

трещины. Отдельные участки стен заморожены и разрушены. Во внутренних несущих стенах центральных лестничных клеток выявлены трещины шириной раскрытия до 10 мм на всю высоту здания. Внутри здания выявлены дефекты и повреждения, обусловленные длительной эксплуатацией здания без проведения своевременных ремонтов.

Данное здание было смоделировано в конечно-элементном программном комплексе ANSYS, выполнены расчеты на действие эксплуатационных нагрузок, замеренных и прогнозируемых осадок. Расчет выполнялся с использованием пространственных конечных элементов SOLID65, позволяющих учесть возможность трещинообразования и раскрашивания материалов конструкций. Характер разрушения конструкций здания, полученный в результате обследования и расчета достаточно близок, что позволило исследовать возможность применения некоторых конструктивных мер защиты здания (введение поясов усиления, возможность разрезки здания на отсеки).

А.С. Кириллов

Оренбургский государственный университет

ПОДХОДЫ К РАЗРАБОТКЕ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ РЕШЕНИЯ КООПЕРАТИВНЫХ ИГР ДЛЯ КЛАСТЕРНЫХ СИСТЕМ¹

В последние годы значение теории игр существенно возросло во многих областях экономических и социальных наук. В экономике она применяется не только для решения хозяйственных задач, но и для анализа стратегических проблем предприятий, разработок организационных структур и систем стимулирования.

¹ Работа выполнена при финансовой поддержке Министерства образования и науки Российской Федерации в рамках реализации ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009–2013 гг.

Аппарат теории игр активно используется при создании систем поддержки принятия решений. При этом очень актуальна проблема разработки эффективных алгоритмов для поиска оптимальных стратегий в игровых моделях. На практике требуется, чтобы алгоритмы были способны обрабатывать большое количество информации об участниках рынка в режиме реального времени. Направление теории игр, связанное с разработкой и анализом эффективных алгоритмов их решения, получило название «Алгоритмическая теория игр» и очень активно развивается именно в последние 15 лет. Коллективная монография [1], подводящая первые итоги становления алгоритмической теории игр, была издана совсем недавно – в 2007 году.

Среди направлений теории игр особое место занимают кооперативные игры, они используются, например, в разработке оптимальных стратегий безопасности систем, эффективной ценовой политики и др. В кооперативных играх игроки имеют возможность принимать совместные решения и, в некоторых случаях, перераспределять выигрыши между собой. В кооперативной игре задаются возможности и предпочтения различных групп игроков (коалиций), для которых строятся оптимальные стратегии и задаются распределения между ними суммарных выигрышей. Результатом решения кооперативной игры являются оптимальная коалиция, в которую следует вступить игрокам, их оптимальные смешанные стратегии и выигрыш коалиции. Распределение выигрыша между игроками определяется дележом.

Особого внимания заслуживают задачи моделирования конкурентных рынков. Примером подобной задачи может быть моделирование конкурентного рынка отрасли информационных технологий, где игроками являются производители компьютерной техники и программного обеспечения, имеющие возможность объединяться в коалиции, конкурирующие за соответствующие ниши рынка.

Существуют различные способы решения кооперативных игр, например: решение на основе функции полезности Неймана-Моргенштерна, решение игры в виде С-ядра и в стратегиях

угроз, однако эти подходы не определяют оптимальных стратегий игроков внутри коалиций. А решение игры в виде С-ядра дает целое множество допустимых решений, не определяя оптимального решения.

Попытки построения количественного подхода к решению кооперативной игры описаны в работах Протасова [2] и Фергюсона [3]. Предложен подход, основанный на представлении кооперативной игры набором биматричных игр для каждой пары игроков. Недостатком этого подхода является его высокая вычислительная сложность, поскольку количество биматричных игр экспоненциально растет при увеличении количества игроков.

Таким образом, разработанные в настоящее время алгоритмы решения кооперативных игр имеют серьезные практические ограничения по количеству игроков и не могут использоваться в современных системах поддержки принятия решений. В нашей работе мы предлагаем подходы к решению кооперативных игр с использованием параллельных вычислений для различных типов параллельных архитектур.

Анализ подходов к распараллеливанию решения кооперативной игры. Один из актуальных современных практических подходов к решению задач высокой вычислительной сложности связан с использованием параллельных вычислений. Разработка эффективных высокомасштабируемых параллельных алгоритмов и широкое распространение многоядерных архитектур, вычислительных кластеров, грид-систем позволяют аккумулировать при решении задачи большие вычислительные ресурсы и резко сократить время поиска решения.

Разработка параллельных алгоритмов для решения игр – новое направление в алгоритмической теории игр. На сегодняшний день имеются лишь единичные публикации в этой области, например работы Уиджера и Гросу для некооперативных игр [4, 5]. Для решения кооперативных игр параллельные алгоритмы пока не разработаны. Отметим, что разнообразие свойств параллельных архитектур требует исследования и адаптации параллельных алгоритмов под различные типы архитектур. Так, алгоритмы для систем с общей памятью могут неэффективно

работать в распределенных системах, для оптимизации скорости алгоритмов требуется учитывать характеристики коммуникационной системы и т.д.

Для решения кооперативных игр можно применить сведение задачи к множеству стратегических игр и решение каждой из них симплекс-методом – для нахождения выигрыша каждой коалиции (более подробно описано в работе [6]). Всего коалиций может быть 2^n , где n – количество игроков, следовательно, необходимо решить 2^n стратегических игр симплекс-методом Дж. Данцига, который обладает экспоненциальной сложностью [7]. При достаточно больших n это занимает длительное время.

Кроме того, большую вычислительную сложность имеет процесс формирования матриц, которыми задаются коалиции. Трудоемкость этого процесса составляет $O(n^2)$. Поскольку матрицы, описывающие коалиции, не зависят друг от друга, а зависят только от матриц, задающих саму игру, генерация матриц может осуществляться параллельно.

Исходя из вышесказанного, можно сделать вывод, что формирование матриц, характеризующих коалиции, и их расчет симплекс-методом являются независимыми участками кода, которые можно выполнять параллельно.

Нами разработан алгоритм решения кооперативных игр для кластерных систем. Суть разработанного параллельного алгоритма состоит в следующем.

Нулевым процессом выполняется считывание матриц, задающих игру, после чего они рассылаются остальным незанятым работающим процессам, которые, в свою очередь, формируют матрицу, характеризующую коалицию. Номера игроков, входящих в коалицию, определяются по соответствию позиций единичных элементов двоичного представления порядкового номера формируемой матрицы. После чего для сформированной коалиции строится соответствующая задача линейного программирования и решается симплекс-методом. Результат отсылается нулевому процессу, который проверяет игру на сущест-

венность, и если игра существенна, то вычисляется дележ по Шепли. Затем все результаты выводятся в выходной файл.

Схема работы параллельной программы представлена на рис. 1.

Алгоритм реализован на С с использованием библиотеки MPI (Message Passing Interface).



Рис. 1. Схема работы параллельного алгоритма

В приведенном выше алгоритме доля последовательного участка подчиняется следующему закону: $f = \frac{k}{k^{n-1} + k}$, из которого видно, что при возрастании n или k доля последовательных вычислений стремится к нулю, где n – количество игроков, k – количество стратегий у игроков.

Тестирование программы производилось на кластере Оренбургского государственного университета, построенного на базе 2-процессорных Xeon 3ГГц под управлением операционной системы SuSe Linux.

Тестирование производилось для 7 игроков, и у каждого игрока имелось по 4 стратегии поведения.

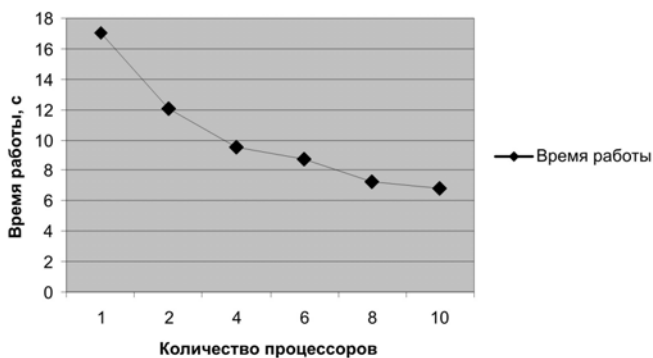


Рис. 2. График зависимости времени работы от количества процессоров

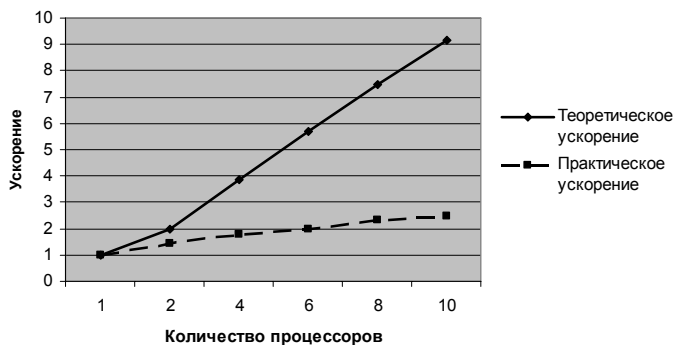


Рис. 3. График зависимости теоретического и практического ускорения от количества процессоров

Зависимость времени работы программы от количества процессоров представлена на рис. 2.

Для наглядного представления построим график зависимости теоретической оценки ускорения по закону Амдала и ускорения работы программы, рассчитанной после тестирования от количества процессоров, который представлен на рис. 3.

Итак, в статье была рассмотрена идея сведения кооперативных игр к множеству биматричных для игроков с большим числом стратегий и предложен способ задания и решения коо-

перативных игр. После анализа алгоритма была предложена схема решения поставленной задачи с использованием высокопроизводительных вычислений, в частности, был разработан и программно реализован параллельный алгоритм. После запуска программы на кластере ОГУ был проведен анализ полученных результатов.

Как видно из графика, приведенного на рис. 3, наблюдается тенденция ускорения работы программы с увеличением количества работающих процессоров, хотя максимальное теоретическое ускорение и не была достигнуто.

Несоответствие теоретического и практического ускорения может быть связано с тем, что теряется время на пересылку данных между процессами и/или с тем, что нулевой процесс простаивает, ожидая завершения работы остальных процессов для продолжения решения задачи.

В дальнейшем, планируется рассмотреть другие способы решения кооперативных игр и модификации алгоритмов для других типов параллельных архитектур.

Список литературы

1. Algorithmic Game Theory / ed. by Noam Nisan, Cambridge University Press, 2007. – 754 p.
2. Протасов И.Д. Теория игр и исследование операций. – М.: Гелиос АРВ, 2006. – 368 с.
3. Thomas S. Ferguson GAME THEORY Class notes for Math 167, Fall, 2000.
4. Widger J. and Grosu D. Computing Equilibria in Bimatrix Games by Parallel Support Enumeration. In Proceedings of the 2008 international Symposium on Parallel and Distributed Computing (July 01–05, 2008). ISPD. IEEE Computer Society. – Washington, DC, 2008. – P. 250–256.

5. Widger J. and Grosu D. Parallel Computation of Nash Equilibria in N-Player Games, Computational Science and Engineering, IEEE International Conference, 2009. – P. 209–215.

6. Колемаев В.А., Калинина В.Н. Теория вероятностей и математическая статистика: учеб. для вузов. – 2-е изд., перераб. и доп. – М.: Юнити, 2003. – 352 с.

7. Данциг Дж. Линейное программирование, его применения и обобщения. – М.: Прогресс, 1966. – 600 с.

Е.А. Козин

Нижегородский государственный университет им. Н.И. Лобачевского

РАЗЛИЧНЫЕ ПОДХОДЫ К ВИЗУАЛЬНОМУ ПРЕДСТАВЛЕНИЮ ПАРАЛЛЕЛЬНЫХ ПРОГРАММ

В настоящей работе рассматривается одна из актуальных проблем написания параллельных программ – снижение сложности и времени разработки. В настоящее время большая часть процессоров в различных вычислительных устройствах имеют более одного ядра. Во всех крупных компаниях используются многопроцессорные вычислительные системы. Как следствие, эффективная программа должна уметь утилизировать данные ресурсы, т.е. быть параллельной. На создание параллельной программы, как правило, уходит существенно больше времени, чем на разработку последовательной программы. Связано это с тем, что в параллельных программах могут возникать «параллельные ошибки», сложность выявления и устранения которых гораздо выше «обычных ошибок». Для снижения сложности написания параллельных программ может быть применено визуальное программирование. Визуальный язык может помочь программисту построить правильный «каркас» параллельной программы. Сгенерированный «каркас» в данном случае берет на себя всю сложность организации параллельных вычислений, оставив на программиста последовательные участки кода, отвечающие за программируемый алгоритм. Такая схема