

УДК 004.65

## **Метод определения порядка сериализации транзакций в системах управления базами данных, использующих протокол строгой двухфазной блокировки**

Аникин Н. А.

### **Аннотация:**

Предлагается метод определения порядка сериализации транзакций в системах управления базами данных (СУБД), использующих протокол строгой двухфазной блокировки. Рассматриваются вопросы применения этого метода при решении задачи управления параллельным доступом (concurrency control) в мобильной системе доступа к данным (МСДД) – распределенной гетерогенной базе данных, клиенты которой работают в мобильной среде.

### **Ключевые слова:**

СУБД, распределенная гетерогенная база данных, мобильная система доступа к данным, управление параллельным доступом, сериализуемость транзакций, порядок сериализации транзакций, строгая двухфазная блокировка.

### **Введение**

Мобильные системы доступа к данным (МСДД) реализуются наложением системы мультибаз данных (распределенной гетерогенной базы данных) на беспроводную мобильную информационную среду [1]. В такой системе базы данных слабо связаны и неоднородны, то есть могут различаться моделями данных, используемыми схемами обработки транзакций и запросов. С одной стороны, МСДД сохраняет ключевое свойство мультибазы данных – автономность входящих в ее состав СУБД (локальных СУБД – ЛСУБД) [2], с другой стороны – обладает рядом особенностей, присущих только мобильной среде.

В работе [3] была подробно описана формальная модель МСДД, а также было показано, что сериализуемость транзакций в каждой ЛСУБД ещё не гарантирует глобальной сериализуемости. В работе [4] была изучена задача управления параллельным доступом [5, с. 599] в МСДД, и было показано, что история транзакций системы мультибаз данных сериализуема тогда и только тогда, когда:

1. истории транзакций во всех ЛСУБД, входящих в систему, сериализуемы;
2. существует порядок выполнения глобальных транзакций, такой, что во всех историях локальных транзакций глобальные субтранзакции сериализуются в соответствии с этим порядком.

В данной работе рассматривается случай, когда ЛСУБД использует протокол двухфазовой блокировки (Two-Phase Locking – 2PL) [6, с. 49, 5, с. 614], и предлагается метод определения порядка сериализации транзакций, не нарушающий локальной автономности этой СУБД.

### **Протокол строгой двухфазной блокировки**

Протокол 2PL довольно прост и формулируется следующим образом:

- прежде чем выполнить операцию с любым элементом данных, транзакция должна приобрести блокировку на этот элемент;
- после снятия любой блокировки транзакция больше не должна приобретать какие-либо дополнительные блокировки.

Блокировка может быть одного из двух типов: блокировка на чтение (приобретается, если транзакция считывает данные, и обозначается как  $rl(x)$ ) или блокировка на запись (приобретается, если транзакция записывает данные, и обозначается как  $wl(x)$ ). Блокировку типа  $o$  для элемента данных  $x$ , приобретенную транзакцией  $T_i$ , будем обозначать как  $ol_i(x)$ . Будем говорить, что две блокировки  $pl_i(x)$  и  $ql_j(y)$  «конфликтуют», если  $x = y$ ,  $i \neq j$  и операции  $p$  и  $q$  конфликтуют, то есть, они получены для одних и тех же данных операциями разных транзакций, причем одна из этих операций или обе – операции записи.

Алгоритм работы планировщика заданий, использующего протокол 2PL, следующий:

1. при поступлении операции  $p_i(x)$  планировщик проверяет, конфликтует ли блокировка  $pl_i(x)$  с какой-либо уже установленной блокировкой  $ql_j(x)$ ; если да, то операция  $p_i(x)$  задерживается, и транзакция  $T_i$  переходит в состояние ожидания получения блокировки; если нет – блокировка  $pl_i(x)$  устанавливается;

2. установленная блокировка  $pl_i(x)$  не может быть снята как минимум до тех пор, пока операция  $p_i(x)$  не будет обработана;
3. как только планировщик снял хотя бы одну блокировку для какой-либо транзакции, он не может больше устанавливать никакие новые блокировки для этой транзакции.

Правило 1 запрещает двум транзакциям одновременно осуществлять доступ к данным, если требуемые блокировки конфликтуют. Правило 2 необходимо для гарантии того, что операции обрабатываются в таком же порядке, в котором поступают. Правило 3 как раз и является правилом «двух фаз», оно необходимо для гарантии того, что все пары конфликтующих операций двух транзакций обрабатывают в одном и том же порядке.

Можно показать, что история транзакций, порождаемая планировщиком заданий, работающим по протоколу 2PL, сериализуема [6, с. 56].

Однако на практике протокол 2PL практически не применяется из-за того, что получаемые с помощью него истории транзакций могут быть не восстановимыми. Вместо него применяются протоколы строгой двухфазной блокировки (S2PL) и сильной строгой двухфазной блокировки (SS2PL):

Strict Two-Phase Locking (S2PL). Данный протокол отличается от протокола 2PL тем, что он требует, чтобы все установленные блокировки на запись  $wl_i(x)$  не снимались не до тех пор, пока не будет обработана операция  $w_i(x)$ , а до окончания всей транзакции.

Strong Strict Two-Phase Locking (SS2PL). Данный протокол отличается от протокола 2PL тем, что он требует, чтобы все установленные блокировки (как на чтение, так и на запись) не снимались до окончания всей транзакции.

### **Метод определения порядка сериализации транзакций**

Докажем следующее утверждение (использованы обозначения из [3]).

Лемма. Пусть локальная СУБД использует протокол сериализации транзакций SS2PL, и  $H$  – история локальных транзакций на множестве транзакций  $\mathbf{T} = \{T_1, \dots, T_n\}$ , полученная в результате использования этого протокола. Тогда  $H$  сериализуема и порядок сериализации транзакций в  $H$  совпадает с порядком их фиксации.

Доказательство. Сначала покажем, что для любых двух зафиксированных транзакций  $T_i \in H, T_j \in H$ , если существуют две конфликтующие операции  $p_i(x) \prec_H q_j(x)$ , то  $c_i \prec_H c_j$ . Это следует напрямую из описания протокола SS2PL: для любых двух конфликтующих операций  $p_i(x) \prec_H q_j(x)$  сначала будет получена блокировка  $pl_i(x)$ , и блокировка  $ql_j(x)$  не

сможет быть получена. Следовательно, операция  $q_j(x)$  (а следовательно, и операция  $c_j$ ) будет задержана до тех пор, пока не будет снята блокировка  $pl_i(x)$ . Так как по протоколу SS2PL все блокировки сохраняются до конца транзакции, это произойдет только после обработки операции  $c_i$ . Таким образом,  $c_i \prec_H c_j$ .

Теперь докажем, что полученная история  $H$  сериализуема. Предположим обратное: допустим,  $H$  не сериализуема. Без ограничения общности можно считать, что существует  $k \geq 2$  пар операций  $\langle op_1(x), op_2(x) \rangle, \langle op_2(x), op_3(x) \rangle, \dots, \langle op_{k-1}(x), op_k(x) \rangle, \langle op_k(x), op_1(x) \rangle$ , таких что  $op_1(x) \prec_H op_2(x), \dots, op_{k-1}(x) \prec_H op_k(x), op_k(x) \prec_H op_1(x)$ . Ранее было показано, что  $\forall p_i(x), \forall q_j(x): p_i(x) \prec_H q_j(x) \Rightarrow c_i \prec_H c_j$ , значит, верно, что  $c_1 \prec_H c_2, \dots, c_{k-1} \prec_H c_k, c_k \prec_H c_1$ . В силу транзитивности отношения  $\prec_H$  получаем, что  $c_1 \prec_H c_1$ . Это противоречит свойству нереклексивности отношения  $\prec_H$  [3], следовательно, предположение неверно, и история  $H$  сериализуема.

Наконец, докажем, что порядок сериализации транзакций в  $H$  совпадает с порядком их фиксации. С одной стороны, для любых двух транзакций  $T_i \in H, T_j \in H$ , если существуют две конфликтующие операции  $p_i(x) \prec_H q_j(x)$ , то транзакция  $T_i$  сериализуется перед транзакцией  $T_j$  в любой истории, эквивалентной последовательной. С другой стороны, опять же, по показанному ранее свойству протокола SS2PL  $\forall p_i(x), \forall q_j(x): p_i(x) \prec_H q_j(x) \Rightarrow c_i \prec_H c_j$ . Следовательно, порядок сериализации транзакций совпадает с порядком их фиксации. Лемма доказана.

Таким образом, для определения порядка сериализации глобальных субтранзакций в локальных СУБД, использующих протокол SS2PL достаточно знать, в каком порядке субтранзакции были зафиксированы в этих СУБД. Этот порядок и будет соответствовать порядку сериализации.

Порядок фиксации транзакций легко получить, выполняя операции фиксации строго последовательно в каждой ЛСУБД. Это ограничение может отрицательно сказаться на производительности, однако, предполагается, что оно скажется несущественно, так как основное время при обработке транзакции тратится на выполнение непосредственно операций чтения/записи данных, а в мобильной среде – также на передачу данных и ожидание действий клиента (в случае непредвиденного разрыва соединения это ожидание может быть на порядок больше, чем время обработки остальных операций). Последовательного же выполнения транзакций целиком не требуется.

На основании информации о порядке сериализации транзакций строится глобальный граф предшествования (Global Serialization Graph – GSG). Во время фиксации очередной глобальной транзакции в GSG добавляется вершина, соответствующая этой транзакции, и проверяется порядок фиксации её субтранзакций [3] в каждой ЛСУБД. Для каждой пары глобальных транзакций, если они имели субтранзакции в одной из ЛСУБД, в GSG добавляется дуга в соответствии с порядком фиксации субтранзакций в этой ЛСУБД. Если после добавления проверяемой транзакции и соответствующих ей дуг GSG содержит цикл, то, очевидно, субтранзакции как минимум двух глобальных транзакций были сериализованы в разном порядке в локальных СУБД, и, следовательно, история глобальных транзакций не является сериализуемой [4]. В этом случае проверяемая глобальная транзакция должна быть отменена, а соответствующая ей вершина – исключена из GSG. В противном случае глобальная транзакция успешно проходит проверку и может быть зафиксирована.

### Выводы

Предложенный метод позволяет легко получить порядок сериализации транзакций в ЛСУБД, использующих протокол строгой двухфазной блокировки, и тем самым обеспечить глобальную сериализуемость транзакций в МСДД, не нарушая локальной автономности входящих в нее СУБД.

Результаты проведенных исследований могут быть использованы во многих отраслях науки, в том числе в авиакосмической промышленности, где требуется построение мультибаз данных, которые бы интегрировали вновь разрабатываемые БД с существующими БД, хранящими данные расчетов конструкций и экспериментального моделирования различных систем.

### Библиографический список:

1. **Аникин, Н. А.** Особенности использования мобильных устройств в системах мульти-БД [Текст] / Николай Александрович Аникин // Высокие технологии, исследования, промышленность: сборник трудов Девятой международной практической конференции «Исследование, разработка и применение высоких технологий в промышленности». – СПб. : Изд-во Политехн. Ун-та, 2010. – Т. 1. – С. 117-124. – ISBN 978-5-7422-2557-7.
2. **Bright, M. W.** A Taxonomy and Current Issues in Multidatabase Systems [Текст] / M. W. Bright, A. R. Hurson, S. H. Pakzad // IEEE Computer. – 1992. – № 25(3). – С. 50-60. – Текст англ.
3. **Храмов, А. А.** Проблемы управления параллельным доступом в мобильных системах доступа к данным и методы их решения [Текст] / Александр Александрович

Храмов, Николай Александрович Аникин // Вестник Московского авиационного института. – М. : Изд-во МАИ, 2010. – Т. 17, № 3. – С. 129-138. – ISSN 0869-6101.

4. **Аникин, Н. А.** Интеграция СУБД, обеспечивающих локальную сериализуемость, и СУБД, использующих Snapshot-изоляция, в рамках системы мульти-БД [Текст] / Николай Александрович Аникин // Сборник научных трудов по материалам международной научно-практической конференции «Современные проблемы и пути их решения в науке, транспорте, производстве и образовании 2010». Том 3. Технические науки. – Одесса : Черноморье, 2010. – С. 28-33. – ISBN 966-555-157-4.

5. **Дейт, К. Д.** Введение в системы баз данных [Текст] / К. Дж. Дейт. – 8-е изд. – М. [и др.] : Вильямс, 2005. – 1314 с. – ISBN 5-8459-0788-8.

6. **Bernstein, P. A.** Concurrency Control and Recovery in Database Systems [Текст] / Philip A. Bernstein, Vassos Hadzilacos, Nathan Goodman. – Addison-Wesley Publishing Company, 1987. – Текст англ. – ISBN 0-201-10715-5.

#### **Сведения об авторах:**

Аникин Николай Александрович , аспирант Национального исследовательского ядерного университета «МИФИ»

Волоколамское ш., д. 6, кв. 259, г. Москва, 125080;

тел.: 8 (903) 775-23-75; e-mail: [nick-anikin@ya.ru](mailto:nick-anikin@ya.ru)