

Реконструкция фазового пространства динамической системы высоконагруженного кэширующего механизма информационных систем

П. А. Курников, Н. В. Крапухина

Национальный исследовательский технологический университет «МИСиС», г. Москва, Россия

Аннотация. В работе предложен подход к определению критических режимов работы компонент объектно-реляционного отображения (ORM), применяемых в информационных системах (ИС). Ключевой особенностью данного подхода является имитационное моделирование проекций ИС различной функциональности. Под проекцией ИС подразумевается сценарий работы выполнения запросов к БД через механизмы кэширования в контексте времени. В работе показано, что поведение размера элементов кэша во времени может быть отождествлено с поведением концентраций веществ химических реакций, протекающих в колебательном режиме. Данные процессы могут быть описаны системой дифференциальных уравнений высоких порядков с нелинейными слагаемыми. Зачастую решить данные системы аналитически или численно не представляется возможным. Предложено рассматривать состояния элементов кэша при работе проекции ИС в дискретные моменты времени и представлять их в многомерные временные ряды. Предложен один из методов реконструкции многомерных временных рядов в фазовое пространство. Использован анализ сечений Пуанкаре и показатели Ляпунова в исследовании фазовых траекторий системы. Показано наличие как диссипативного состояния системы, так и детерминированного хаоса при полной детерминированности запросов в моделируемых ИС.

Ключевые слова: информационная система, производительность, ORM, база данных, кэширование, реконструкция фазового пространства, сечение Пуанкаре, максимальная экспонента Ляпунова, диссипативные структуры, предельный цикл, хаос.

DOI 10.14357/20718632190105

1. Проблематика взаимодействия ORM с базой данных

Производительность информационных систем является одной из основных проблем как на этапе ее проектирования и разработки, так и на этапе эксплуатации. При создании ИС почти всегда используется технология преобразования реляционных данных в объектную модель, которая называется ORM. Данная технология

избавляет программиста от написания большого количества кода, часто однообразного и подверженного ошибкам, тем самым значительно повышая скорость разработки. Кроме того, большинство современных реализаций ORM позволяют программисту при необходимости самому задавать код SQL-запросов, который будет использоваться при тех или иных действиях (сохранение в базу данных, загрузка, поиск и т. д.) с постоянным объектом. ORM-система является

системой с длительным хранением персистентных объектов. Использование ORM-системы уменьшает многократно повторяющийся код манипулирования данными в программе. Достигается это созданием дополнительных слоев абстракции. Тем не менее, это создает дополнительные накладные расходы по использованию ресурсов процессора и памяти. Основными накладными расходами на использование памяти является кэш. Кэш можно разделить на следующие уровни [1].

Кэш первого уровня (First-level cache) – всегда «привязан» к объекту сессии и создается при запросе в базу на получение сущности типа А. Данная сущность кэшируется и при следующем запросе в базу на получение объекта сущности А, при наличии этого объекта в базе, будет просто получен данный объект из кэша при этом запрос на получение сущности даже не будет отправлен в СУБД.

Кэш второго уровня (Second-level cache) – если кэш первого уровня привязан к объекту сессии, то кэш второго уровня привязан к объекту-фабрике сессий (Session Factory object). Что как бы подразумевает, что видимость этого кэша гораздо шире кэша первого уровня. Данный кэш оперирует фабрикой сессий и «живет» в данном контексте, как правило, на стороне сервера. Другими словами, этот кэш общий для всех пользователей. Стоит отметить, что по умолчанию данное кэширование не реализовано или отключено во многих ORM фреймворках. Причем существуют различные стратегии кэширования, определяющие поведение кэша в определенных ситуациях.

Кэш запросов (Query cache) – данный тип кэширования исключен из рассмотрения, так как незначительно влияет на расход оперативной памяти и, как следствие, несущественно снижает быстродействие системы.

На Рис. 1 показано использование уровней кэширования при взаимодействии пользователей с базой данных.

Существующие методы исследования производительности ИС базируются на анализе изменения переменных среды во времени с ростом числа пользователей. Переменными выступают индексы производительности ИС. При этом все существующие исследования – немногочисленны и проводятся в контексте лишь одного сценария работы ИС [2, 3], либо вообще не рассматривают декомпозицию системы до кэширующих механизмов, а лишь дают рекомендательные оценки к повышению производительности ИС средствами оптимизации режима работы сервера [4].

В исследованиях по данной тематике, проводимых автором ранее [5], проводился апостериорный выбор ORM-альтернатив в контексте различных по функциональности ИС. Несмотря на то, что в работе было задействовано большое число метрик производительности, данное исследование не давало ответ о характере состояния ORM-компонента в динамике, а лишь позволяло выбрать производительную ORM. В связи с чем, необходимо продолжить исследования по данной тематике и провести более глубокий и универсальный анализ производительности кэширующих механизмов.

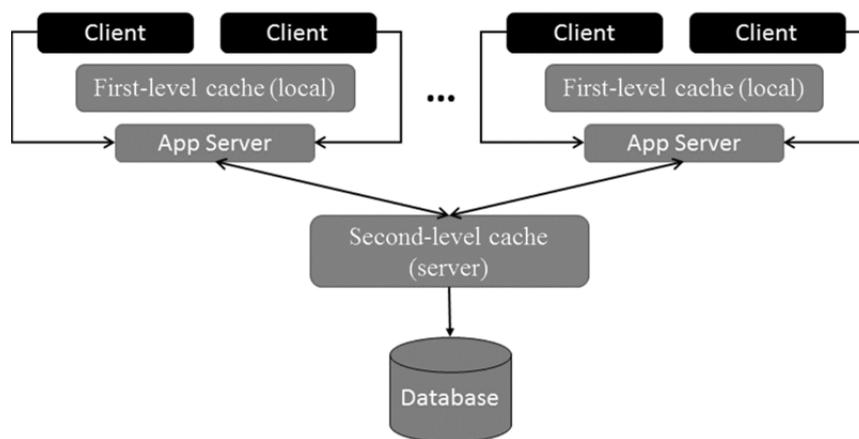


Рис. 1. Архитектура взаимодействия клиентов с БД средствами ORM в информационной системе

В работах [6-8] было установлено, что ORM система является открытой системой и демонстрирует признаки нахождения в неравновесном состоянии в некоторых сценариях работы с ней, а поведение кеширующего механизма можно отождествить с понятием И. Пригожина «диссипативные структуры» [9]. Кроме того, исследования, проведенные в работе [8], показали строгую статистическую зависимость между режимами работы динамической системы и временем обработки SQL запросов. Процесс работы кеширования сущностей базы данных может быть также описан моделями реакций, схожими с реакциями тримолекулярной модели [7]. Данные модели реакций можно представить в виде системы дифференциальных уравнений высоких порядков, однако даже численное решение таких систем уравнений не всегда представляется возможным.

Поскольку можно извлекать состояния кэшей различных типов во время работы модели информационной системы и получать многомерные временные ряды [5], на основе исследования работ [6-8], в данной работе было решено воспользоваться методами реконструкции фазовых пространств, основанными на использовании теоремы Такенса (*Takens theorem*) [10]. Это позволяет наглядно продемонстрировать и облегчить понимание процессов, происходящих в исследуемой динамической системе. Реконструкция аттрактора производится по доступным наблюдениям переменным кеширующей системы, где переменной описывается состояние кэша определенного типа во времени.

Теорема Такенса утверждает, что фазовый портрет динамической системы может быть восстановлен по скалярному временному ряду, где в качестве недостающих координат вектора состояния используется тот же самый ряд, взятый с некоторым запаздыванием. В теореме Такенса ограничения на величину задержки τ не накладываются, однако ее выбор может оказать существенное влияние на значение корреляционной размерности.

Преимущество реконструкции фазового пространства по многомерным временным рядам (*multi-variable phase space reconstruction*) имеет ряд преимуществ, описанных в работах [11-14]. Сама реконструкция имеет множество подходов к вычислению координат фазового

пространства. В работе [15] сравнивают подходы с использованием константных значений временной задержки и размерности вложения для каждой компоненты (*временного ряда*) динамической системы с мультиразмерными их величинами.

Исследования в данной области продолжаются и сегодня, однако универсальный подход к реконструкции мультиразмерного фазового пространства по многомерным временным рядам не был предложен. Опираясь на анализ вышеизложенных подходов, в данной работе предлагается метод реконструкции фазового пространства по многомерным временным рядам, основанный на результатах работы [14]. Как показал анализ исследования [14], данный подход позволяет хорошо реконструировать фазовое пространство по исходным данным, к тому же, обладает относительно небольшой вычислительной сложностью по сравнению с методами, представленными в работах [11, 12].

2. Методология исследования

2.1. Описание дискретно-событийной модели

Для воспроизведения поведения многопользовательской ИС различных типов на определенном ORM источнике, необходимо создать множество тестов различной функциональности к хранимой базе данных. Под тестом понимается запрос к базе данных на извлечение или модификацию нескольких объектов. Причем тест должен быть реализован на стороне сущностной модели и выполняться на стороне сервера через ORM преобразование [16, 17]. Данные тесты, исполняемые на стороне клиентов на временном отрезке $[0, T]$, будут накапливать информацию в общем ресурсе – векторе кэшей ORM [16], образуя мультиразмерное фазовое пространство. После чего данная информация будет анализироваться на предмет характера фазового портрета системы.

Для описания тестов к базе данных будем оперировать терминами реляционной алгебры. Введем понятие отношения R . Отношение представляет собой подмножество декартова произведения одного и более доменов. Отношение может быть представлено в виде табли-

цы, где каждая строка является кортежем, а каждая колонка имеет уникальное имя, называемое атрибутом. [18] Отношение R на множестве атрибутов $A = \{A_1, A_2, \dots, A_n\}$ обозначим $R[A]$. Для описания тестов к базе данных, определим семейство операций, ассоциированных с операциями реляционной алгебры: *projection* (π), *selection* (σ), *union* (\cup), *difference* ($-$), *cartesian product* (\times), *intersection* (\cap), *natural join* (\bowtie), *theta join* ($\bowtie \theta$) и *aggregation* (G) [19].

На примере данных фундаментальных операций реляционной алгебры, рассмотрим спроектированные нами тесты, выполняемые на стороне хранимой реляционной базы на сервере:

1. Селективный тест (σ): выборка из отношения R , обозначается оператором $\sigma F_i(R)$. Результатом данного оператора является подмножество кортежей из R , удовлетворяющее условию, определяемым формулой-предикатом из множества F :

$$\sigma F_i(R) = \{t \mid F_i(t) \wedge t \in R\}, i \in \mathbb{N}, \quad (1)$$

где t – кортеж в R , F_i – формула запроса к базе данных, σ – селективный оператор.

2. Удаление данных ($-$): основная часть выражена селективным запросом. Далее результат запроса удаляется из базы данных:

$$R \leftarrow R - \sigma F(R). \quad (2)$$

3. Добавление данных (\cup): результат добавления данных также выражен селективным запросом. При этом атрибуты значений должны отвечать требованиям домена:

$$R \leftarrow R \cup \sigma F(R). \quad (3)$$

4. Обновление данных (δ): данный тест позволяет вносить изменения значений в кортеж. В терминах реляционной алгебры используется оператор δ :

$$\delta_{A \leftarrow \sigma F(R[A])}, \quad (4)$$

где R – отношение на множестве атрибутов A которому присваиваются значения из выражения $\sigma F(R)$.

На примере проприетарного фреймворка EntityFramework [16, 17], определим компоненту объектно-реляционного отображения ORM. ORM автоматически генерирует концептуальный слой (EDM Data Model) из имеющейся реляционной базы данных. При этом каждая таб-

лица из реляционной модели транслируется в объект концептуального уровня, называемый сущностью. Связи между таблицами также транслируются во множество связей концептуального уровня. При этом ORM отвечает за поддержку так называемой бидиректной связи между EDM моделью (концептуальной моделью) и реляционной моделью. Данная связь позволяет осуществлять запросы и обновления между двумя моделями. Другими словами, ORM создает сущностную схему E из схемы базы данных (store schema R). Под E понимается некое подмножество всех созданных сущностей из исходной базы данных. Под R , в свою очередь, будем подразумевать подмножество хранимых состояний таблиц. Каждому элементу множества R соответствует сущность из E . Связь между объектами множеств E и R определяется бидиректной связью, которая представляется следующими выражениями:

$$E = QV(R), \quad (5)$$

где QV – *query views*, преобразует объекты из реляционной схемы в сущности;

$$R = UV(E), \quad (6)$$

где UV – *update views*, преобразует объекты из сущностной схемы в реляционные.

При этом тесты в терминах реляционной алгебры (формулы 1-4), необходимо переписать на клиентском уровне над схемой сущностей (Табл. 1).

Обозначим группу тестов концептуального уровня множеством F_E . Каждая сущность имеет свое состояние в виде кэша. Под состоянием кэша будем подразумевать занятое дисковое пространство под его хранение. Введем в рассмотрение следующую функцию:

$$\Delta C_i = UpdateObjectCache(\Delta^{+/-} E_i), \Delta^{+/-} E_i \neq \emptyset, \quad (7)$$

где ΔE_i – *ObjectStateEntry*, результат выполнения теста из F_E над сущностью E_i .

Функция *UpdateObjectCache* вносит изменения в типизированный кэш $C\langle E_i \rangle$, ассоциированный с сущностью E , при условии, что ΔE_i является объектом модификации. [16, 17] Каждый тест концептуального уровня, оперирует определенным набором сущностей и модифицирует соответствующий вектор состояний.

Табл. 1. Сравнение реляционных тестов с тестами клиентского уровня

Тест реляционных данных	Тест концептуального уровня	Тип теста
$\sigma F_i(R)$	$\Delta E = \sigma F_i(E)$	Query_Processing
$R \leftarrow R - \sigma F_i(R)$	$\Delta^- E = \{\sigma F_i(E)\}$	Update_Processing
$R \leftarrow R \cup \sigma F_i(R)$	$\Delta^+ E = \{\sigma F_i(E)\}$	Update_Processing
$\delta_{A \leftarrow \sigma F_i(R[A])}$	$\begin{cases} \Delta^- E = \{\sigma F_i(E_{new})\} \\ \Delta^+ E = \{\sigma F_i(E_{old})\} \end{cases}$	Update_Processing

Для тестов группы *Update Processing* происходит следующее преобразование:

$$\Delta R = \Delta UV(E, \Delta^{+/-} E), \Delta^{+/-} E \neq \emptyset, \quad (8)$$

здесь для результата разницы состояний сущностей применяется преобразование *UpdateViews*, которое указывает как извлечь информацию об измененном состоянии хранимой базы данных [18].

Для тестов селективной группы *Update Processing*:

$$\Delta E = QV(R), \quad (9)$$

здесь отображение *QueryViews* генерирует текущее состояние хранимой базы данных и обновляет текущее состояние сущностей. [16]

Перейдем к рассмотрению дискретно событийной системы. Введем в рассмотрение множество пользователей $U = \{u_1, \dots, u_k\}$. Пользователи на клиентском уровне имеют объектное ЕДМ представление базы данных. Серверную часть, хранящую БД, будем подразумевать общим ресурсом описываемой системы. Внутренним состоянием системы является вектор кэшей *C*. Этот ресурс также является общим для каждого пользователя (Рис. 1). Пользователь определяется подмножеством тестов $F_E^{U_i}$ из $\subset F_E$, с которым он работает, как полной группой несовместных событий с вероятностями $\langle P \rangle$. Также законом распределения тестов ω , и временем работы *T* с системой:

$$U = \langle F_E^U, P, \omega, T \rangle \quad (10)$$

Пользователь U_i является источником потока событий (тестов) на $[0, T]$. Зададим многомерную плотность вероятностей интервалов между моментами завершения тестов:

$$\omega_{U_i}(\tau_1, \dots, \tau_n), \tau_k = t_k - t_{k-1}, t_0 = 0, t_n = T, \quad (11)$$

где ω_{U_i} – *n*-мерная функция плотности для пользователя U_i , определяющая закон распределения случайных векторов $\|\tau_k\|, k = \overline{1, n}$, для пользователя U_i .

Многомерная случайная величина моделируется на ЭВМ по методу Неймана, обобщенному на многомерный случай [20]. Выбор теста, который будет выполняться пользователем в момент времени t_k , осуществляется на основе генерации дискретной случайной величины на интервале $[0, 1]$:

$$X = \begin{pmatrix} 1 & \dots & l \\ p_1 & \dots & p_l \end{pmatrix}, \sum_{i=1}^l p_i = 1. \quad (12)$$

Здесь $P = \{p_1, \dots, p_l\}$ – вектор вероятностей полной группы событий (тестов из множества F_E^U) с индексами $\{1, \dots, l\}$.

В момент времени t_k будет выполнен тест с индексом x_m , где $r_k \in \Delta_m$.

$$r \in \begin{cases} \Delta_1, & r \leq p_1 \\ \Delta_k, & p_1 + \dots + p_{k-1} \leq r \leq p_1 + \dots + p_k, (k \geq 2) \end{cases} \quad (13)$$

где Δ_k часть отрезка из $[0, 1]$, длина которого соответствует p_k .

Общее правило, по которому пользователь работает с общим ресурсом *C* можно представить следующим образом:

$$C_j(t_k) = UpdateObjectCache(F_i^E(E_j, t_{k-1}) - F_i^E(E_j, t_k)). \quad (14)$$

Какой именно пользователь осуществил модификацию ресурса – несущественно. Важнее всего отслеживать состояние C на всей временной оси $[0, T]$ с квантом Δt . Для удобства можно просуммировать потоки на одну временную ось.

Теперь проанализируем динамику вектора кэшей C . Для этого будем опрашивать состояние вектора C на $[0, T]$ каждый Δt . В итоге получим фазовое пространство, представленное D -мерными временными рядами X_1, X_2, \dots, X_D где $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,N})$, $i = 1, 2, \dots, D$, $N = T / \Delta t$, $x_{i,n}$ – размер i -го элемента кэша C в момент времени $n\Delta t$. $D = \dim(C)$.

2.2. Реконструкция мультиразмерного фазового пространства

Для систем содержащих D переменных $x_i(t) = [x_{i,1}, x_{i,2}, \dots, x_{i,N}]^T \in \mathbb{R}^D, i = 1, \dots, D$, фазовое пространство можно отобразить следующим образом:

$$X = \begin{bmatrix} x_{1,t}, x_{1,t+\tau_1}, \dots, x_{1,t+(d_1-1)\tau_1}, \\ x_{2,t}, x_{2,t+\tau_2}, \dots, x_{2,t+(d_2-1)\tau_2}, \\ \dots \\ x_{D,t}, x_{D,t+\tau_D}, \dots, x_{D,t+(d_D-1)\tau_D} \end{bmatrix} \in \mathbb{R}^{\sum_{i=1}^D d_i} \quad (15)$$

где τ_i – временная задержка, d_i – размерность вложения для каждого ряда, D – число переменных состояния системы, $t = 1, \dots, M$; $M = N - (d - 1)\tau$.

Любой реконструированный аттрактор представляет собой искаженное изображение исход-

ного. В данной работе предложен метод корреляционных интегралов [21] к нахождению вектора размерностей вложения d_i и вектора временных задержек τ_i . Чем лучше подобраны данные параметры, тем лучше произведена реконструкция фазового пространства и тем меньше искажение аттрактора. Такенс показал [10], что можно восстановить данные свойства аттрактора по временной последовательности одной из составляющих D размерного временного ряда. Производить расчет параметров d и τ будем индивидуально для каждого временного ряда.

2.3. Вычисление временной задержки

При небольшом значении параметра τ значения рядов будут близкими, поэтому именно большое значение приобретает правильный выбор временной задержки. Необходимо стремиться выбрать τ так, чтобы корреляция между x_t и $x_{t+\tau}$ была минимальной. Для этого вычислим автокорреляционную функцию временного ряда:

$$B(\tau) = \frac{1}{m} \sum_{k=0}^{m-1} (x_k - \bar{x})(x_{k+\tau} - \bar{x}), m = N - \tau \quad (16)$$

Задержка τ выбирается равной времени первого пересечения нуля автокорреляционной функции [22, 23] (Рис. 2).

2.4. Вычисление корреляционной размерности аттрактора

По вычисленному параметру τ для временного ряда x в n -мерном фазовом пространстве строится последовательность вида:

$$X_k = (x_k, x_{k+\tau}, \dots, x_{k+(n-1)\tau}), \quad X_k \in \mathbb{R}^n, k = 0, \dots, m-1, m = M - (n-1)\tau \quad (17)$$

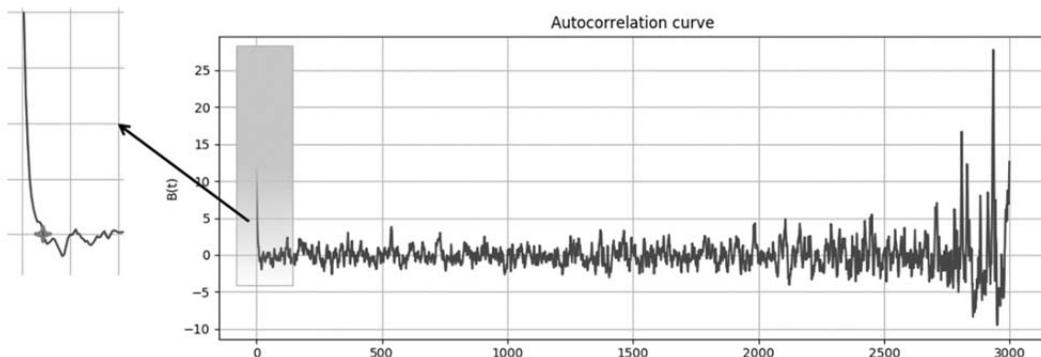


Рис. 2. Пример автокорреляционной динамики (результат получен на реальных данных)

где τ – найденная временная задержка для ряда X_k , n – искомая размерность вложения.

Для оценки размерности аттрактора необходимо произвести оценку ее корреляционной размерности [11, 12, 24].

$$D = \lim_{\varepsilon \rightarrow 0} \frac{\ln(C(\varepsilon))}{\ln(\varepsilon)}, \quad (18)$$

где $C(\varepsilon) = \lim_{m \rightarrow \infty} \frac{1}{m^2} \sum_{k,j=0}^{m-1} \theta(\varepsilon - \|X_k - X_j\|)$ показате-

тель, характеризующий вероятность того, что две точки на фазовой траектории динамической системы будут находиться друг от друга на расстоянии, не превышающем некоторого ε ;

$$\theta = \begin{cases} 1, & (\varepsilon - \|X_k - X_j\|) \geq 0 \\ 0, & (\varepsilon - \|X_k - X_j\|) < 0 \end{cases} \quad \text{– функция Хевисайда;}$$

X_k, X_j – векторы состояния динамической системы.

Для приближенного расчета показателя $C(\varepsilon)$ используется формула:

$$C(\varepsilon) \approx \frac{2}{m(m-1)} \sum_{k=0}^{m-2} \sum_{j=k+1}^{m-1} \theta(\varepsilon - \|X_k - X_j\|). \quad (19)$$

Значения $C(\varepsilon)$ рассчитываются для различных значений ε . Данная процедура повторяется

для нескольких размерностей фазового пространства n . Критерием остановки данной процедуры является следующая оценка:

$$C(\varepsilon) \sim \varepsilon^{D_2} \quad (20)$$

С ростом значения n корреляционная размерность $D_2(n)$ будет возрастать. При некотором n , значение корреляционной размерности выйдет на постоянный уровень $D_2(n) \approx D_2$ (Рис. 3).

Оценкой размерности аттрактора будет являться тангенс угла наклона прямой, аппроксимирующей график корреляционного интеграла $C(\varepsilon)$ в двойном логарифмическом масштабе.

При этом необходимо исследовать только линейные участки графика. Рассмотрим кривую графика зависимости $\ln(C(\varepsilon))$ от $\ln(\varepsilon)$ при некотором значении n . Все расчеты производились на реальных данных (Рис. 4).

При малых значениях ε число соседей у точек аттрактора невелико, и влияние шума нарушает линейность графика. При значениях ε стремящихся к размеру аттрактора линейность также нарушается из-за влияния границ аттрактора, где число соседей у каждой точки меньше (Рис. 4, а). Исследования также показали, что линейная аппроксимация всего графика не дает нужного результата, так как использует лишние окрестности ε при построении аппроксимирующей функции (Рис. 4, б). Таким образом может быть вычислена корреляционная размерность аттрактора и оценка размерности фазового пространства системы.

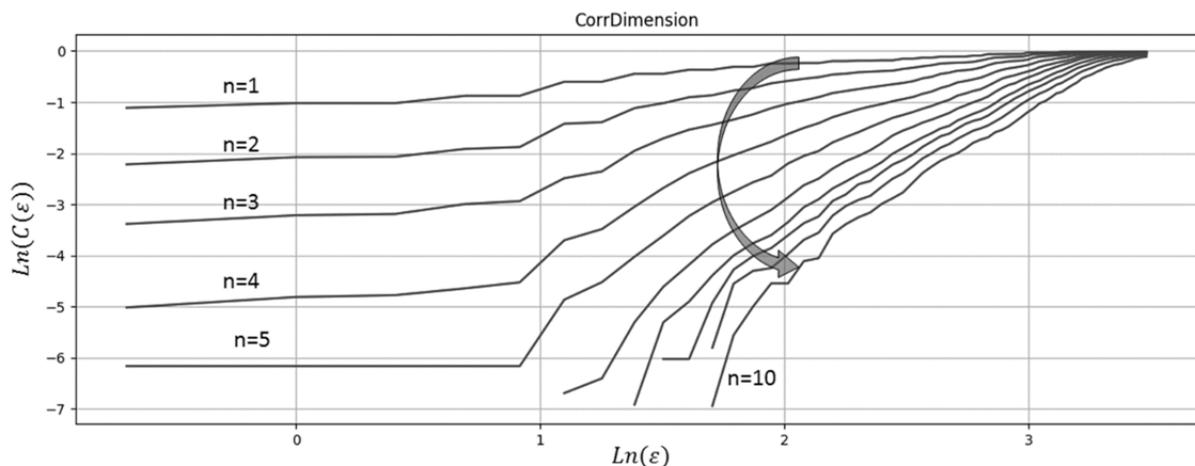


Рис. 3. Логарифмические зависимости показателя C от заданного расстояния ε при различных размерностях фазового пространства

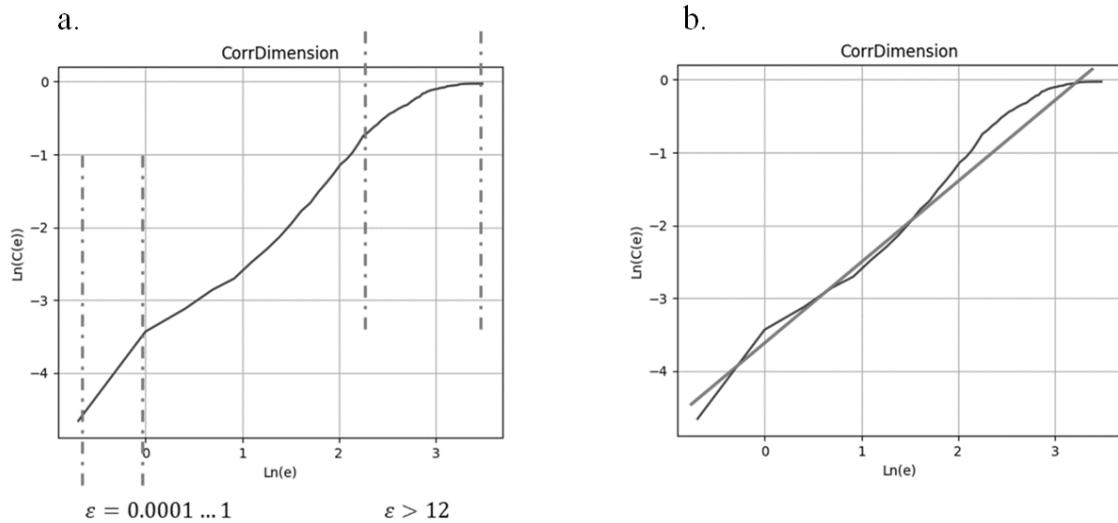


Рис. 4. График корреляционного интеграла в двойном логарифмическом масштабе (а); линейная аппроксимация (б)

2.5. Внедрение реконструированных компонент в единое фазовое пространство

Для внедрения каждой компоненты в единое фазовое пространство необходимо использовать константные параметры τ и d для каждой переменной, определяемые по формулам:

$$\tau = \min(\tau_1, \dots, \tau_D) \quad (21)$$

$$d = \max(d_1, \dots, d_D) \quad (22)$$

Теперь реконструированный аттрактор можно представить следующим образом:

$$X = \begin{bmatrix} x_{1,t}, x_{1,t+\tau}, \dots, x_{1,t+(d-1)\tau}, \\ x_{2,t}, x_{2,t+\tau}, \dots, x_{2,t+(d-1)\tau}, \\ \dots \\ x_{D,t}, x_{D,t+\tau}, \dots, x_{D,t+(d-1)\tau} \end{bmatrix} \in \mathbb{R}^{Dd} \quad (23)$$

На следующем этапе производится адаптивное взвешенное слияние каждого измерения фазового пространства [14]:

$$Z = \begin{bmatrix} z_t, z_{t+\tau}, \dots, z_{t+(d-1)\tau} \end{bmatrix}, \quad (24)$$

$$z_t = \sum_{i=1}^D x_{i,t} w_{i,t}, t = 1, \dots, M; M = N - (d-1)\tau,$$

где $z \in Z$ – координата точки в реконструированном фазовом пространстве, $x_{i,t}$ – t -ое значение переменной i , и $w_{i,t}$ – вес $x_{i,t}$,

$\{z_t, z_{t+\tau}, \dots, z_{t+(d-1)\tau}\}$ – фазовая координата точки t в пространстве Z .

Поскольку природа исследуемой системы нам известна, мы можем подобрать эти веса, исходя из специфики сущностей, запрашиваемых из БД. В данном случае вектором весов w является нормированный вектор размера различных одиночных экземпляров сущностей, занимаемой в оперативной памяти при кэшировании.

2.6. Расчет показателей Ляпунова

Существует спектр показателей Ляпунова, равный по размеру размерности фазового пространства [25, 26]. Данные показатели являются величинами, характеризующими скорость разделения бесконечно близких траекторий фазового пространства. Известно [26], что наибольший из них является максимальным показателем Ляпунова λ . Именно он определяет понятие предсказуемости для исследуемой динамической системы. Данный показатель, например, позволит выявлять хаотичность систем. Аналитическое определение показателей Ляпунова для большинства задач не представляется возможным, поскольку для этого необходимо знать аналитическое решение системы дифференциальных уравнений. Однако существуют достаточно надёжные алгоритмы, позволяющие найти все показатели Ляпунова, используя численные ме-

тоды. В статье использовались численные методы работ [25, 27, 28], где показатель λ рассчитывался по временному ряду. Считая $\delta(0)$ – дистанцией между двумя векторами Такенса в фазовом пространстве, дистанция после времени t между двумя траекториями может быть оценена по следующей формуле:

$$\delta(t) \sim \delta(0) \exp(\lambda t), \tag{25}$$

$$S(t) = \lambda t \sim \log \left(\frac{\delta(t)}{\delta(0)} \right). \tag{26}$$

Линейная регрессия к $S(t)$ и будет оценкой показателя Ляпунова (estimate of the maximal Lyapunov exponent). Вне зависимости от размерности фазового пространства, данная процедура повторяется для нескольких размерностей, чтобы убедиться в том, что экспонента Ляпунова не зависит от размерности пространства.

Результаты

Для апробации предложенных подходов извлечения знаний о поведении состояния кэша ORM источника при высоконагруженном сценарии работы пользователей с различными типами ИС была разработана среда, включающая следующий функционал [5]:

1) создание конфигурационного профиля с некоторым набором и распределением нагрузочных тестов, имитирующих работу различных информационных систем (проекций) на временном отрезке;

2) сериализация состояний типизированного пространства кэшей во время выполнения набора тестов в виде интервальных многомерных временных рядов, пригодных для реконструкции фазового пространства и его анализа.

В качестве эксперимента для апробации метода была выбрана проприетарная ORM система EntityFramework [16, 17]. Разработаны базовые тесты различного функционала с возможностью вариационной компоновки набора тестов, имитирующего работу различных ИС [5]. Под выполнением теста подразумевается отправка запроса на сервер БД с целью получения экземпляра сущности и добавления данной сущности в кэш соответствующего типа. Последовательный запуск тестов во времени T обращается к следующей базе данных (Рис. 5).

Ассоциативная матрица *таблица* ↔ *сущность* выглядит следующим образом (Табл. 2).

Список тестов был разработан с учетом наиболее часто встречающихся операций работы над СУБД с точки зрения ИС.

1. **SearchSimple** - поиск по таблице документов (без ассоциированных с ним адреса и массива, и без файлов) по дате.

2. **GetFile** - получение PDF и XML файлов из таблицы документов.

3. **PutFile** - запись PDF и XML файлов в таблицу документов.

4. **GetStructObject** - получение документа и связанного с ним по ключу адреса из другой таблицы.

5. **PutStructObject** – запись структурированного объекта в несколько таблиц базы: документа, адреса, связанных строк массива документа.

6. **SearchStructObject** - поиск документов по связанной таблице по дате.

7. **RemoveDocument** – удаление документа.

8. **AddDocument** – добавление документа.

В ходе апробации работы было создано несколько проекций информационных систем (Табл. 3).

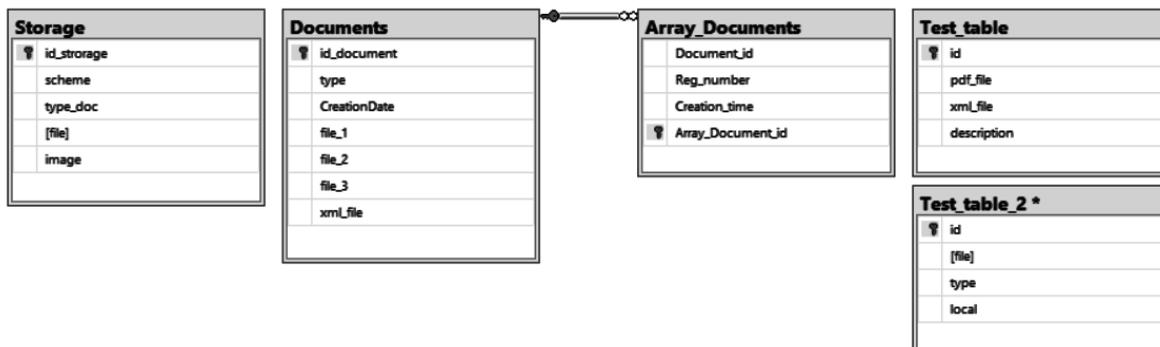


Рис. 5. Диаграмма сущностей базы данных

Табл. 2. Соответствие таблиц с объектом сущности в кэше ORM

Имя таблицы	Тип кэша
Storage	A
Documents	B
Array_documents	C
Test_table_1	D
Test_table_2	E

Табл. 3. Список шаблонов, имитирующих функционал ИС

Тест	Шаблон 1 (распределение тестов по времени их работы)	Шаблон 2	Шаблон 3	Шаблон 4
SearchSimple	30	10	0	40
PutStructObject	0	10	30	5
SearchStructObject	0	10	30	5
GetStructObject	0	50	30	0
GetFile	10	10	5	25
PutFile	15	10	5	25
RemoveDocument	25	0	0	0
AddDocument	20	0	0	0

Табл. 4. Определение числа работающих пользователей с шаблонами при моделировании ИС

Шаблон	Число одновременно работающих пользователей
1	15
2	20
3	50
4	70

Кроме того, набор тестов для каждого шаблона выполняется несколькими пользователями одновременно. Данные правила определены в Табл. 4.

Состояние кэшей различных типов (Табл. 2), сериализуется во временные ряды в равные промежутки времени при работе одного из шаблонов, представленных в Табл. 3.

Для расчетов показателей, рассмотренных в работе, а также реконструкции фазового пространства различных кэширующих схем был задействован язык R с пакетом *nonlinearTseries*. Обоснование численных методов данного пакета рассмотрено в работе [25]. Подобный анализ стохастически возмущенных предельных циклов ранее проводился в работах [29-35]. Данные сведения использовались при оценке результатов, приведенных ниже. Результаты определения реконструированного аттрактора и корреляционной размерности для шаблона 1 представлены на Рис. 6.

Все компоненты (временные ряды) динамической системы сериализовались в равные промежутки времени, а их размерность составила 400000 для каждого ряда. Для шаблона 1 показатель временной задержки составил $\tau = 76$, размерность вложения $d = 9$. Здесь мы видим устойчивый предельный цикл. Это показал и анализ сечения Пуанкаре (Рис. 6, *d*). Несмотря на небольшие стохастические возмущения, вызванные одновременной нагрузкой нескольких пользователей на сервер БД, они не разрушают предельного цикла. Другими словами, система далека от точки стохастической бифуркации.

Для шаблона 2, пучок траекторий выглядит как стохастический. Однако на сечении Пуанкаре прослеживается структура типа цикл. Такой характер поведения системы связан с повышенной нагрузкой кэширующего ORM механизма на оперативную память, так как тесты запускаются параллельно большим числом

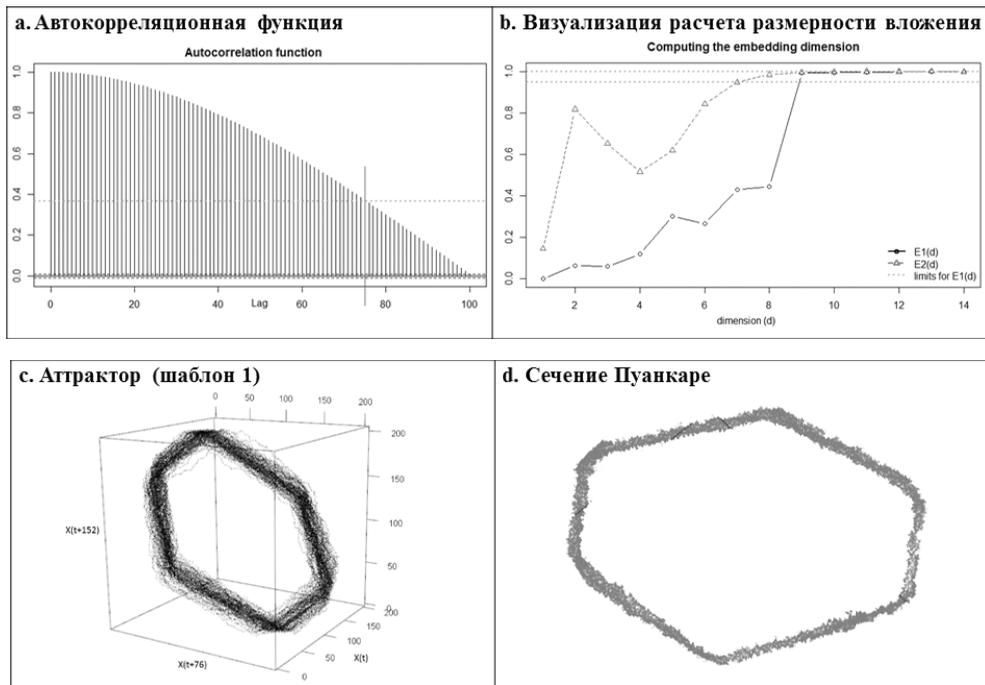


Рис. 6. а – график автокорреляции одной из компонент динамической системы; б – график расчета размерности вложения (пакет **nonlinearTseries**, функция **estimateEmbeddingDim**); с – реконструированный аттрактор в трехмерном пространстве; d – сечение Пуанкаре в двумерном пространстве (пакет **nonlinearTseries**, функция **poincareMap**)

пользователей. Кроме того, в тестах задействована связка сущностей $B - C$. Соответственно при обновлении ORM кэша типа B , кэш C автоматически обновляется по определенному закону. Поскольку данный процесс является внешним воздействием на систему, для сверх-

чувствительных систем данные воздействия могут привести к хаосу. Самая широкая часть пучка проходит в боковых и верхней частях цикла (Рис. 7, d). В нижней части можно наблюдать практически идеальную детерминированную траекторию.

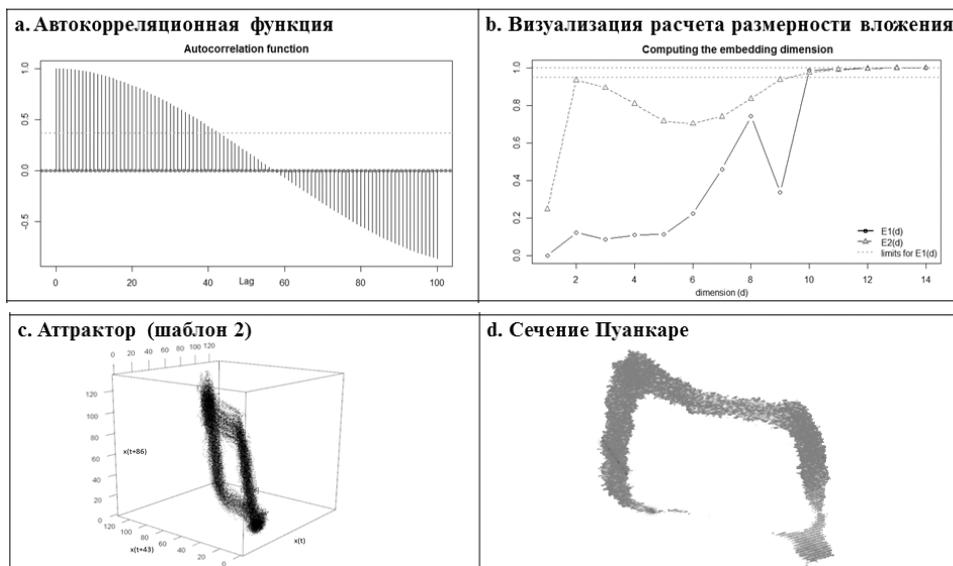


Рис. 7. а – график автокорреляции одной из компонент динамической системы; б – график расчета размерности вложения (пакет **nonlinearTseries**, функция **estimateEmbeddingDim**); с – реконструированный аттрактор в трехмерном пространстве; d – сечение Пуанкаре в двумерном пространстве (пакет **nonlinearTseries**, функция **poincareMap**)

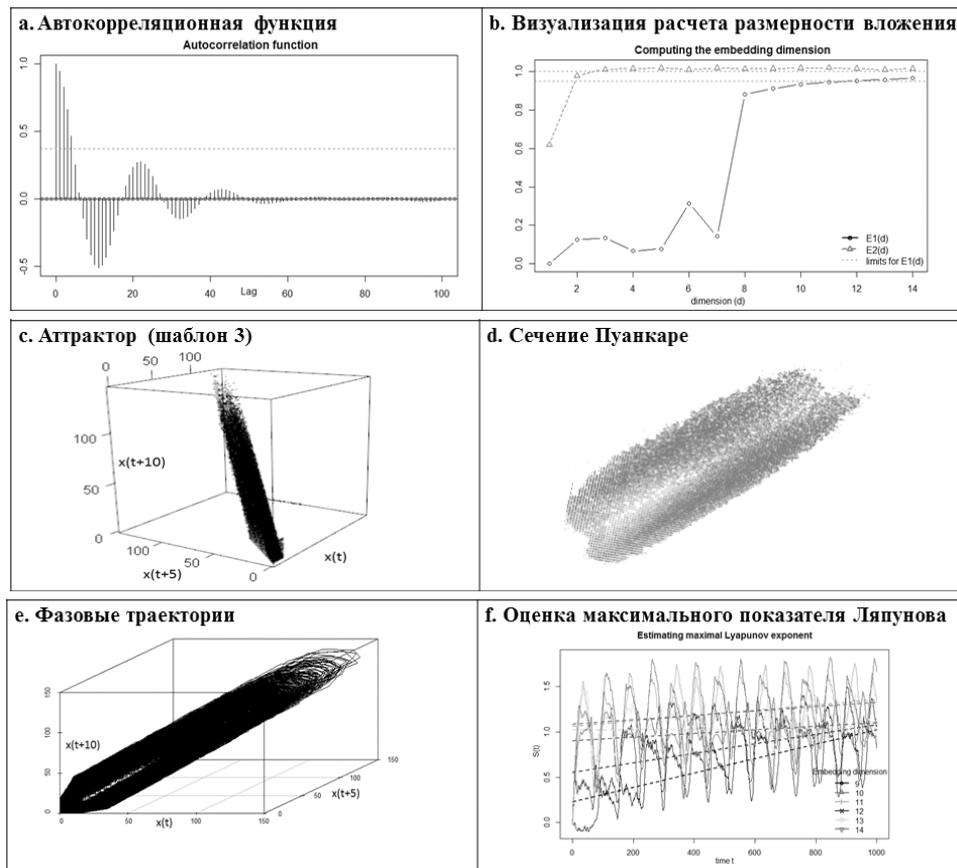


Рис. 8. а – график автокорреляции одной из компонент динамической системы; б – график расчета размерности вложения (пакет **nonlinearTseries**, функция **estimateEmbeddingDim**); с – реконструированный аттрактор в трехмерном пространстве; д – сечение Пуанкаре в двумерном пространстве (пакет **nonlinearTseries**, функция **poincareMap**); е – фазовая траектория; ф – оценка максимального показателя Ляпунова

Результаты реконструкции фазового пространства динамической системы для *шаблона 3* представлены на Рис. 8. Здесь наблюдается переход к хаосу в периодически возмущенной системе. На Рис. 8, *f* представлены оценки показателей (*maximal Lyapunov exponent*) для нескольких размерностей пространства. Несмотря на высокую дисперсию данного показателя, он положителен, что также подтверждает наличие хаотических процессов. И связано это не только с большим числом одновременных запусков модели (50), а с высокой долей тестов `PutStructObject`, `SearchStructObject`, `GetStructObject`. Они составляют 90% от всех остальных тестов, запускаемых в шаблоне. Именно данные процессы генерируют хаос. Это связано, прежде всего, со спецификой кэша ORM.

Для демонстрации данного утверждения, был скомпонован *шаблон 4*. В данном шаблоне количество тестов, связанных с кэшированием

сложно структурированных объектов, существенно снижено. Результаты реконструкции *шаблона 4* представлены на Рис. 9.

Несмотря на большее по сравнению с *шаблоном 3* число одновременно работающих пользователей (70), отсутствие запросов в связанные сущности позволило достичь устойчивого предельного цикла. Следовательно, рост пользователей не является критическим фактором в возникновении хаотических процессов в динамической системе. Кроме того, были произведены замеры времени работы *шаблона 3* и *шаблона 4*. Данные представлены в Табл. 5.

Результаты показали, что несмотря на большую пиковую нагрузку на оперативную память: 200 объектов в *шаблоне 4* и 150 объектов в *шаблоне 3*, время выполнения тестов значительно быстрее именно в *шаблоне 4*. Связано это с устойчивостью предельного цикла и минимальным присутствием стохастических воздействий извне.

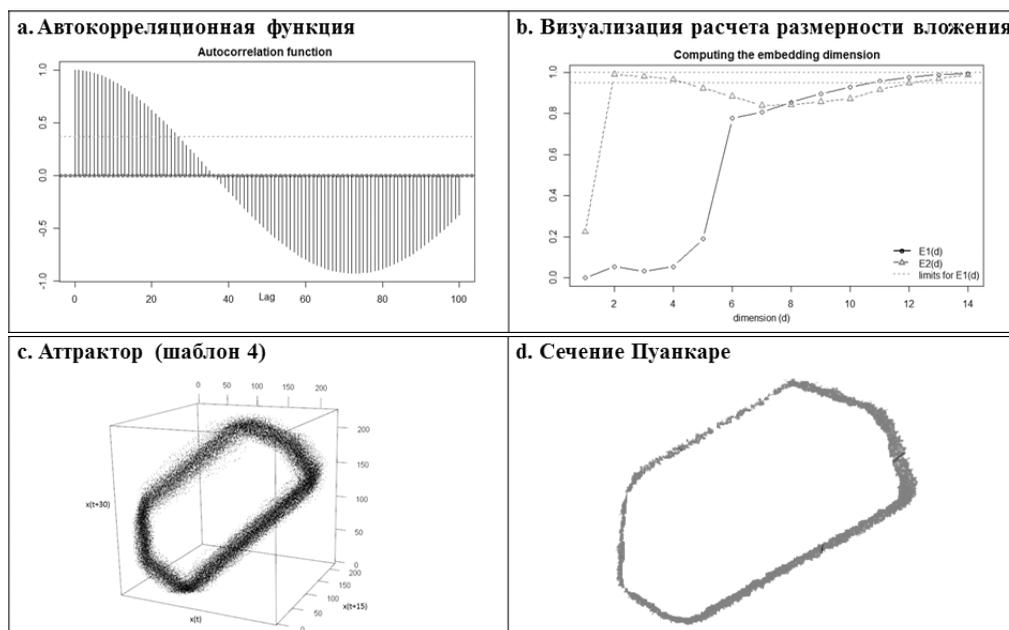


Рис. 9. а – график автокорреляции одной из компонент динамической системы; б – график расчета размерности вложения (пакет **nonlinearTseries**, функция **estimateEmbeddingDim**); с – реконструированный аттрактор в трехмерном пространстве; д – сечение Пуанкаре в двумерном пространстве (пакет **nonlinearTseries**, функция **poincareMap**); е – фазовая траектория

Табл. 5. Среднее время выполнения тестов одним пользователем в миллисекундах

Тест	Шаблон 3	Шаблон 4
SearchSimple	-	1
PutStructObject	192	51
SearchStructObject	125	47
GetStructObject	30	-
GetFile	5	4
PutFile	4	4
RemoveDocument	-	-
AddDocument	-	-

Заключение

В ходе исследования были разработаны и реализованы модели (*шаблоны 1-4*) различных по функциональности ИС. Данные о состояниях каждой переменной (*кэш*) системы во времени были записаны во временные ряды. Для каждого временного ряда были определены необходимые параметры для реконструкции фазового пространства. Исследование данных траекторий позволило выявить характер работы исследуемых динамических систем. Также было показано наличие стационарных и хаотических режимов в исследуемых системах. Оптимальность данных режимов была наглядно

отражена временными показателями (Табл. 4). Кроме того, было показано наличие диссипативных структур в высоконагруженных кэширующих механизмах.

Исследование динамической модели высоконагруженного кэширующего механизма ORM в различных проекциях (*шаблонах*) информационных систем, показало, что в некоторых случаях система имеет устойчивые предельные циклы, *шаблоны 1, 4*. Фазовая траектория стремится к данному предельному циклу, однако устойчивого фокуса не существует (Рис. 6 и Рис. 9). При больших значениях начальных условий динамической системы, траектория будет сходиться к данным циклам.

Фазовый портрет на Рис. 9 говорит о стационарном динамическом режиме исследуемой системы. Данное поведение кэширующего механизма является оптимальным вариантом при работе в ИС *шаблона 4*, в сравнении с ИС *шаблона 3*, поскольку не наблюдается роста оперативной памяти при $T \rightarrow \infty$, что подтверждено результатами, представленными в Табл. 5.

В ходе исследований также было показано, что характер предельного цикла напрямую зависит от производительности всей исследуемой информационной системы. Данное положение было подкреплено непосредственно замерами времени выполнения тестов (Табл. 5). Это объясняется многократным обращением в оперативную память. Кроме того, наличие устойчивого предельного цикла в *шаблоне 1, 4* гарантирует стабильность работы кэширующего ORM механизма и, как следствие, всей информационной системы в целом. В процессе данного исследования было показано, что несмотря на детерминированность правил в архитектуре информационных систем *шаблон 3*, нелинейная динамическая система была подвержена детерминированному хаосу, что привело к более высокой нагрузке на оперативную память и ресурсы центрального процессора (CPU). Данное положение также было подтверждено в ходе сравнения *шаблонов 3 и 4*.

Результаты исследования данной работы применимы для широкого класса программ, использующих механизмы кэширования. Предложенный подход, позволяет производить анализ производительности не только всего подмножества компонент объектно-реляционного отображения, но также и NO SQL систем, что покрывает весь инструментарий обработки и хранения данных в разрабатываемых ИС.

Данный подход эффективен на этапе архитектурного проектирования функционала любой информационной системы, что позволяет производить сравнительный анализ композиций функционала на этапе проектирования ИС и отдавать предпочтение в пользу более устойчивого варианта. Данный подход уже имеет практическую значимость и используется автором при разработке высоконагруженного веб сервиса.

Литература

- Gadkari A. Caching in the distributed environment //Advances in Computer Science: an International Journal. – 2013. – Т. 2. – №. 1. – С. 9-16.
- Bhatti S. N., Abro Z. H., Rufabro F. Performance evaluation of java based object relational mapping tool //Mehran University Research Journal of Engineering and Technology. – 2013. – Т. 32. – №. 2. – С. 159-166.
- Gruca A., Podsiadlo P. Performance Analysis of. NET Based Object-Relational Mapping Frameworks //International Conference: Beyond Databases, Architectures and Structures. – Springer, Cham, 2014. – С. 40-49.
- Jacobson R., Misner S. Microsoft SQL Server (TM) 2005 Analysis Services Step by Step. – Microsoft Press, 2006.
- Крапухина Н. В., Курников П. А., Тарханов И. А. Многокритериальный метод оценки производительности ORM-решений в различных информационных системах //Труды Института системного анализа Российской академии наук. – 2015. – Т. 65. – №. 2. – С. 105-109.
- Мельников С. В. Образование диссипативных структур в ORM-компонентах высоконагруженных порталов //Современные проблемы науки и образования. – 2013. – №. 6. – С. 5.
- Кудж С. А., Мельников С. В. Нестационарный режим работы программной системы при активном «Кэшировании» доступа к базе данных //Перспективы науки и образования. – 2014. – №. 2 (8).
- Мельников С. В. Диссипативные структуры типа «цикл» в ПО, существующем в режиме многопоточного выполнения исполняемого программного кода //Современные проблемы науки и образования (приложение " Технические науки").—2012.—6.—стр. – 2012. – Т. 10.
- Nikolis G., Prigozhin I. Poznanie slozhnogo. Vvedenie [Knowledge of the complex. Intruduction]. Moscow, Mir, 1990.
- Takens F. Detecting strange attractors in turbulence, Dynamical Systems and Turbulence, Warwick 1980 //Lecture notes in mathematics. – 1981. – Т. 898. – С. 366-381.
- Garcia S. P., Almeida J. S. Multivariate phase space reconstruction by nearest neighbor embedding with different time delays //Physical Review E. – 2005. – Т. 72. – №. 2. – С. 027205.
- Hirata Y., Suzuki H., Aihara K. Reconstructing state spaces from multivariate data using variable delays //Physical Review E. – 2006. – Т. 74. – №. 2. – С. 026202.
- Barnard J. P., Aldrich C., Gerber M. Embedding of multi-dimensional time-dependent observations //Physical Review E. – 2001. – Т. 64. – №. 4. – С. 046201.
- Wang R. et al. Data fusion based phase space reconstruction from multi-time series //International Journal of Data-base Theory and Application. – 2015. – Т. 8. – №. 6. – С. 101-110.
- Palit S. K., Mukherjee S., Bhattacharya D. K. A high dimensional delay selection for the reconstruction of proper phase space with cross auto-correlation //Neurocomputing. – 2013. – Т. 113. – С. 49-57.
- Adya A., Bernstein P. A., Melnik S. Generation of query and update views for object relational mapping : pat. 7647298 США. – 2010.

17. Albahari B., Simmons D. G. State transition logic for a persistent object graph : пат. 7526501 США. – 2009.
18. Park H. C. Relational Algebraic Graph Algorithms //International Journal of Computer Science and Network Security (IJCSNS). – 2017. – Т. 17. – №. 1. – С. 25.
19. Silberschatz A. et al. Database system concepts. – New York : McGraw-Hill, 1997. – Т. 4.
20. Быков В. В. Цифровое моделирование в статистической радиотехнике. – Советское радио, 1971.
21. Turcotte D.L. Fractals and chaos in geology and geophysics. Cambridge University Press, Cambridge, 1997.
22. Theiler, J.: Spurious dimension from correlation algorithms applied to limited time-series data. Phys. Rev. A 34, 2427–2432 (1986).
23. Kantz, H., Schreiber, T.: Nonlinear Time Series Analysis. Cambridge University Press, Cambridge (1997).
24. Malinetskii G. G., Potapov A. B. Sovremennye problem nilineinoi dinamiki [Modern Problems of Nonlinear Dynamics]. Moscow, Editorial URSS Publ. 2002. 360 p.
25. Parker, T. S., L. O. Chua, and T. S. Parker (1989). Practical numerical algorithms for chaotic systems. Springer New York.
26. Arnold L., Kliemann W., Oeljeklaus E. Lyapunov exponents of linear stochastic systems //Lyapunov Exponents. – Springer, Berlin, Heidelberg, 1986. – С. 85-125.
27. Розенштейн, Майкл Т и Коллинз, Джеймс Дж и Де Лука, Карло Я. А. Практический метод расчета крупнейших показателей Ляпунова из небольших наборов данных. Physica D: Нелинейные явления, 65-1, 117-134, (1993).
28. Eckmann, J. P., Kamphorst, S. O., Ruelle, D., & Ciliberto, S. (1995). Liapunov exponents from time series. In Turbulence, Strange Attractors And Chaos (pp. 427-435).
29. Anishchenko V. S. et al. Nonlinear dynamics of chaotic and stochastic systems: tutorial and modern developments. – Springer Science & Business Media, 2007.
30. Banerjee M. Turing and non-Turing patterns in two-dimensional prey-predator models //Applications of Chaos and Nonlinear Dynamics in Science and Engineering-Vol. 4. – Springer International Publishing, 2015. – С. 257-280.
31. Bashkirtseva I. A., Isakova M. G., Ryashko L. B. Quasi-potential in stochastic stability analysis of the nonlinear oscillator orbits // J. Neural, Parallel & Scientific Computations. 1999. Vol. 7, N 3. P. 299-310.
32. Bashkirtseva I. A., Ryashko L. B. Sensitivity analysis of the stochastically and periodically forced Brusselator // Phys. A. 2000. Vol. 278. P. 126-139.
33. Bashkirtseva I. A., Ryashko L. B. Sensitivity analysis of stochastically forced Lorenz model cycles under period-doubling bifurcations // Dynamic Systems and Applications. 2002. Vol. 11. P. 293-309.
34. Bashkirtseva I. A., Ryashko L. B. Stochastic sensitivity of 3D-cycles // Mathematics and Computers in Simulation. 2004. Vol. 66. P. 55-67.
35. Bashkirtseva I., Ryashko L. Sensitivity and chaos control for the forced nonlinear oscillations // Chaos, Solitons & Fractals. 2005. Vol. 26. P. 1437-1451.

Курников Павел Александрович. Национальный исследовательский технологический университет «МИСиС», г. Москва, Россия. Аспирант. Количество печатных работ: 3. Область научных интересов: системный анализ, анализ данных, компьютерное зрение. E-mail: two-names@yandex.ru

Крапухина Нина Владимировна. Национальный исследовательский технологический университет «МИСиС», г. Москва, Россия. Профессор, к.т.н. Количество печатных работ: 110. Область научных интересов: системный анализ, принятие решений, математическое моделирование, методы искусственного интеллекта. E-mail: krapuhina@mail.ru

Phase space reconstruction of high-loaded caching mechanism dynamics in information systems

P. A. Kurnikov, N. V. Krapuhina

National University of Science and Technology MISiS, Moscow, Russia

Abstract. This paper proposes an approach to determining the critical operation modes in object-relational mapping (ORM) components applied in information systems. A key feature of the approach is projection of information systems simulation modeling. The projection implies the cases of the execution of queries to databases using caching in the context of time. It was shown that cache behavior can be identified with behavior of reactants concentrations in oscillatory chemical reactions. This process can be described by a system of higher order non-linear differential equations. Often, not possible to solve the systems analytically or numerically. It was proposed that observe cache states at discrete points in time during information system model work and submit data in multidimensional time series. One of the multidimensional phase space reconstruction methods was proposed. Poincare maps and stability analysis with the use maximal Lyapunov exponent was applied. The article illustrates dissipative structures as well as deterministic chaos with the complete determinism of queries in simulated information systems.

Keywords: information system, performance, ORM, database, caching, phase space reconstruction, Poincare map, largest Lyapunov exponent, dissipative system, limit cycle, chaos.

DOI 10.14357/20718632190105

References

- Gadkari, A. (2013). Caching in the distributed environment. [Advances in Computer Science: an International Journal], 2(1):9-16.
- Bhatti, S. N., Abro, Z. H., & Abro, F. R. (2013). Performance evaluation of java based object relational mapping tool. [Mehran University Research Journal of Engineering and Technology], 32(2):159-166.
- Gruca, A., Podsiadlo, P. (2014). Performance Analysis of NET Based Object-Relational Mapping Frameworks [In International Conference: Beyond Databases, Architectures and Structures]. Springer, Cham. 40-49.
- Jacobson, R., & Misner, S. (2006). Microsoft SQL Server (TM) 2005 Analysis Services Step by Step. Microsoft Press.
- Krapuhina N.V, Kurnikov P.A, Tarkhanov I.A. (2015) Multi-criteria method of estimating performance ORM components in information systems [Trudi Instituta sistemnogo analiza Rossiyskoy akademii nauk (ISA RAN)], 65(22):105-109.
- Melnikov S. V. (2013). Formation of dissipative structures in ORM-components of high-loaded portals [Perspectives of Science & Education]. 6(5)
- Kudzh, S. A., & Melnikov, S. V. (2014). Unsteady operation of a software system with the active "caching" to access the database [Perspectives of Science & Education], 8(2).
- Melnikov S. V. (2012). Dissipative structures of the "cycle" in the software, which exists in the mode of multi-threaded execution of the executable code [Perspectives of Science & Education], 10(6).
- Nikolis G., Prigozhin I. (1990). Poznanie slozhnogo. Vvedenie [Knowledge of the complex. Intruduction]. Mir, Moscow.
- Takens F. (1981). "Detecting strange attractors in turbulence". [Dynamical Systems and Turbulence], Lecture Notes in Mathematics, vol. 898. Springer-Verlag. 366-381.
- Garcia, S. P., & Almeida, J. S. (2005). Multivariate phase space reconstruction by nearest neighbor embedding with different time delays [Physical Review E] 72(2), 027205.
- Hirata, Y., Suzuki, H., & Aihara, K. (2006). Reconstructing state spaces from multivariate data using variable delays [Physical Review E], 74(2), 026202.
- Barnard, J. P., Aldrich, C., & Gerber, M. (2001). Embedding of multidimensional time-dependent observations [Physical Review E], 64(4), 046201.
- Wang, R., Gao, J., Gao, Z., Gao, X., Jiang, H., & Cui, L. (2015). Data fusion based phase space reconstruction from multi-time series [Intern J Database Theory]. 8(6):101-110.
- Palit, S. K., Mukherjee, S., & Bhattacharya, D. K. (2013). A high dimensional delay selection for the reconstruction of proper phase space with cross auto-correlation [Neuro-computing]. 113:49-57.
- Adya, A., Bernstein, P. A., & Melnik, S. (2010). U.S. Patent No. 7,647,298. Washington, DC: U.S. Patent and Trademark Office.
- Albahari, B., & Simmons, D. G. (2009). U.S. Patent No. 7,526,501. Washington, DC: U.S. Patent and Trademark Office.
- Park, H. C. (2017). Relational Algebraic Graph Algorithms [International Journal of Computer Science and Network Security (IJCSNS)], 17(1):25.
- Silberschatz, A., Korth, H. F., & Sudarshan, S. 1997. [Database system concepts (Vol. 4).] New York: McGraw-Hill.
- Bykov V. V., (1971). Digital simulation and statistical radio engineering, [Izdat. "Sov. Radio"], Moscow.
- Turcotte D.L. 1997. [Fractals and chaos in geology and geophysics]. Cambridge University Press, Cambridge.
- Theiler, J. (1986). Spurious dimension from correlation algorithms applied to limited time-series data. [Physical review A], 34(3), 2427.
- Kantz, H., & Schreiber, T. (2004). Nonlinear time series analysis (Vol. 7). Cambridge university press.
- Malinetskii, G. G., & Potapov, A. B. (2000). Modern problems of nonlinear dynamics. Moscow: Editorial URSS.
- Parker, T. S., & Chua, L. (2012). Practical numerical algorithms for chaotic systems. Springer Science & Business Media.
- Arnold, L., Kliemann, W., & Oeljeklaus, E. (1986). Lyapunov exponents of linear stochastic systems. In Lyapunov Exponents (pp. 85-125). Springer, Berlin, Heidelberg.
- Rosenstein, M. T., Collins, J. J., & De Luca, C. J. (1993). A practical method for calculating largest Lyapunov exponents from small data sets [Physica D: Nonlinear Phenomena], 65(1-2):117-134.
- Eckmann, J. P., Kamphorst, S. O., Ruelle, D., & Ciliberto, S. (1995). Liapunov exponents from time series. In Turbulence, [Strange Attractors And Chaos] 427-435.
- Anishchenko, V. S., Astakhov, V., Neiman, A., Vadivasova, T., & Schimansky-Geier, L. (2007). Nonlinear dynamics of chaotic and stochastic systems: tutorial and modern developments. [Springer Science & Business Media].
- Banerjee, M. (2015). Turing and non-Turing patterns in two-dimensional prey-predator models [In Applications of Chaos and Nonlinear Dynamics in Science and Engineering-Vol. 4] Springer, Cham. 257-280.
- Bashkirtseva, I. A., Isakova, M. G., & Ryashko, L. B. (1999). Quasipotential in stochastic stability analysis of the nonlinear oscillator orbits [Neural, Parallel & Scientific Computations]. 7(3):299-310.
- Bashkirtseva, I. A., & Ryashko, L. B. (2000). Sensitivity analysis of the stochastically and periodically forced Brusselator [Physica A: Statistical Mechanics and its Applications] 278(1-2): 126-139.
- Bashkirtseva, I. A., & Ryashko, L. B. (2002). Sensitivity analysis of stochastically forced Lorenz model cycles un-

- der period-doubling bifurcations [Dynamic systems and applications] 11(2):293-309.
34. Bashkirtseva, I. A., & Ryashko, L. B. (2004). Stochastic sensitivity of 3D-cycles [Mathematics and Computers in Simulation] 66(1):55-67.
35. Bashkirtseva, I., & Ryashko, L. (2005). Sensitivity and chaos control for the forced nonlinear oscillations [Chaos, Solitons & Fractals] 26(5):1437-1451.

Kurnikov P.A. Postgraduate student, National University of Science and Technology MISiS, 4 Leninsky Prospekt, Moscow, 119049, Russia, e-mail: two-names@yandex.ru

Krapuhina N.V. Professor, National University of Science and Technology MISiS, 4 Leninsky Prospekt, Moscow, 119049, Russia, e-mail: krapuhina@mail.ru