

Норкин О.Р. Модульная сборка параллельных программ с использованием знаний предметной области. // Проблемы информатики в образовании, управлении, экономике и технике: Сб. статей IX Междунар. научно-техн. конф. – Пенза: ПДЗ, 2009. – С. 83-85.

МОДУЛЬНАЯ СБОРКА ПАРАЛЛЕЛЬНЫХ ПРОГРАММ С ИСПОЛЬЗОВАНИЕМ ЗНАНИЙ ПРЕДМЕТНОЙ ОБЛАСТИ

О.Р. Норкин

Технологический институт
Южного федерального университета,
г. Таганрог, Россия

Рассматривается проблема параллельного программирования для решения сложных задач. Предлагается решение задачи путем сборки параллельных программ из параллельно исполняемых модулей, имеющихся в базе данных.

Norkin O.R. Modular integration of parallel programs with the use of knowledge domain.

The problem of parallel programming for solving complex tasks is under the discussion. It is suggested that this problem should be solved by integration of parallel programs from concurrent run-modules in the database.

Обычно под термином предметная область понимают информационную сторону функционирования какой-либо системы. Множество объектов и связей между ними описывают предметную область. Каждый объект определяется набором определённых свойств, значения которых более или менее полно могут быть указаны.

Предлагаемый подход состоит в возможности получения параллельных программ с помощью базы данных модулей (БДМ), объединённой с базой знаний даже для случая, когда вся задача полностью отсутствует в БДМ, но присутствуют её составные части или подзадачи.

В общем случае задачу некоторой предметной области можно представить как упорядоченный список подзадач:

$$P = \langle p_1, p_2, \dots, p_k \rangle. \quad (1)$$

Если в БДМ имеется k модулей подзадач, то с их помощью можно составить или в каком-то смысле «синтезировать» $N(k)$ задач. Для базового набора модулей подзадач, состоящего из одного элемента p_1 , можно составить $V(p_1)$ вариантов реализации задач. Здесь аргументом функции V является множество модулей подзадач, непосредственно решаемых на вычислительной системе (ВС); значение V определяется архитектурными особенностями ВС (число каналов обменов, типы коммутации, число процессоров и т.д.). В общем случае

$$\begin{aligned} N(k) = & \sum_{i=1}^k V(p_i) + \sum_{i=1}^k \sum_{j=i+1}^k V(p_i, p_j) + \sum_{i=1}^k \sum_{j=i+1}^k \sum_{l=j+1}^k V(p_i, p_j, p_l) + \dots \\ & \dots + \sum_{i_1=1}^k \sum_{i_2=i_1+1}^k \dots \sum_{i_{k-1}=i_{k-2}+1}^k V(p_{i_1}, p_{i_2}, \dots, p_{i_{k-1}}) + V(p_1, p_2, \dots, p_k). \end{aligned} \quad (2)$$

Любая задача может быть сформулирована в терминах своей предметной области. При этом предметная область предполагает вполне определённый математический аппарат, который, в свою очередь, включает в себя определённые формы параллелизма. Согласно гипотезе дуальности программной и аппаратной реализации вычислений [1] математическую запись задачи с использованием операций базового набора можно представить в виде информационного графа. Узлами графа являются операции базового набора модулей подзадач проблемно-ориентированной области, а дугами – потоки данных. Число дуг, входящих в один узел, равно числу операндов операции.

Исследования в данной области сводятся, в основном, к решению двух задач: прямой и обратной. Прямая задача состоит в поиске для заданной предметной области адекватной архитектуры многопроцессорной ВС (МВС), а обратная задача, наоборот, – в поиске для заданной архитектуры МВС адекватной предметной области.

Каждой задаче $j \in J$, решаемой с помощью МВС, соответствует множество способов её программных реализаций

$$P_j = (p_{1j}, p_{2j}, \dots, p_{mj}). \quad (3)$$

Важно выполнить параллельную программу с заданными технико-временными параметрами. После определения множества задач J и множеств их программных реализаций p_1, p_2, \dots, p_n (где n – число задач, мощность множества J) находятся потенциально возможные варианты архитектур МВС $A_q (a_1, a_2, \dots, a_n)$. Для каждого варианта архитектуры a_i необходимо выяснить степень его соответствия каждому элементу множества P_j . Для каждой задачи $j_r \in J$ определяется набор архитектур МВС A_j , соответствующих P_j способам решения задачи.

Все запросы на поиск данных будут связаны с конкретной базой данных – БДМ. Определять нужные отношения между модулями декларативно и априорно нецелесообразно. Кроме того, пользователь при этом может просто ошибиться. Поэтому для решения проблемы необходимо использовать расширяемую базу знаний решения параллельных программ [2, 3, 4].

Необходимость интеллектуальной поддержки процесса составления параллельных программ очевидна из-за многообразия и сложности имеющихся и возникающих заново связей и отношений между аппаратно-программными элементами распределенных вычислений. Как и в большинстве систем, основанных на знаниях, важная роль в исследуемой области отводится эвристическим знаниям, которые доказали свою эффективность в областях, где невозможно точно поставить задачу, сформулировать правила её решения или решить поставленную задачу чисто формальными методами.

На представленных в статье подходах базируются разработки элементов автоматизированной системы составления параллельных программ с использованием структурированных знаний о задачах (разбиение на модули-подзадачи) и объектах проектирования (МВС).

Библиографический список

1. Ясинявичус Р. Параллельные пространственно-временные вычислительные структуры. – Вильнюс : Мокслас, 1988. – 183 с.

2. Норкин О.Р. Интеллектуализация разработки программного обеспечения многопроцессорных вычислительных систем. Рук. деп. в ВИНТИ. 09.1323–В2004.127. –16 с.

3. Норкин О.Р., Шевелев В.В. Представление знаний в системе разработки ПО МВС // Труды конференции «Новые информ. технологии. Разработка и аспекты применения» / ТРТУ. – Таганрог, 2004. – С. 217 – 219.

4. Норкин О.Р. Формализация описания аппаратных компонентов МВС, используемая для составления параллельных программ на основе знаний проблемной области // Труды конференции «Новые информ. технологии. Разработка и аспекты применения» / ТРТУ. – Таганрог, 2004. – С. 214 – 216.