DOI 10.37882/2223-2966.2026.01.24

ОПТИМИЗАЦИЯ ЗАТРАТ НА ВЫЧИСЛИТЕЛЬНЫЕ РЕСУРСЫ

OPTIMIZING THE COST OF COMPUTING RESOURCES

N. Muntian

Summary. In the era of digital transformation, the exponential growth of cloud infrastructure costs has become one of the key challenges for modern technology platforms. Naive approaches to allocating computing resources based on peak loads are fundamentally flawed and lead to enormous financial losses. Effective cost management requires a deep synthesis of technical optimization strategies and a granular understanding of the key business processes that this infrastructure serves. This article approves and demonstrates the framework for such synthesis, illustrating the application of cost-saving methodologies in AWS using the example of a real case of a highly loaded service.

Keywords: cost of computing resources, efficient management, financial losses.

Мунтян Никита Валерьевич
Российский государственный социальный университет, Россия, г. Москва
nikita.muntian@icloud.com

Аннотация. В эпоху цифровой трансформации экспоненциальный рост затрат на облачную инфраструктуру стал одной из ключевых проблем для современных технологических платформ. Наивные подходы к выделению вычислительных ресурсов, основанные на пиковых нагрузках, являются фундаментально ошибочными и ведут к колоссальным финансовым потерям. Эффективное управление затратами требует глубокого синтеза технических стратегий оптимизации и гранулярного понимания ключевых бизнес-процессов, которые эта инфраструктура обслуживает. Данная статья утверждает и демонстрирует фреймворк для такого синтеза, иллюстрируя применение методологий экономии средств в AWS на примере реального кейса высоконагруженного сервиса.

Ключевые слова: затраты на вычислительные ресурсы, эффективное управление, финансовые потери.

1. Теоретический базис и ключевые концепции облачной инфраструктуры AWS

Прежде чем приступать к анализу тактик оптимизации, критически важно сформировать фундаментальное понимание архитектурных принципов, лежащих в основе гипермасштабируемых облачных платформ, таких как Amazon Web Services (AWS). Неспособность усвоить эти базовые концепции неизбежно приводит к созданию субоптимальных и экономически неэффективных инфраструктурных решений.

Фундаментальные принципы облачных вычислений — эластичность, масштабируемость и модель оплаты по мере использования (pay-as-you-go) — представляют собой парадигмальный сдвиг по сравнению с устаревшей моделью локальной инфраструктуры (on-premise). Эта новая парадигма не только обеспечивает превосходство в архитектурной гибкости, но и вносит существенные сложности в финансовое планирование, требуя от инженеров и архитекторов компетенций в области финансового менеджмента.

В текущих реалиях AWS является отраслевым стандартом для развертывания масштабируемых приложений. Глубокое владение сервисами этой платформы перестало быть факультативным преимуществом и стало базовым требованием для любого компетентного архитектора решений. Освоение фундаментальных концепций закономерно подводит нас к необходимости изучения более продвинутых методов финансового и операционного контроля.

2. Стратегии и методы управления затратами на вычислительные ресурсы AWS

Несмотря на то, что AWS предоставляет чрезвычайно мощный инструментарий для построения инфраструктуры, по умолчанию эти инструменты агностичны к вопросам экономической эффективности. Проактивное управление затратами, известное как дисциплина FinOps, должно быть заложено в архитектуру системы с самого начала, а не применяться ретроспективно в качестве запоздалой меры.

2.1. Динамическое масштабирование с AWS Auto Scaling

AWS Auto Scaling представляет собой механизм автоматической корректировки вычислительных мощностей в ответ на изменения рабочей нагрузки. Данный сервис является первой и основной линией защиты как от избыточного выделения ресурсов (что ведет к прямым финансовым потерям), так и от их недостатка (что приводит к деградации производительности и ухудшению пользовательского опыта). Правильно настроенные политики Auto Scaling позволяют системе эластично реагировать на колебания трафика, поддерживая оптимальный баланс между стоимостью и производительностью согласно официальной документации AWS по Auto Scaling

[https://docs.aws.amazon.com/autoscaling/ec2/userguide/what-is-amazon-ec2-auto-scaling.html, с. 1].

2.2. Оптимизация затрат через модели закупки: Reserved и Spot Instances

AWS предлагает различные модели ценообразования для своих вычислительных ресурсов (EC2), и их стратегическое использование является краеугольным камнем оптимизации затрат.

- Reserved Instances (RI) предлагают значительную скидку в обмен на обязательство использовать определенный объем вычислительных мощностей в течение одного или трех лет. Этот инструмент идеально подходит для предсказуемых, постоянных рабочих нагрузок, составляющих базовый уровень потребления системы. Экономическая модель здесь проста: долгосрочное обязательство в обмен на снижение почасовой ставки.
- Spot Instances позволяют использовать свободные вычислительные мощности EC2 со скидкой до 90 % от цены по запросу. Однако основной компромисс заключается в том, что AWS может прервать работу такого инстанса в любой момент с уведомлением за две минуты. Следовательно, Spot Instances подходят исключительно для отказоустойчивых, прерываемых рабочих нагрузок, таких как пакетная обработка данных, задачи рендеринга или фоновые аналитические вычисления. Их использование требует более сложной архитектуры приложения, способной выдерживать внезапные прерывания согласно официальной документации по моделям ценообразования AWS. [https://aws.amazon.com/ec2/pricing/, с. 2].

Эффективная реализация этих стратегий немыслима без внедрения надежных систем мониторинга и аналитики, позволяющих принимать решения на основе данных.

3. Инструментарий для мониторинга и анализа затрат

Принцип «нельзя оптимизировать то, что нельзя измерить» является фундаментальной аксиомой дисциплины FinOps. Для получения полного представления о потреблении ресурсов и распределении затрат необходимо освоить нативные инструменты AWS, предназначенные для этой цели.

AWS CloudWatch — это основной сервис мониторинга и наблюдаемости в экосистеме AWS. Он собирает метрики производительности, логи и события со всех ресурсов. Эти данные служат фундаментальной основой для настройки интеллектуальных политик Auto Scaling, создания оповещений о производительности и выявления аномалий в работе приложений. Без данных, предоставляемых CloudWatch, любое масштабирование было бы слепым и неэффективным.

AWS Cost Explorer выступает в роли главного инструмента финансового анализа. Он предоставляет интуитивно понятный интерфейс для визуализации исторических и текущих затрат, прогнозирования будущих расходов и детализации потребления по сервисам, тегам или учетным записям. Cost Explorer позволяет выявлять тренды и аномальные всплески потребления, что является отправной точкой для любой инициативы по оптимизации.

После того как теоретические основы и инструментарий были заложены, фокус должен сместиться на их практическое применение, которое неразрывно связано с бизнес-логикой приложения.

4. Практическое применение методологий оптимизации на основе анализа бизнес-процессов

Наиболее значительные возможности для оптимизации затрат часто скрыты от чисто технического взгляда и могут быть обнаружены только через призму анализа конкретных бизнес-операций, которые обслуживает инфраструктура. Следующий анализ демонстрирует этот принцип на практике.

- Анализ процесса «Send Order» (Отправка заказа): Процесс «Send Order» демонстрирует классический «пилообразный» паттерн нагрузки, напрямую коррелирующий с внешними факторами, такими как час пик или окончание массовых мероприятий. Статическое провижинирование под пиковую нагрузку в таких условиях является формой финансовой халатности, гарантирующей оплату 80% времени простаивающих ресурсов. Архитектурно верным решением является внедрение AWS Auto Scaling с политиками, основанными на метриках CPU Utilization и, что более важно, на длине очереди входящих запросов (SQS queue depth), что позволяет проактивно реагировать на всплески спроса.
- Анализ процесса «Driver Confirmation» (Подтверждение водителем): После подтверждения заказа система начинает в реальном времени отслеживать перемещение водителя. Критически важной, видимой пользователю частью является движение иконки автомобиля на карте, требующее надежной, постоянно доступной инфраструктуры. Однако фоновая агрегация этих же GPS-данных для последующего долгосрочного анализа (например, для выявления паттернов трафика или оценки эффективности водителей) является классической прерываемой задачей. Архитектурно

зрелый подход требует разделения этих нагрузок: критический real-time стриминг выполняется на On-Demand инстансах, в то время как аналитическая обработка данных выносится на Spot Instances, что дает колоссальную экономию без риска для пользовательского опыта.

- Анализ процессов «Trip Completion» и «Payment Processing» (Завершение и оплата поездки): Процессы, связанные с завершением поездки — расчет итоговой стоимости, обработка транзакций через платежные шлюзы, генерация квитанций и обновление финансовых отчетов — характеризуются высокой степенью предсказуемости и стабильности. Объем этих операций прямо пропорционален количеству завершенных поездок и формирует постоянную, базовую нагрузку на финансовые и отчетные микросервисы. Как показывает практика, попытки динамического масштабирования таких систем часто приводят к излишней сложности. Поэтому серверы, обслуживающие эти функции, являются идеальными кандидатами для перевода на Reserved Instances, что позволит максимизировать экономию за счет долгосрочных обязательств на предсказуемой части инфраструктуры.

Чтобы доказать жизнеспособность предложенных стратегий, необходимо перейти от высокоуровневых гипотез к микроскопическому анализу операционного потока данных и триггеров в каждом бизнес-процессе inDrive, что и будет сделано в следующем разделе.

5. Декомпозиция бизнес-процессов организации (Case Study: inDrive)

Этот раздел представляет собой углубленный анализ ключевых бизнес-процессов сервиса заказа такси inDrive. Данный анализ служит практической основой для применения стратегий оптимизации, обсуждавшихся ранее.

5.1. Процесс инициации и отправки заказа (Send Order)

Клиент инициирует процесс, указывая в мобильном приложении пункты отправления и назначения, а также дополнительные параметры поездки. После подтверждения запрос поступает на серверы inDrive, где система начинает его обработку: проверяет наличие доступных водителей в районе, оценивает их загруженность и предоставляет клиенту возможность отменить заказ до момента его подтверждения водителем. В случае наличия подходящих водителей система рассылает им уведомления о новом заказе. С точки зрения инфраструктуры, этот процесс генерирует короткоживущие, высокочастотные транзакции, требующие низкой задержки от API-шлюзов и высокой пропускной способности от системы подбора водителей.

Подтверждение сделки исполнителем

Приходит пуш-уведомление, исполнитель решает о выполнении заказа, принимая его или нет и времени подачи. Он подтверждает заказ, то он готов принять ордер через веб-интерфейс мобильного приложения. Бекенд— серверная часть обрабатывает ордер айди и присылает уведомление с информацией о исполнителе, автомобиле и времени прибытия.

Процесс начала поездки

Водитель прибывает в точку отправления, используя навигацию в приложении. Система в реальном времени отслеживает его перемещение и уведомляет клиента о прибытии. После проверки данных клиента и подтверждения маршрута водитель отмечает начало поездки в приложении. Система фиксирует время старта, обновляя статус заказа в базе данных. Этот процесс характеризуется непрерывным потоком GPS-координат от приложения водителя к серверу, что создает постоянную нагрузку на сервисы геолокации и обработки потоковых данных.

Процесс завершения поездки

По прибытии в пункт назначения водитель завершает поездку в приложении. Система автоматически фиксирует время окончания и рассчитывает итоговую стоимость на основе пройденного расстояния и времени в пути. Клиент получает уведомление с детализацией поездки и итоговой суммой, после чего ему предлагается оценить поездку и оставить отзыв. С технической точки зрения, этот этап инициирует серию транзакционных, высоконадежных операций с интенсивным взаимодействием с базой данных для фиксации итогов поездки.

Процесс обработки платежа

После завершения поездки система инициирует процесс оплаты. В случае безналичного расчета средства автоматически списываются с привязанного платежного метода клиента. Система отправляет подтверждение об успешной оплате и электронный чек клиенту, а также уведомление о зачислении средств водителю. Статус заказа в базе данных обновляется на «завершен и оплачен». Этот процесс требует высоконадежной и безопасной вычислительной среды для взаимодействия с внешними платежными шлюзами и обеспечения целостности финансовых данных.

Предложения по реинжинирингу бизнес-процессов

В рамках стратегического развития сервиса предлагается введение двух новых ролей для углубления ана-

лиза и усиления контроля над операционной деятельностью.

Аналитика поездок: Целью этой роли является надзор за процессом выполнения поездок, оптимизация маршрутов и контроль соблюдения стандартов качества и безопасности. Функционал включает анализ данных о поездках для выявления проблемных зон и взаимодействие с водителями для улучшения качества обслуживания.

Аналитик клиентских данных: Эта роль нацелена на улучшение взаимодействия с клиентами и повышение их лояльности. Функционал подразумевает анализ клиентских данных для выявления предпочтений, разработку персонализированных предложений совместно с отделом маркетинга и мониторинг отзывов для постоянного улучшения сервиса.

Внедрение этих ролей — это не просто операционное улучшение, а стратегическая инвестиция в зрелость FinOps. «Аналитик поездок» может выявлять неэффективные маршруты, сжигающие избыточные вычислительные циклы на GPS-трекинг, а «Аналитик клиентских данных» — сегментировать пользователей, позволяя в будущем принимать решения о дифференцированном уровне сервиса (и, следовательно, разной стоимости инфраструктуры) в зависимости от ценности клиента.

Заключение

Достижение экономической эффективности в облаке — это многомерная задача, выходящая далеко за рамки простых технических корректировок. Основной тезис данной статьи заключается в том, что единственный устойчивый путь к созданию масштабируемых, производительных и финансово жизнеспособных систем лежит через синергетический подход. Этот подход должен органично сочетать глубокое знание технических возможностей сервисов AWS с доскональным анализом бизнес-логики, которую они призваны поддерживать. В конечном счете FinOps перестает быть функцией IT-отдела и трансформируется в ключевую компетенцию правления, где каждый процент сэкономленных облачных затрат напрямую конвертируется в маржинальность бизнеса и акционерную стоимость.

ЛИТЕРАТУРА

1. Официальная документация Amazon Web Services. AWS Auto Scaling User Guide. [Электронный ресурс]. Режим доступа: https://docs.aws.amazon.com/autoscaling/ec2/userguide/what-is-amazon-ec2-auto-scaling.html (дата обращения: 15.10.2025).
2. Официальная документация Amazon Web Services. Amazon EC2 Pricing Models. [Электронный ресурс]. Режим доступа: https://aws.amazon.com/ec2/pricing/ (дата обращения: 15.10.2025).
3. Фаулер М. Архитектура корпоративных программных приложений. — М.: Вильямс, 2016. — 544 с.
4. Робертс С., Арора Б. We Love FinOps. — O'Reilly Media, 2020. — 150 с.

Журнал «Современная наука: актуальные проблемы теории и практики»