
Оглавление

От издательства	12
О научном редакторе русского издания	12
Предисловие	13
Для кого эта книга?	13
Что вы узнаете.....	14
Структура книги	15
Условные обозначения	17
Благодарности	17

Часть I

ПРОИЗВОДИТЕЛЬНОСТЬ — ЭТО ПОЛЬЗОВАТЕЛЬСКИЙ ОПЫТ

Глава 1. Почему производительность — это пользовательский опыт.....	20
Цена задержек: первое впечатление и удовлетворенность пользователя	22
Психология ожидания	23
Оценка UX-производительности: ключевые метрики.....	24
Производительность для всех: устройства и сетевые условия	25
Дизайн с акцентом на производительность	27
Модель RAIL: производительность с ориентацией на пользователя	27
Производительность как отправная точка проектирования.....	28
Глава 2. Измерение важного: основные метрики производительности с ориентацией на пользователя.....	30
Лабораторные тесты и полевые исследования.....	30
CWV: ключевые метрики	32
LCP: быстрое отображение контента	32
FID и INP: готовность к взаимодействию	33
CLS: визуальная стабильность	34
Другие важные метрики	35
Интерпретация и реагирование на отчеты	35
Лабораторные инструменты	36
Lighthouse: аудит производительности.....	36
Панель Performance Chrome DevTools: глубокая диагностика с поддержкой ИИ.....	39
WebPageTest: реалистичное тестирование и углубленный анализ	42

Полевые инструменты	46
Полевые измерения метрик CWV при помощи web-vitals.js	47
Применение CrUX и данных из внешних источников	50
RUM — это не только CWV	50
Данные RUM как путь к улучшению	51
Бюджет производительности и оповещения	52
Практический пример: постоянный мониторинг в действии.....	53
Выводы	54
Глава 3. ИИ-код и парадокс производительности.....	55
ИИ-код: правильный, но не оптимальный	55
Скрытые издержки на ИИ-код	57
Ответственность за качество и производительность.....	60
Оптимизация с помощью ИИ: будущее?	61
Влияние на общую картину производительности.....	62
Роль человека в эпоху ИИ	63
Заключительные мысли об ИИ и производительности	63
 Часть II ОПТИМИЗАЦИЯ ВЕБ-ПРОИЗВОДИТЕЛЬНОСТИ В ЭПОХУ ИИ 	
Глава 4. Оптимизация фронтенда, созданного ИИ	66
CWV и код, сгенерированный ИИ	66
Частые проблемы с производительностью React-компонентов, созданных ИИ.....	67
Нестабильность макета (проблемы с CLS).....	68
Раздутые бандлы и медленная загрузка (проблемы с LCP)	69
Узкие места основного потока и подвисания (проблемы с INP)	71
Игнорирование доступности и обратной связи в UX	72
Улучшение компонентов, созданных ИИ.....	73
Профилирование производительности и новые инструменты на базе ИИ.....	80
Тестирование на реальных устройствах: окончательная проверка.....	82
Повышение скорости при сохранении качества.....	83
Глава 5. Браузер изнутри: загрузка и рендеринг страниц	85
Работа браузера: основы.....	85
Многопроцессная архитектура и основной поток.....	86
Конвейер рендеринга.....	87
RenderingNG.....	91
Сканер предварительной загрузки и приоритизация ресурсов.....	92

Скрипты и цикл событий	93
Модель RAIL	94
Узкие места, связанные с рендерингом.....	95
Работа браузера: сложные вопросы	96
Планировщик задач основного потока Chrome и приоритеты задач	96
Рендеринг на стороне сервера и гидратация.....	98
Сетевой стек и загрузка ресурсов.....	101
Все по полочкам: хронология оптимизированной загрузки.....	105
Выводы	109
Глава 6. Компромиссы при оптимизации производительности	110
Баланс показателей производительности	110
Объективные метрики и субъективное восприятие	111
Распространенные компромиссы	111
Размеры бандла vs сетевые накладные расходы	112
Фреймворки vs ванильный JavaScript.....	113
Микрооптимизации vs сопровождаемость	114
Культура и бюджет производительности	115
Знать, когда остановиться	117
Выводы	118

Часть III

ОПТИМИЗАЦИЯ JAVASCRIPT

Глава 7. Плата за JavaScript	120
Ограничения.....	120
Производительность устройства: процессор, память и вычислительная мощность	120
Сетевые ограничения: полоса пропускания и (особенно) задержка	122
Архитектура браузера: основной поток, парсинг и выполнение.....	123
Движок JavaScript	125
Развитие движка: от интерпретации к JIT-компиляции.....	125
Фоновая компиляция, парсинг и сборка мусора.....	127
Пределы волшебства движков	128
Производительность JavaScript на бэкенде	129
Node.js и цикл событий.....	130
Оптимизации, характерные для бэкенда.....	131
Мониторинг и наблюдаемость	133
Выводы	133

Часть IV

УПРАВЛЕНИЕ ЗАВИСИМОСТЯМИ И ЗАБОТА О ПРОИЗВОДИТЕЛЬНОСТИ

Глава 8. Сторонние скрипты: основы	138
Сторонний код повсюду	139
Влияние сторонних скриптов на производительность	140
Анализ влияния сторонних скриптов на производительность	144
Инвентаризация скриптов, ключевые метрики и индикаторы	144
Особенности сторонних скриптов на основе ИИ	145
Lighthouse и Chrome DevTools	145
WebPageTest (WPT)	146
Принципы оптимизации сторонних скриптов	149
Удаление, упрощение, замена	149
Бюджет производительности	150
Сдвиг влево: вовлечение всех участников	150
Цель — скорость и функциональность	151
Глава 9. Загрузка сторонних скриптов	152
Порядок и приоритеты	152
Выявление критических скриптов	152
Расположение тегов <script>	153
Неблокирующие скрипты: async и defer	154
Ранняя (но не слишком) загрузка критических сторонних скриптов	155
Сохранение порядка выполнения	155
Раннее подключение при помощи ресурсных подсказок	156
Обзор лучших практик для управления загрузкой сторонних скриптов	157
Ленивая загрузка стороннего контента	158
Ленивая загрузка iframe средствами браузера	158
Настраиваемая ленивая загрузка при помощи IntersectionObserver	159
Преимущества и недостатки	161
Пример: ленивая загрузка скриптов аналитики	162
Фасады и загрузка по клику	162
Реализация загрузки по клику	163
Преимущества и недостатки	166
Условная загрузка	167
Оценка устройства с точки зрения производительности	167
Загрузка с учетом сети	168
Преимущества и недостатки	169
Выводы	170

Глава 10. Планирование и оптимизация сторонних скриптов (и роль ИИ)	171
Планирование сторонних скриптов	171
API requestIdleCallback	171
После onload или первого взаимодействия.....	172
Idle Until Urgent	173
Разбиение длительных задач	173
setTimeout как запасной или упрощенный вариант.....	174
Вынос скриптов из основного потока.....	174
Веб-воркеры и Partytown	175
Изоляция в iframe	176
Сторонняя логика на стороне сервера.....	176
Оптимизация средствами фреймворков: Next.js и другие	178
Компонент <Script> из Next.js	178
Оптимизация в других фреймворках	181
Оптимизация с учетом типа сайта	181
Сторонние скрипты в одностраничных (SPA) и многостраничных (MPA) приложениях.....	182
Оптимизация производительности диспетчера тегов.....	184
WordPress и другие системы управления контентом (CMS)	185
Сайты электронной коммерции.....	187
Роль ИИ в оптимизации сторонних скриптов	189

Часть V

БУДУЩЕЕ ВЕБ-ПРОИЗВОДИТЕЛЬНОСТИ

Глава 11. Формирование культуры производительности	194
Производительность как функция	194
Просвещение и оснащение команды.....	195
Ответственный за производительность	196
Интеграция в пайплайн CI/CD и рабочий процесс.....	197
ИИ и культура производительности	198
Выводы	199
Глава 12. Веб-производительность. Кейсы и истории успеха.....	201
Электронная коммерция и розничная торговля: производительность побеждает	204
Rakuten 24: A/B-тестирование говорит об удачном вложении в CWV.....	205
Vodafone: ускорение лендинга повышает продажи	206
Магазины на Shopify: Sunday Citizen	207
Магазины на Shopify: Carpe	207
Ray-Ban: предварительный рендеринг	208

Новости и СМИ: рост вовлеченности вместе со скоростью	209
The Telegraph Media Group	209
The Economic Times: улучшение CWV в масштабе	210
Yahoo! Japan News: улучшение CLS коррелирует с вовлеченностью	212
Путешествия и гостиничный бизнес: ускорение бронирования	215
redBus: улучшение INP и повышение продаж	215
Bookaway: статический контент для международной аудитории	216
Финансы и финтех: завоевание доверия с помощью скорости	218
Financer.com: ускорение WordPress для улучшения конверсии и SEO	218
Искусственный интеллект и производительность: новые истории успеха	220
Vercel: оптимизация изображений с помощью ИИ	220
Cloudflare: машинное обучение для прогнозирования производительности	221
Google Chrome: прогнозирование навигации при помощи ИИ	222
ИИ-помощники для кодинга: внимание к производительности	223
Shopify: ИИ для оптимизации магазинов	224
Основные выводы	225
Выводы	227
Глава 13. Будущее веб-производительности в эпоху ИИ	228
ИИ-революция в веб-разработке	228
Практические стратегии в эпоху ИИ	229
UX и потоковые интерфейсы ИИ-агентов: новая парадигма производительности	230
Потоковая передача текста	232
Метрики	232
Бюджеты производительности	233
Тактики рендеринга	234
Выбор транспортного слоя	235
Паттерны, сохраняющие ощущение скорости	236
Потоковая передача UI-компонентов	236
Метрики потоковой передачи компонентов	237
Тактики рендеринга потоковых компонентов	238
Проблемы, требующие внимания	239
Потоковая передача контента и мультимедиа	239
Разрастающиеся списки и ленты	239
Генерация аудио и видео, синтез речи (TTS)	240
Потоковая передача изображений	240
Планирование и отзывчивость под нагрузкой	241
Особенности сетевой и серверной обработки	242
Издержки холодного запуска	242

Обратное давление	242
Интеллектуальное кэширование	243
Разумный подход к фрагментам	243
CWV в мире потоковой передачи	244
Наблюдаемость и отладка потоковых систем	245
Наблюдение при помощи меток	245
Трассировка потоковых данных	246
Оповещение при частых прерываниях потока	246
Доступность	246
Интернационализация	247
Конфиденциальность	247
Безопасность	248
Энергопотребление и финансовые затраты	248
Рекомендуемые паттерны проектирования	249
Нежелательные антипаттерны	250
Практический чек-лист для продакшена	251
Выводы	252
Об авторе	254
Иллюстрация на обложке	255