

АГЕНТ-ОРИЕНТИРОВАННЫЕ МОДЕЛИ: МИРОВОЙ ОПЫТ И ТЕХНИЧЕСКИЕ ВОЗМОЖНОСТИ РЕАЛИЗАЦИИ НА СУПЕРКОМПЬЮТЕРАХ

В.Л. Макаров^а, А.Р. Бахтизин^а, Е.Д. Сушко^а,
В.А. Васенин^б, В.А. Борисов^б, В.А. Роганов^б

^а Центральный экономико-математический институт РАН, Москва, Россия

^б Московский государственный университет им. М.В. Ломоносова, Москва, Россия

e-mail: makarov@cemi.rssi.ru, albert.bakhtizin@gmail.com, sushko_e@mail.ru, vasenin@msu.ru, wazzy.msu.@gmail.com, radug-a@ya.ru

Поступила в редакцию 04.08.2015 г.

Масштабные социальные процессы, нередко формирующиеся спонтанно под влиянием непредсказуемых факторов, как это наблюдается сейчас в Европейском союзе в связи с массовым притоком мигрантов, требуют разработки новых инструментов мониторинга и прогнозирования, позволяющих не только оперировать большим количеством данных, но и отражать сложную динамику, определяемую действиями отдельных людей. Таким инструментом является агент-ориентированное моделирование, возможности которого усилены за счёт использования суперкомпьютерных вычислительных ресурсов. В статье рассматривается успешный опыт зарубежных учёных по запуску агент-ориентированных моделей на суперкомпьютерах, а также разработанное к настоящему моменту специализированное программное обеспечение для технической реализации агентных моделей на суперкомпьютерах.

Ключевые слова: агент-ориентированные модели, суперкомпьютерные технологии, параллельные вычисления.

DOI: 10.7868/S0869587316030075

В последние годы в число приоритетных направлений в информационных технологиях уверенно вошли отдельные отрасли гуманитарных наук, в частности, создание систем краткосрочного и долгосрочного предсказательного моделирования социальных явлений и событий с использованием суперкомпьютерных технологий.

По оценкам аналитической компании IDC (International Data Corporation), общий объём данных к 2020 г. достигнет 35000 экзабайт, что означает 29-кратный рост за 10 лет (1200 экзабайт в 2010 г.). Как полагают аналитики, наибольший вклад в этот огромный прирост внесут следующие сферы: интернет (блоги, социальные сети и т.д.); финансы (биржевые индексы, аналитические материалы и т.д.); здравоохранение (данные о пациентах, лекарствах, способах лечения и т.д.); астро-

номия (детализированные изображения галактик); биоинформатика (данные о 3.3 млрд. оснований нуклеиновых кислот в геноме человека, протеиновые последовательности и их анализ); библиотеки (текстовые данные, фото, карты и т.д.).

Необходимость обработки такого объёма данных обуславливает появление аналитических систем нового поколения, включающих усовершенствованные методы вычислений, распознавания образов, организации хранилищ, сбора статистики с целью извлечения смысла из данных и получения информационного контекста. Ставка делается в том числе и на агент-ориентированные модели (АОМ), относящиеся к классу моделей, основанных на индивидуальном поведении агентов и создаваемых для компьютерных симуляций. Основная идея, реализуемая в АОМ, заключается

МАКАРОВ Валерий Леонидович — академик РАН, директор ЦЭМИ РАН. БАХТИЗИН Альберт Рауфович — доктор экономических наук, заведующий лабораторией ЦЭМИ РАН. СУШКО Елена Давидовна — кандидат экономических наук, ведущий научный сотрудник ЦЭМИ РАН. ВАСЕНИН Валерий Александрович — доктор физико-математических наук, заведующий отделом НИИ механики МГУ им. М.В. Ломоносова. БОРИСОВ Василий Александрович — студент механико-математического факультета МГУ им. М.В. Ломоносова. РОГАНОВ Владимир Александрович — старший научный сотрудник НИИ механики МГУ им. М.В. Ломоносова.

в построении вычислительного инструмента, представляющего собой совокупность агентов с определённым набором свойств и позволяющего проводить симуляции реальных явлений.

С помощью АОМ можно смоделировать систему, максимально приближённую к реальности. Появление АОМ следует трактовать как результат эволюции методологии моделирования: переход от мономodelей (одна модель — один алгоритм) к мультимоделям (одна модель — множество независимых алгоритмов). При этом агент в АОМ является автономной сущностью, имеющей, как правило, графическое представление, с определённой целью функционирования и возможностью обучения в процессе существования до уровня, определяемого разработчиками соответствующей модели. В качестве примеров агентов могут рассматриваться: люди (или иные живые организмы), роботы, автомобили и другие подвижные объекты; недвижимые объекты, а также совокупности однотипных объектов. В целом агентами в АОМ могут быть любые наблюдаемые в реальной жизни объекты, однако основной задачей учёта этих объектов в рамках модели является их корректная спецификация. Отметим, что общая особенность и одновременно главное отличие всех АОМ от моделей других классов — наличие в них большого числа взаимодействующих друг с другом агентов. Более подробно про агентные модели можно прочитать в [1].

В силу фактически экспоненциального роста данных обозначается дальнейший тренд развития АОМ — построение АОМ с использованием суперкомпьютерных технологий (в том числе на базе геоинформационных систем). Это направление активно развивается, а на мировых конгрессах, посвящённых АОМ, оно уже давно обсуждается не только на специализированных сессиях, но и в рамках пленарных выступлений.

Актуальность использования суперкомпьютерных технологий для разработки АОМ обусловлена тем, что оперативная память обычного персонального компьютера не способна вместить то количество объектов программной среды, которое соответствует, например, населению земного шара или даже отдельных густонаселённых стран. Запуск оригинальной модели в специализированных средах для разработки АОМ с количеством агентов, превышающим несколько миллионов, уже приводит к исчерпанию оперативной памяти персонального компьютера.

Подобным же образом дела обстоят и с производительностью. Для пересчёта состояния масштабной системы с нетривиальной логикой поведения и взаимодействия агентов требуются значительные вычислительные ресурсы, сопоставимые с потребностями вычислительных методов математической физики с аналогичным количеством расчётных ячеек. Но, в отличие от последних, по-

ведение агентов включает элементы случайности, поэтому требуется провести целую серию расчётов и найти вероятностное распределение ключевых характеристик итогового состояния моделируемой среды.

Названные факторы обуславливают необходимость масштабных экспериментов с использованием суперкомпьютерных версий моделей, в которых популяция агентов распределяется по множеству узлов суперкомпьютера и расчёты выполняются параллельно. При этом возникает задача адаптации разрабатываемых в традиционных программных средах моделей для суперкомпьютеров.

Как и при создании суперкомпьютерных программ для решения многих физических задач, потенциал для распараллеливания многоагентных систем кроется в использовании локальности меж-агентного взаимодействия. В модели, как и в реальной жизни, большая часть взаимодействий происходит между субъектами, находящимися неподалёку. Это позволяет использовать подход распараллеливания “по пространству”, то есть разместить популяцию агентов на узлах суперкомпьютера наиболее равномерно и с учётом близости их географического положения. Таким образом, разбиение территории, на которой проживают агенты, на так называемые кварталы, обеспечивает базовую возможность для распараллеливания задачи. Это наиболее часто встречающийся на практике подход для случаев, когда взаимодействие элементов моделируемой системы, будь то агенты в многоагентных системах или отдельные расчётные ячейки с усреднёнными параметрами моделируемой физической среды, удовлетворяет принципу пространственной локализации: связи и обмен данными имеют место преимущественно для элементов с близкими координатами и отрабатываются практически мгновенно в пределах каждого вычислительного узла.

Обратимся к зарубежным разработкам в области построения АОМ с использованием суперкомпьютерных технологий, преимущественно тем, которые направлены на моделирование социальных систем и процессов, связанных с их функционированием.

ЗАРУБЕЖНЫЙ ОПЫТ РЕАЛИЗАЦИИ АГЕНТНЫХ МОДЕЛЕЙ НА СУПЕРКОМПЬЮТЕРАХ

FuturICT — беспрецедентный междисциплинарный проект по моделированию технологической, социальной и экономической систем мира, стартовавший в 2012 г. и вовлекший учёных практически из всех развитых стран. Срок его реализации составляет 10 лет, а начальное финансирование — 1 млрд. евро. Руководители проекта делают

особый акцент на использовании самых передовых информационных технологий.

Финансовый кризис, международные конфликты, терроризм, распространение заболеваний и киберпреступности, а также демографические, технологические и экологические изменения свидетельствуют о том, что число проблем, с которыми сталкивается человечество, и их острота только нарастают. При этом спрос на научные исследования в этой области явно не удовлетворяется. Поскольку технологические, социальные и экономические системы становятся всё более сложными, используемые в повседневной практике инструменты не позволяют получать реалистичные прогнозы и проводить мониторинг состояния этих систем. Кроме того, одной из наиболее актуальных задач, стоящих перед наукой XXI в., является выявление скрытых законов и процессов, лежащих в основе функционирования и развития сложных систем.

Использование новейших достижений в области естественных и общественных наук позволит разработать систему мониторинга состояния глобальной социально-экономической системы мира, с помощью которой возможно в числе прочего прогнозировать:

- развитие социально-экономических систем (страновой уровень),
- общественные настроения,
- миграционные процессы,
- демографическую ситуацию,
- состояние окружающей среды.

По мнению Д. Хелбинга, одного из руководителей проекта, несмотря на актуальность создания таких многоуровневых систем и существование некоторых их компонентов, институциональные барьеры и нехватка ресурсов тормозят разработку комплексного продукта. Его отсутствие открывает перед FuturICT перспективу стать первым в своём роде.

Организационно проект реализуется как сеть национальных центров, каждый из которых представлен несколькими научными учреждениями в рамках одной страны. Национальные научные сообщества имеют определённую степень автономии, но вместе с тем и набор обязательств. Помимо такой сети, проект предполагает ещё и наличие проблемно-ориентированной сети, включающей головные центры нескольких стран и направленной на решение отдельных научных проблем. Таким образом, FuturICT интегрирует организации на институциональном и проблемном уровнях.

Предлагаемая участниками проекта платформа включает в себя три составляющие:

- Planetary Nervous System (нервная система планеты),

- Living Earth Simulator (симулятор живой планеты),

- Global Participatory Platform (глобальная объединённая платформа).

Набор моделей, формирующих “симулятор живой планеты”, благодаря наблюдательным пунктам позволит обнаруживать кризисы и находить эффективные решения для смягчения их последствий. С целью верификации и калибровки этих моделей будут использоваться данные, собираемые в режиме реального времени с помощью “нервной системы планеты”. Результаты работы двух первых частей FuturICT будут сводиться воедино на “глобальной объединённой платформе”, с которой должны взаимодействовать лица, принимающие решения.

Различного рода наблюдательные пункты (финансовые, экономические, энергетические, транспортные) делятся на четыре группы основных направлений исследований: в области общественных отношений, технологического и экономического развития, а также мониторинга состояния окружающей среды. “Нервную систему планеты” можно представить в виде глобальной сети “сенсоров”, собирающих информацию о социально-экономической, технологической и экологической системах мира. Для её построения координаторы проекта FuturICT тесно сотрудничают с командой С. Пентленда из Массачусетского технологического института (MIT) с целью “подключения сенсоров” к современным гаджетам. В рамках “симулятора живой планеты” предполагается реализовать открытую программную платформу, которая, по мнению инициаторов проекта, будет напоминать магазин приложений для продуктов компании “Apple” — “App Store”: учёные, разработчики и заинтересованные лица смогут загружать собственные и скачивать другие модели, относящиеся к различным районам планеты и тематическим областям. При этом главным будет подход к моделированию, основанный на агент-ориентированной парадигме. Унифицированные модельные компоненты в дальнейшем предполагается реализовывать одновременно с использованием суперкомпьютерных технологий.

“Глобальная объединённая платформа” также будет открытой площадкой для коалиционного обсуждения получаемых с помощью FuturICT прогнозов развития мира — с представителями гражданского общества, власти и бизнес-сообщества.

Крупномасштабная АОМ европейской экономики — EURACE, или Europe ACE (Agent-based Computational Economics — агент-ориентированная вычислительная экономика), отличается очень большим числом автономных агентов, взаимодействующих в рамках социально-экономической системы. Этот проект стартовал в сентябре

2006 г. [2]. В него вовлечены экономисты и программисты из восьми научно-исследовательских центров Италии, Франции, Германии, Великобритании и Турции, а также консультант из Колумбийского университета США нобелевский лауреат Д. Стиглиц.

Методология ACE была положена в основу исследования, с тем чтобы преодолеть ограничения широко распространённых моделей, в которых рассматриваются агрегированные агенты и которые предполагают их рациональное поведение в состоянии равновесия. Для модели используется географическая информационная система, охватывающая широкий перечень объектов — предприятия, магазины, школы, транспортные сети и т.д. Фиксируются три типа агентов: домашние хозяйства (около 10^7), предприятия (около 10^5) и банки (около 10^2). Все они имеют географическую привязку, а также связаны друг с другом посредством социальных сетей, деловых отношений и т.д.

С помощью разработанной модели был проведён ряд экспериментов с целью исследования рынка труда. Основной вывод исследования, по мнению авторов, заключается в том, что макропоказатели двух регионов со схожими условиями (ресурсы, развитие экономики и т.д.) в течение продолжительного периода (10 лет и более) могут значительно разойтись в силу первоначальной неоднородности агентов (www.eugace.org).

Полномасштабная распределённая агентная модель эпидемий. Классические модели распространения эпидемий преимущественно основывались на использовании дифференциальных уравнений, однако такой инструментарий затрудняет учёт связей между отдельными агентами и их многочисленными индивидуальными особенностями. АОМ позволяют преодолеть указанные недостатки. В 1996 г. Д. Эпштейн и Р. Акстелл опубликовали описание одной из первых АОМ, в которой рассматривается процесс распространения эпидемий [3]. Агенты модели отличаются друг от друга восприимчивостью к заболеванию, обусловленной состоянием иммунной системы, и распределены на определённой территории. Число агентов в данной модели составляет всего несколько тысяч, причём их поведение достаточно примитивно.

В дальнейшем под руководством Д. Эпштейна и Д. Паркера в Центре социальной и экономической динамики Брукингского института (Center on Social and Economic Dynamics at Brookings) была построена одна из самых больших АОМ, включающая данные обо всём населении США — US National Model [4]. Данная модель имеет ряд преимуществ. Во-первых, она позволяет предсказывать последствия распространения заболеваний различного типа. Во-вторых, она ориентирована

на поддержку двух сред для вычислений: одна среда состоит из кластеров с установленной 64-битной версией Linux, а другая — из серверов с четырёхъядерными процессорами и установленной системой Windows (поэтому в качестве языка программирования был выбран Java, хотя разработчики и не уточняют, какую именно реализацию Java они использовали). В-третьих, модель способна поддерживать численность агентов от нескольких сотен миллионов до шести миллиардов.

Способ распределения агентов между аппаратными ресурсами включает две фазы: сначала агенты распределяются по компьютерам, задействованным в работе, а затем по потокам на каждом из компьютеров. В процессе работы модели каждый поток может остановиться (в заранее обусловленное время) для передачи сообщений другим потокам. Все подготовленные к отправке сообщения до определённого момента хранятся в пуле сообщений, а затем одновременно отправляются. Кроме того, при реализации модели применяются две вспомогательные утилиты: первая управляет потоками на отдельном компьютере, вторая следит за тем, чтобы все сообщения между потоками были выполнены до момента возобновления вычислительного процесса.

При распределении агентов по аппаратным ресурсам следует учитывать два обстоятельства: 1) производительность узла напрямую зависит от числа инфицированных агентов; 2) контакты, предполагающие передачу сообщений между потоками, требуют гораздо больших вычислительных затрат, чем контакты, ограничивающиеся локальной информацией. Исходя из этого можно, с одной стороны, поделить всё рассматриваемое географическое пространство на равные части, число которых должно соответствовать числу узлов, а затем определить для каждого узла какой-либо географический регион. Такое распределение позволяет сбалансировать вычислительную нагрузку между узлами. С другой стороны, можно закрепить определённую территорию, представляющую собой единую административную единицу, за конкретным узлом. В этом случае вычислительную нагрузку удастся сократить за счёт снижения числа контактов, предполагающих передачу сообщений между потоками. Если первый способ распределения агентов влечёт за собой увеличение вычислительной нагрузки за счёт ресурсоёмких контактов, то второй в ряде случаев чреват значительным дисбалансом между аппаратными ресурсами.

US National Model включает 300 млн. агентов, перемещающихся по карте страны в соответствии с матрицей корреспонденций размером 4000×4000 , специфицированной с помощью гравитационной модели. На рассматриваемой модели был проведён вычислительный экспери-

мент, имитирующий 300-дневный процесс распространения болезни, которая характеризуется 96-часовым периодом инкубации и 48-часовым периодом заражения. В исследовании, в частности, было установлено, что распространение заболевания идёт на спад, после того как 65% заражённых выздоровели и приобрели иммунитет. Эту модель неоднократно использовали специалисты Университета Джонса Хопкинса (Johns Hopkins University), а также Департамента национальной безопасности США в рамках исследований, посвящённых стратегии быстрого реагирования на различного рода эпидемии [5].

В 2009 г. была создана вторая версия US National Model, включающая 6.5 млрд. агентов, спецификация действий которых проводилась с учётом имеющихся статистических данных. С её помощью имитировали последствия распространения вируса гриппа А(H1N1/09) в масштабах всей планеты. Ранее подобная модель была разработана в Лос-Аламосской национальной лаборатории (США), и результаты работы с ней были представлены в широкой печати 10 апреля 2006 г. [6]. Для технической реализации использовался один из мощнейших по тем временам суперкомпьютеров, состоящий из 1024 узлов и имеющий два процессора с частотой 2.4 ГГц и 2 ГБ памяти на каждом. С помощью этой крупномасштабной модели, включающей 281 млн. агентов, были рассмотрены сценарии распространения различных вирусов, в том числе и H5N1, с учётом тех или иных оперативных вмешательств: вакцинации, закрытия школ, введения карантина на некоторых территориях.

Имитация на суперкомпьютере средневековых военных действий. По мнению авторов исследования, являющегося составной частью проекта “Средневековые военные действия на решётке” (Medieval Warfare on the Grid – MWGrid), при описании имевших место исторических событий часто не хватает подтверждающих документов, и возникающие пробелы интерпретируются историками во многих случаях с некоторой долей субъективизма и без убедительных доказательств. В качестве альтернативы авторы исследования, представители Школы информатики Бирмингемского университета (School of Computer Science, The University of Birmingham), Института археологии и античности Бирмингемского университета в Великобритании (Institute of Archaeology and Antiquity University of Birmingham) и исторического факультета Принстонского университета в США (History Department, Princeton University) рассматривают агент-ориентированный подход как инструмент детального воссоздания изучаемых событий.

В статье [7] предлагается использовать агентную модель при исследовании средневековой военной логистики для получения новых сведений,

которые могут дополнить или скорректировать известные исторические факты. Разбираемый в статье пример — поход византийской армии на Манцикерт (1071 г. н.э.). Считается, что в битве при Манцикерте войско императора Романа Диогена IV насчитывало до 100 тыс. человек. Однако, учитывая продовольственные запасы во встречающихся на пути населённых пунктах, возможность в начале прошлого тысячелетия осуществить поход армии подобной численности на обозначенное расстояние вызывает сомнения. Авторы исследования уделяют основное внимание описанию конструкции агентов и их взаимодействий. Для большего правдоподобия армия императора Романа Диогена IV смоделирована в реальном масштабе 1 : 1, кроме того, с такой же детализацией представлены потребляемые ресурсы (скот, зерно и др.).

Поскольку имитация деятельности десятков тысяч отдельных агентов в окружающей среде с большим набором параметров требует значительной вычислительной мощности, было решено в целях использования суперкомпьютера приложить дополнительные усилия по распараллеливанию программного кода. По заверениям разработчиков, проект “Средневековые военные действия” в части возможной имитации исторических процессов аналогичен известной игре-симулятору С. Мейера (S. Meier) “Цивилизация”.

Агентные модели для систем с эксафлопсной производительностью. Исследователи из Аргоннской национальной лаборатории (Argonne National Laboratory, США) работают над созданием агентных моделей нового поколения. По мнению разработчиков, эти модели позволяют достичь существенного прогресса в науке и технике (в том числе в общественных дисциплинах). Они будут вычисляться в рамках систем с эксафлопсной производительностью (10^{18} флоп/с), которые могут прийти на смену существующим суперкомпьютерам петафлопсного уровня (то есть порядка 10^{15} флоп/с) уже в ближайшие годы (2016–2018 гг.).

К настоящему времени учёные разработали и используют крупномасштабную агент-ориентированную модель — комплексную адаптивную систему рынка электроэнергии (Electricity Markets Complex Adaptive Systems — EMCAS) для оценки последствий дерегулирования рынка электроэнергии в штате Иллинойс. Агенты — производители и потребители электроэнергии — подстраивают своё поведение к изменениям рыночных условий максимально приближённо к реальности.

Реализуются ещё три пилотных проекта агентных моделей с эксафлопсной производительностью в области: 1) биоразнообразия микроорганизмов; 2) кибербезопасности; 3) социальных аспектов изменения климата. В рамках третьего

проекта предполагается углублённо рассмотреть экономические и социальные эффекты от изменения состояния окружающей среды. По мнению разработчиков, результаты будут иметь важное значение для долгосрочного прогнозирования климата, поскольку изменения в этой области могут оказать существенное влияние на физические факторы. К примеру, выбор топлива и переход на новые технологии в энергетике, скорее всего, повлияют на уровень концентрации двуокиси углерода в атмосфере, и наоборот.

Если в моделировании физических аспектов климата (атмосферные потоки, океаническая циркуляция и т.д.) исследователи добились ощутимых успехов, то в области моделирования населения планеты на долгосрочный период (когда численность может достичь 10^{10} человек) предстоит большая работа. Предполагается, что в модели у каждого агента будет 10^3 (!) внутренних состояний, кроме того, планируется рассмотреть широкий спектр поведенческих типов (<http://www.dis.anl.gov/exp/cas/index.html> и <http://www.alcf.anl.gov/>).

Высокопроизводительные вычисления агентных моделей с использованием графических процессоров. В последнее время всё чаще запуск агентных моделей осуществляется с использованием графических процессоров (Graphics Processing Unit, GPU) вместо обычных центральных процессоров (Central Processing Unit, CPU).

Ядра CPU предназначены для исполнения одного потока последовательных инструкций с максимальной производительностью, в то время как GPU изначально проектировались для параллельного выполнения большого числа инструкций. Различия между CPU и GPU связаны ещё и с кэшированием (механизм сохранения копий в промежуточном буфере с быстрым доступом), а также со спецификой механизмов доступа к встроенным контроллерам памяти. Помимо этого, частота работы CPU имеет ограничения, связанные с физическими условиями, — работа огромного числа (свыше миллиарда) транзисторов создаёт помехи внутри схемы, а также с высоким энергопотреблением. Сравнение частот различных процессоров компании Intel свидетельствует о том, что в последние несколько лет частота процессоров уже практически не увеличивается, колеблясь в районе 3.4 ГГц.

Ресурс дальнейшего увеличения производительности основные разработчики процессоров — компании Intel и AMD — видят в увеличении числа ядер. Так, в 2011 г. получили распространение 8-ядерные процессоры от компании Intel. Эта же компания ещё в 2009 г. продемонстрировала опытный образец 48-ядерного процессора, представляющего собой одночиповый облачный компьютер (Single-chip Cloud Computer, SCC), кото-

рый в ближайшие годы предполагается совершенствовать путём увеличения числа ядер до 100. Однако в настоящее время в CPU число ядер несоизмеримо меньше, чем в GPU.

Поддержка многопоточности у этих двух типов процессоров существенно различается. Если CPU исполняет один-два потока на ядро, то у GPU работа простая и распараллеленная изначально, и видеочипы поддерживают до 1024 потоков на каждый мультипроцессор, которых в чипе несколько штук. Переключение с одного потока на другой в случае CPU предполагает сотни тактов, а GPU переключает несколько потоков за один такт [8].

Для облегчения вычислений на GPU, поддерживающих технологию GPGPU (General-Purpose Graphics Processing Units), то есть технологию произвольных вычислений на видеокартах, компания NVIDIA, мировой лидер в разработке графических ускорителей и процессоров для них, выпустила программно-аппаратную архитектуру CUDA (Compute Unified Device Architecture). CUDA позволяет эффективно управлять памятью графического ускорителя, организовывать доступ к его набору инструкций и эффективно осуществлять параллельные вычисления. В основе CUDA API лежит C-подобный язык программирования со своим компилятором и библиотеками для вычислений на GPU. На русскоязычном сайте компании NVIDIA (www.nvidia.ru) приведены данные по увеличению производительности программ за счёт переноса расчётов на видеочипы. Усреднённые оценки демонстрируют ускорение по сравнению с универсальными процессорами в 30–50 раз.

Представляет интерес деятельность исследовательской группы из Университета Шеффилда (The University of Sheffield), Великобритания, занимающейся разработкой программного обеспечения для построения крупномасштабных пешеходных АОМ с использованием графических процессоров. Городская среда представляет собой сложную пространственную структуру с большим количеством движущихся объектов, её визуализация требует высокой производительности. Разработанное программное обеспечение позволяет описывать поведение агентов, а также изменение их данных на высоком уровне (в виде написания соответствующих скриптов) и без непосредственной работы с памятью GPU¹.

В настоящее время в распоряжении разработчика решений с использованием CUDA имеется множество инструментов: NVIDIA Parallel Nsight, CUDA SDK, библиотеки CUDA BLAS, CUDA

¹ Соответствующая АОМ представлена на сайте разработчика, а также на сайте компании NVIDIA (в разделе с примерами моделей, использующих программно-аппаратную архитектуру CUDA): <http://www.paulrichmond.staff.shef.ac.uk/index.php>; <http://www.nvidia.com/content/cuda/spotlights/paulrichmond-sheffield.html>

FFT и другие продукты. Кроме того, есть компиляторы с автоматическим распараллеливанием — PGI Fortran, CAPS HMPP C и др. Особенно удобно, что CUDA поддерживает большинство пакетов для символьных вычислений (MATLAB, Mathematica), причём многие технические сложности, к примеру, связанные с управлением памятью GPU, преодолеваются благодаря этим программам в автоматическом режиме.

Резюмируя, отметим, что, с одной стороны, GPU приближаются к CPU, становясь более универсальными за счёт увеличивающихся возможностей по расчётам чисел с двойной и одинарной точностью, а с другой — и в CPU растёт число ядер, соответственно, и способность к параллельным расчётам. Более того, известно, что у крупнейшего производителя CPU компании Intel имеются свои разработки в области GPU (к примеру, процессор Larrabee). То же относится и к компании AMD. Для нас же важно, что на данный момент можно говорить о несомненной эффективности GPU для решения задач, связанных с запуском АОМ.

Приход больших игроков, агент-ориентированное программирование и моделирование мира. В 1990 г. профессор Стэнфордского университета Й. Шохам предложил агент-ориентированный подход в качестве новой парадигмы программирования. Его суть заключается в представлении агента в качестве программного модуля, выполняющего поставленные задачи, а его поведения как зависимого от окружающей среды, которую он воспринимает с помощью датчиков и на которую может воздействовать посредством исполнительных механизмов. По мнению Шохам, агент-ориентированный подход является частным случаем более общей и широко известной парадигмы программирования — объектно-ориентированного программирования [9]. Различие состоит в следующем: если объект в рамках объектно-ориентированной парадигмы определяется набором методов и атрибутов, то агент задаётся описанием его поведения в более широком смысле. Идея агент-ориентированного программирования предполагает, что агент является программной сущностью с определённой степенью автономности и целью выполнения задач для пользователя. При этом агенты рассматриваются как более автономные, чем объекты, их поведение — как отличающееся большей гибкостью и являющееся “более социальным”. Последнее означает, что соответствующий программный код выполняется не автоматически, а по решению самого агента в зависимости от ситуации, задач, текущих приоритетов и параметров среды.

В 2009 г. одна из крупнейших транснациональных компаний по производству программного обеспечения Microsoft (а точнее, DevLabs — подразделение Microsoft, занимающееся экспери-

ментальными и инновационными программными разработками) анонсировала новый язык программирования — Ахит, который был изначально предназначен для написания многопоточных параллельных приложений, оптимизированных для работы на многоядерных процессорах. Важной особенностью нового языка стала реализация в нём агент-ориентированной парадигмы программирования. Таким образом, экспериментальный язык вобрал в себя многие прогрессивные концепции программирования, а основной целью его создания стала проверка совместимости перечисленных выше парадигм в рамках одного языка.

По синтаксису и ряду конструкций Ахит наиболее близок к языку C#, однако они различаются концептуально. Главные концепции Ахит — каналы, схемы, слои, домены и агенты. Первое и самое главное отличие — отсутствие в Ахит классов. Вместо них в этом языке есть новый абстрактный тип — агенты, которые отличаются от классов тем, что на них нельзя ссылаться. Взаимодействие между агентами происходит посредством передачи сообщений в рамках каналов. Агенты близки по смыслу потокам, но при этом они гораздо “легче”, а их количество может составлять сотни, тысячи или даже миллионы в рамках одной программы.

Работа над инновационным программным языком Ахит на данный момент закончена, и в виде самостоятельного продукта он на рынок не выйдет. По мнению разработчиков, этот проект был весьма успешным и апробированные в нём концепции найдут применение в будущих релизах .NET: языки C# и VB будут поддерживать асинхронные режимы выполнения программного кода, а также реализованные в Ахит парадигмы агент-ориентированного подхода. С коммерческой точки зрения, вероятно, действительно эффективнее расширять функционал имеющихся и уже достаточно распространённых языков программирования, чем выводить на рынок абсолютно новые. Тем не менее существенным представляется сам факт того, что крупнейший производитель программного обеспечения взял курс на поддержку агентного подхода.

К проявлениям этой же тенденции относится анонсированное в 2010 г. компанией Microsoft создание *компьютерной модели мира* с использованием в том числе агент-ориентированного подхода, то есть, по сути, создание виртуальной реальности, описывающей текущее состояние социальной и экономической системы всего мира. В промороликах, размещённых на сайте компании, её топ-менеджеры отмечали, что на протяжении всей своей истории человечество стремилось к новым знаниям, в результате чего к сегодняшнему дню собрано огромное количество самых разных данных. Учёные и специалисты в области информационных технологий берутся обрабатывать их в

течение ближайших пяти лет с целью моделирования окружающего мира, причём планируется использовать весь потенциал имеющихся у Microsoft технических наработок. Это позволит, полагают руководители Microsoft, перейти на новый уровень понимания наиболее актуальных проблем, с которыми столкнулось человечество, что является необходимым условием их последующего разрешения. В заявлениях компании, кроме того, подчёркивалось, что наука находится в точке бифуркации: новейшие достижения в области суперкомпьютерных технологий, а также в области технологий интеллектуальной обработки информации открывают перспективы разработки сложных моделей, возможности которых соответствуют тем скоростям, с которыми меняется современный мир. По ожиданиям, такие модели не только позволят найти пути трансформации ключевых отраслей — энергетики и здравоохранения, но окажут огромное влияние на бизнес-сообщество и отрасль инженерии знаний, что может привести к созданию принципиально новых продуктов, предприятий и отраслей промышленности. Несомненный интерес крупнейших игроков ИТ-рынка к агентным моделям лишний раз доказывает перспективность этого инструмента и его большое будущее.

СПЕЦИАЛИЗИРОВАННОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ТЕХНИЧЕСКОЙ РЕАЛИЗАЦИИ АГЕНТНЫХ МОДЕЛЕЙ НА СУПЕРКОМПЬЮТЕРАХ

Для АОМ разработано более 100 программных сред с различными функциональными возможностями, но важно отметить, что все эти продукты реализованы в средах, изначально не предназначенных для распараллеливания программного кода (к примеру, в Eclipse, MS Visual Studio и др.). В последнее время появляются первые специализированные средства для построения агентных моделей непосредственно для суперкомпьютеров, о которых мы расскажем подробнее.

Repast for High Performance Computing (RepastHPC) — первый пример программного обеспечения, разработанного для проектирования АОМ с целью последующего запуска на суперкомпьютерах. Данный пакет реализован с использованием языка C++ и MPI — программного интерфейса для обмена сообщениями между процессами, выполняющими задачу в параллельном режиме, а также для библиотеки Boost, расширяющей C++.

В рамках RepastHPC реализован динамический дискретно-событийный планировщик выполнения программных инструкций с консервативными алгоритмами синхронизации, предусматривающими задержку процессов для соблюдения определённой очередности их выполнения. Агенты

распределяются между процессами, и каждый процесс связан с агентом, являющимся *локальным* по отношению к данному процессу. В свою очередь агент локален к процессу, выполняющему программный код, описывающий поведение данного агента. При этом копии остальных — *нелокальных* — агентов могут присутствовать в любом процессе, что позволяет агентам всей модели взаимодействовать с этими копиями. К примеру, пусть пользователь в своей модели, предполагающей параллельные вычисления, использует два процесса — P1 и P2, каждый из которых создаёт определённое количество агентов и имеет собственный планировщик выполнения программных инструкций. Агенты, поведение которых рассчитывается на процессе P1, являются локальными по отношению к данному процессу, и только в рамках данного процесса программный код может изменить их состояние (аналогично и для процесса P2). Предположим, процесс P1 запрашивает копию агента A2 из процесса P2. Агент A2 не является локальным по отношению к процессу P1, и, соответственно, программный код, выполняемый в рамках процесса P1, не может изменить состояние агента A2. При этом агенты, реализуемые в рамках процесса P1, при необходимости могут запросить состояние агента A2, но копия A2 останется неизменной. Изменение оригинального A2 возможно только в рамках процесса P2, но в этом случае RepastHPC синхронизирует изменения состояния агента между всеми процессами².

Высокопроизводительное программное обеспечение для построения крупномасштабных АОМ Пандора (Pandora) разработано в суперкомпьютерном центре Барселоны (Barcelona Supercomputing Centre). Оно предоставляет полную поддержку геоинформационных систем (ГИС), что важно в тех случаях, когда для функционирования моделей необходима географическая привязка агентов. Результат каждой симуляции хранится в иерархическом формате данных (Hierarchical Data Format, HDF), предназначенном для хранения большого количества цифровой информации. Этот формат поддерживается большинством ГИС. Пандора дополняется программой Кассандра (Cassandra), позволяющей визуализировать процесс выполнения симуляции с использованием 2D и 3D-графики.

К ключевым особенностям Пандоры и Кассандры относятся:

- все возможности C++ для разработки, выполнения и анализа агентных моделей и клеточных автоматов;

² Более подробно с данным программным обеспечением можно ознакомиться в руководстве пользователя [10], а скачать сам пакет можно по адресу http://repast.sourceforge.net/repast_hpc.php

- параллельное выполнение программного кода с использованием интерфейсов MPI/OpenMP;
- высокая масштабируемость, то есть способность увеличивать производительность пропорционально увеличению числа узлов суперкомпьютера;
- поддержка ГИС;
- возможность разработки распределённых агентов, построенных с использованием технологий искусственного интеллекта.

Разработчики предлагают использовать Пандору для целого ряда исследовательских направлений, среди которых:

- социальные явления, анализ социальных сетей и межличностные взаимодействия;
- сложные социально-экономические системы;
- миграционные процессы.

Механизм работы Пандоры выглядит следующим образом. После концептуального определения каркаса создаваемой модели её первая реализация может быть разработана с помощью скриптов языка Python, а полученные результаты затем обработаны с использованием нескольких инструментов анализа. Если для реализации модели потребуются дополнительные аппаратные ресурсы, Пандора может конвертировать программный код в C++ версию, автоматически его распараллелив. Таким образом, от пользователя, по заверению разработчиков, не требуется дополнительных усилий по адаптации программного кода для узлов суперкомпьютера.

Среда функционирования агентов в Пандоре определяется как набор слоёв, содержащих растровые карты, соответствующие стандартам ГИС. В процессе симуляции моделируемая среда, а также населяющие её агенты равномерно распределяются по узлам суперкомпьютера. Такой подход наилучшим образом решает проблему распределения вычислительной нагрузки при моделировании древних сообществ, в которых не было средств телекоммуникации.

Разработанные модели являются хорошо масштабируемыми, хотя есть проблема с синхронизацией действий агентов. В Пандоре она решается путём разделения каждого узла на четыре равные части (0, 1, 2, 3). Программный код всех агентов, обрабатываемых в части 0, выполняется одновременно и без конфликтов, поскольку эта часть узла не является смежной по отношению к аналогичной части другого узла. Полученные в результате выполнения кода результаты далее передаются для последовательного выполнения соседним частям (1, 2, 3). После этого данные всех узлов синхронизируются, и выполнение модели переходит на следующий шаг.

Синхронизация в Пандоре осуществляется с помощью интерфейса MPI. Кроме того, директивы OpenMP используются для распределения вы-

числительной нагрузки по различным процессорам внутри узла. При анализе разных пакетов для разработки агентных моделей было выявлено, что наиболее ресурсоёмкими участками кода являются моменты сбора информации и решения о порядке выполнения программных инструкций на следующем шаге. В Пандоре для одного процессора выполняемый шаг модели состоит из трёх методов. В рамках реализации первого из них (updateKnowledge) агенты собирают информацию, но не могут изменять параметры среды и состояния других агентов. Второй метод (SelectAction) позволяет агентам принимать решения и генерировать последовательность дальнейших действий, но по-прежнему не разрешает проводить какие-либо изменения. Наконец, во время выполнения третьего метода (updateState) агенты меняют своё состояние. В случае использования множества процессоров Пандора равномерно распараллеливает между ними объём вычислений первых двух методов (наиболее ресурсоёмких). Что касается процедур третьего метода, то они осуществляются последовательно, с тем чтобы избежать возможных конфликтов. На первый взгляд, разделение шага на три исполняемых метода кажется усложнением. Однако реализация шага с помощью одного метода требует усилий по корректному определению последовательности выполнения различных процедур в рамках моделируемых сценариев, а разделение шага на три метода позволяет снять эти ограничения [11].

Среда построения агентных моделей ABM++. В АОМ EpiSims, разработанной исследователями из Института биоинформатики Вирджинии (Virginia Bioinformatics Institute, США), рассматриваются как перемещения агентов, так и их контакты в рамках среды, максимально приближенной к реальности и содержащей дороги, здания и прочие инфраструктурные объекты [12]. Для построения модели потребовался большой массив данных, включающий информацию о здоровье отдельных людей, их возрасте, доходе, этнической принадлежности и т.д.

Изначальная цель исследования заключалась в построении для запуска на суперкомпьютере АОМ большой размерности, с помощью которой планировалось изучать распространение болезней. Однако в ходе работы пришлось также решать задачу, связанную с созданием специализированного программного обеспечения ABM++, позволяющего осуществлять разработку АОМ на языке C++, а также содержащего функции, которые облегчают распределение исполняемого программного кода на узлах кластеров суперкомпьютера. Помимо этого, ABM++ предоставляет возможность динамического перераспределения потоков вычислений, а также синхронизации происходящих событий.

ABM++, первая версия которого появилась в 2009 г., представляет собой результат модернизации инструмента, разработанного в 1990–2005 гг. в Лос-Аламосской национальной лаборатории в процессе построения крупномасштабных АОМ (EpiSims, TRANSIMS, MobiCom). Межпроцессорные связи между вычислительными узлами в АОМ часто требуют синхронизации происходящих в модели событий. ABM++ позволяет разрабатывать модели, отвечающие этому требованию. Например, в социальных моделях агенты часто перемещаются между различными точками пространства (работа, дом и т.д.), а на программном уровне этому соответствует смена узла кластера, и здесь важно, чтобы модельное время принимающего узла было синхронизировано со временем узла, который агент только что покинул. В ABM++ также реализована библиотека MPIToolbox, соединяющая интерфейс C++ API (Application Programming Interface) и MPI (Message Passing Interface) суперкомпьютера, благодаря чему ускоряется передача данных между узлами кластеров.

ABM++ создавалось в Ubuntu Linux – операционной системе с компиляторами gcc/g++. В качестве интегрированной среды разработки рекомендуется пакет Eclipse с плагином для поддержки C и C++, а также с плагином RTP (Parallel Tools Platform), обеспечивающим разработку и интеграцию приложений для параллельных компьютерных архитектур. Eclipse поддерживает интеграцию с TAU (Tuning and Analysis Utilities) Performance System – инструментом для разностороннего анализа и отладки программ для параллельных вычислений, что также упрощает разработку агентных моделей.

SWAGES – расширяемая распределённая среда для крупномасштабного агент-ориентированного моделирования – детище учёных из Университета Тафтса (Медфорд, Массачусетс, США). SWAGES предоставляет возможность автоматического распределения программного кода и поддержку нескольких языков программирования, а также подключения плагинов для визуализации, статистического анализа и автоматической обработки ошибок. Результаты вычислений могут быть конвертированы в файлы для последующего использования в общедоступных средствах обработки данных (типа R или Scilab).

Данная среда содержит несколько компонентов клиент-серверной архитектуры, обеспечивающей распределённые вычисления (планировка, распределение, запуск, контроль выполнения, восстановление после сбоев и др.). На сервере определяются наборы экспериментов, отличающиеся начальными состояниями, приоритетом выполнения, а также формируются коллекции выходных данных, результаты дополнительного анализа данных и т.д. Агенты могут быть описаны с использованием любого программного языка,

поддерживаемого Poplog (к примеру, Pop11, Prolog, ML, Scheme, CLisp). Кроме того, для разработчика предусмотрена возможность вызова внешних функций, написанных на других языках программирования, посредством специального интерфейса. Имеются встроенные инструменты для статистического анализа, а также библиотеки для поиска данных (в различных форматах, включая HTML, TeX, простой текст). Разработчики SWAGES позиционируют свой продукт в качестве проработанной среды для построения крупномасштабных АОМ, успешно использованной как минимум в нескольких десятках научно-исследовательских проектов (см.: <http://www.hrilib.org>).

Число специализированных программных продуктов для реализации агентных моделей на суперкомпьютерах постоянно растёт. Помимо описанных выше, следует упомянуть также MUSE, CyberGIS и др. При сложившейся конъюнктуре приход на это поле больших игроков (например, Microsoft) является лишь вопросом времени.

* * *

Возрастающий интерес крупнейших игроков ИТ-рынка (Microsoft, Wolfram, ESRI и др.) к АОМ, несомненно, доказывает перспективность этого инструмента и его большое будущее, а экспоненциальный рост общего объёма данных, связанных с жизнедеятельностью людей, и потребность в аналитических системах получения данных нового поколения, необходимых для прогнозирования социальных процессов, обуславливают применение суперкомпьютерных технологий.

В настоящее время существует несколько международных ассоциаций, которые объединяют исследовательские группы из крупнейших институтов и университетов, работающих в данном направлении. Наиболее известными из них являются: 1) North American Association for Computational Social and Organizational Sciences (NAACSOS); 2) European Social Simulation Association (ESSA); 3) Pacific Asian Association for Agent-Based Approach in Social Systems Science (PAAA). Каждая из перечисленных ассоциаций регулярно проводит конференции по социальному моделированию соответственно в Америке, Европе и Азии. Кроме того, раз в два года проводится мировой конгресс по данной тематике.

В России АОМ получили развитие относительно недавно, однако уже существует интернет-сайт, содержащий информацию о разработках в данной области (www.artsoc.ru). Флагманом этого научного направления в нашей стране является Центральный экономико-математический институт РАН. О содержании и результатах проводимых в его стенах исследований будет рассказано в следующей статье.

Работа выполнена при поддержке РНФ, проект № 14-18-01968.

ЛИТЕРАТУРА

1. Макаров В.Л., Бахмузин А.Р. Социальное моделирование — новый компьютерный прорыв (агент-ориентированные модели). М.: Экономика, 2013.
2. Deissenberg C., Sander van der H., Herbert D. EURACE: A Massively Parallel Agent-Based Model of the European Economy / Document de Travail n°2008-39. 24 June. 2008.
3. Epstein J.M., Axtell R.L. Growing Artificial Societies: Social Science from the Bottom Up. Cambridge: MIT Press, 1996.
4. Parker J. A Flexible, Large-Scale, Distributed Agent Based Epidemic Model // Center on Social and Economic Dynamics. 2007. Working Paper. № 52. http://www.brookings.edu/~media/research/files/papers/2007/12/epidemicmodel-parker/12_epidemicmodel_parker.pdf (дата обращения 04.06.2015).
5. Epstein J.M. Modelling to contain pandemics // Nature. 2009. 460, 687, 6 August 2009. <http://www.nature.com/nature/journal/v460/n7256/full/460687a.html> (дата обращения 04.06.2015).
6. Ambrosiano N. Avian Flu Modeled on Supercomputer // Los Alamos National Laboratory NewsLetter. 2006.
7. Murgatroyd P., Craenen B., Theodoropoulos G. et. al. Modelling medieval military logistics: an agent-based simulation of a Byzantine army on the march // Computational and Mathematical Organization Theory. 2012. V. 18. P. 488–506.
8. Берилло А. NVIDIA CUDA — неграфические вычисления на графических процессорах // <http://www.ixbt.com/video3/cuda-1.shtml> (дата обращения 01.11.2013).
9. Shoham Y. Agent Oriented Programming: Technical Report. Stanford: Computer Science Department, Stanford University, 1990.
10. Collier N. Repast HPC Manual // <http://repast.sourceforge.net/docs.php> (дата обращения 01.11.2013).
11. Wittek P., Rubio-Campillo X. Scalable agent-based modelling with cloud HPC resources for social simulations // IEEE 4th International Conference on Cloud Computing Technology and Science (CloudCom). December 3–6. Taipei, Taiwan, 2012. P. 355–362.
12. Roberts D.J., Simoni D.A., Eubank S. A National Scale Microsimulation of Disease Outbreaks. RTI International Research Triangle Park, NC. Virginia Bioinformatics Institute. Blacksburg, VA, 2007.