

ОПТИМИЗАЦИЯ АЛГОРИТМА ВЫВОДА АБСТРАКТНОГО СИНТАКСИЧЕСКОГО ДЕРЕВА ПРОГРАММЫ ИЗ ОБРАТНОЙ ПОЛЬСКОЙ ЗАПИСИ

Лукашенко Владислав Владиславович

научный сотрудник кафедры ИВТ и МПИ Рязанский государственный университет имени С.А. Есенина, РФ, г. Рязань

Романчук Виталий Александрович

канд. техн. наук, доцент кафедры ИВТ и МПИ Рязанский государственный университет имени С.А. Есенина, РФ, г. Рязань

Optimization of the output algorithm of the abstract syntax tree of the program from the reverse Polish record

Lukashenko Vladislav

Scientific employee of the department of ICT and MPI Ryazan State University named after S.A. Yesenin, Russia, Ryazan

Vitaly Romanchuk

Candidate of Technical Sciences, Associate Professor of the Department of ICT and MPI, Ryazan State University named after S.A. Yesenin, Russia, Ryazan

Аннотация. В работе рассматривается вариант реализации вычислительного кластера нейрокомпьютеров. Для реализации основного принципа распределенных вычислений в статье представлен алгоритм разбиения задач, поступающих на выполнение в вычислительный кластер нейрокомпьютеров, на подзадачи. Задача, которой посвящена статья – оптимизация алгоритма преобразования обратной польской записи программы в абстрактное синтаксическое дерево. Для этого в статье предлагается представить поступившую на выполнение в кластер программу в модифицированной постфиксной Польской записи и хранить ее в стеке команд программы. На следующем шаге предлагается получить абстрактное синтаксическое дерево программы, следуя оптимизованному алгоритму перевода модифицированной постфиксной Польской записи из стека команд в абстрактное синтаксическое дерево. Работа выполнена при финансировании государственного задания, проект №2.9519.2017/8.9.

Abstract. The implementation of the computational cluster of neurocomputers is considered in the article. To implement the basic principle of distributed computing, the article presents an algorithm for dividing the tasks included in the computing cluster of neurocomputers into sub-tasks. The task, which is devoted to the article, is the optimization of the algorithm for converting the inverse Polish program into an abstract syntax tree. For this, the article proposes to present the program entered into the cluster for execution in the modified post-entry Polish record, and store it in the stack of

program commands. The next step is to get the abstract syntax tree of the program, following the optimized algorithm to translate the modified Polish postfix record from the command stack to an abstract syntax tree. The work was carried out with the financing of the state task, project N 2.959.2017/8.9.

Ключевые слова: распределенные вычисления; кластерные вычисления; кластеризация вычислительных ресурсов; нейровычисления; нейрокомпьютеры; кластеризация нейрокомпьютеров; оптимизация; оптимизация алгоритмов.

Keywords: distributed computing; cluster computing; clustering of computing resources; neuro computing; neurocomputers; clustering of neurocomputers; optimization; algorithm optimization.

Введение

В современной научной и производственной сфере крайне актуальна задача использования вычислительных систем параллельной обработки информации ввиду недостаточности вычислительных ресурсов. Сегодня обозначенные задачи решают с недостаточно высокой степенью эффективности масштабируемые параллельные вычислительные кластеры на базе нейрокомпьютеров [1]. В работе рассматривается существующая реализация параллельного кластера на базе нейрокомпьютеров MB77.07 ЗАО «НТЦ Модуль», технические характеристики и перспективы использования в работе которых, описаны в работе [2-4].

При выборе нетипичной вычислительной архитектуры требуется решить задачу построения модели вычислений. Модель вычислений связывает между собой понятия архитектуры вычислительной системы и модель программирования. Работа алгоритма разбиения задач на подзадачи позволяет организовать один из основных принципов параллельных вычислений, а именно - параллелизм. В основной части алгоритма решается задача перевода из обратной польской записи в абстрактное синтаксическое дерево программного кода. Абстрактное синтаксическое дерево - основная структура представляющая процесс передачи данных от одной операции программы к другой. Следует, отметить что скорость работы алгоритма перевода из обратной польской записи в абстрактное синтаксическое дерево крайне низка и требует оптимизации. В связи с этим основной проблемой, поставленной и рассматриваемой в статье является оптимизация работы алгоритма преобразования обратной польской записи программы в абстрактное синтаксическое дерево.

Для оптимального разбиения задачи на подзадачи рассмотрим ее с точки зрения алгоритма. В этом случае, каждая, поступающая на выполнение в параллельный кластер задача - есть

алгоритм $AI^{(j)}$, реализованный в понятном вычислительной системе виде. Выполнение

программы $PR^{(j)}$ на вычислительной машине или группе параллельных вычислительных машин осуществляется посредством четких машинных команд. Тогда, для оптимального разбиения программы на подпрограммы, а затем и их выполнения требуется рассмотреть ее представление с позиции теории компиляции.

С теории компиляции всякая программа представляется в виде $PR^{(j)} = (Gpu, V)$, где $Gpu = (W, E, start, stop)$ - граф потока управления программы, а $V = \{v_1, \dots, v_k\}$ - алфавит операторов.

Граф потока управления $Gpu = (W, E, start, stop)$ - это ориентированный граф, в котором выделены две вершины start и stop, связанные между собой посредством множеств вершин W и дуг E.

Для описания программы исполняющейся на вычислительных машинах традиционной архитектуры с теории компиляции этого достаточно. Граф потока управления программы, исполняющейся на нейрокомпьютерах можно представить следующим образом

$G_{pu} = (V, E, start, stop, BitVal)$, где BitVal – множество значений разрядности результатов на каждом шаге выполнения программы.

Для разбиения программы на подпрограммы с позиции компиляции, требуется рассмотреть программу во внутренней форме представления.

Внутренняя форма представления программ - это результат работы синтаксического анализатора. Наиболее распространенным способом представления программ во внутренней форме является обратная польская запись [10,11].

В рассматриваемой работе выбрана обратная польская запись в качестве внутренней формы представления программ.

Модифицированный алгоритм, позволяющий представлять всю программу в постфиксной польской записи, а не только ее логические и математические выражения работает по тому же принципу и был предложен Робертом Седжвиком [17].

Алгоритм перевода из Польской формы в абстрактное синтаксическое дерево

После представления программы в обратной Польской записи необходимо построить граф потока управления программы. Представление программы в расширенной обратной Польской записи хранится и передается в форме абстрактного типа данных класса стек. Такой стек называется стеком команд программы (*Stack*). На следующем шаге необходимо сформировать абстрактное синтаксическое дерево (АСД) программы из стека команд программы. Абстрактное синтаксическое дерево – это ориентированное дерево, где внутренние вершины являются операторами языка, на котором написана программа, а листьями — соответствующие операнды. Листья нижнего слоя – это входные параметры, а, в свою очередь, корень дерева - результат, который должна вернуть программа. Ребра АСД формируются между операторами и соответствующими операндами, требующихся для их работы. Для этого существует алгоритм перевода программы из Польской формы в абстрактное синтаксическое дерево (рисунок 1). Следует, отметить что скорость работы алгоритма перевода из обратной польской записи в абстрактное синтаксическое дерево крайне низка и требует оптимизации. В связи с этим рассмотрим оптимизованный алгоритм преобразования обратной польской записи программы в абстрактное синтаксическое дерево.

Будем хранить в R текущий считываемый из *Stack* элемент, до тех пор пока он не пуст.

- В случае, когда R - идентификатор или константа, его значение считывается из стека и записывается в лист АСД а также в многосвязный список ссылаясь на элемент родитель в дереве, затем переходим к считыванию следующего элемента.
- В случае, когда R является бинарным оператором, его действие осуществляется над двумя последующим элементам, которые являются операндам из стека, так формируется левое поддереву АСД. Затем переходим к считыванию следующего элемента.
- В случае, когда R является унарным оператором, его действие осуществляется над верхним последующим элементом стека, так формируется левое поддереву АСД. Затем переходим к считыванию следующего элемента.
- В противном случае R является n -арным оператором, его действие осуществляется над всеми n верхними последующими операндами, так формируется левое поддереву АСД. Затем переходим к считыванию следующего элемента.

Вывод

Таким образом, в статье рассмотрен вариант оптимизации алгоритма перевода программы,

все операции которой, записаны в модифицированной обратной польской записи, в абстрактное синтаксическое дерево, который отобразит множество всех путей исполнения программы. Результатом работы данного алгоритма является алгоритмическая структура – многосвязный список, посредством которой осуществляется хранение и передача для дальнейшей обработки результатов представления программы в АСД.
Работа выполнена при финансировании государственного задания, проект №2.9519.2017/8.9.

Список литературы:

1. Бурцев В. С. Параллелизм вычислительных процессов и развитие архитектуры супер ЭВМ. М.: ИВВС РАН, 1997. 152 с.
2. Воеводин В. В., Воеводин Вл. В. Параллельные вычисления. СПб.: БХВ-Петербург, 2002. 608 с.
3. Галушкин А.И. Нейронные сети: основы теории. М.: «Горячая линия Телеком», 2010. 496 с
4. Топорков В.В. Модели распределенных вычислений. М.: ФИЗМАТЛИТ, 2004. 320 с.
5. Ручкин В.Н., Романчук В.А., Лукашенко В.В. Обобщенная модель вычислений кластера нейрокомпьютеров // Вестник Рязанского государственного университета им. С.А. Есенина. 2015. № 2 (47). С. 146-150.
6. Гергель В.П., Полежаев П.Н. Исследование алгоритмов планирования параллельных задач для кластерных вычислительных систем с помощью симулятора // Вестник Нижегородского университета им. Н.И. Лобачевского. 2010. № 5-1. С. 201-208.
7. Злобин В.К., Ручкин В.Н. Нейросети и нейрокомпьютеры: учеб. пособие./В.К. Злобин, В.Н. Ручкин.-СПб.: БХВ-Петербург, 2011.-256 с.: ил
8. Коваленко Вп Коваленко Е., Корягин Д. и др. Управление заданиями в распределенной вычислительной среде // Открытые системы. - 2001. - №5-6. -С. 22-28.
9. Комарцова Л.Г., Максимов А.В. Нейрокомпьютеры: учеб. пособие для вузов. -М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. -320 с., ил.
10. Коновалов Н., Крюков В. Параллельные программы для вычислительных кластеров и сетей // Открытые системы. - 2002. - №3. - С. 12-18.
11. Корячко В. П. Алгоритм планирования вычислительного процесса в мультипроцессорной вычислительной системе реального времени // Автоматика и вычислительная техника. 1985. № 3. С. 16-18.
12. Костров Б.В., Ручкин В.Н. Архитектура микропроцессорных систем: учеб. пособие.-М.:ТЕХБУК, 2007.-208 с.
13. Bender M.A., Bunde D.P., Demaine E.D. Communicationn Aware Processor Allocation for Supercomputers // Lecture Notes in Computer Science. V. 3608/2005. - Berlin: Springer, 2005. - P. 1699181.
14. Танненбаум Э., Ван Стен М. Распределенные системы. Принципы и парадигмы. СПб.: Питер, 2003. 877 с.
15. Тель Ж. Введение в распределенные алгоритмы. Пер. с англ. - М.: МЦНМО, 2009. - 616 с.
16. Грис Д. Конструирование компиляторов для цифровых вычислительных машин. – М.: Мир, 1975. – 544 с.

17. Роберт Седжвик. Алгоритмы на C++. Фундаментальные алгоритмы и структуры данных. Algorithms in C++. — М.: «Вильямс», 2011. — 1056 с.