Параллельные вычислительные технологии (ПаВТ'2023)
Санкт-Петербург, 28–30 марта 2023 г.

Восстановление пропущенных значений в потоковом временном ряде с применением центрального и графического процессоров

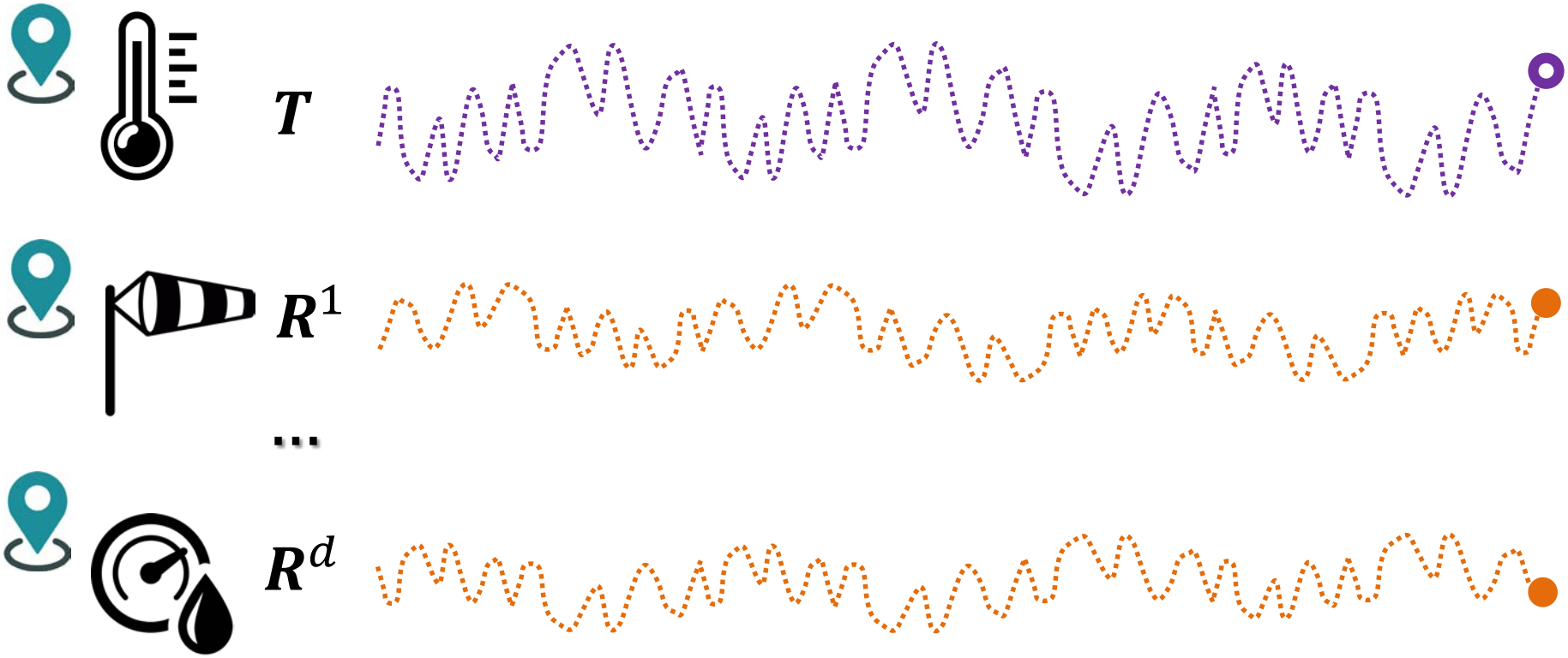
М.Л. Цымблер[1], А.Н. Полуянов[2],

[1]Южно-Уральский государственный университет (Челябинск),

[2]Институт математики им. С.Л. Соболева СО РАН (Омск)

Работа выполнена при финансовой поддержке Российского научного фонда (грант № 23-21-00465)

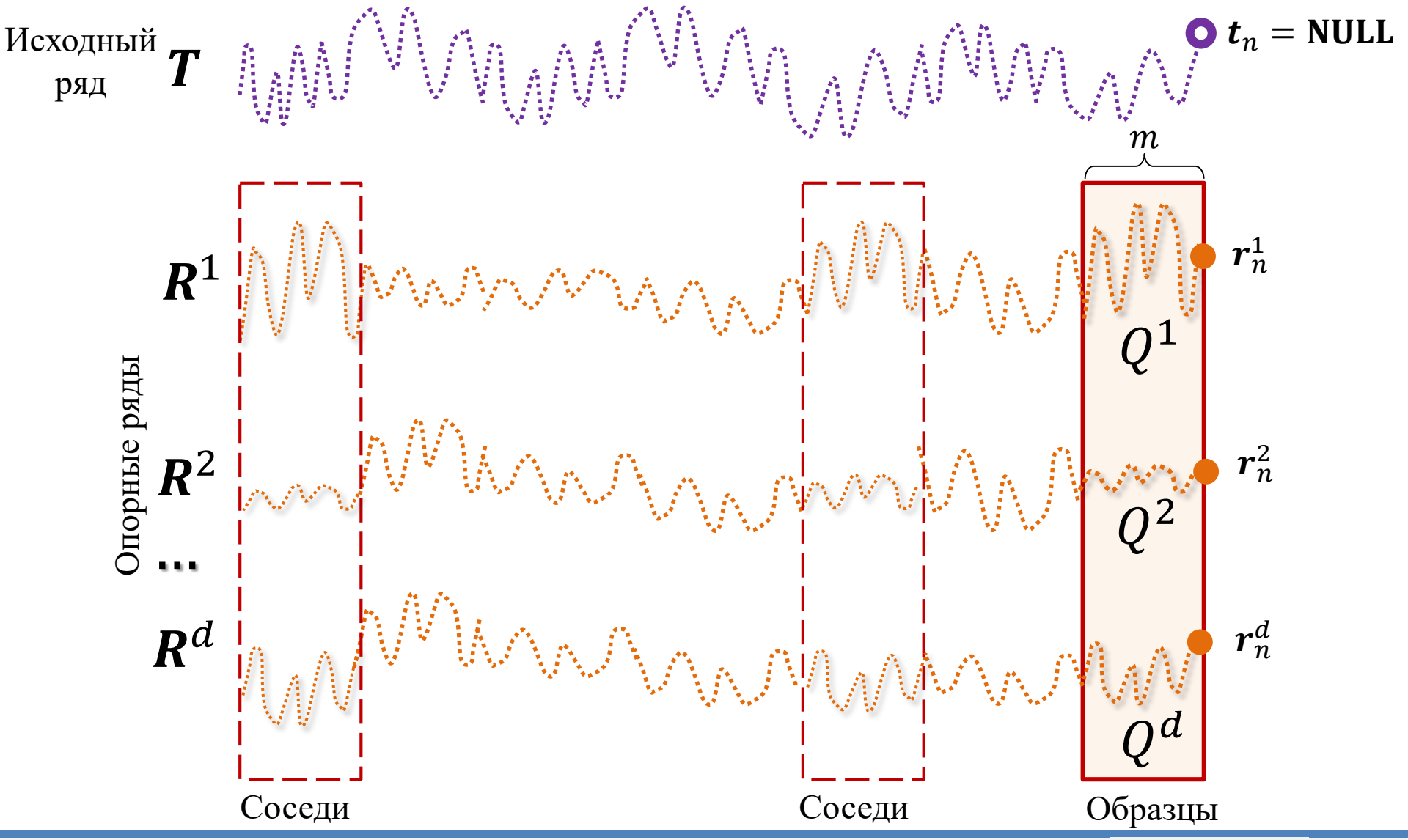
Восстановление по опорным рядам



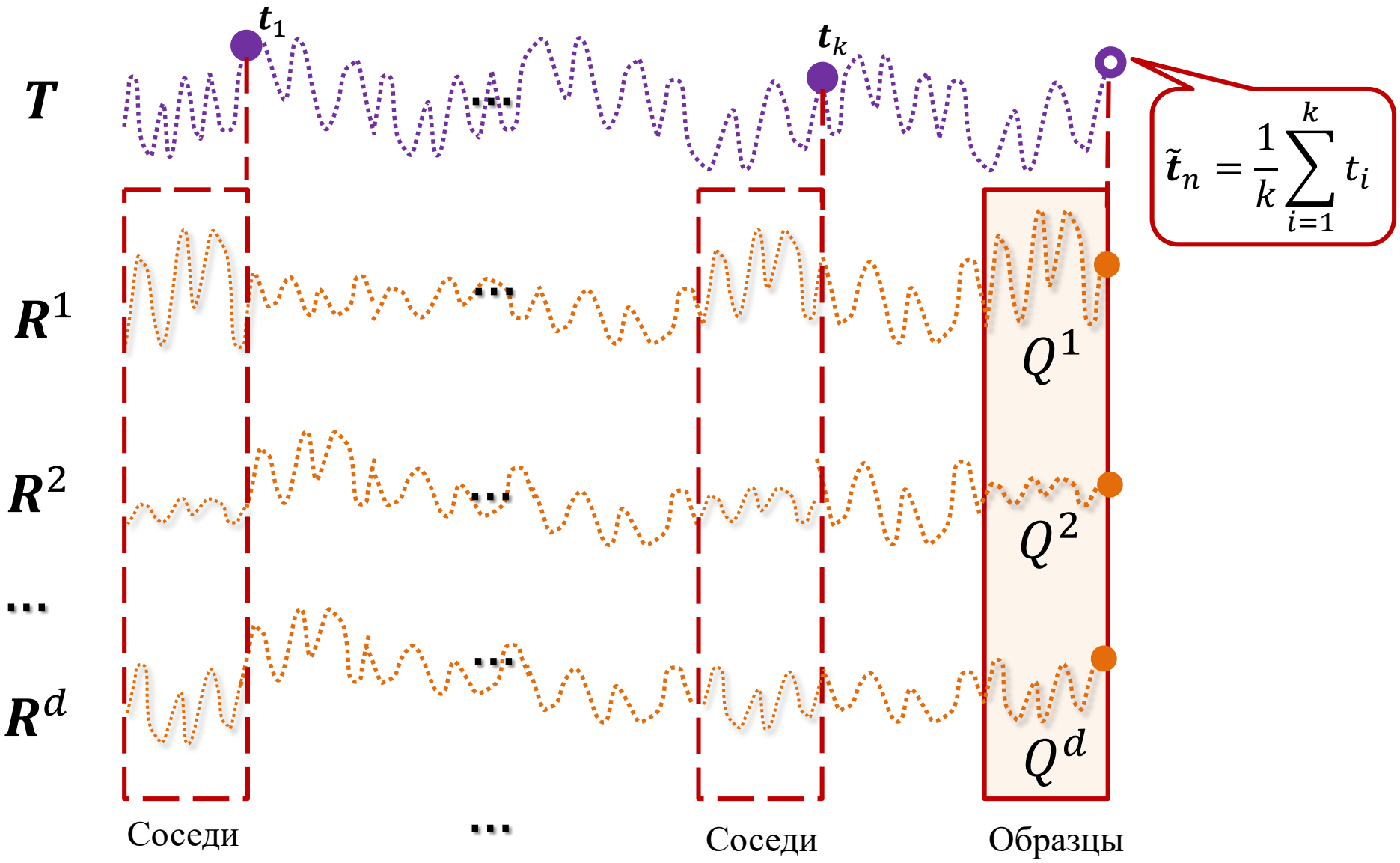
Эвристика

похожие промежутки в исходном ряде возникают в тех же интервалах времени, что и в опорных рядах

Поиск k ближайших соседей



Реконструкция пропущенного значения



ParaDI: Parallel DTW-based Imputation*

ПаВТ’2022 ⟹ ПаВТ’2023

- Версия алгоритма CPU+GPU
  - Вычисление нижних границ на графическом процессоре с применением CUDA
- Инициализация bsf
  - Улучшен выбор начального приближения DTW
- Эксперименты
  - Расширен перечень наборов данных и алгоритмов-конкурентов

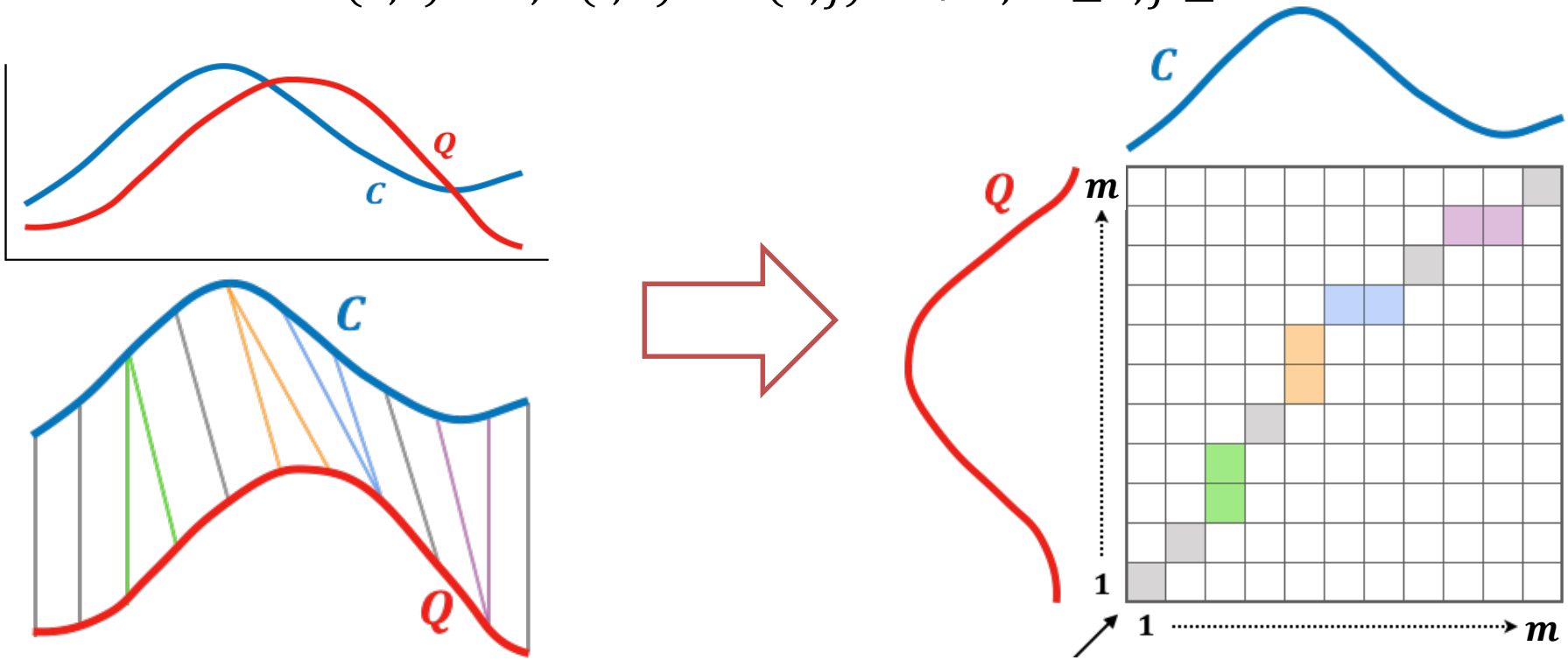
* Цымблер М.Л., Полуянов А.Н., Краева Я.А. Параллельный алгоритм восстановления сенсорных данных в режиме реального времени для многоядерного процессора // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2022. Т. 11, № 3. С. 68–89. DOI: 10.14529/cmse220305.

DTW (Dynamic Time Warping) мера расстояния*

$$\mathbf{DTW}(Q, C) = d(m, m)$$

$$d(i,j) = (q_i - c_i)^2 + \min \begin{cases} d(i-1,j) \\ d(i,j-1) \\ d(i-1,j-1) \end{cases}$$

$$d(0,0) = 0, d(i,0) = d(0,j) = +\infty;\ 1 \le i,j \le m$$



* Berndt D.J., Clifford J. Using Dynamic Time Warping to Find Patterns in Time Series. KDD & AAAI Workshop 1994. TR-WS-94-03. P. 359-370.

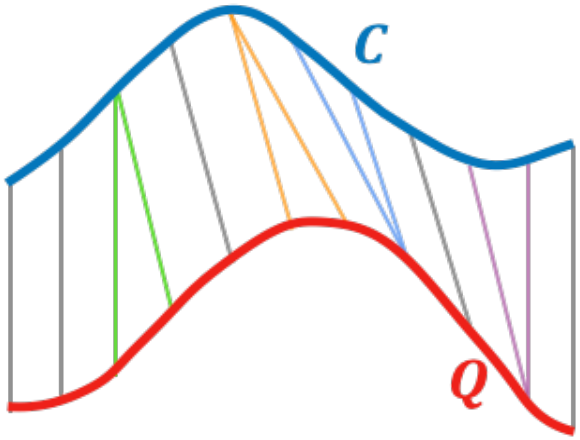
DTW: точность *vs.* сложность

Сложность
$\boldsymbol{O(m^2)}$

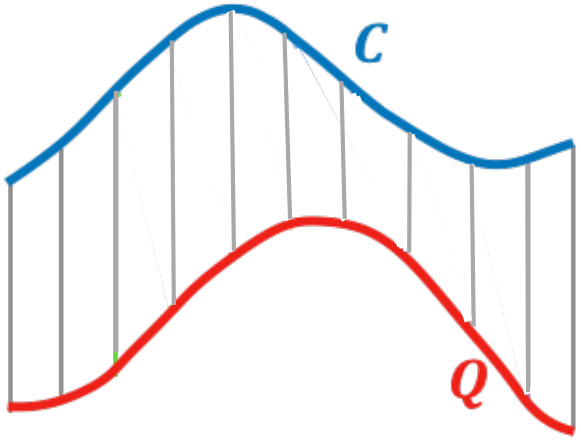
$$\mathbf{DTW}(Q, C) = d(m, m)$$

$$d(i,j) = (q_i - c_i)^2 + \min \begin{cases} d(i-1,j) \\ d(i,j-1) \\ d(i-1,j-1) \end{cases}$$

$$d(0,0) = 0, d(i,0) = d(0,j) = +\infty;\ 1 \le i,j \le m$$



DTW лучше определяет схожесть по форме, чем Евклидова мера



Ускорение подсчета DTW: полоса Сако–Чиба*

Сложность $O(mr)$

$$\mathbf{DTW}(Q, C) = d(m, m)$$

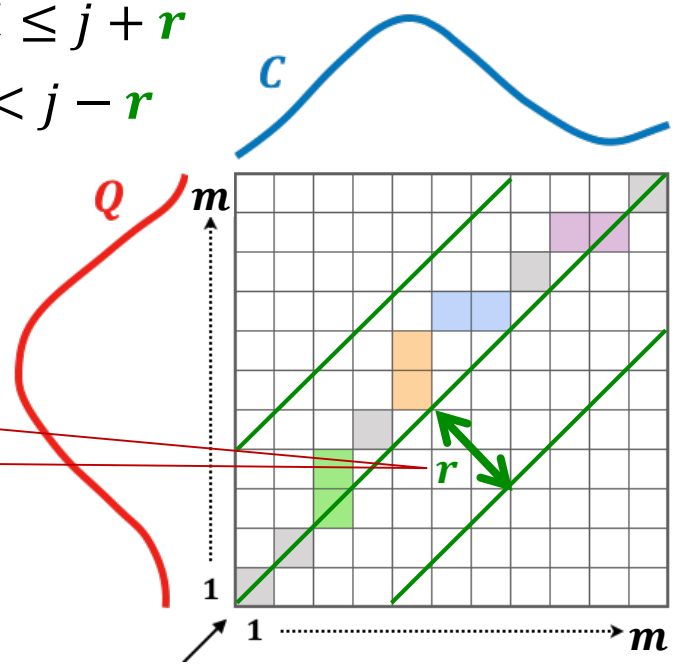
$$d(i,j) = (q_i - c_i)^2 + \min \begin{cases} d(i-1,j) \\ d(i,j-1) \\ d(i-1,j-1) \end{cases}$$

$$d(0,0) = 0, d(i,0) = d(0,j) = +\infty;\ 1 \le i,j \le m;$$

$$\mathbf{0} \le \boldsymbol{r} \le \boldsymbol{m} - \mathbf{1}, j - \boldsymbol{r} \le i \le j + \boldsymbol{r}$$

$$d(i,j) = +\infty, j + \boldsymbol{r} < i < j - \boldsymbol{r}$$

Сокращает сложность за счет огрубления схожести



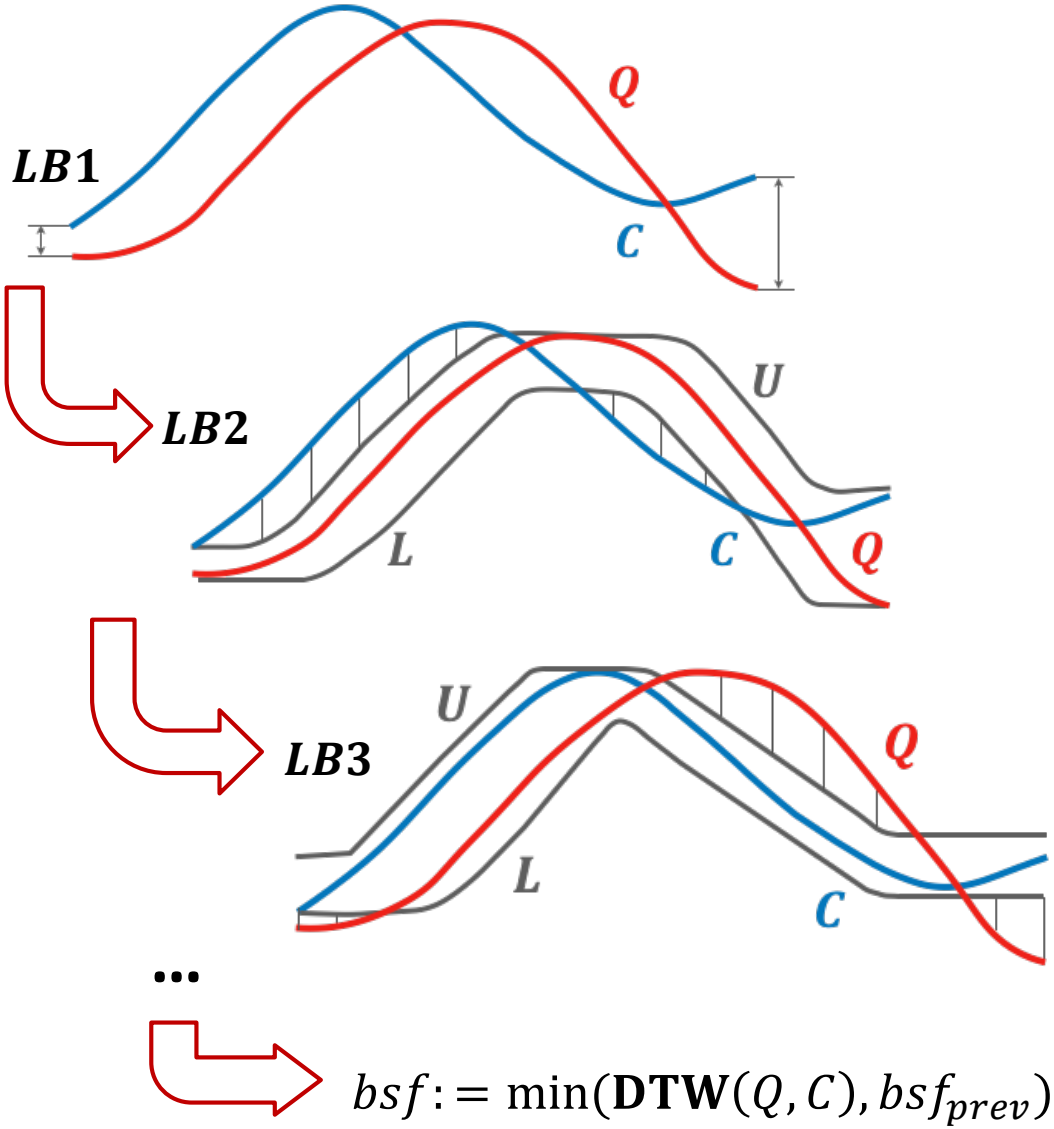
* Sakoe H., Chiba S. Dynamic Programming Algorithm Optimization for Spoken Word Recognition. IEEE Trans. on Acoustics, Speech, and Signal Processing. 1978. Vol. 26. P. 43-49.

Ускорение подсчета DTW: отбрасывание*

- Нижняя граница
  - функция LB: $\mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}_+$ с вычислительной сложностью менее $O(m^2)$
- Текущий минимум DTW
  - bsf (best-so-far)
- Отбрасывание соседей, далеких от образца, без вычисления DTW
  - если $\mathrm{LB}(R[i:m], Q) > bsf$, то $\mathrm{DTW}(R[i:m], Q) > bsf$
- Нормализация
  - Соседей и образец нужно нормализовать

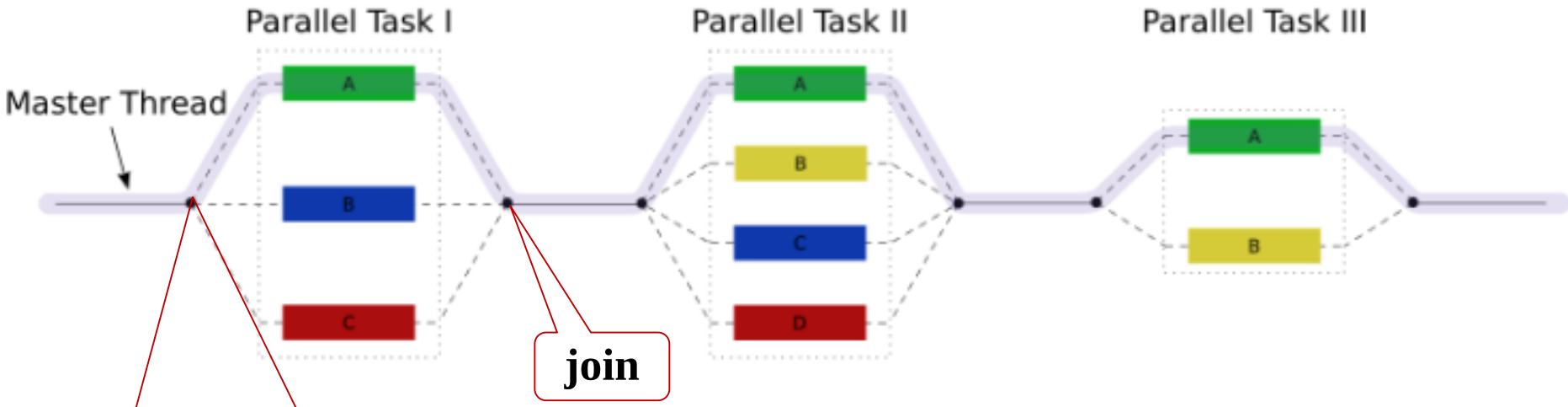
* Rakthanmanon T., *et al.* Addressing Big Data Time Series: Mining Trillions of Time Series Subsequences Under Dynamic Time Warping. ACM Trans. Knowl. Discov. Data. 2013. Vol. 7, no. 3. 10:1–10:31.

Отбрасывание: нижние границы



Нижняя граница	Сложность
LB_KimFL $\boldsymbol{LB1}(Q,C) = (q_1 - c_1)^2 + (q_m - c_m)^2$	$O(1)$
LB_KeoghEC $\boldsymbol{LB2}(Q,C) = \sum_{i=1}^{m} \begin{cases} (c_i - u_i)^2, & c_i > u_i \\ (c_i - \ell_i)^2, & c_i < \ell_i \\ 0, & otherwise \end{cases}$ $u_i = \max_{i-r \le k \le i+r} q_k$ $\ell_i = \min_{i-r \le k \le i+r} q_k$	$O(m)$
LB_KeoghEQ $\boldsymbol{LB3}(Q,C) = \boldsymbol{LB2}(C,Q)$	$O(m)$
LB_Yi, LB_PAA, LB_FTW, ...	...
$\mathbf{DTW}(Q,C)$	$O(mr)$

Параллелизм fork-join

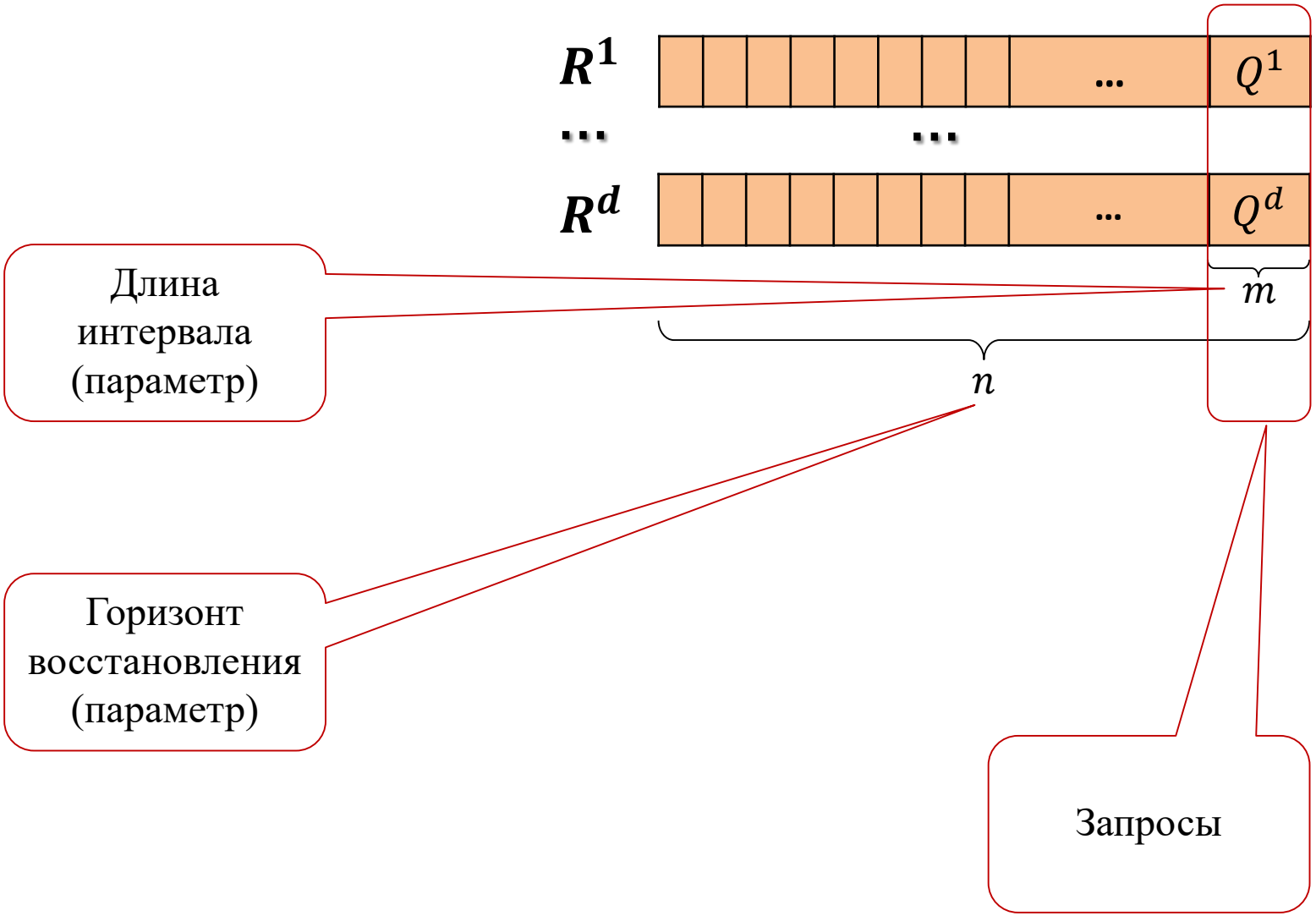


fork

```
CPU: #pragma omp
GPU: kernel<<<blocks,threads>>>
```

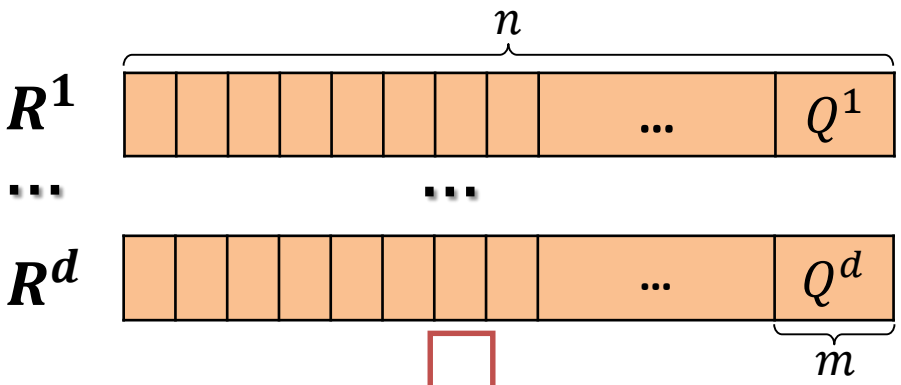
- Многопоточность
  - OpenMP for x86
  - CUDA for GPU

ParaDI: Опорные ряды



ParaDI: поиск соседей по образцу

Расстояние между интервалом $R^j[i:m]$ и образцом Q^j в смысле меры DTW (Dynamic Time Warping)

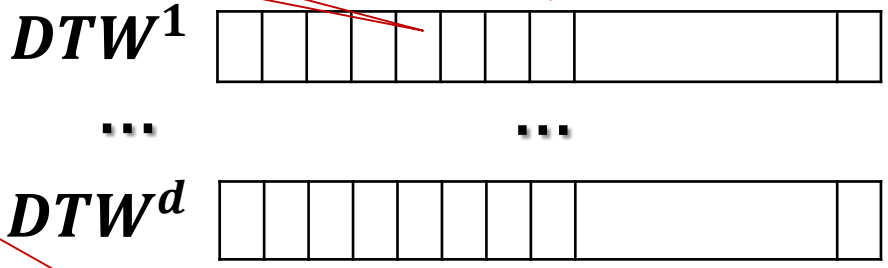


1. Поиск соседей по образцу

for $i := 1$ **to** d **do**

for $j := 1$ **to** $n - m + 1$ **do**

$DTW^i(R^i[j:m], Q^i)$



Зеленый выполняется **параллельно OpenMP**

ParaDI: скоринг интервалов

1. Поиск соседей по образцу

for $i \coloneqq 1$ **to** d **do**

  for $j \coloneqq 1$ **to** $n - m + 1$ **do**

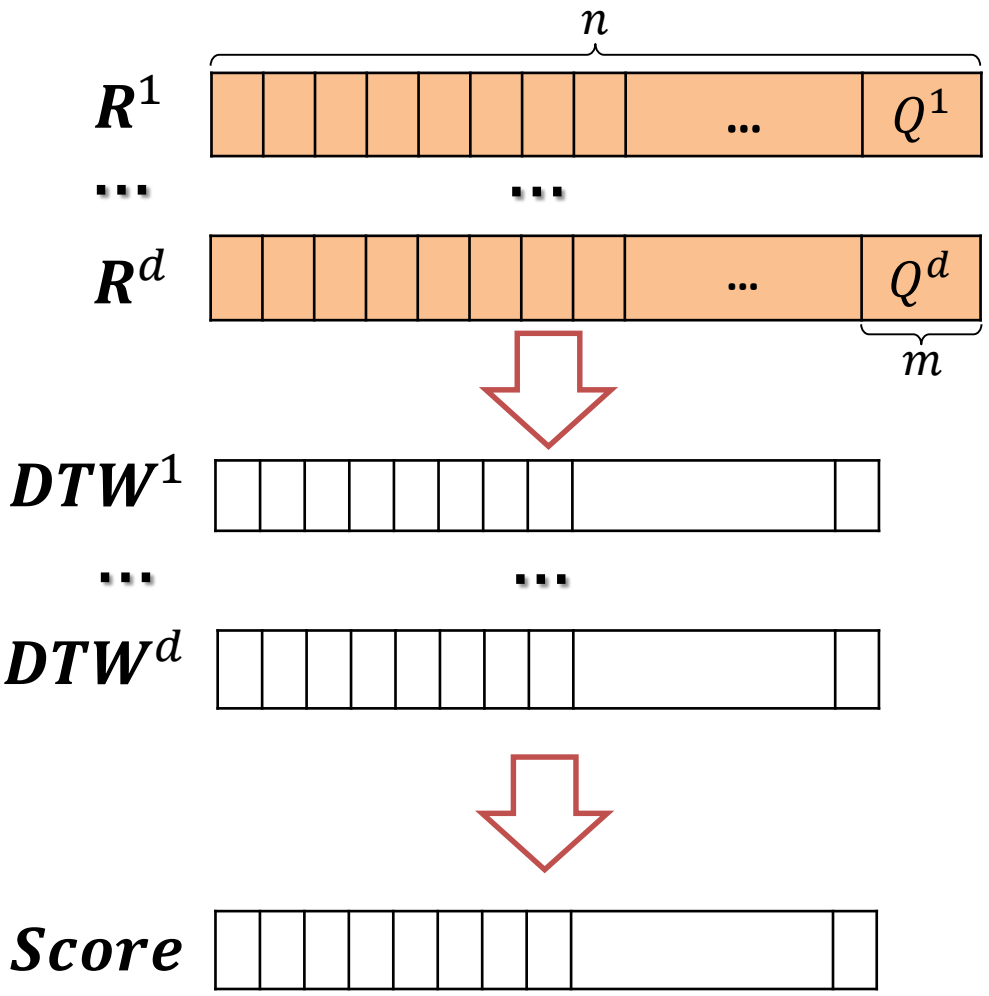
    $DTW^i(R^i[j{:}m], Q^i)$

2. Скоринг интервалов

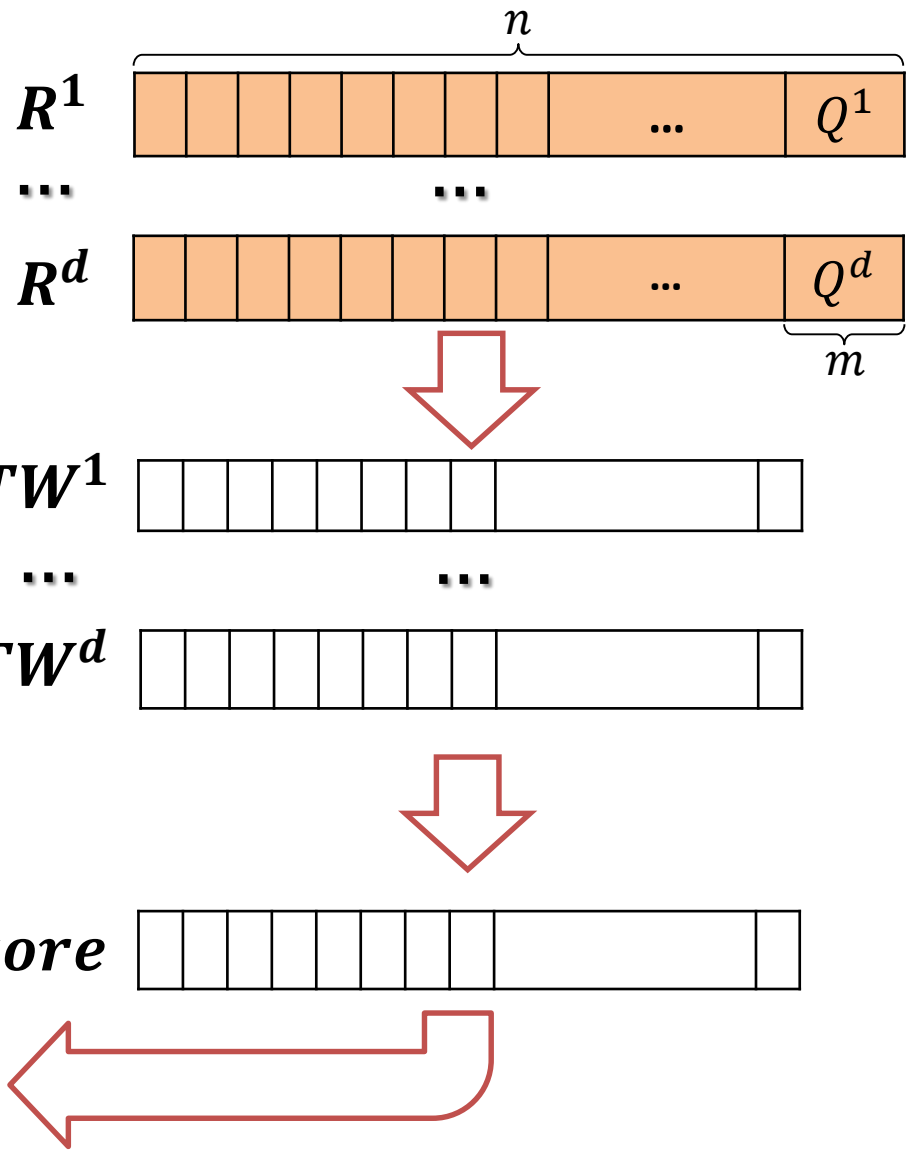
for $j \coloneqq 1$ **to** $n - m + 1$ **do**

  for $i \coloneqq 1$ **to** d **do**

    $Score(j) \coloneqq Score(j) + \frac{weight(j,i)}{DTW^i(j)+\varepsilon}$



ParaDI: Реконструкция



1. Поиск соседей по образцу

for $i \coloneqq 1$ **to** d **do**

for $j \coloneqq 1$ **to** $n - m + 1$ **do**

$DTW^i(R^i[j:m], Q^i)$

2. Скоринг интервалов

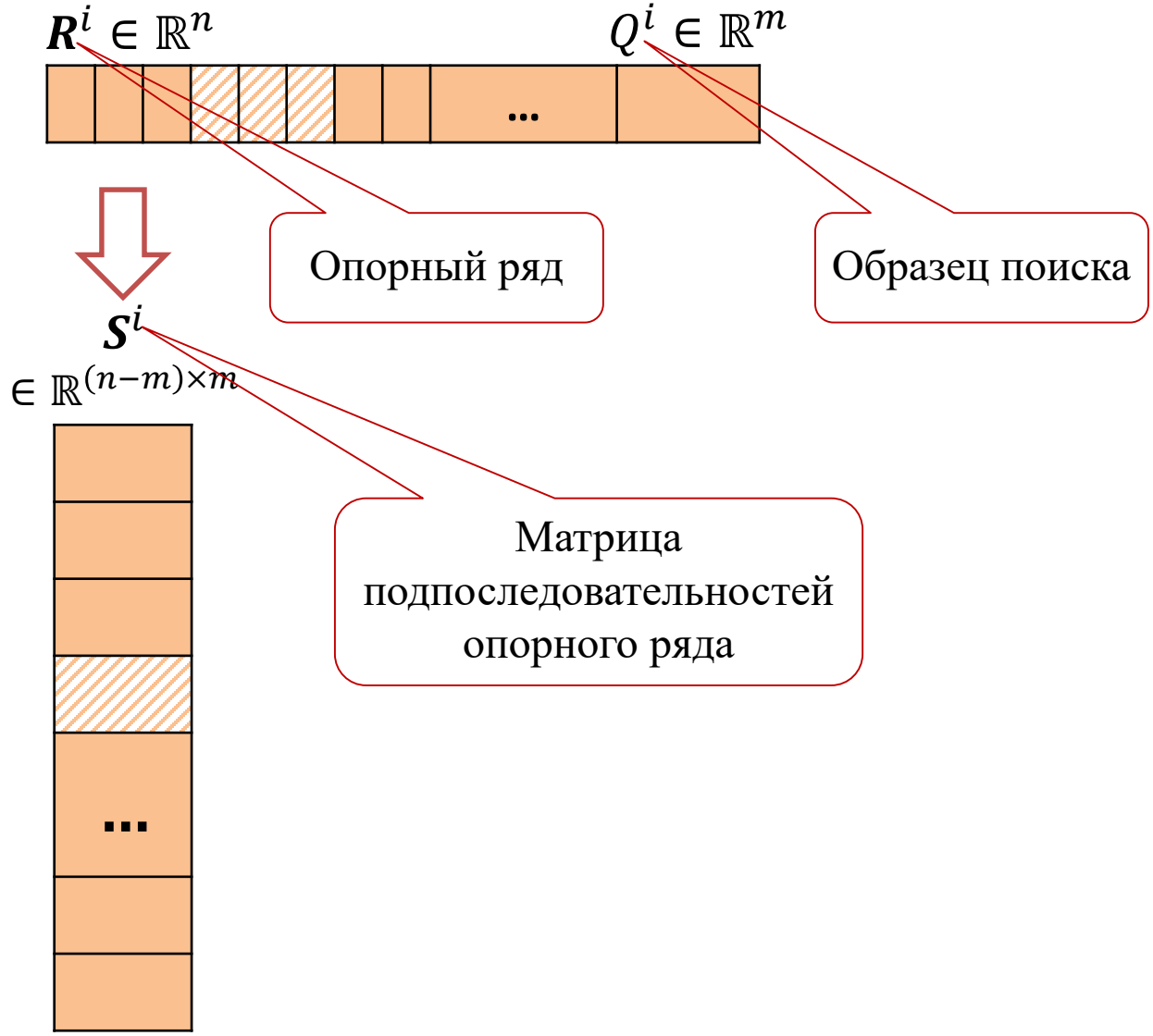
for $j \coloneqq 1$ **to** $n - m + 1$ **do**

for $i \coloneqq 1$ **to** d **do**

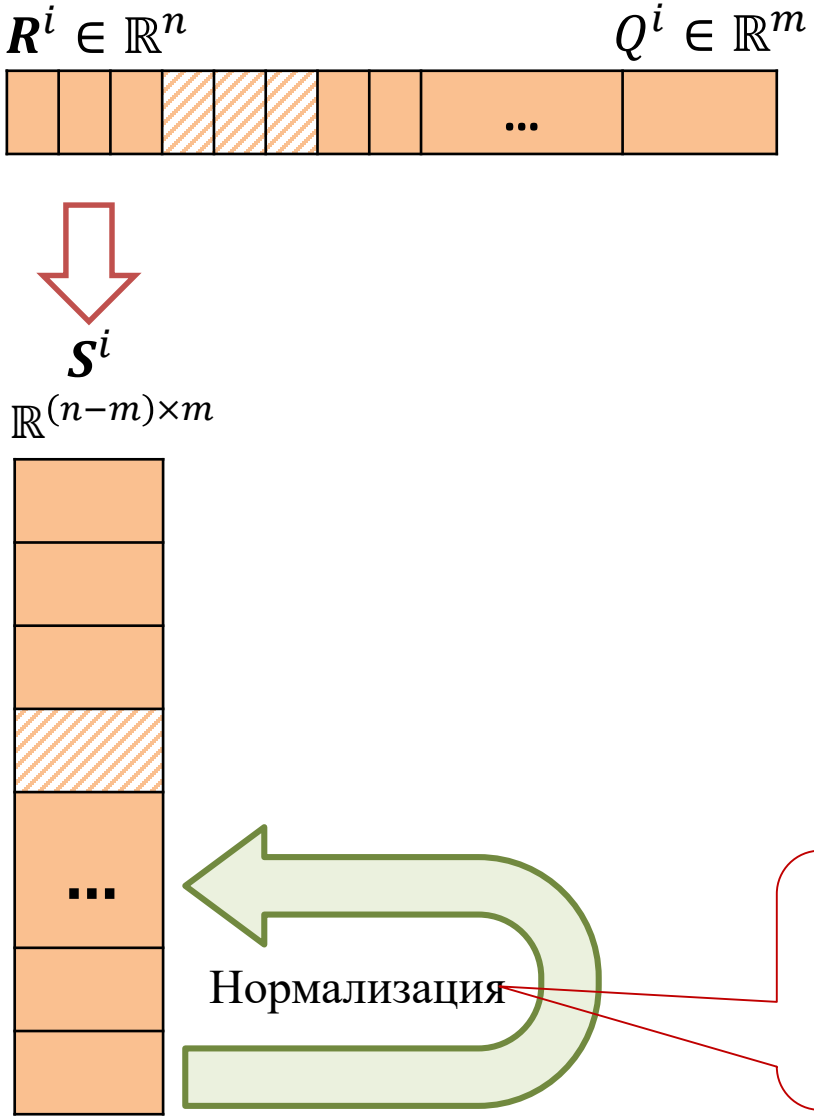
$$Score(j) \coloneqq Score(j) + \frac{weight(j,i)}{DTW^i(j)+\varepsilon}$$

3. Отбор TOP-k интервалов, реконструкция

Параллельный поиск соседей по образцу

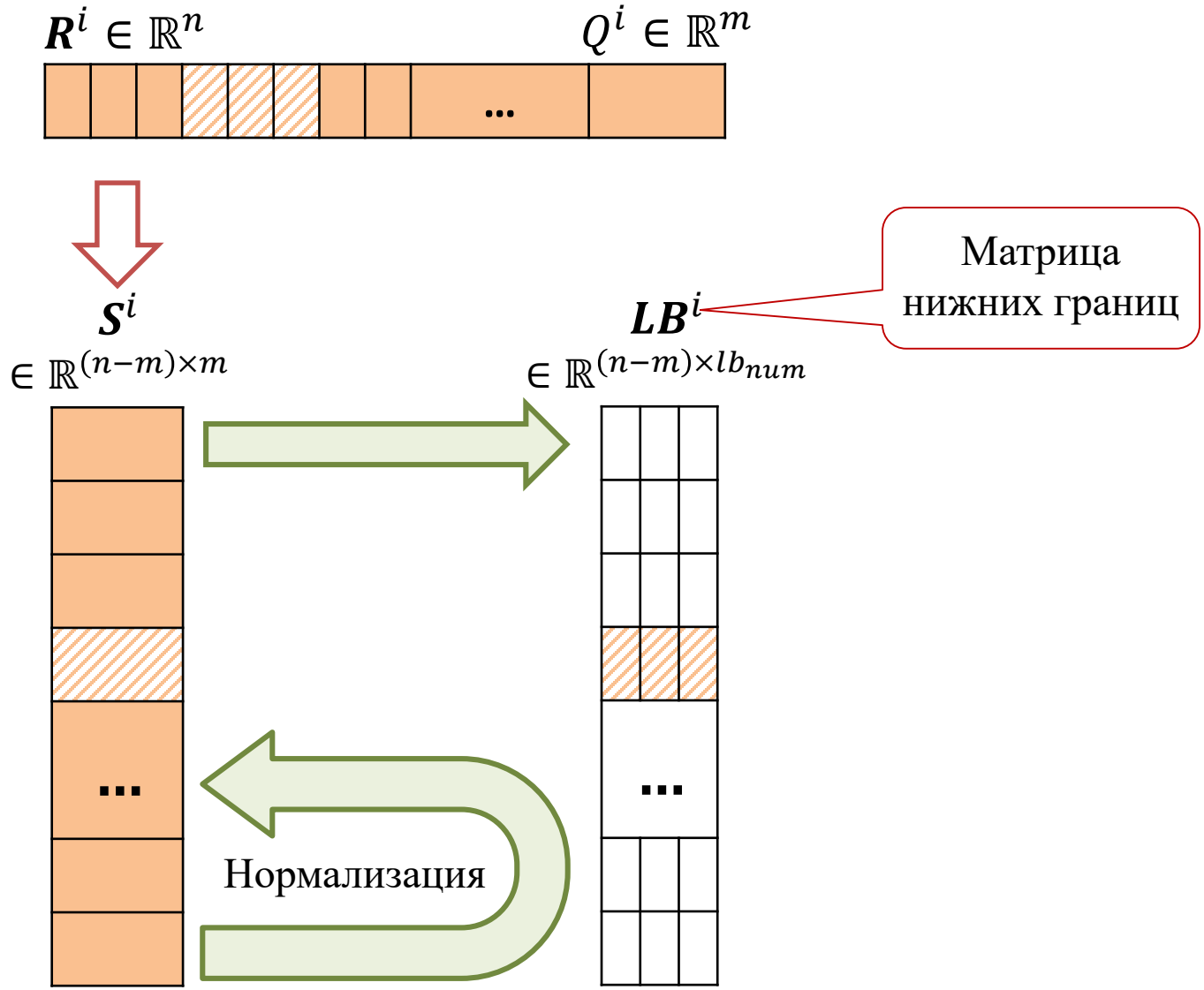


ParaDI: Нормализация

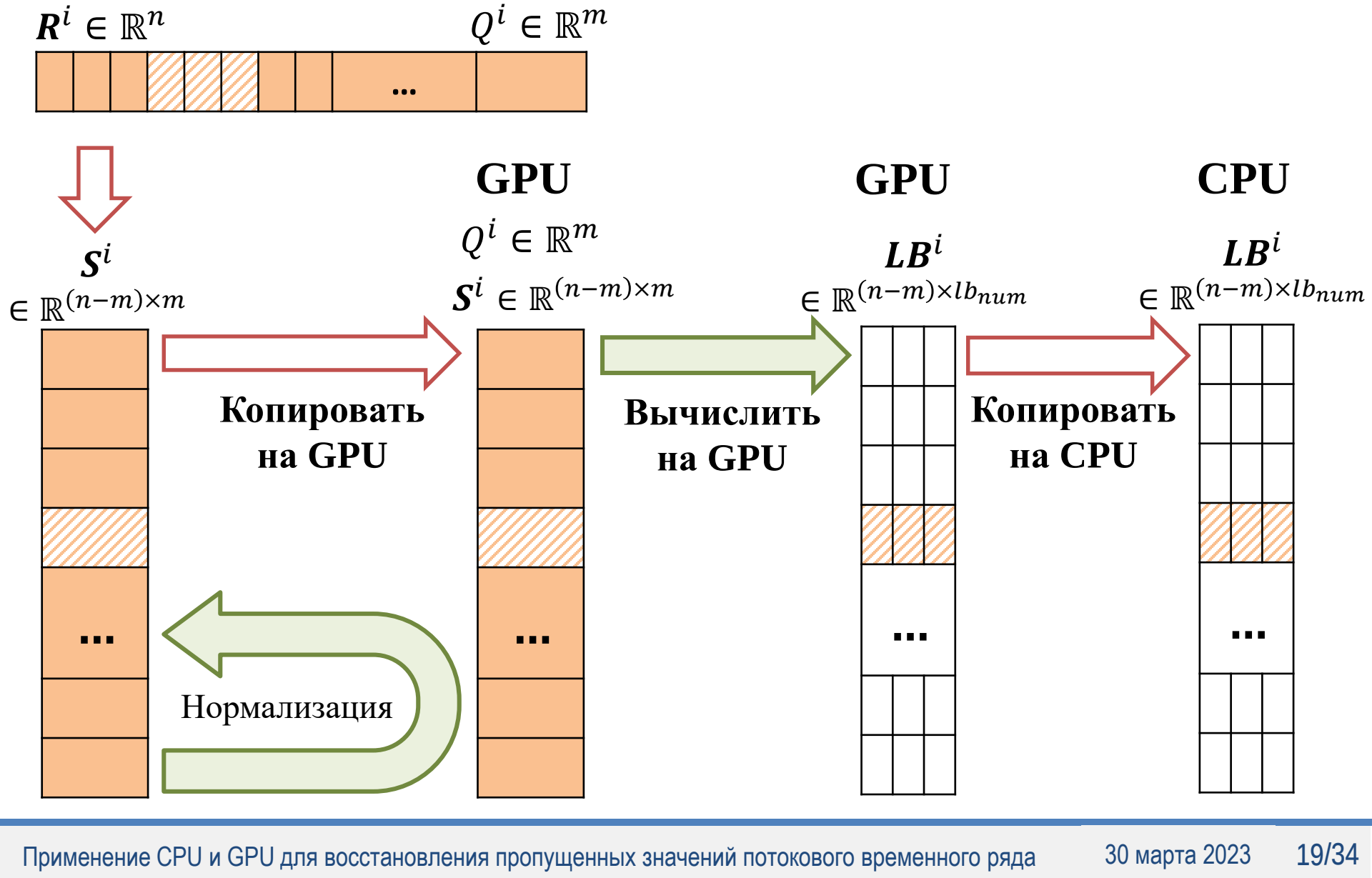


$$R[1:m] \Rightarrow \hat{R}\,[1:m], \qquad \hat{r}_i = \frac{r_i - \mu}{\sigma},$$
$$\mu = \frac{1}{m}\sum_{i=1}^{m} r_i, \qquad \sigma^2 = \frac{1}{m}\sum_{i=1}^{m} r_i^2 - \mu^2$$

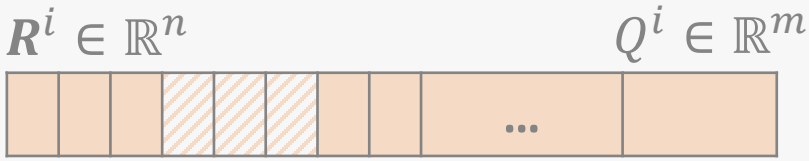
ParaDI: Нижние границы



ParaDI: Вычисление нижних границ на GPU



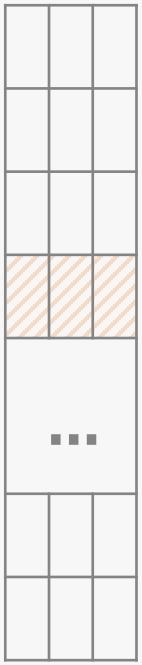
ParaDI: Инициализация bsf



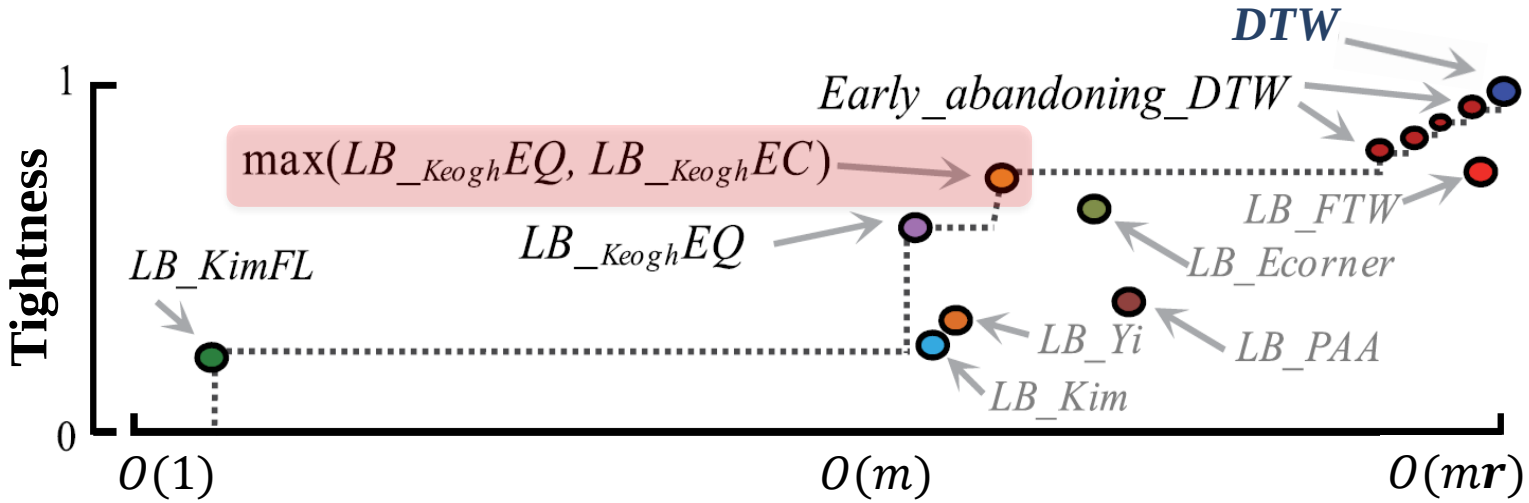
$$bsf_{init} := \mathrm{DTW}(Q^i, C)$$

$$C = \arg \min_{1 \le p \le n-m} \max_{1 \le j \le lb_{num}} \boldsymbol{LB}_j^i(Q^i, \boldsymbol{R}[p:m])$$

$$\boldsymbol{LB}^i \in \mathbb{R}^{(n-m) \times lb_{num}}$$

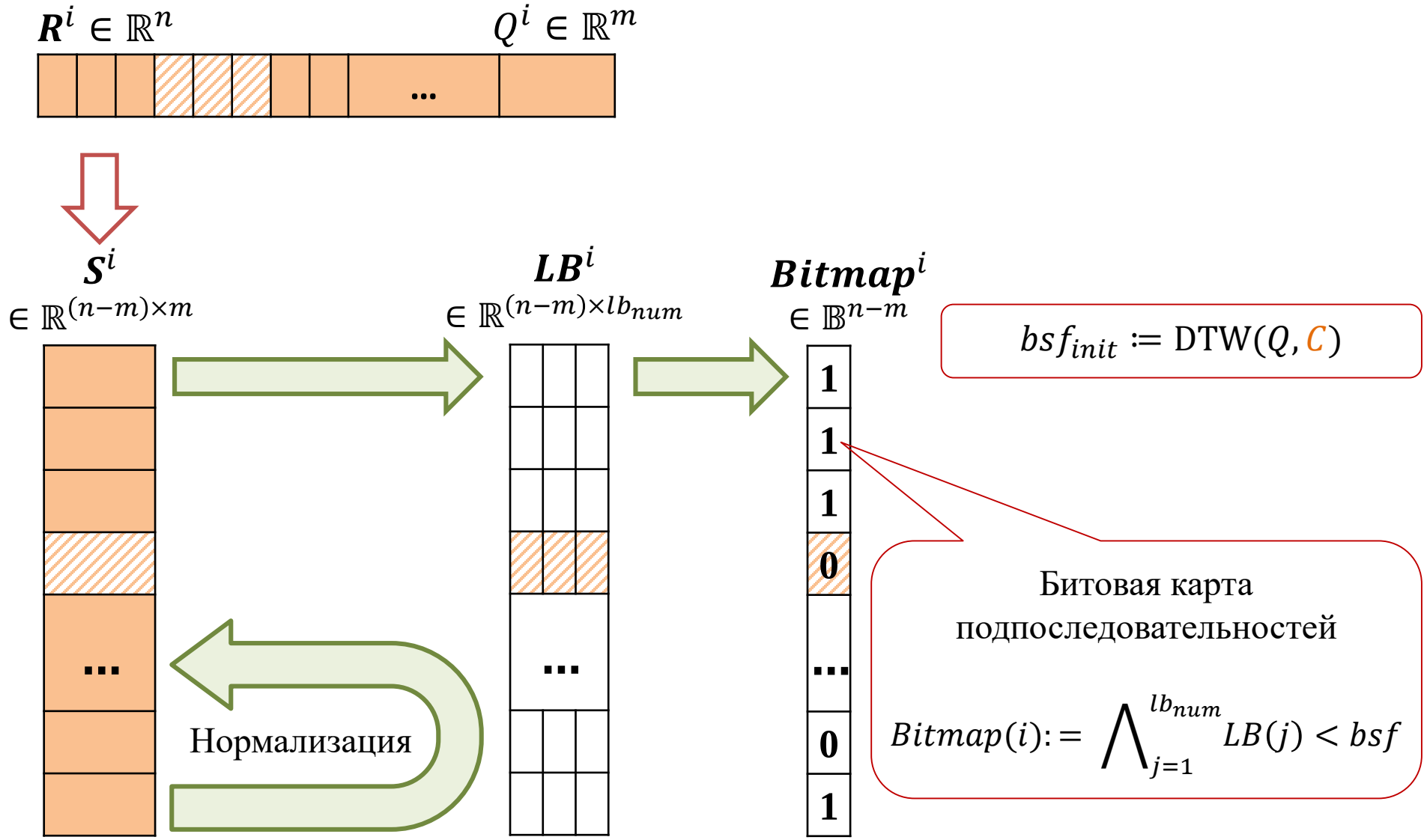


$$Tightness^* = \frac{\boldsymbol{LB}(C,Q)}{\boldsymbol{DTW}(C,Q)}$$

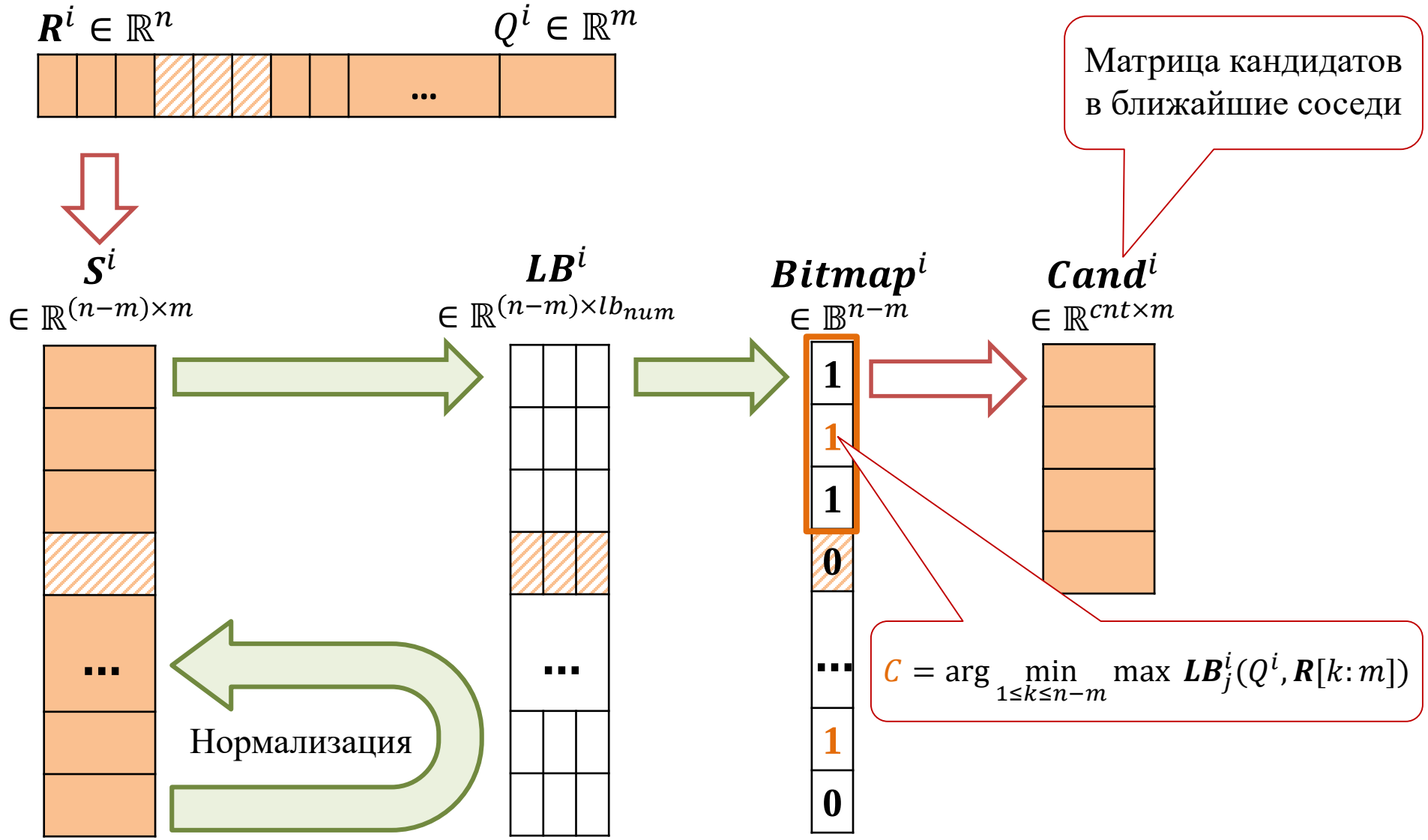


* Rakthanmanon T., *et al.* Addressing big data time series: Mining trillions of time series subsequences under Dynamic Time Warping. ACM Trans. Knowl. Discov. Data. 2013. Vol. 7, no. 3. 10:110:31. DOI: 10.1145/2500489.

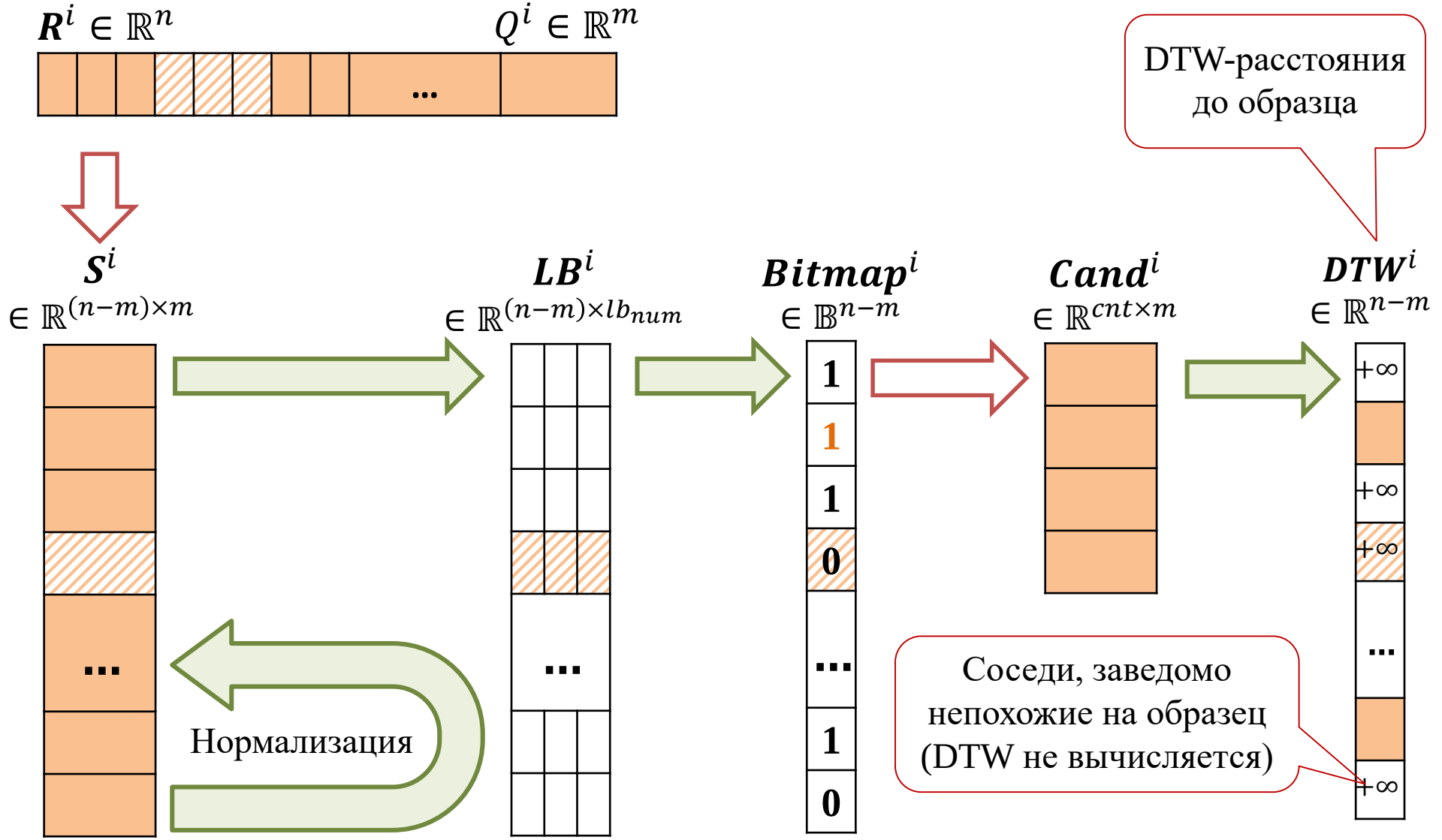
ParaDI: Битовая карта



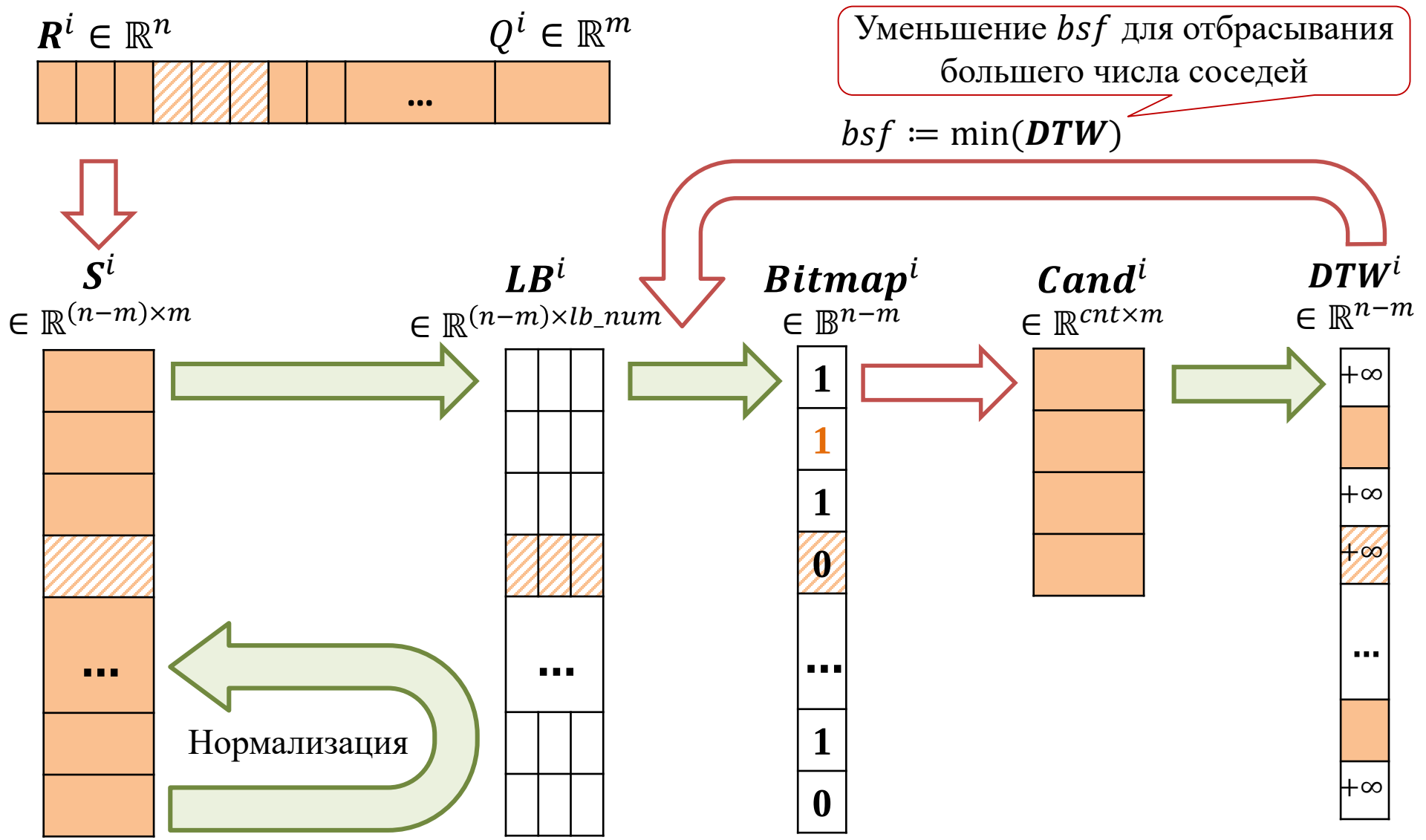
ParaDI: Матрица кандидатов



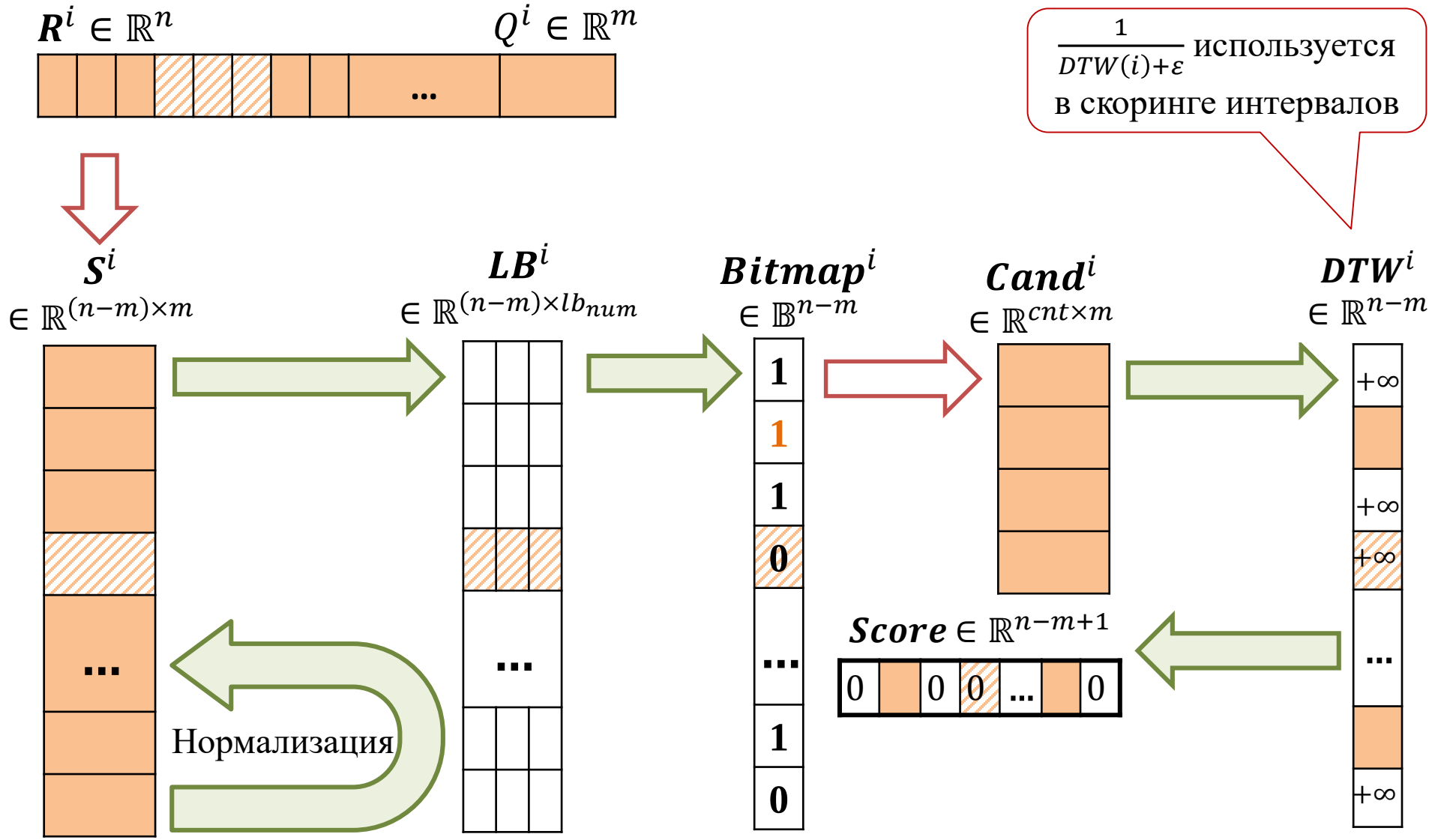
ParaDI: Вычисление DTW



ParaDI: Улучшение порога bsf



ParaDI: Скоринг



Эксперименты: платформа и данные

- **CPU:** Intel Xeon E5-2687W v2 (8 ядер @3.40 GHz)
- **GPU:** NVIDIA Ampere A100 PCIe (6912 ядер CUDA, 80 GB)
- **Данные**

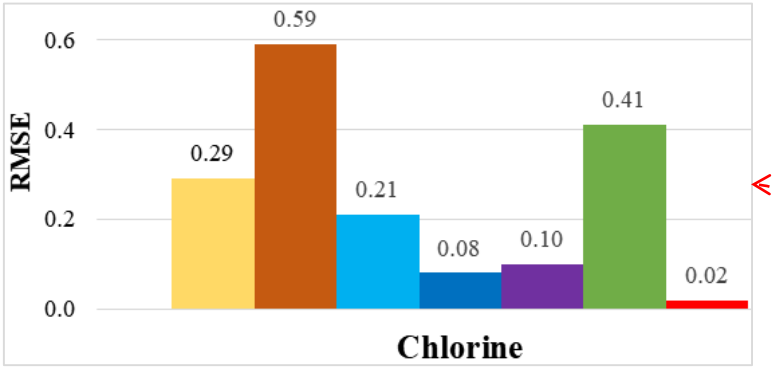
Набор	# рядов, $d+1$	Длина, $n \cdot 10^3$	Предметная область
BAFU	10	50	Сброс воды в реках Швейцарии
Chlorine	50	1	Моделирование концентрации хлора в питьевой воде
Climate	10	5	Погода в различных локациях Северной Америки
MADRID	10	25	Трафик автодорог Мадрида
MAREL	10	50	Характеристики морской воды в проливе Ла-Манш

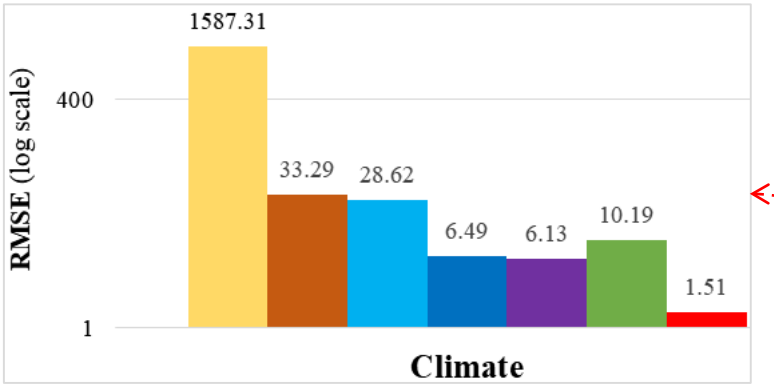
- **Конкуренты**: ORBITS[1], OGD-Impute[2], SPIRIT[3], SAGE[4], TKCM[5]
- **Параметры** (фреймворк ORBITS[1]):
  - Сценарий: пропуски в 10% последних точек
  - Конкуренты: лучшие параметры, рекомендованные их авторами
  - ParaDI: $m = 50$, $k = 3$, $r = 0.25m$

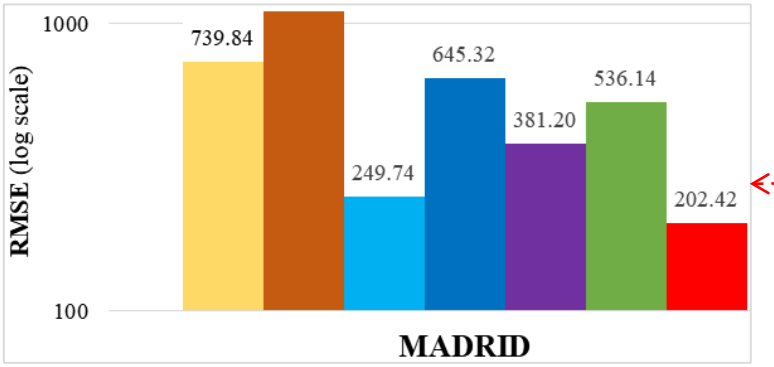
[1] Khayati M., *et al.* ORBITS: Online Recovery of Missing Values in Multiple Time Series Streams. Proc. VLDB Endow. 2020. Vol. 14, no. 3. P. 294-306.
[2] Anava O., *et al.* Online Time Series Prediction with Missing Data. Proc. ICML 2015. P. 2191-2199.
[3] Papadimitriou S., *et al.* Streaming Pattern Discovery in Multiple Time-Series. VLDB 2005. P. 697-708.
[4] Balzano L., *et al.* Streaming PCA and Subspace Tracking: The Missing Data Case. Proc. of IEEE. 2018. Vol. 106, no. 8. P. 1293-1310.
[5] Wellenzohn K., *et al.* Continuous Imputation of Missing Values in Streams of Pattern-Determining Time Series. EDBT 2017. P. 330-341.

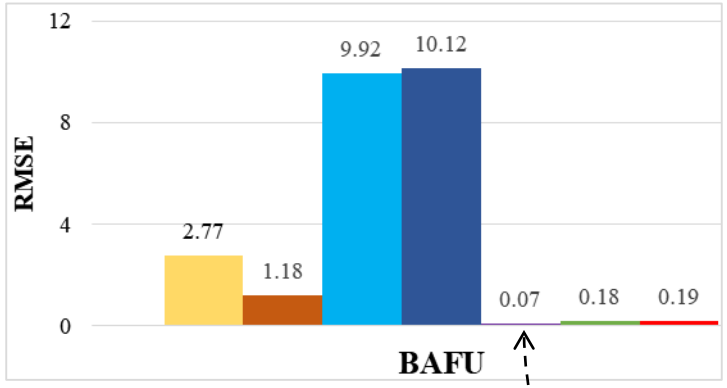
Эксперименты: точность

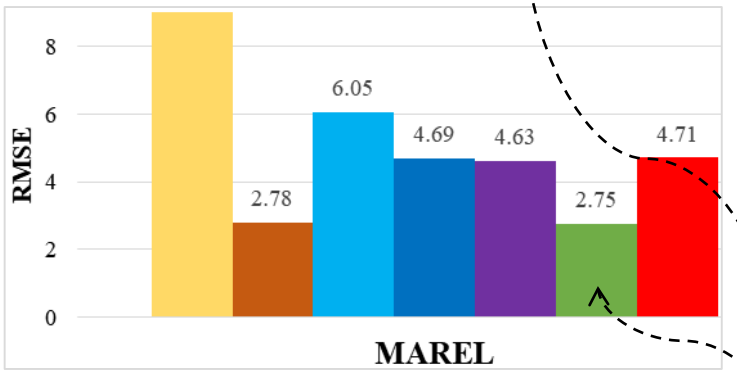
$$\mathrm{RMSE} = \sqrt{\frac{1}{h}\sum_{i=1}^{h}(t_i - \tilde{t}_i)^2}$$







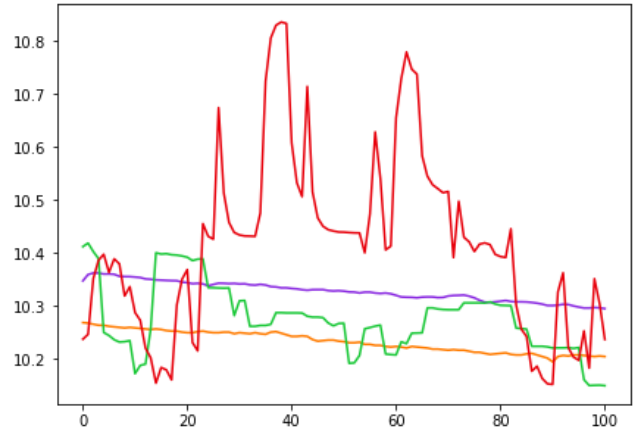




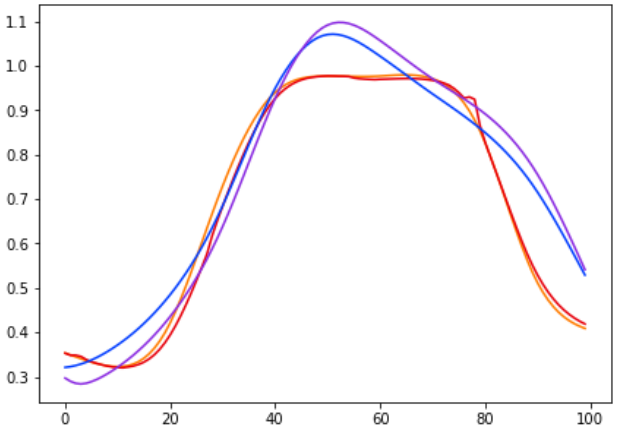
ParaDI точнее, чем аналоги
(но не всегда: SPIRIT на BAFU, TKCM on MAREL

SAGE OGD-Impute ORBITS (k=2) ORBITS (k=3)
SPIRIT TKCM **ParaDI**

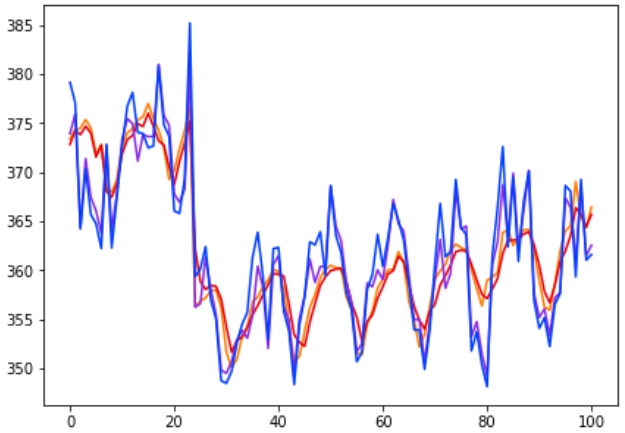
Эксперименты: восстановление 100 точек*



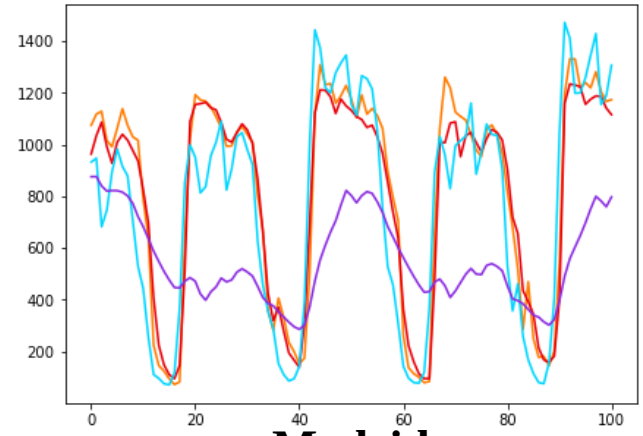
BAFU



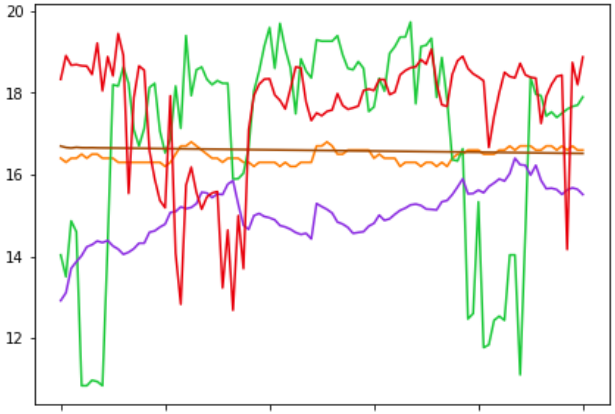
Chlorine



Climate



Madrid

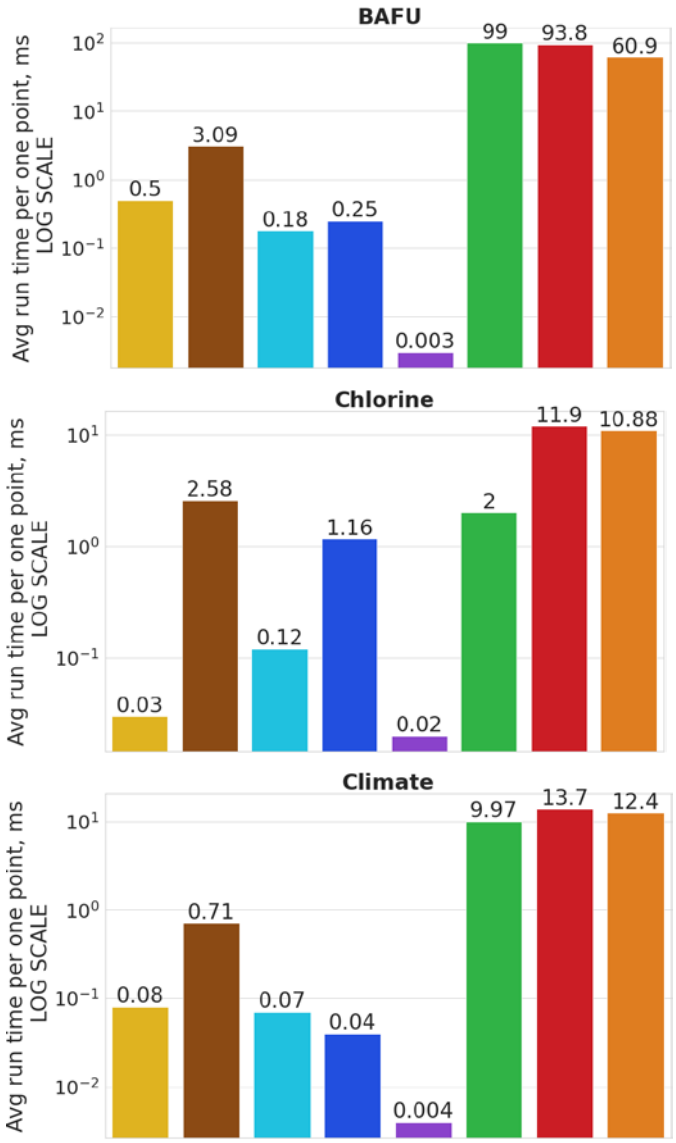


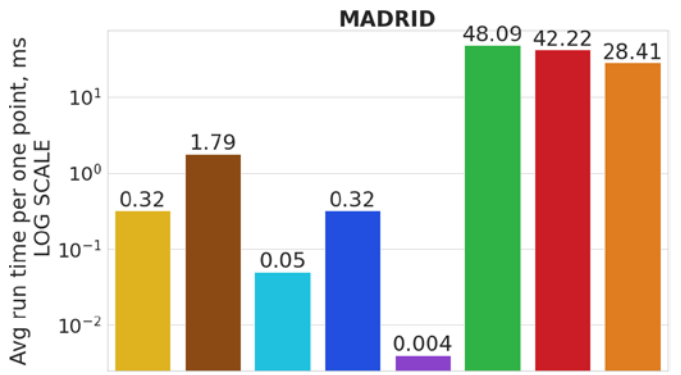
MAREL

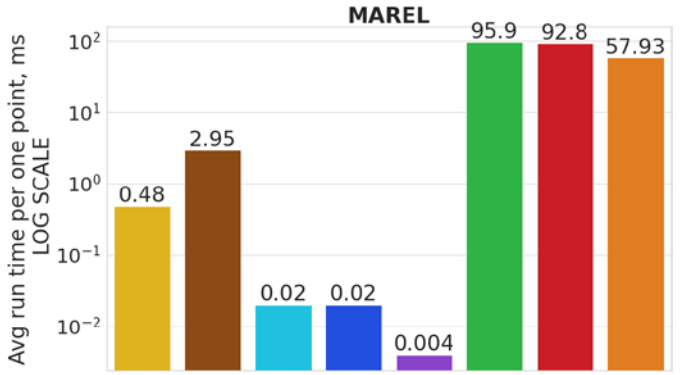
SAGE
OGD-Impute
SPIRIT
TKCM
ORBITS (k=2)
ORBITS (k=3)
ParaDI
реальные данные

* Показаны исходные данные и результаты восстановления трех лучших по точности алгоритмов

Эксперименты: производительность







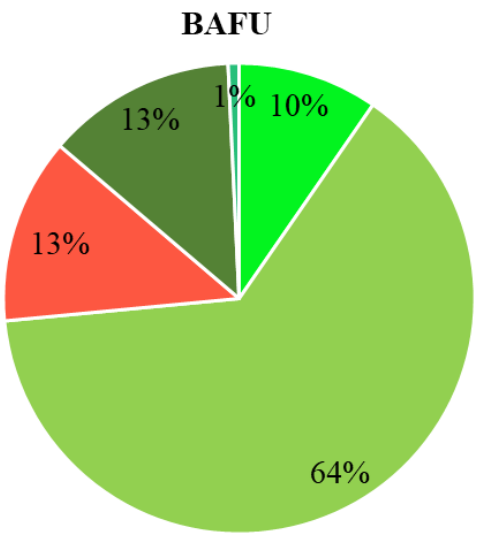
GPU ускоряет ParaDI до 1.5 раз

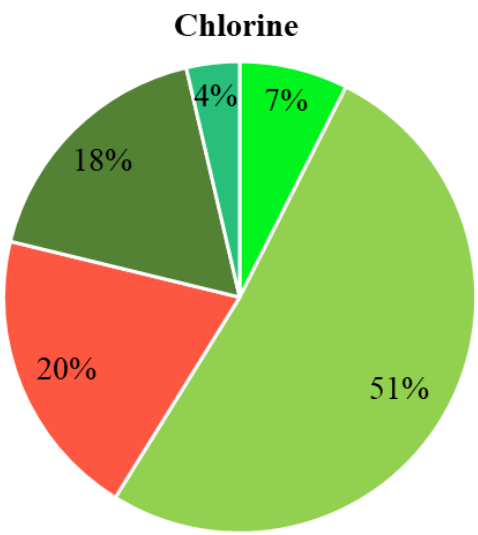
ParaDI медленнее аналогов, кроме TKCM

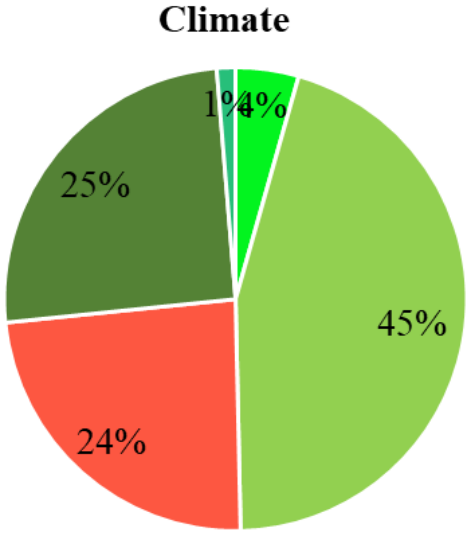
SAGE | OGD-Impute | ORBITS (k=2) | ORBITS (k=3)

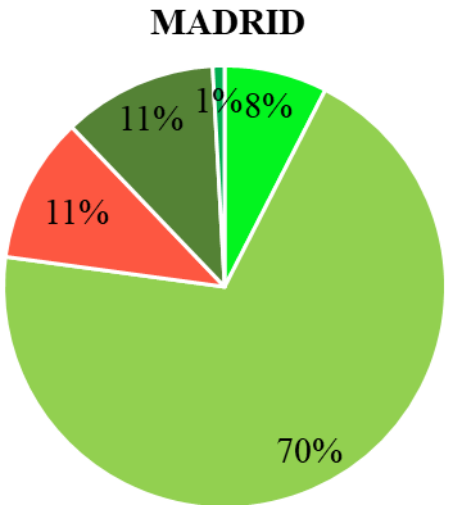
SPIRIT | TKCM | **ParaDI CPU** | **ParaDI CPU+GPU**

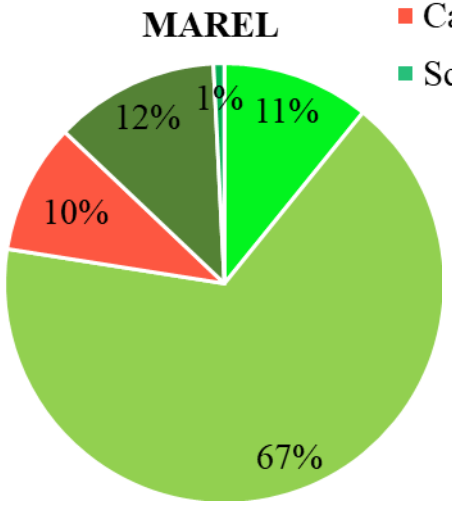
Эксперименты: разбивка по фазам











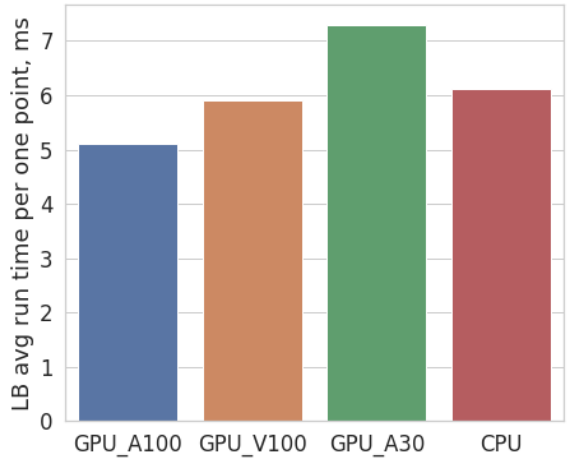
- Z-normalization
- Calculation of LBs
- Candidate selection
- Calculation of DTW
- Scoring and reconstruction

Top-3 фаз по трудоемкости:

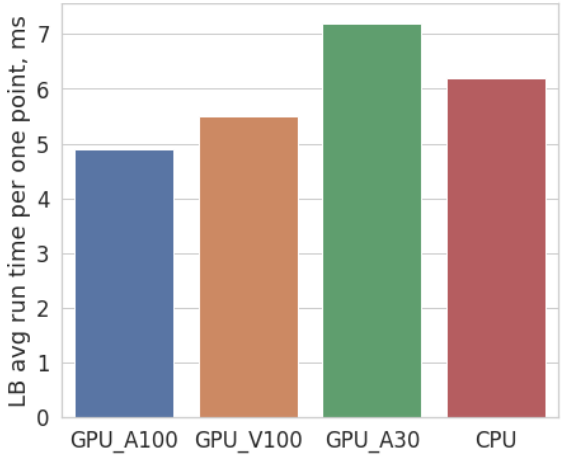
1. **Вычисление LB**
2. **Вычисление DTW**
3. **Отбор кандидатов**

Эксперименты: время вычисления LB

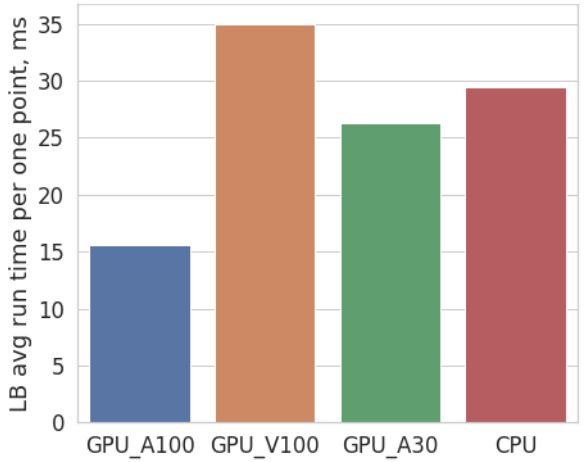
Chlorine



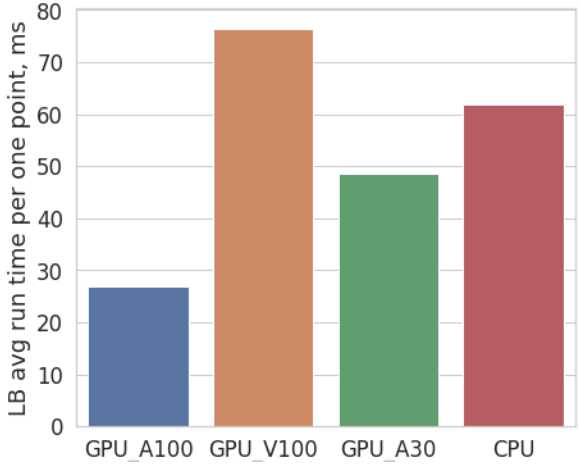
Climate



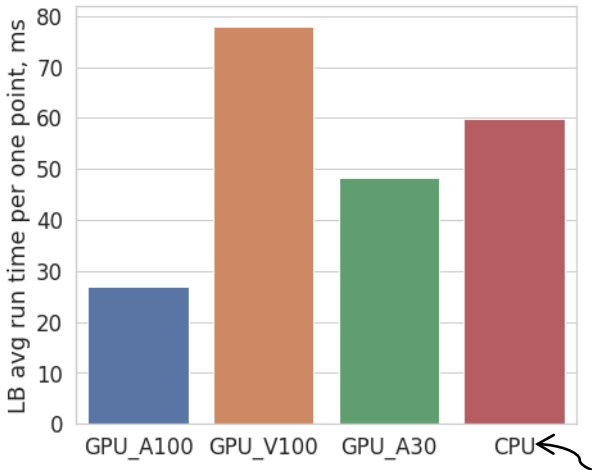
Madrid



MAREL



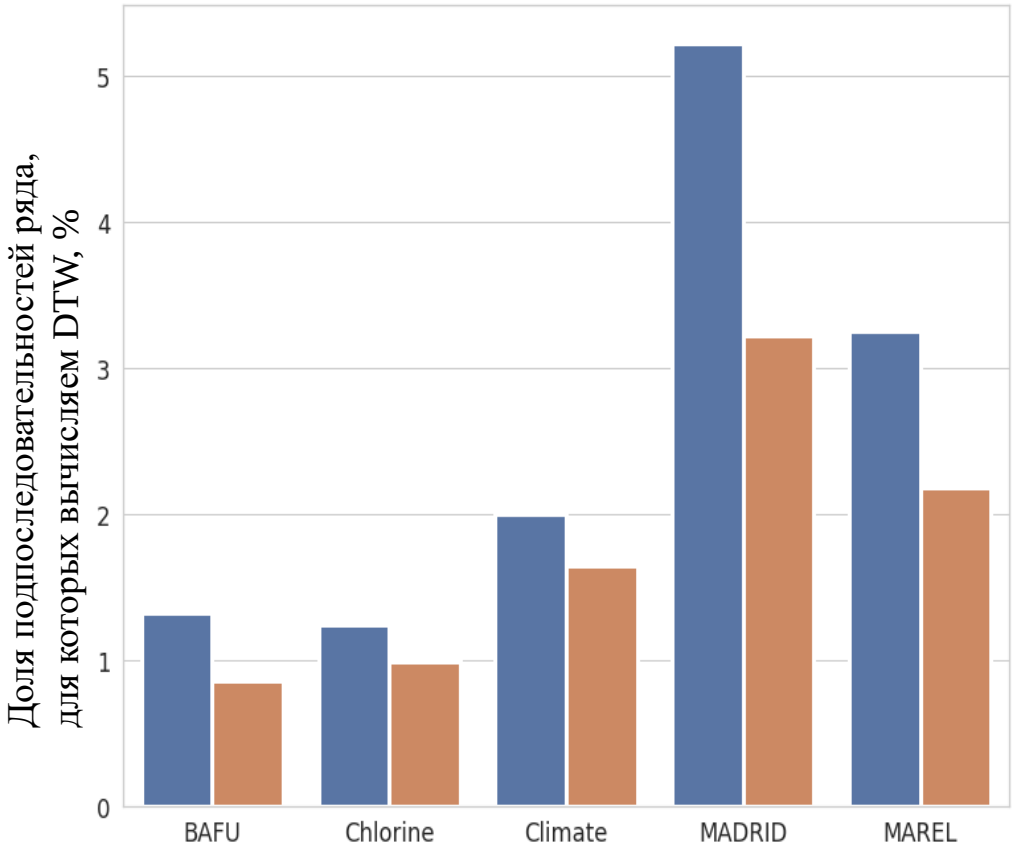
BAFU



GPU ускоряет вычисление нижних границ (самую трудоемкую часть алгоритма) до 2 раз

Intel Xeon E5-2687W v2 (8 ядер @3.40 GHz)

Эффективность применения нижних границ



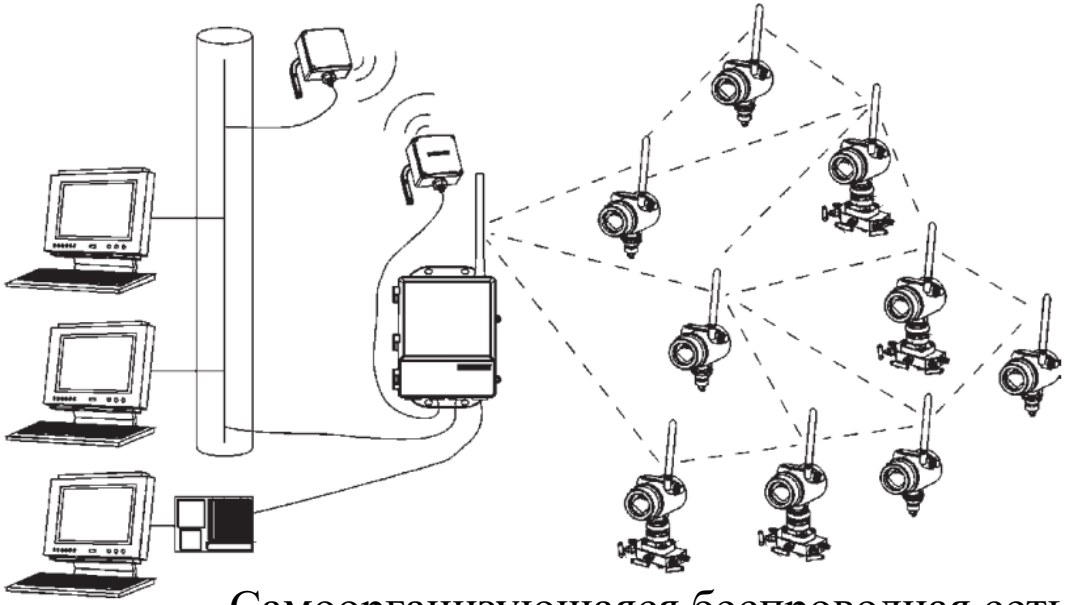
Случайное значение bsf_{init}

$bsf_{init} := \mathrm{DTW}(Q^i, C)$

$$C = argmin \max_{1 \le j \le lb_{num}} \boldsymbol{LB}_j^i(Q^i, \boldsymbol{R}[p:m])$$

Отброшено 96.7–99.1% расчетов DTW

ParaDI: режим реального времени



Самоорганизующаяся беспроводная сеть

Wireless Gateway 1420*		**Время ParaDI, мсек**	
# сенсоров	min время обновления, **сек**	худшее	лучшее
100	8	60.9	0.8
50	4		
12	1		

ParaDI подходит для режима реального времени

* Emerson temperature sensors catalogue 2021. URL: https://www.c-o-k.ru/library/catalogs/emerson/110477.pdf

Заключение

- ParaDI: параллельный CPU+GPU алгоритм восстановления пропусков во временных рядах
  - Точность: лучше многих, но не всех конкурентов
  - Скорость: ниже конкурентов, но приемлемо для режима реального времени
  - Применение: цикличные, сезонные временные ряды
- Будущие исследования:
  - Калибровка алгоритма для автоматического подбора параметров n, m, r, k

Спасибо за внимание! Вопросы?
Михаил Цымблер, Андрей Полуянов