Предсказательная аналитика на основе потоков Больших Данных

Андрей Дмитриев, д.ф.-м.н., профессор, **НИУ ВШЭ**

Дается краткий обзор методов предсказательной аналитики на основе потоков больших массивов данных, генерируемых в режиме реального времени. Более подробно обсуждаются возможности и ограничения методов детектирования кризисов и предкризисных режимов кризисов в данных потоках. Предложена нелинейно-динамическая модель, генерирующая временные ряды рыночных цен спроса и предложения с одним фундаментальным управляющим параметром. Калибровкой данного параметра по историческим данным и выделением интервалов его постоянства при определенных условиях удалось детектировать предкризисные режимы. Данная модель апробирована на биржевых данных по ценам спроса и предложения на драгоценные металлы и стальные биллеты. Разработано приложение, позволяющее на основе данной динамической модели детектировать предкризисные режимы в потоках больших массивов данных, генерируемых в режиме реального времени.

Поиск похожих подпоследовательностей временных рядов на сопроцессорах Intel Xeon Phi

Михаил Цымблер, к.ф.-м.н., доцент, **Александр Мовчан**, **ЮУрГУ**

Временной ряд (time series) представляет собой совокупность вещественных значений, каждое из которых ассоциировано с последовательными отметками времени. Задача поиска похожих подпоследовательностей (subsequence matching) определяется следующим образом (см. рис. 1). Пусть дан временной ряд T, его подпоследовательности мы обозначаем как Tij, где $i<j$ — номера членов ряда; пусть задан запрос Q — временной ряд с длиной, не превышающей длину ряда T; имеется функция схожести $D(t1,t2)$, определяющая схожесть двух временных рядов. Необходимо найти подпоследовательности Tij, имеющие длину, равную длине запроса, для которых значение функции $D(Tij,Q)$ минимально.

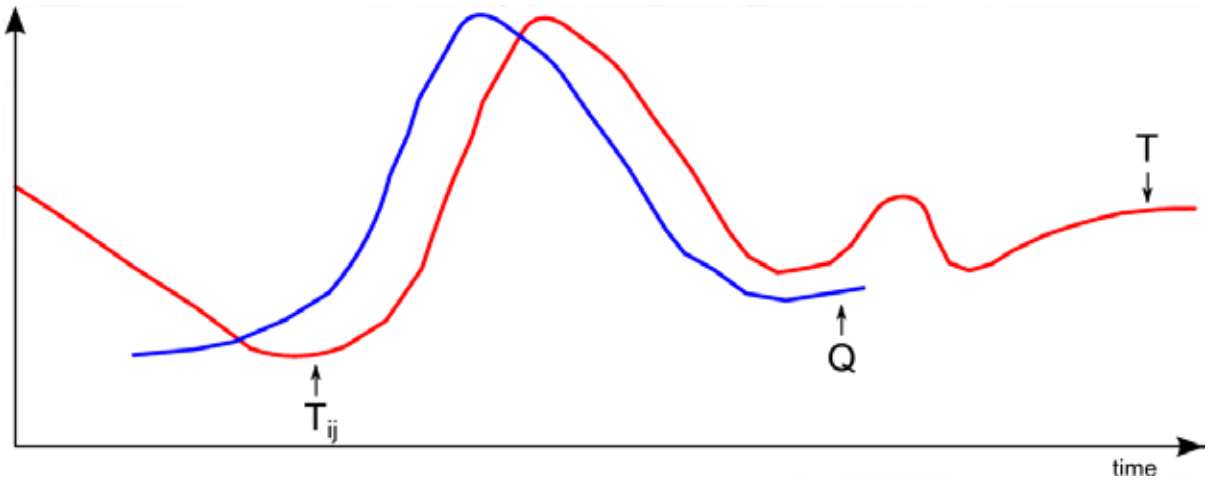

Рис. 1. *Поиск похожих подпоследовательностей*

Для определения схожести временных рядов можно использовать различные функции схожести. В настоящее время одной из наиболее популярных функций схожести временных рядов является *динамическая трансформация шкалы времени (Dynamic Time Warping, DTW)*, которая отличается от традиционной функции евклидова расстояния и вычислительно существенно более сложна. Преимуществом динамической трансформации шкалы времени является возможность сравнивать временные ряды, различающиеся скоростью изменения данных.

На сегодня алгоритм UCR-DTW, предложенный учеными Калифорнийского университета в Риверсайде, является, по-видимому, наиболее быстрым последовательным алгоритмом поиска похожих подпоследовательностей. Идея данного алгоритма заключается в применении каскада предварительных оценок, позволяющих отбросить непохожую подпоследовательность до выполнения вычислительно сложной динамической трансформации шкалы времени. Существуют реализации данного алгоритма для FPGA, а в нашем исследовании алгоритм UCR-DTW адаптируется для сопроцессора Intel Xeon Phi.

Intel Xeon Phi представляет собой сопроцессор, основанный на архитектуре Intel Many Integrated Core (Intel MIC). Intel Xeon Phi содержит 61 ядро, которые соединяются высокопроизводительной шиной. Каждое ядро сопроцессора имеет 4 потока за счет технологии Hyper-Threading и 512-разрядные векторные АЛУ,

обеспечивающие в одной инструкции до 16 операций над типом float или до 8 операций над типом double. В силу совместимости сопроцессора с архитектурой x86 при разработке приложения для Intel Xeon Phi имеется возможность использовать стандартные инструменты и технологии параллельного программирования, предназначенные для процессоров Intel Xeon.

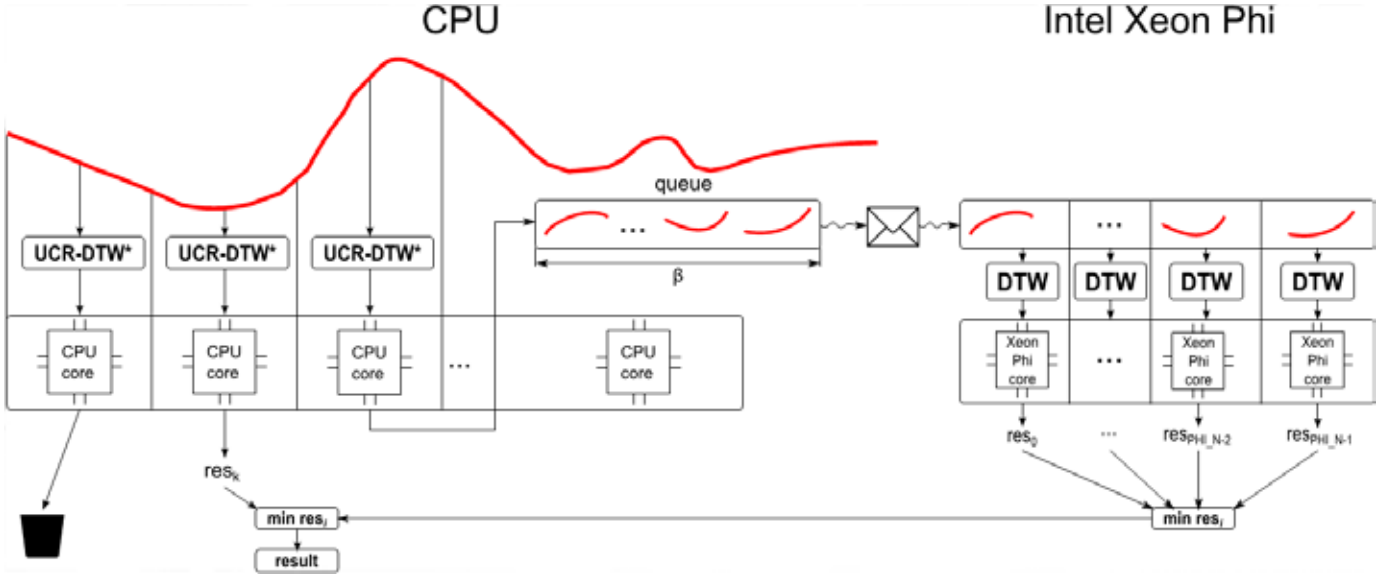

***Рис. 2**. Улучшенный параллельный алгоритм для сопроцессора*

Идея разработанного нами алгоритма (см. рис. 2) заключается в использовании на стороне процессора очереди подпоследовательностей, которые выгружаются на сопроцессор для вычисления динамической трансформации шкалы времени. Одна из нитей, выполняемых на ядрах процессора, объявляется мастером, остальные — рабочими. Мастер осуществляет выгрузку очереди на сопроцессор при ее заполнении. Рабочий вычисляет каскадные оценки и отбрасывает заведомо непохожую подпоследовательность либо добавляет эту подпоследовательность в очередь. Если очередь заполнена, то рабочий вычисляет динамическую трансформацию шкалы времени самостоятельно. По окончании выгрузки на процессор передается информация о найденных на сопроцессоре самых похожих подпоследовательностях. В итоге вычисляется самая похожая подпоследовательность среди найденных на процессоре и сопроцессоре.

Для исследования эффективности предложенного алгоритма нами проведена серия вычислительных экспериментов. В качестве аппаратной платформы экспериментов использовался вычислительный узел суперкомпьютера «Торнадо ЮУрГУ», характеристики которого приведены в табл. 1.

***Таблица 1.** Аппаратная платформа экспериментов*

Характеристики	Процессор	Сопроцессор
Модель	Intel Xeon X5680	Intel Xeon Phi SE10X
Количество ядер	6	61
Частота ядер,ГГц	3,33	1,1
Количество потоков на ядро	2	4
Производительность, TFLOPS	0,371	1,076

Эксперименты на синтетическом временном ряде, состоящем из 109 точек (см. рис. 3), показали преимущество улучшенной версии алгоритма.

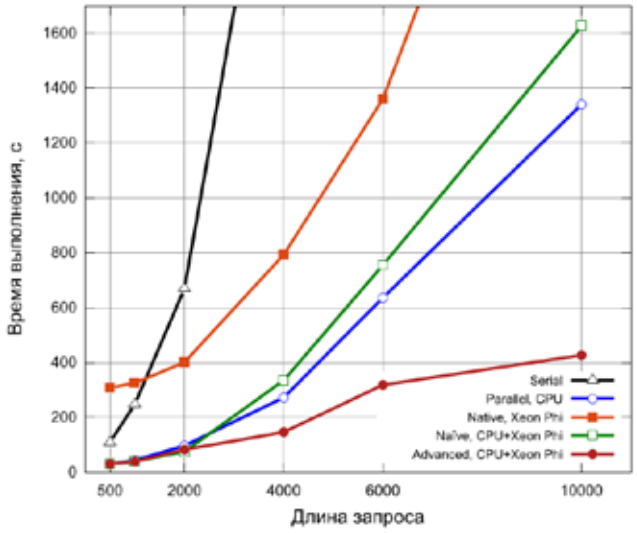

Рис. 3. *Производительность разработанного алгоритма на синтетических данных*

Эксперименты на реальных данных (см. рис. 4), в качестве которых использовались 2 · 108 точек данных ЭКГ (примерно 22 часа при частоте дискретизации 250 Гц), также показали преимущество улучшенной версии алгоритма.

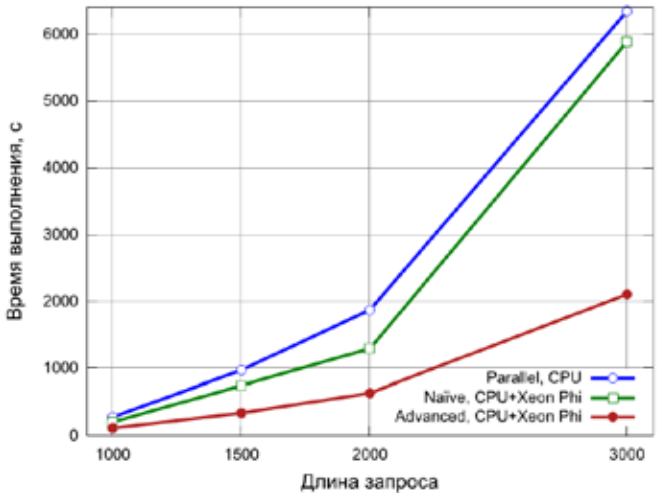

Рис. 4. *Производительность разработанного алгоритма на реальных данных*

Производительность разработанного алгоритма мы сравнили с аналогичными алгоритмами для GPU (NVIDIA Tesla C1060, 77,76 GFLOPS) и FPGA (Xilinx Virtex 5 LX—330, 65 GFLOPS) для запроса длиной 1024 точек данных, результаты показаны на рис. 5.

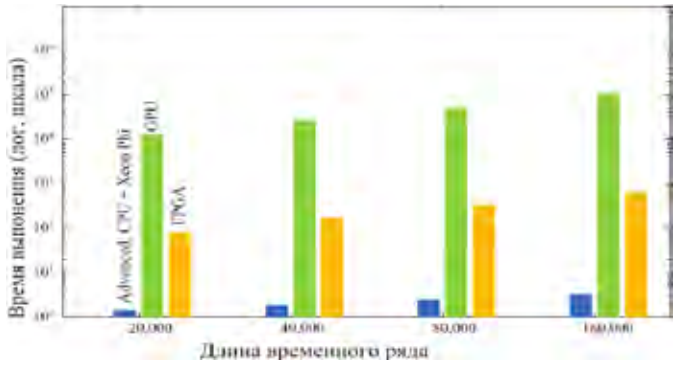

Рис. 5. *Сравнение разработанного алгоритма с алгоритмами для GPU и FPGA*