

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Южно-Уральский государственный университет (национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра системного программирования

Разработка параллельного алгоритма поиска лейтмотивов временного ряда для многоядерных ускорителей

Я.А. Краева

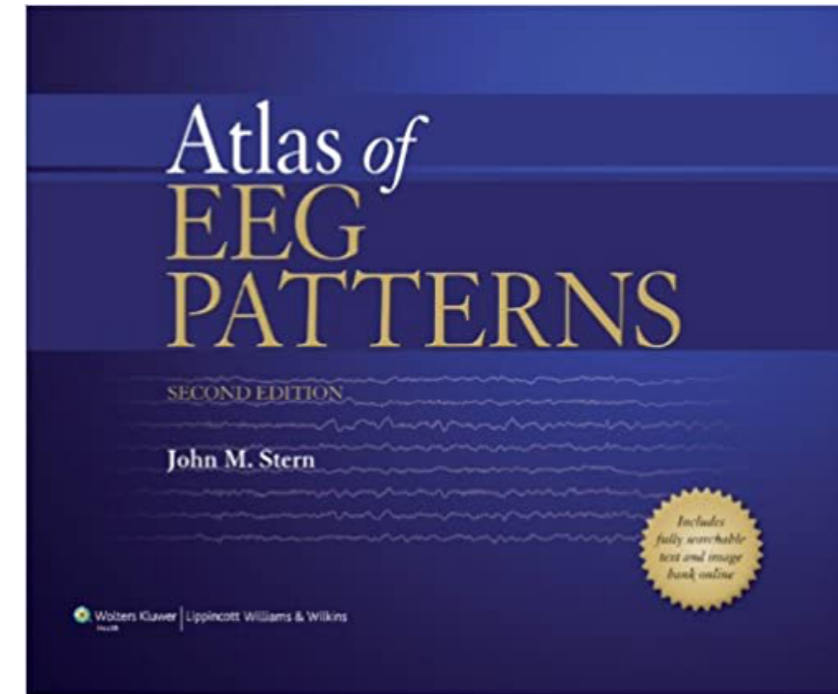
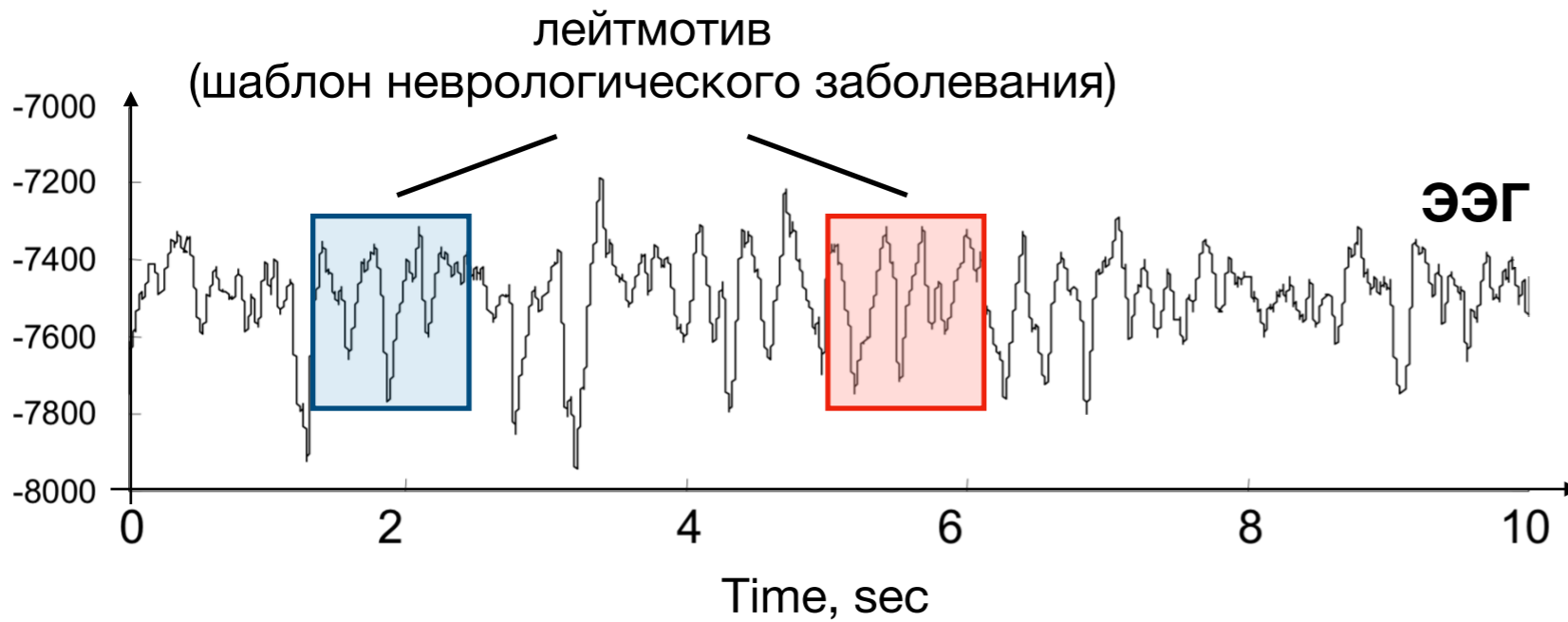
Рецензент:

Вед. программист
ООО «ВОРТЕКСКОД»
П.А. Михайлов

Научный руководитель:

к.ф.-м.н., доцент
М.Л. Цымблер

Актуальность поиска лейтмотивов



Атлас шаблонов болезней

- Лейтмотивы в электроэнцефалограмме (ЭЭГ) пациентов позволяют составить атлас неврологических болезней и выявить заболевания

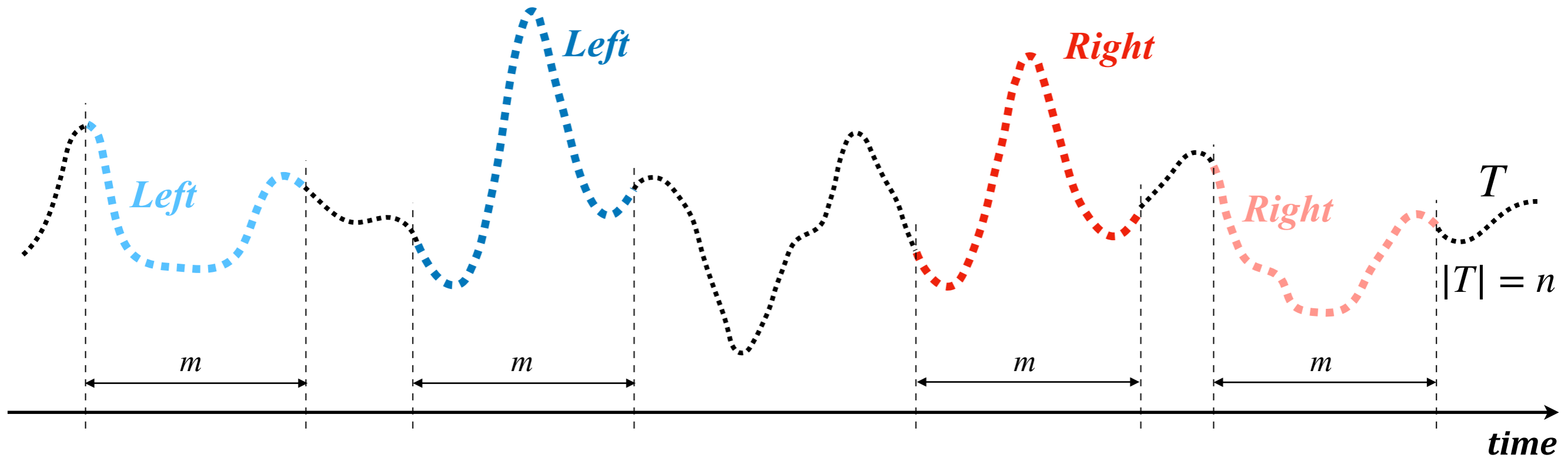
Цель и задачи исследования

Цель: разработать параллельный алгоритм поиска лейтмотивов временного ряда для многоядерных ускорителей.

Задачи:

1. Провести обзор последовательных и параллельных алгоритмов поиска лейтмотивов временного ряда.
2. Изучить аппаратную архитектуру и программную модель многоядерного процессора семейства Intel MIC и графического процессора NVIDIA GPU.
3. Спроектировать и реализовать параллельный алгоритм поиска лейтмотивов временного ряда для многоядерных ускорителей.
4. Провести вычислительные эксперименты по анализу эффективности разработанного алгоритма.

Лейтмотив временного ряда



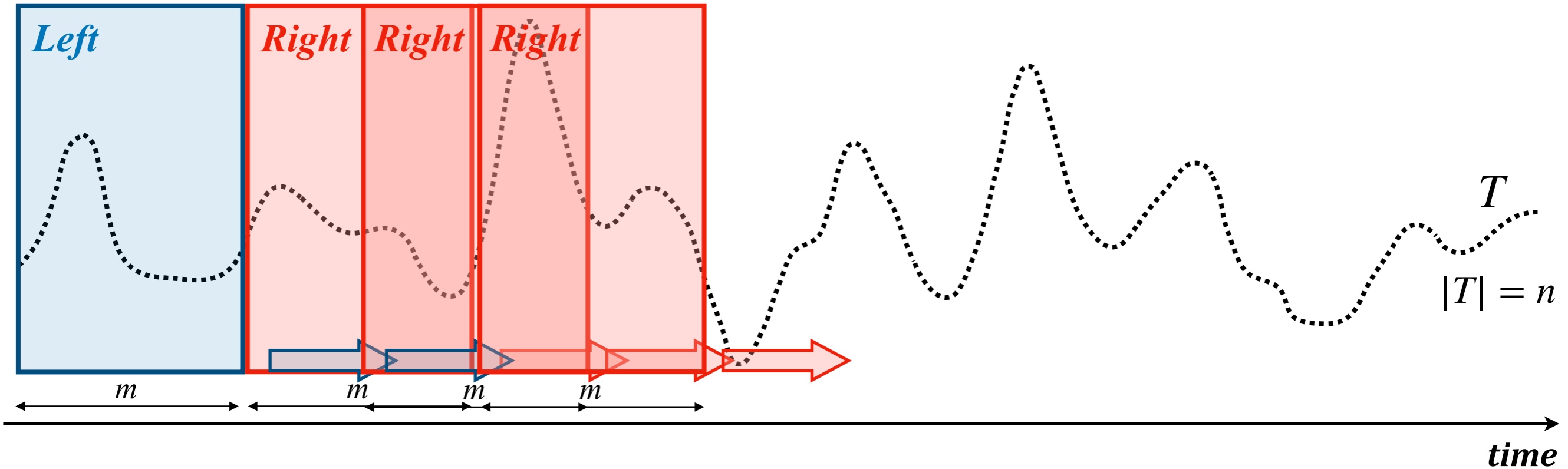
Лейтмотив $\{Left, Right\}$ – пара произвольных непересекающихся подпоследовательностей длины m временного ряда T , наиболее похожих друг на друга

$$\forall Left, Right \quad ED(Left, Right) < ED(Left, Right)$$

$$Left \cap Right = \emptyset,$$

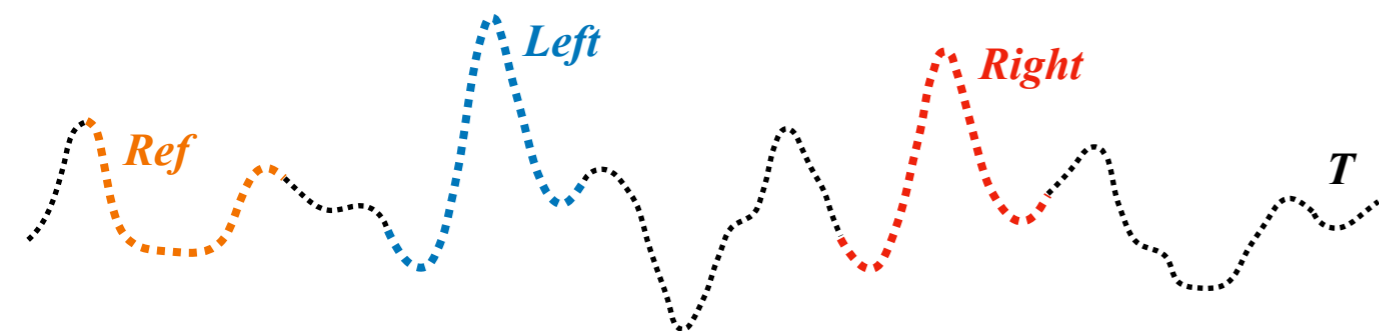
$ED(\cdot, \cdot)$ – евклидово расстояние

Требования к алгоритму

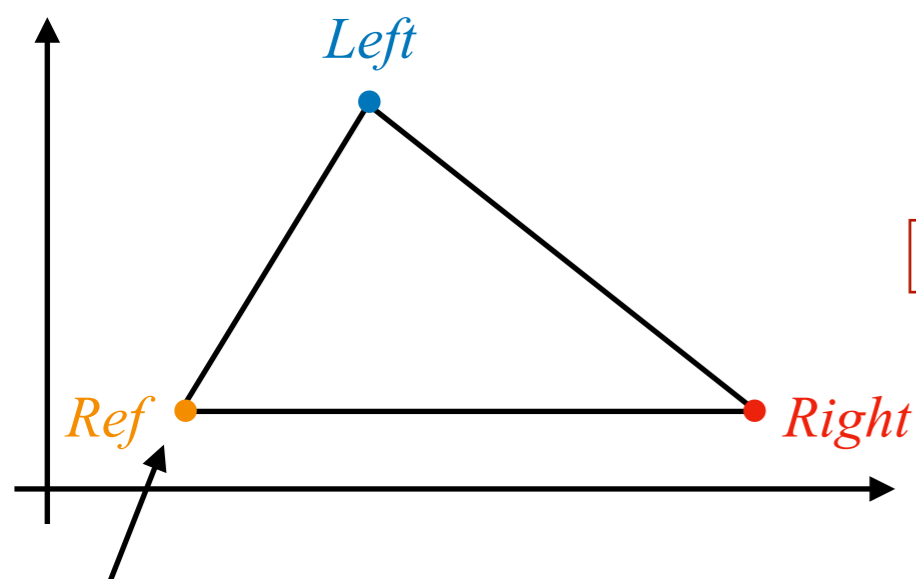


- Вычислительная сложность полного перебора $O(n^2 \cdot m)$
- Требования к параллельному алгоритму:
 1. Сокращение временной сложности.
 2. Максимальное использование возможностей векторных вычислений многоядерного ускорителя.
 3. Высокая масштабируемость.

Сокращение полного перебора

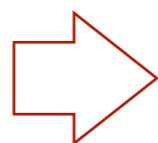


↓ Проецирование
в \mathbb{R}^2

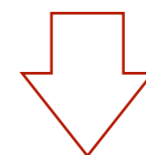


опорная
подп-ть

Неравенство треугольника



$$ED(Ref, Right) + ED(Left, Right) \geq ED(Ref, Left)$$



Обратное неравенство треугольника

$$\underbrace{ED(Ref, Left) - ED(Ref, Right)}_{\text{Нижняя граница LB}} \leq \underbrace{ED(Left, Right)}_{\text{Истинное расстояние}}$$

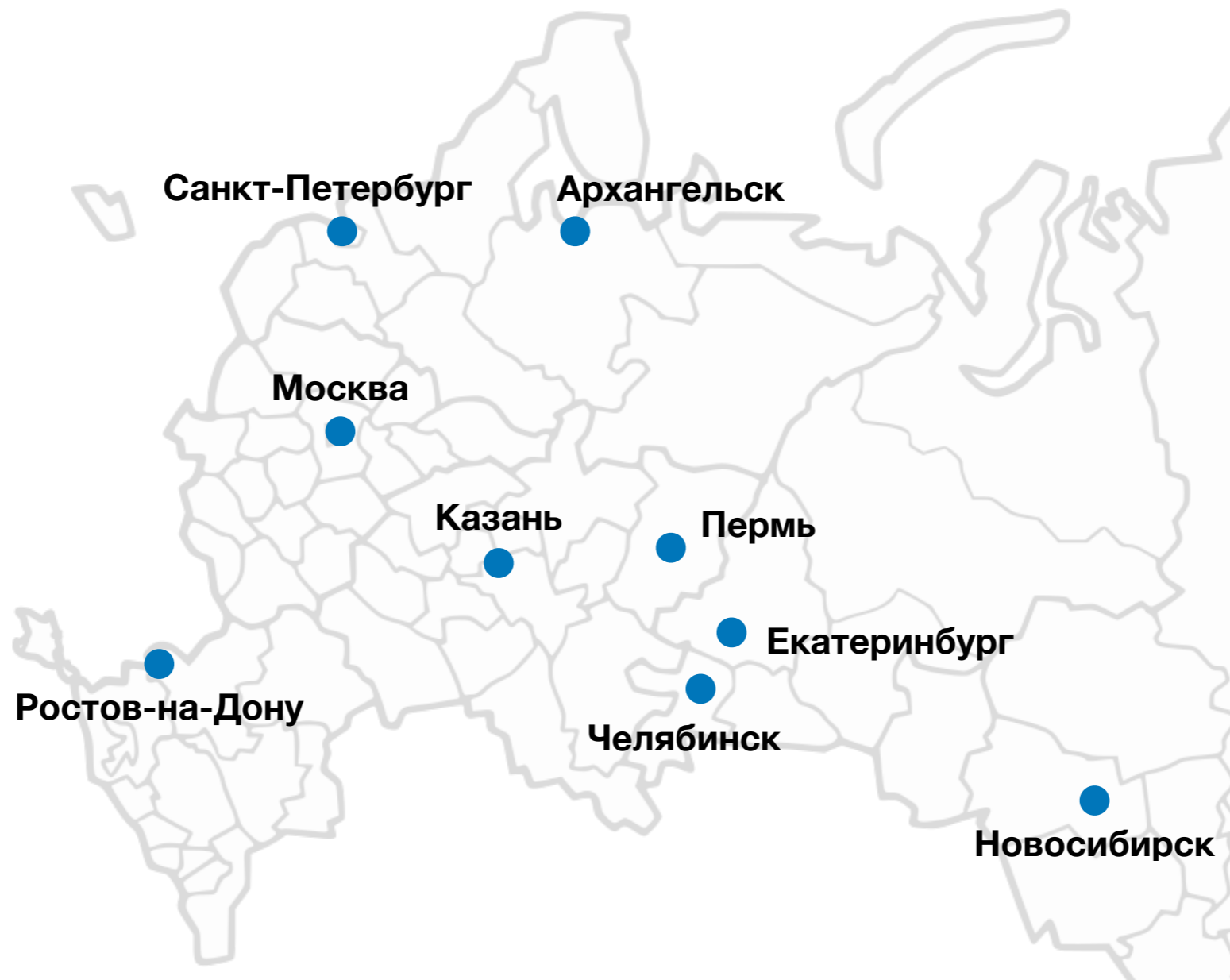
Нижняя граница LB

Истинное расстояние

Пример сокращенного поиска лейтмотивов



↓ Проецирование
в \mathbb{R}^2



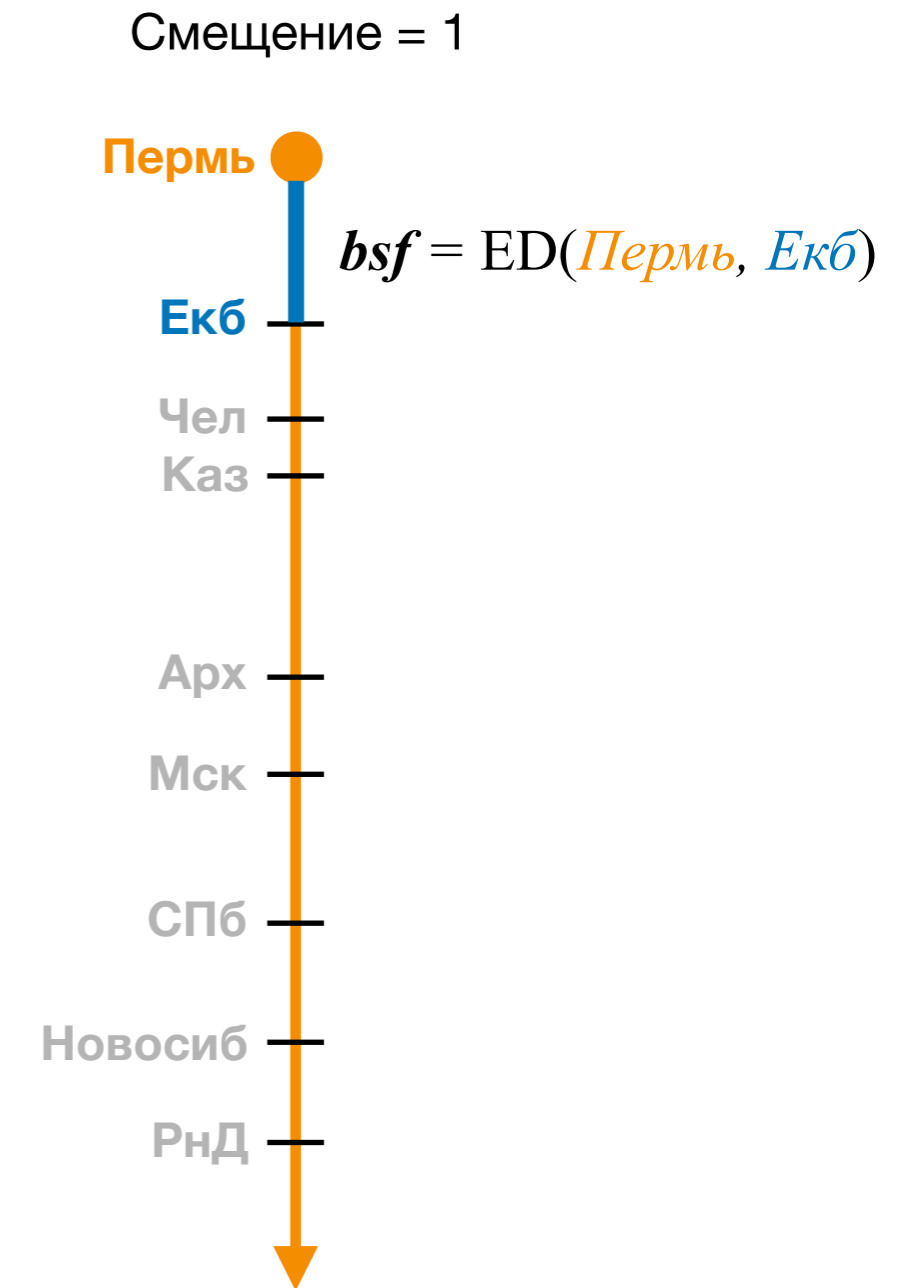
Выбор опорного города



Линейный порядок городов

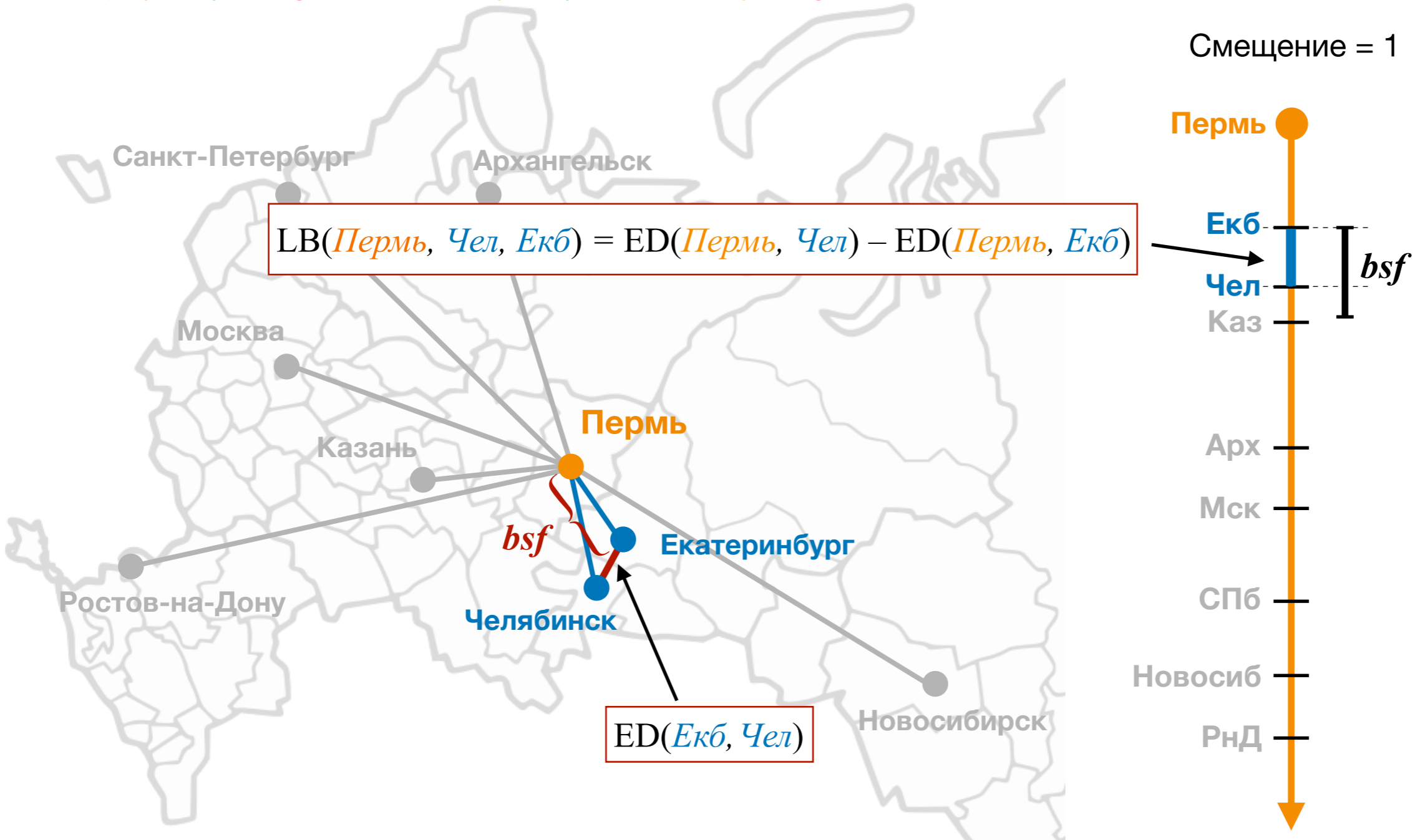


Вычисление начального значения порога



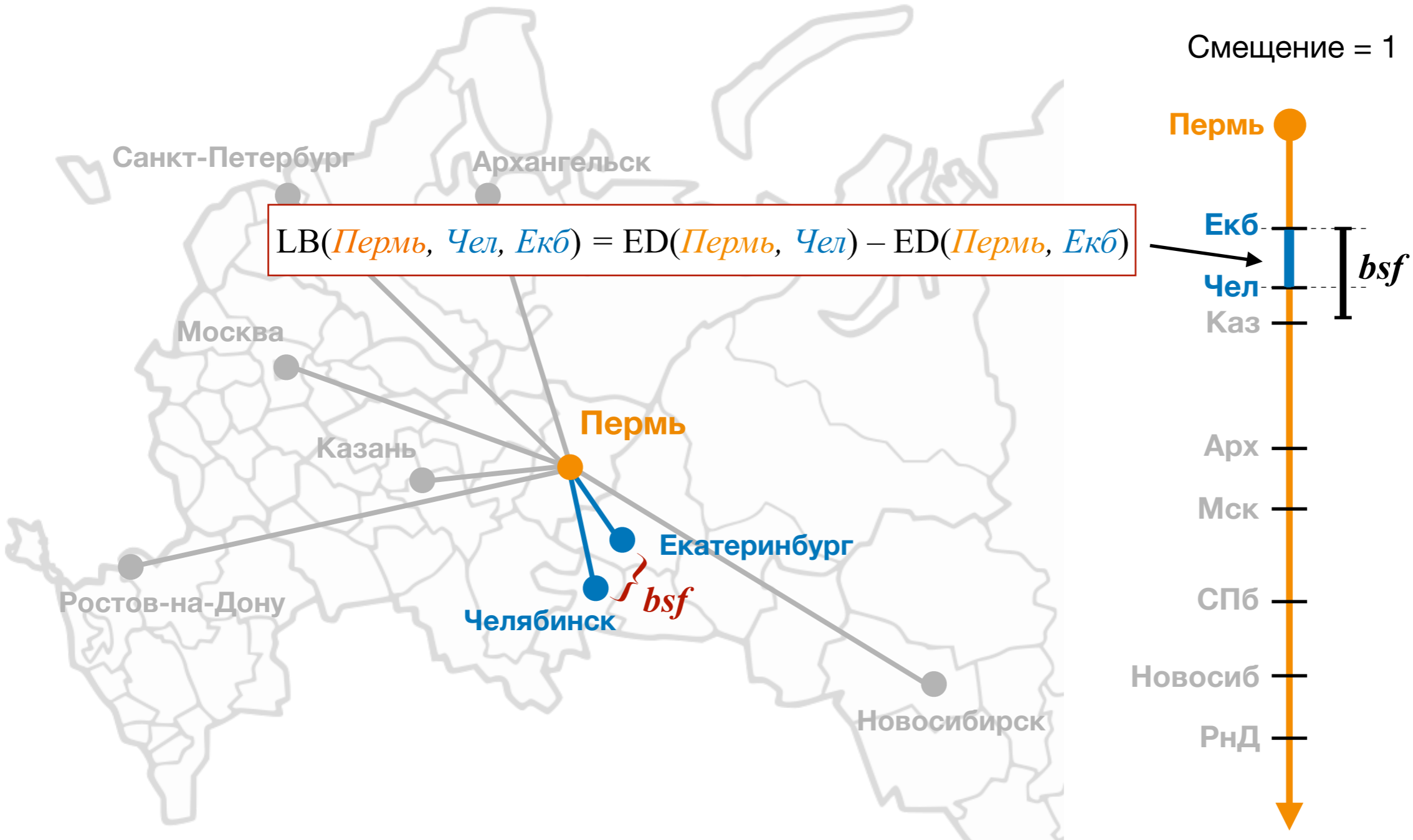
Вычисление нижней границы

$$LB(Ref, Left, Right) = ED(Ref, Left) - ED(Ref, Right)$$



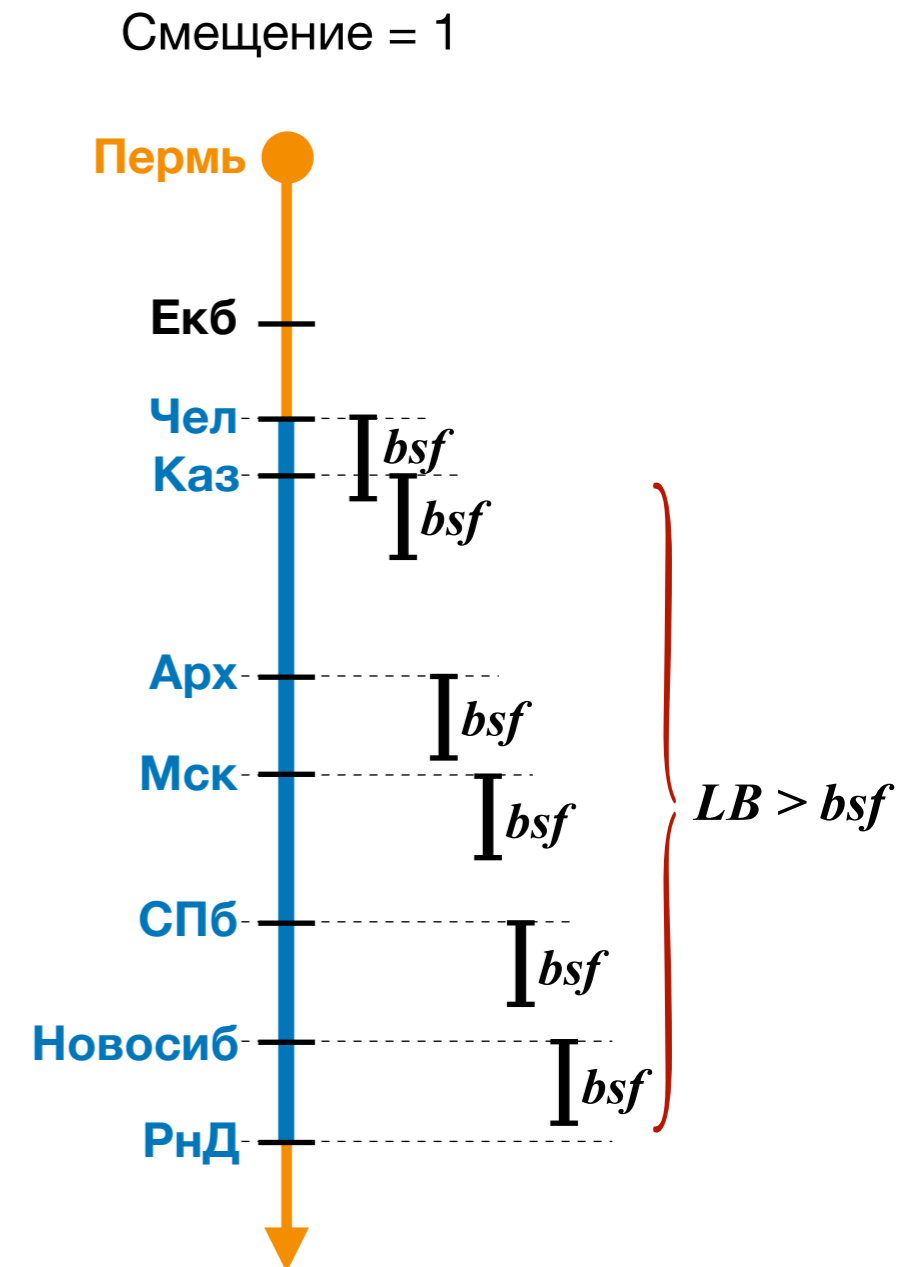
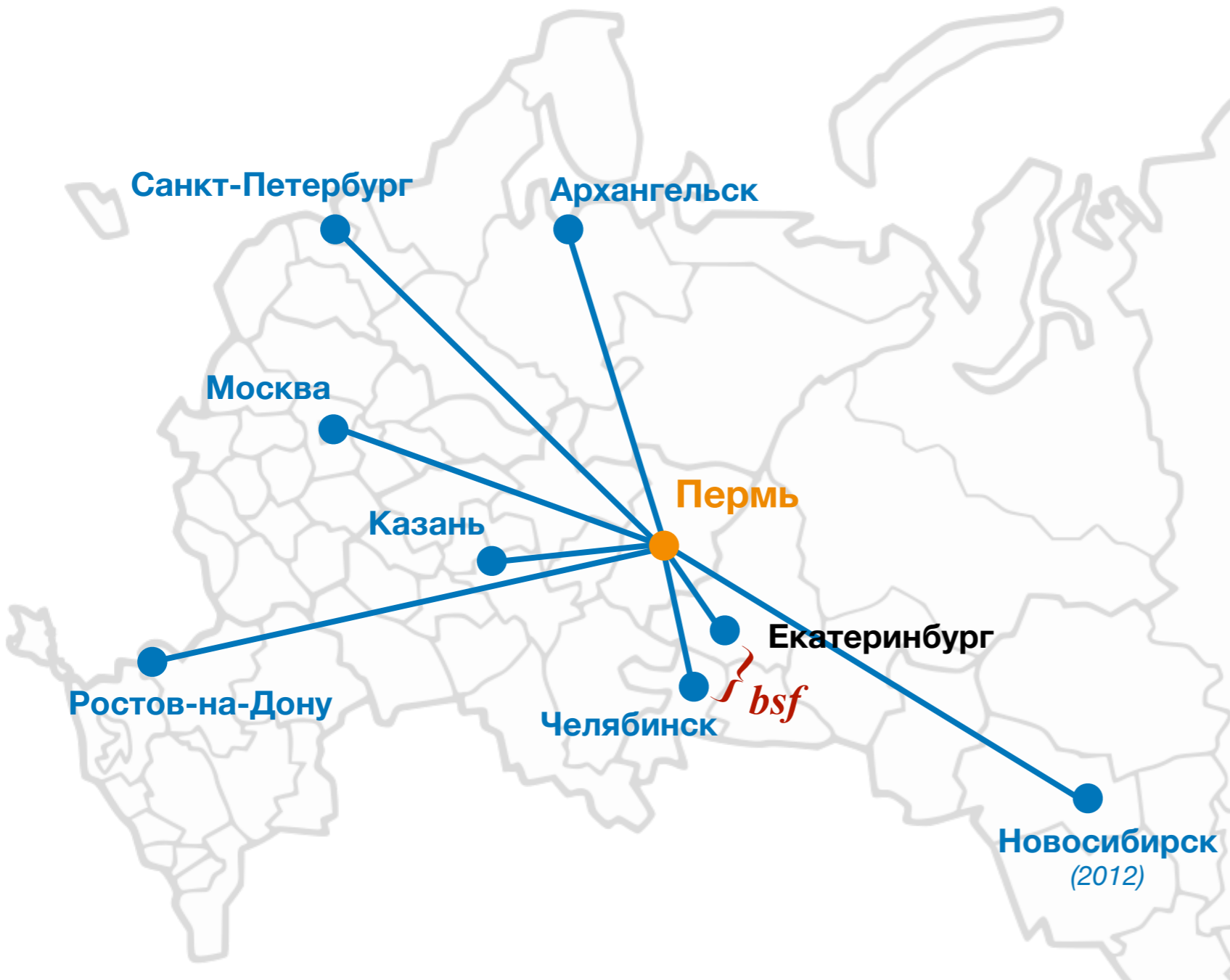
$$LB(Пермь, Чел, Екб) < bsf \Rightarrow bsf = \min(bsf, ED(Екб, Чел))$$

Обновление значения порога



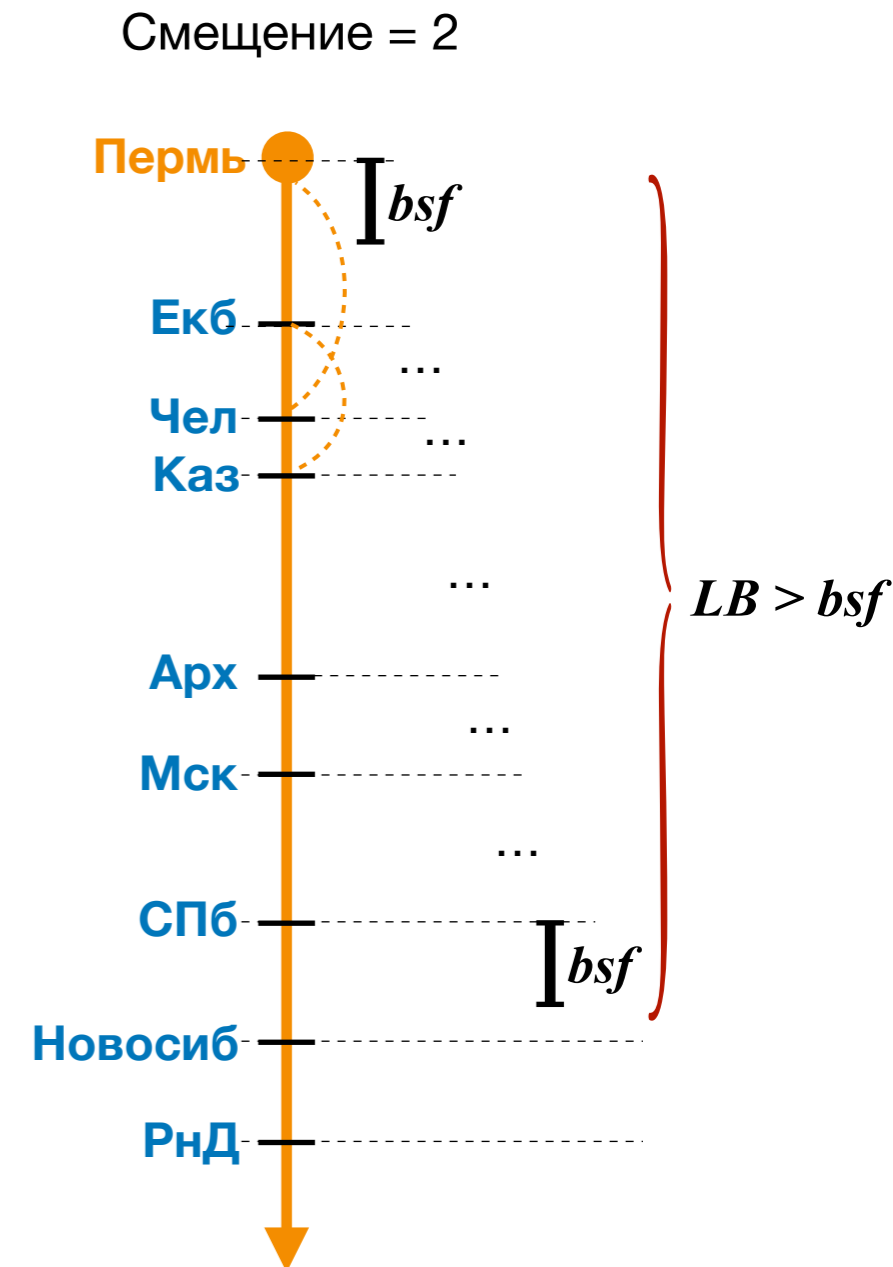
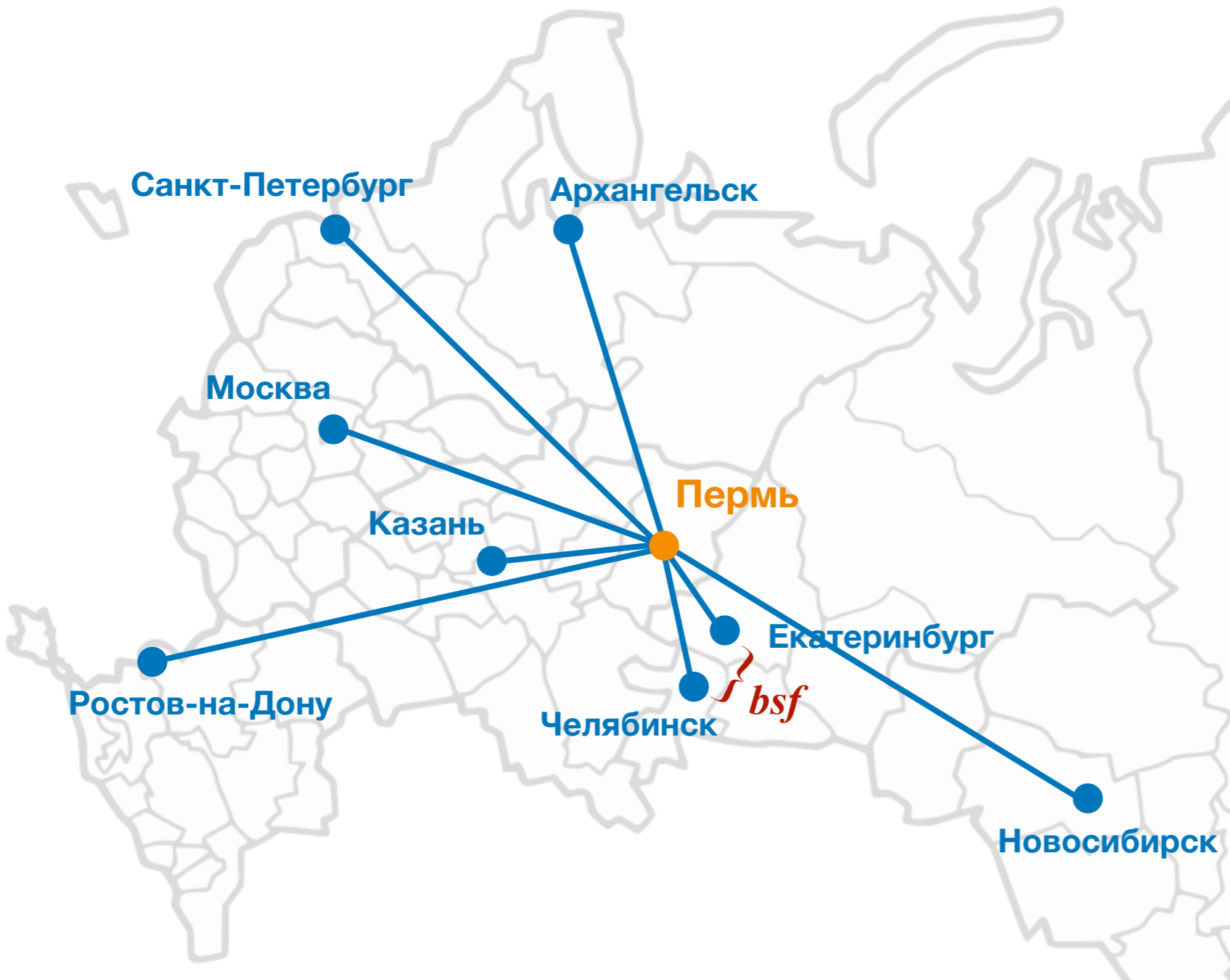
$$LB(Пермь, Чел, Екб) < bsf \Rightarrow bsf = \min(bsf, ED(Екб, Чел))$$

Отбрасывание кандидатов



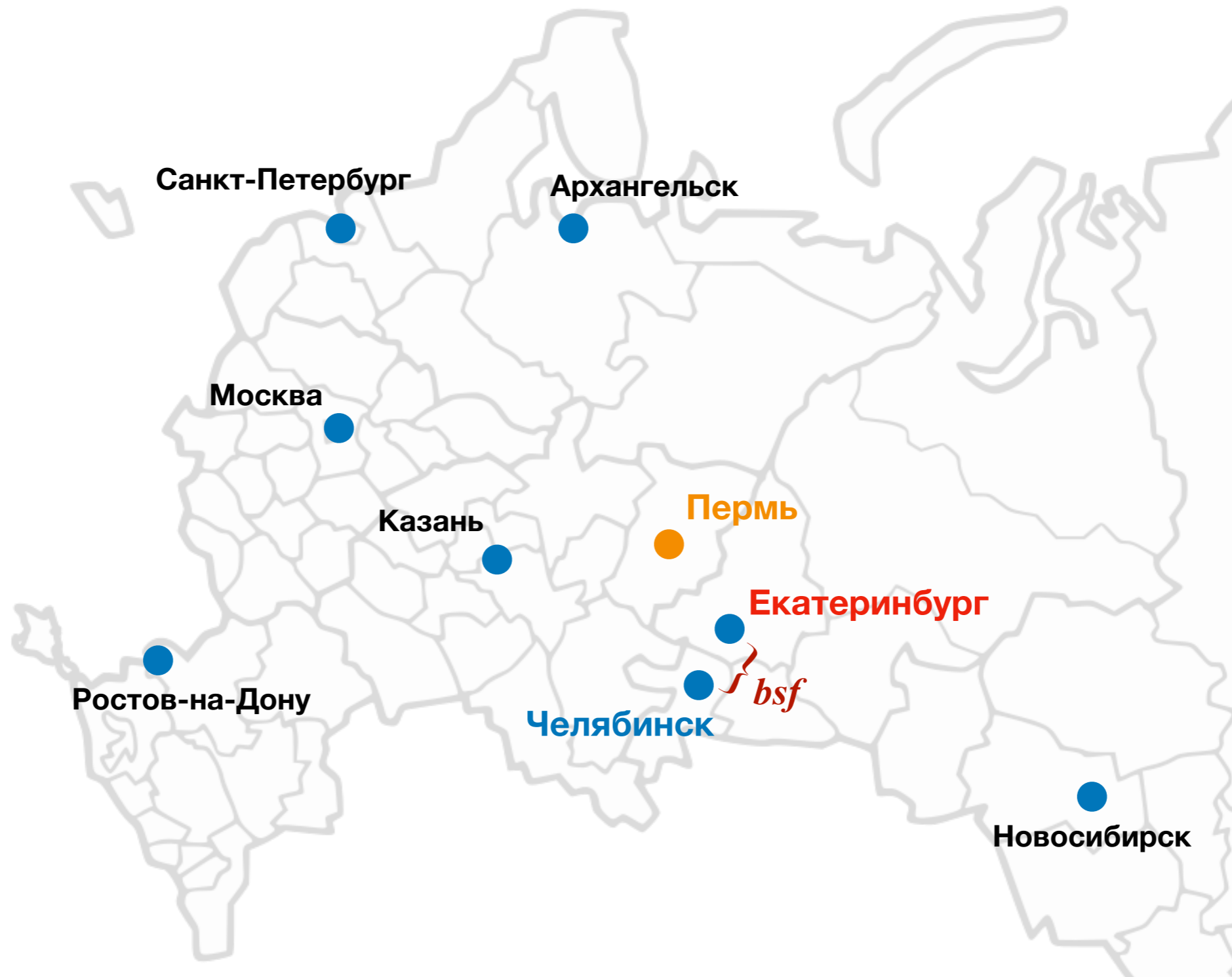
$LB(Ref, Left, Right) > bsf \Rightarrow \{Left, Right\}$ отбрасывается
 без вычисления $ED(Left, Right)$

Отбрасывание кандидатов



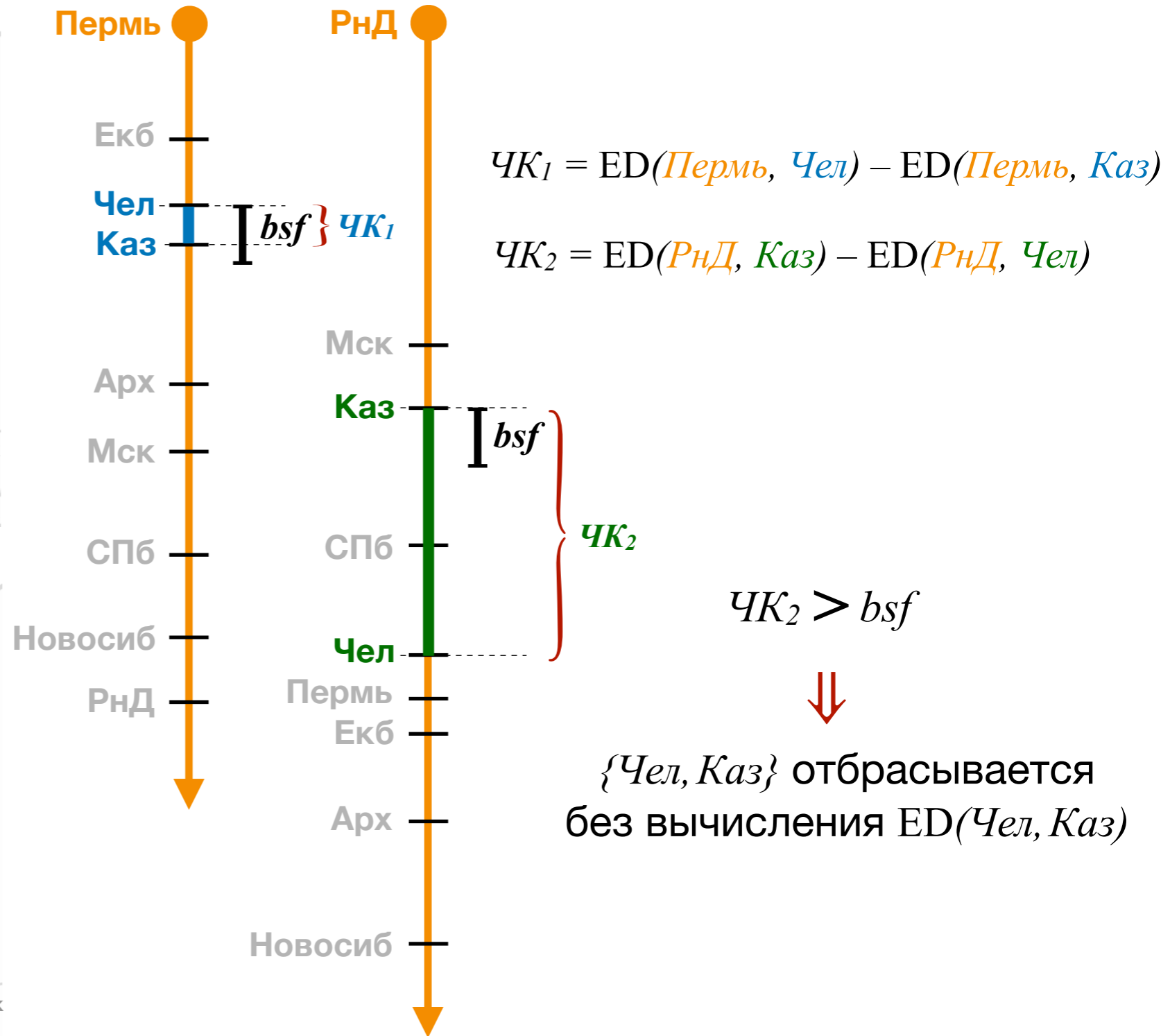
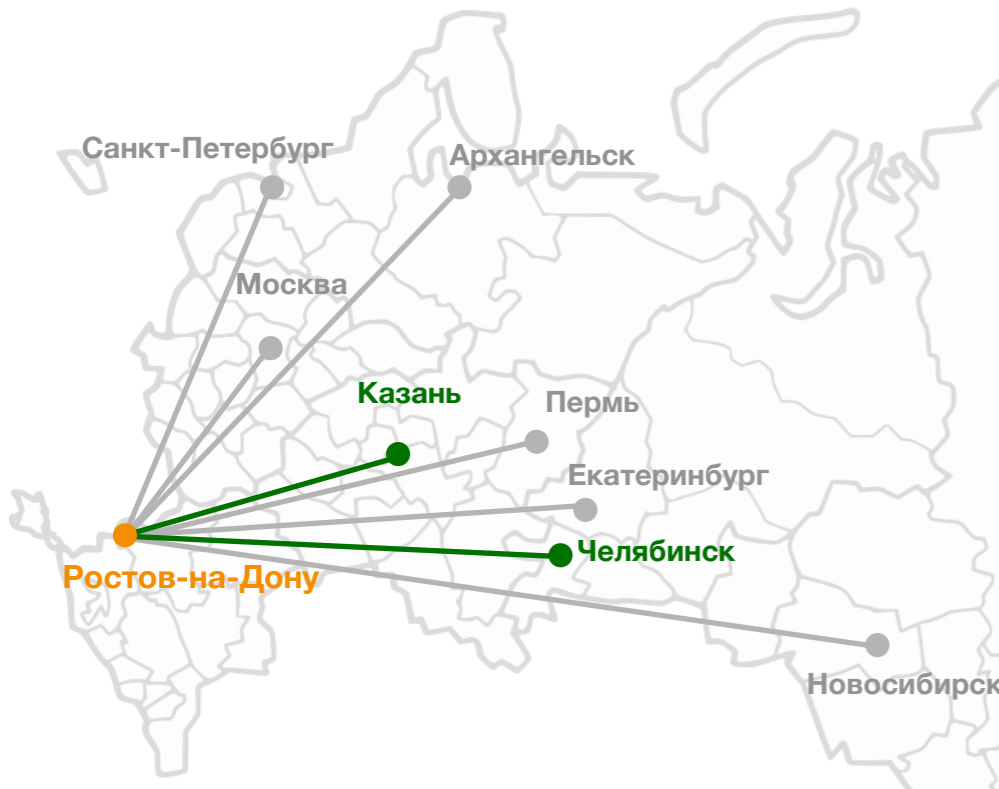
$\exists offset_{stop}$
 $\forall LB(Ref, Left, Right) > bsf \Rightarrow \forall offset_{next} \geq offset_{stop} \forall \{Left, Right\}$ отбрасывается
 без вычисления $ED(Left, Right)$

Пример сокращенного поиска лейтмотивов

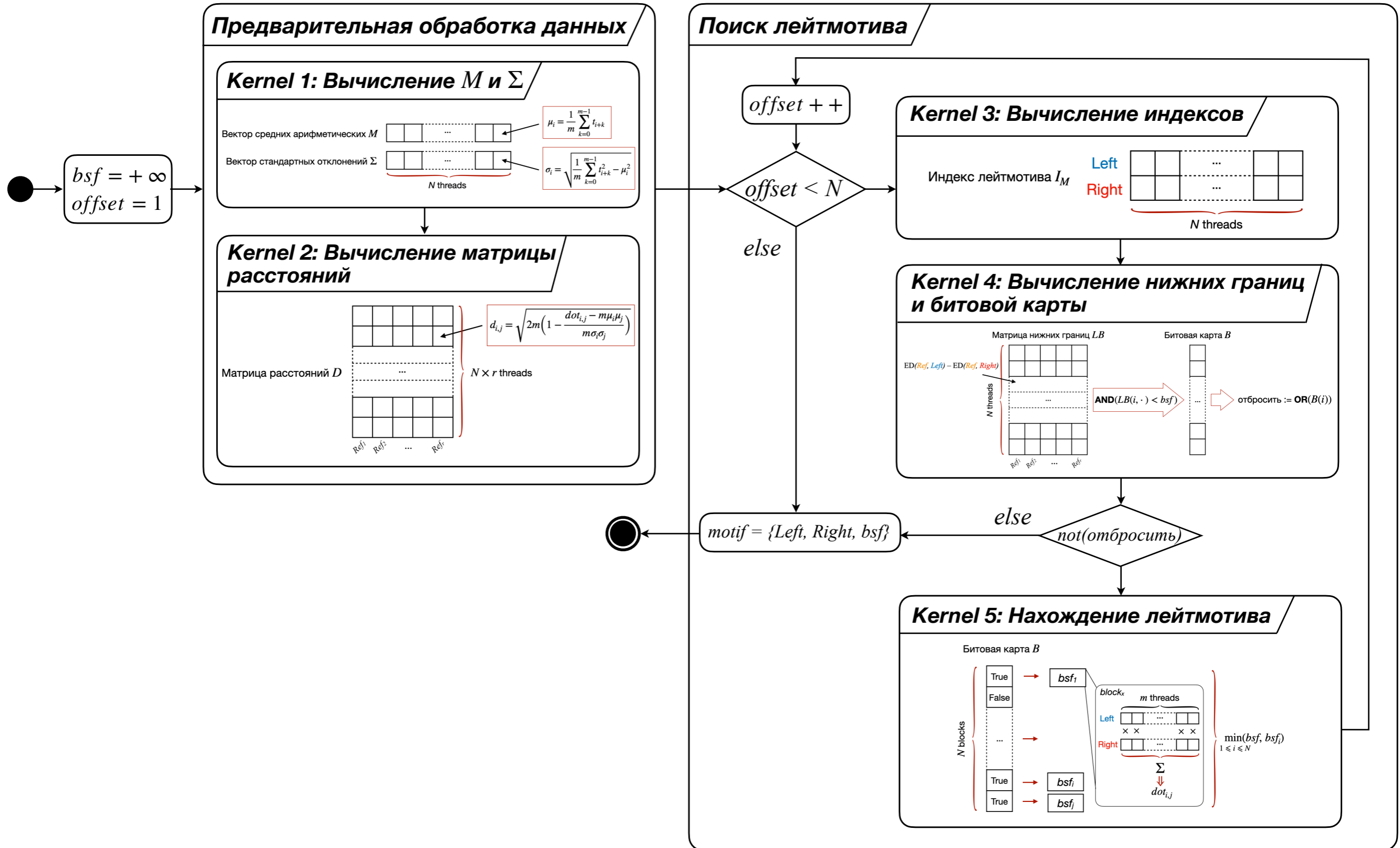


{Челябинск, Екатеринбург} – лейтмотив

Несколько опорных городов



Параллельный алгоритм для GPU



Предварительная обработка данных

$ED_{norm}(\cdot, \cdot)$ вместо $ED(\cdot, \cdot)$

$$ED_{norm}(\cdot, \cdot) = \sqrt{2m \left(1 - \frac{dot_{i,j} - m\mu_i\mu_j}{m\sigma_i\sigma_j} \right)}$$

Kernel 1: Вычисление M и Σ

Вектор средних арифметических M



$$\mu_i = \frac{1}{m} \sum_{k=0}^{m-1} t_{i+k}$$

Вектор стандартных отклонений Σ

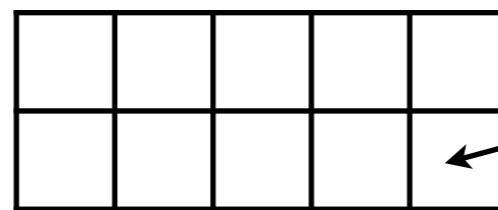


$$\sigma_i = \sqrt{\frac{1}{m} \sum_{k=0}^{m-1} t_{i+k}^2 - \mu_i^2}$$

N threads

Kernel 2: Вычисление матрицы расстояний

Матрица расстояний D



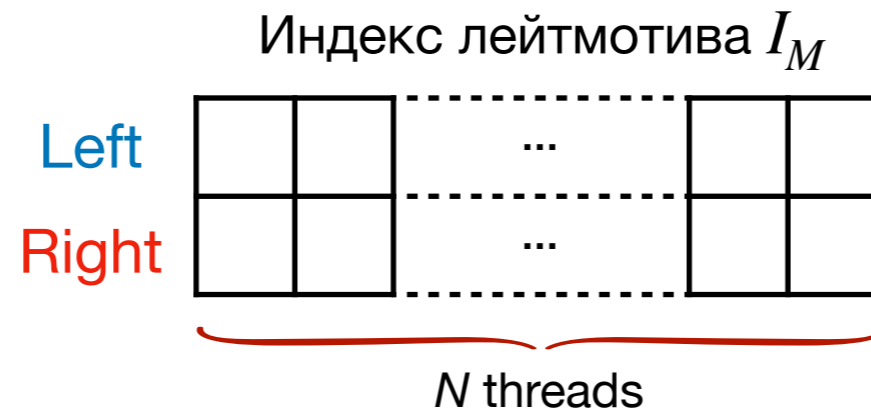
$$d_{i,j} = \sqrt{2m \left(1 - \frac{dot_{i,j} - m\mu_i\mu_j}{m\sigma_i\sigma_j} \right)}$$

$N \times r$ threads

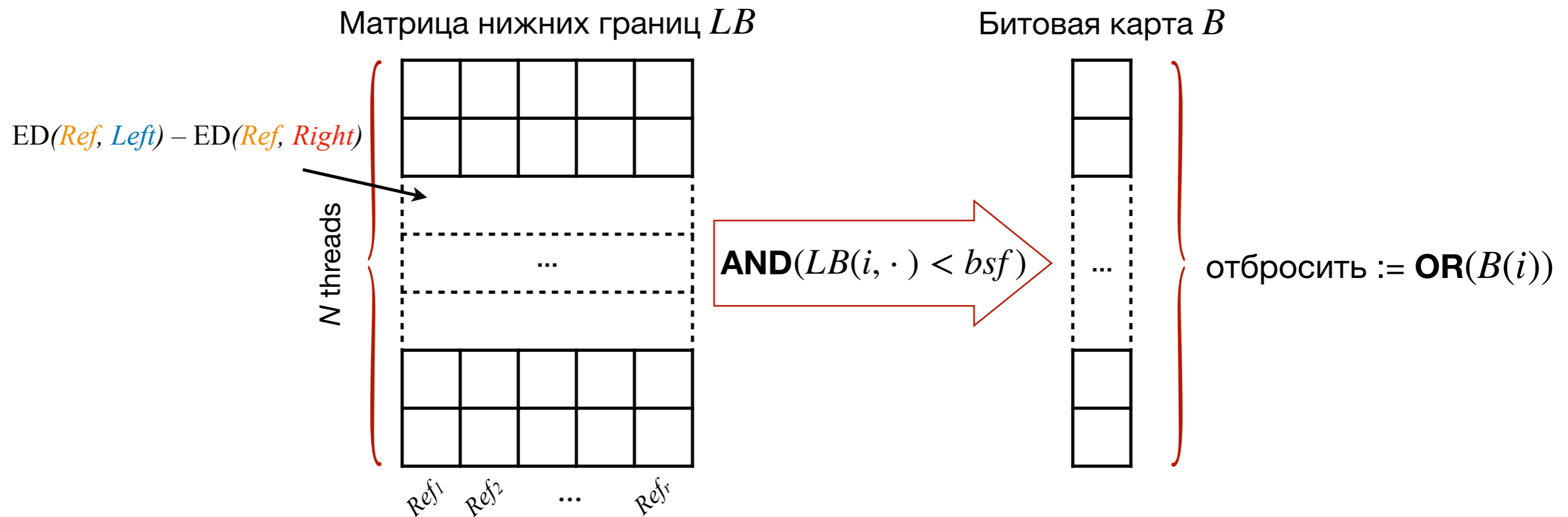
Ref₁ Ref₂ ... Ref_r

Поиск лейтмотива

Kernel 3: Вычисление индексов

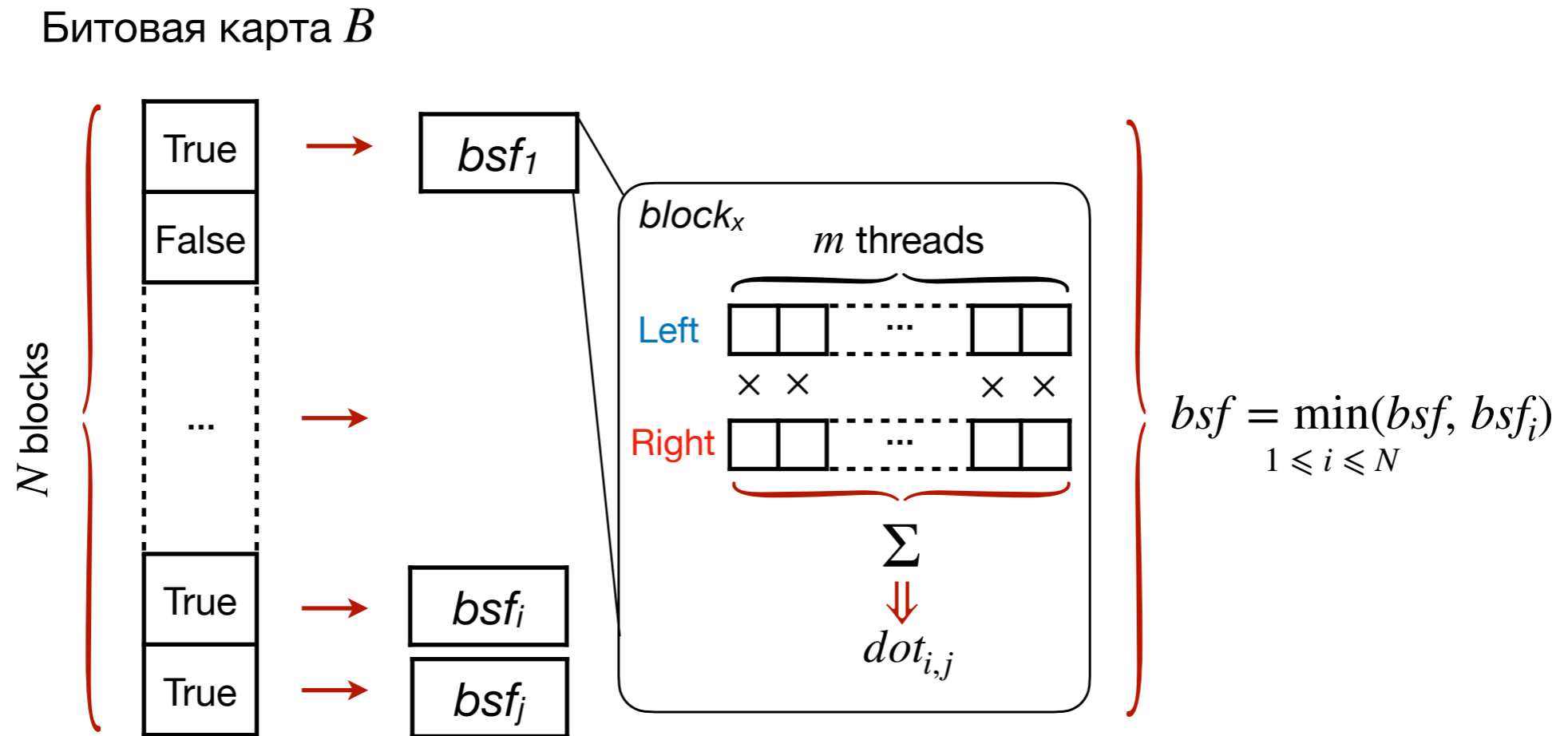


Kernel 4: Вычисление нижних границ и битовой карты



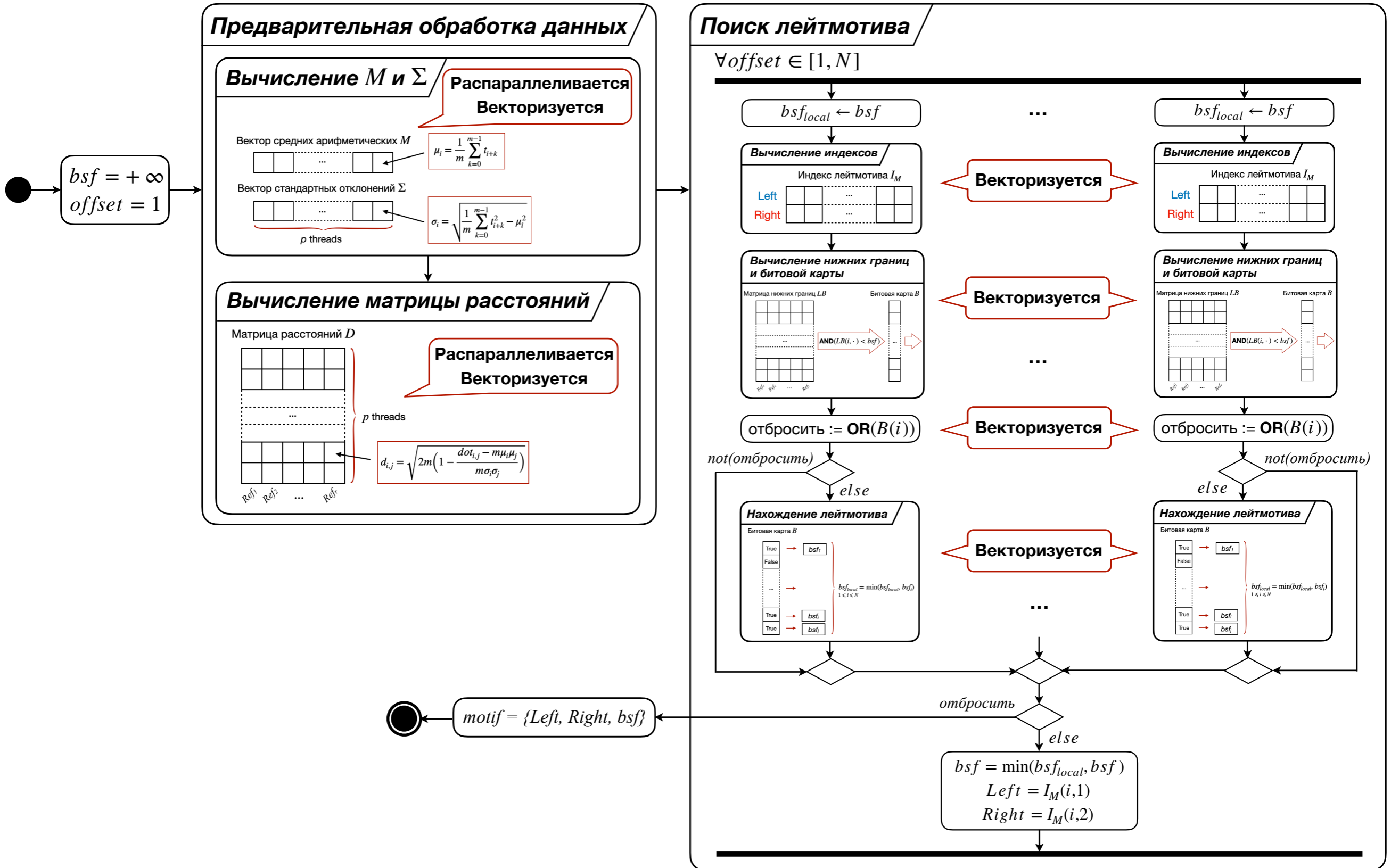
Поиск лейтмотива

Kernel 5: Нахождение лейтмотива



$$ED_{norm} = \sqrt{2m \left(1 - \frac{dot_{i,j} - m\mu_i\mu_j}{m\sigma_i\sigma_j} \right)}$$

Параллельный алгоритм для Intel MIC



- **Аппаратная платформа**

Характеристика	Intel MIC	NVIDIA GPU
Модель	Xeon Phi SE10X	GeForce GTX 2080Ti
Кол-во физ. ядер	61	4 352 (68 мультипроцессоров)
Гиперпоточность	4×	–
Кол-во лог. ядер	244	–
Тактовая частота, ГГц	1.1	1.35
Пиковая произв-ть, TFLOPS	1.076	11

- **Наборы данных**

- Random Walk ¹⁾: $|T| = 4 \cdot 10^5$
- ЭКГ ²⁾: $|T| = 4 \cdot 10^5$

- **Показатели**

- Быстродействие, t – время работы параллельного алгоритма
- Ускорение, $s(k) = \frac{t_1}{t_k}$, k – кол-во мультипроцессоров GPU / нитей Intel MIC

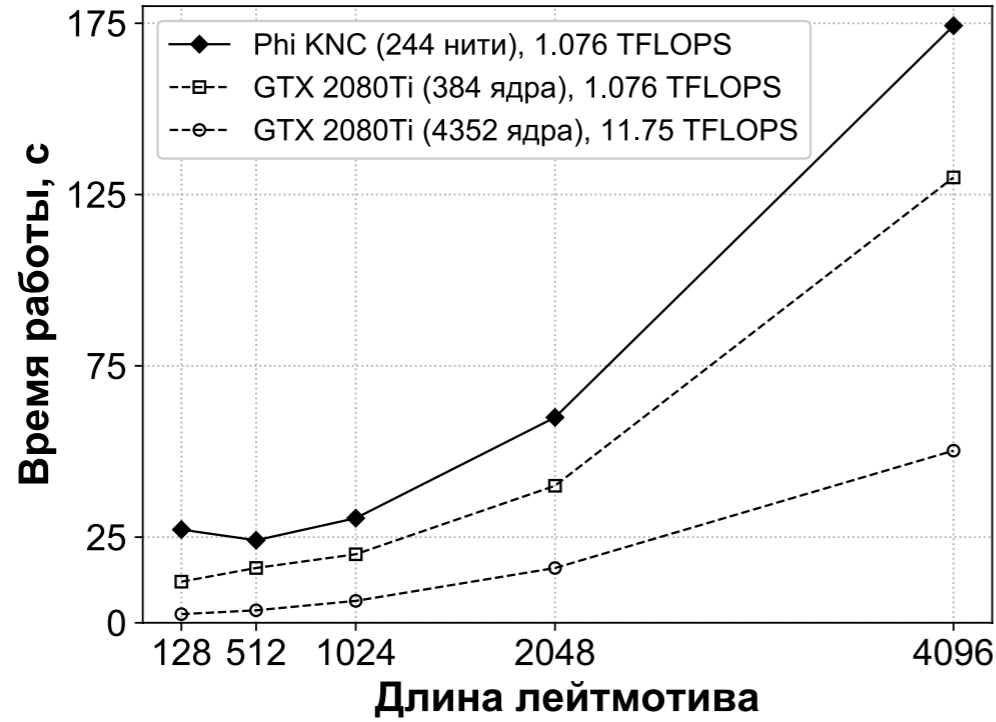
¹⁾ Mueen A., et al. Exact Discovery of Time Series Motifs. Proceedings of the SIAM Int. Conf. on Data Mining. 2009. pp. 473–484.

²⁾ Goldberger A.L., et al. PhysioBank, PhysioToolkit, and PhysioNet. Circulation. 2000. vol. 101. no. 23. pp. e215–e220.

Быстродействие алгоритма

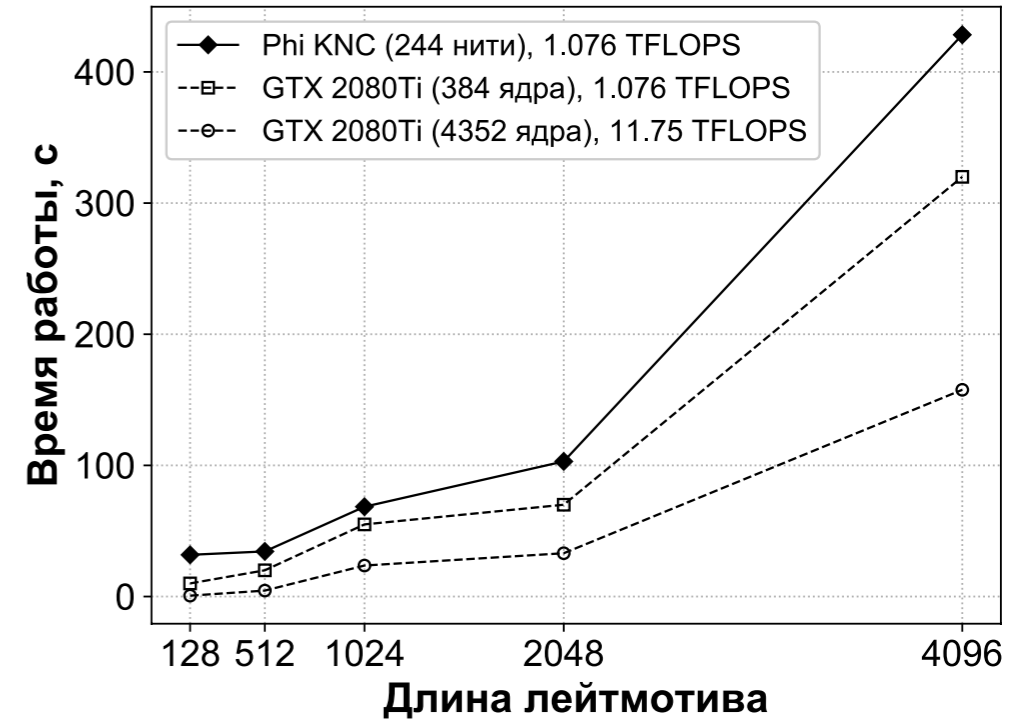
Синтетический ряд, Random Walk

Длина ряда,
 $n = 10^5$

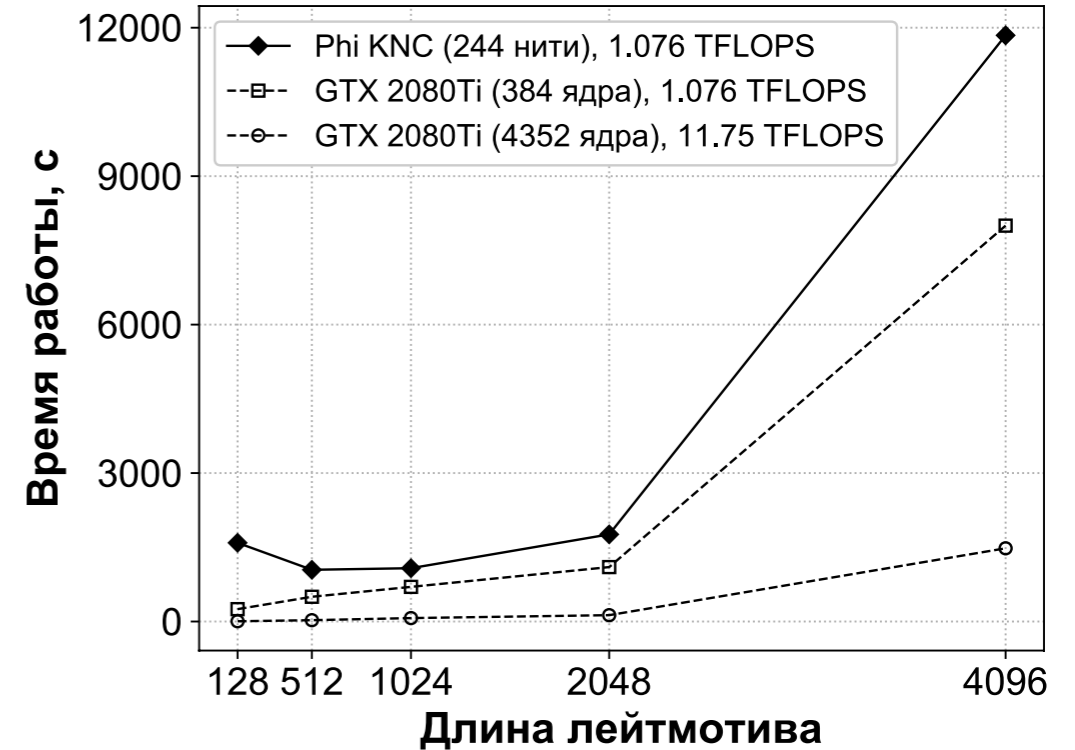
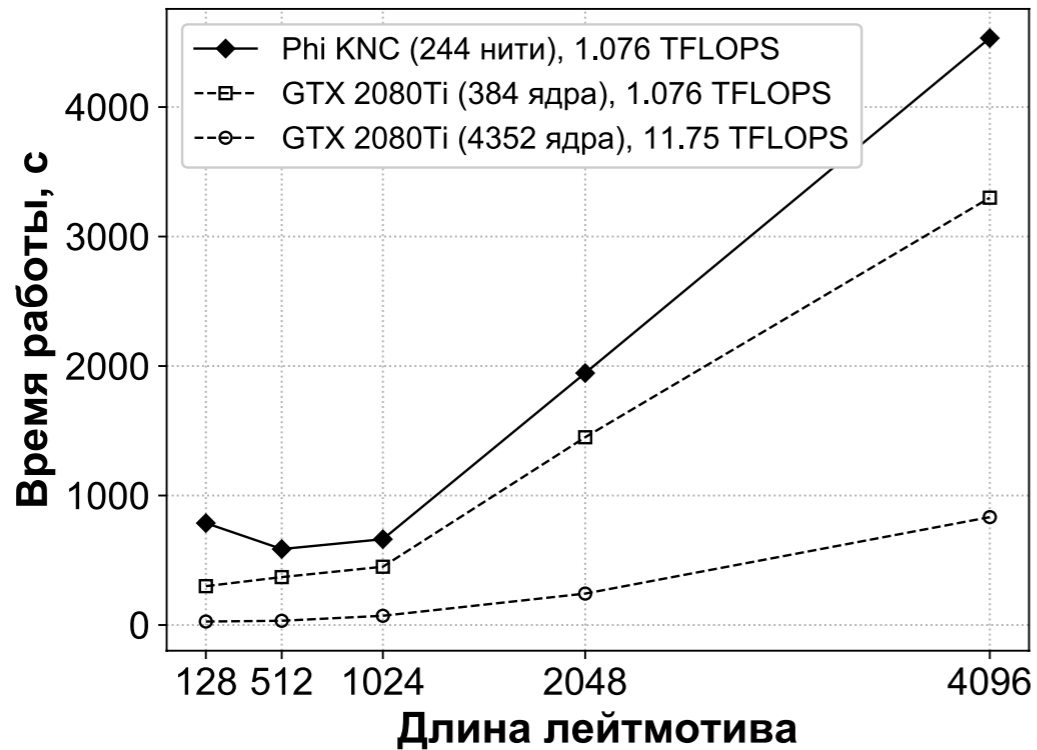


Реальный ряд, ЭКГ

Время работы, с



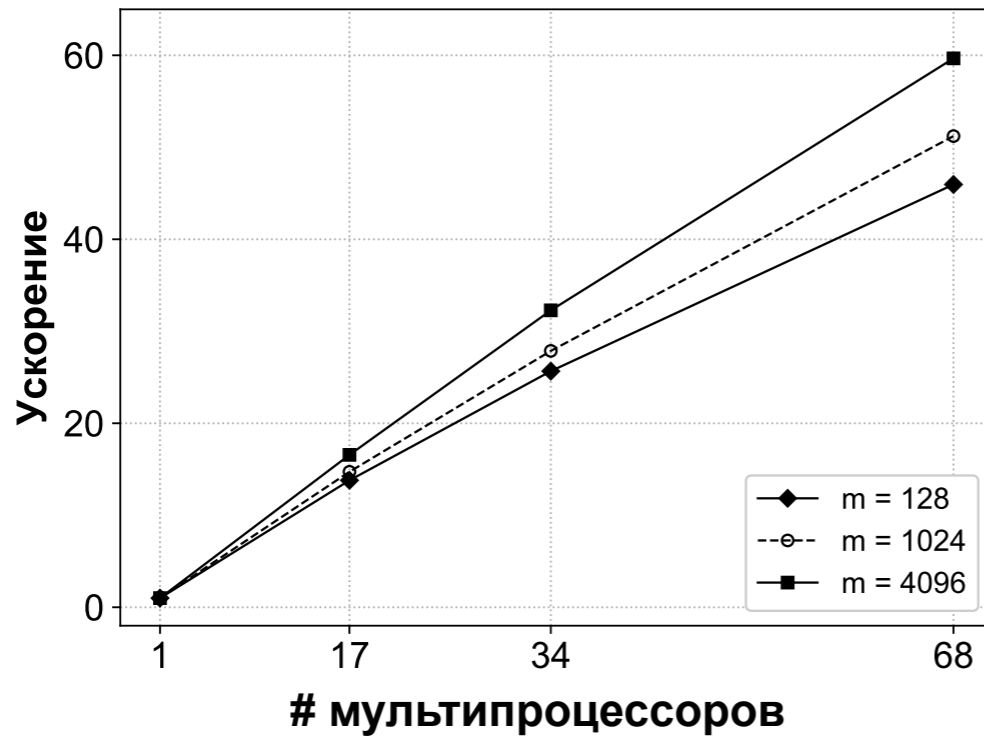
Длина ряда,
 $n = 4 \cdot 10^5$



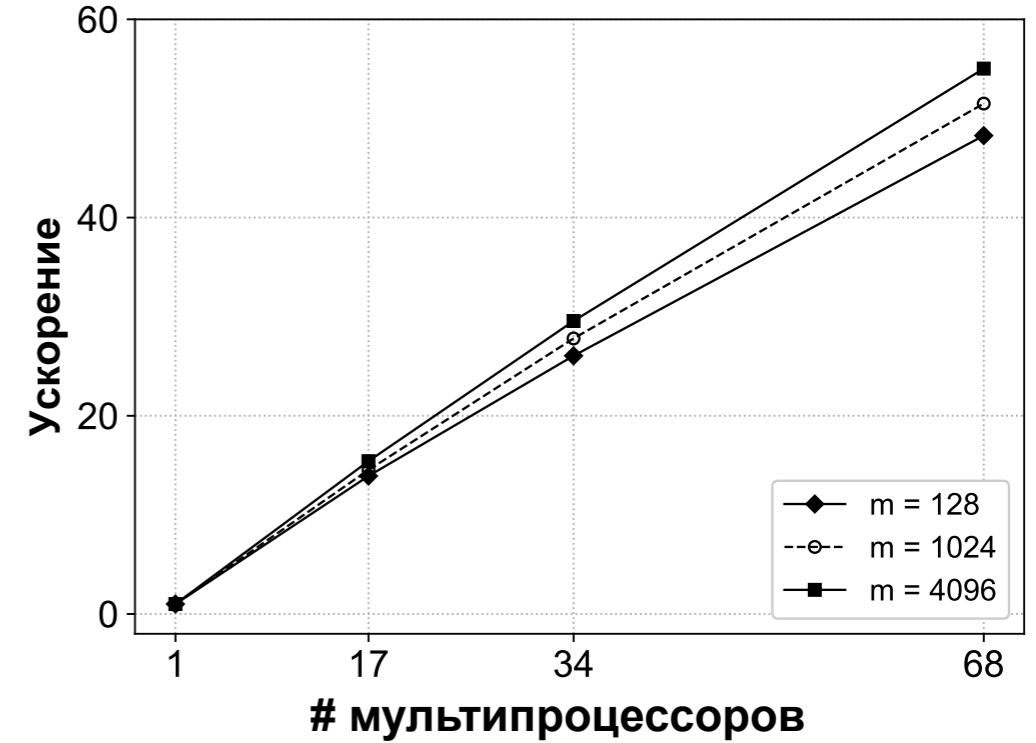
Ускорение алгоритма

GPU

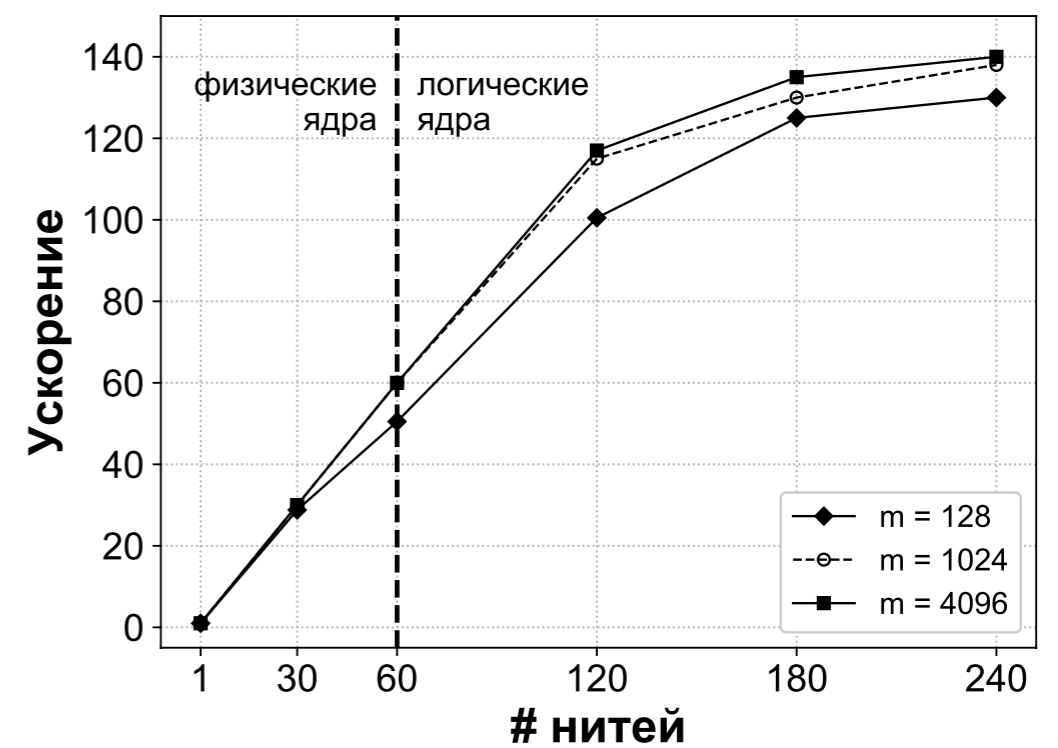
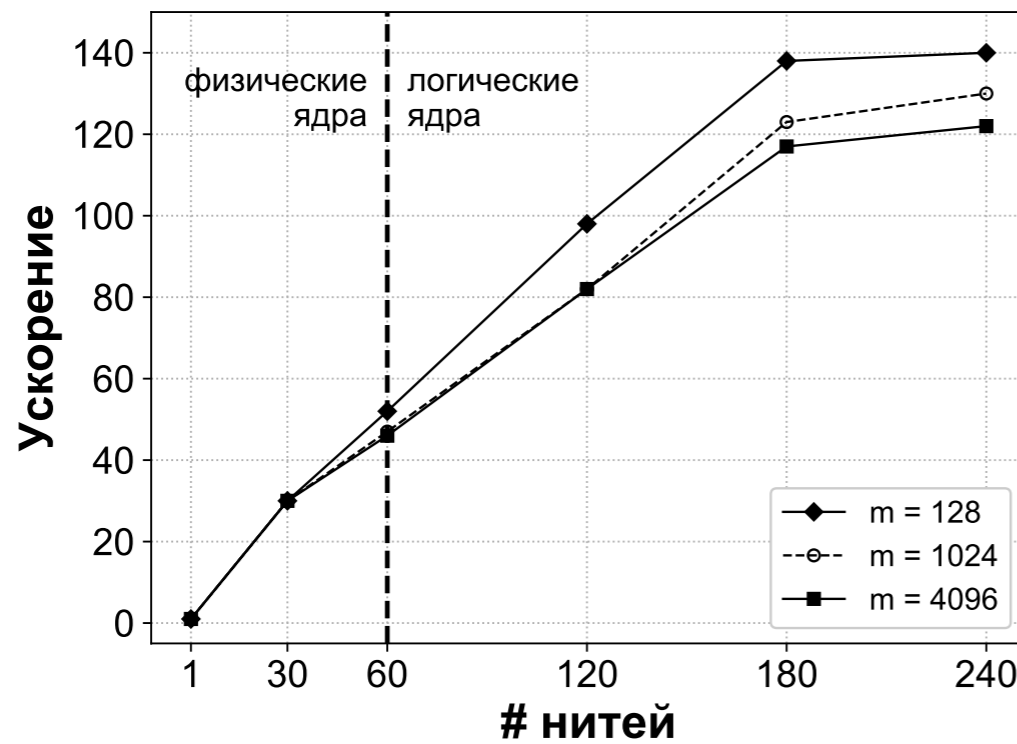
Синтетический ряд, Random Walk



Реальный ряд, ЭКГ



Intel MIC



Публикации, апробация

- **Опубликованы две научные статьи:**

1. Zymbler M., Kraeva Ya. Discovery of Time Series Motifs on Intel Many-Core Systems // Lobachevskii Journal of Mathematics. 2019. Vol. 40, No. 12. P. 2124–2132 (индексируется в **Scopus Q2**).
2. Цымблер М.Л., Краева Я.А. Параллельный алгоритм поиска лейтмотивов временного ряда для графического процессора // Параллельные вычислительные технологии – XIV международная конференция, ПаВТ'2020, г. Пермь, 31 марта – 2 апреля 2020 г. Короткие статьи и описания плакатов. Челябинск: Издательский центр ЮУрГУ, 2020. С. 298–311 (индексируется в **РИНЦ**).

- **Сделан доклад на XIV международной научной конференции «Параллельные вычислительные технологии (ПаВТ) 2020» (31 марта – 2 апреля 2020 г., Пермь).**

Заключение

1. Проведен обзор последовательных и параллельных алгоритмов поиска лейтмотивов временного ряда.
2. Изучены аппаратная архитектура и программная модель многоядерного процессора семейства Intel MIC и графического процессора NVIDIA GPU.
3. Спроектирован и реализован параллельный алгоритм поиска лейтмотивов временного ряда для многоядерных ускорителей.
4. Проведены вычислительные эксперименты на реальных и синтетических данных, показавшие высокую масштабируемость разработанного алгоритма как на платформе Intel MIC, так и на платформе GPU.