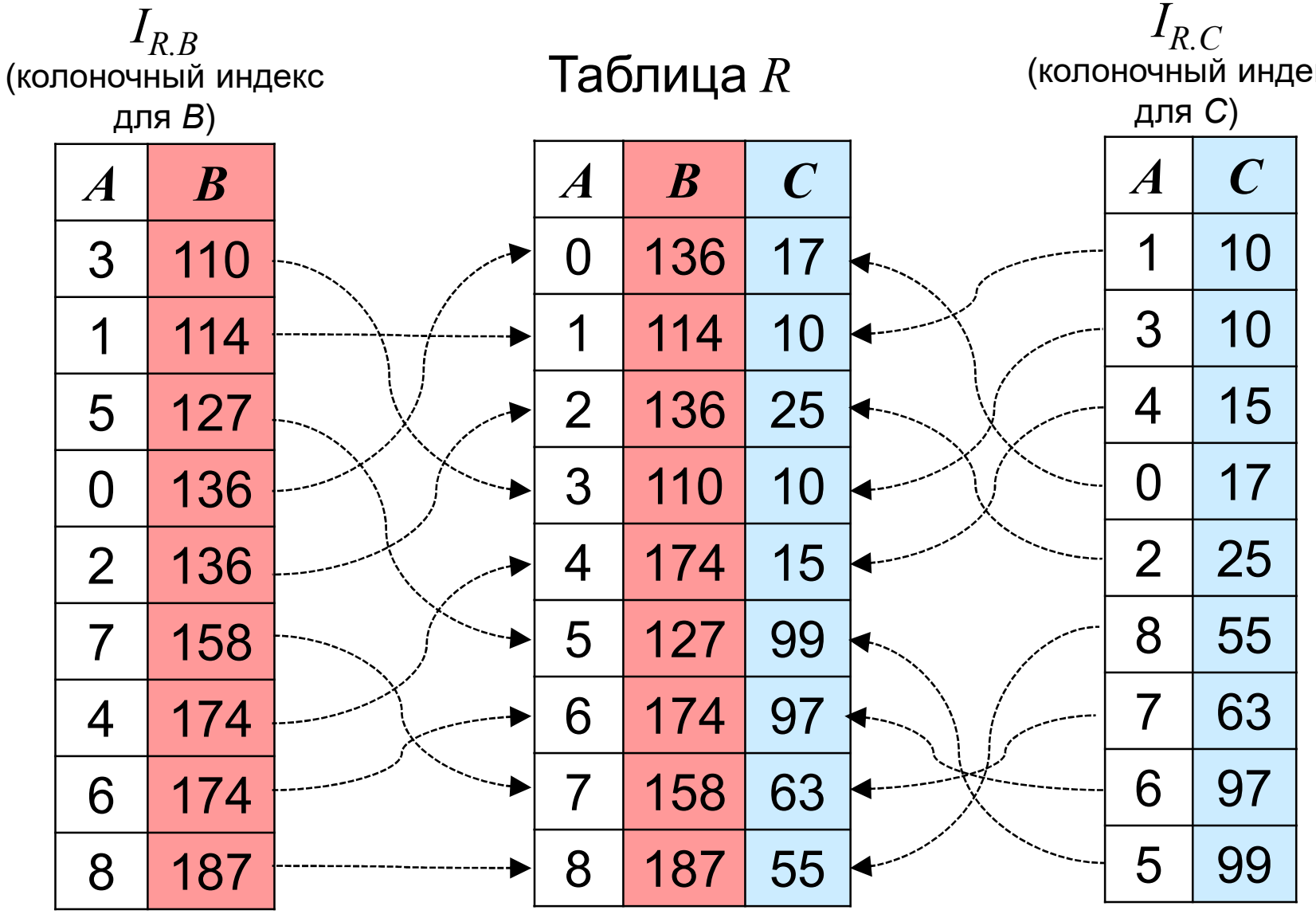
Работы по теме диссертации

1	Ramamurthy R., Dewitt D., Su Q. A case for fractured mirrors // Proceedings of the VLDB Endowment. 2002. Vol. 12, No. 2. P. 89-101.	«Разбитое зеркало»
2	Abadi D.J., Madden S.R., Hachem N. Column-Stores vs. Row-Stores: How Different Are They Really? // Proceedings of the 2008 ACM SIGMOD international conference on Management of data, June 9-12, 2008, Vancouver, BC, Canada. ACM, 2008. P. 967-980.	Эмуляция колоночного представления в строчной СУБД
3	Bruno N. Teaching an Old Elephant New Tricks // Online Proceedings of Fourth Biennial Conference on Innovative Data Systems Research (CIDR 2009), Asilomar, CA, USA, January 4-7, 2009.	C-таблицы
4	El-Helw A., Ross K.A., Bhattacharjee B., Lang C.A., Mihaila G.A. Column-oriented query processing for row stores // Proceedings of the ACM 14th international workshop on Data Warehousing and OLAP (DOLAP '11), October 28, 2011, Glasgow, United Kingdom. ACM, 2011. P. 67-74.	Только индексные планы
5	Larson P.-A., Clinciu C., Hanson E. N., Oks A., Price S. L., Rangarajan S., Surna A., Zhou Q. SQL server column store indexes // Proceedings of the 2011 ACM SIGMOD International Conference on Management of data (SIGMOD '11), June 12-16, 2011, Athens, Greece. ACM, 2011. P. 1177-1184.	Индексы колоночной памяти (column store indexes)
6	Jha S., He B., Lu M., Cheng X., Huynh H. P. Improving main memory hash joins on Intel Xeon Phi processors: an experimental approach // Proceedings of the VLDB Endowment. 2015. Vol. 8, No. 6. P. 642-653.	Использование Xeon Phi

Предлагаемое решение



Колоночный индекс



Формальное определение колоночного индекса

Пусть $R(A, B, \ldots)$ – отношение R с суррогатным ключом A и атрибутом B. A состоит из целочисленных неотрицательных элементов.

$\mathfrak{D}_B$ – *домен атрибута* B. На множестве $\mathfrak{D}_B$ задано отношение линейного порядка. $T(R) = n$ – количество элементов в R.

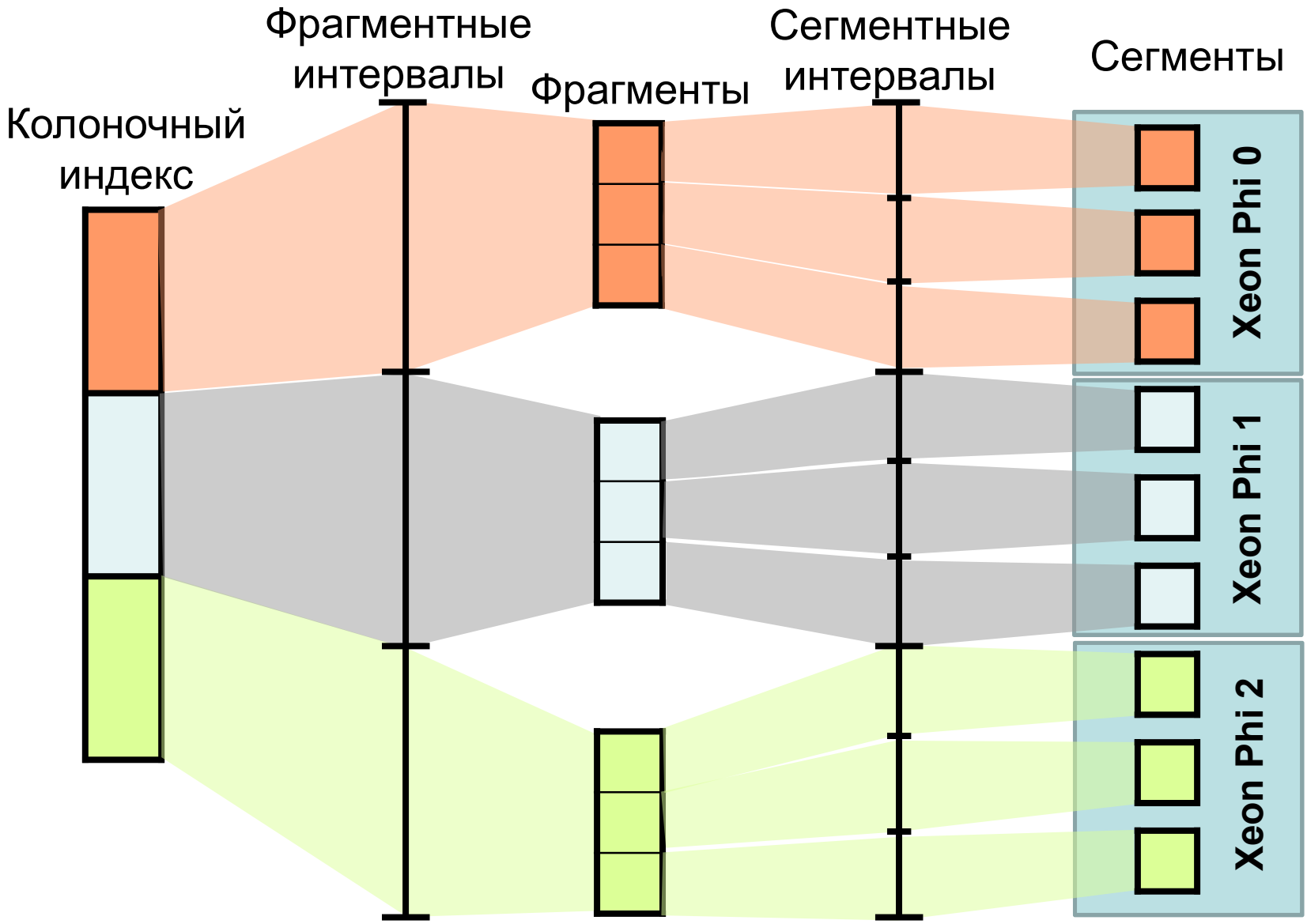
Колоночным индексом $I_{R.B}$ атрибута B отношения R называется упорядоченное отношение, удовлетворяющее следующим требованиям:

$$T(I_{R.B}) = n \quad \text{и} \quad \pi_A\left(I_{R.B}\right) = \pi_A\left(R\right); \qquad (1)$$

$$\forall x_1, x_2 \in I_{R.B}\left(x_1 \le x_2 \Leftrightarrow x_1.B \le x_2.B\right); \qquad (2)$$

$$\forall r \in R\left(\forall x \in I_{R.B}\left(r.A = x.A \Rightarrow r.B = x.B\right)\right). \qquad (3)$$

Распределение данных



Формальное определение доменно-интервальной фрагментации

Разбиение домена $\mathfrak{D}_B$ на k непересекающихся интервалов:

$$V_0 = [v_0; v_1);\ V_1 = [v_1; v_2);\ \ldots;\ V_{k-1} = [v_{k-1}; v_k);$$

$$v_0 < v_1 < \cdots < v_k;$$

$$\mathfrak{D}_B = \bigcup_{i=0}^{k-1} V_i$$

Доменная функция фрагментации

$$\varphi_{\mathfrak{D}_B}: \mathfrak{D}_B \to \{0, \ldots, k-1\}$$

Доменно-интервальная функция фрагментации колоночного индекса: $\forall i \in \{0, \ldots, k-1\} \left(\forall b \in \mathfrak{D}_B \left(\varphi_{\mathfrak{D}_B}(b) = i \Leftrightarrow b \in V_i\right)\right)$

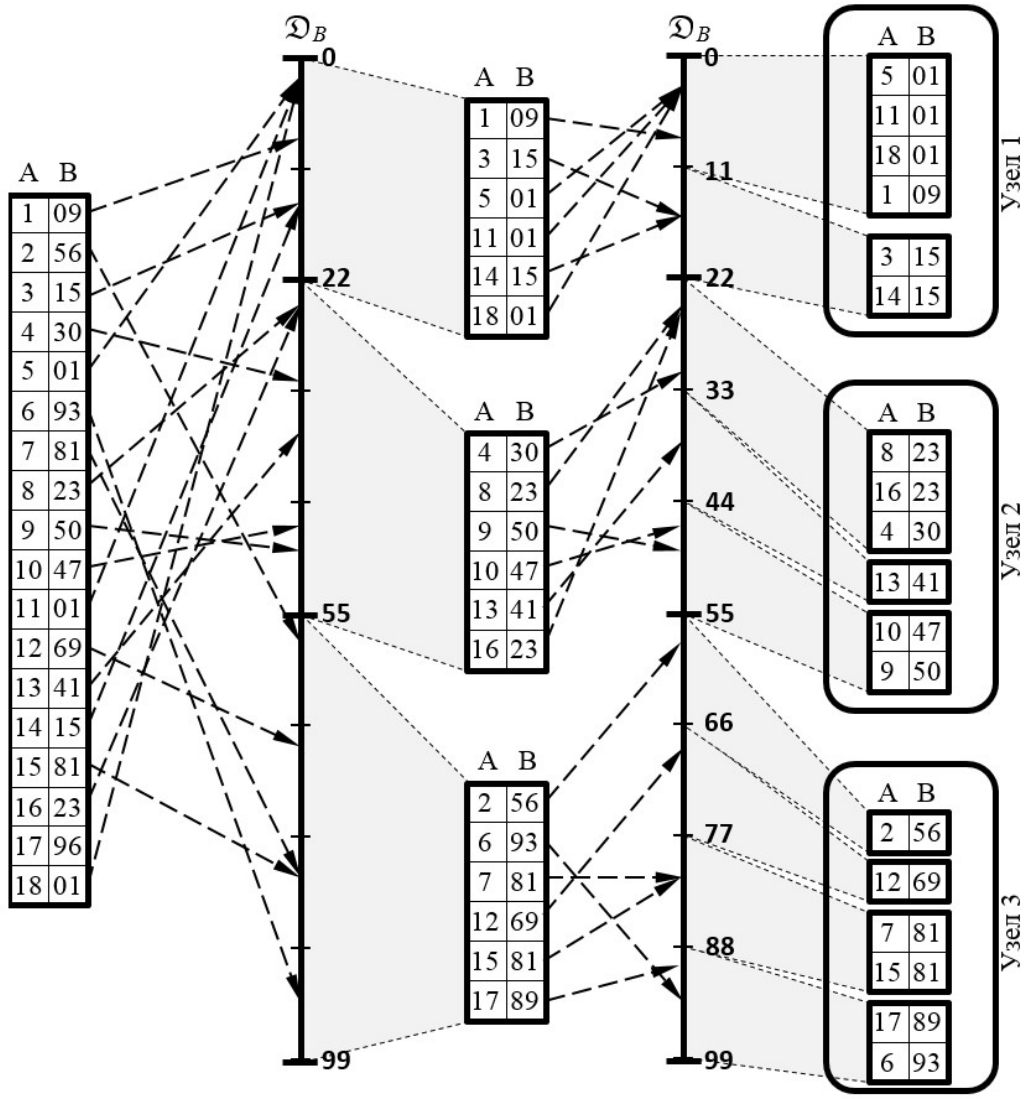
$$\varphi_{I_{R.B}}: I_{R.B} \to \{0, \ldots, k-1\}$$

$$\forall x \in I_{R.B} \left(\varphi_{I_{R.B}}(x) = \varphi_{\mathfrak{D}_B}(x.B)\right)$$

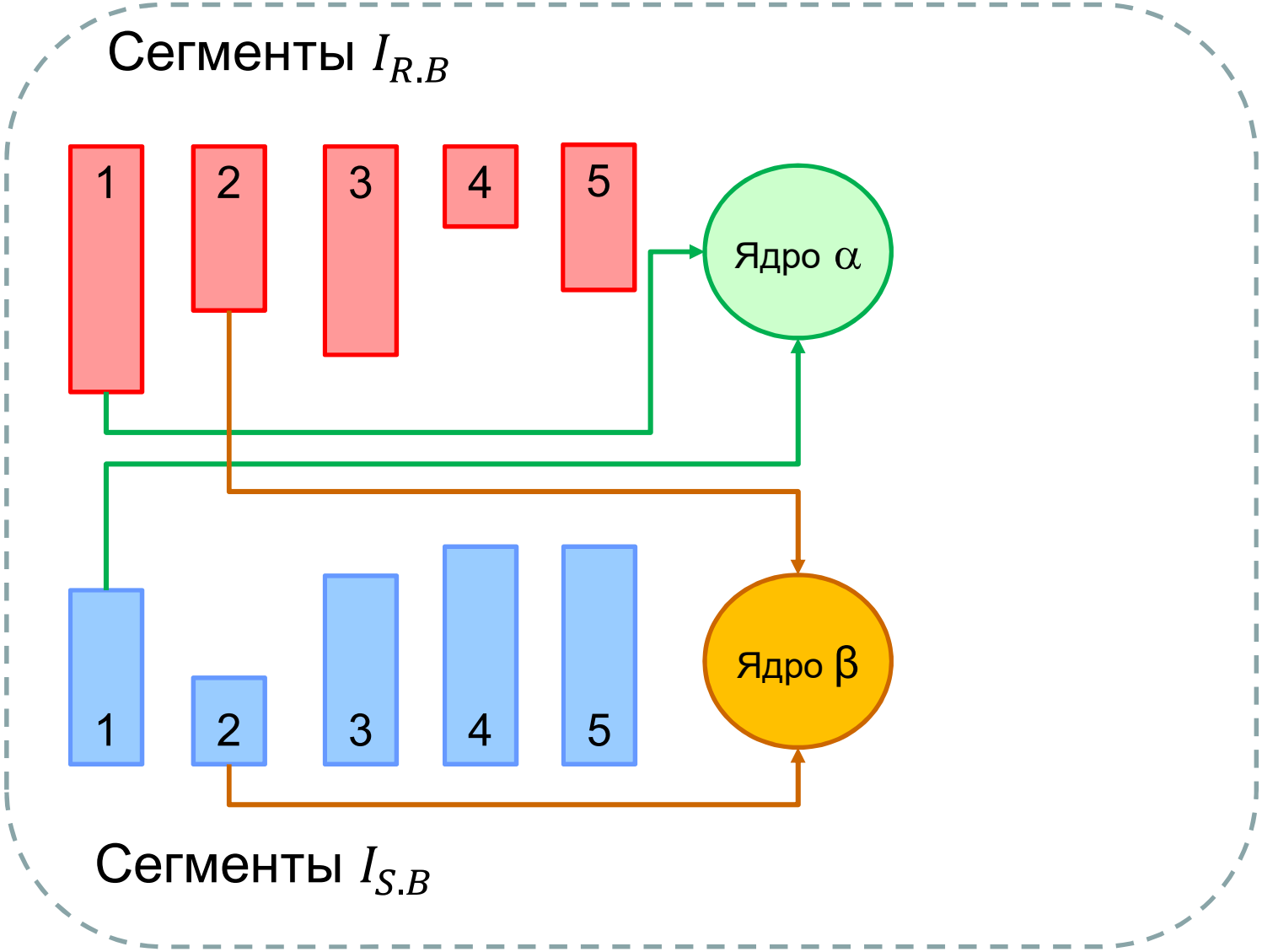
i-тый фрагмент колоночного индекса:

$$I^i_{R.B} = \left\{x \,|\, x \in I_{R.B};\ \varphi_{I_{R.B}}(x) = i\right\}$$

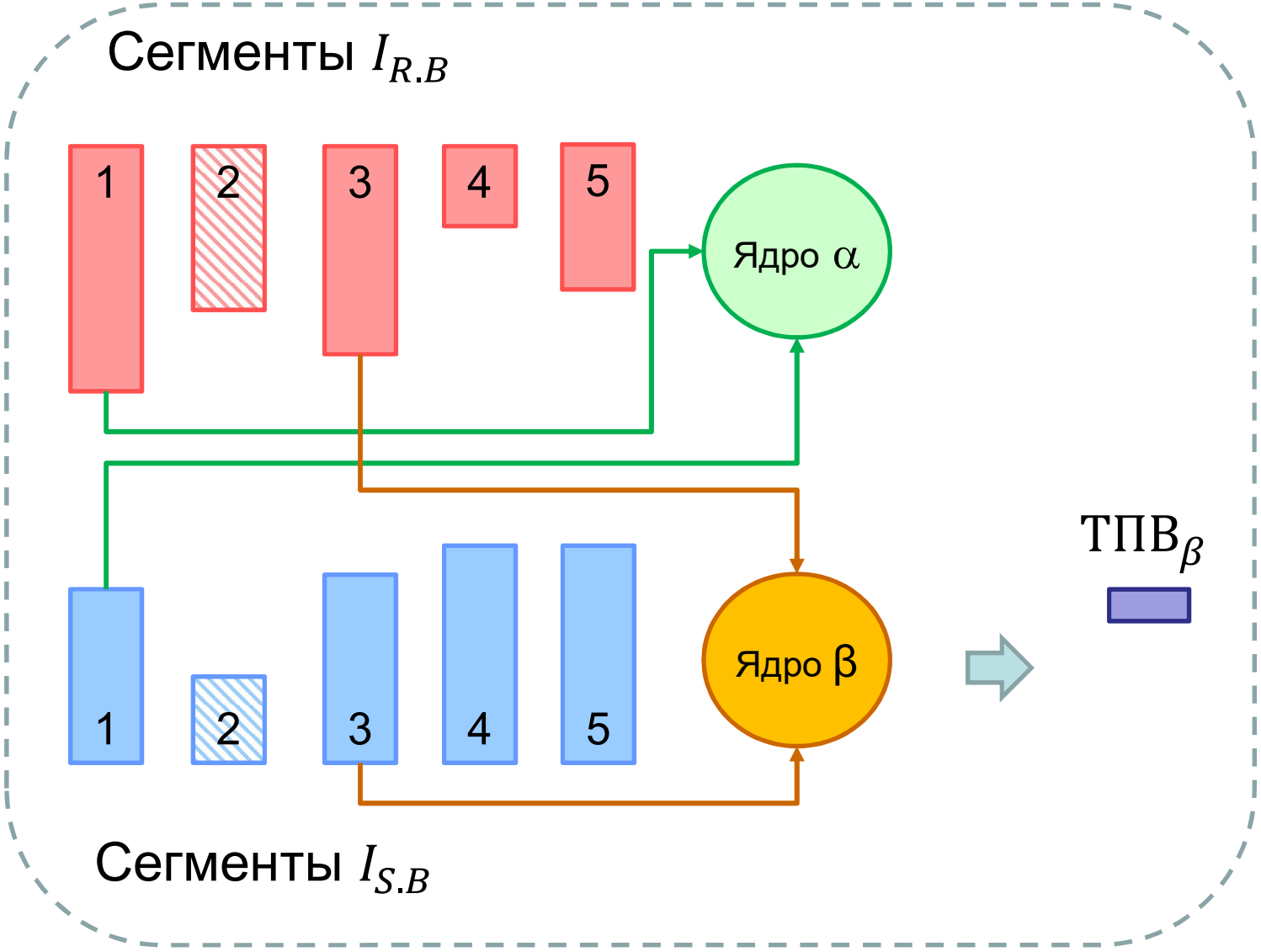
Пример двухуровневого разбиения колоночного индекса на фрагменты и сегменты



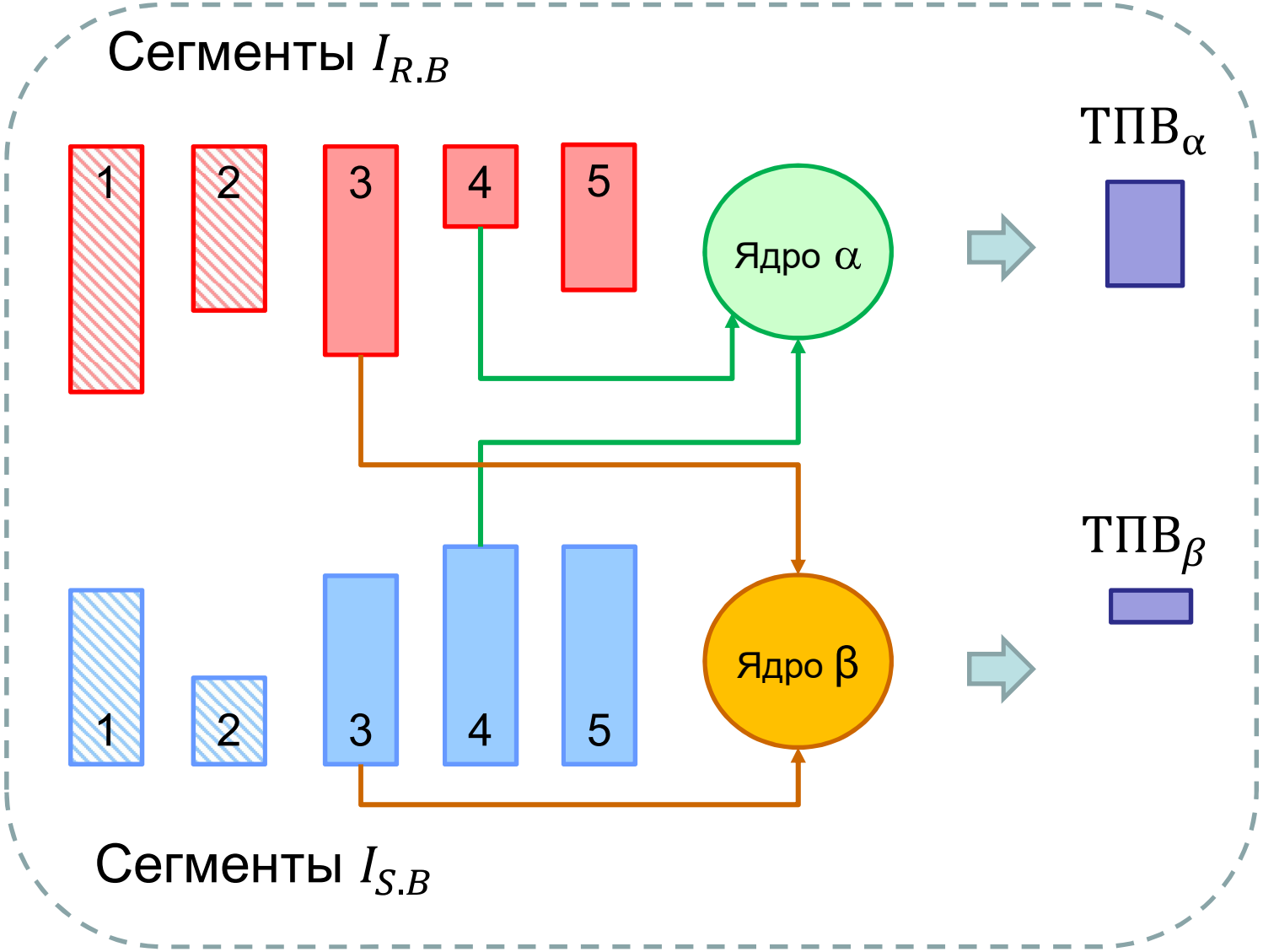
Балансировка загрузки ядер



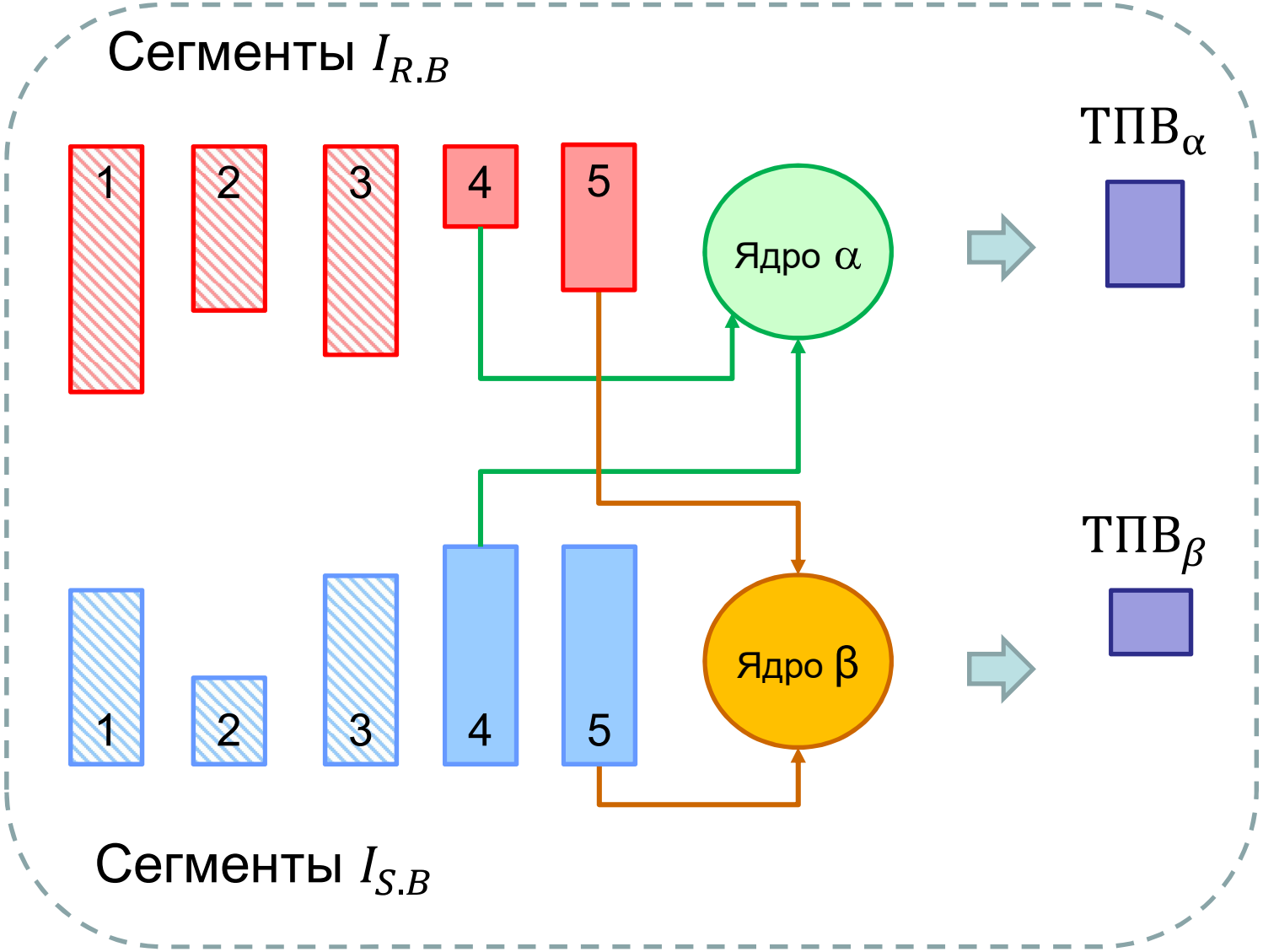
Балансировка загрузки ядер



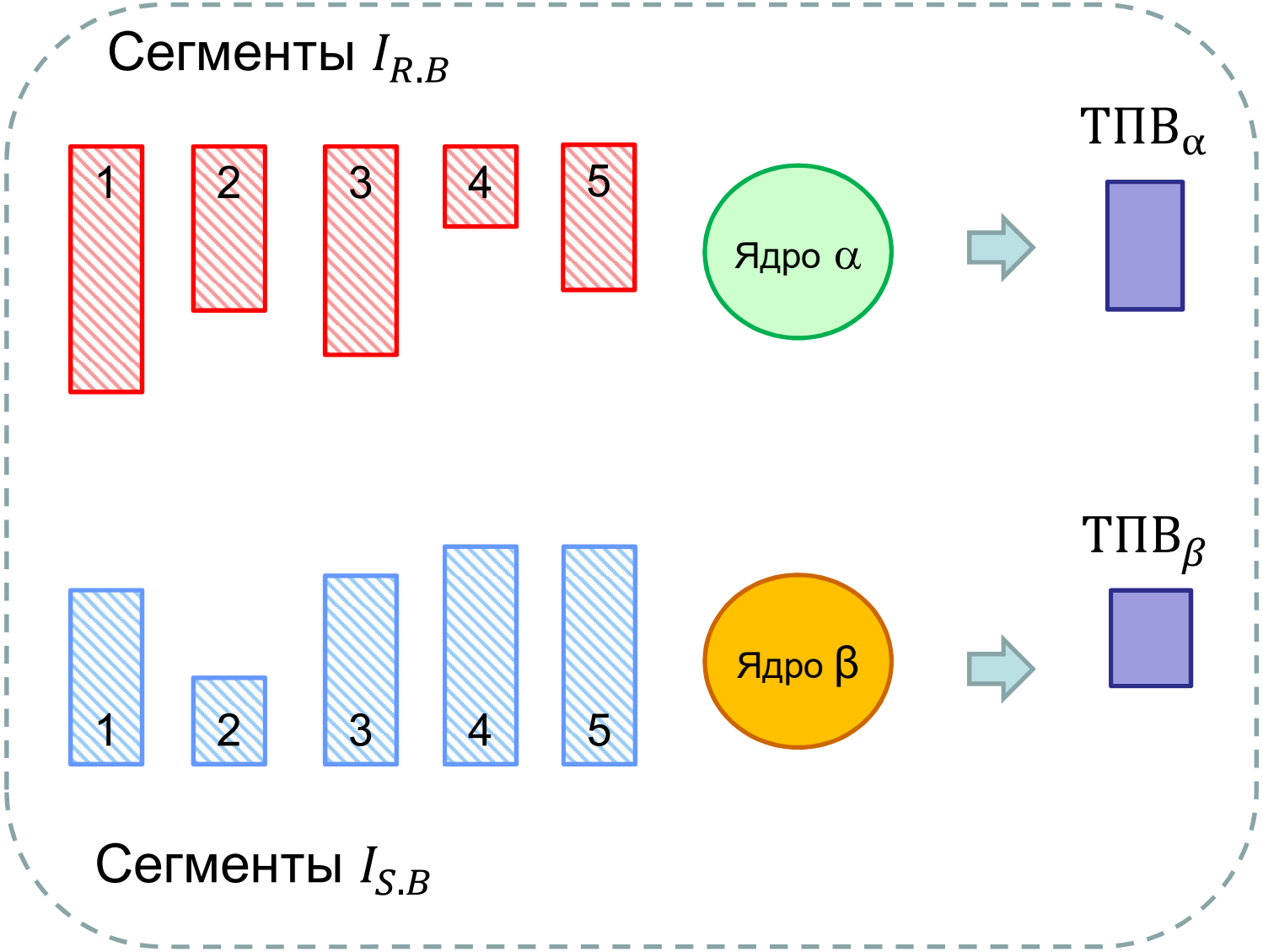
Балансировка загрузки ядер



Балансировка загрузки ядер



Балансировка загрузки ядер



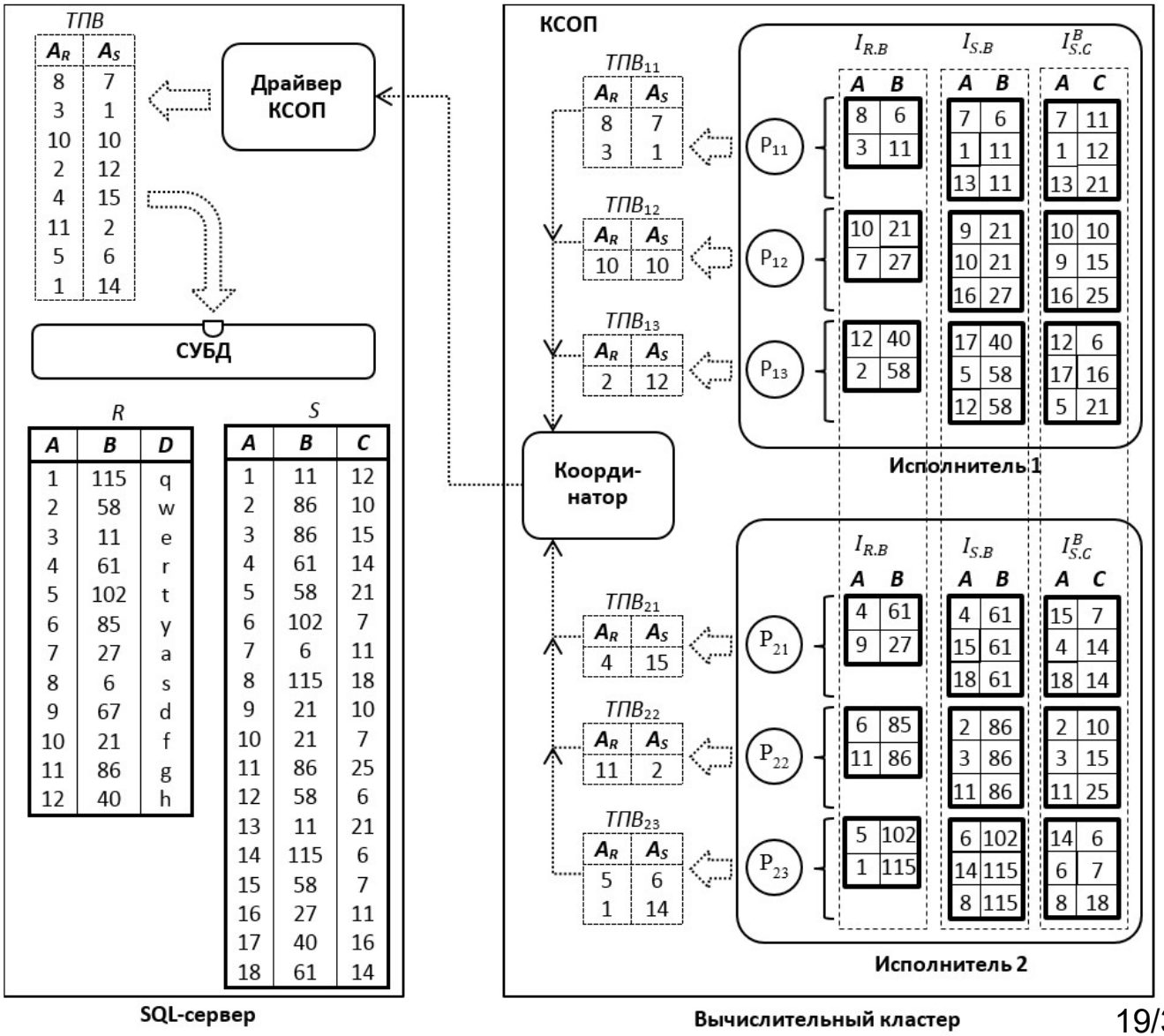
Параллельные алгоритмы выполнения реляционных операций

- Проекция
- Выбор
- Удаление дубликатов
- Пересечение
- Объединение
- Естественное соединение
- Группировка

Взаимодействие SQL-сервера с колоночным сопроцессором КСОП

```
SELECT D, C
FROM R, S
WHERE
    R.B = S.B
    AND C<13;
```

$$\pi_{I_{R.B}.A \to A_R, I_{S.B}.A \to A_S} \left(I_{R.B} \underset{I_{R.B}.B = I_{S.B}.B}{\bowtie} \left(I_{S.B} \bowtie \sigma_{C<13}\left(I^{B}_{S.C}\right)\right)\right)$$



Реконструкция результата на SQL-сервере

R

A	B	D
1	115	q
2	58	w
3	11	e
4	61	r
5	102	t
6	85	y
7	27	a
8	6	s
9	67	d
10	21	f
11	86	g
12	40	h

S

A	B	C
1	11	12
2	86	10
3	86	15
4	61	14
5	58	21
6	102	7
7	6	11
8	115	18
9	21	10
10	21	7
11	86	25
12	58	6
13	11	21
14	115	6
15	58	7
16	27	11
17	40	16
18	61	14

```
SELECT D, C
FROM
   R INNER JOIN (
      ТПВ INNER JOIN S ON (S.A = ТПВ.A_S)
   ) ON (R.A = ТПВ.A_R)
```

ТПВ

A_R	A_S
8	7
3	1
10	10
2	12
4	15
11	2
5	6
1	14

Результат

D	C
s	11
e	12
f	7
w	6
r	7
g	10
t	7
q	6

Формальное определение декомпозиции естественного соединения $Q = \pi_{*\setminus A}(R) \bowtie \pi_{*\setminus A}(S)$

Схема БД: $R(A, B, D)$; $S(A, B, C)$.

Колоночные индексы $I_{R.B}$ и $I_{S.B}$, для которых задана доменно-интервальная фрагментация степени k по атрибуту B:

$$I_{R.B} = \bigcup_{i=0}^{k-1} I_{R.B}^{i} \qquad I_{S.B} = \bigcup_{i=0}^{k-1} I_{S.B}^{i}$$

$$P^{i} = \pi_{I_{R.B}^{i}.A \to A_R,\, I_{S.B}^{i}.A \to A_S} \left(I_{R.B}^{i} \underset{I_{R.B}^{i}.B = I_{S.B}^{i}.B}{\bowtie} I_{S.B}^{i} \right)$$

$$P = \bigcup_{i=0}^{k-1} P^{i}$$

$$Q = \left\{ \left(\&_R(p.A_R).B, \&_S(p.A_S).C, \&_R(p.A_R).D \right) \middle| \, p \in P \right\}$$

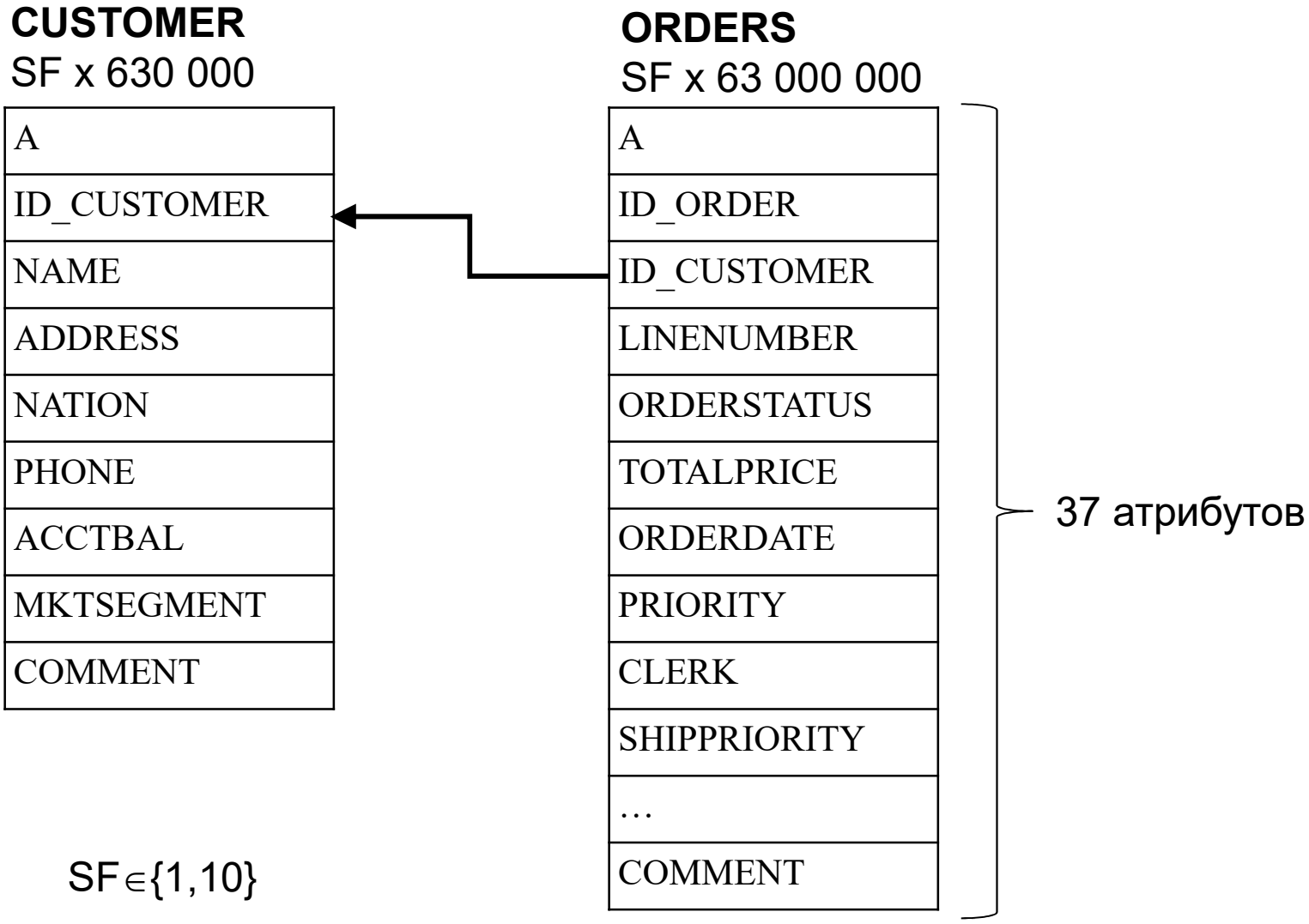
Реализация КСОП

- Язык программирования: Си
- Технологии параллельного программирования: MPI и OpenMP
- Исходные тексты приложения свободно доступны в сети Интернет: https://github.com/elena-ivanova/colomnindices/

Характеристики вычислительных систем

	«Торнадо ЮУрГУ»	**«RSC PetaStream»**
Количество узлов:	384	64
Тип процессоров:	2 x Intel Xeon X5680 (12 ядер по 3.33 ГГц; 2 потока на ядро)	
Оперативная память узла:	24 Гб	
Тип сопроцессора:	Intel Xeon Phi SE10X: (61 ядро по 1.1 ГГц; 4 потока на ядро)	Intel Xeon Phi 7120 (61 ядро по 1.24 ГГц)
Память сопроцессора:	8 Гб	16 Гб
Тип системной сети:	InfiniBand QDR	InfiniBand FDR
Тип управляющей сети:	Gigabit Ethernet	Gigabit Ethernet
Операционная система:	Linux CentOS 6.2	Linux CentOS 7.0

Тестовая база данных

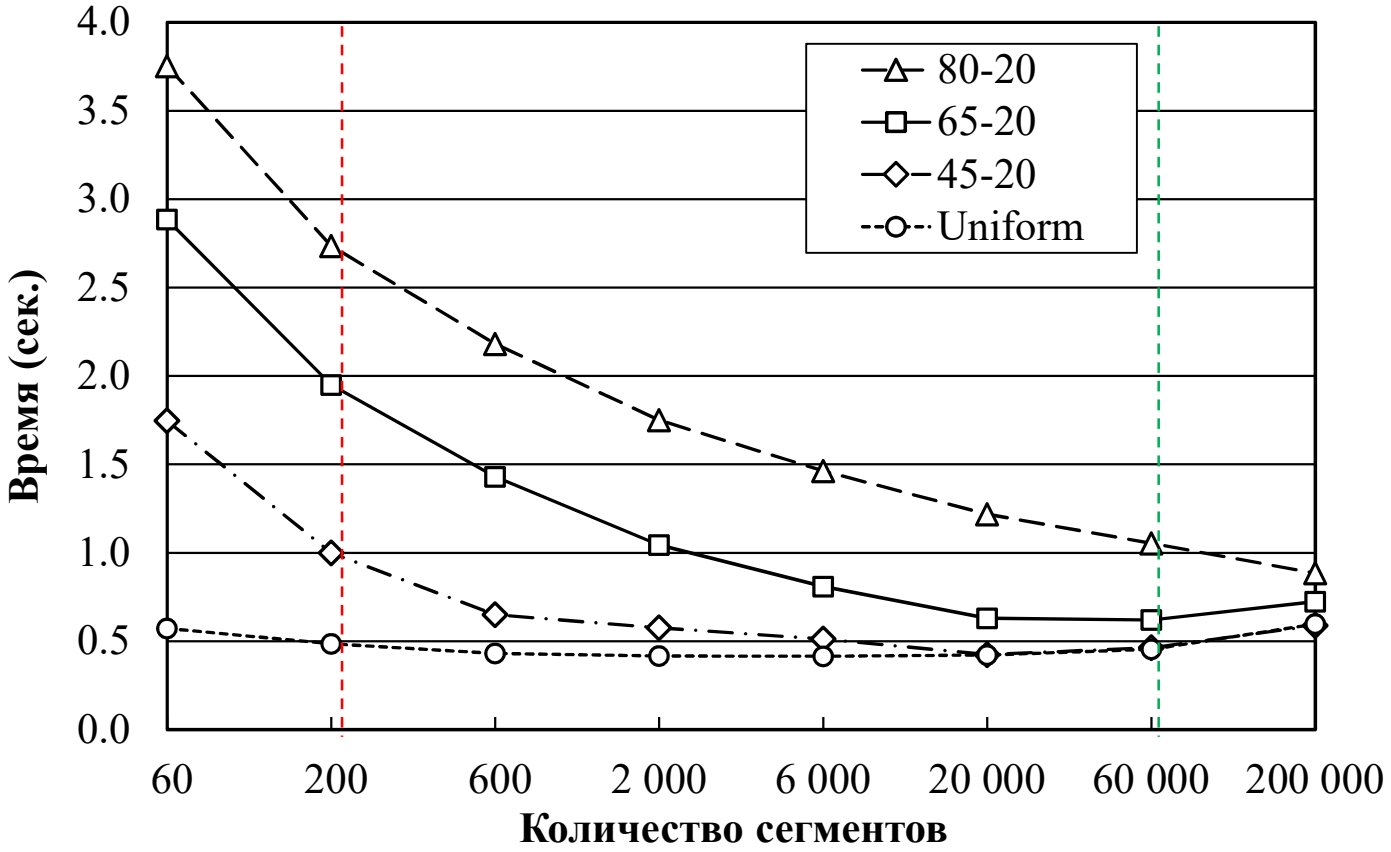


Тестовый запрос

Выдать заказы (с информацией о покупателе), общая стоимость которых не превышает величину Sel*100 000:

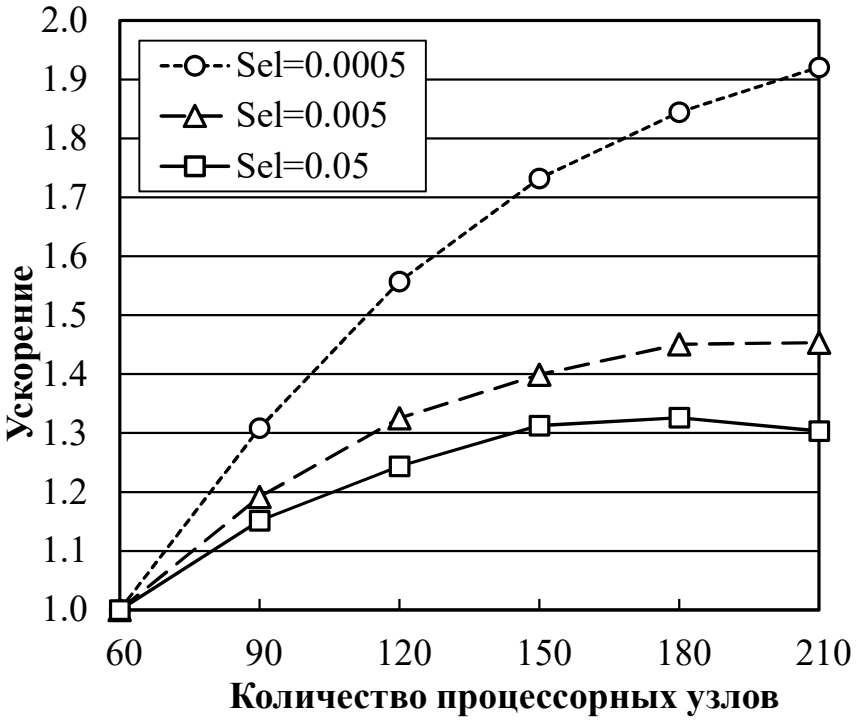
```
SELECT *
FROM CUSTOMER, ORDERS
WHERE (CUSTOMER.ID_CUSTOMER=ORDERS.ID_CUSTOMER)
AND (ORDERS.TOTALPRICE <= Sel*100 000).
```

Балансировка загрузки процессорных ядер Xeon Phi

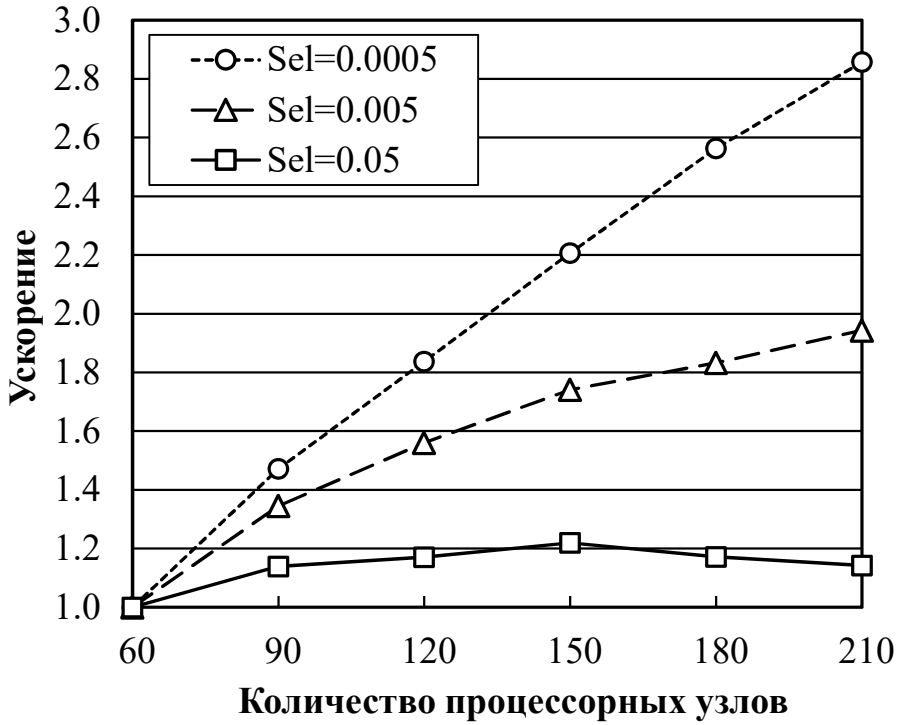


Размер хранилища: 63 млн. записей
Вычислительная система: «Торнадо ЮУрГУ»

Масштабируемость КСОП



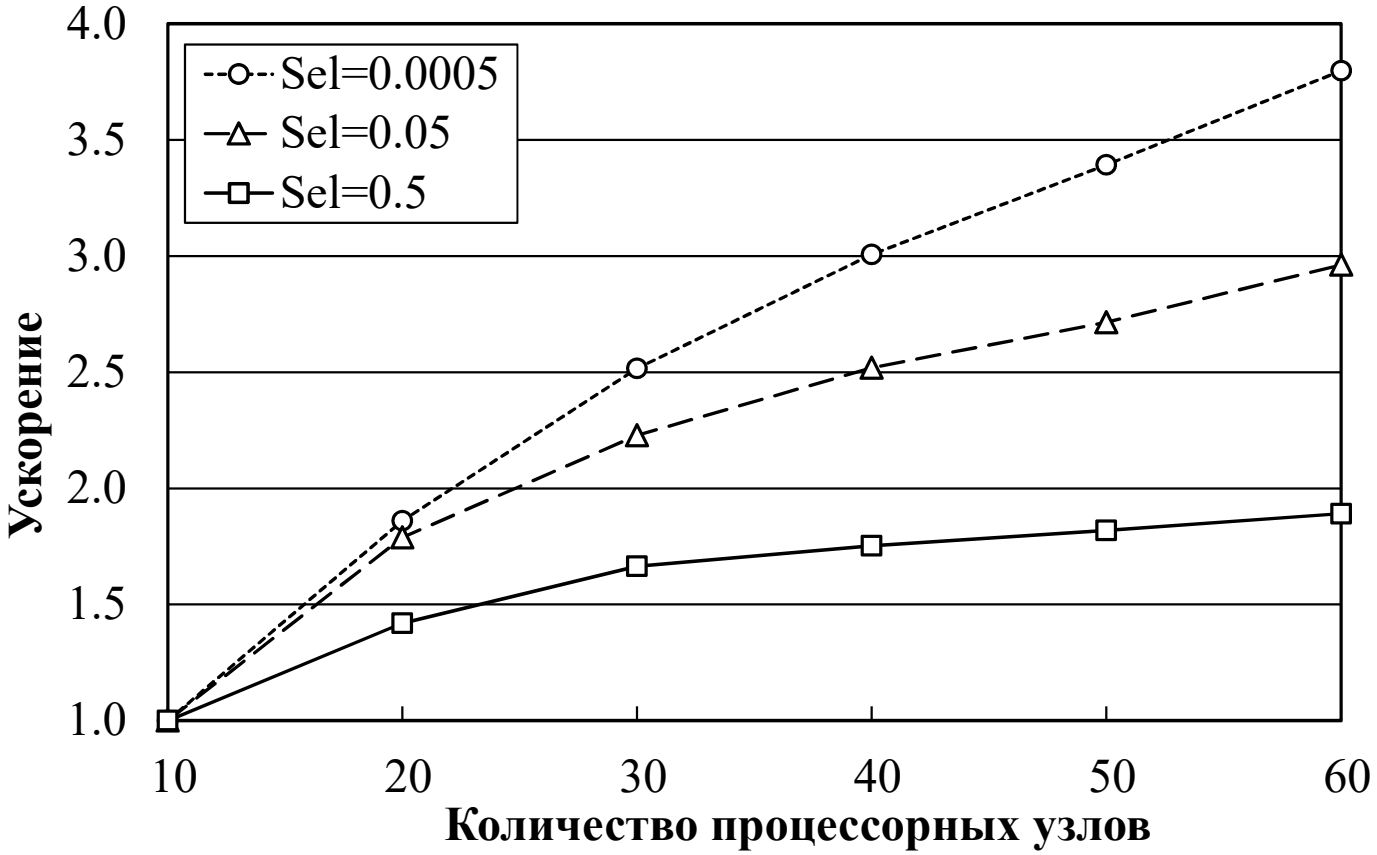
Размер хранилища:
63 млн. записей



Размер хранилища:
630 млн. записей

Вычислительная система: «Торнадо ЮУрГУ»

Масштабируемость КСОП



Размер хранилища: 63 млн. записей

Вычислительная система: «RSC PetaStream»

Использование КСОП при выполнении SQL-запросов

Конфигурация	Время в минутах					
	Sel=0.0005		Sel=0.005		Sel=0.05	
	1-й запуск	2-й запуск	1-й запуск	2-й запуск	1-й запуск	2-й запуск
PostgreSQL	7.3	1.21	7.6	1.29	7.6	1.57
PostgreSQL & B-Trees	2.62	2.34	2.83	2.51	2.83	2.63
PostgreSQL & CCOP	0.073	0.008	0.65	0.05	2.03	1.72
Ускорение						
$\frac{t_{PostgreSQL}}{t_{PostgreSQL\ \&\ CCOP}}$	100	151	12	27	4	0.9
$\frac{t_{PostgreSQL\&B-Trees}}{t_{PostgreSQL\ \&\ CCOP}}$	36	293	4	50	1.4	1.53

Основные результаты, выносимые на защиту

1) Разработана доменно-колоночная модель представления данных, на базе которой выполнена декомпозиция основных реляционных операций с помощью распределенных колоночных индексов.
2) Разработаны высокомасштабируемые параллельные алгоритмы выполнения основных реляционных операций, использующие распределенные колоночные индексы.
3) Выполнена реализация колоночного сопроцессора для кластерных вычислительных систем. Общий объем кода на языке Си составил около 2500 строк. Исходные тексты прототипа свободно доступны в Интернет по адресу: https://github.com/elena-ivanova/colomnindices/.
4) Проведены вычислительные эксперименты, подтверждающие эффективность предложенных подходов.