

**МОДЕЛИ И МЕТОДЫ МАРШРУТИЗАЦИИ
ИНФОРМАЦИОННЫХ РЕСУРСОВ В СЕТЯХ
ДОСТАВКИ КОНТЕНТА НА ОСНОВЕ
ОТЕЧЕСТВЕННОГО ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ**

Прокуровский А. А.¹, Тутова Н. В.², Андреев И. А.³
*(Московский технический университет связи
и информатики, Москва)*

Сети доставки контента предназначены для ускорения загрузки веб-сайтов, приложений и другого цифрового контента для пользователей, находящихся в различных регионах. Основная цель таких сетей заключается в том, чтобы уменьшить задержку при передаче данных между сервером и пользователем, обеспечивая быструю доставку контента независимо от географического положения пользователя. Импортзамещение сетей доставки контента является актуальной и важной научно-технической задачей. В работе предложены модели и методы эффективного распределения цифровых информационных ресурсов в сети доставки контента эффективного распределения цифровых информационных ресурсов в сети доставки контента с кэшированием контента на пограничных серверах. Сеть доставки контента впервые реализована в отечественном программном продукте 1С:Шина. Для решения поставленных задач оптимизации использовался муравьиный алгоритм, который позволяет найти эффективное решение в вычислительных сложных задачах с большой размерностью за приемлемое время. Приведено описание эксперимента с использованием существующей инфраструктуры сети доставки контента и показана эффективность предложенного метода. По сравнению с случайным формированием маршрута предложенные методы формирования маршрута позволяют ускорить загрузку контента до 20%. Многопоточная передача позволила ускорить загрузку контента еще в два раза.

Ключевые слова: сеть доставки контента, распределение контента, цифровые ресурсы, муравьиный алгоритм, поиск минимального пути, 1С:Шина.

¹ Алексей Андреевич Прокуровский, аспирант (a.a.prokurovskiy@mtuci.ru).

² Наталья Владимировна Тутова, к.т.н., доцент (n.v.tutova@mtuci.ru).

³ Илья Александрович Андреев, к.э.н. (kis.dep@mtuci.ru).

1. Введение

Сети доставки контента (Content Delivery Network, CDN) предназначены для ускорения загрузки веб-сайтов, приложений и другого цифрового контента для пользователей по всему миру. CDN широко используются в приложениях, где требуется быстрая доставка видеоконтента пользователям, таких как видеохостинги (YouTube, RuTube), социальные сети, онлайн-магазины, игровые платформы, где важна минимальная задержка при передаче игровых данных. Среди крупнейших сетей доставки контента можно выделить Akamai CDN, которая обрабатывает от 20% до 30% всего веб-трафика. Другим примером сети является Amazon Cloudfront, поддерживающая более 100 пограничных узлов по всему миру [18].

CDN кэширует часто запрашиваемые статические файлы, например, изображения, видео, CSS-файлы, JavaScript-коды, что ускоряет их повторную загрузку. Когда пользователь повторно обращается к сайту, ему передается уже закэшированный контент с ближайшего сервера, что уменьшает нагрузку на исходный сервер и сокращает время отклика. За счет резервирования повышается надежность и отказоустойчивость сети.

Сети доставки контента находят применение и в новых приложениях, таких как мониторинг транспортных средств, мониторинг беспилотных летательных аппаратов, умное здравоохранение, умные города и электронное обучение [18].

В области электронного обучения были разработаны отдельные прототипы CDN [14, 16], но они были построены на иностранном программном обеспечении.

Однако в текущих условиях возникла необходимость импортозамещения и разработки отечественного программного обеспечения, что указано в соответствующих национальных программах «Цифровая экономика Российской Федерации» и «Экономика данных и цифровая трансформация государства».

1С:Шина – это технология, разработанная компанией «1С» для обеспечения взаимодействия между различными программными продуктами линейки 1С через защищенные каналы связи. Она позволяет организовать обмен данными и сервисами между

приложениями, работающими на разных платформах и в различных сетях, обеспечивая при этом безопасность передачи данных. В настоящее время 1С:Шина является важным инструментом для создания комплексных информационных систем на базе продуктов 1С, обеспечивающим надёжную и безопасную связь между ними.

В данной работе представлены модели и методы маршрутизации контента и их реализация в сеть доставки контента для дистанционного обучения на основе отечественного программного обеспечения 1С:Шина.

2. Выбор кэширующего сервера и маршрутизация контента в CDN

В CDN можно выделить следующие подсистемы, отвечающие за функционирование сети [12, 18].

- Подсистема перенаправления запроса на выбранный сервер. Обеспечивает перенаправление запроса на ближайший подходящий кэширующий сервер.
- Подсистема распределения и управления контентом. Отвечает за управление кэшами, обновление данных на серверах и другие подобные задачи.
- Подсистема учета и аналитики. Проводит измерения параметров работы сети, собирает данные о доступе к серверам и использовании контента.

Существует три основные модели для распространения контента с исходных серверов на пограничные серверы [13, 15].

- push-модель: контент заранее распространяется, если ожидается его запрос на пограничном сервере;
- pull-модель: пограничный сервер извлекает контент с исходного сервера при получении запроса от конечного пользователя; такая модель используется в большинстве CDN в связи с простотой реализации;
- гибридная push-pull-модель динамически адаптируется к изменяющимся запросам конечного пользователя, проактивно проталкивая часть контента и реактивно извлекая другой кон-

тент. В частности, в сети Akamai для определения маршрута доставки контента используется фирменный алгоритм SureRoute [17]. Принцип работы данного алгоритма заключается в следующем: на исходном сервере размещается тестовый объект. С заданной периодичностью проводятся «гонки» по извлечению данного объекта пограничным сервером с источника по трем доступным маршрутам. Минимальный по длительности маршрут отмечается и считается основным до тех пор, пока не будет найден другой маршрут. Недостатком такого подхода является то, что создается дополнительный трафик для постоянного тестирования маршрутов, что нагружает сеть.

В специализированных CDN гибридные алгоритмы могут дать лучший результат, если можно сделать достаточно точный прогноз использования контента. В данной работе предложены модели и методы маршрутизации контента в CDN для дистанционного обучения на основе гибридной модели кэширования.

3. Постановка задачи маршрутизации контента в CDN с кэшированием контента на промежуточных узлах

3.1. МОДЕЛЬ МАРШРУТИЗАЦИИ В ВИДЕ СВЯЗНОГО ОРИЕНТИРОВАННОГО ГРАФА

Сформулируем задачу распределения контента по всем узлам сети *CDN* в форме задачи оптимизации на графах. Рассмотрим связный ориентированный граф $G = (V, E, h)$ (рис. 1), в котором $V = \{v_1, v_2, \dots, v_n\}$ – конечное множество вершин, $E = \{e_1, e_2, \dots, e_m\}$ – конечное множество дуг, $h : E \rightarrow Z_+$ – весовая функция дуг.

Контент распространяется по всем узлам сети, и желательно это сделать оптимальным образом с учетом значений весовой функции дуг: $c_{ij} = h(e_k)$, где дуга $e_k \in E$ соответствует упорядоченной паре вершин (v_i, v_j) . Значения весовой функции дуг $c_{ij} = h(v_i, v_j)$ соответствуют длине участка сети от i -го узла до j -го. Длина любого подмножества дуг $E_k \subset E$ в графе G равна сумме весов дуг, входящих в это подмножество.

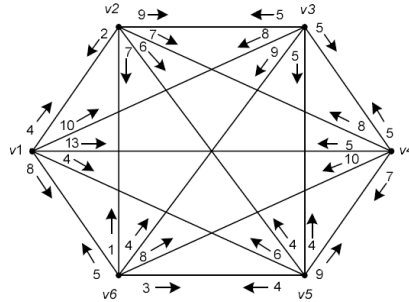


Рис. 1. Иллюстрация задачи маршрутизации контента в CDN
в виде связанного ориентированного графа

Требуется определить такое подмножество дуг, которое образует в графе G замкнутый путь, проходит через каждую вершину ровно один раз и обладает минимальной длиной.

Не все узлы CDN могут быть включены в рассмотрение. Это зависит от прогноза использования контента на кэширующих узлах. Например, если пользователи, записанные на онлайн-курс, просмотрели первый урок, то с большой степенью вероятности они перейдут ко второму уроку. Главное, чтобы граф оставался связным, что в условиях работы в интернете легко реализуется.

Данная постановка задачи соответствует известной задаче коммивояжера и может быть представлена в виде модели булева программирования.

Пусть булевы переменные x_{ij} означают, входит ли дуга (v_i, v_j) в искомый маршрут. Если переменная $x_{ij} = 1$, то дуга (v_i, v_j) входит, т.е. контент передается непосредственно из i -го узла в j -й узел, а если $x_{ij} = 0$, то соответствующая дуга не входит в маршрут.

Тогда математическая постановка задачи маршрутизации контента с кэшированием на всех узлах сети может быть сформулирована следующим образом:

$$(1) \quad \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min, \quad x \in \Delta_\beta,$$

где множество допустимых альтернатив Δ_β формируется следующей системой ограничений:

$$(2) \quad \left\{ \begin{array}{l} \sum_{j=1}^n x_{ij} = 1 \quad (\forall i \in \{1, 2, \dots, n\}); \\ \sum_{i=1}^n x_{ij} = 1 \quad (\forall j \in \{1, 2, \dots, n\}); \\ u_i - u_j + n \cdot x_{ij} \leq n - 1 \quad (\forall i, j \in \{2, 3, \dots, n\}, i \neq j); \\ x_{ij} \in \{0, 1\}, \quad (\forall i, j \in \{1, 2, \dots, n\}); \\ u_i \in R^1, \quad (\forall i \in \{2, 3, \dots, n\}). \end{array} \right.$$

В данной математической модели используются также вспомогательные переменные u_i ($\forall i \in \{2, 3, \dots, n\}$), которые могут принимать любые действительные значения. При этом первое и второе ограничения в системе (2) обеспечивают выполнение условий, что искомый путь должен проходить через каждую вершину графа ровно один раз. Третья группа ограничений обеспечивает выполнение условий, что искомый путь не должен распадаться на отдельные циклы. Всего в задаче $n^2 - n + 2$ ограничений.

3.2. МОДЕЛЬ МАРШРУТИЗАЦИИ В ВИДЕ НЕСВЯЗНОГО ОРИЕНТИРОВАННОГО ГРАФА

Рассмотрим ориентированный граф, представленный на рис. 2, также состоящий из вершин, дуг и весовых коэффициентов дуг $G = (V, E, h)$.

Отличием данного графа от представленного на рис. 1, является его несвязность. Маршрут всегда начинается в вершине v_s , соответствующей основному узлу сети CDN, и заканчивается в конечной вершине v_t , соответствующей наиболее отдаленному серверу от основного узла. Длина любого маршрута в графе равна сумме весов дуг, входящих в этот маршрут. Веса дуг $c_{ij} = h(v_i, v_j)$ назначаются с учетом прогноза потребности пользователей контента в ближайшем будущем. Для образовательного контента данная потребность может быть оценена исходя из записей пользователей того или иного региона на онлайн-курсы.

Контент передаётся от узла к узлу сети и кэшируется на всех промежуточных узлах. Требуется определить маршрут минимальной длины из начальной вершины v_s в конечную вершину v_t .

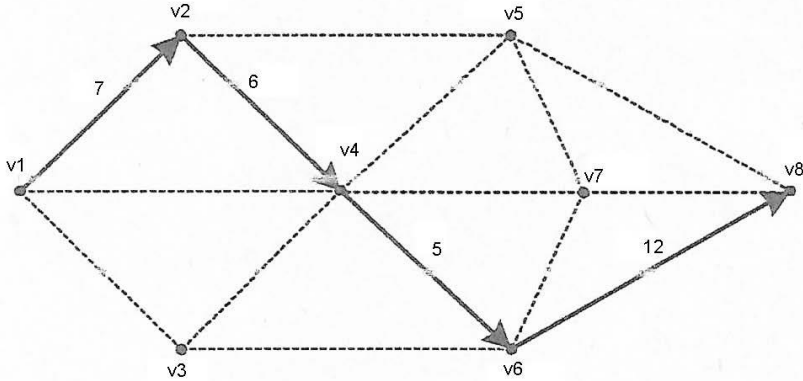


Рис. 2. Иллюстрация задачи нахождения минимального пути кэширования контента

Математическая постановка задачи о минимальном маршруте контента может быть сформулирована следующим образом:

$$(3) \quad \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min, \quad x \in \Delta_\beta$$

где множество допустимых альтернатив Δ_β формируется следующей системой ограничений:

$$(4) \quad \left\{ \begin{array}{l} \sum_{j=1}^n x_{sj} - \sum_{i=1}^n x_{is} = 1; \\ \sum_{j=1}^n x_{tj} - \sum_{i=1}^n x_{it} = -1; \\ \sum_{j=1}^n x_{ij} - \sum_{j=1}^n x_{ji} = 0 \quad (\forall i \in \{1, 2, \dots, n\}, i \neq s, i \neq t); \\ x_{ij} \in \{0, 1\}, \quad (\forall i, j \in \{1, 2, \dots, n\}). \end{array} \right.$$

При этом первое ограничение (4) требует выполнения следующего условия – искомый путь должен начинаться в вершине v_s , второе ограничение задаёт, что искомый путь должен заканчиваться в вершине v_t . Третье ограничение гарантирует связность минимального пути, т.е. искомый путь должен проходить через промежуточные вершины графа. Общее количество ограничений (4) равно n .

4. Метод решения задач оптимизации

Метод муравьиных колоний – это один из эвристических методов оптимизации, который основан на поведении муравьёв в природе. Данный метод используется для решения различных задач оптимизации, таких как задача коммивояжера, задача маршрутизации и другие [1–3, 6–9]. В этих задачах необходимо найти оптимальный маршрут или распределение ресурсов, удовлетворяющее определённым ограничениям.

В основе метода лежит идея о том, что муравьи способны находить кратчайший путь от муравейника до источника пищи благодаря феромонам – особым химическим веществам, которые они оставляют на земле. Когда муравей находит источник пищи, он возвращается в муравейник, оставляя за собой след из феромонов. Другие муравьи следуют по этому следу и также находят пищу. Со временем след из феромонов испаряется и муравьи начинают искать новые пути. Если какой-то путь оказывается более коротким, то количество муравьёв, следующих по нему, увеличивается, а след из феромонов на этом пути становится более «заметным». Таким образом, муравьи постепенно находят кратчайший путь до источника пищи.

Суть метода заключается в следующем.

- Создаётся популяция муравьёв (или агентов), каждый из которых представляет возможное решение задачи.
- Муравьи перемещаются по графу задачи, выбирая следующий узел на основе вероятности, которая зависит от количества феромона на ребре и его качества (например, длины ребра).

- После каждого перемещения муравья количество феромона на рёбрах графа обновляется в соответствии с качеством найденного решения.

- Процесс повторяется до тех пор, пока не будет найдено приемлемое решение или не будет выполнено заданное количество итераций.

Преимуществом метода является его простота и эффективность при решении сложных задач оптимизации. Однако метод может потребовать большого количества итераций для нахождения хорошего решения. Параметры (скорость испарения, количество феромонов) могут сильно влиять на результаты. Иногда подвержен преждевременной сходимости (закрепляется на неоптимальном пути).

Рассмотрим основные блоки схемы алгоритма определения маршрута методом муравьиной колонии, приведенной на рис. 3:

- Начало: старт алгоритма.
- Инициализация параметров: на этом этапе задаются начальные параметры для работы алгоритма, такие как количество муравьёв, количество феромона и другие настройки.
- Качественный критерий пройден – проверяется, достигнуто ли удовлетворительное решение. Если да, алгоритм завершается.
- k = числу муравьёв – проверяется, прошло ли необходимое количество итераций для каждого муравья. Если нет, то происходит выбор следующего маршрута для муравья.
- Выбор маршрута k , оценка его стоимости: для текущего муравья k .
- k происходит выбор маршрута и оценка его стоимости (длины, стоимости и т.д.).
- Маршрут $k <$ текущего – проверка, является ли найденный маршрут короче текущего лучшего маршрута.
- Если да, феромон не добавляется.
- Если нет, происходит расчёт количества наносимого феромона.
- Нанесение феромона на маршрут: обновление количества феромонов на графе.

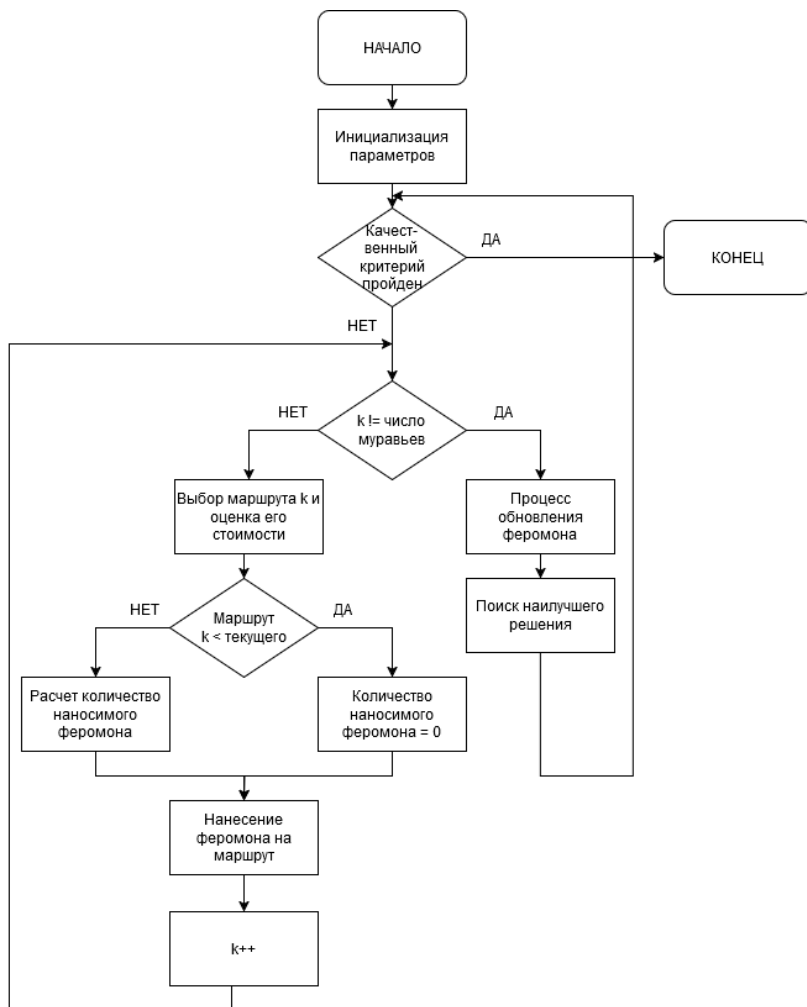


Рис. 3. Блок-схема алгоритма определения маршрута методом муравьиной колонии

- $k++$: увеличение счётчика муравьёв.
- Процесс обновления феромона: после того как все муравьи прошли свои маршруты, обновляется информация о феромонах.

- Поиск наилучшего решения: определяется текущее лучшее решение, исходя из найденных маршрутов.
- Конец: завершение работы алгоритма.

Таким образом, блок-схема описывает процесс итерационного поиска оптимального решения с использованием муравьиных агентов.

Вероятность выбора пути муравьём

Каждый муравей выбирает следующий узел (маршрут) на основе вероятности, которая зависит от количества феромонов и длины пути. Формула для вероятности выбора муравьём пути j из узла i такова:

$$P_{ij}(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{k \in N_i} [\tau_{ik}(t)]^\alpha \cdot [\eta_{ik}]^\beta},$$

где $P_{ij}(t)$ – вероятность, что муравей выберет узел j из узла i на итерации t ; $\tau_{ij}(t)$ – количество феромона на ребре (i, j) на итерации t ; $\eta_{ij} = \frac{1}{d_{ij}}$ – эвристическая информация, обратная длине

пути d_{ij} ; α – параметр, влияющий на важность феромона; β – параметр, влияющий на важность длины пути; N_i – множество доступных узлов для перехода из узла i .

Обновление феромонов

После того как все муравьи завершили свой путь, феромоны на каждом ребре обновляются по формуле

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \sum_k \Delta\tau_{ij}^k(t),$$

где $\tau_{ij}(t+1)$ – новое значение феромона на ребре (i, j) ; ρ – коэффициент испарения феромона ($0 < \rho < 1$); $\Delta\tau_{ij}^k(t)$ – количество феромона, добавляемое муравьём k на ребро (i, j) , которое определяется как

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L_k}, & \text{если муравей } k \text{ прошёл через ребро } (i, j), \\ 0 & \text{иначе,} \end{cases}$$

где Q – константа, представляющая общее количество феромонов, которые муравей может оставить; L_k – длина пути, пройденного муравьём k .

Испарение феромонов

Процесс испарения феромонов необходим для того, чтобы алгоритм не закикливался на локальных решениях и мог исследовать новые пути. Этот процесс моделируется уменьшением количества феромонов на каждом ребре:

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t).$$

Чем больше ρ , тем быстрее испаряются феромоны.

Функция стоимости (длина пути)

Для каждого муравья рассчитывается длина пути L_k , которую он прошёл за одну итерацию:

$$L_k = \sum_{(i,j) \in \text{путь муравья } k} d_{ij},$$

где d_{ij} – длина ребра между узлами i и j .

Лучший путь

На каждой итерации алгоритма также может сохраняться лучший найденный путь. Это используется для поиска глобально оптимального решения. Лучший путь B может обновляться либо каждый раз, когда найдено лучшее решение, либо с учётом нескольких итераций:

$$B = \min (L_k),$$

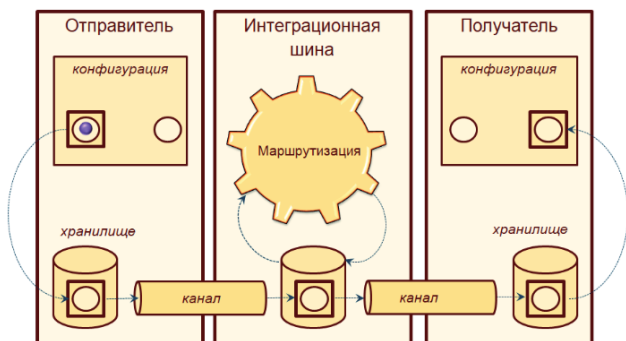
где L_k – длина пути, пройденного муравьём k .

5. Вычислительный эксперимент

5.1. ТЕСТОВЫЙ СТЕНД

Предложенный метод маршрутизации контента был внедрен в сеть доставки контента учебного центра 1С, служащую для распространения обучающих материалов, преимущественно в формате видеоуроков. Сеть имеет более 50 физических географически распределенных серверов в различных городах России. CDN реализована на интеграционной шине «1С:Шина», построенной на базе стандарта открытой архитектуры SOA, что обеспечивает возможность её применения не только для объ-

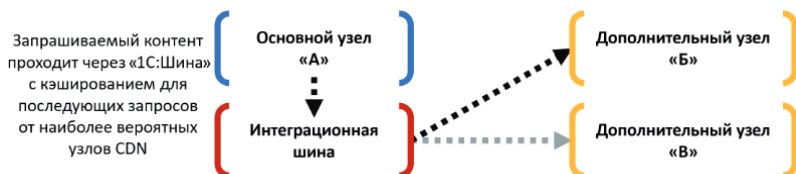
единения различных систем на платформе 1С, но и для взаимодействия с внешними системами [4]. Шина поддерживает открытые протоколы, такие как SOAP, REST, JMS и другие, благодаря чему она совместима с разными архитектурными решениями. Принцип функционирования интеграционной шины «1С:Шина» показан на рис. 4.



*Рис. 4. Принцип функционирования интеграционной шины
«1С:Шина»*

1С:Шина выполняет функции маршрутизатора данных между узлами, центра управления кэшированием и аналитического центра для мониторинга и оптимизации нагрузки на узлы сети доставки контента [11].

На рис. 5 схематично представлена интеграция 1С:Шины в CDN.



*Рис. 5. Схематическое представление интеграции 1С:Шины
в CDN*

В случае запроса контента узлом «Б» с основного сервера данные передаются через 1С:Шину по протоколу AMQP. При наличии промежуточных кэширующих узлов запрашиваемый контент будет кэшироваться на всех промежуточных узлах.

Для проведения экспериментов использовалась инфраструктура сети доставки контента, имеющая 57 физических узлов. Для масштабирования сети до 500 узлов были использованы контейнеры Docker, имеющие техническую конфигурацию, приведенную в таблицах 1 и 2. Использование принципа контейнеризации позволяет значительно повысить показатель устойчивости сети. Это связано с возможностью дополнительной инициализации необходимого количество контейнеров Docker в тех случаях, когда нагрузка на сеть характеризуется как повышенная или критическая.

Таблица 1. Конфигурация сервера с интеграционной шиной

Параметр	Значение
Операционная система	Linux (Ubuntu 24.0)
Оперативная память	8 GB RAM
Центральный процессор	4 ядра по 2.4 ГГц
Внешняя память	100 Гб (SSD)
Веб-сервер	Nginx
Интеграционная шина	1С:Шина

Таблица 2. Конфигурация кэширующих серверов сети доставки контента

Параметр	Значение
Операционная система	Linux (Ubuntu 24.0)
Оперативная память	4 GB RAM
Центральный процессор	4 ядра по 2.4 ГГц
Внешняя память	80 Гб (SSD)
Веб-сервер	Nginx
API сети доставки контента	Скрипт-файлы «1С:Исполнитель»

Для условного отображения построенной сети представлена схема-аналог из пяти узлов (рис. 6).



Рис. 6. Диаграмма тестовой сети доставки контента

5.2. МАСШТАБИРУЕМОСТЬ МУРАВЬИНОГО АЛГОРИТМА

Метод маршрутизации контента, использующий муравьиный алгоритм, реализован в интеграционной шине. Для того чтобы оценить масштабируемость муравьиного алгоритма было зафиксировано время формирования маршрута с числом узлов от 50 до 500. Результаты приведены на рис. 6.



Рис. 7. Время просчета маршрута в зависимости
от размерности задачи

Муравьиный алгоритм позволяет получить маршрут за полиномиальное время ($O(n^2)$). Время просчета маршрута для 500 узлов составило 1 минуту 36 секунд, что позволяет использовать его в больших сетях.

5.3. ЭФФЕКТИВНОСТЬ МЕТОДА МАРШРУТИЗАЦИИ

Для решения задачи было проведено несколько итераций эксперимента.

Сначала было реализовано кэширование контента на всех узлах с поэтапным распространением контента без использования какого-либо метода маршрутизации контента, когда каждый последующий промежуточный узел, на который будет передаваться файл, выбирался случайным образом. Далее был реализован предложенный метод маршрутизации.

В ходе экспериментов использовалось 50 файлов. Размер файлов варьировался от 1 Кб до 1 Гб. Скорость передачи и приема на узлах была ограничена 15 Мбайт/с. Передача осуществлялась последовательно. Фиксировалось суммарное время на передачу всех файлов. Максимальное число серверов – 500, а длительность загрузки составила 7 часов 20 минут для случайного формирования маршрута и 6 часов 14 минут предложенным методом.

Результаты эксперимента приведены на рис. 8. После внедрения алгоритма среднее время загрузки контента на кэширующие серверы уменьшилось в среднем на 17–20%.

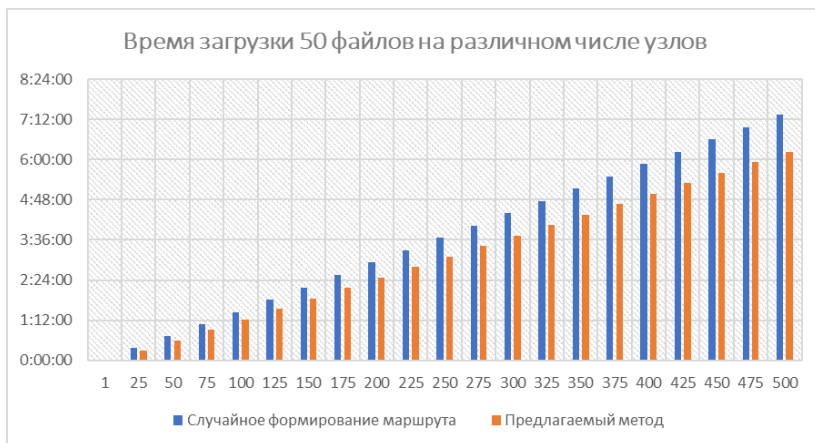


Рис. 8. Результаты эксперимента внедрения в CDN метода маршрутизации контента

Для повышения скорости загрузки контента была реализована многопоточная передача с каждого узла. Результаты эксперимента по передаче файла размером 1 Гб приведены в таблице 3 и на рис. 9.

Таблица 3. Время загрузки файла при однопоточной и многопоточной передаче в зависимости от числа узлов

Число узлов	Однопоточная передача	Многопоточная передача
1	00:00:25	00:00:14
5	00:02:06	00:01:10
10	00:04:13	00:02:20
15	00:06:19	00:03:31
20	00:08:25	00:04:41
25	00:10:32	00:05:51
30	00:12:38	00:07:01
35	00:14:44	00:08:11
40	00:16:51	00:09:21
45	00:18:57	00:10:32
50	00:21:03	00:11:42

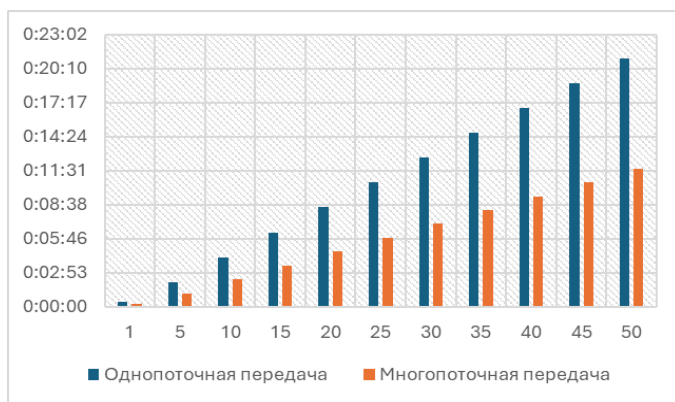


Рис.9. Длительность загрузки файла размером 1 Гб при однопоточной и многопоточной передаче в зависимости от числа узлов

Как видно, при многопоточной передаче длительность загрузки файлов сократилась еще почти в два раза.

6. Заключение

В работе предложены модели и методы маршрутизации контента в сети доставки контента, реализованной на отечественном программном продукте 1С:Шина. В основу методов положены задачи оптимизации коммивояжера и о минимальном пути в направленном графе. Для решения поставленных задач использовался муравьиный метод, который позволяет найти эффективное решение в NP-полных задачах с большой размерностью за полиномиальное время. Приведено описание эксперимента с использованием реальной сети доставки контента и показана эффективность предложенных методов. По сравнению с случайным формированием маршрута предложенные методы формирования маршрута позволяют ускорить загрузку контента до 20%. Многопоточная передача позволила ускорить загрузку контента еще в два раза.

Литература

1. БОРОНИХИНА Е.А. и др. *Исследование гибридных эвристических методов решения задачи коммивояжера* // Материалы IV Международной молодежной научной конференции «Математическое и программное обеспечение информационных, технических и экономических систем». – 2016. – С. 7–10.
2. ВАСИЛЬЕВА Л.И., СУГАЛ Ф.Н., КАРТАК В.М. *Обзор алгоритмов построения маршрутов на примере задачи коммивояжера* // Современные проблемы и перспективы развития естествознания. – 2020. – С. 64–68.
3. ВАТУТИН Э.И., ТИТОВ В.С. *Анализ результатов применения алгоритма муравьиной колонии в задаче поиска пути в графе при наличии ограничений* // Известия Южного федерального университета. Технические науки. – 2014. – №12(161). – С. 111–120.

4. ДИГО С.М., НУРАЛИЕВ Б.Г. *Взаимодействие экосистемы ИС с системой образования в эпоху цифровой экономики* // Новые информационные технологии в образовании. – 2022. – С. 7–24.
5. ИТУРРИАГА ФАБРА С.Д., НЕСМАЧНОВ КАНОВАС С.Е., ГОНИ БОФРИСКО Н. и др. *Конструирование и оптимизация сетей распространения контента* // Труды ИСП РАН. 2019. – №2. – URL: <https://cyberleninka.ru/article/n/konstruirovanie-i-optimizatsiya-setey-rasprostraneniya-kontenta> (дата обращения: 02.01.2025).
6. ЛЕБЕДЕВ Б. и др. *Моделирование адаптивного поведения муравьиной колонии при поиске решений, интерпретируемых деревьями* // Известия Южного федерального университета. Технические науки. – 2012. – Т. 132, №7. – С. 27–34.
7. МЕЛЕХИН В.Б., ХАЧУМОВ М.В. *Об одном подходе решения задачи коммивояжера для планирования автономным беспилотным летательным аппаратом маршрутов облета целей* // Вестник Дагестанского государственного технического университета. Технические науки. – 2021. – Т. 48. – №1. – С. 108–118.
8. МУХИНА Д.А. *Муравьиный алгоритм в решении задачи маршрутизации транспорта* // В сб.: Информационные технологии и высокопроизводительные вычисления: Материалы IV Всероссийской научно-практической конференции. – 2017. – С. 91–103.
9. ПОПОВ Д.В. *Некоторые аспекты развития практических методов решения задач класса NP* // Исторические, философские, методологические проблемы современной науки. – 2018. – С. 254–258.
10. ПРОКУРОВСКИЙ А.А., АНДРЕЕВ И.А., БУЛАНОВ Г.А. *Архитектура сети доставки контента на основе технологии блокчейна* // Технологии информационного общества. – 2022. – С. 245–247.
11. ПРОКУРОВСКИЙ А.А., АНДРЕЕВ И.А., ТУТОВА Н.В. *Повышение эффективности распределения ресурсов в сетях доставки образовательного контента с использовани-*

ем интеграционной шины «IC: Шина» // Информатика и образование. – 2024. – Т. 39, №4. – С. 90–95.

12. BUYYA R., PATHAN M., VAKALI A. (eds.). *Content delivery networks*. – Springer Science & Business Media, 2008. – Vol. 9.
13. CARLSSON N. et al. *Caching and optimized request routing in cloud-based content delivery systems* // Performance Evaluation. – 2014. – Vol. 79. – P. 38–55.
14. ESTEVE M. et al. *A streaming content distribution network for e-learning support* // Interactive Technology and Smart Education. – 2006. – Vol. 3, No. 1. – P. 9–19.
15. PODLIPNIG S., BÖSZÖRMENYI L. *A survey of web cache replacement strategies* // ACM Computing Surveys (CSUR). – 2003. – Vol. 35, No. 4. – P. 374–398.
16. SAFAVI M., BASTANI S., LANDFELDT B. *Online learning and placement algorithms for efficient delivery of user generated contents in telco-CDNs* // IEEE Trans. on Network and Service Management. – 2019. – Vol. 17, No. 1. – P. 637–651.
17. *SureRoute*, – URL: <https://techdocs.akamai.com/property-mgr/docs/sureroute-beh> (дата обращения: 19.04.2025).
18. ZOLFAGHARI B. et al. *Content delivery networks: State of the art, trends, and future roadmap* // ACM Computing Surveys (CSUR). – 2020. – Vol. 53, No. 2. – P. 1–34.

ROUTING MODELS AND METHODS OF INFORMATION RESOURCES ON THE CONTENT DELIVERY NETWORK BASED ON DOMESTIC SOFTWARE

Alexey Prokurovsky, Moscow Technical University of Communications and Informatics, Moscow, Postgraduate (a.a.prokurovskiy@mtuci.ru).

Natalia Toutova, Moscow Technical University of Communications and Informatics, Moscow, Assistant Professor (n.v.tutova@mtuci.ru).

Iliya Andreev, Moscow Technical University of Communications and Informatics, Moscow, Assistant Professor (kis.dep@mtuci.ru).

Abstract: Content Delivery Networks are designed to accelerate the loading of web-sites, applications and other digital content for users located in different regions.

*Программы и системы моделирования объектов,
средств и систем управления*

The main goal of such networks is to reduce the delay in data transmission between the server and the user, ensuring fast content delivery regardless of the user's geographic location. Import substitution of content delivery networks is a relevant and important scientific and technical task. The paper proposes a method for efficient distribution of digital information resources in a content delivery network implemented on the domestic software product 1C:Bus. The method is based on the problem of optimizing the route for placing content on caching servers, which corresponds to the problem of the minimum path in a directed graph. To solve the problem, we used the ant colony method to find an efficient solution in the computationally complex problem in an acceptable time. Compared with random route generation, the proposed route generation methods can speed up content loading by up to 20%. Multithreaded transmission made it possible to speed up content loading by two more times.

Keywords: content delivery network, content distribution, digital resources, ant colony algorithm, minimum path search, 1C:Bus.

УДК 004.41

ББК 32.972

*Статья представлена к публикации
членом редакционной коллегии А.А. Печниковым.*

Поступила в редакцию 24.01.2025.

Опубликована 31.07.2025.