Подход к представлению вычислительных задач в терминах dataflow на распределенных системах*

С.О. Кисляков, А.Н. Сальников

Московский государственный университет имени М.В.Ломоносова

Многие университеты и исследовательские группы активно изучают концепцию управления выполнением программы с помощью потока данных (dataflow). Данный подход лишен недостатков управления с помощью единственного счетчика команд, препятствующего распараллеливанию на уровне инструкций. В dataflow единственным критерием для выполнения следующей операции является готовность данных. Нами были рассмотрены существующие dataflow системы: Dryad [1], LuNA [2], PARUS [3]; выявлены преимущества и недостатки подходов представления задач в синтаксисе языка и способов генерации исполняемого кода для распределенных вычислительных систем.

Нами разрабатывается система Frigate — следующий виток развития системы PARUS. Frigate позволяет программисту описать алгоритм в специальном формате. По данному представлению генерируется набор функций на языке C++, для обмена данными используется библиотека MPI. Затем код компилируется и связывается с библиотекой, которая реализует подсистему времени запуска и размещает функции по узлам вычислительной системы. Среди MPI-процессов выделяются процессы координаторы, управляющие выполнением программы. Полученная программа запускается на вычислительном кластере.

Пользователю необходимо представить алгоритм решения как граф, состоящий из набора ориентированных ациклических подграфов. Вершинам в графе соответствуют вычисления, обрабатывающие входные данные и получающие в результате данные на выход. Рёбра в графе бывают трёх типов: внутренние, внешние и управляющие. Внутренние рёбра представляют собой зависимости по данным между вершинами одного подграфа, Внешние рёбра соединяют вершины разных подграфов. Данные, передаваемые по внешнему ребру, буферизуются до окончания выполнения текущего подграфа. Управляющее ребра направлены от вершин подграфа к сущности координатора, их назначение — изменять глобальные данные на тех MPI-процессах, которые являются координаторами.

Система Dryad предоставляет широкие возможности для описания графа программы, но, в отличии от Frigate, не дает возможности перестраивать граф в процессе выполнения. За счет введения конструкций, явно описывающих граф, процесс фрагментирования программы более интуитивен для программиста, чем в системе LuNA, хоть и остается полностью за ним.

Литература

1. M. Isard, M. Budiu, Y. Yu, A. Birrell, D. Fetterly Dryad: distributed data-parallel programs from sequential building blocks //European Conference on Computer Systems (EuroSys), March 21–23, 2007, Lisbon, Portugal. P. 59–72.

2. Малышкин В.Э. Технология фрагментированного программирования //Параллельные вычислительные технологии (ПаВТ'2012): труды международной научной конференции Новосибирск: 26–30 марта 2012 г. С. 592–599.

3. Alexey N. Salnikov PARUS: A Parallel Programming Framework for Heterogeneous Multiprocessor Systems //Lecture Notes in Computer Science (LNCS 4192) Recent Advances in Parallel Virtual Machine and Message Passing Interface, 2006. P. 408–409.

*Работа проводится при поддержке грантов РФФИ 11-07-00756-а, 11-07-00614-а.