

Использование программного пакета GULP на системах добровольных вычислений

М.А. Посыпкин^{1,a}, Н.П. Храпов^{2,b}, В.В. Ройзен^{3,c}

¹Вычислительный центр им. А. А. Дородницына
Федерального исследовательского центра «Информатика и управление» Российской академии наук
Россия, 119333, г. Москва, ул. Вавилова, д. 44, к. 2

²Институт Проблем Передачи Информации Российской академии наук
Россия, 127051, г. Москва, Большой Каретный пер., д. 19, стр.1

³Лаборатория компьютерного дизайна материалов, Московский физико-технический институт
Россия, 141700, Московская область, г.Долгопрудный, ул.Первомайская, д.5

E-mail: ^a mposypkin@gmail.com, ^b nkhrapov@gmail.com, ^c valmipt@gmail.com

Предсказание структуры вещества является важным направлением в химии и в физике. На настоящий момент существует много реализаций программного обеспечения для решения задач в данной области. Работа содержит обзор основных пакетов ПО. Главный акцент в статье сделан на адаптацию пакета GULP (General Utility for Lattice Program) к системам добровольных вычислений.

Программа GULP предназначена для решения широкого спектра задач моделирования структур, состоящих из множества частиц, методом силового поля. Работа содержит описание технических аспектов функционирования данной программы и различных подходов её адаптации для запуска в рамках инфраструктуры BOINC. Статья также содержит описание проведённых вычислительных экспериментов, автоматизации анализа результатов вычислений и генерации новых порций вычислительных заданий на основе проведённого анализа. В заключении приводятся рекомендации по организации вычислений с системами подобного рода.

Ключевые слова: предсказание структуры вещества, добровольные вычисления, эволюционный алгоритм, методы оптимизации

© 2016 Михаил Анатольевич Посыпкин, Николай Павлович Храпов, Валерий Валерьевич Ройзен

Введение¹

Кристаллическая структура вещества является наиболее важным носителем информации о нём – зная структуру, можно предсказать широкий набор его свойств. Это позволяет создавать материалы с нужными характеристиками путём подбора нужной конфигурации атомов. С точки зрения математики это является задачей глобальной многопараметрической оптимизации, численное решение которой требует разработки надёжного и эффективного метода.

Рассмотрим в качестве примера задачу предсказания кристаллической структуры. Из простых соображений комбинаторики [2] легко получить оценку числа потенциально возможных кристаллических структур:

$$C = \binom{V/\beta^3}{N} \prod_i n_i$$

где N - это полное число атомов в элементарной ячейке объёмом V , β – разумный параметр дискретизации (например, 1 ангстрем) и n_i число атомов типа i в элементарной ячейке.

Даже для небольших систем ($N \approx 10-20$), параметр C принимает астрономическое значение (приблизительно $10N$).

Используя особенности изучаемых систем (кристаллов, полимеров, наночастиц), реально понизить размерность решаемой задачи на несколько порядков. Тем не менее, даже в этом случае нужны большие масштабные вычислительные ресурсы. Поэтому актуальна проблема создания вычислительных систем нового поколения, адаптированных для компьютерного дизайна материалов.

Эволюционный алгоритм

USPEX - это эволюционный алгоритм [2], первоначально разработанный для предсказания кристаллических структур. В настоящее время он также адаптирован для предсказания нанокластеров [4] и полимеров [5]. Его эффективность обусловлена сочетанием глобальной и локальной оптимизации, что позволяет эффективно исследовать пространство поиска и избегать "залипания" в локальных минимумах.

Общая схема работы алгоритма представлена на рис. 1. Ниже приводится краткое описание его работы.

¹ Работа была выполнена при финансовой поддержке Российского Научного Фонда (грант № 16-11-10352).



Рис.1. Схема работы алгоритма USPEX

Весь расчёт разбит на поколения, состоящие из структур. Первое поколение генерируется случайным образом с помощью операторов симметрии, характерных для изучаемой системы (например, пространственных групп симметрии для кристаллов). Затем все структуры релаксируются в несколько этапов для приведения их к локальным минимумам. Для этого используются сторонние программы для квантово-химических расчётов (например, VASP - Vienna Ab-initio Simulation Package) или для классических расчётов методами потенциалов силового поля (например, GULP – General Utility for Lattice Program). В данной работе применялась программа GULP [8]. Это свободное ПО, позволяющее проводить расчёты с помощью различных потенциалов силового поля.

После релаксации заданный процент лучших структур в поколении отбирается для создания следующего поколения с помощью эволюционных операторов. В коде USPEX реализовано несколько функций соответствия, использование конкретной из них определяется при инициализации расчёта. Детали работы эволюционных операторов описаны ранее [2, 9]. Для сохранения разнообразия в популяции, критично важного для эффективной работы эволюционного алгоритма, также добавляется некоторое количество случайных структур. После чего все итерации повторяются ещё раз, пока не выполняются критерий остановки расчёта (по умолчанию это неизменность лучшей структуры в течение фиксированного числа поколений).

Наиболее затратным с точки зрения вычислительных ресурсов является этап релаксации структур. Использование системы добровольных вычислений позволяет ускорить расчёты в несколько раз. Далее мы подробно опишем принципы работы разработанной системы.

Системы добровольных вычислений

Если сравнивать принцип функционирования грид-систем из персональных компьютеров (далее ГСПК) с сервисными грид-системами и кластерами, то можно выделить основные отличия:

1. Большинство кластерных систем предоставляют возможность обмена данными между различными нитями распределённого приложения. А в ГСПК обмен данными осуществляется только между программным обеспечением промежуточного уровня, установленного на вычислительном узле, и сервером проекта, прямого обмена данными между вычислительными узлами нет.
2. Вычислительные кластеры предполагают безотказное функционирование вычислительных узлов в режиме. В ГСПК требуется относительно большое время на инициализацию задания, включающее в себя подготовку, копирование данных на вычислительный узел и получение результатов обратно. Время на инициализацию скачанного задания составляет от 10-ти до 20-ти секунд (зависит от состояния вычислительного узла).
3. Вычислительные кластеры предполагают постоянное безотказное функционирование вычислительных узлов. В ГСПК любой узел может в любой момент на неопределённое время прекратить выполнять расчёты. Это приводит к очень существенной разнице во времени обработки заданий одинаковой сложности различными вычислительными узлами.

Для исследования влияния перечисленных факторов на производительность распределённой реализации метода USPEX и выработки оптимальной стратегии такой реализации был разработан программный прототип для платформы BOINC. BOINC (Berkeley Open Infrastructure for Network Computing) представляет собой платформу с открытым кодом для организации проектов добровольных вычислений. Разработка системы ведётся в U.C. Berkeley Spaces Sciences Laboratory (США) исследовательской группой, которая также разрабатывала проект SETI@home. Работа над BOINC была начата в 2002 году с целью создания универсальной программной платформы для проектов добровольных вычислений и интеграции вычислительных ресурсов внутри предприятия. Использование технологии BOINC позволяет упростить процесс развертывания необходимой инфраструктуры и разработки приложений. Первый проект добровольных вычислений на основе BOINC был запущен в 2004 году. В настоящее время насчитывается более 80 публичных проектов на основе BOINC, делая платформу фактическим отраслевым стандартом в данной области.

Все программное обеспечение BOINC можно разделить на две основные компоненты: клиентскую и серверную части программного обеспечения. Клиентская часть устанавливается на вычислительном узле. В её задачи входит:

1. Подключиться к одному из проектов, к какому именно указывает владелец машины.
2. Запрашивать задания у центрального сервера.
3. Скачивать задания с сервера, если они там есть.
4. Запускать у себя скачанные задания.
5. Результаты работы заданий отсылать обратно на сервер.

Серверная часть программного обеспечения BOINC выполняет следующие действия:

1. Создает задания для пересылки на вычислительные узлы.
2. Отвечает на клиентские запросы, отправляет задания на вычислительные узлы.
3. Получает результаты работы задания и передает их для дальнейшей обработки.
4. Содержит в себе web-сервер для получения информации о проекте через web-интерфейс.

Клиентская часть распределённого приложения и есть исполняемый файл, запускаемый на вычислительном узле. Она выполняет основную вычислительную нагрузку. Серверная часть распределённого приложения создает задания (расчётные блоки) для вычислительных узлов.

Как правило, расчётный блок состоит из исполняемого файла клиенткой части, объединённый со специфическим для конкретного задания входным файлом с данными. После отправки задания в вычислительную инфраструктуру серверная часть распределённого приложения ждёт результатов задания. Получив из инфраструктуры все результаты заданий, серверная часть производит их обработку, и создает единый результат работы распределённого приложения.

Интеграция алгоритма USPEX и систем добровольных вычислений

Принцип взаимодействия алгоритма USPEX с вычислительными ресурсами хорошо сочетается с системами добровольных вычислений. Система USPEX предоставляет возможность добавлять интерфейсы для интеграции с различными системами параллельных и распределённых вычислений. Таким образом, можно осуществить интеграцию систем, используя встроенные возможности алгоритма USPEX.

Одной из основных проблем интеграции является балансировка нагрузки между вычислительными узлами. Время выполнения отдельного USPEX-задания может варьироваться от 10-ти секунд до нескольких часов. Таким образом, получается, что генерация одного задания BOINC на основе одного задания USPEX неэффективна для малых заданий, т.к. издержки на инициализацию превышают полезную работу. По этой причине при интеграции систем использовался подход на основе агрегации нескольких заданий USPEX в одно задание BOINC. Такой подход позволяет снизить издержки на инициализацию одного задания. Параметры для агрегации подбирались эмпирическим способом.

Для осуществления интеграции был разработан набор программных компонентов, осуществляющий взаимодействие между USPEX и BOINC-сервером. На стороне USPEX запускаются скрипты, осуществляющие отправку заданий в инфраструктуру BOINC, а на стороне BOINC-сервера были реализованы скрипты, обрабатывающие запросы системы USPEX.

Заключение

Для тестирования стабильности разработанной системы было проведено несколько серий экспериментальных расчётов. Использовались две задачи. Первая задача является стандартным примером USPEX по предсказанию кристаллической структуры $MgAl_2O_4$ при давлении 100 ГПа. Это вычисление с переменными параметрами ячейки с помощью потенциала Букингема, реализованного в коде GULP. Вторая задача состояла в моделировании кластера, состоящего из 150-ти молекул, взаимодействие между молекулами описывается потенциалом Морзе.

Полученные результаты позволяют заключить, что разработанная система успешно работает. Практика проведения расчётов выявила направления для дальнейшей модернизации существующего вычислительного комплекса.

Во-первых, необходима дальнейшая адаптация самого алгоритма USPEX к работе с нестабильными вычислительными узлами. USPEX изначально разрабатывался для проведения расчётов на ПК и суперкомпьютерных кластерах. Данные системы имеют стабильную архитектуру и обеспечивают высокую скорость передачи данных, что важно для работы эволюционного алгоритма, который не может перейти к генерации нового поколения без полного завершения текущего.

Во-вторых, необходимо портирование решателей на наиболее часто используемые платформы. Данный подход позволит существенно повысить потенциал имеющейся инфраструктуры. Одним из способов портирования является использование технологий виртуализации.

Также для дальнейших исследований в данной области необходима разработка BOINC-интерфейса для более широкого спектра ПО, связанного с квантово-химическими расчётами.

Список литературы

- Oganov A.R., Schön J.C., Jansen M., Woodley S.M., Tipton W.W., Hennig R.G. Modern Methods of Crystal Structure Prediction (ed. A.R. Oganov), pp. 223-231. Berlin: Wiley-VCH. 2010.
- Oganov A.R., Glass C.W. Crystal structure prediction using ab initio evolutionary techniques: principles and applications. *J. Chem. Phys.* 124, art. 244704. 2006.
- Zhang W.W., Oganov A.R., Goncharov A.F., Zhu Q., Boulfelfel S.E., Lyakhov A.O., Stavrou E., Somayazulu M., Prakapenka V.B., Konopkova Z. Unexpected stoichiometries of stable sodium chlorides. *Science* 342, 1502-1505. 2013.
- Matsko N. L., Tikhonov E. V., Baturin V. S., Lepeshkin S. V., Oganov A. R. The impact of electron correlations on the energetics and stability of silicon nanoclusters. *J. Chem. Phys.* 145, 074313. 2016.
- Sharma, V., Wang, C., Lorenzini, R. G., Ma, R., Zhu, Q., Sinkovits, D. W., Pilania, G., Oganov, A. R., Kumar, S., Sotzing, G. A., Boggs S.A., Ramprasad R. Rational design of all organic polymer dielectrics. *Nat. Commun.* 5, art. 4845. 2014.
- Anderson, D. P. Boinc: A system for public-resource computing and storage. In *Grid Computing, 2004. Proceedings. Fifth IEEE/ACM International Workshop on* (pp. 4-10). 2004. November.
- Simons K, Kooperberg C, Huang E, Baker D. Assembly of Protein Tertiary Structures from Fragments with Similar Local Sequences using Simulated Annealing and Bayesian Scoring Functions. *J. Mol. Biol.* 268:209-225. 1997.
- J.D. Gale and A.L. Rohl. The General Utility Lattice Program (GULP). *Mol. Simul.*, 29:291–341, 2003.
- Oganov A.R., Lyakhov A.O., Valle M.. How evolutionary crystal structure prediction works - and why. *Acc. Chem. Res.* 44, 227-237. 2011.

Using General Lattice Program on the desktopgrid systems

M.A. Posypkin^{1,a}, N.P. Khrapov^{2,b}, V.V. Rozen^{3,c}

¹Dorodnicyn Computing Centre, Federal Research Center “Computer Science and Control” of Russian Academy of Sciences, 44, b. 2, Vavilov st., Moscow, 119333, Russia

²Institute for Information Transmission Problems of Russian Academy of Sciences, 19, b. 1, Bolshoy Karetny., Moscow, 127051, Russia

³Computational Materials Discovery Lab, Moscow Institute of Physics and Technology, 5, Pervomayskaya st., Dolgoprudny, Moscow region, 141700, Russia

E-mail: ^amposypkin@gmail.com, ^bnkhrapov@gmail.com, ^cvalmipt@gmail.com

Crystal structure prediction (CSP) is an important direction in chemistry and physics. At the moment there are a lot of software implementations allowing to solve this problem. The article contains the overview of the of the basic packages of this software. The main emphasis of this article is made on package GULP (General Utility for Lattice Program) adaptation to volunteer computing systems.

The GULP program is designed for a wide range of tasks of modeling many-particle structures by the force field. The report contains description of the technical aspects of the program and different approaches to its adaptation for running under the BOINC infrastructure. The article also contains a description of conducted computational experiments and various approaches to automation of analysis of the calculation results, and also the generation of new tasks based on the computational results. The recommendations on organization of computations with such kind of systems a given in summary.

Keywords: structure prediction, volunteer computing, evolutionary algorithm, optimization methods