

С.С. ТАНЯНСКИЙ, канд. техн. наук, *В.В. ТУЛУПОВ*, канд. техн. наук,
Д.А. РУДЕНКО, канд. техн. наук (г. Харьков)

ДЕЦЕНТРАЛИЗОВАННОЕ УПРАВЛЕНИЕ ПОТОКАМИ ТРАНЗАКЦИЙ

Функціонування багатьох корпоративних інформаційних систем вимагає забезпечення специфікації і планування міжзадачних залежностей. Крім цього, необхідно управляти транзакціями при зміні структури баз даних. Для рішення цих задач пропонується передати частину функцій управління інформаційній системі. У зв'язку з цим, потрібно забезпечити погодженість між даними, переданими транзакцією, і структурою бази даних. У статті розглядається спосіб уявлення даних, необхідний для функціонування інформаційної системи.

The functioning of many corporate information systems requires maintenance of the specification and planning of intertasks dependences. Besides, it is necessary to operate transactions at change of structure of databases. For the decision of these tasks it is offered to transfer a part of functions of management to information system. In this connection, it is required to supply a coordination between the data transmitted transaction, and structure of a database. In clause the way of performance given, necessary for functioning information system is considered.

Постановка проблеми. Функціонування многих інтегрованих інформаційних систем (ИС), використовуючих бази даних, залежить від характеру управління передачею даних. Прикладом такого функціонування є забезпечення ефективного планування потоків транзакцій для досягнення потрібної мети з певної початкової ситуації. В зв'язку з цим розробляється план дій, представляючий упорядочену сукупність завдань. Пошук плану дій виникає в ІС коли виникає нестандартна ситуація з певною відомою послідовністю дій, що призводять до потрібної мети.

Система управління потоками транзакцій повинна забезпечувати специфікацію і планування міжзадачних залежностей. Як правило, така система складається з планувальника і агентів завдань [1]. Агент контролює виконання завдання оброблюваним об'єктом, а планувальник оброблює потоки, запускаючи на виконання різні завдання, оцінюючи умови, пов'язані з міжзадачними залежностями. Планувальник представляє агенту завдання на виконання або припиняє виконання раніше запущених завдань. В відповідності до специфікації потоку робіт, планувальник забезпечує планування залежностей і є відповідальним за забезпечення того, що всі завдання досягнуть прийнятної стану завершення.

Існує три підходи до побудови систем управління потоками транзакцій:

- централизованный, один планировщик управляет всеми параллельно выполняемыми задачами;
- частично централизованный, один планировщик для каждого потока задач;
- децентрализованный, планировщик отсутствует, при этом агенты задач координируют свое выполнение, взаимодействуя друг с другом для разрешения межзадачных зависимостей.

На практике, в основном, используются централизованный или частично централизованный типы планировщиков. Однако большинство из предлагаемых решений затрагивают только часть проблем и могут быть использованы в ограниченных случаях. Таким образом, разработка методов управления потоками транзакций расширяющих класс решаемых задач является актуальной.

Анализ литературы. Планировщики, основанные на моделях предикатных сетей Петри и логического параллельного языка L.O, разработаны для управления гибкими транзакциями. Планировщик как интерпретатор спецификаций транзакций может использовать язык VPL. Одним из наиболее простых и эффективных методов управления транзакциями является интерпретатор правила “событие – условие – действие” (ЕСА-правило). Модель управления потоком транзакций можно описать как “игру” планировщика против своего окружения, представленного локальными системами баз данных. Подробное обсуждение этих и других методов управления потоками транзакций можно найти в [2, 3, 4].

При решении задачи управления транзакциями основной проблемой является сохранение автономности систем, участвующих в обработке. Поскольку многие системы до интеграции рассчитаны на выполнение автономных операций, они, как правило, не предоставляют информацию необходимую для выполнения параллельных транзакций, поддерживая при этом требуемую семантику. Кроме того, даже если такие возможности есть, может потребоваться полное переписывание существующих алгоритмов и значительное изменения прикладных программ.

В связи с этим, **целью статьи** является описание модели управления потоками транзакций, в которых используются известные структуры, учитываются требования координации наборов задач и семантика функционирования систем, выполняющих эти задачи. Для решения этого вопроса предлагается использовать децентрализованный подход к управлению потоками транзакций, в которых агенты задач сами координируют свои действия.

Управления потоками транзакций. Задачу в потоке транзакций будем определять как единицу работы, которая может быть выполнена обрабатывающей программой. Задача может быть определена независимо от

обрабатываемого объекта или зависеть от его поведения. Для управления потоком задач нет необходимости рассматривать структурные подробности транзакции, так как каждая задача выполняет конкретную операцию над определенной системой, и нет необходимости учитывать свойства этой операции. Таким образом, для корректного выполнения потока транзакций достаточно проанализировать только “внешние” аспекты задач, влияющие на управление всего потока.

В качестве модели децентрализованного управления потоками транзакций будем использовать модель конечных автоматов [5] для анализа зависимостей между задачами и определения достижимости приемлемых состояний.

Определим структуру транзакции как:

- множество состояний задачи;
- множество переходов между состояниями;
- допустимые условия переходов.

Таким образом, абстрактной моделью транзакции может являться автомат состояний, поведение которого определено заданием для ее выполнения или переходом в другое состояние. Переходы между различными состояниями могут вызываться различными событиями планирования.

Рассмотрим детерминированный автомат \mathfrak{Z} заданный каноническими уравнениями [5]

$$\varphi(t+1) = \Phi(\varphi(t), s(t+1)), \quad (1)$$

$$f(t) = F(\varphi(t)). \quad (2)$$

В этих уравнениях переменная t является единицей времени выполнения операции, при этом значение функции $s(t) = 0$ соответствует единичному выигрышу, а значение $s(t) = 1$ – единичному проигрышу автомата \mathfrak{Z} . Выходная переменная $f(t)$ может принимать n значений f_1, f_2, \dots, f_n , называемых действиями автомата. Автомат \mathfrak{Z} произведет k -е действие, если $f(t) = f_k$, где $k = 1, 2, \dots, n$. Функция $\varphi(t)$ может принимать m различных значений $\varphi_1, \varphi_2, \dots, \varphi_m$ называемых состояниями автомата. Автомат \mathfrak{Z} в момент времени t находится в j -ом состоянии, если $\varphi(t) = \varphi_j$. Действие f_k называется соответствующим состоянию φ_j , если $F(\varphi_j) = f_k$. При этом уравнение (1) описывает зависимость действий автомата от его состояний, а уравнение (2) – изменения его состояний в зависимости от функции $s(t)$.

Состояние переходов можно записать в виде матрицы $\|a_{ij}(s)\|$, где $i, j = 1, 2, \dots, m$. Если в момент времени t автомат находится в состоянии φ_i , то в момент времени $t+1$ он перейдет в такое состояние φ_j , для которого $a_{ij}(s(t+1)) = 1$.

Стохастический автомат, как и детерминированный, имеет конечное число состояний $\varphi_1, \varphi_2, \dots, \varphi_m$, конечное число действий f_1, f_2, \dots, f_n и $s(t)$ может принимать два значения 0 и 1. Отличной является матрица переходов, в

которой $a_{ij}(s)$ имеет смысл вероятности перехода i -го состояния в j -е при заданном значении s .

Автомат \mathfrak{Z} находится в стационарной случайной среде $C(a_1, a_2, \dots, a_n)$, если действие f_k , произведенное в момент времени t , влечет в момент времени $t+1$ значение $s = 1$ с вероятностью $p_k = \frac{1-a_k}{2}$ и $s = 0$ с вероятностью

$$q_k = \frac{1+a_k}{2}, \text{ очевидно, что } |a_k| \leq 1.$$

Вероятность p_{ij} перехода автомата из состояния φ_i в состояние φ_j определяется формулой (3).

$$p_{ij} = p_{k_i} a_{ij}(1) + q_{k_i} a_{ij}(0), \quad i, j = 1, 2, \dots, m. \quad (3)$$

Математическое ожидание выигрыша для автомата \mathfrak{Z} в среде C , в обозначении $W(\mathfrak{Z}, C)$ выражается формулой (4).

$$W(\mathfrak{Z}, C) = \sum_{k=1}^n \sigma_k a_k. \quad (4)$$

При этом автомат обладает целесообразным поведением, если выполняется условие (5):

$$W(\mathfrak{Z}, C) > \frac{1}{n}(a_1 + a_2 + \dots + a_n). \quad (5)$$

Переходы состояний для каждого значения s можно изобразить в виде графа состояний, в котором каждому состоянию φ_i ставится в соответствие вершина i , а каждому ненулевому матричному элементу $a_{ij}(s)$ стрелка, направленная из вершины i в вершину j .

Рассмотрим пример использования автомата для транзакции с двумя возможными состояниями. Задача A_1 не может стартовать, пока задача A_2 не завершится, или задача A_2 должна быть прервана, если задача A_1 будет зафиксирована.

Пусть общее поведение автомата \mathfrak{N} определяется двумя состояниями φ_1 и φ_2 и двумя действиями $f_1 = F(\varphi_1)$ и $f_2 = F(\varphi_2)$. При этом матрицы выигрыша и проигрыша имеют вид:

$$\|a_{i,j}(0)\| = \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix}, \quad (6)$$

$$\|a_{i,j}(1)\| = \begin{vmatrix} 0 & 1 \\ 1 & 0 \end{vmatrix}. \quad (7)$$

В данном случае (6) соответствует состоянию выигрыша, а (7) – состоянию проигрыша. Автомат \mathfrak{N} , описывающий данную ситуацию, сохраняет свои состояния при выигрыше и изменяет при проигрыше. Граф переходов состояний автомата \mathfrak{N} изображен на рисунке.

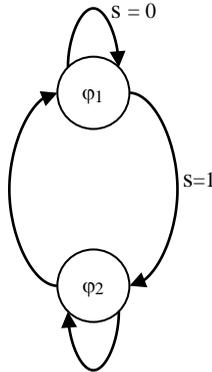


Рис. Граф возможных состояний

Пусть \aleph зависит от некоторой случайной среды $C(a_1, a_2)$, тогда согласно (3), матрица переходных вероятностей будет иметь вид $P = \begin{vmatrix} q_1 & p_1 \\ p_2 & q_2 \end{vmatrix}$. Записав вероятность $a_{ij}(0)$ как r_1 , а вероятность $a_{ij}(1)$ как r_2 и воспользовавшись условием нормировки $r_1 + r_2 = 1$, получим что $r_1 = \frac{p_1}{p_1 + q_2}$, а $r_2 = \frac{p_2}{p_1 + q_2}$, и соответственно математическое ожидание выигрыша будет соответствовать $W(\aleph, C) = \frac{p_1 a_2 + p_2 a_1}{p_1 + p_2}$.

Согласно неравенству (5) можно заметить, что $W(\aleph, C) > \frac{a_1 + a_2}{2}$ при $a_1 \neq a_2$ и, следовательно, автомат \aleph обладает целесообразным поведением в стационарной случайной среде.

Таким образом, транзакции можно моделировать их состояниями вместе с событиями и соответствующими переходам между состояниями. Межадачные зависимости между значимыми событиями формально определяются соответствующим автоматом зависимостей. Такие зависимости могут поддерживаться путем проверки целесообразности поведения этого автомата.

В рассматриваемой постановке задачи предполагается, что каждый поток работ выполняется под управлением своего автомата и обеспечивает локальное управление параллельным выполнением в рамках, требуемых семантикой каждой задачи. Это гарантирует, что каждая отдельная обработка транзакции остается в согласованном состоянии. Если предположить, что

каждый поток транзакций выполняется корректно, то параллельное выполнение нескольких потоков также может быть корректным, если оно эквивалентно последовательному, без взаимного влияния, выполнению этих потоков.

Выводы. В статье рассмотрен подход к управлению потоками транзакций при отсутствии планировщика задач. Показано, что поток транзакций можно рассматривать как целесообразное поведение автомата в стационарной случайной среде.

При отсутствии дополнительной информации об ограничениях, налагаемых на состояния систем, участвующих в выполнении транзакции и о свойствах задач этой транзакции, такой подход может гарантировать согласованность при обработке потока транзакций.

С другой стороны, иерархические структуры расширенных моделей транзакций, часто являются слишком негибкими для приложений обработки. Основной проблемой, возникающей при разработке системы управления потоками транзакций в рамках рассматриваемой модели, является то, что предопределенный набор свойств, обеспечиваемых моделью, может требоваться, а может и не требоваться семантикой транзакций. Например, межсистемные зависимости, могут быть более сложными, чем те, которые поддерживаются данной моделью.

В связи с этим развитая система транзакционных потоков должна поддерживать многозадачную, мультисистемную работу. При этом различные задачи при выполнении должны обладать различным поведением и различными свойствами, обеспечивать координацию выполнения транзакций, включая обмен данными и реакцию на возможные сбои системы.

Таким образом, очевидно, что дальнейшее исследование в этой области необходимо направить на возможность использования коллективного поведения автоматов с многоуровневой организацией модели согласованной работы систем с дополнительными ограничениями на межсистемные связи.

Список литературы: 1. *Рузинкевич М., Цикоцки А.* Определение выполнения потоков транзакций // Системы управления базами данных. – 1995. – № 2. – С. 114 – 129. 2. *Рузинкевич М., Цикоцки А.* Определение выполнения потоков транзакций // Системы управления базами данных. – 1995. – № 4. – С. 84 – 97. 3. *Оззу М.Т., Валдуриз П.* Распределенные и параллельные системы баз данных // Системы управления базами данных. – 1996. – № 4. – С. 8 – 33. 4. *Гарсиа-Моллина Г., Ульман Дж., Уидом Дж.* Системы баз данных. Полный курс: Пер. с англ. – М.: Издательский дом “Вильямс”, 2003. – 1088 с. 5. *Цетлин М.А.* Исследование по теории автоматов и моделирование биологических систем. – М.: Наука, 1969. – 316 с. 6. *Карпов Ю.Г.* Теория автоматов. – СПб.: Питер, 2002. – 224 с. 7. *Goldchlager L., Lister A.* Computer Science. A modern introduction // Prentice Hall, 1988. – 330 p. 8. *Дейкстра Э.* Взаимодействие последовательных процессов // Языки программирования. – М.: Мир, 1972. – С. 9 – 86.

Поступила в редакцию 10.04.04