

Организация добровольных вычислений на платформе *BOINC* на примере проектов *OPTIMA@home* и *SAT@home*

О.С. Заикин, М.А. Посыпкин, А.А. Семёнов, Н.П. Храпов

Статья посвящена вопросам организации проектов добровольных вычислений, базирующихся на платформе *BOINC* (boinc.berkeley.edu) и предназначенных для решения задач в распределенных средах с низкой интенсивностью обменов между вычислительными узлами. Детали и особенности процесса проиллюстрированы на примере реально функционирующих проектов добровольных вычислений *OPTIMA@home* и *SAT@home*. Первый предназначен для решения задач глобальной оптимизации, второй – для решения комбинаторных задач, сведенных к задачам о булевой выполнимости (*SAT*).

Введение

Добровольные вычисления – относительно новое направление в распределенных вычислениях, идеология которого предполагает предоставление личных вычислительных ресурсов для организации масштабных расчетов так называемыми добровольцами, или “волонтерами” (*volunteers*). В роли волонтеров обычно выступают частные лица, располагающие собственными персональными компьютерами (ПК), ресурсы которых они согласны предоставить для решения разнообразных научных задач. Следует отметить, что при этом задействуются только свободные ресурсы, не используемые ПК для других целей. Таким образом, участие в добровольных вычислениях не мешает собственной работе пользователей-волонтеров.

Компьютеры добровольцев, которые могут находиться в географически удаленных друг от друга точках, объединяются в *grid* под управлением некоторой среды. Такие *grid*-системы создаются под конкретные задачи (их называют проектами добровольных вычислений), на решение которых могут уходить месяцы и даже годы. В основном, эти проекты инициируются университетами и научными центрами, которые не располагают необходимыми вычислительными ресурсами.

За последние несколько лет были организованы проекты добровольных вычислений, предполагающие решение широкого круга задач – от задач теории чисел (*ABC@home*) и криптографии (*distributed.net*) до поиска новых космических объектов (*Einstein@home*), предсказания изменения климата Земли (*Climate Prediction*), поиска новых лекарств от пока неизлечимых болезней (*World Community Grid*).

Наиболее популярная открытая программная платформа для организации добровольных вычислений – это *BOINC* (*Berkeley Open Infrastructure for Network Computing*). Она изначально разрабатывалась для

проекта *SETI@home* в *U.C. Berkeley Spaces Sciences Laboratory* (США). С 2002 года платформа *BOINC* была сделана открытой (по лицензии *LGPL*), и с 2004 года на её основе стали создаваться другие проекты. По состоянию на 02.04.2012 г. проекты на платформе *BOINC* суммарно обладают следующими характеристиками:

- 295 тысяч активных участников;
- 949 тысяч активных компьютеров;
- 6.4 *PFLOPS*, что сопоставимо с производительностью мощнейших суперкомпьютеров мира.

Основные принципы организации добровольных вычислений

Вначале мы кратко коснемся истории добровольных вычислений. В 1996 году стартовал проект *GIMPS* по поиску простых чисел Мерсенна, а в 1997 году – *distributed.net*, направленный на решение задачи криптоанализа шифра *RC5* (изначально проект предполагал поиск 56-битного секретного ключа). В июле 1996 года на 5-й международной конференции по биоастрономии была представлена концепция проекта *SETI@home*, предназначенного для обработки интенсивных потоков данных, поступающих от мощных радиотелескопов. Этот проект был запущен в 1999 году, а в 2002 году на его основе была разработана открытая платформа *BOINC*. И если изначально для создания проектов добровольных вычислений требовались ресурсы больших научных коллективов, то с появлением *BOINC* построение каждого нового проекта стало вполне по силам небольшим лабораториям и даже отдельным энтузиастам. Из 70-ти активных на данный момент проектов добровольных вычислений 65 базируются на платформе *BOINC*.

Проект на платформе *BOINC* состоит из трех основных частей (рис. 1):

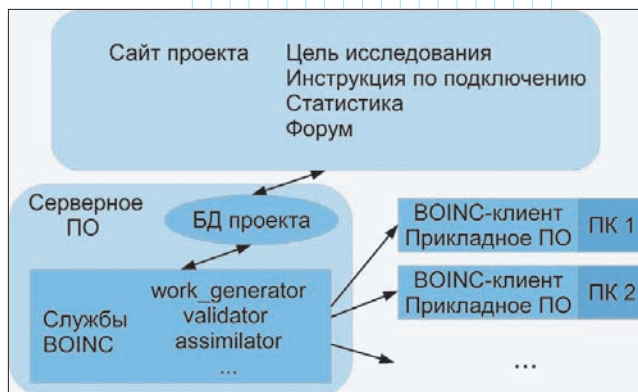


Рис. 1. Общая схема работы проекта добровольных вычислений на платформе *BOINC*

- 1 серверное ПО;
- 2 веб-сайт;
- 3 прикладное ПО.

Серверное и прикладное ПО образуют распределенное приложение.

Ниже перечислены основные службы, входящие в состав серверного ПО проекта:

- *work_generator* – создает задания для обработки;
- *validator* – проверяет корректность результатов, присланных с ПК волонтеров, а также начисляет кредиты за корректные результаты;
- *assimilator* – обрабатывает корректные результаты.

Для того чтобы подключиться к проекту, пользователь-доброволец должен установить на свой ПК стандартную программу “BOINC-клиент”. Она осуществляет обмен данными с сервером выбранного проекта и выделяет ресурсы ПК пользователя для работы прикладного ПО. Пользователь должен указать адрес (*URL*) сайта выбранного проекта и зарегистрироваться на нём (достаточно сообщить только адрес своей электронной почты). После этого вычисления на стороне пользователя происходят автоматически: BOINC-клиент сам скачивает с сервера проекта прикладное ПО (ориентируясь на тип ОС и процессора ПК пользователя) и задания для обработки. Гибкая система настроек BOINC-клиента позволяет эффективно задействовать свободные ресурсы ПК и распределять их между проектами. После настройки BOINC-клиента волонтер становится полноправным участником выбранных им проектов.

Сайт проекта содержит следующую информацию: цели проводимых исследований, полученные результаты, список публикаций авторов проекта, производительность проекта, рейтинг лучших (в смысле количества набранных за участие в проекте кредитов) участников и т.п. Кроме того, участнику доступен форум, на котором идет общение с разработчиками проекта.

При организации проекта следует учитывать ряд факторов, способствующих его активному развитию. Обычно выделяются следующие основные причины, мотивирующие пользователей на участие в проекте:

- за каждое выполненное задание участникам проекта начисляются, пропорционально затраченным вычислительным ресурсам, так называемые “кредиты”. Количество кредитов является характеристикой, по которой участники соревнуются между собой. Участники могут объединяться в команды по разным признакам (национальному, региональному, пр.), которые также соревнуются между собой;
- при обнаружении очередного объекта поиска на сайте проекта публикуется информация об участнике, на ПК которого данный объект был найден;
- ощущение причастности к важным научным исследованиям – именно поэтому большой

популярностью пользуются медицинские проекты, направленные на поиск новых лекарств (*World Community Grid, Rosetta@home*).

Поскольку в проектах добровольных вычислений задействованы ПК частных лиц, всегда существует вероятность того, что результаты вычислений будут сфальсифицированы пользователем, либо окажутся некорректными в результате аппаратного или программного сбоя. Чтобы предотвратить это, обычно используются избыточные вычисления: для каждого задания формируются несколько копий, которые отправляются на ПК разных участников из разных команд. Полученные результаты сравниваются, и если они совпадают, то задание считается решенным. Если же имеются расхождения, то результаты аннулируются, после чего формируются новые копии задания, которые отправляются на ПК других участников.

Существуют три основных статуса проектов добровольных вычислений: “альфа”, “бета” и “релиз”. Можно выделить следующие критерии, по которым проектам присваивают соответствующие статусы:

- стабильность (доступность сайта и сервера проекта в режиме 24/7, постоянное наличие готовых к отправке заданий);
- степень завершенности серверного и прикладного ПО;
- наличие версий прикладного ПО для основных операционных систем (*Windows, Linux, Mac OS*);
- полнота и наглядность описания целей проекта, запланированных и полученных результатов.

Статусы проектов, использующих платформу BOINC, отображены на сайте *Formula BOINC* (formula-boinc.org). Даже для получения статуса “альфа” (это означает, что проект находится на начальном этапе разработки) проект проходит специальную экспертизу.

Следует отметить, что в СНГ на данный момент активно действуют четыре проекта добровольных вычислений на основе BOINC: три – на территории России (включая *OPTIMA@home* и *SAT@home*, которые будут описаны ниже) и один – Украины; все они имеют статус “альфа”.

Проект добровольных распределенных вычислений *OPTIMA@home*

Проект *OPTIMA@home* был запущен 20.06.2011 г. в Институте системного анализа РАН (boinc.isa.ru/dcsdg). Данный проект на основе платформы BOINC предназначен для распределенной реализации различных методов и стратегий глобальной оптимизации.

На *рис. 2* приведена общая схема работы распределенного приложения проекта *OPTIMA@home*.

Управляющая часть (серверное ПО) работает на сервере проекта. Основная задача серверного ПО – генерировать новые задания и обрабатывать результаты, полученные с ПК участников.

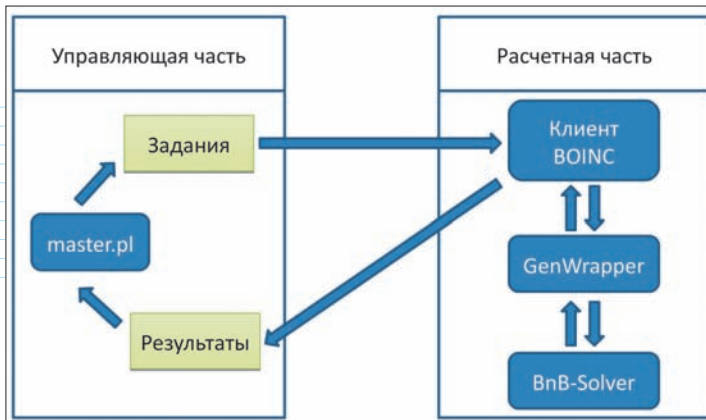


Рис. 2. Распределенное приложение проекта OPTIMA@home

Управляющая часть в проекте OPTIMA@home – это программа *master.pl* на языке *Perl*, которая запускается каждые полчаса. Она просматривает папку с полученными результатами, обрабатывает их и генерирует на их основе новые задания; сгенерированные задания помещаются в базу данных проекта.

Расчетная часть (клиентское ПО) представляет собой архив выполняемых файлов, который загружается на ПК участников. Получив очередное задание, BOINC-клиент запускает расчетную часть, которая обрабатывает полученное задание. Результат возвращается на сервер. Основой расчетной части в проекте OPTIMA@home является выполняемый модуль библиотеки *BnB-Solver*. Входной файл приложения содержит исходные данные задачи и параметры алгоритма, выходной – найденные решения. Оба файла используют формат *XML*.

Для запуска *BnB-Solver* на ПК участников служит программный инструмент *GenWrapper*, предназначенный для портирования в грид существующих приложений. При этом не требуется переписывать само приложение – *GenWrapper* самостоятельно дополняет его функциями из BOINC API, необходимыми для взаимодействия приложения с BOINC-клиентом.

На данный момент в рамках проекта OPTIMA@home реализован метод *Basin-Hopping* для решения задач глобальной оптимизации. Это эвристический метод, который в результате последовательных возмущений некоторого решения улучшает его. Перед началом работы на сервер загружается набор случайных точек. Каждая из точек является начальной для алгоритма *Basin-Hopping* на стороне участника. Улучшенное решение пересылается на сервер и заносится в список заданий, чтобы послужить новой начальной точкой. Таким образом, происходит постоянное улучшение решений.

Результатом реализации перечисленных пунктов стал устойчиво работающий проект OPTIMA@home. В качестве тестовой задачи была

выбрана задача поиска минимума энергии молекулярного кластера Морзе из 150-ти атомов. Полученное на данный момент решение оказалось лучше (по значению оптимизируемой функции), чем решения, найденные ранее на суперкомпьютере [1].

По состоянию на 02.04.2012 г. проект OPTIMA@home имеет более 2500 подключенных ПК, из которых 192 – активные. Средняя производительность проекта составляет порядка 0.5 TFLOPS.

SAT-задачи и сводимые к ним комбинаторные проблемы

Задачи о булевой выполнимости (сокращенно SAT – от английского *Satisfiability*) представляют собой задачи поиска наборов, выполняющих булевы формулы. Обычно рассматриваются формулы, приведенные к виду конъюнктивной нормальной формы (КНФ). В своей общей постановке задача о булевой выполнимости NP-трудна – то есть, полиномиальные алгоритмы для нее на текущий момент неизвестны. Но поскольку эта задача очень важна для современной прикладной кибернетики, то алгоритмы, показывающие высокую эффективность на различных “индустриальных” классах SAT-задач, чрезвычайно востребованы. Одной из областей использования таких алгоритмов является теория верификации дискретных систем. В настоящее время во многих научных центрах, занимающихся проблемами верификации, ведутся разработки и в области алгоритмики SAT-решателей.

Кроме проблем верификации, к SAT-задачам эффективно сводятся многие комбинаторные задачи на графах, задачи поиска различных экстремальных комбинаторных структур и, конечно же, задачи криптоанализа. Следует отметить, что SAT-подход в криптоанализе часто проигрывает по эффективности специализированным методам (для тех шифров, для которых такие методы существуют). Тем не менее, сама по себе способность SAT-решателей справляться с задачами криптоанализа является аргументом (хоть и нестрогим) вычислительной эффективности соответствующих алгоритмов и обосновывает их применение для решения комбинаторных проблем, не являющихся искусственно трудными (в отличие от задач криптоанализа).

Техника сводимости различных задач к SAT представляет отдельное важное и интересное направление, хотя с чисто теоретической точки зрения эффективность соответствующих процедур есть прямое следствие теоремы С. Кука. На практике можно каждую конкретную задачу сводить к SAT особым способом, используя либо системы логического проектирования (например, [2]), либо специализированные системы трансляции алгоритмических описаний в булевы уравнения (и, в конечном счете, в SAT) [3].

Когда SAT-задача, кодирующая некоторую комбинаторную проблему, построена, необходимо определиться с алгоритмом её решения. На подавляющем множестве SAT-задач, имеющих реальные “прототипы”, лучшие результаты показывают SAT-решатели, базирующиеся на алгоритме *DPLL*.

Параллельные SAT-решатели стали массово появляться совсем недавно, хотя первые теоретические работы по распараллеливанию соответствующих алгоритмов были опубликованы в 90-х годах прошлого века. Конкурсы параллельных SAT-решателей регулярно проводятся с 2008 года (www.satlive.org). В своем подавляющем большинстве современные параллельные SAT-решатели предполагают интенсивный обмен данными между вычислительными узлами – конкретно, происходит обмен параллельно накапливаемыми булевыми ограничениями (т.н. “конфликтными дизъюнктами”). Конечно же, использование таких решателей в проектах добровольных вычислений (да и вообще в грид) весьма проблематично. Тем не менее, отметим очень интересную работу [4], где описывается подход к построению распределенного SAT-решателя, обмен ограничениями в котором реализован в *peer-to-peer* сети, состоящей из ПК.

Описываемый далее подход к построению распределенных SAT-решателей в рамках проектов добровольных вычислений предполагает крупноблочный параллелизм [5]. Основной упор при этом делается на исследование исходной комбинаторной задачи и препроцессинг, результатом которого является построение параллельного списка заданий. Для обработки данного списка используются независимые друг от друга узлы распределенной вычислительной среды. Управление обработкой списка (рассылка заданий, получение и анализ ответов) происходит на специально выделенном для этой цели сервере.

Анализ исходной задачи и процесс декомпозиции обычно выполняются на параллельном вычислителе, на котором возможно интенсивное межпроцессорное взаимодействие. Как правило, для этой цели достаточно ресурсов маломощного кластера. При этом суммарные вычислительные затраты на данный этап существенно меньше тех, которые потом будут необходимы для обработки построенного параллельного списка. Весьма детально процесс декомпозиции SAT-задач, кодирующих проблемы обращения дискретных функций, описан в [6]. Здесь мы приведем совсем краткое описание этой техники.

Рассматривается некоторая дискретная функция, являющаяся преобразованием на множестве двоичных слов, которая задается программой *A*, написанной, например, на языке *Verilog* или на языке какой-либо специализированной системы трансляции алгоритмов в булевы уравнения [3].

Результатом работы данной программы является булево уравнение вида $KNF=1$, решая которое можно получить ответ на исходную задачу: например, найти секретный ключ анализируемого

шифра, доказать эквивалентность микрочипа эталонному образцу и т.п.

Базовая схема крупноблочного распараллеливания полученной SAT-задачи заключается в следующем. Среди переменных исходной КНФ *C* выбирается некоторое подмножество, состоящее из *d* переменных, после чего варьируются всевозможные их значения. Подстановка каждого такого вектора значений истинности в КНФ *C* дает новую КНФ *C'*, которая и является элементарным заданием параллельного списка. Всего, таким образом, в получаемом списке имеется 2^d элементарных заданий.

Если при переходе от текста программы *A* к КНФ *C* опираться на ряд простых правил [6], то можно гарантировать, что мощность списка, каждое задание в котором является вычислительно простым, заведомо не превосходит 2^n , где *n* – длина входного слова программы *A*. Как правило, эту мощность можно весьма существенно уменьшить, используя довольно естественные вычислительные процедуры, которые в совокупности составляют этап препроцессинга [5]. После того как параллельный список сформирован, он загружается в грид-систему с архитектурой клиент-сервер.

Описанный подход был с успехом применен в криптоанализе ряда генераторов поточного шифрования. В 2009 году этот подход был реализован в вычислительной системе *BNB-Grid* [7] в применении к криптоанализу широко известного генератора ключевого потока *A5/1*.

Дополнительно подчеркнем, что задачи криптоанализа для нас не являются самоцелью, а выступают в роли аргументированно трудных тестов. При этом еще раз повторим, что, с нашей точки зрения, успешная апробация вычислительной технологии на криптографических тестах означает её принципиальную применимость для решения задач, вычислительная трудность в которые не заложена искусственно. Успешность решения задач криптоанализа в грид-системе *BNB-Grid* стимулировала наши исследования по разработке проекта добровольных вычислений, ориентированного на решение SAT-задач. Таким проектом стал *SAT@home*.

Проект добровольных распределенных вычислений *SAT@home*

Совместный проект Института системного анализа РАН и Института динамики систем и теории управления (ИДСТУ) СО РАН, получивший наименование *SAT@home* (sat.isa.ru/pdsat), был запущен 29.09.2011 г. При создании проекта была использована открытая платформа *BOINC* (boinc.berkeley.edu) и пакет *SZTAKI Desktop Grid* [8]. С применением библиотеки *DC-API* было создано распределенное приложение, состоящее из управляющей и расчетной части. Управляющая часть отвечает за создание заданий в базе данных проекта, а также за обработку результатов выполнения заданий, присылаемых с ПК участников. Отправкой

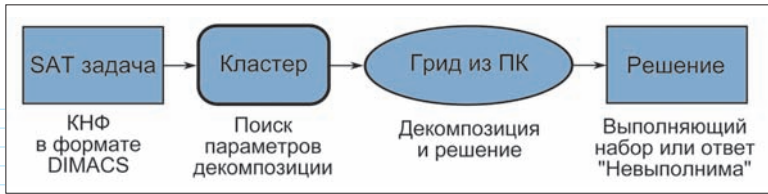


Рис. 3. Схема решения задач в проекте SAT@home

заданий на ПК участников и получением результатов занимаются стандартные службы BOINC.

В настоящий момент работа идет по схеме, представленной на рис. 3. Решатель PD-SAT [9], запущенный на вычислительном кластере (кластер Blackford ИДСТУ СО РАН – www.mos.iccr.ru), получает исходную SAT-задачу (КНФ в формате DIMACS – www.satlive.org) и вычисляет для нее “хорошие” параметры декомпозиции. Под “хорошими” имеются в виду параметры, найденные в результате применения специальной техники прогнозных функций, описанной в [5]. Параметрами декомпозиции являются способ выбора переменных, включаемых в декомпозиционное множество, и число этих переменных. Данный этап требует относительно небольшого времени вычислений (от нескольких минут до нескольких часов). Найденные параметры передаются управляющей части приложения проекта, которая осуществляет декомпозицию исходной SAT-задачи – разбивает её на подзадачи.

Основу расчетной части приложения составляют SAT-решатели *minisat-1.14.1* и *minisat-2.0* (minisat.se/MiniSat.html), модифицированные с учетом особенностей КНФ, кодирующих задачи обращения дискретных функций [6]. Исполняемые файлы расчетной части взаимодействуют с BOINC-клиентом при помощи функций библиотеки DC-API, что позволяет, в частности, реализовать периодическое сохранение промежуточных данных вычислений в виде контрольных точек. Решение исходной SAT-задачи находится в результате обработки всех (в общем случае) подзадач.

По состоянию на 02.04.2012 г. SAT@home имеет следующие характеристики:

- 2381 участников, из них активных – 990;
- 5179 ПК, из них активных – 2029;
- версии расчетного приложения – *windows x86*, *linux x86*, *linux x64*;
- средняя реальная производительность грида – 1.5 TFLOPS, пиковая – 4.3 TFLOPS.

На рис. 4 отображена динамика роста количества подключающихся к проекту ПК с 29.09.2011 г. по 02.04.2012 г. Каждый столбец отображает количество подключенных ПК с момента старта проекта по конкретный день. Из графика видно, что с середины октября 2011 года динамика подключений

существенно возросла. Это объясняется тем, что 10.10.2011 г. информация о проекте была размещена на статистическом сайте *Free-DC* (www.free-dc.org), который содержит сведения о проектах на платформе BOINC. Кроме того, в проекте был включен экспорт статистики, что позволило другим статистическим сайтам добавлять информацию о проекте в свои

списки. Наличие информации о проекте на основных статистических сайтах – важный фактор для привлечения новых участников. На рис. 5 представлена динамика изменения реальной производительности грида в гигафлопсах с 17.01.2012 г. по 14.02.2012 г.

Следует отметить, что в процессе работы проектов добровольных вычислений целесообразно организовывать командные соревнования среди участников. За время функционирования проекта SAT@home было организовано два таких соревнования на сайте *BOINCStats* (ru.boincstats.com). На период каждого соревнования производительность проекта возрастала примерно в два раза, в сравнении с его средней производительностью. Это можно проследить на графике (рис. 5) производительности проекта (с 10.02.2012 г. началось семидневное соревнование).

Важной составляющей частью успешности любого проекта добровольных вычислений является его популяризация. Для этого желательно

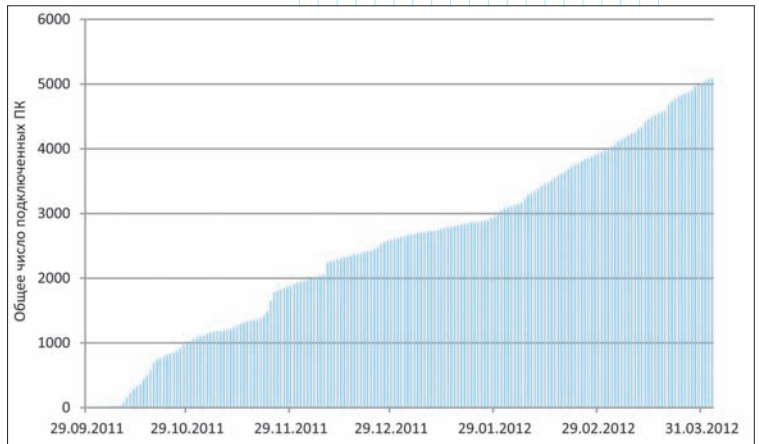


Рис. 4. Динамика подключения ПК участников к проекту SAT@home

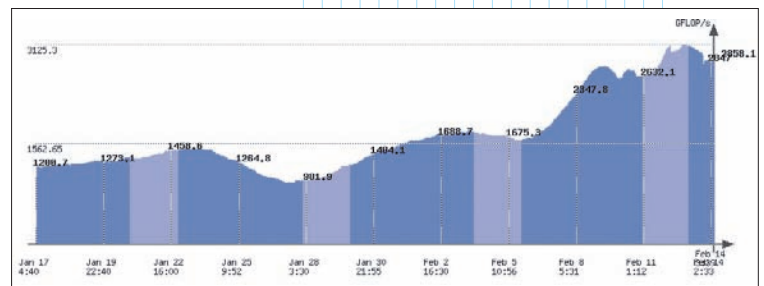


Рис. 5. Динамика изменения реальной производительности проекта SAT@home

сопровождать проект форумом на сайте проекта, а также размещать и обновлять информацию о проекте на специализированных информационных ресурсах, посвященных добровольным вычислениям (<http://boinc.ru>, <http://distributed.ru>).

Результаты, полученные в проекте SAT@home

В рамках проекта SAT@home, в середине октября 2011 года запущен эксперимент, который направлен на решение серии SAT-задач, кодирующих обращение функции, порождающей ключевой поток в генераторе A5/1 (длина инициализирующей последовательности – 64 бита; анализируется фрагмент ключевого потока длиной 114 бит, соответствующий одному фрейму ключевого потока).

На сегодняшний день наиболее эффективным методом криптоанализа A5/1 является т.н. “rainbow-метод” (reflector.com/trac/a51/wiki). Имеющиеся в открытом доступе rainbow-таблицы покрывают ключевое пространство примерно на 88% (при реалистичных условиях криптоанализа). На текущий момент в проекте SAT@home решаются 10 примеров криптоанализа шифра A5/1, которые не покрываются этими rainbow-таблицами.

Декомпозиция SAT-задачи, кодирующей криптоанализ A5/1, на семейство подзадач проводится по 31 переменной. Таким образом, для решения одного примера криптоанализа требуется решить, в худшем случае, 2^{31} (то есть более 2 миллиардов) относительно простых подзадач, которые объединяются в пакеты по 32 768 подзадач в каждом. На обработку каждого пакета обычно требуется не более четырех часов работы одного ядра среднего по мощности процессора. На данный момент в SAT@home успешно решены 7 (из 10) примеров криптоанализа A5/1. Эти результаты можно найти в разделе “Найденные решения” сайта проекта (sat.isa.ru/pdsat).

В ближайшем будущем предполагается использовать проект SAT@home для поиска различных совершенных комбинаторных структур. В частности, в проект будут загружены задачи поиска пар и троек ортогональных латинских квадратов, гипотезы о существовании которых на текущий момент не подтверждены.

Заключение

Концепция добровольных вычислений идеально подходит для реализации вычислительных экспериментов, привлекающих существенные ресурсы распределенных вычислителей на протяжении длительного времени (месяцы и даже годы). Описанные в работе технологии могут использоваться для организации специализированных проектов, ориентированных на решение крупномасштабных научных задач. Ключевые пункты таких технологий проиллюстрированы на примере двух реально работающих проектов – OPTIMA@home и SAT@home. Отметим,

что проект SAT@home является на текущий момент единственным действующим на территории СНГ проектом добровольных распределенных вычислений, входящим в список активных проектов, составленный разработчиками платформы BOINC (boinc.berkeley.edu/projects.php), куда он был внесен 07.02.2012 г.

Работа выполнена при поддержке РФФИ (гранты № 11-07-00377-а и 10-07-00301-а) и Седьмой рамочной программы Европейского Союза (FP7/2007-2013), грант № 261561 (DEGISCO). Авторы благодарят участников проектов OPTIMA@home и SAT@home за предоставленные вычислительные ресурсы. ☺

Литература

1. Посыпкин М.А. Методы и распределенная программная инфраструктура для численного решения задачи поиска молекулярных кластеров с минимальной энергией // Вестник Нижегородского университета им. Н.И. Лобачевского, 2010, № 1, с. 210–219.
2. Thomas D., Moorby P. The Verilog hardware description language. Springer, 2002, 381 p.
3. Отпущенников И.В., Семенов А.А. Технология трансляции комбинаторных проблем в булевы уравнения // Прикладная дискретная математика. 2011, № 1, с. 96–115.
4. Schulz S., Blochinger W. Parallel SAT Solving on Peer-to-Peer Desktop Grids // Journal Of Grid Computing, 2010, Vol. 8, No. 3, pp. 443–471.
5. Заикин О.С., Семёнов А.А. Технология крупноблочного параллелизма в SAT-задачах // Проблемы управления, 2008, № 1, с. 43–50.
6. Semenov A., Zaikin O., Bepalov D., Posypkin M. Parallel algorithms for SAT in application to inversion problems of some discrete functions. arXiv:1102.3563v1 [cs.DC].
7. Посыпкин М.А., Заикин О.С., Беспалов Д.В., Семёнов А.А. Решение задач криптоанализа поточных шифров в распределенных вычислительных средах // Труды ИСА РАН, 2009, т. 46, с. 119–137.
8. Kacsuk P., Kovacs J., Farkas Z., Marosi A. Cs., Gombás G., Balaton Z. SZTAKI Desktop Grid (SZDG): A Flexible and Scalable Desktop Grid System // Journal Of Grid Computing, 2009, Vol. 7, No. 4, pp. 439–461.
9. Заикин О.С. Реализация процедур прогнозирования трудоемкости параллельного решения SAT-задач // Вестник УГАТУ, 2010, т. 14, № 4, с. 210–220.

Об авторах

О.С. Заикин (zaikin.icc@gmail.com), А.А. Семёнов (biclop.rambler@yandex.ru) – сотрудники Института динамики систем и теории управления СО РАН
М.А. Посыпкин (mposypkin@gmail.com), Н.П. Храпов (nkhrapov@gmail.com) – сотрудники Института системного анализа РАН