

УДК 681.3.06:519.68

УПРАВЛЕНИЕ ИЕРАРХИЧЕСКИМИ СТРУКТУРАМИ ПОСРЕДСТВОМ МЕТОДА ПОСЛЕДОВАТЕЛЬНОГО УГЛУБЛЕНИЯ

Халилов А.И.

Дагестанский научно-исследовательский и технологический институт информатики, Махачкала,
e-mail: volilah-40@rambler.ru

Среди множества проблем создания систем коллективного пользования параллельного действия существенными являются проблемы общего подхода к вопросу, структуризации системы, её реорганизации и оптимизации, управления функционированием и др. В структурно-базовой технологии, которую мы рассматриваем как инструментарий создания таких систем, не последнее место занимают методы решения этих проблем. В качестве одного весьма эффективного метода рассматривается метод последовательного углубления, который позволяет развить известные методы обработки иерархических структур в направлении минимизации избыточных действий, повышения эффективности параллельной обработки информации, параметризации процессов управления и повышения уровня алгоритмической абстракции. В данной статье рассматриваются назначение метода последовательного углубления, обобщённый механизм его реализации, примеры использования и некоторые особенности как самого метода, так и его применения.

Ключевые слова: система коллективного пользования, структурно-базовая технология, иерархическая структура, распараллеливание, метод последовательного углубления

MANAGING HIERARCHICAL STRUCTURES THROUGH THE SERIAL DEEPENING METHOD

Khalilov A.I.

The Daghestan Scientific Research and Technological Institute of Informatics,
Makhachkala, e-mail: volilah-40@rambler.ru

Among the many problems of creation of systems of collective use of the parallel operations of the essential problems are a common approach to the issue, structuring system, its restructuring and optimization, control, etc. In the structurally-underlying technology, which we see as a toolkit to create such systems, most are methods of solving these problems. As a very effective increase is a hierarchical structure that becomes the algorithm for a particular structure (subject area), once defined the values of parameters such as environment analysis, restructuring and optimization of the system, this procedure is performed when fleshed out and maybe, constraint condition. This method allows you to develop the known methods of processing of hierarchical structures to minimize excess action, improving the efficiency of parallel information processing, parameterization of management process and enhance algorithmic abstraction. This article discusses the purpose of the method is a consistent, generalized mechanism for its implementation, usage examples and some features of the method and its application.

Keywords: system of collective use, structurally-underlying technology, hierarchical structure, paralleling, the serial deepening method

Структурно-базовая технология (СБТ) [5] создания систем коллективного пользования (СКП) [6] предполагает блочность (вложенную), модульность, структурированность и структурную однородность модулей системы на всех уровнях иерархии, что обуславливает стандартность и рекурсивность управляющих процессов, простоту и надежность системы. Эта технология в достаточной мере соответствует естественным структурам сложных объектов автоматизации и обуславливает простоту и естественность реализации свойств коммуникабельности, мультипликативности и параметричности систем баз данных (БД) и баз программ (БП). Она предполагает также использование единых принципов организации структур различных классов (типов) данных и концепции БД при формировании и ведении информационного фонда системы.

Такой подход обуславливает определённое обобщение методов анализа и реорганизации иерархических структур и процессов управления ими. Для этого эффективно

может быть использован метод последовательного углубления (МПУ) [4].

Механизм метода последовательного углубления

МПУ рассматривает анализируемую структуру как частично упорядоченное иерархическое множество A , на котором определено некоторое отношение порядка δ из множества Δ допустимых на этом множестве отношений. Анализ A производится на истинность некоторого предиката $p \in P = \{p_i\}$. Предикат p_0 называют управляющим, p_i ($i = 1, 2, \dots, m$) – функциональными. В каждом конкретном случае p_0 выбирает p_i , истинность которого проверяется. В случае истинности p_i выполняется некоторая процедура q_j ($j = 1, 2, \dots, k$) из множества процедур $Q = \{q_j\}$. Процедура q_0 управляющая и обеспечивает выбор q_j соответствующего p_i .

Пусть $A = \{A_{i_1, i_2, \dots, i_n} \mid i_k (k=1, 2, \dots, n)$ и n – натуральные числа}, k – уровень

иерархии, n – число уровней иерархии, i_k – номер элемента на k -м уровне иерархии. Пусть δ есть отношение порядка на множестве A . Назовём множество элементов $k + 1$ -го уровня, составляющих элемент A_{i_1, i_2, \dots, i_k} k -го уровня, составляющим множеством этого элемента и обозначим его через $S(A_{i_1, i_2, \dots, i_k})$. Будем считать, что количество элементов составляющего множества

$S(A_{i_1, i_2, \dots, i_k})$ задаётся функцией $f(i_1, i_2, \dots, i_k)$, а количество элементов на самом верхнем уровне иерархии равно m . Для составного элемента $A_{i_1, i_2, \dots, i_k} f(i_1, i_2, \dots, i_k) > 1$, а для простого, очевидно, значение функции f равно единице. Тогда механизм метода последовательного углубления состоит в выполнении следующих шагов.

Шаг 1. Для $\{A_{i_1} | i_1 = 1, 2, \dots, m; m - \text{натуральное число}\}$ выполнить p .

Шаг 2. Для $i_1 = 1$ с шагом 1 до m выполнить: если $f(i_1) > 1$ для $\{A_{i_1, i_2} | i_2 = 1, 2, \dots, f(i_1)\}$ выполнить P .

Шаг 3. Для $i_1 = 1$ с шагом 1 до m выполнить: если $f(i_1) > 1$, для $i_2 = 1$ с шагом 1 до $f(i_1)$ выполнить: если $f(i_1, i_2) > 1$, для $\{A_{i_1, i_2, i_3} | i_3 = 1, 2, \dots, f(i_1, i_2)\}$ выполнить p .

.....

Шаг n . Для $i_1 = 1$ с шагом 1 до m выполнить: если $f(i_1) > 1$, для $i_2 = 1$ с шагом 1 до $f(i_1)$ выполнить: ... выполнить: если $f(i_1, i_2, \dots, i_{n-1}) > 1$, для $\{A_{i_1, i_2, \dots, i_n} | i_n = 1, 2, \dots, f(i_1, i_2, \dots, i_{n-1})\}$ выполнить p .

Особенности некоторых применений МПУ

Механизм МПУ позволяет естественным образом составлять простые алгоритмы для многих приложений – для любых иерархических структур. Он, по существу, является схемой алгоритма и превращается в последний, как только будут определены A , Δ , P и Q . Далее в качестве иллюстрации приведём несколько примеров.

1. Распараллеливание программ [3].

Применение на всех уровнях (структурное, операторное и операционное) структуры программы единого, простого и естественного механизма МПУ позволяет выделять в программе максимальный параллелизм на данном языковом уровне. Т.е. при условии использования алгоритмов распараллеливания циклов и процедур он позволяет получить полную параллельную форму (ППФ) программы на языке высокого уровня. При этом степень статического параллелизма равна

$$N_{\text{ст}} = \max_i \sum_j n_{ij}^0 + n_{ij}^b,$$

где n_{ij}^0 – степень статического параллелизма j -го модуля (множества составляющих операторов) i -го уровня иерархии; n_{ij}^b – степень статического параллелизма j -го выражения i -го уровня иерархии;

$$n_{ij}^b = \sum_k \max_l N_{kl},$$

где k – количество выражений в i -м операторе j -го уровня; l – номер уровня иерархии

k -го выражения в i -м операторе j -го уровня; N_{kl} – степень параллелизма k -го выражения l -го уровня иерархии.

Заметим, что в эту формулу как частный случай входит также параллелизм (на уровне выражений) операторов последовательных участков программы.

Отметим некоторые особенности как самого МПУ, так и алгоритмов распараллеливания:

– При распараллеливании посредством МПУ сохраняется естественная иерархическая структура исходного множества и отношение порядка между элементами структуры. Вместе с тем формируется другая иерархическая структура, базирующаяся на новом отношении порядка – параллелизме операторов или операций.

– Параллелизм выделяется процедурой q внутри модуля. При этом в каждом модуле выделяется максимальное число минимальных параллельных модулей. Отсюда, естественно, следует максимальный параллелизм, включающий внутримодульный и межмодульный параллелизм.

Возможность выделения максимально-го параллелизма на данном языковом уровне следует также из теоремы Дильворта [2] и соответствующих методов выделения независимых цепочек.

Частично упорядоченное множество иерархической структуры A будем интерпретировать как иерархическую сеть. Обозначим $n_{i_1}, n_{i_2}, \dots, n_{i_j}$ число независимости подмножества $A_{i_1, i_2, \dots, i_j} \subseteq A$, составляющего элемент A_{i_1, i_2, \dots, i_l} l -го уровня. Тогда имеет

место утверждение: минимальное число цепей, покрывающих A , равно

$$N = \sum_{i_k=1}^J \sum_{j=1}^k \max n_{i_1, i_2, \dots, i_j} - M + 1,$$

где J – глубина иерархии; M – количество элементов, для каждого из которых число независимости его составляющего множества отлично от нуля.

Вычитаемое $M - 1$ имеет место, поскольку одна из цепей, покрывающих составляющее множество каждого элемента, является составной частью некоторой цепи верхнего уровня иерархии (за исключением самого верхнего уровня). Очевидно, что приведенное утверждение представляет собой обобщение теоремы Дильворта для иерархических сетей. Задача выделения минимального числа непересекающихся цепей, покрывающих данное частично упорядоченное множество (т.е. числа независимости n_{i_1, i_2, \dots, i_j}), может быть сформулирована и решена как задача линейного программирования транспортного типа.

– МПУ допускает простое рекурсивное описание его алгоритма. Однако рекурсивность и естественная распараллеливаемость требуют реентерабельности реализующей его программы.

– Возможность использования одних и тех же полей памяти при применении процедуры к различным составляющим множествам повышает экономность и эффективность реализующей программы.

– Механизм МПУ допускает естественное распараллеливание собственного алгоритма. При этом степень параллелизма применения процедуры q равна максимальному числу модулей, расположенных на одном и том же уровне иерархии (исходной структуры), исключается многократное повторение без необходимости значительного числа действий.

– Повышение производительности системы от распараллеливания пропорционально величине $1/(F(1 - F)/N)$, где F – последовательная часть программы, N – число процессоров.

2. *Параллельный синтаксический контроль.*

Назовём модулем оператор программы. Операторы, входящие в данный оператор, представляют собой его составляющее множество. Множество модулей данной программы составляет A .

Под отношением δ будем понимать отношение вложенности модулей.

Процедура p выделяет в пределах данного составляющего множества данного уровня множество не вложенных друг

в друга модулей, снабжает их соответствующими описаниями (дублирует необходимые для данного блока описания объемлющих блоков) и осуществляет синтаксический контроль модулей данного составляющего множества.

Полученный в результате такой интерпретации алгоритм позволит выделять на каждом уровне минимальные самостоятельно контролируемые модули. Поскольку пооператорный контроль связан с большими затруднениями (в частности, с чрезмерным дублированием описаний), может оказаться целесообразным определить модуль как блок программы.

Заметим также, что использование метода последовательного углубления при синтаксическом контроле программы позволит существенно сократить время синтаксической отладки программы. Такое же замечание можно было бы сделать и относительно общей отладки программы.

3. *Анализ, реорганизация и оптимизация систем организационного управления.*

Структура организационного управления (напр., регионом, муниципальным образованием, отраслью, отдельным предприятием или организацией и т.д.) может быть представлена как иерархическая система, на разных уровнях которой требуется разный уровень агрегирования информации. Очевидно, используя МПУ, может быть составлена обобщённая схема управления, из которой посредством задания на входе (в процессе настройки системы) значений параметров (типа организационной структуры, условий анализа, реорганизации или оптимизации и процедур их исполнения и др.) в каждом конкретном случае может быть получен соответствующий алгоритм.

Выводы

1. Предикаты P и процедуры Q могут быть сколь угодно сложными. Они могут устанавливаться для элементов A , не удовлетворяющих δ , новые типы отношений и строить иерархии, удовлетворяющие этим новым отношениям порядка.

В приведенном выше первом примере устанавливается отношение параллельности между элементами, что позволяет в пределах каждого модуля, а следовательно, и в пределах всего множества установить иерархию в отношении параллельности.

Процедура Q может включать в себя также элементы оптимизации. Однако ни оптимизация, ни установка новых отношений и соответствующих им иерархий не должны нарушать исходного отношения порядка между элементами и соответству-

ющей этому отношению иерархической структуры.

Заметим, что иногда возможно нарушение между некоторыми элементами исходного отношения порядка без нарушения иерархической структуры системы. Например, распараллеливание частично независимых операторов с введением соответствующих средств синхронизации.

В третьем примере условия, отношения и процедуры могут быть чрезвычайно многообразными.

2. Если желательно укрупнение элементов, между которыми устанавливается новое отношение, можно ввести некоторый коэффициент, ограничивающий углубление. Например, в первом примере это может быть некоторый вес в стандартных алгоритмических единицах сложности [1], во втором примере – количество символов (длина конструкции в символах), такое, чтобы каждая формируемая минимальная конструкция, удовлетворяющая новому отношению порядка, имела минимальное из возможных отклонение от этого коэффициента.

3. Кроме такой перспективной области, как параллельное программирование, МПУ может быть использован и в других разделах мультиобработки информации. Он может быть также весьма полезным при организации работы вычислительных сетей и комплексов, сетей вычислительных центров, в области сетевого планирования и управления и т.д. При этом, вероятно, потребуются воспользоваться понятиями мультибаз данных, иерархических сетей и др. и механизмами их реализации.

4. Несмотря на сходство с методом обхода деревьев, МПУ существенно отличаются от него возможности ограничения углубления посредством весовых функций и параллельного выполнения процедур Q .

Список литературы

1. Глушков В.М. Два универсальных критерия эффективности вычислительных машин // ДАН УССР. – К., 1960. – № 4.
2. Форд Л.Р., Фалкерсон Д.Р. Потoki в сетях. – М.: Мир, 1966. – 272 с.
3. Халилов А.И. К вопросу о распараллеливании программ / Проблемы кибернетики. – М.: Наука, 1974. – № 28. – С. 157–176.
4. Халилов А.И. Метод последовательного углубления и некоторые его приложения // Теория и практика системного программирования: сб. – Киев: РИО ИК АН УССР, 1976. – С. 180–191.
5. Халилов А.И. Структурно-базовая технология создания систем коллективного пользования: монография. – 2-е. изд. – Махачкала: ИД «МавраевЪ», 2011. – 133 с.
6. Шон Б. Еом. Компьютерные средства коллективной работы в сети: Информационные технологии в бизнесе. Бизнес-класс; под ред. М. Желены. – СПб: Питер, 2002.

References

1. Glushkov V.M. Dva universalnikh kriteriya effektivnosti vychislitelnykh mashin // DAN USSR. K. 1960. no. 4.
2. Ford L.R., Falkerson D.R. Potoki v setyakh. M: Mir. 1966. 272 p.
3. Khalilov A.I. K voprosu o rasparallelevanii programm / Problemi kibernetiki. M: Nauka. 1974. no. 28. pp. 157–176.
4. Khalilov A.I. Metod posledovatel'nogo uglublenniya i nekotorie ego prilozheniya / Sb. «Teoriya i praktika systemnogo programmirovaniya». K: RIO IK AN USSR. 1976. pp. 180–191.
5. Khalilov A.I. Strukturno-bazovaya tekhnologiya sozdaniya system kollektivnogo polzovaniya: Monografiya: izd. 2-e. Makhachkala: ID «Mavraev»/ 2011. 133 p.
6. Shon B. Eom. Kompyuternie sredstva kollektivnoy raboty v seti: Informatsionnie tekhnologii v biznese. Biznes-klass; [Pod red. M. Zheleny]. SPb: Piter. 2002.

Рецензенты:

Адамдзиев К.Р., д.т.н., профессор, зав. кафедрой информационных технологий и моделирования экономических процессов Дагестанского государственного университета, г. Махачкала;

Курбанмагомедов К.Д., д.т.н., профессор, директор Дагестанского института (филиала) Московского государственного открытого университета, г. Махачкала.

Работа поступила в редакцию 25.12.2013.