

Разработка агент-ориентированной демографической модели России и ее суперкомпьютерная реализация*

В.Л. Макаров, А.Р. Бахтизин, Е.Д. Сушко, Г.Б. Сушко

ФГБУН Центральный экономико-математический институт РАН, Москва

В работе показано применение агент-ориентированного подхода при моделировании естественного движения населения. Представлена демографическая модель России с учетом ее административного деления, в которой на основе моделирования поведения отдельных членов искусственного общества имитируются процессы смертности, рождаемости и миграции. Для моделирования поведения искусственного общества в целом требуется проведение симуляций с числом агентов до 10^9 и использование суперкомпьютерных технологий. Важной задачей в таких расчетах становится оптимальное распределение агентов по процессорам кластера. Показано применение декомпозиции модели с использованием алгоритма METIS с учетом особенностей агентной модели. Показаны результаты апробации модели.

Ключевые слова: агент-ориентированное моделирование, демография, суперкомпьютерные технологии, применение графовой декомпозиции METIS.

1. Введение

Статья посвящена ведущейся в течение ряда лет в ЦЭМИ РАН масштабной работе по созданию методологии разработки больших агент-ориентированных моделей (АОМ), воссоздающих в искусственной среде структуру и особенности функционирования реальных социально-экономических систем, а также технологии реализации подобных АОМ в суперкомпьютерной среде. Последовательно разрабатываются типовые блоки подобных АОМ, имитирующие отдельные процессы, происходящие в обществе. Так, базовым блоком для любых АОМ, включающих популяции агентов-людей, служит блок, имитирующий естественное движение населения, то есть, такие процессы как смертность, рождаемость и миграция. Смертность в разрабатываемых авторами моделях имитируется на основе использования данных статистики о возрастно-половой структуре умерших для моделируемого региона. В то время как подход авторов к имитации рождаемости и миграции основан на индивидуальном поведении агентов, зависящем и от внутренних свойств агентов, и от факторов внешней для них среды. Каждый агент-человек выбирает партнера, создает семью и «рождает» ребенка (создает нового агента-человека) в соответствии со свойственной именно ему репродуктивной стратегией, зависящей, в свою очередь, от приверженности агента определенному типу воспроизводства. В моделях представлены два таких типа, существенным образом отличающихся друг от друга, – традиционный (предполагающий высокую рождаемость) и современный (с низкой рождаемостью). Принадлежность к тому или иному типу определяет такие характеристики агента как желаемое число детей в семье и вероятное время их рождения, так как кривые распределения числа рождений по возрасту матери, как свидетельствует статистика, также различаются для указанных типов репродуктивных стратегий. Общая схема работы демографической модели России представлена на рис. 1.

К миграции агентов в модели побуждает разница в уровне среднедушевых доходов между различными регионами, а также наличие привлекательных рабочих мест в других регионах или их отсутствие в регионе-месте жительства. Кроме того, большую роль в принятии миграционных решений играют социальные связи агентов (в частности, родственные), от которых во мно-

* Работа выполнена при финансовой поддержке Российского научного фонда (грант № 14-18-01968) с использованием вычислительных ресурсов Межведомственного суперкомпьютерного центра Российской академии наук (МСЦ РАН), а также суперкомпьютера Тяньхэ-2 Национального суперкомпьютерного центра университета оборонных технологий Китайской народной республики в г. Гуанчжоу.

гом зависит уровень информированности агентов об условиях рынка труда. Подробнее об этом можно прочесть в работе [1].

Имитация действий агентов, связанных с их участием в процессах естественного движения населения, предполагает их взаимодействие с другими агентами и с объектами внешней среды (например, с регионами), причем, это взаимодействие осуществляется в АОМ в форме обмена информацией. Соответственно, чем больше у агента «социальных связей» (в частности, родственных), тем интенсивнее он обменивается со своими контрагентами сообщениями для сбора информации, необходимой ему для принятия решения о создании семьи или о переезде в другой регион.

Очевидно, что для полноценной имитации естественного движения населения моделируемого региона, а тем более страны в целом, необходимо создавать в модели популяции агентов большой численности, что требует использования высокопроизводительной техники и специализированного программного обеспечения.

2. Особенности организации параллельных вычислений многоагентных симуляций

Резюмируя сказанное выше, отметим те специфические особенности агент-ориентированных моделей, предназначенных для имитации динамики больших социально-экономических систем с соответствующей численностью популяций агентов, которые важны с точки зрения организации распараллеливания их работы:

- агенты популяции имеют место жительства, то есть, привязку к тому или иному региону (юрисдикции), а также к координатам на карте региона и страны;
- агенты могут самостоятельно менять место жительства – мигрировать из одного региона в другой;
- численность популяции агентов меняется в течение периода симуляции, так как агенты могут уничтожаться и создаваться в ходе компьютерного эксперимента;
- агенты обладают «социальными связями» с другими агентами, с которыми они с разной интенсивностью обмениваются сообщениями, причем, эти связи могут образовывать агенты, проживающие в разных юрисдикциях;
- и место жительства агентов, и их связи с другими агентами существенным образом влияют на поведение агентов при образовании семьи и выборе места жительства.

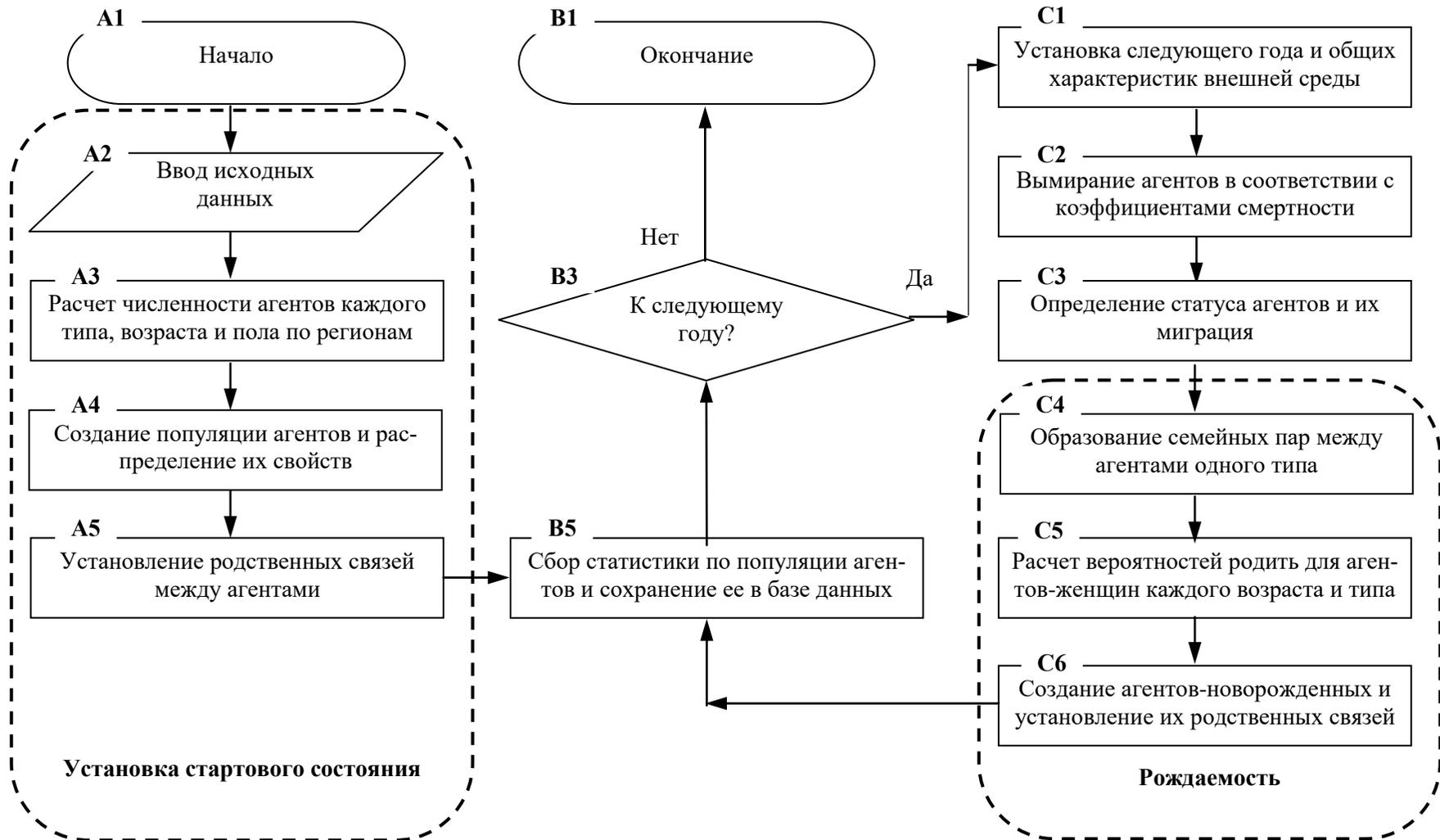


Рис. 1. Общая схема работы демографической модели России

Таким образом, для проведения многоагентного моделирования демографических процессов с большим числом агентов необходимо создать начальное состояние популяции агентов с заданным распределением их по регионам, возрастам, полу и прочим параметрам. А для эффективного распараллеливания работы симулятора и обеспечения хорошей масштабируемости модели необходимо осуществить распределение популяции агентов по процессорам именно таким образом, чтобы минимизировать связи агентов с агентами на других процессорах.

Выбор технологии распараллеливания работы больших АОМ определялся следующими критериями:

1. Система должна быть масштабируемой на множество узлов вычислительного кластера (то есть использовать ресурсы нескольких узлов для ускорения счета), поэтому многопоточная модель расчета оказалась не подходящей, так как она ограничена одним узлом кластера.

2. Модель должна быть легкой в разработке и при дальнейшем развитии, поэтому был выбран язык программирования высокого уровня C#.

3. Система должна быть эффективной и использовать системную библиотеку MPI вместо стандартного механизма сетевого взаимодействия TCP/IP или библиотек .Net, таких как Windows Communication Foundation, поскольку эти технологии не оптимальны для суперкомпьютеров. Библиотеки MPI, установленные на каждом кластере, обычно настраиваются на конкретную сетевую систему кластера и обеспечивают максимальную пропускную способность и минимальные задержки передачи данных. Кроме того, библиотека MPI доступна и для использования на обычных персональных компьютерах, что позволяет применять одну технологию и при разработке, и при проведении расчетов с использованием суперкомпьютеров без специальной доработки.

На уровне отдельных агентов должно поддерживаться моделирование эволюции внутреннего состояния агента, формирование постоянных и временных связей между агентами, обмен сообщениями, а также появление и исчезновение агентов в системе. Кроме того, для того чтобы эффективно проводить моделирование, в системе необходимо реализовать динамический механизм балансировки нагрузки между вычислительными процессами, т.е. переносить агентов с более загруженных процессов на другие с сохранением их состояния и связей.

Для балансировки нагрузки между процессами были выбраны алгоритмы METIS/ParMETIS [2], которые обычно используются для декомпозиции больших графов (до 10^9 вершин), расчетных сеток и матриц. Эти алгоритмы позволяют разделить граф агентов и связи между ними на части сходного размера с наименьшим возможным числом связей между частями. Алгоритм можно применять рекурсивно для эффективного расчета иерархического распределения графа. Использование алгоритма позволяет строить как начальную декомпозицию (распределение) системы, так и проводить уточнение декомпозиции в процессе расчета, что необходимо для поддержания балансировки нагрузки по мере добавления новых агентов в систему и удаления части старых агентов. Динамическая декомпозиция и перераспределение агентов должны позволить эффективно использовать до 1000 процессорных ядер современных суперкомпьютеров.

3. Программная реализация

Предложенный подход к созданию больших АОМ для запуска на суперкомпьютерах был апробирован на примере демографической модели России с учетом ее административного деления на отдельные регионы (субъекты РФ). При имитации процесса вымирания агенты действуют независимо друг от друга, что обеспечивает идеальную масштабируемость модели. Наиболее сложный с точки зрения дальнейшего распараллеливания работы модели процесс воспроизводства населения делится на каждом шаге имитации (соответствующем одному году) на следующие этапы:

- для каждого региона (субъекта РФ) выделяется множество взрослых агентов, не имеющих партнеров, из которого формируются две коллекции одиноких агентов – отдельно мужчин и женщин;
- агенты из сформированных коллекций внутри каждого региона обмениваются сообщениями о желаемом числе детей в семье, и если разница не превышает двух, то они образуют семью. Если агент-женщина в образовавшейся паре находится в репродуктивном возрасте, и

еще не родила столько детей, сколько хотела, то она пополняет коллекцию агентов-женщин, желающих родить ребенка в текущем году;

- исходя из полученной численности коллекций агентов-женщин, коэффициента фертильности и распределения рождений детей по возрасту матери рассчитываются текущие вероятности рождения детей для агентов разного возраста;
- полученные вероятности рождения детей агентами-женщинами в зависимости от их возраста используются для выбора матерей из коллекции претенденток.

Следует заметить, что эти алгоритмы имитации воспроизводства населения были апробированы ранее на последовательной версии модели и показали хорошую сходимость с наблюдавшимися фактическими данными. Так, численность населения, предсказанная в эксперименте на пятилетний период, ежегодно отличалась от фактической не более чем на 1%. Однако с точки зрения распараллеливания работы модели здесь действуют противоречивые факторы. С одной стороны, создание коллекций агентов, участвующих в том или ином процессе, значительно уменьшает потоки сообщений, которыми агенты обмениваются. В то же время само деление процесса воспроизводства на этапы означает создание точек синхронизации, что должно отрицательно влиять на показатели эффективности распараллеливания. Поэтому для исследования эффективности выбранных методов суперкомпьютерной реализации больших АОМ был сконструирован тестовый вариант модели, в которой заданная часть агентов популяции просто обменивались сообщениями со случайными агентами из своего региона.

В тестовом примере было реализовано считывание данных по регионам (численность населения) из таблицы в формате CSV и геометрии регионов из изображения в растровом формате, в котором регионы были закрашены разными цветами. Введенные данные использовались для построения прямоугольной сетки, каждая ячейка которой была привязана к определённому пикселю картинке, а цвет пикселя определял номер региона. Таким образом, при считывании этого изображения устанавливалась связь ячеек прямоугольной сетки с номерами регионов и другими их характеристиками. После чего осуществлялось случайное распределение числа агентов-жителей по ячейкам сетки с заданным общим числом агентов в системе и регионе, а также расчёт декомпозиции ячеек сетки по процессорам.

Для расчёта декомпозиции сетки использовался графовый алгоритм METIS с заданием весов (вариант METIS_PartGraphRecursive). Алгоритм METIS принимает на вход граф, заданный через матрицу связей в формате CSR[3] и массив весов узлов графа и возвращает оптимальное распределение графа на заданное число частей с минимизацией связей между частями. Матрица связей задается через число строк (N) и два массива IA и JA. Массив IA содержит последовательность частичных сумм числа ненулевых элементов в матрице. Этот массив используется для разделения элементов массива JA по строкам. Массив JA содержит списки индексов ненулевых элементов во всех строках.

В результате вызова алгоритма METIS для заданной матрицы связей и весов и заданного числа процессоров создаётся массив чисел, описывающий оптимальную привязку ячеек сетки к процессорам. Эти данные используются для распределения ячеек и агентов по процессорам во время расчёта, но не влияют на процедуру агрегирования полученных результатов расчёта, так как агрегировать результаты необходимо по регионам, независимо от используемого варианта декомпозиции.

С использованием описанного выше алгоритма было получено распределение исходной системы по процессорам. При реализации важно, что такое распределение нужно производить до создания агентов, поскольку создаваемые агенты должны исходно быть правильно распределены между узлами кластера.

Размер матрицы в нашем случае соответствовал числу непустых ячеек сетки, а связанными считались соседние (не по диагонали) ячейки сетки. Общая численность популяции агентов задавалась как параметр модели, так же, как и число используемых процессоров. После чего определялась доля агентов, приходящихся на каждый регион, как доля численности населения этого региона в общей численности населения страны. Внутри региона агенты распределялись равномерно по ячейкам. Такая схема расчёта позволяет менять общее число агентов, сохраняя их правильное географическое распределение.

Были опробованы два варианта автоматической декомпозиции сетки с использованием алгоритма METIS: без учёта числа агентов в ячейках (вес каждой ячейки = 1) (рисунок 2) и с учё-

том числа агентов (вес ячейки равен числу агентов) (рисунок 3). На рисунках показаны результаты декомпозиции на 8 процессорах, где разными оттенками отмечены области, относящиеся к разным процессорам. В первом случае алгоритм выравнивает число ячеек сетки на разных процессорах при минимальной длине границы между ними. Во втором случае выравнивается число агентов, приписанных к ячейкам конкретного процессора при минимизации длины границы. Очевидно, что во втором случае площади областей на карте, приписанных к разным процессорам, не равны из-за разницы в плотности населения. Ещё одним эффектом, связанным с формулировкой критериев оптимизации распределения, является то, что ячейки, расположенные на островах и эксклавах (пример – Калининградская область) изолированы, то есть, не связаны напрямую с ближайшими регионами, и могут считаться на любом из процессоров.



Рис. 2. Декомпозиция ячеек сетки на 8 процессорах без учета числа агентов в ячейках



Рис. 3. Декомпозиция ячеек сетки на 8 процессорах в соответствии с числом агентов в ячейках

Для анализа полученного распределения агентов по процессорам можно использовать закон Амдала, который связывает максимальное достижимое ускорение при параллельных вычислениях с числом процессоров и с долей последовательных вычислений [4]. В нашем случае, последовательные вычисления могут возникать из-за дисбаланса числа агентов на узлах, по-

этому полученные варианты декомпозиции были также оценены с точки зрения их масштабируемости при параллельном расчете.

На рисунке 4 показано сравнение графиков оценки ускорения при параллельном расчёте в зависимости от числа процессоров для двух вариантов декомпозиции. Видно, что неравномерность распределения агентов по ячейкам приводит к сильному перекосу в распределении числа агентов по процессорам в случае, если распределение не привязано к весам ячеек, что приводит к очень плохой масштабируемости. В то же время, из графика оценки масштабируемости видно, что весовая декомпозиция с использованием алгоритма METIS получается достаточно равномерной и не приводит к заметному дисбалансу вычислений.

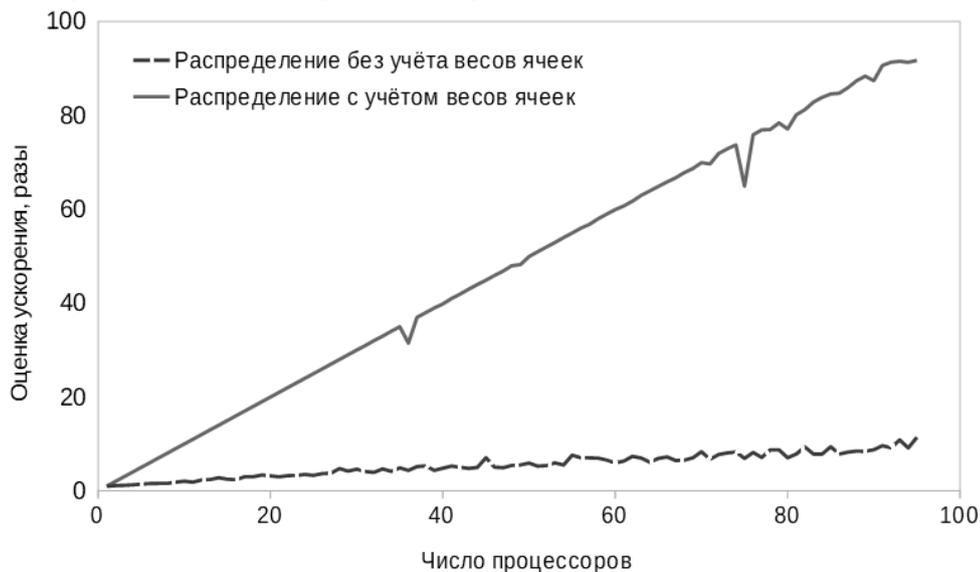


Рис. 4. Оценка ускорения при параллельном расчёте в зависимости от числа процессоров для двух вариантов декомпозиции модели в случае, если декомпозиция производится без учёта числа агентов в ячейках и с учётом числа агентов

4. Результаты экспериментов

Для проведения тестового расчёта была выбрана система, состоящая из восьми миллионов агентов, каждый из которых имеет 10 связей (агентов, с которыми он обменивается сообщениями) в своей ячейке на карте. Для проведения тестового расчёта использовался кластер МВС 100К межведомственного суперкомпьютерного центра РАН, каждый узел кластера представлял из себя двухпроцессорный сервер с процессорами Intel Xeon E5450 (суммарно 8 ядер на узел) с 8 ГБ оперативной памяти. Узлы кластера соединены сетью Infiniband DDR.

В таблице 1 представлены следующие результаты экспериментов для случаев 8 миллионов агентов и 32-х миллионов агентов: время расчёта шага и полученное ускорение относительно базового варианта. Данные для случая 32-х миллионов агентов приведены только для числа процессоров больше 48, из-за нехватки оперативной памяти при меньшем числе задействованных узлов кластера.

Таблица 1. Сравнение результатов распараллеливания работы модели, полученных при разном числе процессоров для случаев 8 и 32-х миллионов агентов.

Число процессоров	Время расчета одного шага для $8 \cdot 10^6$ агентов, с.	Ускорение	Время расчета одного шага для $32 \cdot 10^6$ агентов, с.	Ускорение
1	55.683	1.		
2	28.872	1.92		
4	16.114	3.45		
8	10.262	5.42		

12	7.006	7.94		
16	5.268	10.56		
24	3.796	14.66		
48	2.161	25.75	9.778	25.75*
96	1.378	40.38	4.840	52.01*
192	0.846	65.78	3.146	80.02*

Результаты, приведенные в таблице 1, показывают, что распределение агентов по процессорам в результате декомпозиции получилось достаточно равномерным, что привело к достаточно хорошим показателям параллельной эффективности.

Равномерное распределение числа агентов по процессорам позволяет также равномерно распределять загрузку оперативной памяти каждого узла кластера. Но при существенном увеличении числа агентов возникают сложности уже не только в обеспечении эффективности распараллеливания, но и в обеспечении самой возможности расчетов. Так, при расчете примера многоагентной системы с 148 миллионами агентов (что примерно соответствует реальной численности населения России) на 192 процессорах кластера основной сложностью было то, что при суммарном объеме оперативной памяти выделенных узлов в 192 Гб на каждом узле кластера находилось только 8 Гб оперативной памяти, при необходимом потреблении памяти примерно 500-600 байт на одного агента. Таким образом, необходимо было распределить агентов равномерно, чтобы ни на одном узле кластера процессы симуляции не превышали лимит потребления памяти.

На рисунке 5 показано распределение расчётной сетки по 192 процессорам, построенное с учётом плотности населения. В построенном распределении среднее число агентов, приходящихся на один процессор, было равно 772000. Численность агентов, приходящихся на один процессор варьировалась в диапазоне от 743711 до 787266, т.е. разброс числа агентов на одном процессоре составлял ~6% от среднего значения. Расчёт одного шага в данной симуляции занимал примерно 15 секунд. В дальнейшем разброс числа агентов может быть снижен за счёт использования более мелкой расчётной сетки, что повысит точность декомпозиции.

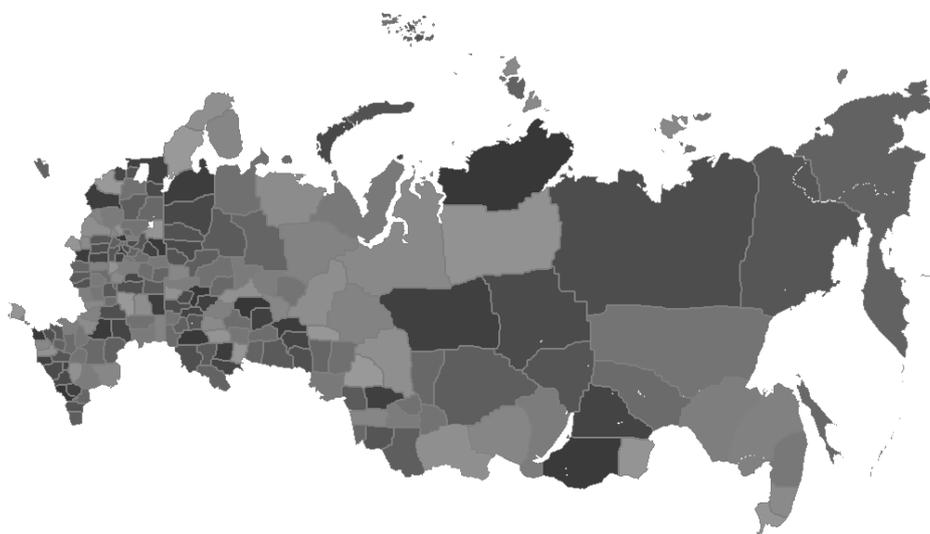


Рис. 5. Декомпозиция расчётной сетки на 192 процессора с учётом плотности населения

5. Обзор программных средств для реализации агент-ориентированных моделей в суперкомпьютерной среде

Существуют несколько реализаций агент-ориентированного подхода, адаптированных к использованию на суперкомпьютерах. Здесь уместно упомянуть о наиболее близких из них:

- Repast для высокопроизводительных вычислений (Repast HPC) 2.2.0, выпущенный в 2016 году. Основан на принципах и концепциях, разработанных в инструментальном пакете Repast Symphony и написан на C++ с использованием библиотеки MPI для параллельного исполнения кода программы. Он также широко использует библиотеку boost¹. Repast HPC предназначен для работы на разных платформах и операционных системах. Он может использоваться на рабочих станциях, кластерах и суперкомпьютерах под управлением Apple Mac OS X, Linux или Unix. Модели могут быть написаны либо в стандартном, либо в упрощенном синтаксисе C++ (в стиле языка Logo) [5].

- Pandora – представляет собой библиотеку с открытым исходным кодом, созданную исследовательской группой социального моделирования в Суперкомпьютерном центре Барселоны². Этот инструмент предназначен для реализации агентных моделей и их выполнения в высокопроизводительных вычислительных средах. Pandora была разработана с целью выполнения крупномасштабных симуляций на основе агентов и способна обрабатывать тысячи агентов, осуществляющих сложные действия. Pandora имеет тесную интеграцию с GIS-системами для того, чтобы осуществлять симуляции, в которых пространственные координаты актуальны, как с точки зрения взаимодействий агентов, так и с точки зрения их связи с окружающей средой. Результаты каждого моделирования хранятся в часто используемом иерархическом формате данных (HDF), который может быть загружен большинством ГИС. Эта функция особенно полезна, так как мы также будем использовать ГИС для анализа результатов моделирования. Библиотека Pandora дополняется программой Cassandra, разработанной для анализа результатов, генерируемых симуляцией. Cassandra позволяет пользователю визуализировать весь ход выполнения симуляций и полученные в результате симуляции статистические данные, используя комбинацию 2D и 3D графики.

- SWAGES – распределенная среда моделирования и экспериментов на основе агентов, которая использует автоматическое динамическое распараллеливание и распределение в гетерогенных вычислительных средах для минимизации времени счета. Среда SWAGES позволяет создавать агентов с использованием разных языков программирования, используя общую модульную архитектуру. SWAGES также позволяет использовать модули для подключения разных графических сред и механизмов моделирования и интегрировать их в общую среду SimWorld. Библиотека содержит множество средств для анализа, статистической обработки и визуализации результатов. SWAGES также включает в себя средства для проведения численных экспериментов, удобный web-интерфейс, средства обнаружения сбоев и проведения крупномасштабных симуляций [6].

- HPAVM – иерархическая среда параллельного моделирования для пространственно-распределённых агентно-ориентированных моделей, предназначенная для разработки сложных агентных моделей для исследования крупномасштабных задач, связанных с ГИС. В HPAVM модель на основе агентов декомпозируется на набор подмоделей, которые функционируют как вычислительные единицы для параллельных вычислений. Каждая подмодель состоит из подмножества агентов и их отдельных сред. Субмодели агрегируются в группу супермоделей, которые представляют вычислительные задачи. HPAVM, основанный на дизайне супер- и подмоделей, приводит к ослаблению связи между агентами и базовыми архитектурами параллельных вычислений. Эффективность HPAVM для создания параллельных агентных моделей была изучена в ходе вычислительных экспериментов, которые показывали, что HPAVM масштабируется для разработки широкомасштабных агент-ориентированных моделей и, таким образом, демонстрирует эффективную поддержку для расширения возможностей крупномасштабного комплексного агентного и геопространственного моделирования [7].

Можно также упомянуть Microsoft Axum – параллельный язык программирования, основанный на модели акторов, которая активно разрабатывалась Microsoft в период с 2009 по 2011 годы. Это объектно-ориентированный язык, основанный на .NET Common Language Runtime с использованием C-подобного синтаксиса, который, будучи предметно-ориентированным языком, предназначен для разработки хорошо распараллеливаемых частей программных приложе-

¹ URL: <http://boost.org> (дата обращения 14.04.2018)

² Дополнительная информация доступна на странице URL: <https://www.bsc.es/computer-applications/pandora-hpc-agent-based-modelling-framework> (дата обращения 14.04.2018)

ний. Кроме того, он содержит достаточно универсальных средств, которые позволяют не переключаться на язык программирования общего назначения (например, C#) для разработки последовательных частей программы. Число зависимостей между агентами минимально, они не используют совместно ресурсы, а чтобы координировать агентов или запросить у агента ресурсы другого, должно быть отправлено сообщение. В языке Ахум для этих целей реализован механизм каналов. Однако проект Ахум был доведен только до состояния прототипа с интеграцией Microsoft Visual Studio, который был представлен общественности, после чего проект был закрыт, а версия была удалена.

Сравнивая наш подход к разработке средств для эффективного распараллеливания работы многоагентных моделей с показанными в обзоре, следует подчеркнуть, что мы стремились учесть все наши специфические требования к таким средствам, которые связаны со множеством факторов. К таким факторам относятся: специфика больших социально-экономических систем, для имитации структуры и поведения которых мы предполагаем использовать АОМ; используемая нами собственная методология разработки многоагентных моделей; большая численность агентов, необходимая для адекватного воссоздания в модели социальной и экономической структуры населения; требования пользователей создаваемых АОМ в части их быстрой работы для проведения серий экспериментов и при этом требование разработчиков больших АОМ к снижению трудоемкости их разработки и др.

Можно отметить, что наш подход позволил учесть весь этот комплекс различных требований (иногда конкурирующих). К недостаткам представленной работы следует отнести слабую проработку связи АОМ с ГИС-системами, что является сейчас приоритетным направлением развития нашей библиотеки.

Если же обратиться к наиболее близкому опыту практической реализации АОМ в суперкомпьютерной среде, то следует отметить работы [8-9]. Так, в работе [8] описывается распределенная АОМ эпидемий, в рамках которой, по утверждению авторов, можно создавать социум с численностью агентов от нескольких сотен миллионов до 6 млрд. В модели здоровые агенты при контакте заражаются от больных. Модель реализована на языке Java. В экспериментах агенты численностью 300 млн. распределялись между двумя узлами, а внутри каждого узла на 8 потоков. Представлен алгоритм, позволяющий распределить агентов таким образом, чтобы снизить межгрупповые контакты агентов, что сходно с применением алгоритма METIS в этой статье. Приводится сравнение результатов, полученных на двух версиях АОМ: одна распараллелена без использования данного алгоритма, а другая – с использованием. Использование этого алгоритма позволило заметно снизить время, затрачиваемое на сетевые обмены, и повысить производительность.

В работе [9] также рассматривается подход к распараллеливанию ресурсоемких АОМ, агенты которых обмениваются информацией и имеют пространственную привязку. Параллельная версия агентной модели была реализована в C++ с использованием интерфейса OpenMP, предназначенного для программирования многопоточных приложений на многопроцессорных системах с общей памятью, что принципиально ограничивает возможность роста числа моделируемых агентов размером оперативной памяти одного узла кластера. В рамках проведенных экспериментов тестировалась производительность параллельной версии модели в зависимости от двух свойств: 1) размера пространства, связанного с количеством распределенных по нему агентов, участвующих во взаимодействиях (определяет общий объем межагентных взаимодействий); и 2) радиуса взаимодействия (определяет максимальное расстояние, на котором пара агентов может связываться). Ускорение, полученное авторами по сравнению с последовательной версией модели, составило в среднем 20 раз.

Общий вывод, который делают авторы, и с которым можно согласиться, заключается в том, что распараллеливание агентных моделей в любом случае позволяет значительно повысить их производительность, однако способы распараллеливания во многом зависят от специфики конкретной модели (частота общения агентов, их концентрация, сложность внутреннего устройства и т.д.). Применение явной декомпозиции задачи и технологий, предназначенных для работы в распределенной среде, должно позволить производить симуляции с большим числом агентов и достигать хороших показателей параллельной эффективности.

6. Заключение

Для моделирования поведения искусственного общества в целом требуется создание агент-ориентированных моделей с числом агентов до 10^9 и использование суперкомпьютерных технологий для проведения симуляций. Важной задачей в таких расчётах становится оптимальное распределение агентов по процессорам кластера с учётом основных особенностей агент-ориентированных моделей. В этой работе была представлена новая библиотека для параллельного выполнения многоагентного моделирования и показаны результаты ее апробации. Эта библиотека соединяет использование высокоуровневого языка программирования C# и высокопроизводительной платформы для обмена сообщениями, написанной с использованием C++-библиотеки и системной библиотеки MPI, установленной на суперкомпьютере, а также средства автоматической и динамической декомпозиции моделей с использованием алгоритма METIS. Представленные результаты тестового моделирования показывают хорошую масштабируемость программы на множестве вычислительных узлов, даже в случае создания в модели популяции агентов, сопоставимой по численности с населением России. На наш взгляд, эксперименты с крупномасштабной демографической агент-ориентированной моделью России, подтверждают применимость предложенного подхода к моделированию больших социально-экономических систем со сложными взаимодействиями агентов.

Данная разработка была представлена в рамках пленарного доклада "Development of a system for projecting an agent-based model with running on supercomputers" на первой встрече рабочей группы стран БРИКС, посвящённой инновационному сотрудничеству в сфере информационно-коммуникационных технологий и высокопроизводительных вычислений (The First BRICS Working Group Meeting and Innovation Collaboration Forum on Information Technology and Communication and High-Performance Computing). Встреча проходила в апреле 2017 года в университете Гуанчжоу (Китай), а также на базе Aloft Guangzhou University Park. Основная цель организаторов встречи – достигнуть существенных результатов в развитии искусственного интеллекта, больших данных, высокопроизводительных вычислений, кибербезопасности, моделирования социально-экономических систем стран БРИКС с использованием агент-ориентированного подхода и суперкомпьютерных технологий в рамках принятого ранее рабочего плана стран БРИКС в сфере науки, технологий и инноваций на следующие пять лет.

В ходе дискуссии были определены как критерии для выбора приоритетных проектов БРИКС, так и сами эти приоритетные направления, одним из которых было признано «Крупномасштабное многоагентное моделирование виртуального общества (Large Scale Multi-agent based Simulation of Virtual Society)».

Литература

1. Макаров В.Л., Бахтизин А.Р., Сушко Е.Д., Агеева А.Ф. Искусственное общество и реальные демографические процессы // Экономика и математические методы. 2017. Т. 53. № 1. С. 3-18.
2. Karypis G., Kumar V. METIS-unstructured graph partitioning and sparse matrix ordering system, version 2.0. URL: <http://dm.kaist.ac.kr/kse625/resources/metis.pdf> (дата обращения 31.03.2018).
3. W. Tinney and J. Walker. Direct solutions of sparse network equations by optimally ordered triangular factorization. Proceedings of the IEEE, 55(11):1801–1809, Nov. 1967
4. Amdahl, G.M. Validity of the single processor approach to achieving large scale computing capabilities. In Proceedings of the April 18-20, 1967, spring joint computer conference. P. 483-485. ACM. 1967.
5. Collier N. Repast HPC Manual. URL: <http://repast.sourceforge.net/docs.php> (дата обращения 14.04.2018).
6. Scheutz M., Connaughton R., Dingler A., Schermerhorn P. SWAGES – An Extendable Distributed Experimentation System for Large-Scale Agent-Based Alife Simulations / In Proceedings of Artificial Life X, 2006. P. 412-419.

URL: <https://hrilab.tufts.edu/publications/scheutzetal06alifeswages.pdf> (дата обращения 14.04.2018).

7. Tang W., Wang S. HPABM: A Hierarchical Parallel Simulation Framework for Spatially-explicit Agent-based Models // Transactions in GIS, 2009. № 13(3). P. 315-333.
8. Parker, J. A Flexible, Large-Scale, Distributed Agent Based Epidemic Model. Center on Social and Economic Dynamics, Working Paper № 52, 2007.
9. Gong, Z., Tang, W., Bennett, D.A., and Thill J.C. Parallel agent-based simulation of individual-level spatial interactions within a multi-core computing environment // International Journal of Geographical Information Science, 2013. № 27 (6). P. 1152-1170.