УДК 004.62, 004.65 DOI: 10.14529/cmse210305

ВНЕДРЕНИЕ КОНЦЕПЦИИ МАТРИЧНОГО ПРОФИЛЯ В РЕЛЯЦИОННУЮ СУБД ДЛЯ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ВРЕМЕННЫХ РЯДОВ

© 2021 Е.В. Иванова, М.Л. Цымблер

Южно-Уральский государственный университет
(454080 Челябинск, пр. им. В.И. Ленина, д. 76)
E-mail: elena.ivanova@susu.ru, mzym@susu.ru

Поступила в редакцию: 05.07.2021

В настоящее время большие временные ряды используются в широком спектре предметных областей. Современные системы управления базами данных временных рядов (СУБД-ВР) предлагают, однако, скромный набор встроенных инструментов и средств для интеллектуального анализа данных. Использование сторонних систем интеллектуального анализа временных рядов приводит в связи с этим к нежелательным накладным расходам на экспорт данных вне СУБД-ВР, преобразование данных и импорт результатов анализа. В то же время актуальной научной задачей является внедрение методов интеллектуального анализа данных в реляционные СУБД (РСУБД), которые доминируют на рынке средств управления данными. Однако пока отсутствуют разработки по внедрению методов интеллектуального анализа временных рядов в РСУБД. В статье предлагается подход к управлению и интеллектуальному анализу временных рядов внутри РСУБД на основе концепции матричного профиля. Матричный профиль представляет собой структуру данных, которая для каждой подпоследовательности временного ряда сохраняет индекс и расстояние до ее ближайшего соседа (подпоследовательности ряда, наиболее похожей на данную). Матричный профиль служит основой для обнаружения лейтмотивов (шаблонов), аномалий и других примитивов интеллектуального анализа временных рядов. Описанный подход реализован в РСУБД PostgreSQL. Представлены результаты вычислительных экспериментов, показавшие более высокую эффективность предложенного подхода по сравнению с СУБД-ВР InfluxDB и OpenTSDB.

Ключевые слова: временные ряды, матричный профиль, PostgreSQL, InfluxDB, OpenTSDB.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Иванова Е.В., Цымблер М.Л. Внедрение концепции матричного профиля в реляционную СУБД для интеллектуального анализа временных рядов // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2021. Т. 10, № 3. С. 72–87. DOI: 10.14529/cmse210305.

Введение

В настоящее время, несмотря на широкое использование систем NoSQL, реляционные СУБД (РСУБД) остаются основным инструментом для хранения и обработки данных в широком спектре предметных областей, занимая до 75 % рынка СУБД [5]. РСУБД по своей природе не предоставляют встроенных функций интеллектуального анализа данных, и в связи с этим актуальной задачей является разработка методов внедрения интеллектуального анализа данных в РСУБД [15]. Действительно, если РСУБД рассматривается только как инструмент обслуживания хранилища данных, то выполнение интеллектуального анализа данных будет связано со значительными накладным расходам на экспорт больших объемов данных из РСУБД в стороннюю аналитическую систему, изменение формата данных и импорт результатов анализа обратно в РСУБД. Отметим также, что интеллектуальный анализ данных внутри РСУБД позволяет без дополнительных накладных расходов использовать системные сервисы, заложенные в архитектуре СУБД (оптимизация исполнения запросов,

целостность и безопасность данных и др.).

Методы интеллектуального анализа данных в РСУБД охватывают широкий спектр задач: поиск шаблонов [3, 21], кластеризация [17, 27], анализ графов [12, 16] и др., а также разработка соответствующих многоцелевых библиотек и фреймворков для РСУБД [6, 8, 11, 19]. Детальный обзор методов интеграции интеллектуального анализа данных в РСУБД может быть найден в работе [2].

В данной статье рассматривается проблема внедрения в РСУБД методов интеллектуального анализа временных рядов. Временной ряд представляет собой хронологическую последовательность числовых значений, отражающих течение некоторого процесса или явления. Временные ряды возникают во многих предметных областях: мониторинг показателей функциональной диагностики организма человека, моделирование климата, финансовое прогнозирование, генная инженерия и др. Однако современные системы обработки временных рядов (СУБД-ВР) предоставляют достаточно узкий спектр встроенных средств интеллектуального анализа данных (как правило, для восстановления пропусков или/и прогноза значений временного ряда) [1]. В то же время, как показал предпринятый авторами тщательный поиск научной литературы, по-видимому, пока отсутствуют разработки, обеспечивающие интеллектуальный анализ временных рядов в РСУБД.

В работе предлагается подход к управлению и интеллектуальному анализу временных рядов внутри РСУБД, который использует концепцию матричного профиля временного ряда. Матричный профиль временного ряда [23] представляет собой структуру данных, которая используется для интеллектуального анализа ряда и неформально определяется следующим образом. Матричный профиль данного временного ряда представляет собой временной ряд, i-м элементом которого является расстояние i-й подпоследовательности исходного ряда до ее ближайшего соседа (ближайшей к ней подпоследовательности исходного ряда). Матричный профиль позволяет естественным образом обнаруживать лейтмотивы (повторяющиеся шаблоны) и аномалии временного ряда как подпоследовательности ряда, которым сопоставляются минимальные и максимальные элементы профиля соответственно. Матричный профиль также служит основой для обнаружения более сложных примитивов интеллектуального анализа временных рядов: смысловые сегменты [7], эволюционирующие шаблоны [25], типичные шаблоны [9] и др.

В настоящее время концепция матричного профиля и сопутствующие алгоритмы широко используются для интеллектуального анализа временных рядов в различных приложениях цифровой индустрии. Например, в работе [24] матричные профили временных рядов энергопотребления используются для обнаружения краж электроэнергии. В работе [20] матричный профиль временного ряда синхрофазора (прибора, выполняющего синхронизированное по времени измерение параметров энергосистемы) применен для автоматического распознавания событий в энергосистеме предприятия. В [14] с помощью матричного профиля обнаруживаются периоды аномального потребления электроэнергии. В работе [10] матричный профиль временного ряда показаний датчика вибрации применяется для мониторинга рабочего состояния промышленного вентилятора. В работе [18] матричный профиль временных рядов Интернета вещей используется для реализации технического обслуживания промышленного производства.

Статья организована следующим образом. В разделе 1 приводится формальное определение матричного профиля. В разделе 2 описан подход к интеллектуальному анализу временных рядов в РСУБД на основе концепции матричного профиля. В разделе 3 пред-

ставлены результаты вычислительных экспериментов по оценке эффективности предложенного подхода. В заключении резюмированы полученные результаты и указаны направления будущих исследований.

1. Матричный профиль временного ряда

В данном разделе приводится формальное определение матричного профиля в соответствии с оригинальными работами [20, 23].

Временной ряд представляет собой хронологически упорядоченную последовательность числовых значений: $T = (t_1, \ldots, t_n), t_i \in \mathbb{R}$.

Подпоследовательность $T_{i,m}$ временного ряда T представляет собой непрерывное подмножество T, состоящее из m элементов и начинающееся с позиции i: $T_{i,m} = (t_i, \ldots, t_{i+m-1})$, где $1 \leq i \leq n - m + 1$, $1 \leq m \ll n$.

Множество всех подпоследовательностей S_T^A временного ряда T является упорядоченным набором всех возможных подпоследовательностей длины m, содержащихся в T. Подпоследовательности сортируются в порядке возрастания индекса их первого элемента: $S_T^A = \{T_{1,m}, T_{2,m}, \ldots, T_{n-m+1,m}\}$.

Профиль расстояния D_i^T представляет собой вектор расстояний между заданной подпоследовательностью $T_{i,m}$ и каждой подпоследовательностью $T_{j,m}$ из множества всех подпоследовательностей: $D_i^T = (dist(T_{i,m}, T_{1,m}), \ldots, dist(T_{i,m}, T_{n-m+1,m}))$, где $dist(\cdot, \cdot)$ означает евклидово расстояние между нормализованными подпоследовательностями.

С помощью профиля расстояния выполняется поиск ближайшего соседа для каждой подпоследовательности временного ряда, за исключением тривиальных соседей. Подпоследовательность $T_{j,m}$ называется *ближайшим соседом* подпоследовательности $T_{i,m}$, если $dist(T_{i,m}, T_{j,m}) = min(D_i^T)$. Подпоследовательность $T_{j,m}$ является *тривиальным соседом* для $T_{i,m}$, если $|i - j| \leq \frac{m}{2}$.

На основе вышеприведенных определений матричный профиль формально определяется следующим образом. *Матричный профиль* P^T временного ряда T представляет собой вектор расстояний между каждой подпоследовательностью $T_{i,m}$ и ее ближайшим нетривиальным соседом: $P^T = (nn_{nt}(D_1^T), \ldots, nn_{nt}(D_{n-m+1}^T))$, где $nn_{nt}(D_i^T)$ означает минимальное расстояние между $T_{i,m}$ и ее нетривиальными соседями.

С матричным профилем ассоциируется дополнительная структура данных для определения местоположения ближайших нетривиальных соседей. *Индекс матричного профиля* I^T временного ряда T представляет собой вектор, хранящий индекс ближайшего нетривиального соседа для каждой подпоследовательности: $I^T = (I_1^T, \ldots, I_{n-m+1}^T)$, где $I_i^T = j$, если $nn_{nt}(D_i^T) = dist(T_{i,m}, T_{j,m})$.

Матричный профиль можно рассматривать как метаданные для аннотирования соответствующего временного ряда. Например, максимальное значение профиля соответствует аномальной подпоследовательности (называемой *диссонансом* [22]), тогда как минимальные значения соответствуют наиболее похожей паре подпоследовательностей в ряде (называемой *лейтмотивом*).

Матричный профиль представляет собой агностический (не зависящий от предметной области) инструмент интеллектуального анализа временного ряда, в котором исследователь должен указать лишь один параметр — длину подпоследовательности. Среди других преимуществ матричного профиля отметим, что он может вычисляться параллельно [26] и допускает инкрементное обновление [24].

2. Внедрение концепции матричного профиля в PostgreSQL

Внедрение концепции матричного профиля в РСУБД имеет следующие основные преимущества. Выполняя интеллектуальный анализ временных рядов в РСУБД, мы избавляемся от накладных расходов на экспорт-импорт больших данных. Кроме того, будучи вычисленным и сохраненным в базе данных один раз, матричный профиль и связанные с ним примитивы интеллектуального анализа временных рядов могут затем использоваться многократно.

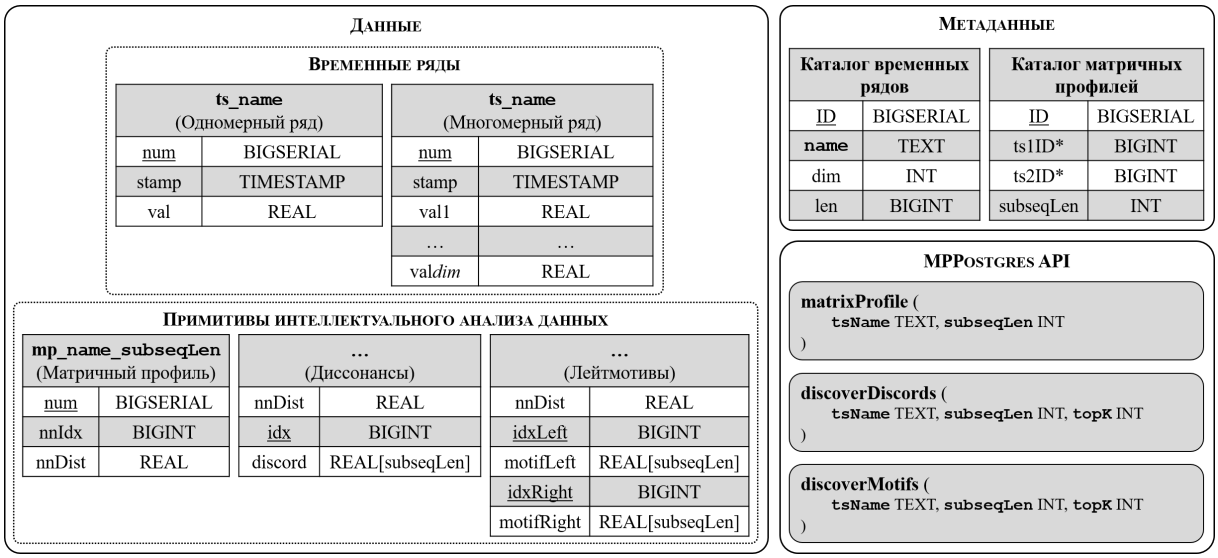

Рис. 1. Структура базы данных для управления матричным профилем в РСУБД

Внедрение матричного профиля в РСУБД предполагает создание *базы данных временных рядов*, представленной на рис. 1. Для хранения данных временных рядов предусмотрены *набор таблиц одномерных временных рядов* и *набор таблиц многомерных временных рядов*. Каждый одномерный временной ряд реализован в виде таблицы со следующими основными столбцами: индекс элемента (первичный ключ), отметка времени и собственно вещественный элемент ряда. Многомерный временной ряд реализован в виде таблицы с аналогичной структурой, где каждое измерение представлено в виде отдельного столбца с вещественными значениями.

Для хранения матричных профилей хранимых временных рядов предусмотрен *набор таблиц матричных профилей*. Каждая такая таблица хранит один матричный профиль для заданной длины подпоследовательности и содержит один вещественный столбец для хранения расстояния до ближайшего соседа и один целочисленный столбец для хранения индекса ближайшего соседа.

Помимо указанных выше таблиц, в базе данных предусмотрены следующие таблицы метаданных. В *таблице каталога временных рядов* хранится следующая основная информация о каждом временном ряде: уникальный идентификатор, имя (таблицы с данными), размерность и длина. В *таблице каталога матричных профилей* хранятся следующие основные данные о хранимых матричных профилях: уникальный идентификатор, длина подпоследовательности, внешний ключ на запись о соответствующем временном ряде в таблице каталога временных рядов.

Именование объектов базы данных временных рядов отражает связь матричного профиля с временным рядом и длиной подпоследовательности. Например, пусть данные вре-

менного ряда, получаемые с акустического датчика, требуется анализировать, используя длины подпоследовательности 64 и 128. Тогда в базе данных будут хранится таблицы `ts_acoustic`, `mp_acoustic_64` и `mp_acoustic_128` для хранения собственно временного ряда и двух его матричных профилей для соответственно.

```
1  -- Найти диссонансы временного ряда и сохранить результат в таблице
2  discoverDiscords(tsName TEXT, subseqLen INT, topK INT) RETURNS
3    TABLE (nnDist REAL, idx BIGINT, discord REAL[subseqLen])
4  -- Найти лейтмотивы временного ряда и сохранить результат в таблице
5  discoverMotifs(tsName TEXT, subseqLen INT, topK INT) RETURNS
6    TABLE (nnDist REAL, idxLeft BIGINT, motifLeft REAL[subseqLen],
7      idxRight BIGINT, motifRight REAL[subseqLen])
8  -- Вычисление матричного профиля
9  matrixProfile(tsName TEXT, subseqLen INT) RETURNS
10   TABLE (num BIGINT, nnDist REAL, nnIdx BIGINT)
```

Рис. 2. API для интеллектуального анализа данных временных рядов, MPPostgres

Помимо базы данных временных рядов, предлагаемый подход предусматривает *интерфейс прикладного программиста* (API, Application Programming Interface) для интеллектуального анализа временных рядов. API представляет собой библиотеку функций, обеспечивающих вычисление матричного профиля, обнаружение диссонансов, лейтмотивов временного ряда и др. аналитических примитивов для специфицированной длины подпоследовательности ряда и представление указанных примитивов в табличном виде (см. рис. 2). РСУБД должна поддерживать соответствующее процедурное расширение языка запросов SQL.

Диссонанс представляется в виде кортежа со следующими атрибутами: индекс диссонанса во временном ряде, расстояние до ближайшего соседа и собственно диссонанс в виде массива. Лейтмотив представляет собой кортеж со следующими атрибутами: левая и правая подпоследовательности-части лейтмотива, их индексы во временном ряде и расстояние между ними. Результат выполнения библиотечной функции может храниться в базе данных в виде таблицы или использоваться как представление (*view*, виртуальная таблица). Диссонансы и лейтмотивы вычисляются с помощью SQL-запросов, которые выбирают строки соответствующей таблицы матричного профиля с максимальными и минимальными значениями расстояния до ближайшего соседа (столбец `nnDist`) соответственно. Библиотечная функция вычисления матричного профиля реализуется как обертка над алгоритмом в оперативной памяти, предложенным в работе [26]. Указанная функция реализуется таким образом, что ее вызов осуществляется только в том случае, если соответствующая таблица матричного профиля еще не создана. API может быть дополнен функциями для поиска других примитивов интеллектуального анализа временных рядов (эволюционирующие шаблоны [25], типичные шаблоны [9] и др.).

Представленный подход был реализован нами для свободной СУБД PostgreSQL как ее расширение и получил название *MPPostgres*. Реализация выполнена на языке PL/pgSQL, который представляет собой полнофункциональный язык программирования, поддерживаемый PostgreSQL, и обеспечивает более широкий спектр процедурных возможностей, чем традиционный SQL. Примерный сценарий работы с MPPostgres показан на рис. 3: создание таблицы для хранения временного ряда показаний датчика температуры и вставка в нее данных, подключение MPPostgres к PostgreSQL, поиск диссонансов и лейтмотивов на основе матричного профиля (для длины подпоследовательности 128).

```
-- Создание таблицы временного ряда и вставка в нее данных
CREATE TABLE ts_Temperature (num BIGSERIAL, stamp TIMESTAMP, val REAL)
INSERT INTO ts_Temperature VALUES (NOW(), 21.0)
...
-- Подключение MPPostgres и вычисление матричного профиля ряда
CREATE EXTENSION MPPostgres
SELECT * FROM matrixProfile('ts_Temperature', 128)
-- Поиск диссонансов и лейтмотивов, формирование их top-100 представлений
CREATE VIEW discords_Temperature_128 AS
  SELECT * FROM discoverDiscords('ts_Temperature', 128, 100)
CREATE VIEW motifs_Temperature_128 AS
  SELECT * FROM discoverMotifs('ts_Temperature', 128, 100)
-- Вывод top-10 диссонансов и лейтмотивов
SELECT * FROM discords_Temperature_128 LIMIT 10
SELECT * FROM motifs_Temperature_128 LIMIT 10
```

Рис. 3. Пример использования MPPostgres для интеллектуального анализа временного ряда

3. Вычислительные эксперименты

Разработанный подход был реализован нами для свободной СУБД PostgreSQL 13.2 и были проведены эксперименты по исследованию его эффективности на платформе рабочей станции со следующими характеристиками: процессор Intel Xeon Gold 6254 4 ГГц, оперативная память 64 Гб, дисковая память 1 Тб. В экспериментах нами были рассмотрены два реальных приложения цифровой индустрии, связанные с анализом сенсорных данных. В указанных приложениях мы полагаем, что сенсорные данные хранятся и анализируются в PostgreSQL, и оцениваем быстродействие выполнения анализа данных. Для сравнения собственной разработки с аналогами нами также была выполнена реализация упомянутых приложений на основе СУБД-ВР InfluxDB и OpenTSDB. Конкурирующие аналоги исполняются в соответствии со следующим сценарием: сенсорные данные сперва экспортируются и преобразуются в формат, необходимый для работы стороннего аналитического приложения [4], и после выполнения им анализа данных полученные результаты импортируются в СУБД-ВР.

3.1. Выявление периодов аномального энергопотребления

В работе [14] авторы применили концепцию матричного профиля для выявления периодов аномального энергопотребления в административном, учебном и лабораторном корпусах университетского кампуса в Цюрихе (известных под псевдонимами Тревис, Трейси и Тери соответственно). Авторами использовался набор временных рядов потребления электроэнергии Building Data Genome (BDG) [13], в которых показания снимаются ежечасно. Недельным периодам аномального энергопотребления соответствуют диссонансы с длиной 168 элементов (7 дней по 24 показания).

Реализация описанного приложения в MPPostgres представлена на рис. 4. Для поиска диссонансов выполняется вызов функции разработанной библиотеки (трижды по числу корпусов кампуса, с соответствующими параметрами). Далее листинг демонстрирует способ реализации указанной функции. Строки результирующей таблицы формируются с помощью следующего цикла. Используя SQL-запрос, выполняется отбор K строк с максимальным расстоянием до подпоследовательности-ближайшего соседа из таблицы матричного профиля, после чего мы берем значения полей индекса и расстояния для строки результирующей

таблицы. Наконец, используя найденный индекс, с помощью SQL-запроса выполняется отбор элементов соответствующей подпоследовательности-диссонанса.

```
-- Найти периоды аномального энергопотребления как top-3 диссонанса
SELECT discoverDiscords('Office_Travis', 168, 3);
SELECT discoverDiscords('Classroom_Terrie', 168, 3);
SELECT discoverDiscords('Laboratory_Tracy', 168, 3);
-- Реализация функции поиска диссонансов
CREATE FUNCTION discoverDiscords(tsName TEXT, subseqLen INT, topK INT)
  RETURNS TABLE(nnDist REAL, idx BIGINT, discord REAL[subseqLen]) AS
DECLARE
  tsTabName, mpTabName TEXT;
  tsRow, mpRow RECORD;
BEGIN
  tsTabName:='ts_' || tsName; mpTabName:='mp_' || tsName || '_' || subseqLen;
  FOR mpRow IN EXEC_QUERY
    -- Получить K строк таблицы матричного профиля с max расстоянием
    SELECT * FROM mpTabName ORDER BY nnDist DESC LIMIT topK
  nnDist:=mpRow.nnDist; idx:=mpRow.num;
  FOR tsRow IN EXEC_QUERY
    -- Получить диссонанс из таблицы временного ряда
    SELECT val FROM tsTabName
    WHERE num BETWEEN idx AND idx+subseqLen-1
  discord:=array_append(discord, tsRow.val);
  RETURN NEXT ROW;
END;
```

Рис. 4. Реализация поиска периодов аномального энергопотребления в MPPostgres

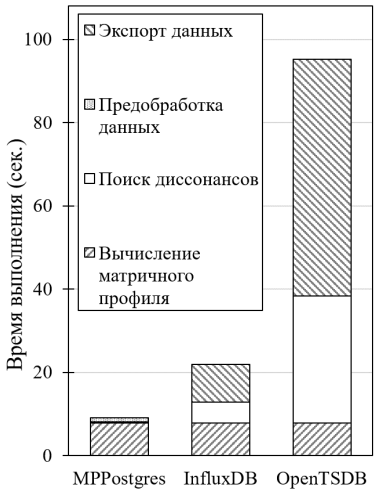

Рис. 5. Производительность различных СУБД-ВР при поиске периодов аномального энергопотребления

Результаты эксперимента представлены на рис. 5 (в иллюстративных целях набор BDG был реплицирован, чтобы соответствовать данным энергопотребления за 32 года). Можно видеть, что все конкуренты одинаково быстро вычисляют матричный профиль ряда. MPPostgres требует небольшого времени на выполнение запросов SQL, извлекающих данные для их подготовки к вычислениям. InfluxDB и OpenTSDB как решения, выполняющие

анализ данных вне СУБД-ВР, терпят значительные накладные расходы на экспорт данных, в отличие от MPPostgres. MPPostgres также превосходит конкурентов на этапе обнаружения диссонансов, поскольку использует встроенную индексацию базы данных по первичному ключу таблицы при поиске значений top-K и извлечении строк из таблиц. Следует отметить также, что при типичном сценарии, когда матричный профиль уже вычислен и сохранен в соответствующей таблице, предлагаемый подход будет еще более значительно превосходить конкурентов.

3.2. Мониторинг состояния промышленного оборудования

В работе [10] авторы применяют концепцию матричного профиля для решения задачи мониторинга промышленного вытяжного вентилятора, анализируя данные временного ряда, собранные установленным на вентиляторе датчиком вибрации. При сборе данных преднамеренно контролируется продолжительность и скорость обдува вентилятора. Переключение режимов вентилятора влияет на вибрации, в то время как лейтмотивы временного ряда указывают на моменты, когда состояние машины меняется. Поиск лейтмотивов выполняется с помощью матричного профиля ряда и вычисляется общая продолжительность временных интервалов, когда вентилятор находится в режимах «высокая скорость» и «низкая скорость», а также соотношение длительности таких интервалов. Полученные авторами результаты показывают, что соотношение, вычисленное с помощью матричного профиля, практически идентично данным натурных экспериментов.

Нами проведены эксперименты, аналогичные описанным в работе [10], поскольку ее авторы не предоставляют в отрытом доступе набор данных. Для экспериментов были собраны данные датчика вибрации, установленного на малогабаритной дробильной машине, которая находится в Южно-Уральском государственном университете и используется для подготовки инженеров. Собранный временной ряд состоит из более чем 240 тыс. элементов, соответствующих трем минутам работы машины. В ходе натурного эксперимента дробильная машина запускалась восемь раз, и статус машины менялся шестнадцать раз с «дробление остановлено» на «дробление выполняется» и обратно. Подобно работе [10], соотношение общей длительности временных интервалов, когда машина работает в разных режимах, вычисленное в наших экспериментах с помощью поиска лейтмотивов, практически совпадает с данными натурных экспериментов.

Реализация описанного случая в MPPostgres показана на рис. 6. После вычисления матричного профиля входного временного ряда используется SQL-запрос для извлечения K первых лейтмотивов в виде первых K строк таблицы матричного профиля с минимальным расстоянием до ближайшего соседа соответствующей подпоследовательности. Затем временной ряд сканируется слева направо по индексам найденных лейтмотивов, вычисляя длину интервала между предыдущим и текущим лейтмотивом, и поочередно добавляя результат к общей продолжительности, которая указывает, когда машина работает в первом или втором режиме. Наконец, после аналогичных однократных вычислений по оставшейся части входного временного ряда мы получаем результирующее соотношение.

```
1  CREATE FUNCTION trackMachineStatus(tsName TEXT, subseqLen INT, topK INT)
2    RETURNS ratio REAL[2] AS
3  DECLARE
4    status1, status2, tsLength, prevMotifIdx, i BIGINT;
5    tsTabName, mpTabName TEXT; motifRow RECORD;
6  BEGIN
7    prevMotifIdx:=0; i:=0; status1:=0; status2:=0;
8    tsTabName:='ts_' || tsName; mpTabName:='mp_' || tsName || '_' || subseqLen;
9    FOR motifRow IN EXEC_QUERY
10   -- Найти top-K лейтмотивы ряда, ...
11   SELECT * FROM (
12     SELECT num FROM _matrixProfile(tsName, subseqLen)
13     ORDER BY nnDist LIMIT topK)
14   ORDER BY num
15   -- ... сканировать их и вычислить длительность каждого статуса
16   i:=i+1;
17   IF i%2=1 THEN
18     status1:=status1+motifRow.idxLeft-prevMotifIdx;
19   ELSE
20     status2:=status2+motifRow.idxLeft-prevMotifIdx;
21   prevMotifIdx:=motifRow.idxLeft;
22   -- Вычислить длительность для остатка временного ряда, выдать результат
23   EXEC_QUERY SELECT COUNT(*) FROM tsTabName INTO tsLength;
24   IF i%2=1 THEN
25     status1:=status1+tsLength-prevMotifIdx;
26   ELSE
27     status2:=status2+tsLength-prevMotifIdx;
28   ratio[1]:=status1*100/tsLength; ratio[2]:=status2*100/tsLength;
29   RETURN ratio;
30 END;
```

Рис. 6. Реализация мониторинга состояния промышленного оборудования в MPPostgres

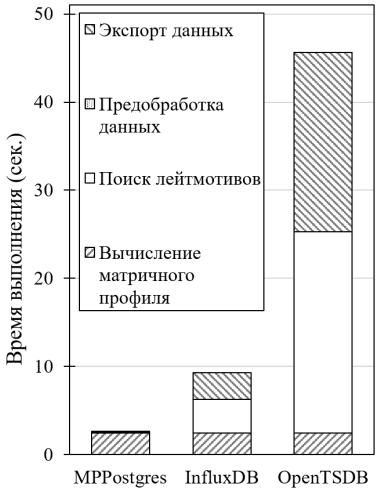

Рис. 7. Производительность различных СУБД-ВР при мониторинге состояния промышленного оборудования

На рис. 7 представлены результаты эксперимента. Как и в предыдущем случае, MPPostgres превосходит конкурентов, поскольку последние вынуждены экспортировать

данные перед их обработкой. Кроме того, в типичном сценарии, когда матричный профиль уже вычислен и сохранен в таблице матричного профиля, MPPostgres будет демонстрировать еще более высокую производительность.

Заключение

В статье представлен подход к управлению и интеллектуальному анализу временных рядов внутри реляционной СУБД (РСУБД), основанный на концепции матричного профиля временного ряда [23]. Матричный профиль представляет собой структуру данных, которая резюмирует временной ряд, сохраняя для каждой подпоследовательности временного ряда индекс и расстояние до ее ближайшего соседа (подпоследовательности ряда, наиболее похожей на данную). Матричный профиль служит основой для обнаружения лейтмотивов (шаблонов) и диссонансов (аномалий) временного ряда.

В рамках подхода данные одномерных и многомерных временных рядов хранятся в реляционных таблицах, а метаданные предоставлены таблицей каталога временных рядов. Кроме того, предусмотрены таблицы для хранения матричных профилей и каталога матричных профилей. Предложенный подход реализован в виде расширения свободной РСУБД PostgreSQL, получившего название MPPostgres. MPPostgres предоставляет прикладному программисту набор библиотечных функций для вычисления матричного профиля и следующих примитивов интеллектуального анализа временных рядов, представляющих результаты в виде реляционных таблиц: лейтмотивы и диссонансы.

Для оценки эффективности предложенного подхода нами проведены вычислительные эксперименты, использующие сценарии и данные временных рядов из двух реальных приложений цифровой индустрии: выявление периодов аномального потребления электроэнергии в зданиях и мониторинг состояния промышленного оборудования. Результаты экспериментов показали, что MPPostgres опережает по эффективности современные СУБД-ВР InfluxDB и OpenTSDB, поскольку в предложенном нами подходе отсутствуют накладные расходы на экспорт-импорт и преобразование данных.

В дальнейших исследованиях мы планируем дополнить MPPostgres более сложными примитивами интеллектуального анализа временных рядов (эволюционирующие шаблоны, типичные шаблоны и др.).

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (грант № 20-07-00140) и Министерства науки и высшего образования РФ (государственное задание FENU-2020-0022).

Литература

1. Иванова Е.В., Цымблер М.Л. Обзор современных систем обработки временных рядов // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2020. Т. 9, № 4. С. 79–97. DOI: 10.14529/cmse200406.

2. Цымблер М.Л. Обзор методов интеграции интеллектуального анализа данных в СУБД // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2019. Т. 8, № 2. С. 32–62. DOI: 10.14529/cmse190203.

3. Baralis E., Cerquitelli T., Chiusano S. Index Support for Frequent Itemset Mining in a Relational DBMS // Proceedings of the 21st International Conference on Data Engineering, ICDE 2005 (Tokyo, Japan, April 5–8, 2005). IEEE Computer Society, 2005. P. 754–765. DOI:

10.1109/ICDE.2005.80.

4. Benschoten A.V., Ouyang A., Bischoff F., *et al.* MPA: a novel cross-language API for time series analysis // Journal of Open Source Software. 2020. Vol. 5, no. 49. Article 2179. DOI: 10.21105/joss.02179.

5. DB-Engines Ranking of Time Series DBMS. URL: https://db-engines.com/en/ranking/time+series+dbms (дата обращения: 26.06.2021).

6. Feng X., Kumar A., Recht B., *et al.* Towards a unified architecture for in-RDBMS analytics // Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2012 (Scottsdale, AZ, USA, May 20–24, 2012). P. 325–336. DOI: 10.1145/2213836.2213874.

7. Gharghabi S., Ding Y., Yeh C.M., *et al.* Matrix Profile VIII: Domain Agnostic Online Semantic Segmentation at Superhuman Performance Levels // 2017 IEEE International Conference on Data Mining, ICDM 2017 (New Orleans, LA, USA, November 18–21, 2017). P. 117–126. DOI: 10.1109/ICDM.2017.21.

8. Hellerstein J.M., Re C., Schoppmann F., *et al.* The MADlib analytics library or MAD skills, the SQL // PVLDB. 2012. Vol. 5, no. 12. P. 1700–1711. DOI: 10.14778/2367502.2367510.

9. Imani S., Madrid F., Ding W., *et al.* Matrix Profile XIII: Time Series Snippets: A New Primitive for Time Series Data Mining // 2018 IEEE International Conference on Big Knowledge, ICBK 2018 (Singapore, November 17–18, 2018). IEEE Computer Society, 2018. P. 382–389. DOI: 10.1109/ICBK.2018.00058.

10. Lee Y.Q., Beh W.L., Ooi B.Y. Tracking Operation Status of Machines through Vibration Analysis using Motif Discovery // Journal of Physics: Conference Series. 2020. Vol. 1529. Article 052005. DOI: 10.1088/1742-6596/1529/5/052005.

11. Mahajan D., Kim J.K., Sacks J., *et al.* In-RDBMS Hardware Acceleration of Advanced Analytics // Proc. VLDB Endow. 2018. Vol. 11. P. 1317–1331. DOI: 10.14778/3236187.3236188.

12. McCaffrey J.D. A Hybrid System for Analyzing Very Large Graphs // 9th International Conference on Information Technology: New Generations, ITNG 2012 (Las Vegas, Nevada, USA, 16–18 April, 2012). IEEE Computer Society, 2012. P. 253–257. DOI: 10.1109/ITNG.2012.43.

13. Miller C., Meggers F. The Building Data Genome Project: An open, public data set from non-residential building electrical meters // Energy Procedia. 2017. Vol. 122. P. 439–444. DOI: 10.1016/j.egypro.2017.07.400.

14. Nichiforov C., Stancu I., Stamatescu I., *et al.* Information Extraction Approach for Energy Time Series Modelling // 24th International Conference on System Theory, Control and Computing, ICSTCC 2020 (Sinaia, Romania, October 8–10, 2020). IEEE, 2020. P. 886–891. DOI: 10.1109/ICSTCC50638.2020.9259635.

15. Ordonez C. Can we analyze big data inside a DBMS? // Proceedings of the 16th international workshop on Data warehousing and OLAP, DOLAP 2013 (San Francisco, CA, USA, October 28, 2013). ACM, 2013. P. 85–92. DOI: 10.1145/2513190.2513198.

16. Pan C.S., Zymbler M.L. Very Large Graph Partitioning by Means of Parallel DBMS // Advances in Databases and Information Systems – 17th East European Conference, ADBIS

2013 (Genoa, Italy, September 1–4, 2013). Lecture Notes in Computer Science. Vol. 8133. Springer, 2013. P. 388–399. DOI: 10.1007/978-3-642-40683-6_29.

17. Pelekis N., Tampakis P., Vodas M., *et al.* In-DBMS Sampling-based Sub-trajectory Clustering // Proceedings of the 20th International Conference on Extending Database Technology, EDBT 2017 (Venice, Italy, March 21–24, 2017). OpenProceedings.org, 2017. P. 632–643. DOI: 10.5441/002/edbt.2017.84.

18. Pizoń J., Kulisz M., Lipski J. Matrix profile implementation perspective in Industrial Internet of Things production maintenance application // Journal of Physics: Conference Series. 2021. Vol. 1736. Article 012036. DOI: 10.1088/1742-6596/1736/1/012036.

19. Rechkalov T., Zymbler M.L. Integrating DBMS and Parallel Data Mining Algorithms for Modern Many-Core Processors // Data Analytics and Management in Data Intensive Domains – XIX International Conference, DAMDID/RCDL 2017 (Moscow, Russia, October 10–13, 2017). Communications in Computer and Information Science. Vol. 822. Springer, 2017. P. 230–245. DOI: 10.1007/978-3-319-96553-6_17.

20. Shi J., Yu N., Keogh E., *et al.* Discovering and Labeling Power System Events in Synchrophasor Data with Matrix Profile // 2019 IEEE Sustainable Power and Energy Conference (iSPEC) (Beijing, China, November 21–23, 2019). Article 19303617. DOI: 10.1109/iSPEC48194.2019.8975286.

21. Sidló C.I., Lukács A. Shaping SQL-Based Frequent Pattern Mining Algorithms // Knowledge Discovery in Inductive Databases, 4th International Workshop, KDID 2005 (Porto, Portugal, October 3, 2005), Revised Selected and Invited Papers. Lecture Notes in Computer Science. Vol. 3933. Springer, 2005. P. 188–201. DOI: 10.1007/11733492_11.

22. Yankov D., Keogh E.J., Rebbapragada U. Disk aware discord discovery: finding unusual time series in terabyte sized datasets // Knowledge and Information Systems. 2008. Vol. 17. P. 241–262. DOI: 10.1109/ICDM.2007.61.

23. Yeh C.-C.M., Zhu Y., Ulanova L., *et al.* Time series joins, motifs, discords and shapelets: a unifying view that exploits the matrix profile // Data Min. Knowl. Discov. 2018. Vol. 32, no. 1. P. 83–123. DOI: 10.1007/s10618-017-0519-9.

24. Zhu Y., Gharghabi S., Silva D.F., *et al.* The Swiss army knife of time series data mining: ten useful things you can do with the matrix profile and ten lines of code // Data Mining and Knowledge Discovery. 2020. Vol. 34. P. 949–979. DOI: 10.1007/s10618-019-00668-6.

25. Zhu Y., Imamura M., Nikovski D., *et al.* Matrix Profile VII: Time Series Chains: A New Primitive for Time Series Data Mining // 2017 IEEE International Conference on Data Mining, ICDM 2017 (New Orleans, LA, USA, November 18–21, 2017). P. 695–704. DOI: 10.1109/ICDM.2017.79.

26. Zhu Y., Zimmerman Z., Senobari N.S., *et al.* Matrix Profile II: Exploiting a Novel Algorithm and GPUs to Break the One Hundred Million Barrier for Time Series Motifs and Joins // IEEE 16th International Conference on Data Mining, ICDM 2016 (Barcelona, Spain, December 12–15, 2016). IEEE Computer Society, 2016. P. 739–748. DOI: 10.1109/ICDM.2016.0085.

27. Zymbler M.L., Kraeva Y., Grents A., *et al.* An Approach to Fuzzy Clustering of Big Data Inside a Parallel Relational DBMS // Data Analytics and Management in Data Intensive Domains – 21st International Conference, DAMDID/RCDL 2019 (Kazan, Russia, October 15–

18, 2019). Communications in Computer and Information Science. Vol. 1223. Springer, 2019. P. 211–223. DOI: 10.1007/978-3-030-51913-1_14.

Иванова Елена Владимировна, к.ф.-м.н., кафедра системного программирования, Южно-Уральский государственный университет (национальный исследовательский университет) (Челябинск, Российская Федерация)

Цымблер Михаил Леонидович, д.ф.-м.н., доцент, кафедра системного программирования, Южно-Уральский государственный университет (национальный исследовательский университет) (Челябинск, Российская Федерация)

DOI: 10.14529/cmse210305

EMBEDDING OF THE MATRIX PROFILE CONCEPT INTO A RELATIONAL DBMS FOR TIME SERIES MINING

© 2021 E.V. Ivanova, M.L. Zymbler

South Ural State University (pr. Lenina 76, Chelyabinsk, 454080 Russia)

E-mail: elena.ivanova@susu.ru, mzym@susu.ru

Received: 05.07.2021

Currently, large time series are used in a wide range of subject areas. Modern time series DBMSs (TSDBMS) offer, however, a modest set of built-in tools for data mining. The use of third-party time series mining systems to undesirable overhead costs for exporting data outside the TSDBMS, converting data and importing analysis results. At the same time, there is a topical issue of the embedding of data mining methods into relational DBMSs (RDBMS), which dominate the market of data management tools. However, there are still no developments of time series mining methods in RDBMS. The article proposes an approach to the management and mining of time series data within the RDBMS based on the matrix profile concept. A matrix profile is a data structure that, for each subsequence of a time series, stores the index of and the distance to its nearest neighbor. The matrix profile serves as the basis for detecting motifs, anomalies and other primitives of time series mining. The proposed approach is implemented in the PostgreSQL RDBMS. The experimental results showed a higher efficiency of the proposed approach compared to the TSDBMS InfluxDB and OpenTSDB.

Keywords: time series, matrix profile, PostgreSQL, InfluxDB, OpenTSDB.

FOR CITATION

Ivanova E.V., Zymbler M.L. Embedding of the Matrix Profile Concept Into a Relational DBMS for Time Series Mining. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering.* 2021. Vol. 10, no. 3. P. 72–87. (in Russian) DOI: 10.14529/cmse210305.

References

1. Ivanova E.V., Zymbler M.L. Overview of Modern Time Series Management Systems. Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering. 2020. Vol. 9, no. 4. P. 79–97. DOI: 10.14529/cmse200406. (in Russian)

2. Zymbler M.L. Overview of Methods for Integrating Data Mining into DBMS. Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering.

2019. Vol. 8, no. 2. P. 32–62. DOI: 10.14529/cmse190203. (in Russian)

3. Baralis E., Cerquitelli T., Chiusano S. Index Support for Frequent Itemset Mining in a Relational DBMS. Proceedings of the 21st International Conference on Data Engineering, ICDE 2005 (Tokyo, Japan, April 5–8, 2005). IEEE Computer Society, 2005. P. 754–765. DOI: 10.1109/ICDE.2005.80.

4. Benschoten A.V., Ouyang A., Bischoff F., *et al.* MPA: a novel cross-language API for time series analysis. Journal of Open Source Software. 2020. Vol. 5, no. 49. Article 2179. DOI: 10.21105/joss.02179.

5. DB-Engines Ranking of Time Series DBMS. URL: https://db-engines.com/en/ranking/time+series+dbms Available at: `https://docs.timescale.com/` (accessed: 26.06.2021).

6. Feng X., Kumar A., Recht B., *et al.* Towards a unified architecture for in-RDBMS analytics. Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2012 (Scottsdale, AZ, USA, May 20–24, 2012). P. 325–336. DOI: 10.1145/2213836.2213874.

7. Gharghabi S., Ding Y., Yeh C.M., *et al.* Matrix Profile VIII: Domain Agnostic Online Semantic Segmentation at Superhuman Performance Levels. 2017 IEEE International Conference on Data Mining, ICDM 2017 (New Orleans, LA, USA, November 18–21, 2017). P. 117–126. DOI: 10.1109/ICDM.2017.21.

8. Hellerstein J.M., Re C., Schoppmann F., *et al.* The MADlib analytics library or MAD skills, the SQL. PVLDB. 2012. Vol. 5, no. 12. P. 1700–1711. DOI: 10.14778/2367502.2367510.

9. Imani S., Madrid F., Ding W., *et al.* Matrix Profile XIII: Time Series Snippets: A New Primitive for Time Series Data Mining. 2018 IEEE International Conference on Big Knowledge, ICBK 2018 (Singapore, November 17–18, 2018). IEEE Computer Society, 2018. P. 382–389. DOI: 10.1109/ICBK.2018.00058.

10. Lee Y.Q., Beh W.L., Ooi B.Y. Tracking Operation Status of Machines through Vibration Analysis using Motif Discovery. Journal of Physics: Conference Series. 2020. Vol. 1529. Article 052005. DOI: 10.1088/1742-6596/1529/5/052005.

11. Mahajan D., Kim J.K., Sacks J., *et al.* In-RDBMS Hardware Acceleration of Advanced Analytics. Proc. VLDB Endow. 2018. Vol. 11. P. 1317–1331. DOI: 10.14778/3236187.3236188.

12. McCaffrey J.D. A Hybrid System for Analyzing Very Large Graphs. 9th International Conference on Information Technology: New Generations, ITNG 2012 (Las Vegas, Nevada, USA, April 16–18, 2012). IEEE Computer Society, 2012. P. 253–257. DOI: 10.1109/ITNG.2012.43.

13. Miller C., Meggers F. The Building Data Genome Project: An open, public data set from non-residential building electrical meters. Energy Procedia. 2017. Vol. 122. P. 439–444. DOI: 10.1016/j.egypro.2017.07.400.

14. Nichiforov C., Stancu I., Stamatescu I., *et al.* Information Extraction Approach for Energy Time Series Modelling. 24th International Conference on System Theory, Control and Computing, ICSTCC 2020 (Sinaia, Romania, October 8–10, 2020). IEEE, 2020. P. 886–891. DOI: 10.1109/ICSTCC50638.2020.9259635.

15. Ordonez C. Can we analyze big data inside a DBMS? Proceedings of the 16th international workshop on Data warehousing and OLAP, DOLAP 2013 (San Francisco, CA, USA, October 28, 2013). ACM, 2013. P. 85–92. DOI: 10.1145/2513190.2513198.

16. Pan C.S., Zymbler M.L. Very Large Graph Partitioning by Means of Parallel DBMS. Advances in Databases and Information Systems – 17th East European Conference, ADBIS 2013 (Genoa, Italy, September 1–4, 2013). Lecture Notes in Computer Science. Vol. 8133. Springer, 2013. P. 388–399. DOI: 10.1007/978-3-642-40683-6_29.

17. Pelekis N., Tampakis P., Vodas M., *et al.* In-DBMS Sampling-based Sub-trajectory Clustering. Proceedings of the 20th International Conference on Extending Database Technology, EDBT 2017 (Venice, Italy, March 21–24, 2017). OpenProceedings.org, 2017. P. 632–643. DOI: 10.5441/002/edbt.2017.84.

18. Pizoń J., Kulisz M., Lipski J. Matrix profile implementation perspective in Industrial Internet of Things production maintenance application. Journal of Physics: Conference Series. 2021. Vol. 1736. Article 012036. DOI: 10.1088/1742-6596/1736/1/012036.

19. Rechkalov T., Zymbler M.L. Integrating DBMS and Parallel Data Mining Algorithms for Modern Many-Core Processors. Data Analytics and Management in Data Intensive Domains – XIX International Conference, DAMDID/RCDL 2017 (Moscow, Russia, October 10–13, 2017). Communications in Computer and Information Science. Vol. 822. Springer, 2017. P. 230–245. DOI: 10.1007/978-3-319-96553-6_17.

20. Shi J., Yu N., Keogh E., *et al.* Discovering and Labeling Power System Events in Synchrophasor Data with Matrix Profile. 2019 IEEE Sustainable Power and Energy Conference (iSPEC) (Beijing, China, November 21–23, 2019). Article 19303617. DOI: 10.1109/iSPEC48194.2019.8975286.

21. Sidló C.I., Lukács A. Shaping SQL-Based Frequent Pattern Mining Algorithms. Knowledge Discovery in Inductive Databases, 4th International Workshop, KDID 2005 (Porto, Portugal, October 3, 2005), Revised Selected and Invited Papers. Lecture Notes in Computer Science. Vol. 3933. Springer, 2005. P. 188–201. DOI: 10.1007/11733492_11.

22. Yankov D., Keogh E.J., Rebbapragada U. Disk aware discord discovery: finding unusual time series in terabyte sized datasets. Knowledge and Information Systems. 2008. Vol. 17. P. 241–262. DOI: 10.1109/ICDM.2007.61.

23. Yeh C.-C.M., Zhu Y., Ulanova L., *et al.* Time series joins, motifs, discords and shapelets: a unifying view that exploits the matrix profile. Data Min. Knowl. Discov. 2018. Vol. 32, no. 1. P. 83–123. DOI: 10.1007/s10618-017-0519-9.

24. Zhu Y., Gharghabi S., Silva D.F., *et al.* The Swiss army knife of time series data mining: ten useful things you can do with the matrix profile and ten lines of code. Data Mining and Knowledge Discovery. 2020. Vol. 34. P. 949–979. DOI: 10.1007/s10618-019-00668-6.

25. Zhu Y., Imamura M., Nikovski D., *et al.* Matrix Profile VII: Time Series Chains: A New Primitive for Time Series Data Mining. 2017 IEEE International Conference on Data Mining, ICDM 2017 (New Orleans, LA, USA, November 18–21, 2017). P. 695–704. DOI: 10.1109/ICDM.2017.79.

26. Zhu Y., Zimmerman Z., Senobari N.S., *et al.* Matrix Profile II: Exploiting a Novel Algorithm and GPUs to Break the One Hundred Million Barrier for Time Series Motifs and Joins. IEEE

16th International Conference on Data Mining, ICDM 2016 (Barcelona, Spain, December 12–15, 2016). IEEE Computer Society, 2016. P. 739–748. DOI: 10.1109/ICDM.2016.0085.

27. Zymbler M.L., Kraeva Y., Grents A., *et al.* An Approach to Fuzzy Clustering of Big Data Inside a Parallel Relational DBMS. Data Analytics and Management in Data Intensive Domains – 21st International Conference, DAMDID/RCDL 2019 (Kazan, Russia, October 15–18, 2019). Communications in Computer and Information Science. Vol. 1223. Springer, 2019. P. 211–223. DOI: 10.1007/978-3-030-51913-1_14.