

УДК 004.75:658.7

6. ОТ ТОННО-КИЛОМЕТРОВ К ТЕРАБАЙТАМ: ЛОГИСТИЧЕСКИЕ ПРИНЦИПЫ В АРХИТЕКТУРЕ СОВРЕМЕННЫХ IT-СИСТЕМ

Спарш Д.Е., студент гр. 373901

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Фещенко С.Л. – старший преподаватель

Аннотация. В работе исследуются параллели между принципами логистики и архитектурными решениями современных высоконагруженных IT-систем. На примерах CI/CD-конвейеров, сетевой маршрутизации, контейнеризации (Docker/Kubernetes) и инфраструктуры Netflix демонстрируется, что IT-индустрия воспроизводит логистические модели управления потоками, стандартизации и оптимизации маршрутов.

Ключевые слова. логистика, IT-архитектура, CI/CD, маршрутизация, Docker, Kubernetes, CDN, Netflix, управление потоками, контейнеризация, складская логистика, транспортная задача

Современные IT-системы, обеспечивающие доставку цифрового контента миллиардам пользователей, решают, по существу, ту же задачу, которую логистика формализовала задолго до появления интернета – управление потоками в условиях ограниченных ресурсов, неопределенного спроса и требований к скорости обслуживания. Целью данной работы является выявление и систематизация параллелей между логистическими принципами и архитектурными решениями в современных высоконагруженных информационных системах. Практическая значимость состоит в том, что понимание этих параллелей позволяет специалисту на стыке экономики и IT переносить проверенный аналитический аппарат логистики на проектирование программных систем – и наоборот.

В качестве теоретической основы используются классические концепции транспортной, складской и распределительной логистики, изложенные в работах А. М. Гаджинского [1], а также фундаментальные труды по архитектуре распределенных систем М. Клеппмана [2] и компьютерным сетям Э. Таненбаума [3]. Объектом исследования выступают три ключевые области IT-инфраструктуры: конвейеры непрерывной доставки кода (CI/CD), маршрутизация сетевого трафика и контейнеризация приложений с оркестрацией. Предметом – их соответствие принципам и моделям логистики.

Первая параллель. CI/CD-конвейер и транспортная логистика. Конвейер непрерывной интеграции и доставки (CI/CD) – это последовательность автоматизированных этапов, через которые проходит программный код от коммита разработчика до его развертывания на сервере, доступном конечному пользователю [4]. В терминах транспортной логистики здесь прослеживается прямая аналогия. Код-коммит представляет собой единицу груза (грузовую единицу), сам конвейер – транспортный маршрут с последовательными остановками (сборка, тестирование, staging, production), а деплой – момент доставки груза получателю. Понятие «ездка» в логистике, обозначающее законченный цикл перемещения транспортного средства, соответствует одному прогону pipeline – от триггера до завершения. Грузооборот (тонно-километры) находит аналог в объеме данных, проходящих через CI/CD за единицу времени (например, число деплоев в сутки, умноженное на размер артефакта). Время простоя транспортного средства соответствует времени ожидания в очереди задач CI-сервера, а коэффициент использования пробега – доле «полезных» прогонов (дошедших до production) в общем числе запусков pipeline.

Вторая параллель. Сетевая маршрутизация и транспортные маршруты. Алгоритмы маршрутизации в компьютерных сетях – OSPF (Open Shortest Path First), BGP (Border Gateway Protocol) – решают задачу выбора оптимального пути передачи пакета данных от источника к получателю [3]. Это прямой аналог задачи маршрутизации в транспортной логистике. Маятниковый маршрут ($A \rightarrow B \rightarrow A$) соответствует простейшей модели «клиент-сервер», где запрос уходит на сервер и ответ возвращается тем же путем. Кольцевой маршрут ($A \rightarrow B \rightarrow C \rightarrow D \rightarrow A$) воспроизводится в топологии Token Ring и в маршрутах CDN, где контент последовательно обновляется на цепочке узлов. Развозочный маршрут ($A \rightarrow B, A \rightarrow C, A \rightarrow D$) соответствует multicast- и broadcast-рассылке, при которой один источник направляет данные одновременно нескольким получателям. Выбор транспортного протокола (TCP или UDP) по своей сути аналогичен задаче выбора типа транспортного средства: TCP – это «тягач с прицепом», обеспечивающий гарантированную доставку ценой снижения скорости из-за механизмов подтверждения, тогда как UDP – курьерская доставка, где скорость приоритетна, а допускается некоторый процент потерь. Расчет равномерного расстояния в логистике, при котором два вида транспорта становятся одинаково эффективными, находит прямой аналог в выборе протокола в зависимости от характера трафика: для потокового видео «равноценная точка» смещена в сторону UDP, для финансовых транзакций – в сторону TCP.

Третья параллель. Контейнеризация (Docker/Kubernetes) и складская логистика. Docker-контейнер – это стандартизированная единица упаковки приложения с его зависимостями, обеспечивающая

идентичное поведение в любой среде выполнения. Появление морского контейнера стандарта ISO (TEU – Twenty-foot Equivalent Unit) в 1956 г. произвело революцию в мировой торговле за счет унификации тары и совместимости между видами транспорта [5]. Docker воспроизводит тот же принцип: стандартизированный формат образа, переносимый между средами (ноутбук разработчика → тестовый сервер → production). Kubernetes, выполняющий функции оркестрации контейнеров, по сути является WMS (Warehouse Management System) – системой управления складом, которая решает, куда разместить груз (контейнер), как распределить ресурсы и обеспечить масштабирование. Классификация складов в логистике (классы A, B, C, D по уровню оснащенности) находит точное соответствие в системе тиров дата-центров (Tier I–IV) по классификации Uptime Institute [6]: чем выше тир, тем выше отказоустойчивость и стоимость – аналогично тому, как склад класса «А» обеспечивает наилучшие условия хранения, но при значительно более высоких затратах. Метод центра масс грузопотоков, используемый в логистике для определения оптимального расположения складов, концептуально совпадает с алгоритмами размещения CDN-узлов: сервер следует размещать как можно ближе к центру «тяжести» пользовательского трафика.

Четвертая параллель, дополняющая складскую тему: кэширование данных и управление запасами. Классическая модель управления запасами EOQ (Economic Order Quantity) определяет оптимальный размер партии заказа, балансируя затраты на хранение и затраты на размещение заказа. Политики кэширования в IT-системах решают аналогичную задачу: TTL (Time-To-Live) определяет, как долго «товар» (данные) хранится на промежуточном «складе» (кэш-сервере) до истечения срока годности; алгоритм LRU (Least Recently Used) реализует принцип «залежавшийся товар списывается первым», освобождая место для более востребованных позиций; а механизм prefetch – это упреждающий заказ наиболее популярных товаров до момента обращения клиента, аналог страхового запаса.

Подход JIT (Just-In-Time), минимизирующий складские запасы за счет синхронизации поставки с потреблением, отражается в стриминговых протоколах, где контент подгружается порционно «ровно к моменту» воспроизведения – вместо полной предварительной загрузки файла. Примечательно, что тот же принцип реализован и на более низком уровне – в самих языках программирования. JIT-компиляция в среде CLR (Common Language Runtime) платформы .NET работает по аналогичной схеме: промежуточный код (IL) не транслируется в машинный целиком заранее, а компилируется «точно в момент» первого вызова каждого метода [9]. Это прямое воспроизведение логистического JIT: ресурс (машинный код) поставляется ровно тогда, когда в нем возникает потребность, что экономит память (аналог складских площадей) и сокращает время начальной загрузки приложения (аналог сокращения времени комплектации заказа).

Отдельно стоит отметить параллель между балансировкой нагрузки в IT-системах и задачей распределения грузопотоков в логистике. Балансировщик нагрузки (load balancer), распределяющий входящие запросы между несколькими серверами, решает, по существу, ту же задачу, что и диспетчер распределительного центра, направляющий грузовики по разным маршрутам для минимизации суммарного времени доставки. Алгоритмы Round Robin, Least Connections и Weighted Load Balancing имеют прямые аналоги в методах распределения транспортных потоков: равномерное распределение по маршрутам, выбор наименее загруженного маршрута и приоритизация на основе грузоподъемности транспортного средства соответственно. Механизмы автомасштабирования (auto-scaling), динамически добавляющие вычислительные ресурсы при пиковых нагрузках, воспроизводят практику привлечения дополнительного транспорта в периоды сезонных пиков спроса.

Сеть Open Connect CDN компании Netflix включает более 14 000 серверов (OCA – Open Connect Appliance), развернутых у интернет-провайдеров и на точках обмена трафиком (IXP) в 142 странах [8]. Их размещение подчиняется логике, идентичной задаче оптимального размещения распределительных центров. Контент заранее «развозится» на ближайшие к пользователю узлы в ночные часы (аналог предварительной расстановки товаров на региональных складах), а в пиковые часы пользователь получает видео с ближайшего OCA, а не от центрального хранилища – что значительно сокращает задержки, подобно тому как региональный склад сокращает время доставки последней мили. Алгоритмы адаптивного битрейта Netflix также имеют логистическую природу. Система в реальном времени выбирает «размер упаковки» видеопотока (разрешение и степень сжатия) в зависимости от пропускной способности канала – это аналог подбора типа транспортного средства под объем и характер груза.

Особого внимания заслуживает система управления запасами контента на OCA-серверах. Netflix использует алгоритмы прогнозирования спроса, определяющие, какие фильмы и сериалы будут наиболее востребованы в каждом конкретном регионе. Популярный контент размещается на локальных OCA (аналог товаров группы «А» по ABC-анализу – они обеспечивают основную долю «оборота»), тогда как менее востребованные позиции остаются на региональных серверах или в центральном хранилище AWS S3. Это прямое воспроизведение принципа ABC-классификации запасов: 20 % наименований обеспечивают 80 % обращений, и именно они должны находиться на ближайшем к потребителю «складе». Применение методов chaos engineering (например, Chaos Monkey) для тестирования отказоустойчивости инфраструктуры также можно рассматривать через призму логистики – как имитацию сбоев в цепи поставок для проверки устойчивости логистической системы к форс-мажорам.

Описанные аналогии, а также другие архитектурные решения в Netflix, основанные на логистических принципах, представлены в таблице 1.

Таблица 1 – Логистические аналогии в инфраструктуре Netflix

Логистический элемент	Аналог в инфраструктуре Netflix
Распределительный центр (РЦ)	Open Connect Appliance (OCA) у ISP
Центральный склад	AWS S3 – основное хранилище контента
Предварительная развозка товаров	Ночное заполнение OCA популярным контентом
Маршрутизация «последней мили»	DNS-маршрутизация к ближайшему OCA
Выбор типа ТС по характеру груза	Адаптивный битрейт (подбор разрешения)
WMS (система управления складом)	Kubernetes-оркестрация микросервисов
ABC-классификация запасов	Алгоритмы приоритетного кэширования контента

Таким образом, проведенный анализ демонстрирует, что архитектурные решения, лежащие в основе современных IT-систем, не просто напоминают логистические модели – они воспроизводят их принципы: стандартизацию грузовых единиц (Docker/TEU), оптимизацию маршрутов доставки (BGP/транспортные задачи), иерархическую систему складирования (CDN/Tier-классификация дата-центров), конвейерную организацию потока (CI/CD-pipeline/транспортный конвейер) и рациональное управление запасами (кэширование/EOQ). Логистика по своей сути уже давно является наукой не о грузовиках и складах, а об управлении потоками. Самая быстрорастущая отрасль мировой экономики строит свою инфраструктуру на тех же принципах, зачастую даже не осознавая этого. Осознанное применение логистического аппарата специалистами на стыке экономики и IT-индустрии открывает путь к более обоснованным архитектурным решениям и кросс-дисциплинарным инновациям.

Список использованных источников:

1. Гаджинский, А. М. *Логистика : учебник* / А. М. Гаджинский. – 21-е изд. – М. : Дашков и К°, 2017. – 420 с.
2. Kleppmann, M. *Designing Data-Intensive Applications* / M. Kleppmann. – Sebastopol : O'Reilly Media, 2017. – 616 p.
3. Tanenbaum, A. S. *Computer Networks* / A. S. Tanenbaum, D. J. Wetherall. – 5-th ed. – Boston : Pearson, 2011. – 960 p.
4. What is CI/CD? [Электронный ресурс] / Red Hat. – URL: <https://www.redhat.com/en/topics/devops/what-is-ci-cd> (дата доступа: 14.03.2026).
5. Levinson, M. *The Box: How the Shipping Container Made the World Smaller and the World Economy Bigger* / M. Levinson. – 2nd ed. – Princeton : Princeton University Press, 2016. – 491 p.
6. Uptime Institute Tier Classification System [Электронный ресурс] / Uptime Institute. – URL: <https://uptimeinstitute.com/tiers> (дата доступа: 14.03.2026).
7. Böttger, T. *Open Connect Everywhere: A Glimpse at the Internet Ecosystem through the Lens of the Netflix CDN* / T. Böttger [Электронный ресурс] // ACM SIGCOMM Computer Communications Review. – 2018. – Vol. 48, No. 1. – URL: <https://arxiv.org/abs/1606.05519>
8. *Open Connect Overview* [Электронный ресурс] / Netflix. – URL: <https://openconnect.netflix.com/Open-Connect-Overview.pdf> (дата доступа: 14.03.2026).
9. Richter, J. *CLR via C#* / J. Richter. – 4th ed. – Redmond : Microsoft Press, 2012. – 896 p.

UDC 004.75:658.7

FROM TON-KILOMETERS TO TERABYTES: LOGISTICS PRINCIPLES IN THE ARCHITECTURE OF MODERN IT SYSTEMS

Sparish D.E¹, student of group 373901

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus¹

Feshchenko S.L. – Master of Economics

Annotation. The paper examines parallels between logistics principles and architectural solutions in modern high-load IT systems. Using examples of CI/CD pipelines, network routing, containerization (Docker/Kubernetes), and Netflix infrastructure, it demonstrates that the IT industry reproduces fundamental logistics models of flow management, standardization, and route optimization.

Keywords: logistics, IT architecture, CI/CD, routing, Docker, Kubernetes, CDN, Netflix, flow management, containerization, warehouse logistics, transportation problem.