УДК 004.921

Современные многоагентные системы для скрапинга данных

Блюм Владислав Станиславович — канд. техн. наук, доцент кафедры бизнес-информатики и менеджмента. Научные интересы: иммунокомпьютинг, технологии искусственного интеллекта. E-mail: vladblum7@gmail.com

Лапшин Андрей Евгеньевич — магистрант 2-го курса направления 09.04.03 «Прикладная информатика». Научные интересы: интеллектуальный анализ данных. E-mail: andreyka.lapshin.2002@mail.ru

Институт технологий предпринимательства и права, Санкт-Петербургский государственный университет аэрокосмического приборостроения, Россия, 190000, Санкт-Петербург, ул. Большая Морская, 67, лит. А

Для цитирования: Блюм В. С., Лапшин А. Е. Современные многоагентные системы для скрапинга данных // Интеллектуальные технологии на транспорте. 2026. № 1 (45). С. 16–22. DOI: 10.20295/2413-2527-2026-145-16-22

Аннотация. *Рассматриваются архитектура многоагентных систем (МАС), свойства агентов, особенности коммуникации и применимость данного подхода к задачам веб-скрапинга. Актуальность исследования определяется стремительным ростом объемов данных в сети Интернет и ограниченностью классических централизованных систем веб-скрапинга, сталкивающихся с проблемами масштабирования, блокировок и недостаточной устойчивости. В этих условиях возрастает потребность в использовании децентрализованных архитектур, способных адаптироваться к динамичной среде и эффективно собирать большие объемы информации. Одним из наиболее перспективных подходов являются многоагентные системы, обеспечивающие распределенный сбор, обработку и хранение данных.* ***Цель:*** *разработка и структурирование подхода к использованию многоагентных систем для веб-скрапинга, а также описание обобщенного алгоритма, обеспечивающего масштабируемый, отказоустойчивый и адаптивный сбор данных.* ***Методы:*** *теоретический анализ свойств многоагентных систем, архитектурных моделей и коммуникационных механизмов между агентами; изучение существующих практических решений распределенного краулинга; синтез обобщенного алгоритма на основе выделения типовых ролей агентов (планировщик, сборщик, парсер, обработчик данных, агент обхода защиты).* ***Результаты:*** *описана трехуровневая архитектура МАС, включающая уровни сбора, обработки/координации и хранения данных. Выделены ключевые свойства агентов и показаны их роли в задаче скрапинга. Представлены функции пяти типов агентов, применяемых в распределенном веб-скрапинге, и предложена схема взаимодействия между ними. На основе анализа существующих решений сформирован обобщенный алгоритм распределенного скрапинга, отражающий взаимодействие специализированных агентов, который включает этапы инициализации, распределения задач, загрузки страниц, обработки ошибок блокировки, парсинга контента и сохранения данных. Показано, что многоагентный подход обеспечивает параллелизм, масштабируемость, отказоустойчивость и гибкость при работе с веб-ресурсами.* ***Практическая значимость:*** *результаты исследования могут быть использованы при проектировании систем массового сбора данных, построении распределенных веб-краулеров и создании платформ анализа информации на основе МАС. Обобщенный алгоритм может служить основой для реализации гибких и масштабируемых систем, способных эффективно функционировать в условиях больших объемов данных, динамических изменений веб-страниц и наличия защитных механизмов.* ***Обсуждение:*** *в статье описывается интеграция свойств и принципов многоагентных систем в контекст веб-скрапинга с формированием единой обобщенной модели взаимодействия агентов. Представленный алгоритм*

отражает практическую структуру функционирования распределенного краулера и демонстрирует, как различные типы агентов могут обеспечивать координацию, сбор, анализ и фильтрацию данных при работе с динамичными и защищенными веб-ресурсами. Подчеркнута значимость децентрализации и адаптивности для современного веб-скрапинга, включая работу в условиях ограничений, связанных с антибот-защитами.

Ключевые слова: *многоагентные системы, скрапинг, масштабирование, проактивность, автономность*

1.2.1 — *искусственный интеллект и машинное обучение (технические науки)*

Введение

В современном мире объем данных, публикуемых в интернете, растет экспоненциально. Согласно данным Forbes, к 2025 году глобальный объем данных достигнет 175 зеттабайт [1]. Для их обработки бизнес и наука все чаще используют веб-скрапинг — технологию автоматизированного извлечения информации с веб-страниц. Он применяется в маркетинге (мониторинг цен конкурентов), финансовой аналитике (сбор котировок), науке (агрегация данных из PubMed, arXiv) и даже правительственных структурах (мониторинг открытых данных).

Классические решения, построенные на централизованных скриптах, сталкиваются с рядом ограничений:

- затрудненная работа при больших объемах [2];
- блокировки со стороны сайтов (Cloudflare, reCAPTCHA);
- сложности при масштабировании.

Для преодоления этих проблем развивается подход на основе мультиагентных систем (МАС), где вместо одного монолитного скрипта работает распределенная команда агентов. Такой метод обеспечивает гибкость, устойчивость и эффективность при работе с динамичным веб-контентом [3].

Основная часть

Многоагентная система — это система, состоящая из множества автономных взаимодействующих вычислительных единиц, называемых агентами. Каждый агент обладает следующими качествами:

- автономностью — способностью действовать без прямого вмешательства человека;
- реактивностью — способностью воспринимать окружающую среду и реагировать на ее изменения;
- проактивностью — способностью к целенаправленному поведению и преследованию целей;
- социальной способностью — способностью взаимодействовать с другими агентами для решения задач.

В контексте сбора и хранения информации агенты могут быть программными модулями, роботами, датчиками в IoT-сетях или даже бизнес-процессами.

Архитектура такой системы обычно является частным случаем классической трехуровневой модели Presentation Layer — Business Logic Layer — Data Access Layer, широко описанной в литературе по программной инженерии [4].

Уровень сбора данных (агенты-сборщики) отвечает за непосредственное взаимодействие с источниками информации, будь то физические датчики, веб-страницы, API или пользовательский ввод. Эти агенты формируют первичный поток данных, необходимый для последующего анализа. Примерами могут служить дроны, обследующие сельскохозяйственные угодья, программные агенты, отслеживающие биржевые котировки, или температурные датчики в инфраструктуре умного города. Как правило, такие агенты обладают ограниченными вычислительными ресурсами, мобильностью и узкой специализацией, что позволяет им эффективно выполнять одну конкретную задачу.

Уровень обработки и координации (агенты-координаторы или обработчики) получает

и анализирует сырые данные, поступающие от агентов-сборщиков. На этом уровне выполняется предварительная обработка: фильтрация, агрегация, очистка, устранение шума и сжатие данных. Кроме того, агенты этого уровня координируют деятельность сборщиков, распределяя зоны ответственности и оптимизируя нагрузку в системе. В качестве примеров можно привести шлюзы (gateway) в IoT-сетях, серверы, управляющие роем дронов, или координационные центры, обеспечивающие согласованное функционирование распределенных компонентов системы.

Уровень хранения и предоставления доступа (агенты-хранилища или интерфейсные агенты) обеспечивает надежное и структурированное хранение уже обработанных данных. Эти агенты выполняют функции доступа к информации, отвечая на запросы пользователей или взаимодействующих систем. Их роль заключается в предоставлении релевантных данных в удобной форме и поддержании целостности информационного пространства. К типичным примерам относятся распределенные базы данных, реализованные, например, на основе блокчейн-технологий, децентрализованные файловые системы (IPFS), а также программные агенты, предоставляющие интерфейсы взаимодействия через API.

Коммуникация между агентами происходит через специальные языки коммуникации, например FIPA ACL (Agent Communication Language), где сообщения имеют определенную структуру (перформатив, отправитель, получатель, содержание) [5].

Можно выделить следующие ключевые преимущества МАС для этой задачи:

1. Масштабируемость. Новые агенты могут быть легко добавлены в систему без ее полной перестройки. Система может охватывать огромные географические области;

2. Отказоустойчивость и надежность. Отказ одного или нескольких агентов не приводит к коллапсу всей системы. Задачи могут быть динамически перераспределены между другими агентами;

3. Параллелизм и эффективность. Агенты работают параллельно, что значительно ускоряет сбор и обработку данных в больших объемах (например, мониторинг сети Интернет или большого склада);

4. Гибкость и адаптивность. Агенты могут реагировать на изменения в окружающей среде (новые источники данных, поломки, изменение приоритетов) и перестраивать свою работу;

5. Распределенность и автономность. Нет единой точки отказа. Данные могут собираться и обрабатываться на месте (на edge-устройствах), что снижает нагрузку на сеть и задержку.

Многоагентные системы предлагают радикально иной, децентрализованный подход к сбору и хранению информации по сравнению с классическими клиент-серверными моделями [6]. Они идеально подходят для задач, требующих масштабируемости, отказоустойчивости и работы в динамичных распределенных средах. Несмотря на трудности, связанные со сложностью проектирования и безопасностью, именно за МАС будущее в таких областях, как IoT, умные города, Web3 и автономные роботизированные системы.

В [7] предложен подход к разработке и оптимизации системы для сбора и анализа данных из социальных сетей (в частности, Twitter) с использованием распределенного краулера, встроенного в многоагентную систему (Multi-Agent System, MAS). Система предназначена для эффективного извлечения данных на основе ключевых слов.

При этом ключевыми технологиями и методами являются:

1. Веб-краулер. Программа, которая автоматически обходит веб-страницы, загружает их содержимое, извлекает текст и метаданные, а также находит новые ссылки для последующего обхода.

2. Многоагентная система. Иерархическая структура агентов, в которой:

• главный агент (Master Agent) управляет задачами, распределяет URL-адреса и координирует работу;

• агенты-краулеры загружают и анализируют содержимое веб-страниц.

3. Распределенные вычисления. Нагрузка распределяется между множеством агентов, что повышает производительность и отказоустойчивость системы.

Система использует иерархическую архитектуру (дерево агентов), что обеспечивает масштабируемость и балансировку нагрузки. Агенты

обмениваются сообщениями в формате XML с использованием TCP-сокетов. Сообщения могут шифроваться алгоритмом MD5. Поддерживается как распределенное (на нескольких компьютерах), так и локальное выполнение [8–10].

В развитие этой идеи можно выделить алгоритм веб-скрапинга с использованием мультиагентных систем, который отражает практическое применение описанных принципов. Алгоритм скрапинга (веб-скрапинга) с помощью МАС — это эффективный подход к сбору данных в сложных и динамичных условиях.

Классический скрапинг часто представляет собой единый централизованный скрипт. Мультиагентный подход заменяет его роем интеллектуальных агентов, каждый из которых выполняет свою специализированную задачу, координируясь с другими для достижения общей цели — эффективного и надежного сбора данных.

Обычно система состоит из нескольких типов агентов:

- агент-планировщик (Coordinator Agent);
- агент-сборщик (Crawler Agent);
- агент-парсер (Parser Agent);
- агент-обработчик (Data Handler Agent);
- агент — обходчик защиты (Anti-Block Agent).

Агент-планировщик получает целевую задачу (например, «собрать все товары с сайта X»), дробит задачу на подзадачи (списки категорий, страниц), распределяет URL между агентами-сборщиками, а также следит за общей очередью задач.

Агент-сборщик непосредственно загружает веб-страницы, обладает человекоподобным поведением (рандомизация задержек, соблюдение robots.txt), использует пул прокси для смены IP, обходит простые антибот-системы, передает загруженный HTML агенту-парсеру.

Агент-парсер извлекает структурированные данные из сырого HTML, содержит логику парсинга (XPath, CSS-селекторы, регулярные выражения), умеет адаптироваться к небольшим изменениям в верстке. Извлеченные данные передаются агенту-обработчику.

Агент-обработчик выполняет очистку, валидацию и сохранение данных, а также приводит их к единому формату. После проверки данных на полноту и корректность следует сохранение в БД, файл или отправка в очередь сообщений.

Функции агента — обходчика защиты — это решение капчи и обход сложных систем защиты (например, Cloudflare). Может использовать сервисы по распознаванию капч (например, Anti-Captcha, RuCaptcha). Анализирует ответы сервера на наличие JS-челленджей и сообщает другим агентам о необходимости изменить поведение. Этот агент может быть не у всех систем, но для сложных целей он критически важен.

На рис. 1 представлена обобщенная схема взаимодействия агентов, отражающая логику функционирования предложенной системы. Схема демонстрирует основные этапы обмена данными и распределения ролей между агентами.

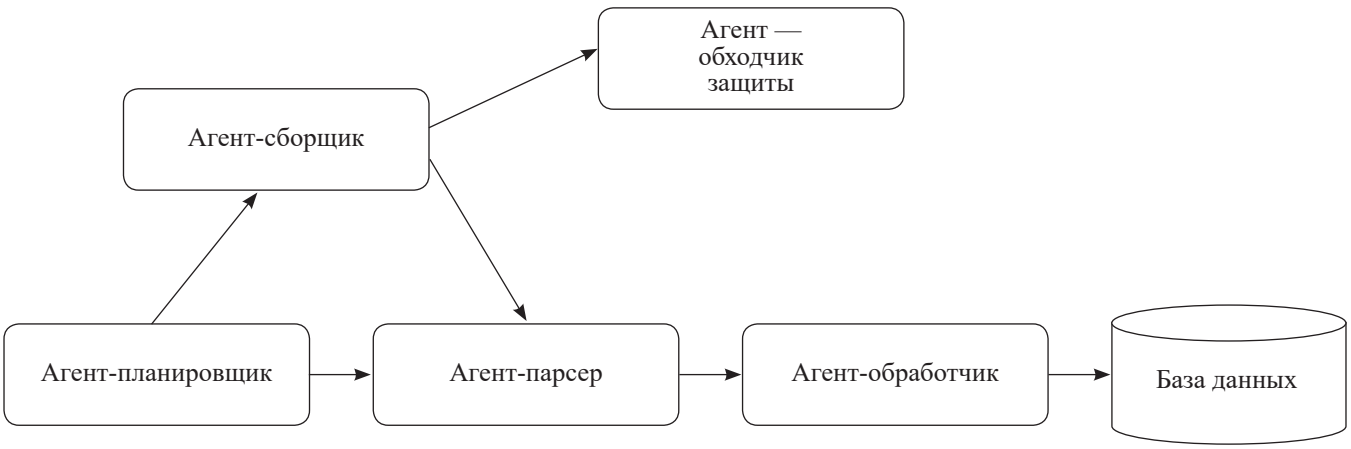

Рис. 1. Схема мультиагентной системы по сбору данных

На основе представленной схемы можно описать последовательность действий агентов в процессе работы системы. Алгоритм функционирования включает следующие этапы:

1. Инициализация входных данных. Формируются стартовые URL и конфигурационные параметры (глубина обхода, задержки, прокси).

2. Создание очереди задач. Агент-планировщик формирует первоначальную очередь на основе стартовых URL.

3. Распределение задач. Планировщик передает свободным агентам-сборщикам ссылки для обработки.

4. Загрузка страниц. Агент-сборщик получает страницу, используя прокси и соблюдая задержки. При возникновении ошибки (капча, код 403) задача передается агенту обхода защиты, после чего запрос повторяется.

5. Парсинг контента. Агент-сборщик передает HTML агенту-парсеру, который извлекает целевые данные и новые ссылки.

6. Обновление очереди. Найденные ссылки, удовлетворяющие фильтрам (тот же домен, допустимая глубина), передаются планировщику для добавления в очередь.

7. Сохранение данных. Агент-обработчик записывает структурированные данные в хранилище.

8. Завершение процесса. Алгоритм останавливается, когда очередь задач пуста и все агенты простаивают. Планировщик отправляет сигнал завершения.

Заключение

В результате анализа можно сделать вывод, что применение мультиагентных систем в задачах веб-скрапинга представляет собой перспективное направление развития интеллектуальных технологий сбора данных. В отличие от традиционных централизованных решений мультиагентный подход обеспечивает масштабируемость, адаптивность и устойчивость к отказам, что особенно важно при работе с динамичными и защищенными веб-ресурсами. Иерархическая структура агентов позволяет эффективно распределять нагрузку, ускоряя процесс извлечения и обработки информации. Практическое использование подобных систем открывает возможности для построения децентрализованных платформ анализа данных, интегрирующих краулинг, обработку и хранение в едином информационном контуре. В дальнейшем развитие таких систем может быть связано с применением методов машинного обучения для интеллектуального управления агентами и повышением уровня автономности при взаимодействии со сложными веб-средами.

СПИСОК ИСТОЧНИКОВ

1. Coughlin T. 175 Zettabytes By 2025 // Forbes. 2018. 27 November. URL: http://www.forbes.com/sites/tomcoughlin/2018/11/27/175-zettabytes-by-2025 (дата обращения: 05.10.2025).

2. Barrett A. How to Scrape Websites at Large Scale // Octoparse Web Scraping Blog. 2022. 30 August. URL: http://www.octoparse.com/blog/scrape-websites-at-large-scale (дата обращения: 05.10.2025).

3. Jennings N. R., Wooldridge M. J. Applications of Intelligent Agents // Agent Technology: Foundations, Applications, and Markets / N. R. Jennings, M. J. Wooldridge (eds). Heidelberg: Springer, 1998. Pp. 3–28. DOI: 10.1007/978-3-662-03678-5_1.

4. Фаулер М. Архитектура корпоративных программных приложений / пер. с англ. М.: Вильямс, 2006. 544 с.

5. De Ridder A. An Introduction to FIPA Agent Communication Language: Standards for Interoperable Multi-Agent Systems // SmythOS AI Blog. URL: http://smythos.com/developers/agent-development/fipa-agent-communication-language (дата обращения: 22.11.2025).

6. Кияев В. И., Граничин О. Н. Информационные технологии в управлении предприятием: краткий учебный курс. 2-е изд., испр. М.: ИНТУИТ, 2016. 361 с.

7. The Data Extraction Using Distributed Crawler Inside the Multi-Agent System / K. Tomala [et al.] // Advances in Electrical and Electronic Engineering, 2013. Vol. 11, no. 6. Pp. 455–460. DOI: 10.15598/aeee.v11i6.867.

8. Extensible Markup Language (XML) 1.0 (Fifth Edition) — W3C Recommendation 26 November 2008 / T. Bray [et al.] (eds). URL: http://www.w3.org/TR/xml (дата обращения: 22.11.2025).

9. Transmission Control Protocol // Wikipedia. URL: http://en.wikipedia.org/wiki/Transmission_Control_Protocol (дата обращения: 22.11.2025).

10. MD5 // Wikipedia. URL: http://en.wikipedia.org/wiki/MD5 (дата обращения: 22.11.2025).

Дата поступления: 29.11.2025
Решение о публикации: 09.02.2026

Modern Multi-Agent Systems for Data Scraping

Vladislav S. Blyum — PhD in Engineering, Associate Professor of the Business Informatics and Management Department. Research interests: immunocomputing, artificial intelligence technologies. E-mail: vladblum7@gmail.com

Andrey E. Lapshin — 2nd year Master's Degree Student in 09.04.03 Applied Informatics. Research interests: data mining. E-mail: andreyka.lapshin.2002@mail.ru

Institute of Entrepreneurship Technologies and Law, Saint Petersburg State University of Aerospace Instrumentation, Bolshaya Morskaya str., 67, lit. A, Saint Petersburg, 190000, Russia

For citation: Blyum V. S., Lapshin A. E. Modern Multi-Agent Systems for Data Scraping, *Intellectual Technologies on Transport*, 2026, no. 1 (45), pp. 16–22. DOI: 10.20295/2413-2527-2026-145-16-22. (In Russian)

Abstract. *This paper examines the architecture of multi-agent systems (MAS), agent properties, communication features, and the applicability of this approach to web scraping tasks. The relevance of the study is determined by the rapid growth of data volumes on the Internet and the limitations of traditional centralized web scraping systems that encounter challenges related to scalability, blocking, and insufficient robustness against dynamic website changes. In this context, there is an increasing demand for decentralized architectures that adapt to evolving environments and efficiently collect vast quantities of information. One of the most promising approaches is the deployment of multi-agent systems, which enable distributed data collection, parallel processing, and resilient storage.* ***Purpose:*** *to develop and structure an approach for utilizing multi-agent systems in web scraping, as well as to describe a generalized algorithm that ensures scalable, fault-tolerant, and adaptive data collection.* ***Methods:*** *the study employs theoretical analysis of multi-agent system properties, architectural models, and inter-agent communication mechanisms; an examination of existing practical implementations of distributed web crawling; and the synthesis of a generalized algorithm constructed upon the identification of typical agent roles: scheduler, collector, parser, data processor, and protection bypass agent.* ***Results:*** *the findings reveal a three-tiered architecture for the multi-agent system, including levels for data collection, processing/coordinating, and storage. Key properties of agents are highlighted, demonstrating their distinct contributions to the scraping task. The functions of five types of agents used in distributed web scraping are presented, alongside a proposed interaction scheme illustrating their collaborative engagement. Based on the analysis of existing solutions, a generalized algorithm for distributed scraping has been formulated, reflecting the interaction of these specialized agents. This algorithm encompasses distinct stages: initialization, task distribution, page loading, error handling in blocking scenarios, content parsing, and data storage. The findings indicate that the multi-agent approach provides parallelism, scalability, fault tolerance, and flexibility,*

adapting to diverse web resources and evolving challenges. ***Practical significance:*** *the results of this research can be used in the design of mass data collection systems, the construction of distributed web crawlers, and the creation of information analysis platforms based on multi-agent systems. The generalized algorithm can serve as the basis for implementing flexible and scalable systems capable of functioning effectively in the context of vast data volumes, dynamic web page alterations, and robust protective mechanisms.* ***Discussion:*** *this article describes the integration of multi-agent system properties and principles into web scraping processes, culminating in the formation of a unified generalized model of agent interaction. The presented algorithm mirrors the practical structure of a distributed crawler and demonstrates how different types of agents can coordinate, collect, analyze, and filter data when interacting with dynamic and secure web resources. The importance of decentralization and adaptability for modern web scraping is emphasized, particularly in scenarios constrained by anti-bot protection.*

Keywords: *multi-agent systems, scraping, scaling, proactivity, autonomy*

REFERENCES

1. Coughlin T. 175 Zettabytes By 2025, *Forbes*. Published online at November 27, 2018. Available at: http://www.forbes.com/sites/tomcoughlin/2018/11/27/175-zettabytes-by-2025 (accessed: October 05, 2025).

2. Barrett A. How to Scrape Websites at Large Scale, *Octoparse Web Scraping Blog*. Published online at August 30, 2022. Available at: http://www.octoparse.com/blog/scrape-websites-at-large-scale (accessed: October 05, 2025).

3. Jennings N. R., Wooldridge M. J. Applications of Intelligent Agents. In: *Jennings N. R., Wooldridge M. J. (eds) Agent Technology: Foundations, Applications, and Markets.* Heidelberg, Springer, 1998, pp. 3–28. DOI: 10.1007/978-3-662-03678-5_1.

4. Fowler M. Arkhitektura korporativnykh programmnykh prilozheniy [Patterns of enterprise application architecture]. Moscow, Williams Publishing House, 2006, 544 p. (In Russian)

5. De Ridder A. An Introduction to FIPA Agent Communication Language: Standards for Interoperable Multi-Agent Systems, *SmythOS AI Blog*. Available at: http://smythos.com/developers/agent-development/fipa-agent-communication-language (accessed: November 22, 2025).

6. Kiyaev V. I., Granichin O. N. Informatsionnye tekhnologii v upravlenii predpriyatiem: kratkiy uchebnyy kurs [Information Technology in Business Management: A Concise Educational Course]. Moscow, INTUIT, 2016, 361 p. (In Russian)

7. Tomala K., et al. The Data Extraction Using Distributed Crawler Inside the Multi-Agent System, *Advances in Electrical and Electronic Engineering*, 2013. Vol. 11, no. 6. Pp. 455–460. DOI: 10.15598/aeee.v11i6.867.

8. Bray T., et al. (eds) Extensible Markup Language (XML) 1.0 (Fifth Edition) — W3C Recommendation 26 November 2008. Available at: http://www.w3.org/TR/xml (accessed: November 22, 2025).

9. Transmission Control Protocol, *Wikipedia*. Available at: http://en.wikipedia.org/wiki/Transmission_Control_Protocol (accessed: November 22, 2025).

10. MD5, *Wikipedia*. Available at: http://en.wikipedia.org/wiki/MD5 (accessed: November 22, 2025).

Received: 29.11.2025
Accepted: 09.02.2026