Параллельная реализация поиска самой похожей подпоследовательности временного ряда для систем с распределенной памятью*

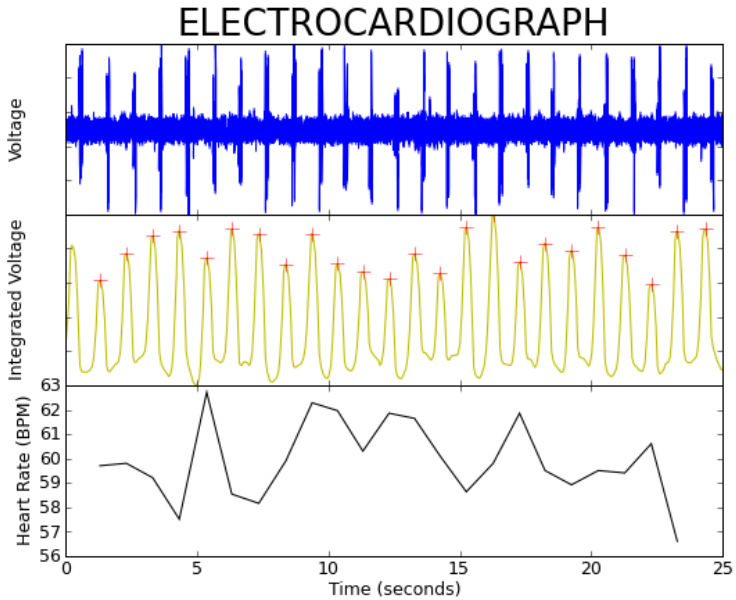
Александр Вячеславович Мовчан, Михаил Леонидович Цымблер

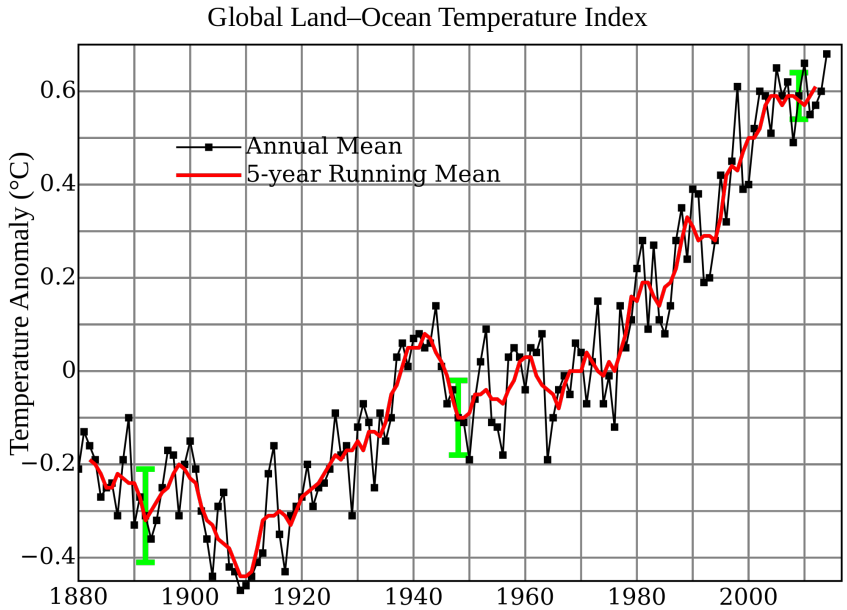
Южно-Уральский государственный университет (НИУ), Челябинск

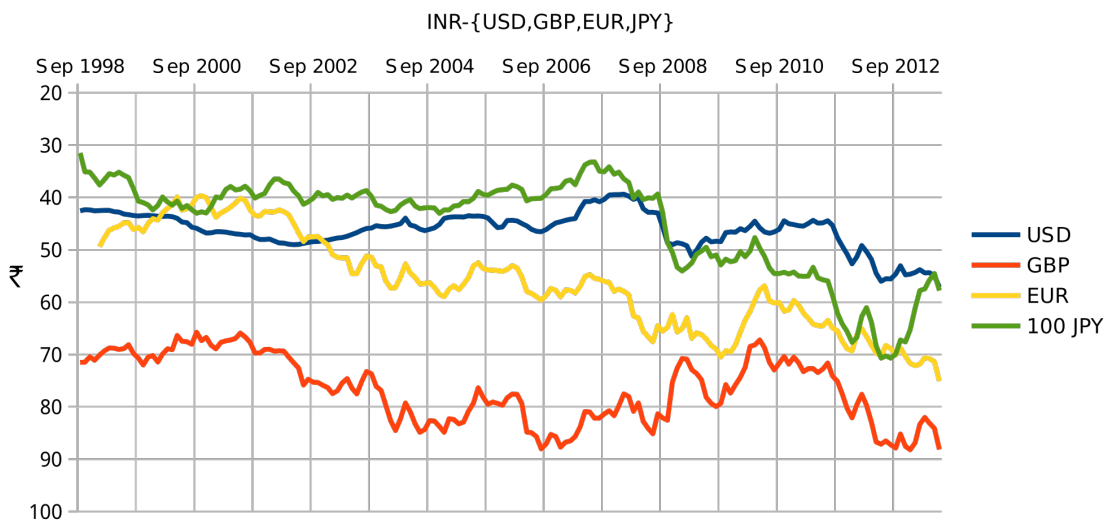
Параллельные вычислительные технологии (ПаВТ) 2016
Архангельск, 29–31 марта 2016

* Работа выполнена при финансовой поддержке Минобрнауки России в рамках ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2014–2020 гг.» (Госконтракт 14.574.21.0035).

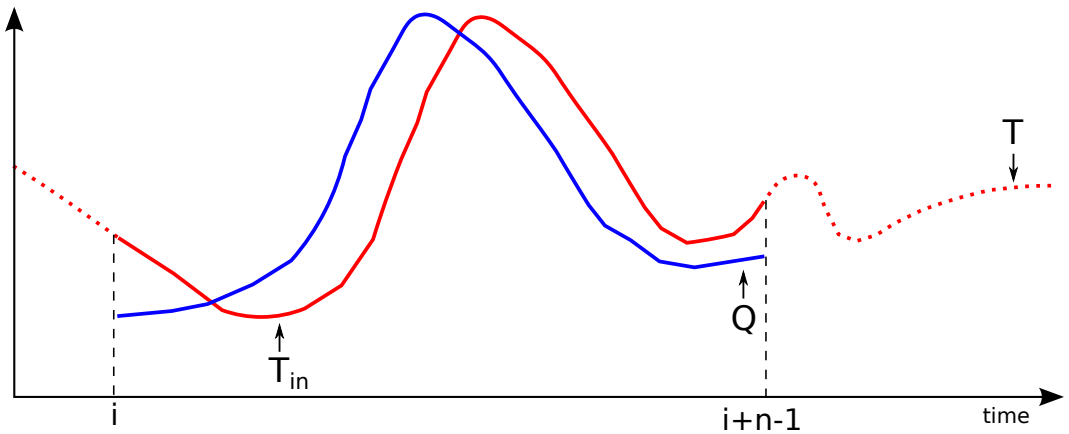
Сверхбольшие временные ряды





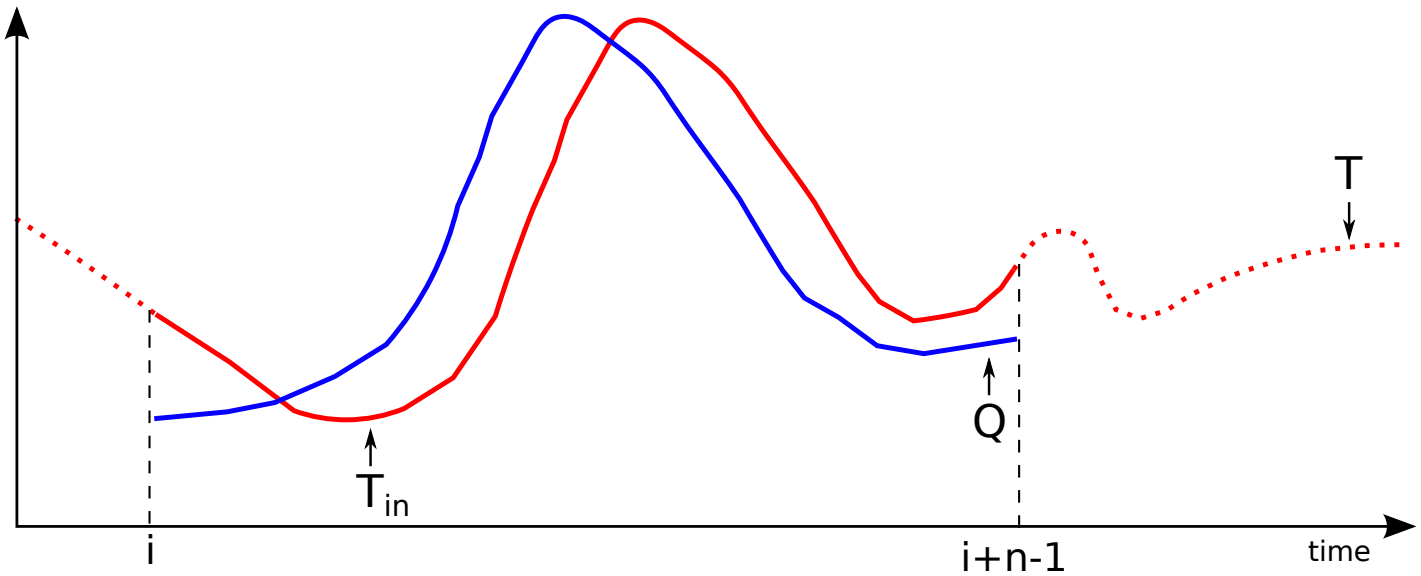


Формальные определения



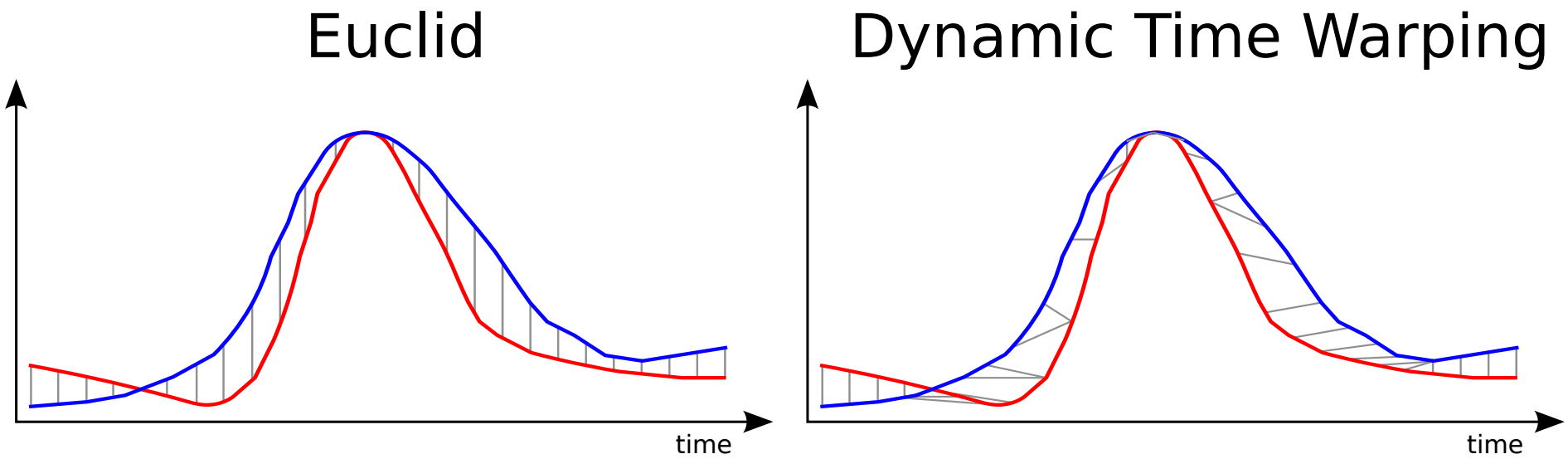
- *Временной ряд* T
  - $T = t_1, t_2, \ldots, t_N$, где $t_i \in \mathbb{R}$
  - N — длина ряда
- *Запрос* Q
  - Q — ряд, который требуется найти в T
  - n — длина запроса, $n \ll N$
- *Подпоследовательность* $T_{i\,m}$
  - $T_{i\,m} = t_i, t_{i+1}, \ldots, t_{i+m-1}$
  - $1 \leq i \leq N$ и $i + m \leq N$

Поиск самой похожей подпоследовательности



- Найти $T_{j\,n} = \underset{1 \le i \le N-n}{\operatorname{argmin}} D(T_{i\,n}, Q)$.
- D — *мера схожести*.

Мера схожести временных рядов DTW

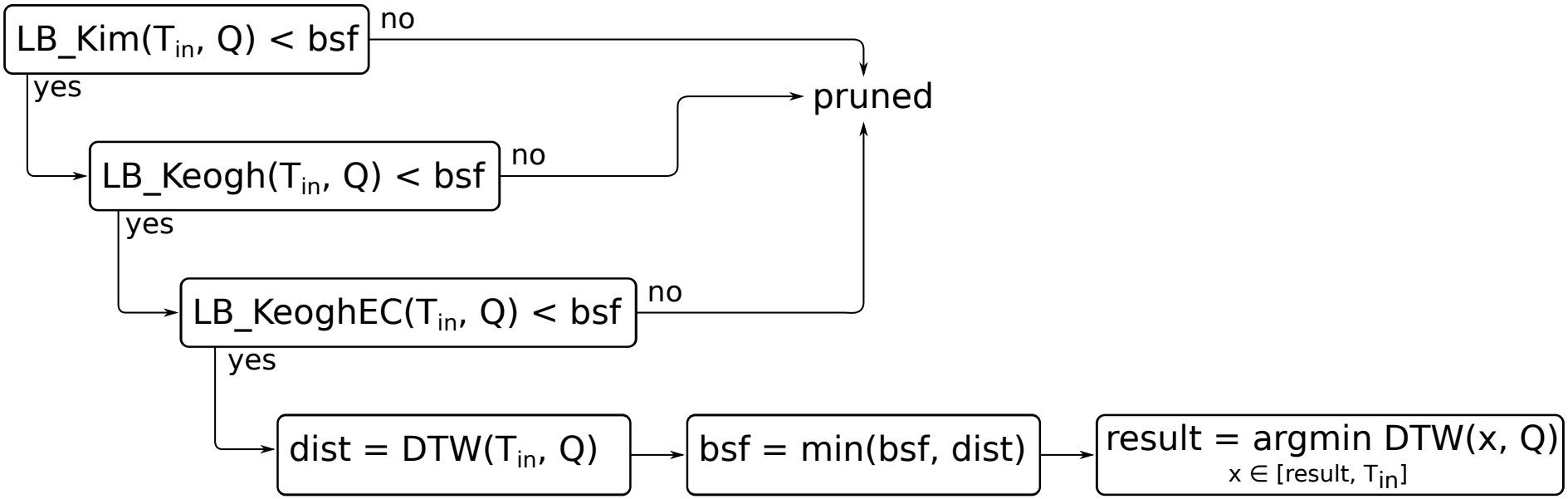


$$DTW(X,Y) = d(N,N),$$

$$d(i,j) = |x_i - y_j| + min \begin{cases} d(i-1,j) \\ d(i,j-1) \\ d(i-1,j-1), \end{cases}$$

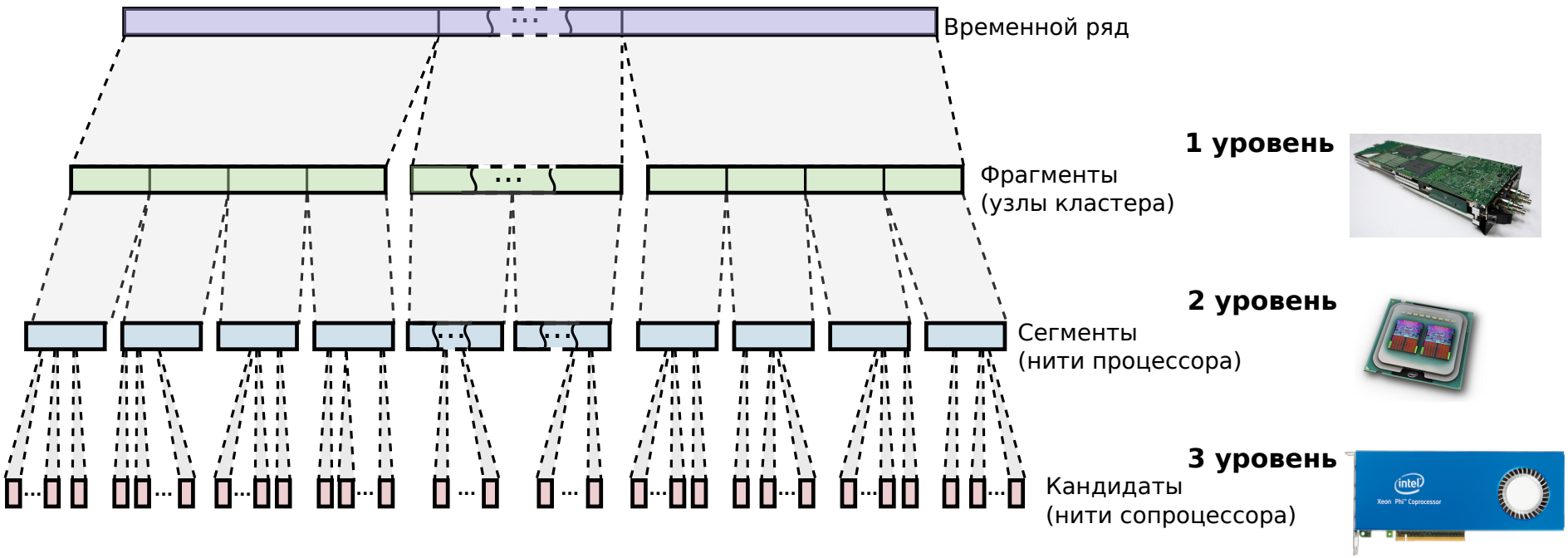
$$d(0,0) = 0; d(i,0) = d(0,j) = \infty; i = 1,2,\ldots,N; j = 1,2,\ldots,N.$$

Последовательный алгоритм UCR-DTW

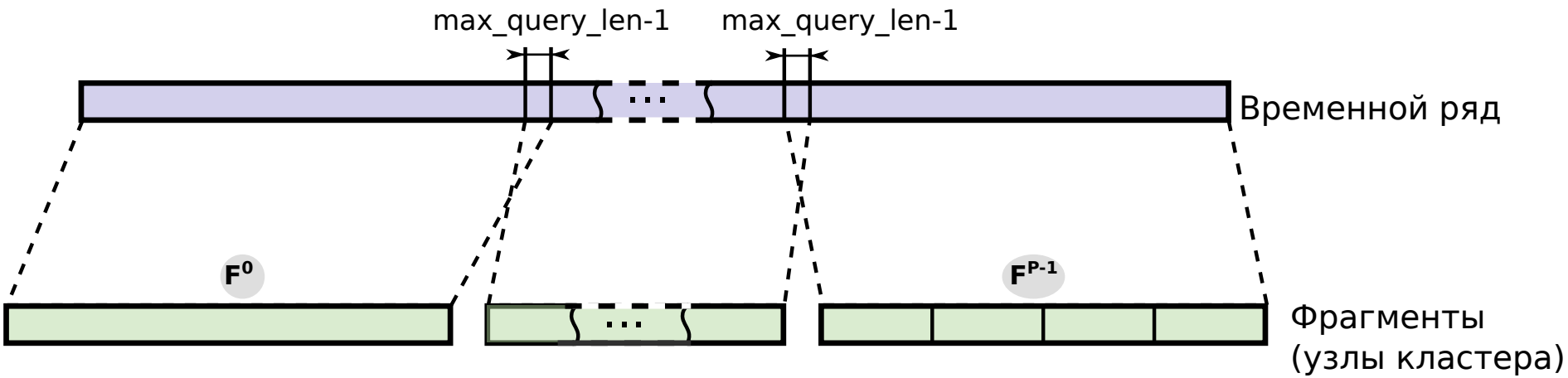


Rakthanmanon T., et al. Searching and Mining Trillions of Time Series Subsequences under Dynamic Time Warping // ACM SIGKDD, 2012. P. 262–270.

Уровни параллелизма алгоритма

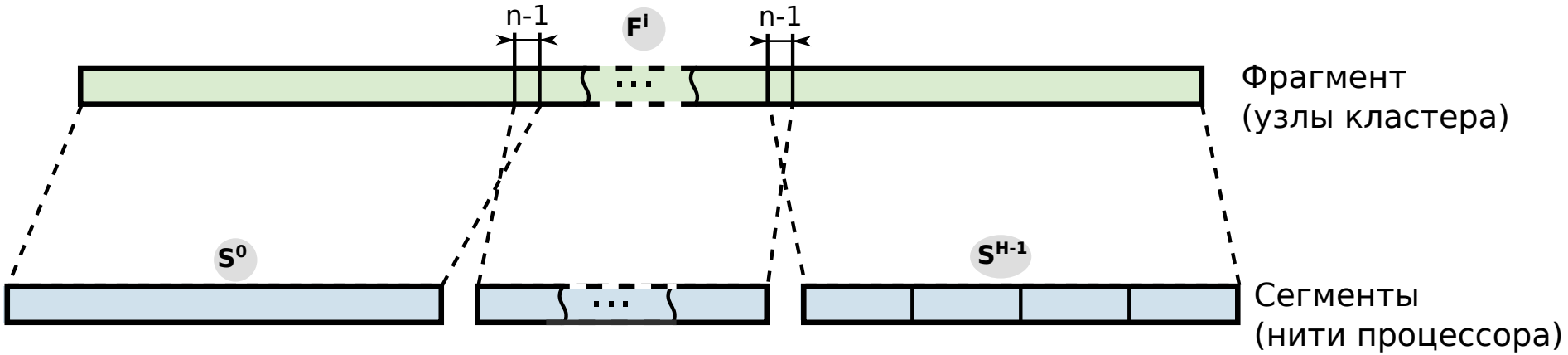


Фрагментация временного ряда



- P — количество фрагментов (узлов кластера);
- F^k — k-й фрагмент временного ряда $T = t_1, t_2, \ldots, t_N$, где $0 \leq k \leq P - 1$.

Сегментация фрагментов ряда

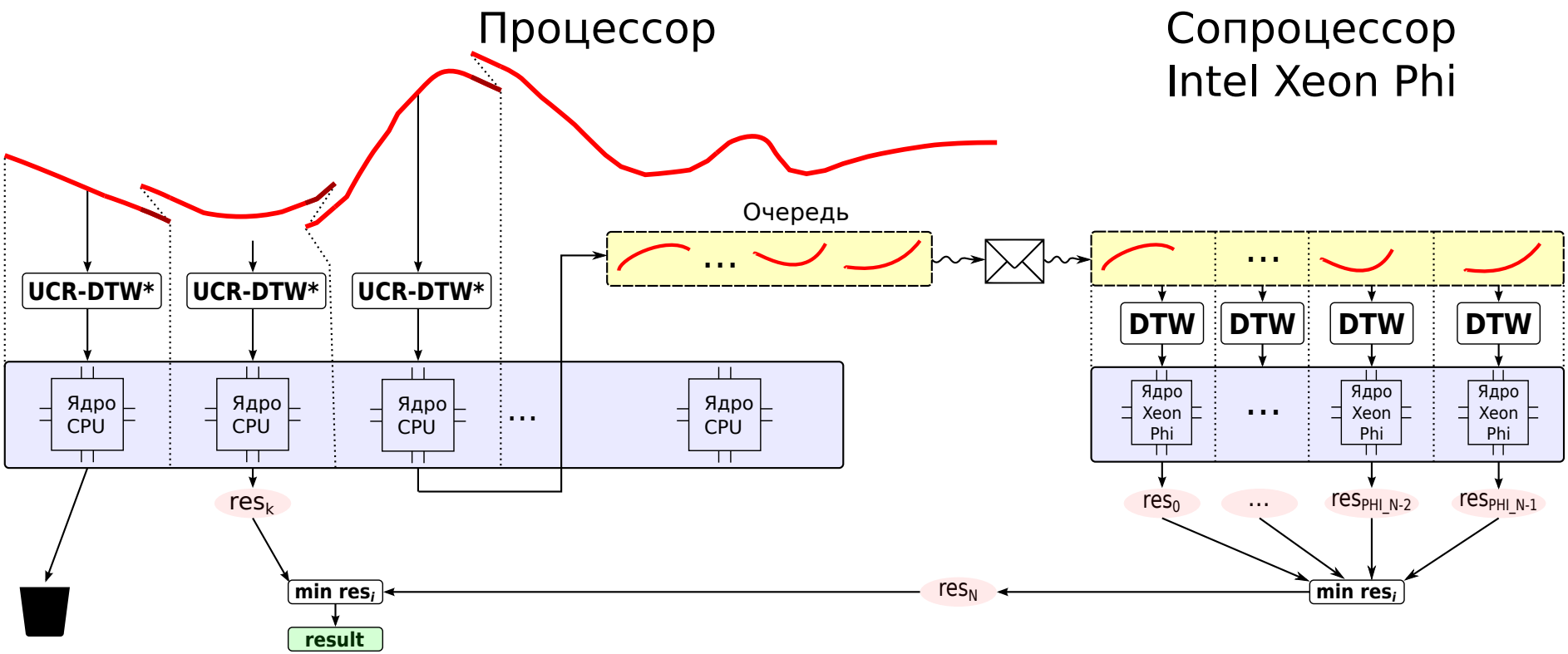


- S^k — k-й сегмент фрагмента F^i, где $0 \leq k \leq H - 1$;
- H — количество сегментов в фрагменте F^k:

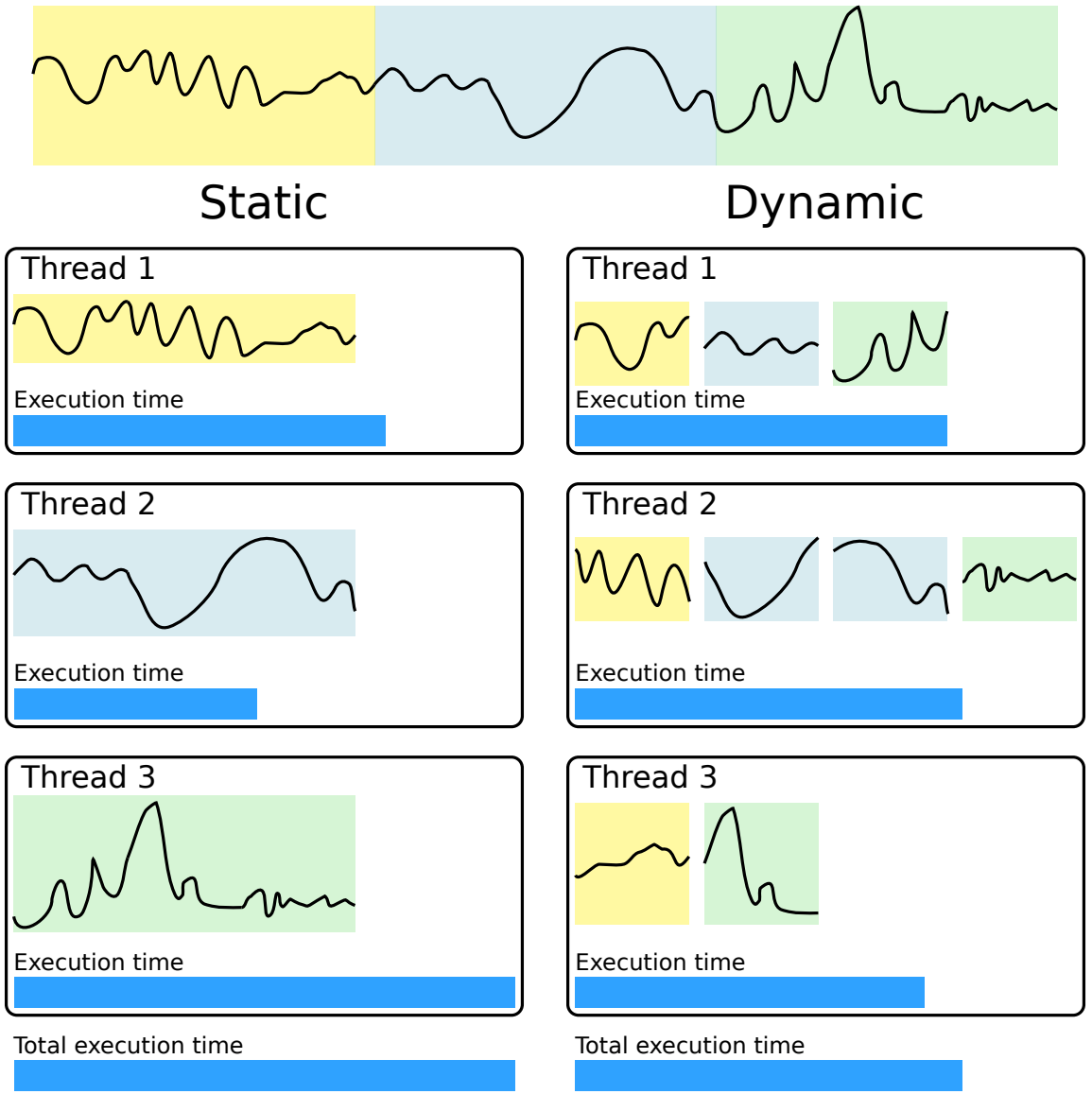
$$H = \lceil \frac{len}{S \cdot L} \rceil \cdot S,$$

  - len — длина фрагмента;
  - S — количество нитей для параллельной обработки сегментов;
  - L — длина сегмента.

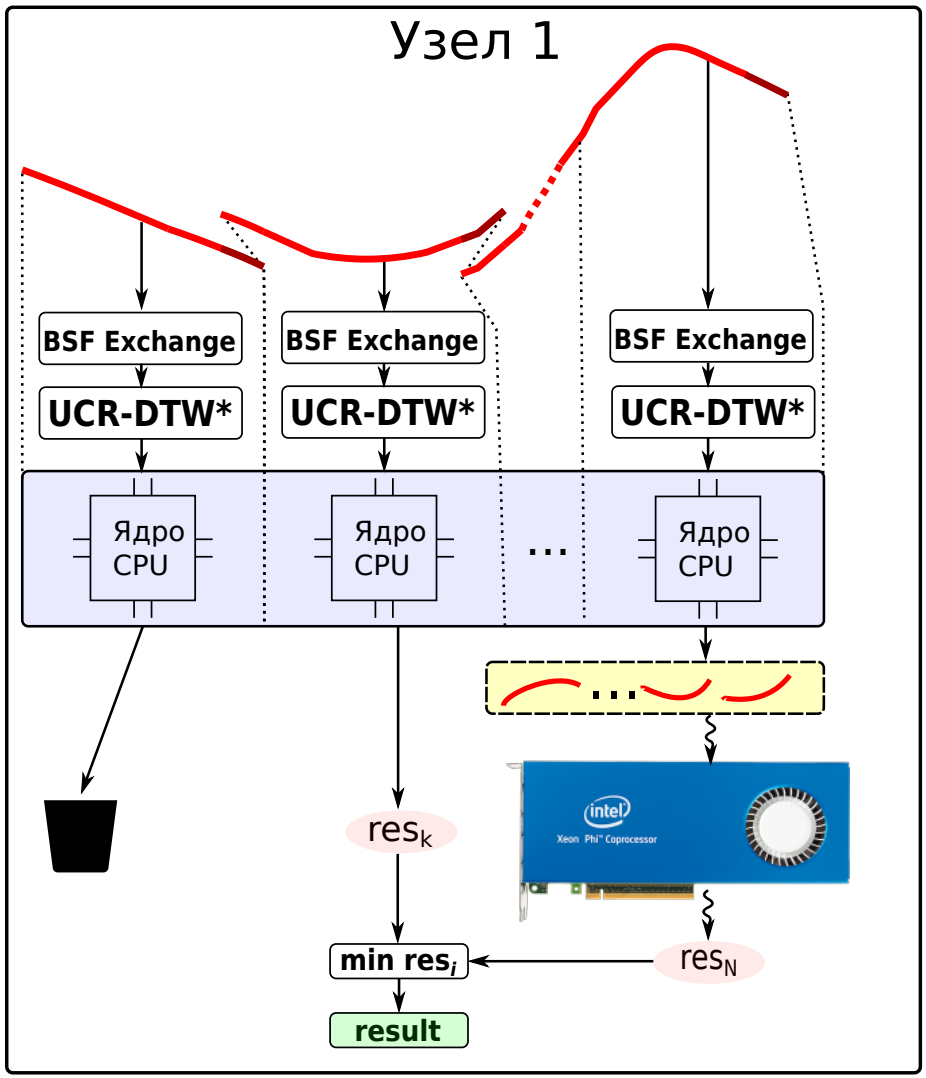
Параллельная обработка сегментов

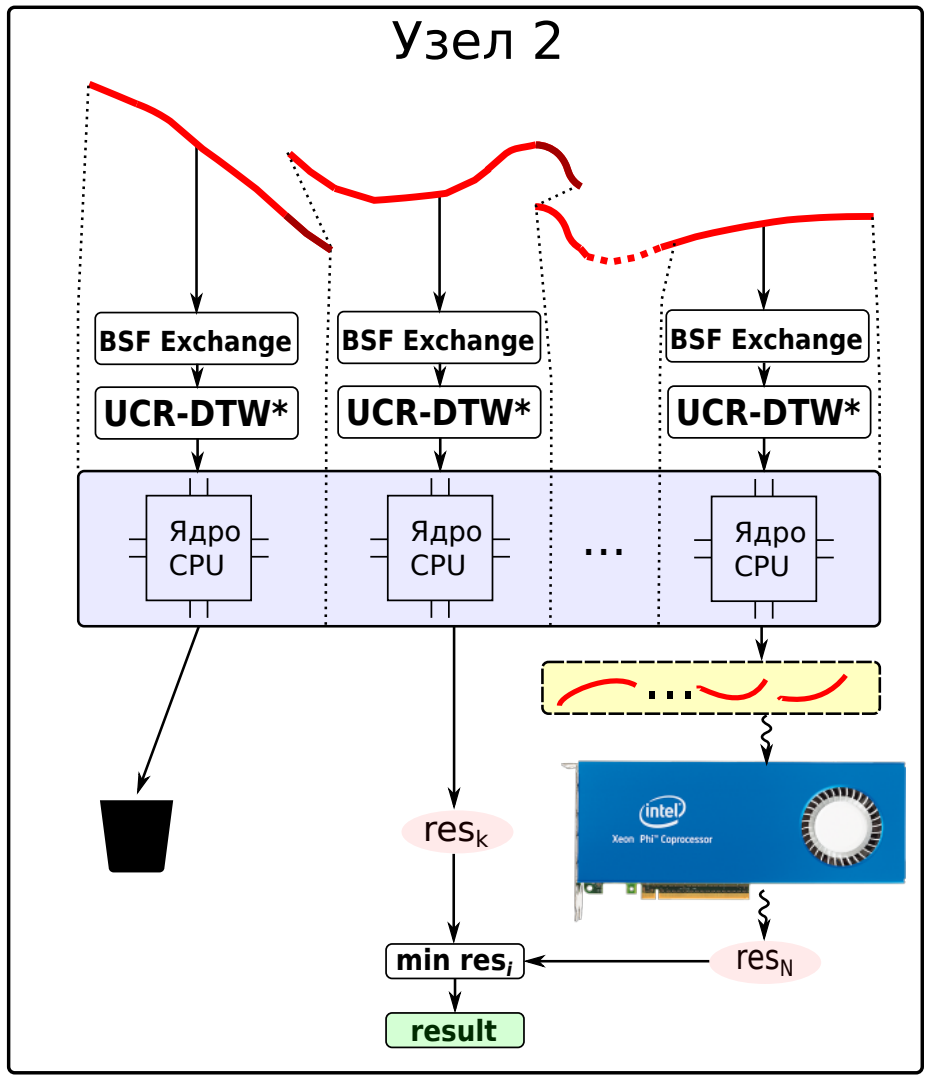


Динамическое распределение сегментов

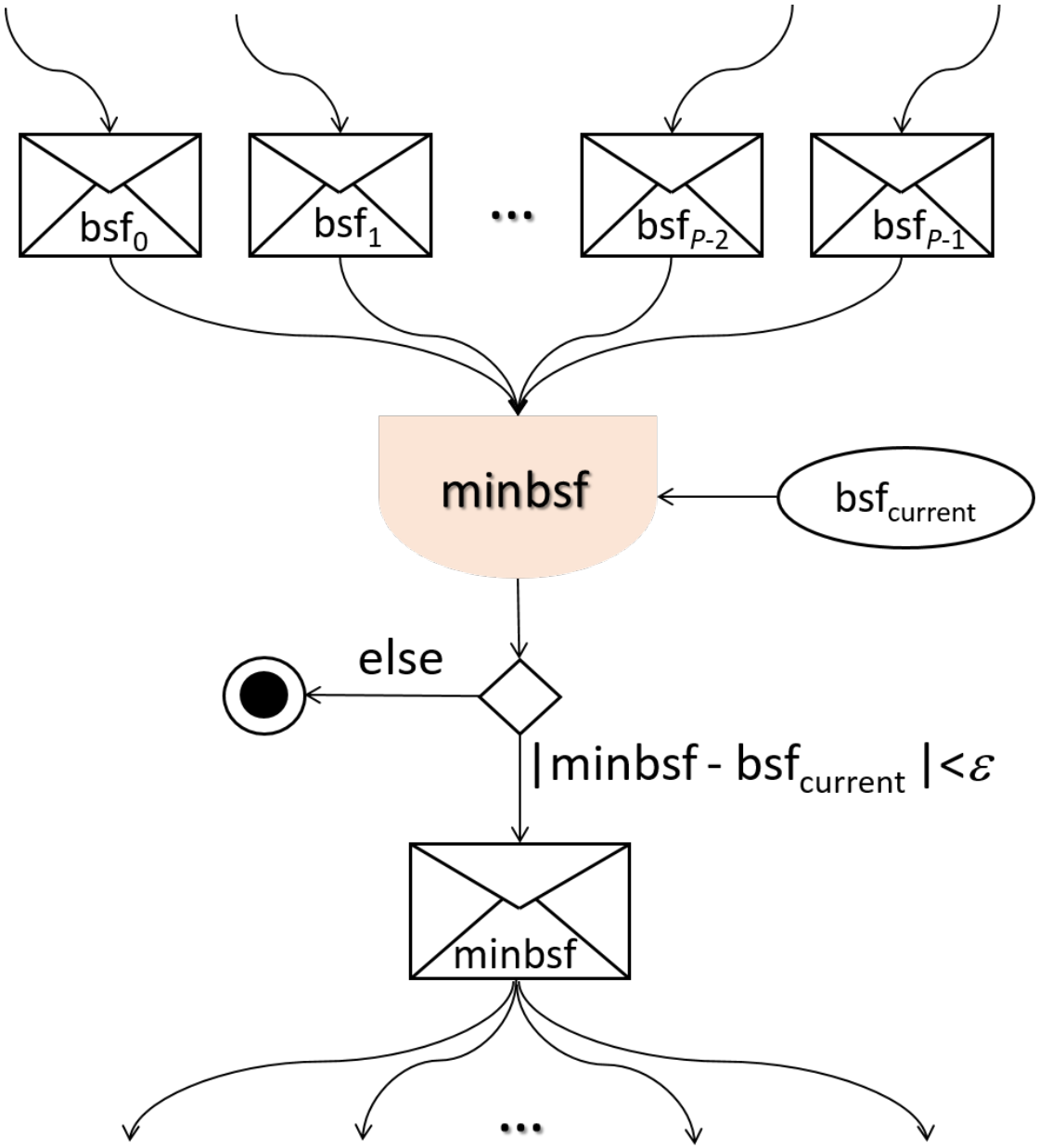


Параллельная обработка фрагментов





Обмен оценками схожести между узлами



Вычислительные эксперименты

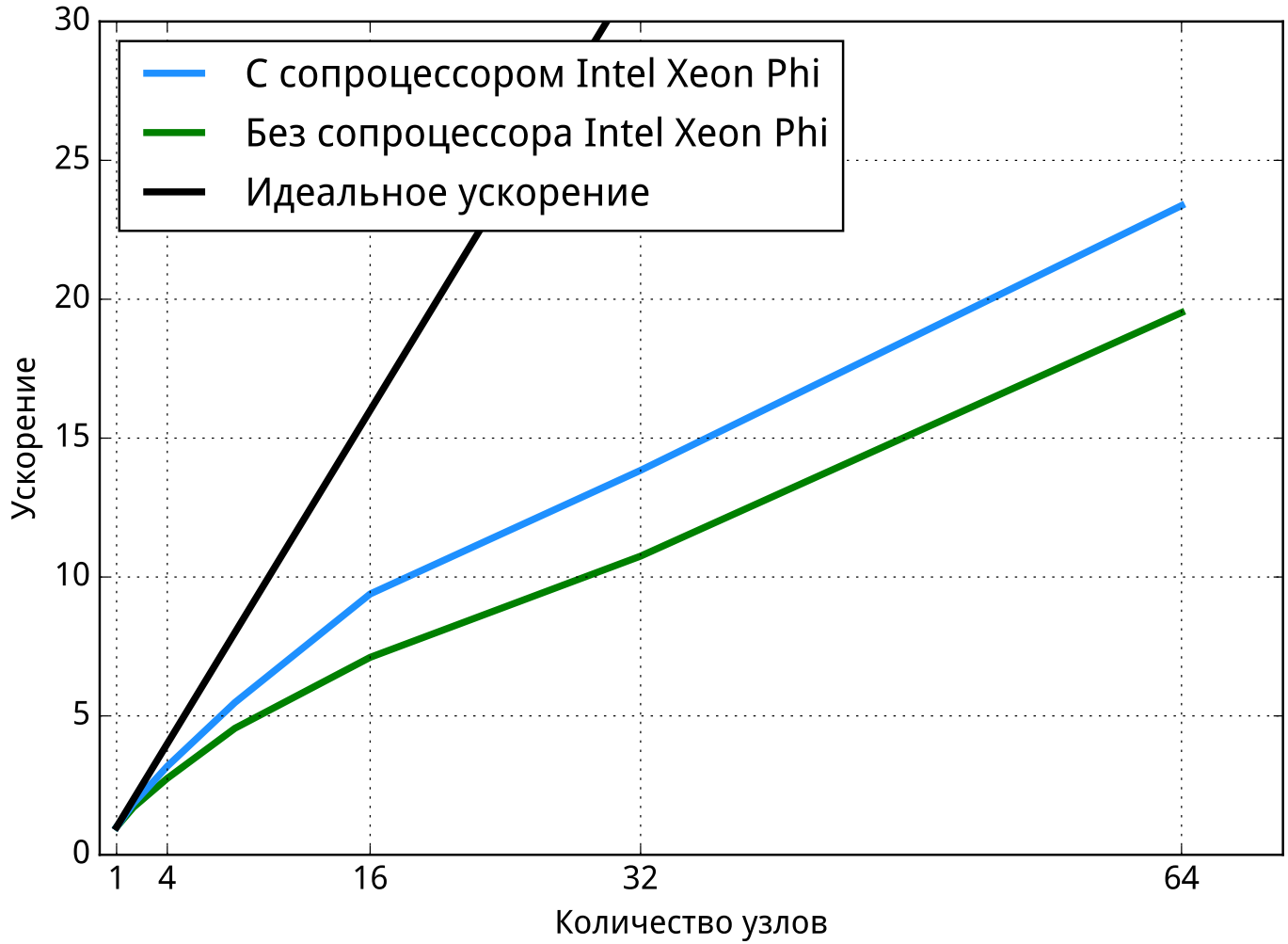
- Платформа: «Торнадо ЮУрГУ», 1..64 узла

Наименование	Процессор	Ускоритель
Модель, Intel	Xeon X5680	Xeon Phi SE10X
K-во ядер	6	61
Частота, GHz	3.33	1.1
Гиперпоточность	2	4
Пик. произв-ть, TFLOPS	0.371	1.076
Память, Gb	24	8
Кэш, Mb	12	30.5

- Данные: синтетические (модель RANDOM WALK)
- Цели:
  - Ускорение
  - Расширяемость
  - Вклад ускорителей
  - Сравнение с аналогами

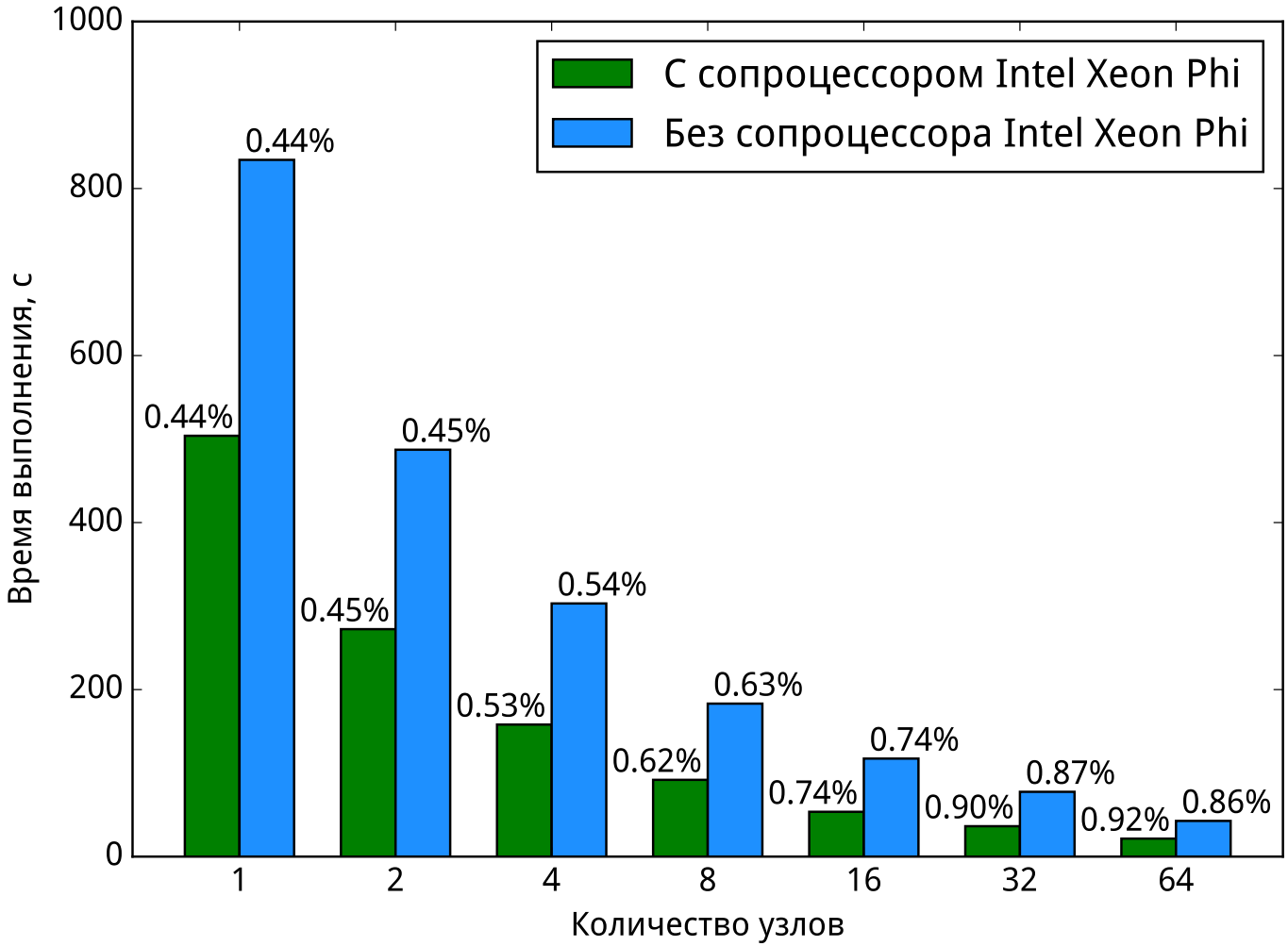
Ускорение

Длина временного ряда $N = 8 \cdot 10^8$ (6 Гб).
Длина запроса $n = 4000$.
Длина сегмента $L = 10^6$.
Пороговое значение улучшения оценки $\mathcal{E} = 0.01$.



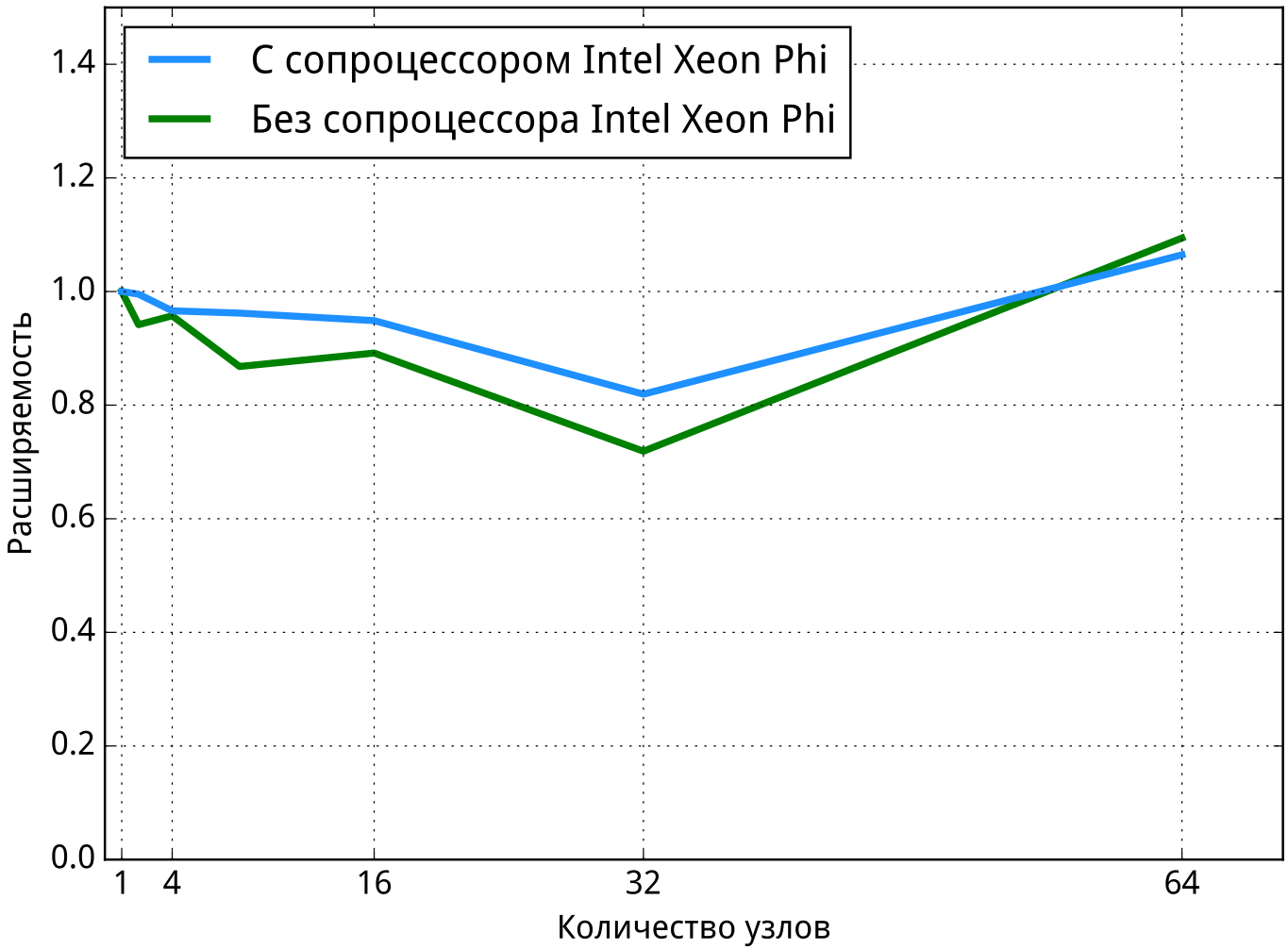
Ускорение

Длина временного ряда $N = 8 \cdot 10^8$ (6 Гб).
Длина запроса $n = 4000$.
Длина сегмента $L = 10^6$.
Пороговое значение улучшения оценки $\mathcal{E} = 0.01$.



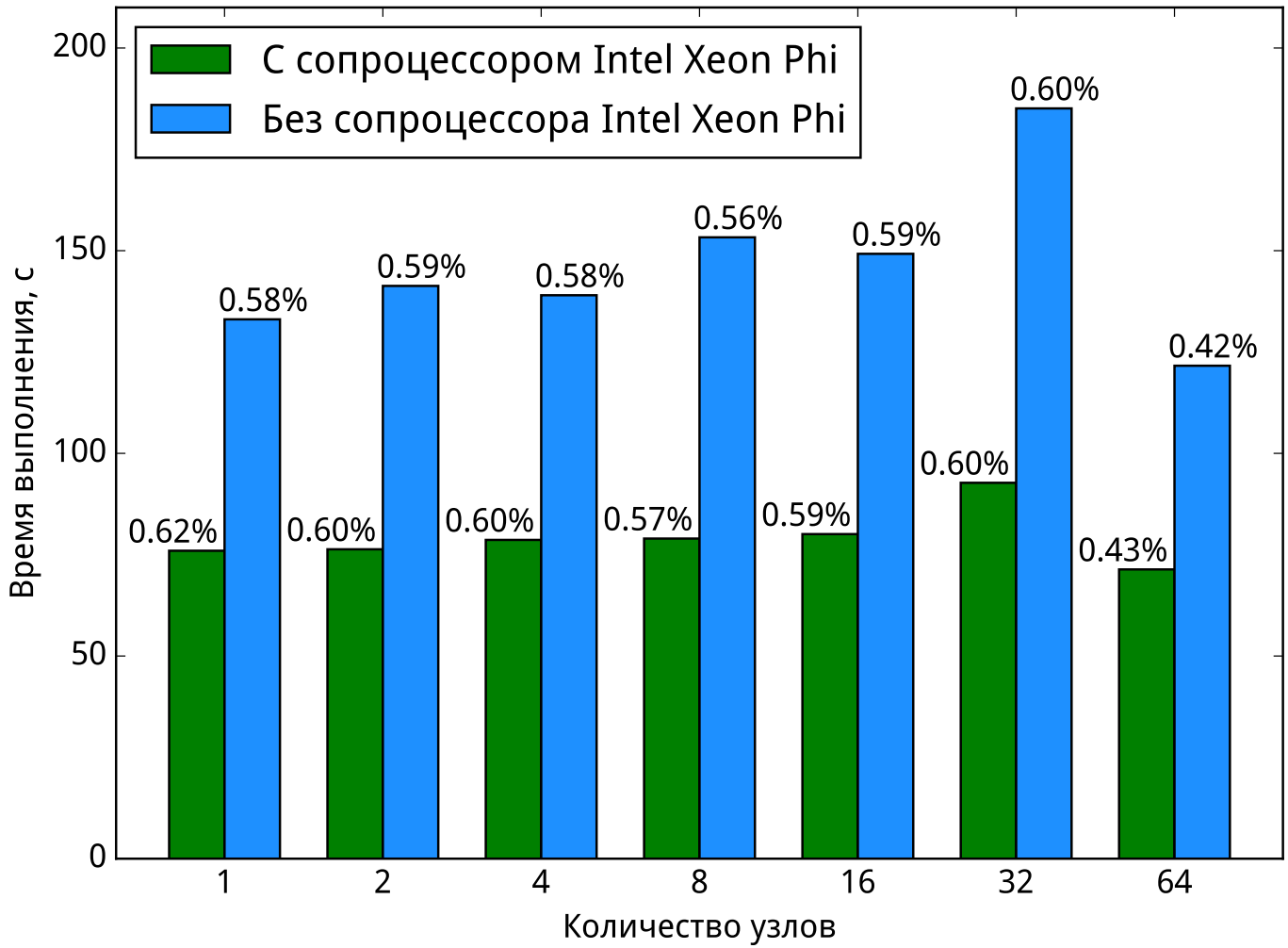
Расширяемость

Длина временного ряда $N = 8 \cdot 10^8$ (0.7 Гб до 47.7 Гб).
Длина запроса $n = 4000$.
Длина сегмента $L = 10^6$.
Пороговое значение улучшения оценки $\mathcal{E} = 0.01$.



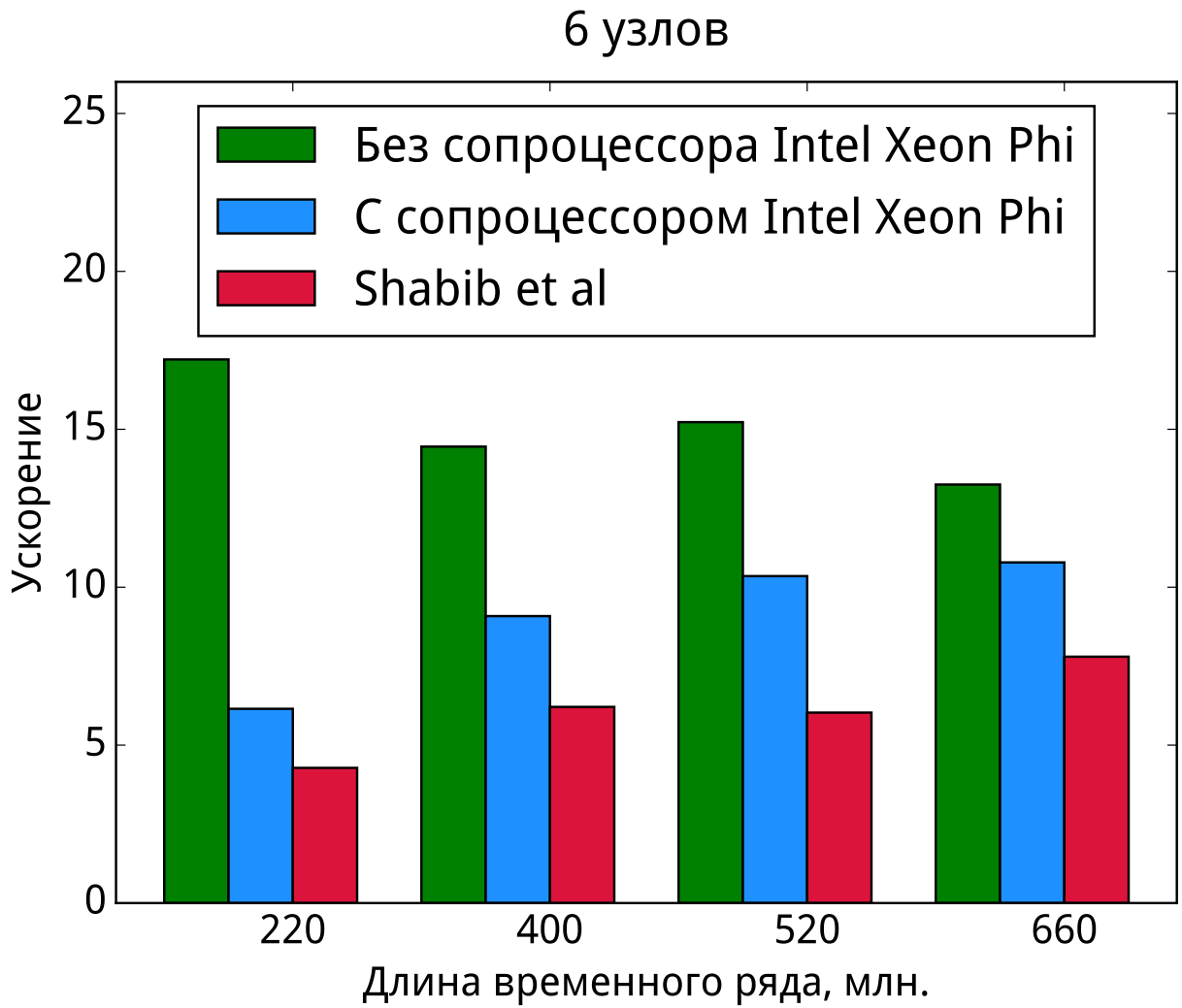
Расширяемость

Длина временного ряда $N = 8 \cdot 10^8$ (0.7 Гб до 47.7 Гб).
Длина запроса $n = 4000$.
Длина сегмента $L = 10^6$.
Пороговое значение улучшения оценки $\mathcal{E} = 0.01$.



Сравнение с аналогами

Длина запроса $n = 128$.



Shabib A. et al. Parallelization of Searching and Mining Time Series Data Using Dynamic Time Warping // ICACCI 2015. IEEE, 2015. P. 343–348.

Заключение

- Реализован параллельный алгоритм поиска самой похожей подпоследовательности временного ряда для вычислительного кластера с узлами на базе ускорителей Intel Xeon Phi.
- Результаты вычислительных экспериментов показывают эффективность алгоритма и его превосходство над аналогами.