

BNVTEST - РЕАЛИЗАЦИЯ МЕТОДОВ ВЕТВЕЙ И ГРАНИЦ НА ПЛАТФОРМЕ BOINC¹

Тянь Бо(yesyestian@gmail.com)

Факультет вычислительной математики и кибернетики МГУ имени М. В. Ломоносова

Задачи комбинаторной оптимизации возникают во многих областях применения вычислительных методов, в частности, таких как искусственный интеллект, исследование операций и т. д. Метод ветвей и границ — общий алгоритмический метод для нахождения оптимальных решений различных задач оптимизации, особенно дискретной и комбинаторной оптимизации. Для задач большой размерности это требует значительных и, в известной степени, неоправданных с практической точки зрения затрат времени. Применение параллельных или распределенных вычислений в задачах глобальной оптимизации дает возможность получить оптимальный вариант за приемлемое время.

Распараллеливания метода ветвей и границ для различных параллельных систем исследовалась многими авторами. Но эти методы полезны только для тесно связанных систем или систем с общей памятью. Крупномасштабные добровольные гриды еще не были рассмотрены в качестве целевой платформы для метода ветвей и границ. Наш подход к проектированию и реализации программ позволяет решить эту проблему.

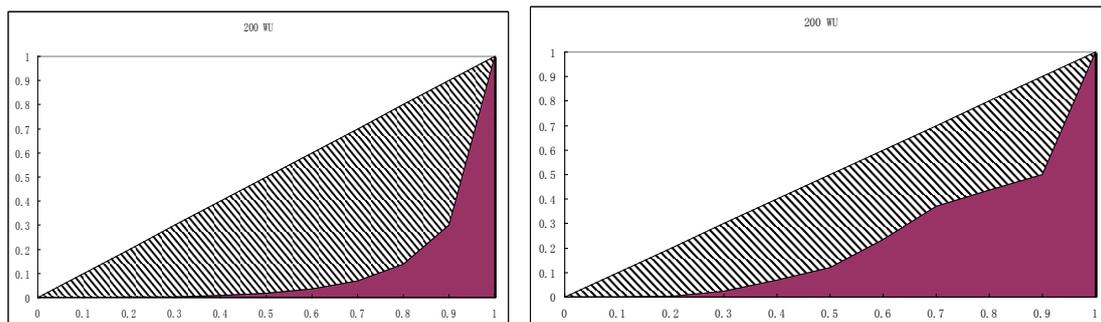
Наиболее популярная открытая программная платформа для организации добровольных вычислений – это BOINC (Berkeley Open Infrastructure for Network Computing). Она изначально разрабатывалась для проекта SETI@home [1] в Berkeley Spaces Sciences Laboratory (США). С 2002 года платформа BOINC [2] была сделана открытой, и с 2004 года на ее основе стали создаваться другие проекты. В данной работе исследуется реализация метода ветвей и границ (МВГ) в BOINC системах на примере задачи о булевом ранце с одним ограничением.

Проект BNVTesT@HOME разделяет вычисление метода ветвей и границ на несколько меньших подзадач, обрабатываются на клиентской стороне. Результаты, полученные клиентами, объединили в одно решение на стороне сервера.

Мы используем поиск в ширину на стороне сервера, для ветвления каждый раз берется задача, находящаяся на слое с минимальным номером. Эта стратегия порождает подзадачи, которые распределены более равномерно по сравнению с другими стратегиями поиска. На стороне клиента используется поиск в глубину. Требования к объему памяти при поиске в глубину существенно меньше, чем при поиске в ширину. Такая стратегия обеспечивает повышение эффективности использования ресурсов в вычислительных волонтеров.

После того, как сервера генерирует определенное количество подзадач, рассматриваются две возможные стратегии. Обе стратегии создают расчетные блоки (WorkUnit, WU), которые состоят из равного количества подзадач. В первой «плотной» стратегии, мастер-сервер подбирает непрерывно подзадачи из дерева ветвлений и помещает их в WU (Рис. 1).

¹Работа выполнена при финансовой поддержке Стипендии правительства Китая (грант № 201308090004).



(a) Плотная стратегия

(b) Разреженная стратегия

Рис. 3 Кривая Лоренца для двух стратегий распределения

Для сравнения обеих стратегий распределения работы были проведены численные эксперименты. Использовались три компьютера, обладающие суммарно с 10-ю процессорными ядрами качестве волонтеров. Полное (суммарное) время выполнения всех расчетных блоков составило 16902.693 секунды. Время выполнения распределенного приложения составило 3944.558 секунд для Плотной стратегии и 3703.242 секунды для Разреженной.

В будущем мы планируем попробовать различные стратегии, и изучение их влияния на эффективность алгоритма.

Список литературы

- [1] David P. Anderson, Jeff Cobb, Eric Korpela, Matt Lebofsky, and Dan Werthimer, SETI@home: an experiment in public-resource computing, *Commun. ACM*45(11):56--61 (2002)
- [2] BOINC: A System for Public-Resource Computing and Storage. David P. Anderson. 5th IEEE/ACM International Workshop on Grid Computing. November 8, 2004, Pittsburgh, USA.
- [3] David Pisinger, Where are the hard knapsack problems? *Computers & Operations Research*, Volume 32, Issue 9, September 2005, Pages 2271–2284.