

## Математическая модель параллельной линеаризации дерева логического вывода в распределённых системах управления дедуктивными базами данных

Пушкаренко С.А.

Донецкий национальный технический университет  
S.Pushkarenko@it.stirol.net

### Abstract

*Pushkarenko S.A. Mathematical model of parallel linearization process of logical deduction tree in deductive database distributed control systems. Implementation problems of deductive database distributed control systems are considered. The proposals regarding increasing of effectiveness of linearization process of compound data structures – arbitrary arity tree, are described.*

### Введение

Базы данных (БД) позволяют хранить значительные объёмы информации. Но извлечение и обработка этой информации недостаточно эффективны. Используемый в настоящее время язык запросов SQL не позволяет в полной мере осуществить извлечение всей необходимой информации из БД. Далеко не все запросы могут быть представлены в рамках одного SQL-оператора. При этом SQL-запросы теряют свою декларативность, становятся процедурными, сложными для написания и восприятия. Это повышает вероятность возникновения ошибок, что в свою очередь приводит к неверным результатам запросов. Поэтому возникает необходимость в альтернативных средствах извлечения и обработки информации. Наиболее перспективным является использование в качестве языков запросов к реляционным БД (РБД) языков логического программирования (ЯЛП), таких как Пролог и специально разработанный для работы с БД язык Дейталог. Такие системы называют дедуктивными БД (ДБД) [1].

### Общая постановка проблемы

Наиболее удобной структурой данных для внутреннего представления логической программы является дерево. Дерево логического вывода представляет собой дерево произвольной арности. Каждый узел такого дерева может ссылаться на произвольное заранее неопределённое количество дочерних узлов, т.е. узлов нижнего уровня.

Используемый в ДБД процесс логического вывода является достаточно трудоёмкой задачей. Основной проблемой логического вывода является экспоненциальный

рост пространства поиска при увеличении количества узлов в дереве логического вывода. В настоящее время наиболее эффективным способом повышения производительности ДБД является распараллеливание процесса логического вывода [2].

В многопроцессорных вычислительных системах (МВС) с распределённой памятью используются интерфейсы, основанные на передаче сообщений, такие как MPI, PVM и др. Все эти интерфейсы способны осуществлять передачу линейных структур данных. В литературе, посвящённой использованию интерфейсов передачи сообщений, обсуждаются в основном способы передачи именно таких структур данных [3, 4].

В ЯЛП, используемых в ДБД в качестве языков запросов, все структуры данных являются нелинейными. В связи с этим возникает задача передачи нелинейных структур между вычислительными узлами МВС.

Для решения поставленной задачи необходимо осуществлять линеаризацию, т.е. преобразование нелинейных структур в линейные, с целью дальнейшей передачи их по коммуникационной среде на удалённые узлы МВС. На удалённых узлах системы необходимо выполнить обратную операцию – делинеаризацию, т.е. преобразование принятой линейной структуры в исходную [5, 6, 7].

При большом количестве узлов в дереве процесс линеаризации является трудоёмкой операцией, требующей значительных временных и вычислительных затрат. В связи с этим возникает необходимость в программно-аппаратных средствах повышения эффективности процесса линеаризации.

### Анализ публикаций

Первые публикации по теме ДБД относятся

к концу 70-х - началу 80-х годов прошлого века: [8, 9, 10, 11].

В работе [11] традиционное представление баз данных характеризуется как модельно-теоретическое. Наряду с модельно-теоретическим подходом рассматривается более предпочтительная в некоторых отношениях альтернатива – доказательно-теоретическое представление.

Применение ЯЛП в качестве языков запросов к БД наиболее полно изложено в работах [1, 12, 13, 14].

В работе [1] изложены теоретические основы и технологии построения основных компонентов дедуктивных баз данных. Работа содержит описание языка Дейталога, парадигмы вычисления программ Дейталога, алгоритмы вычисления целей Дейталога и методы оптимизации запросов. Рассмотрена также возможность трансляции программ на Дейталоге в систему уравнений на языке реляционной алгебры.

Работа [14] является введением в исчисление высказываний и исчисление предикатов с точки зрения баз данных. В ней описываются различные языки запросов и модели баз данных. Рассматриваются приложения логического и аппликативного программирования к базам данных.

Использование параллелизма в процессе логического вывода подробно представлено в работах Р. Логанантхараджа [15, 16, 17] а также в работах [18, 19, 20].

Одни из последних исследований по теме параллельного логического вывода опубликованы в работах: [2, 21, 22].

В работе [2] рассмотрены различные методы дедуктивного вывода: на графовых структурах (на графе связей, на графе дизъюнктов), на иерархических структурах, метод аналитических таблиц. Рассмотрены возможности распараллеливания в процессе дедуктивного вывода. Даются различные виды параллелизма при выводе на графовых структурах.

### **Постановка задачи и целей исследований**

С целью повышения эффективности процесса линеаризации предложено процесс линеаризации выполнять параллельно [7]. Для осуществления процесса параллельной линеаризации управляющий узел МВС (узел 0), осуществляющий автоматическое распараллеливание логической программы и распределяющий задания между узлами МВС, должен являться SMP-системой либо иметь многоядерный процессор. Программа, выполняющая распараллеливание, в данном случае будет являться многопоточной. Основной поток - планировщик - работающий на ядре 0, осуществляет распараллеливание логической

программы в зависимости от особенностей этой программы, данного пользователем запроса и имеющегося в наличии количества вычислительных узлов. Планировщик создаёт дополнительные потоки. Количество дополнительных потоков соответствует количеству вычислительных узлов, на которых будет выполняться выбранный планировщиком параллельный алгоритм логического вывода. Для каждого из вычислительных узлов системы отдельный поток выполняет линеаризацию выделенного планировщиком пространства в дереве логического вывода.

Целью данной работы является разработка математической модели процесса параллельной линеаризации дерева логического вывода в распределённых системах управления ДБД, а также формальное описание системы управления ДБД и процесса линеаризации/делеаризации дерева логического вывода.

### **Формальное описание структуры распределённой системы управления ДБД**

Предложенную распределённую систему управления ДБД можно описать двойкой:

$$\langle N, C \rangle \quad (1)$$

где  $N$  – множество вычислительных узлов МВС;

$C$  – коммуникационная среда.

Множество вычислительных узлов МВС включает в себя управляющий узел и удалённые вычислительные узлы:

$$N = \{ N_0, N_1, \dots, N_{n-1} \} \quad (2)$$

где  $N_0$  – управляющий узел МВС (SMP-система);

$\{ N_0, N_1, \dots, N_{n-1} \}$  – удалённые вычислительные узлы;

$n$  – количество вычислительных узлов.

Управляющий вычислительный узел представляет собой множество процессоров или ядер:

$$N_0 = \{ CPU_0, CPU_1, \dots, CPU_{n-1} \} \quad (3)$$

где  $CPU_0$  – управляющий процессор SMP-системы (управляющее ядро);

$\{ CPU_1, CPU_2, \dots, CPU_{n-1} \}$  – дополнительные процессоры (ядра);

$n$  – количество процессоров (ядер) SMP-системы управляющего узла.

Коммуникационная среда описывается двойкой:

$$C = \langle N, E \rangle \quad (4)$$

где E – множество связей,  $E = \{ e_1, e_2, \dots \}$ .  
С помощью C описывается топология МВС.

**Формальное описание процесса параллельной линейаризации на управляющем узле системы**

Процесс параллельной линейаризации на управляющем вычислительном узле можно описать следующим набором:

$$\langle T, Th, G, f, ST, \text{linear}, SO \rangle \quad (5)$$

где T – исходное дерево;  
Th – множество потоков команд;  
G – запрос пользователя;  
f – функция распараллеливания;  
ST – множество поддеревьев;  
linear – функция линейаризации поддерева;  
SO – множество выходных потоков данных.

Исходными данными для функции распараллеливания являются дерево T и введенный пользователем Дейталог-запрос G.

Дерево можно описать двойкой:

$$T = \langle V, E \rangle \quad (6)$$

где V – множество узлов,  $V = \{ v_1, v_2, \dots \}$ ;  
E – множество рёбер,  $E = \{ e_1, e_2, \dots \}$ .

В данной работе исходное дерево логического вывода T рассматривается как множество поддеревьев ST:

$$T = \{ ST_1, ST_2, \dots \}; \quad (7)$$

Множество потоков команд составляет в целом многопоточный процесс:

$$Th = \{ Th_0, Th_1, \dots Th_{n-1} \}; \quad (8)$$

где Th<sub>0</sub> – управляющий поток (планировщик);

{Th<sub>1</sub>, Th<sub>2</sub>, ... Th<sub>n-1</sub>} – вспомогательные потоки, осуществляющие линейаризацию;

n – количество потоков.

Количество потоков n равно количеству вычислительных узлов МВС, использующихся для решения задачи.

На управляющем процессоре (ядре) CPU<sub>0</sub> управляющего узла N<sub>0</sub> выполняется управляющий поток Th<sub>0</sub>, осуществляющий распараллеливание f и задачу заданий (пространств поиска) для i-го узла системы N<sub>i</sub>. Результатом его работы является поддерево ST<sub>i</sub>:

$$ST_i = f_i(T, G) \quad (9)$$

где G – введенный пользователем Дейталог-запрос.

Поддерево ST<sub>i</sub> представляет собой задание

(пространство поиска) для i-го узла МВС.

Множество выходных потоков данных, предназначенных для передачи на удалённые узлы МВС:

$$SO = \{ SO_1, SO_2, \dots SO_{n-1} \} \quad (10)$$

Исходными данными для функции линейаризации являются выделенное планировщиком (функцией распараллеливания f) пространство поиска (поддерево ST<sub>i</sub>):

$$\text{linear}(f_i(T, G)) \quad (11)$$

Результатом линейаризации является выходной поток данных (линейаризованное поддерево) SO<sub>i</sub>:

$$SO_i = \text{linear}(f_i(T, G)) \quad (12)$$

**Формальное описание процесса делинеаризации на удалённых узлах системы**

Процесс делинеаризации на i-ом вычислительном узле можно описать следующим образом:

$$\langle \text{delinear}(SI_i), ST_i \rangle \quad (13)$$

где ST<sub>i</sub> – поддерево, делинеаризованное на i-ом вычислительном узле МВС;

delinear – функция делинеаризации;

SI<sub>i</sub> – входной поток данных для i-го вычислительного узла.

Исходными данными для функции делинеаризации является полученный от управляющего узла N<sub>0</sub> входной поток данных SI<sub>i</sub>, представляющий собой линейаризованное поддерево ST<sub>i</sub>:

$$ST_i = \text{delinear}(SI_i) \quad (14)$$

**Формальное описание процесса параллельной линейаризации для системы в целом**

Процесс параллельной линейаризации для системы в целом можно описать следующим образом:

$$\langle T, Th, G, f_i, ST_i, \text{linear}, SO_i, SI_i, \text{delinear} \rangle \quad (15)$$

где SO<sub>i</sub> = SI<sub>i</sub> – выходной поток данных для N<sub>0</sub>, являющийся входным потоком для i-го узла N<sub>i</sub>.

**Выводы**

В результате проведенных исследований была разработана математическая модель процесса параллельной линейаризации дерева

логического вывода в распределённых системах управления БД.

В данной работе представлено описание математической модели процесса параллельной линеаризации дерева логического вывода, а также дано формальное описание системы управления БД.

В работе [7] представлены результаты экспериментальных исследований по данному вопросу и анализ эффективности использования процесса параллельной линеаризации с использованием двух потоков по сравнению с одним потоком составил 58,6 %.

Дальнейшим направлением исследований является разработка архитектуры специализированного для аппаратной реализации предложенных алгоритмов.

### Литература

1. Чери С, Готлоб Г., Танка Л. Логическое программирование и базы данных: Пер. с англ. - М.: Мир, 1992. - 352 с.
2. Вагин В.Н. и др. Достоверный и правдоподобный вывод в интеллектуальных системах /Под ред. В.Н. Вагина, Д.А. Поспелова. М.: Физматлитгиз, 2004. - 704 с.
3. MPI: A Message-Passing Interface Standart, June 12, 1995.
4. MPI-2: Extensions to the Message-Passing Interface, November 15, 2003.
5. Пушкаренко С.А., Дацун Н.Н. Сравнительная оценка способов реализации распределенного дерева вывода в дедуктивных базах данных. / Научные труды Донецкого национального технического университета. Серия "Информатика, кибернетика и вычислительная техника" (ИКВТ-2008). Выпуск 9. - Донецк: ДонНТУ. - 2008. - 316 с.
6. Пушкаренко С.А., Дацун Н.Н. Проблемы аппаратно-программной реализации дедуктивных баз данных в многопроцессорных системах с распределённой памятью / "Информатика и компьютерные технологии 2007". Материалы третьей научно-технической конференции мо-лодых учёных и студентов. - Донецк: ДонНТУ. - 2007. - 580 с.
7. Пушкаренко С.А., Дацун Н.Н. Параллельная линеаризация дерева логического вывода в распределённых системах управления дедуктивными базами данных / "Информатика и компьютерные технологии 2008". Материалы четвертой научно-технической конференции студентов, аспирантов и молодых учёных. - Донецк: ДонНТУ. - 2008. - 530 с.
8. Gallaire H., Minker J. Logic and Data Bases. - New York, N.Y.: Plenum Publishing Corp., 1978.
9. Minker J. (ed.). Foundations of Deductive Databases and Logic Programming. - San Mateo, Calif.: Morgan Kaufmann, 1988.
10. Dahl V. On Database System Development through Logic//ACM TODS/ -March, 1982. - 7, № 1.
11. Reiter R. Towards a Logical Reconstruction of Relational Database Theory // On Conceptual Modelling: Perspectives from Artificial Intelligence, Databases, and Programming Languages (eds. M.L. Brodie, J. Mylopoulos, J.W. Schmidt). - New York, N.Y.: Springer-Verlag Pub., 1984. - P. 191-233.
12. Логический подход к искусственному интеллекту: От классической логики к логическому программированию: Пер. с франц. / Тейз А., Грибомон П., Луи Ж- и др. - М.: Мир, 1990. - 432 с.
13. Логический подход к искусственному интеллекту: От модальной логики к логике баз данных: Пер. с франц. / Тейз А., Грибомон П., Юлен Г. и др. - М.: Мир, 1998. - 494 с.
14. Грэй П. Логика, алгебра и базы данных: Пер. с англ. Х. И. Килова, Г. Е. Минца; Под ред. Г. В. Орловского, А. О. Слисенко. - М.: Машиностроение, 1989. - 368 с.
15. Loganantharaj R. Theoretical and Implementational Aspects of Parallel Link Resolution in Connection Graphs. PhD thesis, Department of Computer Science, Colorado State University, 1985.
16. Loganantharaj R., Mueller R. A. Parallel Theorem Proving with Connection Graph // 8th Int. Conf. on Autom. Deduc, LNCS 230, 1986. - P. 337-352.
17. Loganantharaj R., Mueller R., Oldenhoef R.R. Connection Graph Refutation: Aspects of AND-parallelism // Tech. Report CS-85-10, Department of Computer Science, Colorado State University, 1985.
18. Cheng P.D., Juang J. Y. A Parallel Resolution Procedure Based on Connection Graph // Proc. of 6th Nat. Conf. on Art. Intell. (AAAI 87), Seattle, Washington, 13-17 July 1987. - P. 13-17.
19. Hornung G., Knapp A., Knapp U. A Parallel Connection Graph Proof Procedure // German Workshop on Artificial Intelligence, LNCS. - Berlin: Springer-Verlag, 1981. - P. 160-167.
20. Компьютеры на СБИС: В 2-х кн. Кн. 2: Пер. с япон. / Мотоока Т., Хорикоси Х., Сакаути М. и др. - М.: Мир, 1988. - 366 с.
21. Аверин А. И., Вагин В. Н., Хамидулов М. К. Методы параллельного вывода на графовых структурах // Седьмая нац. конф. по иск. инт. с межд. участием (КИИ-2000). - М.: Физматлит, 2000. Т. 1. с. 181-189.
22. Аверин А. И., Вагин В.Н. Параллелизм в дедуктивном выводе на графовых структурах // Автоматика и телемеханика. 2001. №10. с.54-64.

Поступила в редколлегию 13.03.2009