

Нахождение похожих подпоследовательностей временного ряда с помощью многоядерного сопроцессора Intel Xeon Phi

Александр Вячеславович Мовчан, Михаил Леонидович Цымблер

Южно-Уральский государственный университет

Параллельные вычислительные технологии (ПаВТ) 2015

*Работа выполнена при финансовой поддержке Минобрнауки России в рамках ФЦП
«Исследования и разработки по приоритетным направлениям развития
научно-технологического комплекса России на 2014–2020 годы» (Госконтракт
№ 14.574.21.0035).*

Применение временных рядов

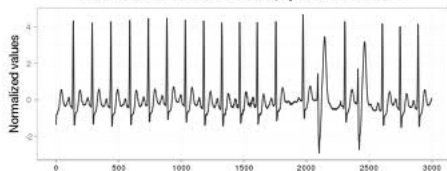
DJ INDU AVERAGE (Dow Jones & Co
as of 6-Dec-2004



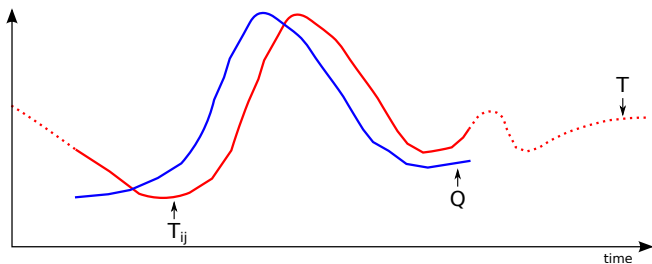
Copyright 2004 Yahoo! Inc.

<http://Finance.yahoo.com/>

Normalized heartbeat series, qtdbsele0606.arff



Поиск похожей подпоследовательности



Самая похожая подпоследовательность:

$$T_{ij}, \forall T_{mn}, |T_{mn}| = |T_{ij}| = |Q|, D(T_{ij}, Q) < D(T_{mn}, Q)$$

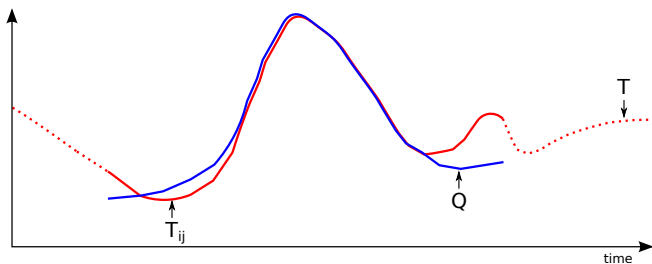
T – временной ряд, по которому осуществляется поиск

T_{ij} – подпоследовательность

Q – запрос

D – функция расстояния

Поиск похожей подпоследовательности



Самая похожая подпоследовательность:

$$T_{ij}, \forall T_{mn}, |T_{mn}| = |T_{ij}| = |Q|, D(T_{ij}, Q) < D(T_{mn}, Q)$$

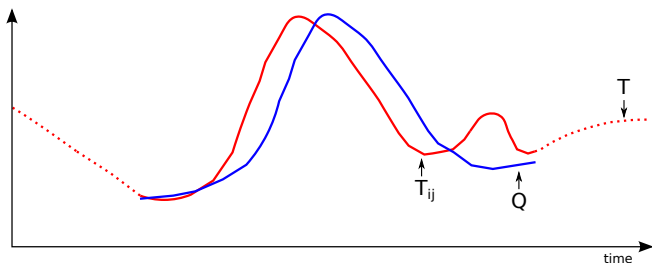
T – временной ряд, по которому осуществляется поиск

T_{ij} – подпоследовательность

Q – запрос

D – функция расстояния

Поиск похожей подпоследовательности



Самая похожая подпоследовательность:

$$T_{ij}, \forall T_{mn}, |T_{mn}| = |T_{ij}| = |Q|, D(T_{ij}, Q) < D(T_{mn}, Q)$$

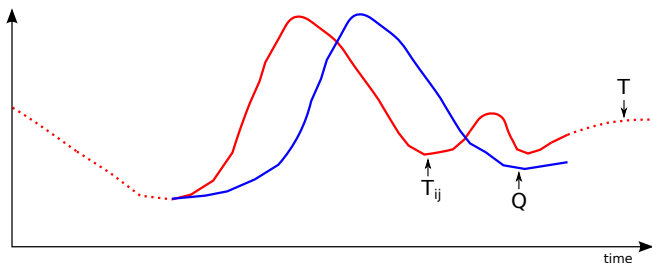
T – временной ряд, по которому осуществляется поиск

T_{ij} – подпоследовательность

Q – запрос

D – функция расстояния

Поиск похожей подпоследовательности



Самая похожая подпоследовательность:

$$T_{ij}, \forall T_{mn}, |T_{mn}| = |T_{ij}| = |Q|, D(T_{ij}, Q) < D(T_{mn}, Q)$$

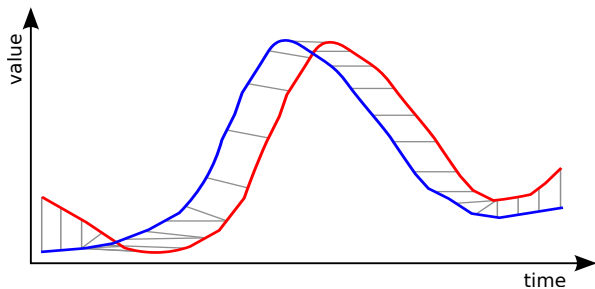
T – временной ряд, по которому осуществляется поиск

T_{ij} – подпоследовательность

Q – запрос

D – функция расстояния

Динамическая трансформация шкалы времени

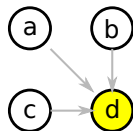
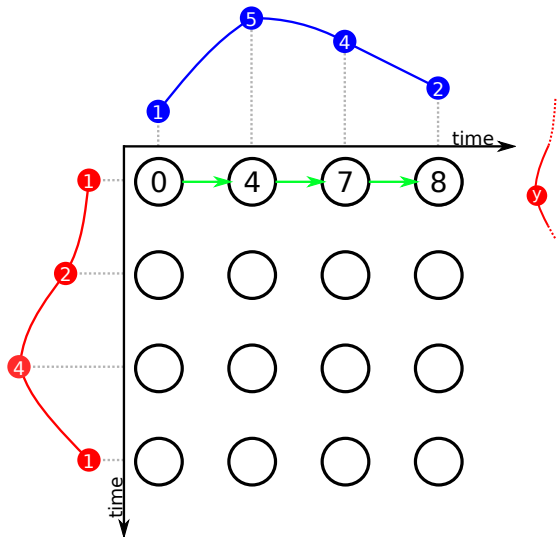


$$DTW(X, Y) = d(N, N),$$

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

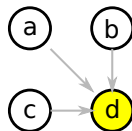
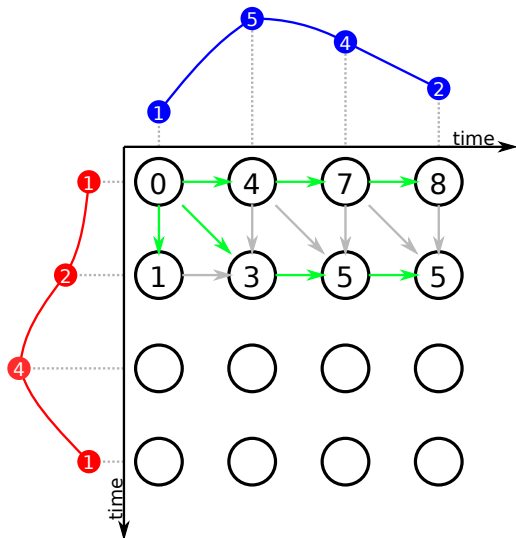
$$d(0, 0) = 0; d(i, 0) = d(0, j) = \infty; i = 1, 2, \dots, N; j = 1, 2, \dots, N.$$

Динамическая трансформация шкалы времени



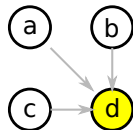
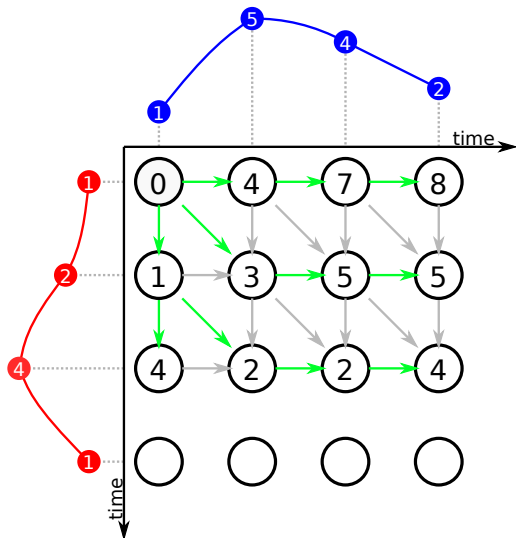
$$d = \text{cost} + \min(a, b, c)$$
$$\text{cost} = |x - y|$$

Динамическая трансформация шкалы времени



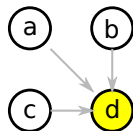
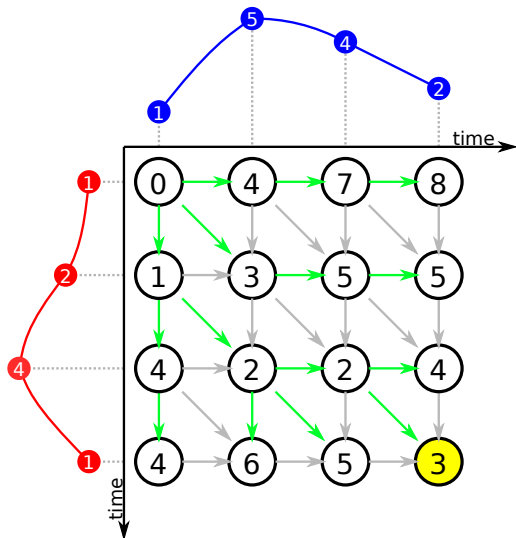
$$d = \text{cost} + \min(a, b, c)$$
$$\text{cost} = |x - y|$$

Динамическая трансформация шкалы времени



$$d = \text{cost} + \min(a, b, c)$$
$$\text{cost} = |x - y|$$

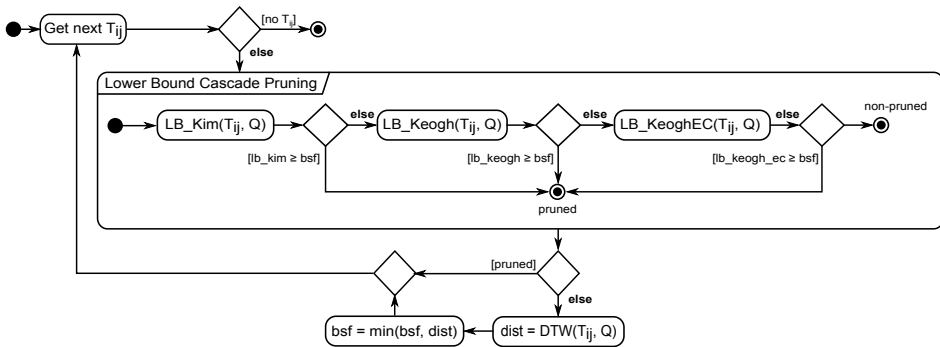
Динамическая трансформация шкалы времени



$$d = \text{cost} + \min(a, b, c)$$

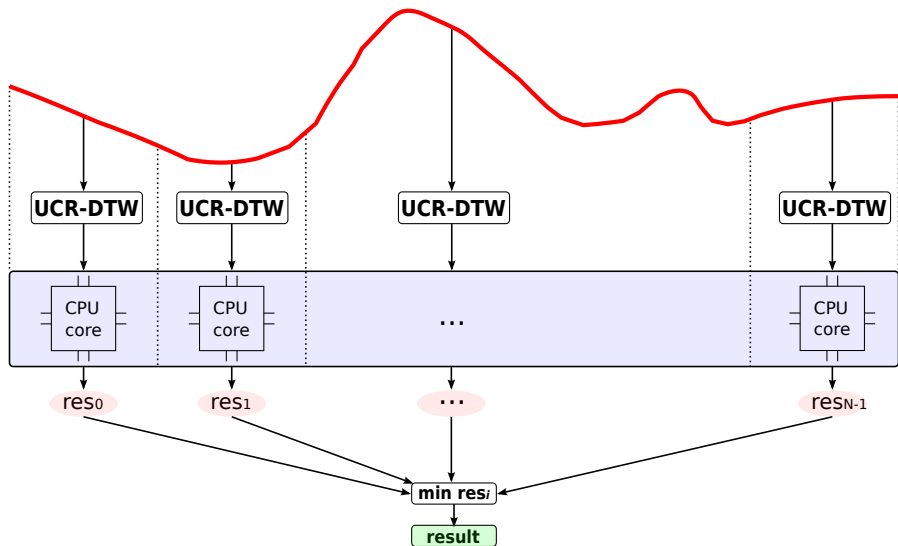
$$\text{cost} = |x - y|$$

Последовательный алгоритм



Rakthanmanon T., et al. Searching and Mining Trillions of Time Series Subsequences under Dynamic Time Warping // The 18th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Beijing, China, 12-16 August, 2012. ACM, 2012. P. 262–270.

Параллельный алгоритм для процессора

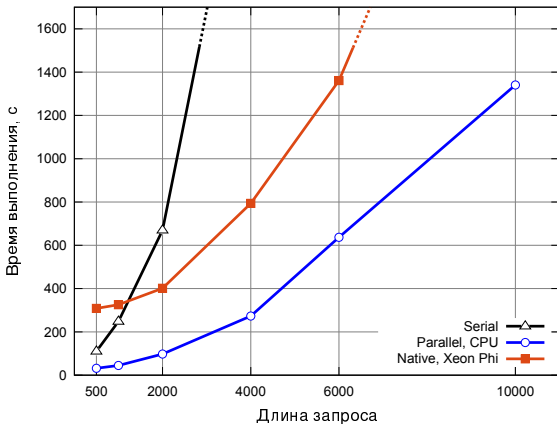


Производительность параллельного алгоритма для процессора

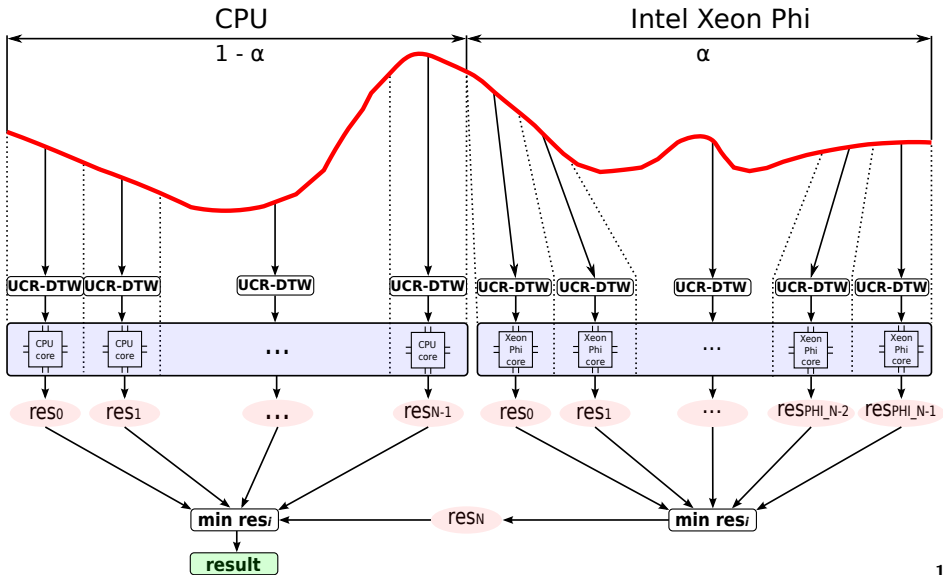
LB_Kim	$O(1)$
LB_Keogh	$O(n)$
LB_KeoghEC	$O(n)$
DTW	$O(n^2)$

Время загрузки данных с диска в память Xeon Phi:
 ≈ 300 с

Данные: random walk, 10^9 точек

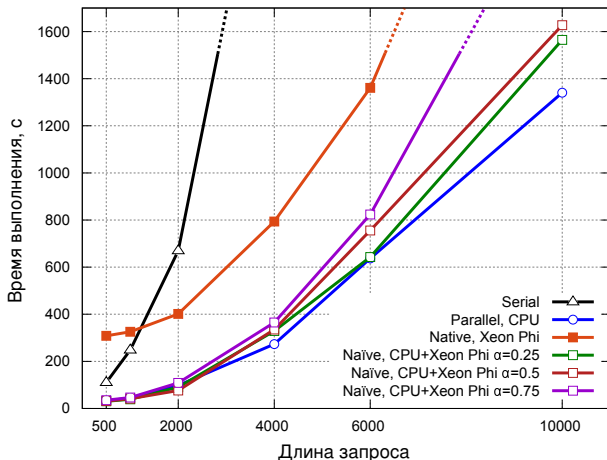


Наивный параллельный алгоритм для сопроцессора

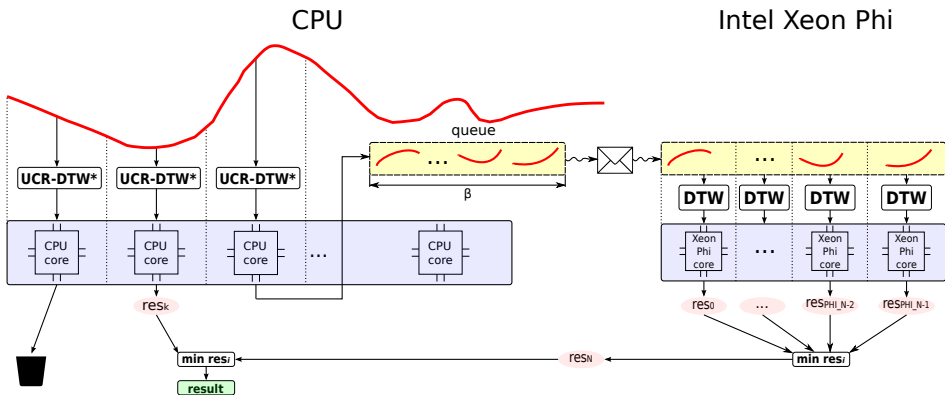


Производительность параллельного алгоритма для Intel Xeon Phi

Данные: random walk, 10^9 точек



Улучшенный параллельный алгоритм для сопроцессора

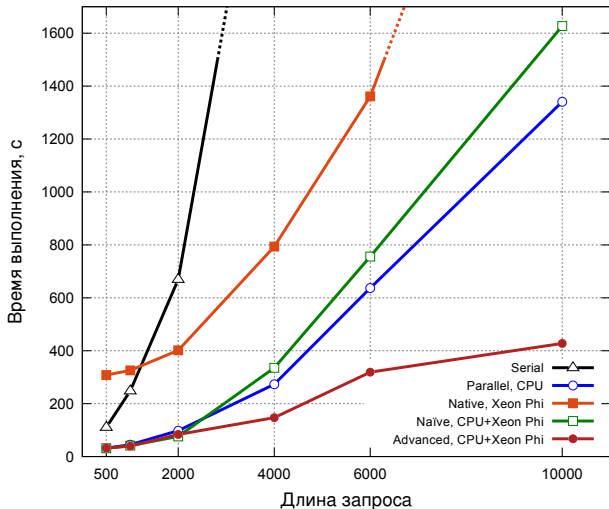


Эксперименты

- ▶ Аппаратная платформа
 - Процессор
 - Intel Xeon X5680
 - 6 ядер по 3.33 GHz
 - 0.371 Тфлопс
 - Сопроцессор
 - Intel Xeon Phi SE10X
 - 61 ядро по 1.1 GHz
 - 1.076 Тфлопс
- ▶ Данные
 - Синтетические
 - random walk, 10^9 точек данных
 - Реальные
 - ЭКГ, 2×10^7 точек данных (22 час. при частоте дискретизации 250 Гц)

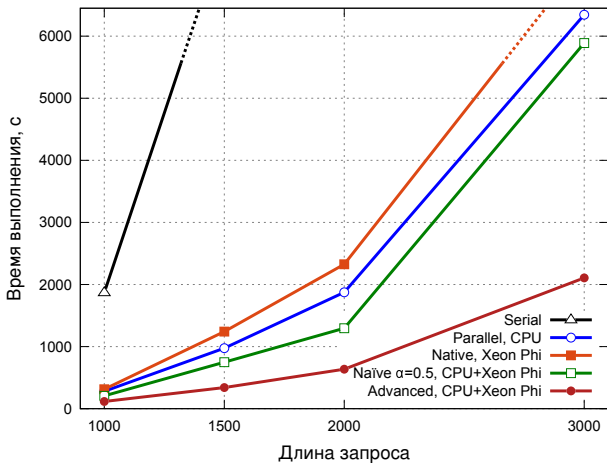
Производительность на синтетических данных

Данные: random walk, 10^9 точек

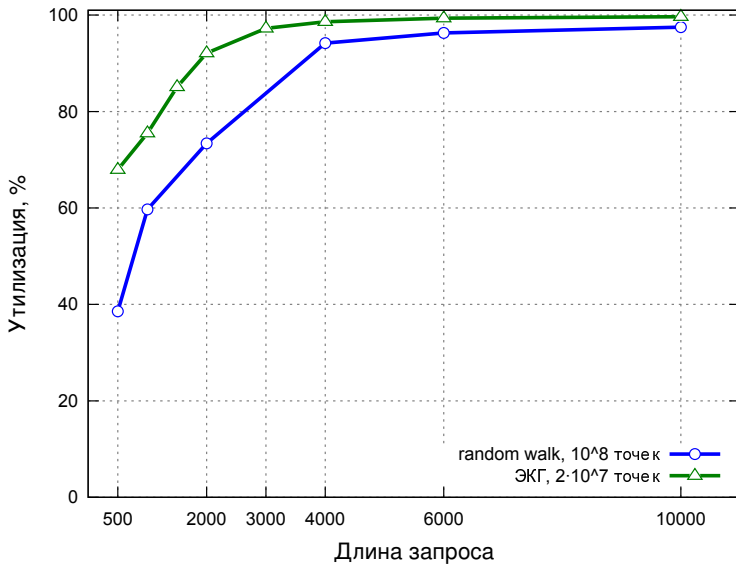


Производительность на реальных данных

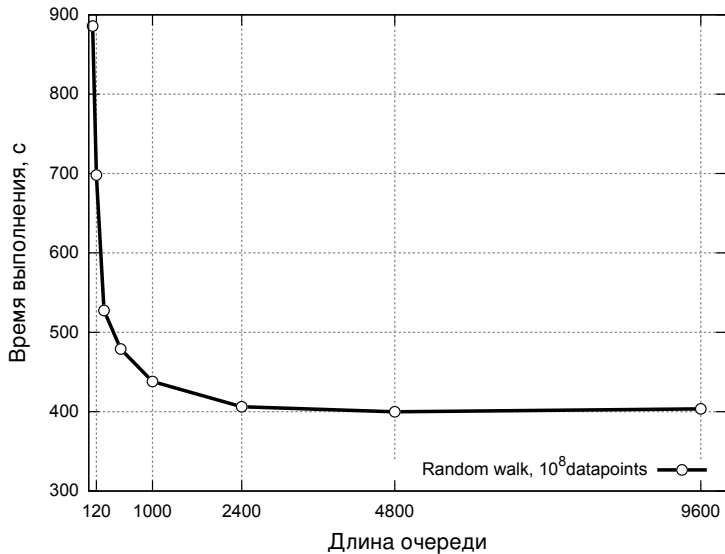
Данные: ЭКГ, 2×10^7 точек



Утилизация сопроцессора

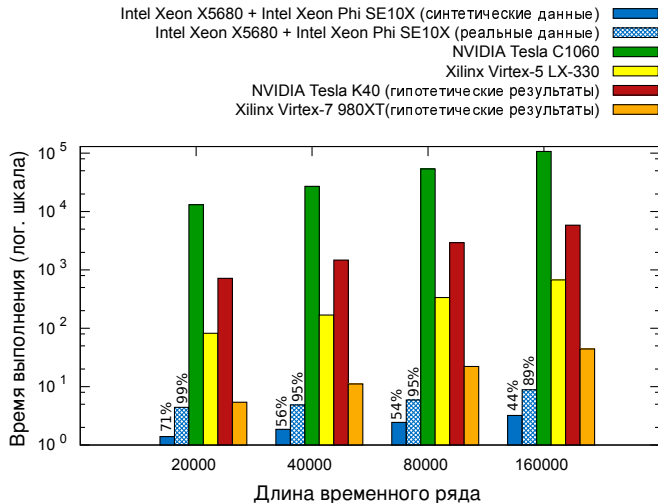


Влияние размера очереди на производительность



Сравнение с аналогами

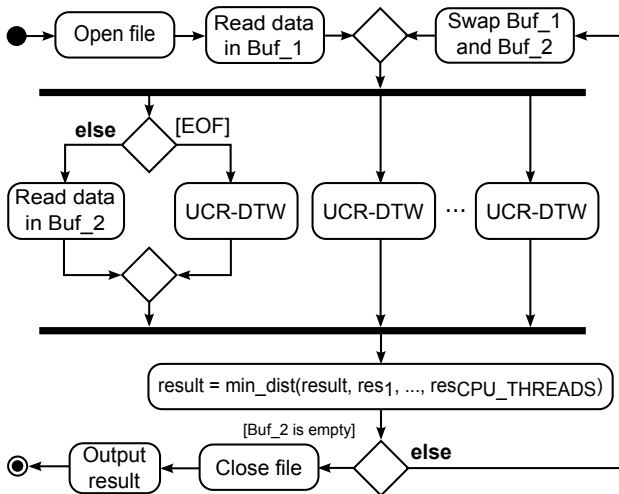
Длина запроса: 1024



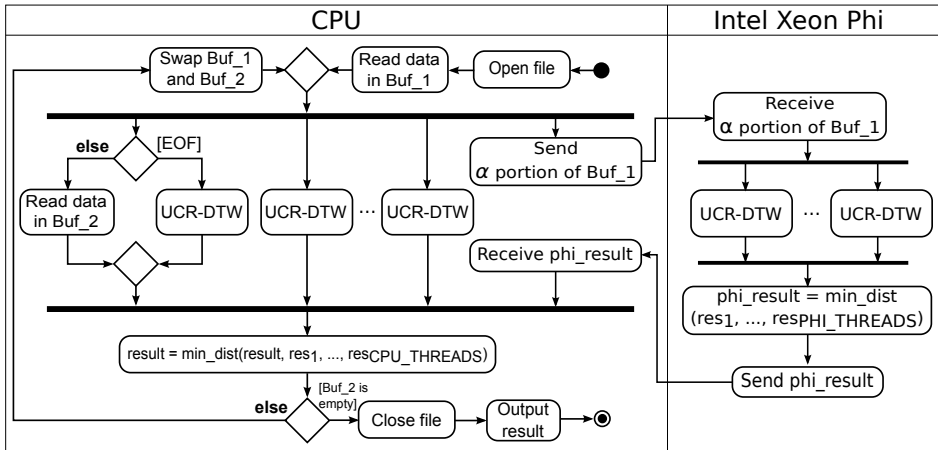
Заключение

- ▶ Разработан параллельный алгоритм поиска похожей подпоследовательности временного ряда для сопроцессоров Intel Xeon Phi
- ▶ Эксперименты показали высокую эффективность алгоритма при большой длине запроса
- ▶ Будущие исследования:
 - алгоритм для узла с несколькими сопроцессорами Intel Xeon Phi
 - алгоритм для кластерной системы с узлами на базе сопроцессоров Intel Xeon Phi

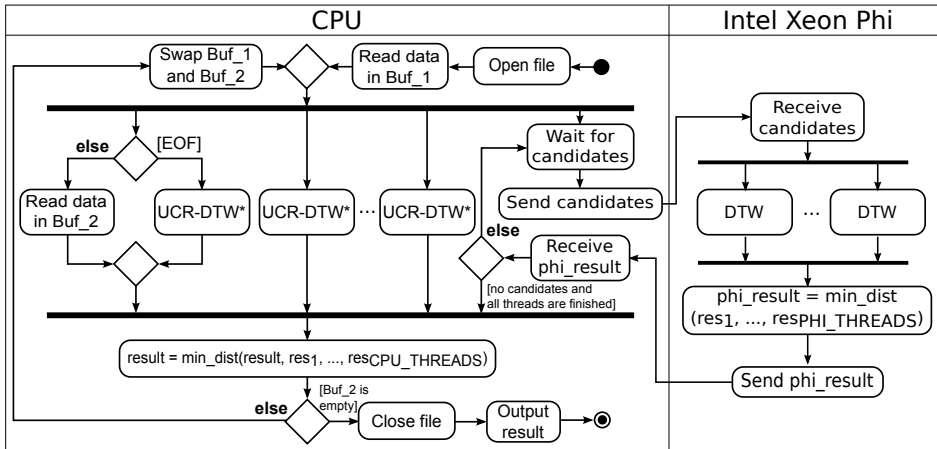
Параллельный алгоритм для процессора



Наивный параллельный алгоритм для сопроцессора



Улучшенный параллельный алгоритм для сопроцессора



UCR-DTW*

