

УДК 519.71

**ИЗВЛЕЧЕНИЕ ЗНАНИЙ АГЕНТАМИ В
СИСТЕМЕ УПРАВЛЕНИЯ РАСПРЕДЕЛЕННЫМИ ВЫЧИСЛЕНИЯМИ**

Феоктистов Александр Геннадьевич

К.т.н., доцент, с.н.с., e-mail: agf65@yandex.ru

Костромин Роман Олегович

Аспирант, e-mail: romang70055@gmail.com

Институт динамики систем и теории управления имени В.М. Матросова СО РАН,
664033, г. Иркутск, ул. Лермонтова, д. 134

Аннотация. Управление масштабируемыми приложениями для решения больших научных задач в гетерогенной распределенной вычислительной среде является нетривиальной проблемой. Такие приложения генерируют конкурирующие потоки заданий, выполняемые с помощью общих ресурсов среды. Эффективным подходом к решению этой проблемы является использование мультиагентных технологий. С этой целью разработана мультиагентная система для управления масштабируемыми приложениями. В отличие от существующих мультиагентных систем, она основана на применении специальной концептуальной модели вычислительной среды, включающей различные компоненты комплексных знаний как об окружающей среде, так и о предметных областях решаемых задач. Предложен новый подход к извлечению этих компонентов знаний посредством комплексного использования концептуального моделирования распределенных вычислений, классификации заданий и ресурсов, а также параметрической настройки алгоритмов функционирования агентов. Практические эксперименты подтверждают эффективность мультиагентного управления масштабируемыми приложениями.

Ключевые слова: распределенные вычисления, мультиагентное управление, самоорганизация, извлечение знаний, имитационное моделирование

Введение. Стремительный рост числа вычислительных компонентов в НРС-системах актуализирует разработку масштабируемых приложений. Масштабируемость вычислений означает то, что время решения задачи пропорционально уменьшается с увеличением числа узлов, используемых приложением. Таким образом, решаемая задача должна обладать возможностью декомпозиции на подзадачи, которые могут быть решены с максимальной степенью независимости друг от друга. Суперкомпьютеры и кластеры обеспечивают решение задач в вычислительных средах, включающих гетерогенные, автономные и географически распределенные узлы, которые обладают сложной гибридной структурой, различными вычислительными характеристиками и административными политиками [15]. Владельцы узлов определяют критерии их использования (эффективность, балансировка нагрузки, энергопотребление, стоимость и т. д.). Grid-технологии и облачные инфраструктуры, как правило, образуют такую гетерогенную распределенную вычислительную среду (ГРВС).

Системы управления ресурсами (Resource Management System, RMS) применяются для работы на уровне узлов ГРВС. Традиционными RMS являются такие системы, как PBS Torque, HTCondor и SLURM. Промежуточное программное обеспечение используется для управления вычислениями в узлах ГРВС посредством взаимодействия с RMS или путем ее замещения. Как правило, ГРВС поддерживает коллективное использование своих ресурсов для решения широкого класса задач. Спецификация задач включает информацию о необходимых вычислительных ресурсах, исполняемых программах, входных и выходных данных, критериях качества решения задачи и другие необходимые данные. Спецификация описывает задание для традиционных RMS и метапланировщиков, используемых в узлах ГРВС.

В общем случае задание включает множество взаимосвязанных подзаданий. Существует широкий спектр систем управления потоками заданий (Workflow Management Systems, WMS), таких, как: Condor DAGMan, Pegasus, Taverna, UNICORE и др. [14]. Тем не менее, не все проблемы управления потоком заданий в ГРВС полностью решены [12]. WMS функционирует на уровне приложений. Ее целью является получение наилучших ресурсов для своего потока заданий, что обуславливает конкуренцию за ресурсы с другими приложениями.

В отличие от WMS, RMS и мета-планировщики работают на уровне среды. Администраторы RMS и мета-планировщиков определяют общие правила использования ресурсов и политики распределения квот, а также прав пользователей и их заданий. Однако эффективное согласование пожеланий пользователей и владельцев ресурсов в процессах распределения этих ресурсов для решения задач, с учетом специфики предметных областей, часто остается за пределами возможностей администраторов.

Перспективным подходом к решению этой нетривиальной проблемы является применение мультиагентных технологий, основанных на принципах самоорганизации агентов [8]. Агенты могут представлять пользователей и владельцев ресурсов, а также общаться друг с другом с целью достижения собственных целей. Известны МАС, которые успешно используются на практике [1, 5, 9, 10]. Тем не менее, существуют компоненты знаний о предметной области, свойствах заданий и т.д., которые не в полной мере используются этими системами.

В статье представлена мультиагентная система управления масштабируемыми приложениями, которая позволяет специалистам в различных областях распределенных вычислений применять собственные знания на разных уровнях процесса решения задач.

1. Мультиагентная система. Разработанная МАС имеет иерархическую структуру, которая включает два или более уровней и работает на основе принципов самоорганизации. На каждом уровне агенты выполняют различные функции. Управление вычислениями основано на общении агентов. На каждом уровне иерархии агенты могут объединяться в виртуальные сообщества (ВС), кооперироваться и конкурировать в рамках этих сообществ. Формирование ВС позволяет МАС адаптировать процесс управления к новым задачам. МАС применяет алгоритм статического распределения ресурсов на основе экономических механизмов регулирования их спроса и предложения. Используется модель аукциона Викри [13].

Пользователи формулируют постановку задачи и определяют критерии ее решения. Пользовательские агенты создают и классифицируют задания, основываясь на постановке

задачи. Затем эти агенты передают задания агентам распределения ресурсов. Создается ВС агентов, наиболее подходящее для решения задачи. Каждый агент ВС делает ставки для всех подзаданий, отражающие истинную стоимость их выполнения и максимальную выгоду. В конце аукциона достигается устойчивое состояние ВС, аналогичное равновесию в теоретико-игровых моделях, определенному Нэшем [11]. Правила аукциона Викри обеспечивают также простоту и необходимую скорость принятия решения агентами.

ВС обеспечивает управление выполнением задания. В процессе вычислений агенты могут перераспределять свою вычислительную нагрузку среди других агентов посредством локального общения с соседями. Агенты обеспечивают динамическую декомпозицию задачи на подзадачи, когда появляются дополнительные свободные ресурсы. Декомпозиция позволяет ускорить решение задачи и сделать ее масштабируемой.

Известны различные методы извлечения знаний [7]. Концептуальное и имитационное моделирование являются основными методами, используемыми агентами.

2. Концептуальная модель. Методы концептуального моделирования являются мощным инструментом для разработки сложных систем [2]. Концептуальная модель позволяет системным проектировщикам и разработчикам представлять результаты своего экспертного анализа предметной области. Концептуальная модель предметно-ориентированной ГРВС включает в себя следующие компоненты знаний: знания о программных модулях (программах) для решения задач в предметных областях и работы с объектами ГРВС; схемные знания о модульной структуре моделей и алгоритмов; продукционные знания, обеспечивающие поддержку принятия решений для выбора оптимальных алгоритмов в зависимости от состояния ГРВС; знания о программно-аппаратной инфраструктуре ГРВС и административных политиках в ее узлах, включая информацию о пользователях, их задачах и заданиях.

Для создания концептуальных моделей разработан специализированный инструментарий высокого уровня SIRIUS II [4]. Этот инструментарий используется для извлечения предметно-ориентированных знаний разработчиков приложений. Более детальную информацию о концептуальной модели можно найти в [6].

3. Система классификации заданий. Разработана система классификации заданий, предназначенная для улучшения процесса распределения ресурсов. Эта система позволяет получать знания от администраторов о соответствии заданий и ресурсов.

Требования задания к ГРВС представляют конкретные характеристики, такие, как время решения задачи, ОЗУ, объем дисковой памяти, число узлов, процессоров и ядер, библиотеки программ, компиляторы, их ключи и другую необходимую информацию.

Распознавание свойств заданий осуществляется с помощью набора характеристических функций. После классификации задание конкретизируется путем добавления в него директивы для RMS, определяющей допустимые для распределения ресурсы. MAC создает ВС агентов, соответствующее этим ресурсам, и отправляет ему конкретизированное задание. ВС выполняет окончательный выбор и распределение ресурсов.

4. Настройка алгоритмов агентов. Алгоритмы распределения ресурсов имеют ряд управляющих параметров: ограничения загрузки компонентов узла; бонусы за удовлетворение этим ограничениям; штрафы за их превышение; приоритеты классов заданий; степень желанья выполнять задания определенных классов. Администраторы

задают начальные значения этих параметров при настройке МАС. Эти параметры автоматически корректируются на основе имитационного моделирования и метамониторинга в процессе работы МАС.

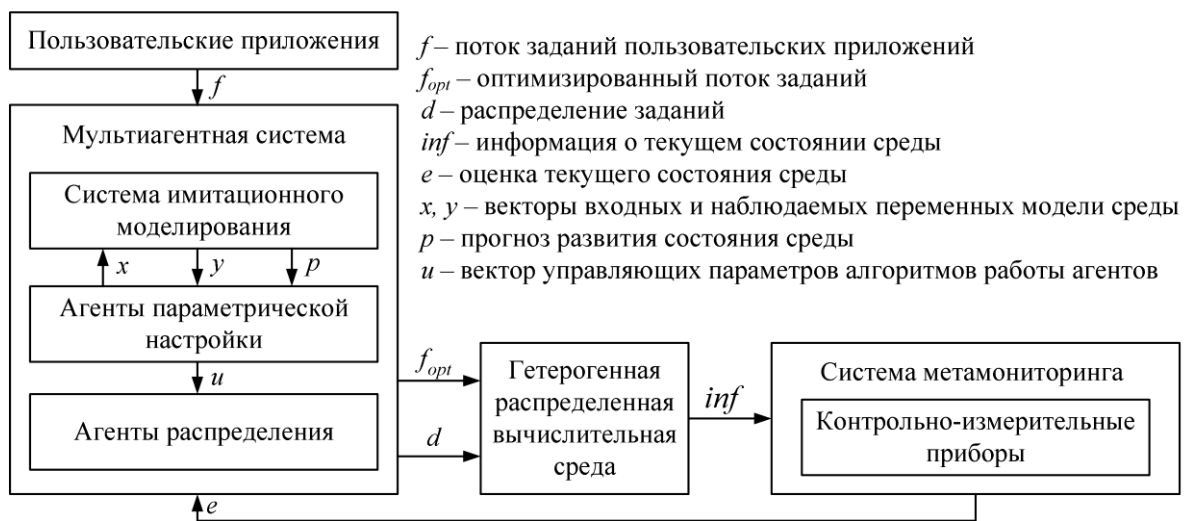


Рис. 1. Схема параметрической настройки алгоритмов работы агентов

На рис. 1 показана схема параметрической настройки. Для параметрической настройки используется имитационное моделирование. Решается следующая задача:

$$v_i(z) \rightarrow \min(\max) \quad (1)$$

$$v_i^{\min} \leq v_i \leq v_i^{\max}, i = \overline{1, n}. \quad (2)$$

Имитационная модель создается в виде приложения, поддерживающего многовариантные расчеты. Система мета-мониторинга обеспечивает оценку текущего состояния ГРВС. Результаты моделирования представлены вариантами значений вектора v . Для формирования вектора u применяется многокритериальный выбор вариантов вектора v с использованием условий, представленных в (1) и (2) [3]. Экстремумы для каждого элемента вектора v назначаются перед началом моделирования. Параметрическая настройка позволяет агентам делать эффективные ставки на аукционе Викри и применяется как элемент их обучения. Обратная связь в процессе обучения осуществляется на основе бонусов и штрафов. Агенты корректируют свои собственные решения с учетом этой обратной связи.

5. Вычислительные эксперименты. Проведен анализ вычислительной истории более 80000 реальных заданий, выполненных на трех кластерах с различными ресурсами (табл. 1). Кластеры 1 и 3 включают узлы, которые используются только в рамках кластера. Узлы кластера 2 задействуются как их владельцами, так и пользователями кластера. В системе Cleo все ядра узла выделяются по запросу задания, независимо от числа запрошенных ядер, что зачастую приводит к неэффективному использованию ресурсов.

В данной работе эта проблема решается с помощью системы классификации. В ней определен набор родительских классов заданий, основанных на анализе их вычислительной истории. Эти классы отличаются количеством запрашиваемых ядер и временем выполнения. На их основе созданы дополнительные классы заданий с учетом требуемого ОЗУ, дискового пространства, операционной системы, прикладного программного обеспечения и т.д. Каждый дополнительный класс соответствует одному из кластеров.

Таблица 1. Вычислительные кластеры

Кластер	Пиковая производит ельность кластера	Число узлов/ процессоров/ ядер	RMS	Средняя загрузка процессора узла без классификации заданий	Средняя загрузка процессора узла с классификацией заданий
Кластер 1	1,50 TFlops	20/40/160	Cleo	70 %	86 %
Кластер 2	0,77 TFlops	16/16/64	HTCondor	41 %	64 %
Кластер 3	0,17 TFlops	16/32/32	СУППЗ	28 %	56 %

Для проведения эксперимента разработана имитационная модель ГРВС. В ней сформирован поток заданий в соответствии с полученной вычислительной историей. Каждое задание классифицируется и передается RMS соответствующего кластера. Средняя загрузка процессора для трех кластеров при обработке потока заданий с использованием системы классификации улучшена на 16%, 23% и 28%. Было перераспределено более 12000 заданий, сохранивших все характеристики определенных для них классов.

В следующем эксперименте использован тот же самый поток заданий. Выполнен сравнительный анализ эффективности загрузки процессоров узлов кластеров под управлением мета-планировщиков HTCondor и GridWay, а также разработанной МАС, используемых в качестве промежуточного программного обеспечения, взаимодействующего с RMS. Результаты, приведенные в табл. 2, иллюстрируют преимущества МАС по сравнению с другими системами. Эти преимущества достигаются за счет параметрической настройки, представленной в разделе 4.

Таблица 2. Загрузка процессоров узлов кластеров

Кластер	HTCondor	GridWay	МАС
Кластер 1	76,98	80,87	81,56
Кластер 2	73,55	72,97	74,65
Кластер 3	76,85	77,96	78,89

Все эксперименты выполнены в распределенной вычислительной среде на базе ресурсов Иркутского государственного университета и Центра коллективного пользования «Иркутский суперкомпьютерный центр СО РАН».

Заключение. В статье рассмотрена проблема мультиагентного управления в ГРВС. Показано, что эффективность управления в значительной степени зависит от способности использования предметно-ориентированных знаний. Предложен подход к извлечению, представлению и использованию всесторонних знаний о ГРВС и предметных областях решаемых задач. Специалисты в различных областях распределенных вычислений могут генерировать эти знания на разных уровнях процесса решения задач на основе собственных навыков и опыта. Разработана МАС для управления масштабируемыми приложениями, которые генерируют потоки заданий. МАС функционирует на основе принципов самоорганизации. Имитационное моделирование и практические эксперименты с многовариантными расчетами показывают эффективность предложенного подхода к управлению в рамках разработанной МАС благодаря извлечению концептуальных знаний.

Исследование выполнено при частичной финансовой поддержке РФФИ, проекты № 15-29-07955-офи_м и № 16-07-00931-а, Программы фундаментальных исследований Президиума РАН 1.33П, проект «Разработка новых подходов к созданию и исследованию моделей сложных информационно-вычислительных и динамических систем с приложениями», а также Совета по грантам Президента Российской Федерации для государственной поддержки ведущих научных школ Российской Федерации (НШ-8081.2016.9).

СПИСОК ЛИТЕРАТУРЫ

1. Каляев А.И., Каляев И.А. Метод мультиагентного диспетчирования ресурсов в облачных вычислительных средах // Известия РАН. Теория и системы управления. 2016. № 2. С. 51–63.
2. Тыугу Э.Х. Концептуальное программирование. М.: Наука, 1984. 256 с.
3. Феоктистов А.Г. Управление сложной системой на основе методологии многокритериального выбора управляющих воздействий // Фундаментальные исследования. 2015. № 9-1. С. 82–86.
4. Феоктистов А.Г., Корсуков А.С., Дядькин Ю.А. Аспекты имитационного моделирования процессов мультиагентного управления распределенными вычислениями // Информационные и математические технологии в науке и управлении, 2016. №4-1. С. 118–126.
5. Amato A., Venticinque S. A Distributed Agent-Based Decision Support for Cloud Brokering // Scalable Computing: Practice and Experience. 2014. Vol. 15. № 1. Pp. 65–78.
6. Bychkov I., Oparin G., Tchernykh A., Feoktistov A., Bogdanova V., Gorsky S. Conceptual Model of Problem-Oriented Heterogeneous Distributed Computing Environment with Multi-Agent Management // Procedia Computer Science. 2017. Vol. 103. Pp. 162–167.
7. Cooke N. J. Varieties of knowledge elicitation techniques // International Journal of Human-Computer Studies. 1994. Vol. 41. № 6. Pp. 801–849.
8. Di Marzo Serugendo G., Gleizes M.-P., Karageorgos A. Self-organization in Multi-agent Systems // The Knowledge Engineering Review. 2005. Vol. 20. № 2. Pp. 165–189.
9. Ezugwu A.E., Frincu S.B., Adewumi A.O., Buhari S.M., Junaidu S.B. Neural network-based multi-agent approach for scheduling in distributed systems // Concurrency and Computation: Practice & Experience. 2017. Vol. 29. № 1. Режим доступа: <http://dx.doi.org/10.1002/cpe.3887> (дата обращения 25.04.2017).
10. Leitao P., Inden U., Ruckemann C.-P. Parallelising Multi-agent Systems for High Performance Computing // 3rd International Conference on Advanced Communications and Computation, Red Hook NY: IARIA. 2014. Pp. 1–6.
11. Nash J. Equilibrium points in n-person games // Proceedings of the National Academy of Sciences of the United States of America. 1950. Vol. 36. № 1. Pp. 48–49.
12. Talia D. Workflow Systems for Science: Concepts and Tools // ISRN Software Engineering. 2013. Vol. 2013. Режим доступа: <http://dx.doi.org/10.1155/2013/404525> (дата обращения 18.04.2017).
13. Vickrey W. Counterspeculation, Auctions, and Competitive Sealed Tenders // Journal of Finance. 1961. Vol. 16. № 1. Pp. 8–37.

14. Yu J., Buyya R. A Taxonomy of Workflow Management Systems for Grid Computing // Journal of Grid Computing. 2005. Vol. 3. № 3. Pp. 171–200.
 15. Zeng L., Xu L., Shi Z., Wang M., Wu W. Distributed Computing Environment: Approaches and Applications // IEEE International Conference on Systems, Man and Cybernetics: IEEE Publisher. 2007. Pp. 3240–3244.
-

UDK 519.71

**KNOWLEDGE ELICITATION BY AGENTS
IN DISTRIBUTED COMPUTING MANAGEMENT SYSTEM**

Alexander G. Feoktistov

Ph.D. of Engineering Sciences, Associate Professor, Senior Research Officer,
e-mail: agf65@yandex.ru

Roman O. Kostromin

Postgraduate Student, e-mail: romang70055@gmail.com
Matrosov Institute for System Dynamics and Control Theory of SB RAS
134, Lermontov Str., 664033, Irkutsk, Russia

Abstract. The scalable applications management for solving large-scale problems in a heterogeneous distributed computing environment is the non-trivial problem. Applications generate competitive job flows that have to be carried out using shared resources of the environment. The effective approach to solve this problem is to apply multi-agent technologies. In this regard, we develop a multi-agent system for the scalable applications management. In comparison to existing multi-agent systems, our system is based on employing a special conceptual model of the environment. It has various components of a complex knowledge about both the environment and subject domains of solved problems. We provide a new approach to an elicitation of these knowledge components through an integrated use of the conceptual modelling of distributed computing, classification of jobs and resources, and the parameters adjustment for agent algorithms. Practical experiments demonstrate the high speed and efficiency of computations due to using the developed multi-agent system for the scalable applications management.

Keywords: distributed computing, multi-agent management, self-organization, knowledge elicitation, simulation modeling

References

1. Kalyaev A.I., Kalyaev I.A. Metod mul'tiagentnogo dispetchirovaniya resursov v oblachnyh vychislitel'nyh sredah [Method of multiagent scheduling of resources in cloud computing environments] // Izvestija RAN. Teorija i sistemy upravlenija = Proceedings of the RAS. Theory and control systems. 2016. №. 2. Pp. 51–63. (in Russian)

2. Tyugu E.H. *Konceptual'noe programmirovaniye* [Conceptual Programming]. Moscow. Nauka = Science. 1984. 256 p. (in Russian)
3. Feoktistov A.G. *Upravlenie slozhnoj sistemoy na osnove metodologii mnogokriterial'nogo vybora upravljajushhih vozdeystvij* [The management of complex systems based on the methodology of multi-criteria choice of control actions] // *Fundamental'nye issledovaniya = Fundamental Research*. 2015. № 9-1. Pp. 82–86. (in Russian)
4. Feoktistov A.G., Korsukov A.S., Dyadkin Y.A. *Aspekty imitacionnogo modelirovaniya processov mul'tiagentnogo upravlenija raspredelennymi vychislenijami* [The aspects of simulation modeling of processes of multiagent control of distributed computing] // *Informacionnye i matematicheskie tehnologii v nauke i upravlenii = Information and Mathematical Technologies in Science and Management*. 2016. № 4-1. Pp. 118–126. (in Russian)
5. Amato A., Venticinque S. *A Distributed Agent-Based Decision Support for Cloud Brokering. Scalable Computing: Practice and Experience*. 2014. vol. 15. № 1. Pp. 65–78.
6. Bychkov I., Oparin G., Tchernykh A., Feoktistov A., Bogdanova V., Gorsky S. *Conceptual Model of Problem-Oriented Heterogeneous Distributed Computing Environment with Multi-Agent Management* // *Procedia Computer Science*. 2017. Vol. 103. Pp. 162–167.
7. Cooke N. J. *Varieties of knowledge elicitation techniques* // *International Journal of Human-Computer Studies*. 1994. Vol. 41. № 6. Pp. 801-849.
8. Di Marzo Serugendo G., Gleizes M.-P., Karageorgos A. *Self-organization in Multi-agent Systems* // *The Knowledge Engineering Review*. 2005. Vol. 20. № 2. Pp. 165–189.
9. Ezugwu A.E., Frincu S.B., Adewumi A.O., Buhari S.M., Junaidu S.B. *Neural network-based multi-agent approach for scheduling in distributed systems* // *Concurrency and Computation: Practice & Experience*. 2017. Vol. 29. № 1. Available at: <http://dx.doi.org/10.1002/cpe.3887>, accessed 25.04.2017.
10. Leitao P., Inden U., Ruckemann C.-P. *Parallelising Multi-agent Systems for High Performance Computing* // *3rd International Conference on Advanced Communications and Computation*. Red Hook NY: IARIA. 2014. Pp. 1–6.
11. Nash J. *Equilibrium points in n-person game* // *Proceedings of the National Academy of Sciences of the United States of America*. 1950. Vol. 36. № 1. Pp. 48–49.
12. Talia D. *Workflow Systems for Science: Concepts and Tools* // *ISRN Software Engineering*. 2013. vol. 2013. Available at: <http://dx.doi.org/10.1155/2013/404525>, accessed 18.04.2017.
13. Vickrey W. *Counterspeculation, Auctions, and Competitive Sealed Tenders* // *Journal of Finance*. 1961. Vol. 16. № 1. Pp. 8–37.
14. Yu J., Buyya R. *A Taxonomy of Workflow Management Systems for Grid Computing* // *Journal of Grid Computing*. 2005. Vol. 3. № 3. Pp. 171–200.
15. Zeng L., Xu L., Shi Z., Wang M., Wu W. *Distributed Computing Environment: Approaches and Applications* // *IEEE International Conference on Systems, Man and Cybernetics: IEEE Publisher*. 2007. Pp. 3240–3244.