

## Мультиагентная технология автоматизации параллельного решения булевых уравнений в распределенной вычислительной среде

И. В. Бычков\*, Г. А. Опарин, В. Г. Богданова, С. А. Горский, А. А. Пашинин

Институт динамики систем и теории управления им. В.М. Матросова СО РАН, Иркутск, Россия

\*Контактный e-mail: idstu@icc.ru

Рассмотрены архитектура, функциональные возможности и технология применения инструментального комплекса РЕБУС 2.0, ориентированного на использование в фундаментальных и прикладных исследованиях при решении задач с помощью высокопроизводительных вычислений в разнообразных областях знаний, где естественным образом возникают дискретные модели в виде систем булевых уравнений (ограничений).

*Ключевые слова:* булевы уравнения и ограничения, комбинаторные задачи, булева выполнимость.

### Введение

В последние годы активно ведутся исследования с привлечением высокопроизводительных вычислений для решения нелинейных систем булевых уравнений и связанной с ними задачи выполнимости булевых ограничений (SAT) — одной из фундаментальных проблем математической логики и теории вычислений. В [1, 2] отмечается, что булевы уравнения естественным образом возникают при исследованиях, имеющих дело с отображениями, области определения и значений которых лежат в пространствах векторов с двоичными компонентами  $B_2^n$  ( $B_2 = \{0, 1\}$ ). Таковы, например, уравнения алгебры логики; соотношения, описывающие бинарные отношения со свойствами типа транзитивности, связности, асимметрии и т. д.; булевы неравенства; задачи, описывающие нахождение оптимальных переключательных схем; задачи проверки специальных свойств булевых функций, таких как линейность, монотонность, самодвойственность, существенная зависимость булевой функции от переменной и т. д. Известно, что многие важные для практических приложений задачи могут быть сформулированы как задачи булевой выполнимости. Высокая вычислительная сложность SAT-задач актуализирует разработку новых эффективных методов и параллельных программных средств их решения.

Решение такого класса задач в распределенной среде позволяет в значительной степени сократить общее время проведения расчетов. Однако, для того чтобы в полной мере использовать ресурсы среды, необходимо научиться управлять этими ресурсами и освоить параллельные и распределенные технологии программирования.

В последние годы активно ведутся исследования по разработке технологий, методов и средств параллельного решения SAT-задач для различных типов архитектур многопроцессорной техники. В зависимости от реализуемого подхода в многообразии современных параллельных SAT-решателей можно выделить два семейства — решатели, использующие кооперацию, и решатели, использующие конкуренцию.

При первом подходе поисковое пространство делится на части, затем организуется поиск параллельно в каждой части пространства отдельным решателем. Этот подход используется в основном в SAT-решателях, предназначенных для запуска на больших параллельных системах, таких как кластеры или сети компьютеров. В качестве примеров первых таких программных разработок следует выделить решатели GridSAT [3], GradSat [4], разработанный авторами распределенный пакет знаний РЕБУС [5, 6], содержащий, наряду со средствами решения задачи в булевых ограничениях, средства построения этих ограничений и, в отличие от [3, 4], использующий независимое выполнение параллельно решаемых в узлах кластера подзадач. Среди более поздних программных разработок можно выделить решатели Pmsat [7], C-sat [8] и SatCiety [9], анализ эффективности которых приводится в [10]. Большинство этих решателей разработано для небольших вычислительных кластеров (до 64 процессоров). Решатель GridSAT может работать на большой гетерогенной сети компьютеров, содержащих сотни узлов, но приведенные в [3] результаты решения трудных на момент написания этой работы SAT-задач в настоящее время могут быть решены, как подчеркивается в работе [10], с помощью современных последовательных SAT-решателей, например minisat [11] и Lingeling [12], за несколько минут. Однако эти современные SAT-решатели не справляются с новыми SAT-задачами, размерность которых существенно повысилась.

При втором подходе несколько SAT-решателей параллельно работают с одной и той же формулой, но с различными установками параметров и обменом набором конфликтных дизъюнктов, сформированным в результате анализа попадания в тупиковые ситуации. В качестве примеров таких решателей можно привести ManySat [13] и более поздние решатели Plingeling [12] и Treengeling [12]. Однако возникает вопрос об их масштабируемости на большом числе процессорных ядер.

В булевых ограничениях существует еще один аспект программирования. Известно, что с точки зрения человеческого восприятия булевы системы имеют сложную специфическую природу. В связи с этим актуальной, практически значимой и весьма нетривиальной проблемой остается построение средств, позволяющих человеку моделировать задачи на языке булевых ограничений.

В силу вышесказанного представляется актуальным создание инструментальных средств, обеспечивающих автоматизацию всех этапов решения рассматриваемого класса задач. К таким этапам, наряду с построением булевой модели, параллельным решением полученной системы булевых уравнений и обработкой результатов решения, относится также этап обеспечения доступа предметных специалистов к высокопроизводительным вычислительным ресурсам. Кроме того, такой инструментарий должен предоставлять возможность использования этих ресурсов без необходимости углубленного знания вычислительных архитектур и низкоуровневых средств разработки параллельных приложений.

При создании такого инструментария прослеживаются тенденции, с одной стороны, использования сервис-ориентированного подхода, подразумевающего организацию сервисов доступа к вычислительным ресурсам посредством сети Интернет, с другой — применения мультиагентных технологий для управления вычислениями.

Новизна рассматриваемой в статье технологии автоматизации параллельного решения булевых уравнений с помощью инструментального комплекса (ИК) РЕБУС 2.0, представляющего развитие инструментальных средств [14–16], заключается в совместном использовании мультиагентного и сервис-ориентированного подходов для организации высокопроизводительных вычислений при решении этих задач в распределенной вычислительной среде (РВС) [17, 18]. Эта технология обеспечивается с помощью разработанной авторами мультиагентной системы (МАС) HrcSoMas [19], предоставляющей возможности управления распределенным решением SAT-задачи на уровне РВС, гибридного SAT-решателя hrcsat [20, 21] и набора сервисов, предназначенных для решения SAT-задач на вычислительных кластерах с SMP-узлами на основе совместного применения технологий MPI и OpenMP.

## 1. Постановка задачи

Формально задача решения булевых уравнений формулируется следующим образом [3]: пусть  $X(x_1, \dots, x_n)$  — упорядоченное множество булевых переменных ( $x_i \in B_2$  для всех  $i = \overline{1, n}$ ;  $B_2 = \{0, 1\}$ ). Задано  $m$  булевых соотношений (система уравнений) вида

$$f_i(x_{i_1}, \dots, x_{i_{k(i)}}) = 0, \quad (1)$$

где  $f_i (i = \overline{1, m})$  — произвольные булевы функции своих аргументов. Требуется найти одно, несколько или все решения системы (1) или установить ее несовместность.

Следует заметить, что первая NP-полная задача — задача “выполнимости конъюнктивной нормальной формы (КНФ)” (задача булевой выполнимости или SAT-задача) есть, по существу, задача распознавания совместности систем уравнений [22]

$$x_{i_1}^{\sigma_{i_1}} \vee \dots \vee x_{i_{k(i)}}^{\sigma_{i_{k(i)}}} = 1, \quad i = 1, \dots, m, \quad (2)$$

где переменная

$$x_{i_p}^{\sigma_{i_p}} = \begin{cases} \bar{x}_{i_p} & \text{при } \sigma_{i_p} = 0, \\ x_{i_p} & \text{при } \sigma_{i_p} = 1, \end{cases} \quad p = \overline{1, k(i)}.$$

Двойственная к (2) система уравнений вида (1) записывается следующим образом:

$$\bar{x}_{i_1}^{\sigma_{i_1}} \wedge \dots \wedge \bar{x}_{i_{k(i)}}^{\sigma_{i_{k(i)}}} = 0, \quad i = 1, \dots, m. \quad (3)$$

Известно, что любая система вида (1) может быть представлена одним эквивалентным уравнением  $f(x_1, \dots, x_n) = \bigvee_{i=1}^m f_i(x_{i_1}, \dots, x_{i_{k(i)}}) = 0$ . Поэтому далее будем рассматривать одно стандартное уравнение  $f(x_1, \dots, x_n) = 0$ .

## 2. Основы автоматизации программирования в булевых ограничениях в ИК РЕБУС 2.0

Подход к спецификации задачи как задачи удовлетворения ограничений требует создания специальных средств представления и обработки знаний. Это обусловлено следующими обстоятельствами. Во-первых, такой подход позволяет избежать процедурных деталей спецификации путем задания модели задачи. Оперировать декларативными

спецификациями намного проще, чем алгоритмическими [23]. Во-вторых, выделение задач удовлетворения булевых ограничений в отдельный класс привело к появлению универсальных алгоритмов их решения, а также к разработке стандартного формата для представления системы ограничений, позволяющих применять к задачам, представленным в этом формате, известные решатели. Новые специальные средства представления и обработки знаний должны обеспечивать совместимость с этими ранее разработанными средствами.

Постановка задачи в виде системы ограничений привносит свою специфику в аппарат знаний. Эта специфика определяется следующими свойствами:

- расслоением знания, возникающим при формализации сложной предметной области. При программировании в булевых ограничениях естественным образом выделяется ряд тесно связанных слоев, представляющих: 1) булевы модели дискретных задач; 2) методы решения систем булевых уравнений, алгоритмы, разработанные на основе этих методов, и программы, реализующие алгоритмы на одном из языков программирования; 3) статистические данные, накапливаемые в результате проведения вычислительных экспериментов, и результаты их обработки;
- присутствием в предметной области как формализованных знаний (имеются в виду булевы модели, методы и алгоритмы их решения), так и экспертных знаний (речь идет о выборе алгоритма и управляющих параметров вычислительного эксперимента);
- разнообразием форм описания булевой модели (формулы алгебры логики, DIMACS-формат и др.);
- высокой размерностью вектора булевых переменных;
- сильной зависимостью вычислительного алгоритма от представления булевой модели (структуры данных, которая ее характеризует).

Со спецификой предметной области тесно связаны требования к организации проведения вычислительного эксперимента, главными из которых являются:

- возможность построения, редактирования булевой модели и преобразования этой модели из одной формы представления в другую;
- генерация программы-вычислителя булевой функции;
- наличие механизмов специфицирования, составляющих программу-решатель модулей;
- наличие гибких управляющих программных структур реконфигурации программ-решателей;
- возможность включения новых алгоритмов, решателей и моделей;
- необходимость проведения быстрых расчетов в диалоговом режиме и ресурсоемких в автономном режиме;
- необходимость проведения многовариантных расчетов и средств их обеспечения;
- наличие инструментальных средств для автоматизации проведения распределенных и параллельных вычислений.

Методологической основой автоматизации проведения вычислительных экспериментов в области решения дискретных задач является подход, базирующийся на знаниях. В связи с этим вопросы представления, накопления, модификации и использования знаний занимают центральное место в технологии, применяемой ИК РЕБУС 2.0.

Технология решения задачи в ИК РЕБУС 2.0 включает четыре этапа обработки информации: моделирование, декомпозицию булевой модели, параллельное вычисление и обработку результатов.

Технология построения булевой модели состоит из таких этапов, как:

- классификация задачи, структуризация множества булевых переменных;
- построение булевой модели с помощью задания системы булевых ограничений над выбранным регулярным или нерегулярным способом подмножеством этого множества;
- представление системы ограничений в одном из форматов входных данных для решателей вычислительного ядра ИК РЕБУС 2.0, включая общепринятый международный формат DIMACS для SAT-решателей.

Для автоматизации построения булевой модели разработана специальная методика, подробно изложенная в [14]. Эта методика предоставляет два различных способа построения булевой модели: синтез булевой модели в виде текста программ на языке программирования (Фортран, С++) и синтез булевых ограничений в формате DIMACS. Технология построения булевых моделей позволяет формализовать постановку задачи на содержательном уровне. Компактная запись булевой модели на языке ЯСФОР [14] повышает эффективность ее переноса на вычислительные узлы.

Для распараллеливания булевой модели наиболее предпочтительным с точки зрения решения уравнения (1) является метод расщепления, позволяющий выполнить декомпозицию уравнения (1) на произвольное число независимых уравнений, объединение решений которых даст множество решений (возможно пустое) исходного уравнения. Полученные независимые уравнения могут решаться параллельно. Анализатор булевой модели выбирает тактику расщепления и схему расщепления, используемую для последующей декомпозиции этой модели.

### 3. Параллельное решение

Вычислительное ядро ИК РЕБУС 2.0 было дополнено (по сравнению с исходной версией [14]) рядом решателей. В данной статье рассматривается разработанный авторами параллельный гибридный решатель `hpcsat` [20, 21], который представляет собой МРІ-приложение на языке С++. Главный процесс приложения организует динамическую очередь подзадач, получаемых в результате декомпозиции исходной булевой модели. В дочернем процессе может выполняться как многопоточное, так и однопоточное приложение. В качестве такого приложения может использоваться программа, осуществляющая декомпозицию булевой модели методом расщепления, либо SAT-решатель (собственный или сторонней разработки). При использовании последовательного решателя в `hpcsat` реализуется одноуровневый параллелизм по данным, при использовании многопоточного — двухуровневый параллелизм на основе гибридного подхода. Для распараллеливания на уровне узла кластера применяется технология работы с общей памятью.

Для представления узла дерева разработан ряд методов, в том числе рекурсивных, для работы с деревом. С помощью этих методов диспетчер подзадач осуществляет управление состоянием дерева. Следует отметить, что ветви дерева, для которых в процессе расщепления на этапе упрощения остаточной функции получено значение UNSAT, отсекаются. Узел дерева становится готовым для формирования подзадачи базовому решателю, когда количество переменных остаточной функции принимает граничные значения, заданные в начале работы `hpcsat`. Однако в процессе работы может возникнуть ситуация, когда из очереди первыми попадают на выполнение подзадачи, не имеющие решения. Поэтому в разработанном решателе применены механизмы квантования

времени и отложенных подзадач. Под квантованием времени понимается прерывание процесса выполнения базового решателя при достижении определенного лимита по времени. Более подробное описание `hpcsat`, включая описание алгоритма работы и оценку эффективности выполнения на кластере с SMP-узлами, приведено в [20].

Для организации трехуровневого параллелизма, под которым в данной статье понимается дополнение двухуровневого параллелизма, используемого в `hpcsat`, запуском нескольких копий этого решателя в распределенной кластерной среде, применяется мультиагентное управление, организованное с помощью инструментальных средств `HpcSoMas`.

#### 4. Мультиагентное управление параллельным решением

Мультиагентное управление параллельным решением системы булевых уравнений выполняется с организацией распределения ресурсов на уровне приложения пользователя. Структурная схема системы управления вычислениями и аспекты ее применения для такого распределения ресурсов приведены в [18]. Управление осуществляется виртуальным сообществом агентов, создаваемым предварительно с помощью инструментария `HpcSoMas framework` [19]. Основное назначение агентов виртуального сообщества — обеспечение выбора наименее загруженного ресурса (кластера) для запуска на нем параллельного приложения, мониторинг его выполнения и передача результатов решения пользователю.

Виртуальное сообщество агентов включает пользовательский агент, агенты классификации и планирования, агент-менеджер и динамически меняющийся набор локальных агентов. Первые три агента осуществляют прием и классификацию запроса пользователя, построение плана вычислений. Сформированное в результате совместных действий агентов задание передается агенту-менеджеру, который, взаимодействуя с агентом мониторинга ресурсов, распределяет задание по локальным агентам вычислительного кластера (ВК). Агент-менеджер отвечает также за выполнение следующих функций: разбиение задачи на подзадачи, автоматический перезапуск задач с новыми параметрами (для определенного класса задач); управление, мониторинг выполнения задания пользователя. Локальные агенты выполняют отправку заданий менеджеру ресурсов ВК, анализ текущего состояния ВК, передачу результатов агенту-менеджеру.

Для распределения задач по локальным агентам применяется модель тендера, где в качестве лотов выступают вычислительные работы, а в качестве участников — представители вычислительных ресурсов, претендующие на выполнение этих работ. Использование аукциона Викри для реализации этой модели, рассмотренное подробно в [17], позволяет достичь согласованного устойчивого состояния участников аукциона по окончании торгов, в также обеспечить более сбалансированное распределение заданий локальных пользователей кластеров.

#### 5. Аспекты программной реализации `HpcSoMas framework`

Инструментарий `HpcSoMas` разработан на основе сервис-ориентированной технологии, применение которой для реализации агентов в виде web-сервисов рассмотрено в [24]. В виртуальном сообществе агентов, создаваемых с помощью этого инструментария, можно выделить три уровня иерархии: уровень пользовательских агентов, представлен-

ный клиентами в браузере; уровень агентов-менеджеров (сервисы управления); уровень предметно-ориентированных реактивных агентов выполнения заданий (вычислительные сервисы) и системных реактивных агентов (системные сервисы). Многоуровневая архитектура позволяет реализовывать функциональные части виртуального сообщества по отдельности и при необходимости повторно использовать их в других приложениях. Все основные предметно-ориентированные функции агента-менеджера вынесены в отдельные вычислительные сервисы. Системные сервисы могут выполнять следующие функции: расчет ставок на выполнение задания, уведомление пользователей, постановку задания в очередь системы управления заданиями (СУПЗ) вычислительного кластера, получение информации о состоянии ВК, расчет оценки состояния ВК, аутентификацию пользователей и др.

SAT-решатель также представлен как вычислительный сервис с помощью разработанных авторами средств ABCSW (Agents Basic Classes Service Wizard), предназначенных для автоматизации построения вычислительного сервиса на основе приложения пользователя. Эти средства являются подсистемой HrcSoMas и ориентированы на профессиональных программистов и специалистов-предметников. В состав ABCSW входят: набор библиотек (API), графический интерфейс пользователя (GUI), формат JFSOMAS для описания сервиса, описания web-интерфейса сервиса и спецификаций настроек сервиса, разработанный на основе языка JSON [25], и конвертор, считывающий этот формат при создании сервиса.

## 6. Работа с сервисом решения SAT-задачи с помощью hrcsat

С применением инструментария ABCSW на основе решателя hrcsat создан сервис для решения SAT-задач. Для запуска сервиса пользователь может подключаться к агенту-менеджеру через web-интерфейс пользовательского агента, доступный для компьютеров и мобильных устройств, подключенных к сети Интернет, при наличии соответствующей учетной записи пользователя на вычислительном кластере.

С помощью web-интерфейса пользователь формулирует постановку задачи (передает файл с исходной КНФ) и инициирует процесс решения задачи. На основе полученных исходных данных пользовательский агент генерирует множество подзадач и передает их агенту планирования вычислений, который, в свою очередь, формирует задание для каждой подзадачи. Сформированный пул заданий направляется агенту-менеджеру. Агент-менеджер, получив задание, при помощи агента мониторинга состояния системы, параметров, заданных пользователем, а также полученных из самого задания, оценивает дальнейшие возможности по его обработке, отправляя запросы другим агентам-менеджерам для проведения тендера. Агенты взаимодействуют на основе координированного сотрудничества, достигаемого в процессе торгов за выполнение задания, которое не смог самостоятельно выполнить получивший его агент. Агент, получивший задание, ставит его в очередь СУПЗ ВК. На этапе выполнения задания запускается решатель hrcsat. Агенты мониторинга отслеживают состояние каждого задания. В случае появления в РВС дополнительных свободных вычислительных ресурсов и пустого пула заданий у агента-менеджера процесс вычислений приостанавливается, одна из активных подзадач снимается с решения, для нее производится дальнейшая декомпозиция, после чего формируются дополнительные задания и процесс вычислений возобновляется. Если решение найдено, задания снимаются с решения или удаляются из очереди (в зависимости от текущего статуса). Если в результате выполнения всех заданий ре-

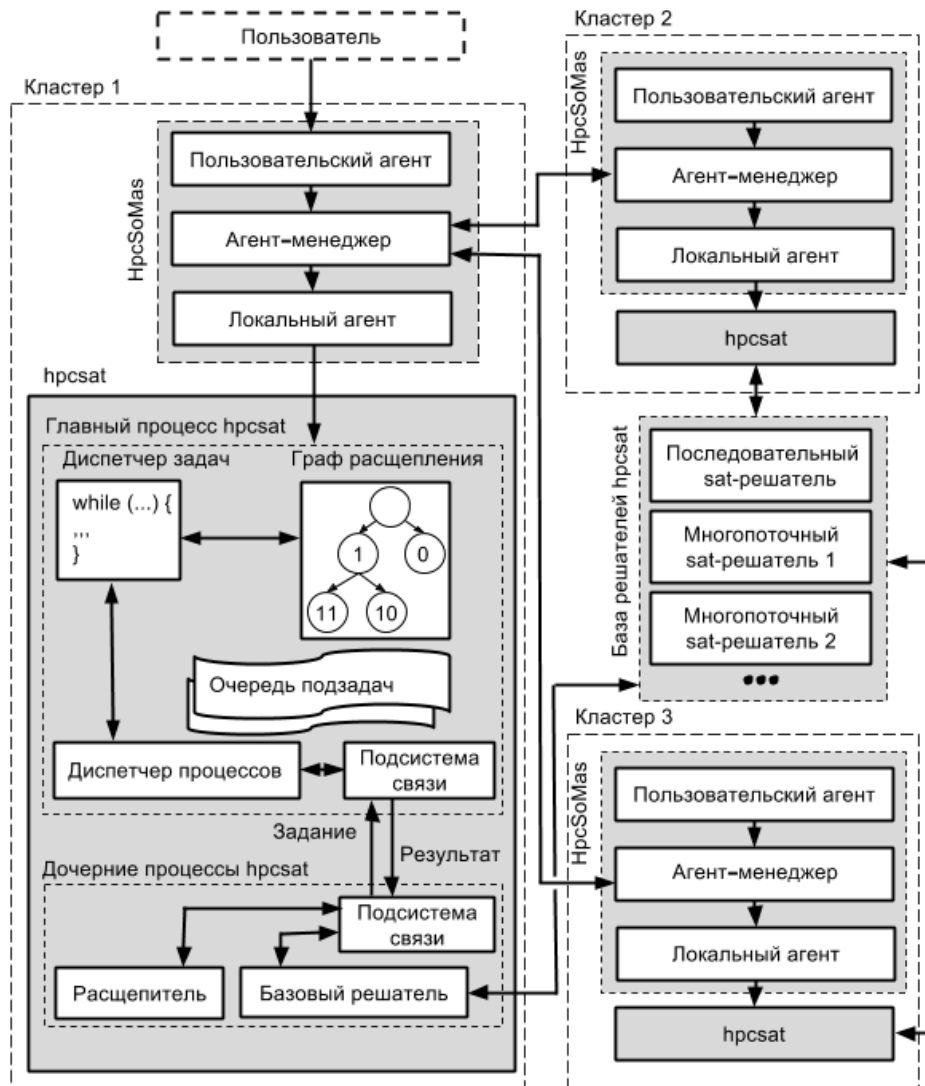


Рис. 1. Схема организации работы MAC HrcSoMas и hpcsat

шение не найдено, пользователь получает уведомление о завершении процесса решения с ответом UNSAT (функция невыполнима). Схема работы решателя hpcsat под управлением виртуального сообщества агентов, созданных с помощью HrcSoMas, показана на рис. 1.

## 7. Тестовый пример

Эксперименты для оценки эффективности разработанных инструментальных средств выполнялись поэтапно. Сначала с помощью ABCSW были разработаны и реализованы сервисы для создания булевой модели в формате DIMACS и решения SAT-задачи. Затем был подобран SAT-решатель HordeSat [10], разработанный для вычислительных кластеров с SMP-узлами, исходный текст которого предоставлен в открытом доступе. Для проведения адекватного сравнения времени выполнения оба решателя, HordeSat и hpcsat, были запущены на одном и том же кластере “Академик В.М. Матросов” [26] на ряде тестовых SAT-задач.



Характеристики тестовых задач

Номер теста	Название	Количество задач	Количество переменных	Выполнимость
1	sat340	15	340	sat
2	sat380	14	380	sat
3	knight11	1	14641	sat
4	knight12	1	20736	sat
5	hole12	1	132	unsat
6	hole13	1	156	unsat

Для тестов выбраны три вида SAT-задач: два блока выполнимых 3SAT-задач [27], задача о ходе шахматного коня (незамкнутое движение) и невыполнимая задача о голубях и клетках (см. таблицу), сформулированные в терминах задачи булевой выполнимости с помощью сервиса построения булевой модели. Целью эксперимента было сравнение результатов решения тестовых задач под управлением параллельным решением с помощью рассматриваемых в статье инструментальных средств HrcSoMas и под управлением стандартизованного менеджера заданий ВК, на котором производился удаленный запуск решателя HordeSat [10] с помощью SSH-клиента PuTTY.

Результаты решения выполнимых SAT-задач небольшой размерности (тесты 1 и 2), большой размерности (тесты 3 и 4) и невыполнимых SAT-задач (тесты 5 и 6) представлены соответственно на рис. 2–4. На горизонтальной оси приводится количество ядер  $\sigma$ , выделенное решателям hpcsat и HordeSat при решении SAT-задачи. Вертикальная ось представляет собой шкалу времени  $t$ . Над маркерами показано время в секундах, потребовавшееся для решения SAT-задачи. При достижении значения времени 48 000 с задача снималась со счета. Эта ситуация возникла при решении невыполнимых SAT-задач для решателя HordeSat (значение 48 000 выделено на рис. 4 жирным шрифтом с подчеркиванием). При выполнении первых двух тестов на графике (рис. 2) отмечалось среднее время решения всех задач, входящих в тест.

В результате анализа данных, приведенных на рис. 2–4, можно отметить, что в большинстве случаев время работы решателя hpcsat значительно меньше, чем решателя HordeSat. Тест для задачи hole13 решателем HordeSat не был завершён в рамках вы-

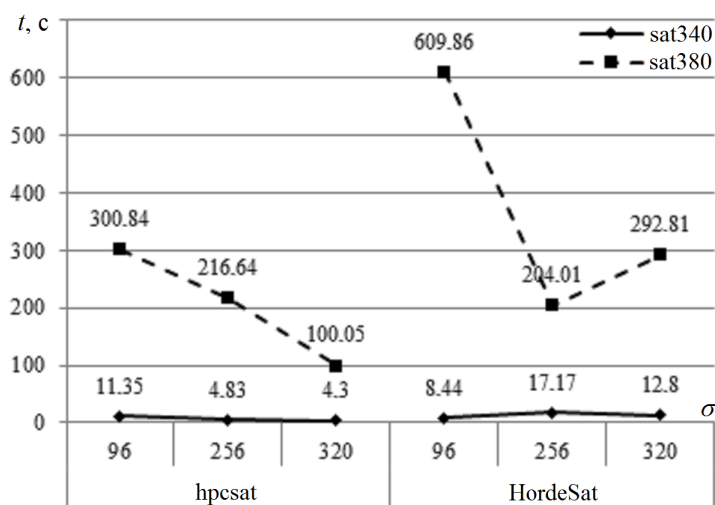


Рис. 2. Результаты выполнения тестов для 3SAT-задач

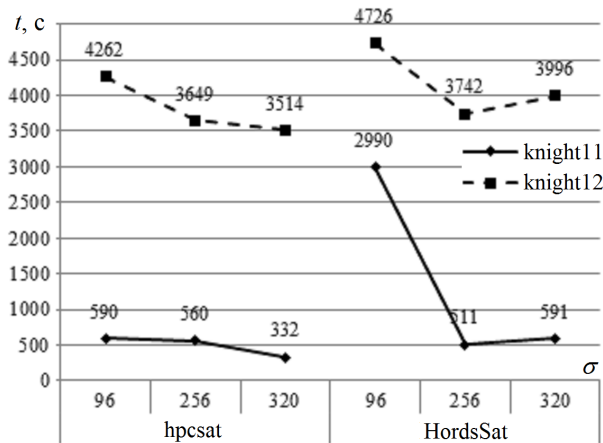


Рис. 3. Результаты выполнения тестов для задачи о ходе шахматного коня

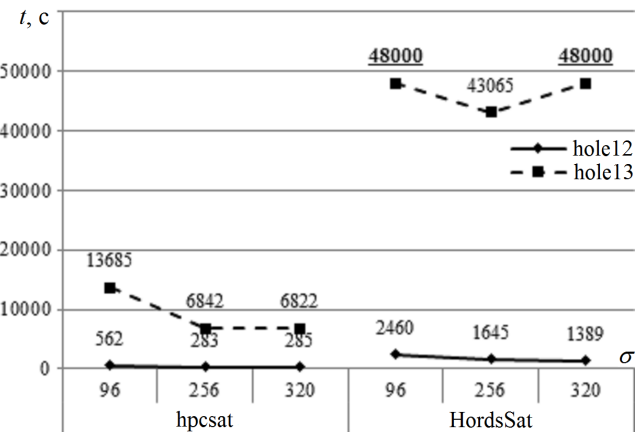


Рис. 4. Результаты выполнения тестов для задачи о голубях и клетках

бранного лимита времени. Для всех трех видов тестовых задач характерно стабильное сокращение времени выполнения у hpcsat с увеличением числа ядер (график зависимости времени решения от количества ядер на рис. 2–4 для решателя hpcsat монотонно убывает).

## Заключение

Рассмотрен сервис-ориентированный подход к организации проблемно-ориентированных распределенных вычислений в РВС. В рамках данного подхода выработана общая концепция реализации многоуровневого параллелизма в РВС, для решения булевых уравнений разработан SAT-решатель на основе гибридного метода полного поиска с интеллектуальным возвратом, сочетающий технологии MPI и OpenMP. Разработаны мультиагентные средства управления вычислениями, новые высокоуровневые инструментальные средства построения интерфейсов сервис-ориентированных приложений. Предложенный подход позволяет существенно сократить время решения задачи под управлением мультиагентной системы по сравнению со стандартизированными менеджерами ресурсов за счет средств динамической декомпозиции заданий и возможности миграции заданий по узлам вычислительной среды.

**Благодарности.** Работа выполнена при финансовой поддержке РФФИ (грант № 15-29-07955-офи-м).

## Список литературы / References

- [1] Левченков В.С. Булевы уравнения. М.: ДИАЛОГ МГУ, 1999. 72 с.  
Levchenkov, V.S. Boolean equations. Moscow: Dialog MGU, 1999. 72 p. (In Russ.)
- [2] Бохманн Д., Постхофф Х. Двоичные динамические системы. М.: Энергоатомиздат, 1986. 401 с.  
Bohmann, D., Posthoff, H. Binary dynamical systems. Moscow: Enegoatomizdat, 1986. 401 p. (In Russ.)
- [3] Chrabakh, W., Wolski, R. GridSAT: A chaff-based distributed SAT solver for the grid // Proc. of the 2003 ACM/IEEE Conf. on Supercomp. Washington: IEEE Comp. Soc., 2003. P. 37.

- [4] **Chrabakh, W., Wolski, R.** GrADSAT: A Parallel SAT Solver for the Grid. UCSB Computer Science Technical Report. No. 2003-05. USA: University of California Santa Barbara, 2003. 9 p.
- [5] **Опарин Г.А., Феоктистов А.Г., Богданова В.Г., Новопашин А.П.** Решение булевых уравнений большой размерности в распределенной вычислительной среде // Тр. Междунар. конф. “Распределенные вычисления и Грид-технологии в науке и образовании”. Дубна: ОИЯИ, 2004. С. 164–169.  
**Oparin, G.A., Feoktistov, A.G., Bogdanova, V.G., Novopashin, A.P.** The solution of high dimensionality Boolean equations in distributed computing environment // Proc. of Distributed Computing and Grid-technologies in Science and Education. Dubna: OIYaI, 2004. P. 164–169. (In Russ.)
- [6] **Опарин Г.А., Феоктистов А.Г., Богданова В.Г., Новопашин А.П.** Инструментальная среда булева моделирования и решения булевых уравнений в сети Интернет // Тр. Междунар. конф. “Распределенные вычисления и Грид-технологии в науке и образовании”, Дубна, июнь, 2006. Дубна: ОИЯИ, 2006. Д11–2006–167. С. 369–378.  
**Oparin, G.A., Feoktistov, A.G., Bogdanova, V.G., Novopashin, A.P.** The toolkit for boolean modelling and solving Boolean equations in the internet // Proc. of Distributed Computing and Grid-technologies in Science and Education. Dubna: OIYaI, 2006. P. 369–378. (In Russ.)
- [7] **Gil, L., Flores, P., Silveira, L.M.** PMSat: A parallel version of Minisat // J. on Satisfiability, Boolean Modeling and Computation. 2008. Vol. 6. P. 71–98.
- [8] **Ohmura, K., Ueda, K.** c-sat: A parallel SAT solver for clusters // Theory and Applications of Satisfiability Testing-SAT, ser. LNCS. Vol 5584. Berlin; Heidelberg: Springer, 2009. P. 524–537.
- [9] **Schulz, S., Blochinger, W.** Parallel SAT solving on peer-to-peer desktop grids // J. of Grid Computing. 2010. Vol. 8, No. 3. P. 443–471.
- [10] **Balyo, T., Sanders, P., Sinz, C.** HordeSat: A massively parallel portfolio SAT solver. Theory and Applications of Satisfiability Testing — SAT 2015: Lecture Notes in Computer Science // Proc. of the 18th Intern. Conf. Austin, TX, USA, September 24–27, 2015. Vol. 9340. P. 156–172.
- [11] **Een, N., Sorensson, N.** MiniSat solver. Available at: <http://minisat.se/> (accessed 12.12.2015).
- [12] **Armin Biere.** Lingeling, Plingeling, PicoSAT and PrecoSAT at SAT Race 2010. Technical Report 10/1, August 2010, FMV Reports Series, Institute for Formal Models and Verification, Johannes Kepler University, Altenbergerstr. 69, 4040 Linz, Austria, 2010. 4 p.
- [13] **Hamadi, Y., Jabbour, S., Sais, L.** ManySAT: A parallel SAT solver // J. on Satisfiability, Boolean Modeling and Computation. 2009. Vol. 6. P. 245–262.
- [14] **Опарин Г.А., Богданова В.Г.** Ребус — интеллектуальный решатель комбинаторных задач в булевых ограничениях // Вест. НГУ. Информационные технологии. 2008. Т. 6, № 1. С. 61–69.  
**Oparin, G.A., Bogdanova, V.G.** Rebus — intellectual solver for combinatorial problems in Boolean constraints // NSU J. of Inform. Technologies. 2008. Vol. 6, No. 1. P. 61–69. (In Russ.)
- [15] **Опарин Г.А., Богданова В.Г., Макеева Н.Г.** Инструментальная среда параллельного решения систем булевых уравнений // Современные технологии. Системный анализ. Моделирование. 2009. № 3. С. 62–68.  
**Oparin, G.A., Bogdanova, V.G., Makeeva, N.G.** Toolkit for the the parallel solving of systems Boolean equations // Modern Technologies. System Analysis. Modeling. 2009. No. 3. P. 62–68. (In Russ.)
- [16] **Опарин Г.А., Богданова В.Г.** Инструментальные средства автоматизации параллельного решения булевых уравнений на многоядерных процессорах // Программные продукты и системы. 2012. № 1. С. 10–14.

- Oparin, G.A., Bogdanova, V.G.** Parallel computing toolkit for solving Boolean Equations on multi-core processors // *Programmnye Produkty i Sistemy*. 2012. No. 1. P. 10–14. (In Russ.)
- [17] **Bogdanova, V.G., Bychkov, I.V., Korsukov, A.S., Oparin, G.A., Feoktistov, A.G.** Multiagent approach to controlling distributed computing in a cluster grid system // *J. of Computer and Systems Sci. Intern.* 2014. Vol. 53, No. 5. P. 713–722.
- [18] **Bychkov, I.V., Oparin, G.A., Feoktistov, A.G., Bogdanova, V.G., Pashinin, A.A.** Service-oriented multiagent control of distributed computations // *Automation and Remote Control*. 2015. Vol. 76, No. 11. P. 2000–2010.
- [19] **Бычков И.В., Опарин Г.А., Феоктистов А.Г., Богданова В.Г., Пашинин А.А.** Мультиагентные методы и инструментальные средства управления в сервис-ориентированной распределенной вычислительной среде // *Тр. Ин-та системного программирования РАН*. 2014. Т. 26, вып. 5. С. 65–82.
- Bychkov, I.V., Oparin, G.A., Feoktistov, A.G., Bogdanova, V.G., Pashinin, A.A.** Multiagent methods and tools of management in a service-oriented distributed computing environment // *Proc. of the Institute for System Programming of the RAS*. 2005. Vol. 26, No. 5. P. 65–82. (In Russ.)
- [20] **Богданова В.Г., Горский С.А.** Технология параллельного решения систем булевых уравнений на вычислительном кластере // *Современные технологии. Системный анализ. Моделирование*. 2013. № 1(37). С. 54–60.
- Bogdanova, V.G., Gorski, S.A.** The technology of parallel solution of nonlinear systems of boolean equations on a computing cluster // *Modern Technologies. System Analysis. Modeling*. 2013. No. 1(37). P. 54–60. (In Russ.)
- [21] **Богданова В.Г., Горский С.А., Пашинин А.А.** Сервис-ориентированные инструментальные средства для решения задач булевой выполнимости // *Фундаментальные исследования*. 2015. № 2, ч. 6. С. 1151–1156.
- Bogdanova, V.G., Gorskiy, S.A., Pashinin, A.A.** Service-oriented tools for solving of Boolean satisfiability problem // *Fundamental Research*. 2015. No. 2, pt 6. P. 1151–1156. (In Russ.)
- [22] **Горшков С.П.** Применение теории NP-полных задач для оценки сложности решения систем булевых уравнений // *Обозрение прикладной и промышленной математики. Дискретная математика*. 1995. Т. 2, вып. 3. С. 325–398.
- Gorshkov, S.P.** Application of the theory of NP-complete problems for estimation of the complexity for solutions of systems of Boolean equations // *Review of Applied and Industrial Mathematics. Discrete Math.* 1995. Vol. 2, No. 3. P. 325–398. (In Russ.)
- [23] **Ушаков Д.М., Телерман В.В.** Системы программирования в ограничениях // *Системная информатика: Сб. науч. тр. Вып. 7: Проблемы теории и методологии создания параллельных и распределенных систем*. Новосибирск: Наука, 2000. С. 275–310.
- Ushakov, D.M., Telerman, V.V.** Systems of constraint programming // *System Informatics: Coll. of Scientific Papers. Vol. 7: Problems of the Theory and Methodology for Development of Parallel and Distributed Systems*. Novosibirsk: Nauka, 2000. P. 275–310. (In Russ.)
- [24] **Бычков И.В., Опарин Г.А., Феоктистов А.Г., Богданова В.Г., Корсуков А.С.** Сервис-ориентированный подход к организации распределенных вычислений с помощью инструментального комплекса DISCENT // *Информ. технологии и вычисл. системы*. 2014. № 2. С. 7–15.
- Bychkov, I.V., Oparin, G.A., Feoktistov, A.G., Bogdanova, V.G., Korsukov, A.S.** The service-oriented approach to distributed computing on the basis of the toolkit DISCENT // *Inform. Technologies and Computer Systems*. 2014. No. 2. P. 7–15. (In Russ.)
- [25] JavaScript object notation format. Available at: <http://www.json.org/> (accessed 12.12.2015).

- [26] Иркутский суперкомпьютерный центр Сибирского отделения РАН. Адрес доступа: <http://hpc.icc.ru> (дата обращения 12.12.2015).  
Irkutsk supercomputer center of the Siberian Branch of the Russian Academy of Sciences. Available at: <http://hpc.icc.ru> (accessed 12.12.2015). (In Russ.)
- [27] Karlsruhe Institute of Technology. HordeSat: A massively parallel portfolio SAT solver. Available at: <http://baldur.iti.kit.edu/hordesat> (accessed 12.12.2015).

*Поступила в редакцию 11 января 2016 г.*

### **A multiagent technology of automation for parallel solution of Boolean equations in a distributed computing environment**

BYCHKOV, IGOR V.\*, OPARIN, GENNADIY A., BOGDANOVA, VERA G., GORSKY, SERGEY A., PASHININ, ANTON A.

Matrosov Institute for System Dynamics and Control Theory of SB RAS, Irkutsk, 664033, Russia

\*Corresponding author: Bychkov, Igor V., e-mail: [dstu@icc.ru](mailto:dstu@icc.ru)

Many practically important combinatorial problems can be reduced to the problem of determining the satisfiability of Boolean formulas (SAT). In recent years, there is a tendency to widespread the use of parallel technology for solving such problems in a distributed computing environment. The purpose of our research is to development the technology and to build the tools that provide the automation of all steps for solving this class of problems. These steps include the construction of a Boolean model, the parallel approach to solution of the resulting system of Boolean equations. It also include processing of solutions and providing an access to high-performance computing (HPC) resources for subject specialists. The special technique was developed to automate the construction of the Boolean model. This technique provides different ways of constructing Boolean model, which include the synthesis of Boolean constraints in DIMACS format. The sat-solver oriented for using a hybrid MPI+OpenMP programming model for HPC systems with clusters of shared memory nodes was designed to solve non-trivial SAT problem. The multi-agent system (MAS) was created for the organization of a three-level parallelism and management of parallel computations using the HpcSoMas framework which was also developed by the authors. This MAC generates a packet of tasks for the problem, allocates resources for this package, manages the execution of the tasks, and informs the user about the status of the task. The author provides experimental tests for a number of SAT instances have been performed using the presented approach and these experiments indicate the usefulness and effectiveness of SAT approach for automation technology.

*Keywords:* Boolean equations and constraints, combinatorial problems, Boolean satisfiability.

**Acknowledgements.** This work was supported by RFBR (grant No. 15-29-07955).

*Received 11 January 2016*