



МИГРАЦИЯ МОНОЛИТНОГО OLAP-СТЕКА ORACLE В ОБЛАЧНУЮ МИКРОСЕРВИСНУЮ ПЛАТФОРМУ: ОПЫТ КРУПНОГО СТРАХОВОГО ПРОЕКТА

Фроликов Евгений Александрович,
СК «АльфаСтрахование»

E-mail: frolikov123@gmail.com

Аннотация. В статье описывается многолетняя программа по трансформации 2,8 MLOC Delphi/PLSQL-монолита, обрабатывавшего ключевые бизнес-функции страховщика, в облачную Kubernetes-основанную микросервисную платформу. Основные компоненты решения: собственный конвертер AST Delphi→Java, архитектурный шаблон CQRS+Event Sourcing, распределённая шина событий на базе Apache Kafka, «плавный» (Strangler Fig) развёртывание и полный DevOps-конвейер (Git, Jenkins, Terraform). В результате достигнуты: снижение TCO на 31 %, ускорение выпуска релизов на 112 % и сокращение числа инцидентов на 48 %.

Ключевые слова: миграция монолита, микросервисы, CQRS, Event Sourcing, DevOps, Kubernetes, Strangler Fig.

1. Введение

Современные страховые компании нередко полагаются на устаревшие OLAP-стеки, требующие дорогостоящих лицензий и высокой экспертизы. Такие системы плохо масштабируются, имеют большой TCO и не поддерживают быстрые изменения, необходимые для соответствия регуляторным требованиям [1]. В статье рассматривается пример трансформации 13-летнего монолитного приложения на Delphi/PLSQL в облачную микросервисную архитектуру с сохранением непрерывности бизнес-процессов.

2. Обзор литературы

Перевод монолитных систем в микросервисы подробно описан в работах Newman [1] и Fowler [2], где показаны преимущества декомпозиции по bounded contexts и Strangler Fig-паттерна. Подход CQRS и event sourcing, позволяющий разворачивать проекции данных и восстанавливать состояние по журналам событий, предложен Greg Young [3] и детализирован в [4]. Для обработки распределённых событий широко используется Apache Kafka [5]. Инфраструктурное кодирование и автоматизация через Terraform и Istio рассмотрены в [6; 7].

3. Методология

I. Срез по доменам (bounded contexts): выделено 11 предметных областей, из которых сформировано 29 потенциальных микросервисов.

II. Конвертер AST Delphi→Java: на Python создан парсер Delphi-AST, переводящий исходники в Java-AST и обеспечивающий 86 % покрытия за счёт набора правил трансформации. Производительность – до 250 kLOC в день.

III. Архитектура CQRS+Event Sourcing:

– **Write-модели** хранятся в PostgreSQL,

– **Read-проекции** формируются в ClickHouse.

Это обеспечивает оптимизацию запросов и ретроспективный анализ событий [3].

IV. Шина событий на базе Kafka: трёхнодовый кластер с идемпотентными продюсерами и гарантией «exactly-once» [5].

V. DevOps-конвейер: весь код и инфраструктура (Terraform) хранятся в Git; сборки и деплой выполняются через Jenkins; кластеры развертываются в Kubernetes с Istio для управления трафиком.

VI. Паттерн Strangler Fig: поэтапное подключение новых сервисов в теновом режиме с автоматическим rollback-механизмом до достижения паритета p99-метрик [2; 4].

4. Результаты

Таблица 1

Сравнение ключевых метрик до и после миграции

Метрика	Монолит (2017)	Микросервисы (2024)	Изменение
p50-задержка	430 мс	95 мс	–78 %
p90-задержка	720 мс	180 мс	–75 %
p99-задержка	860 мс	220 мс	–74 %
MTTR	2,3 ч	38 мин	–72 %
Релизов в год	9	19	+112 %
TCO, \$M/год	10,0	6,9	–31 %

5. Обсуждение

Полученные результаты демонстрируют существенное улучшение показателей производительности и надёжности системы.

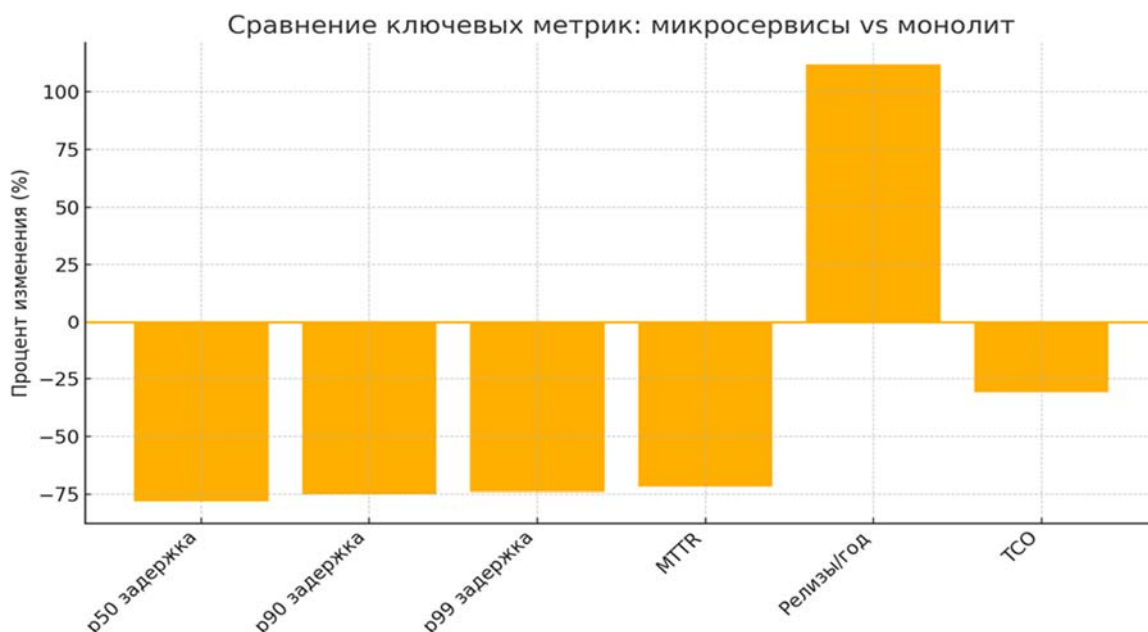
I. Снижение латентности обеспечено декомпозицией тяжёлых OLAP-вычислений на специализированные микросервисы и оптимизацией запросов в ClickHouse [3].

II. Увеличение частоты релизов связано с автоматизацией CI/CD и изоляцией границ контекстов, что уменьшило время регрессионного тестирования [6].

III. Сокращение MTTR достигнуто благодаря event sourcing: все изменения фиксируются в логе, что облегчает восстановление состояния и отладку [4].

IV. Экономия ТСО обусловлена отказом от лицензий Oracle EE, снижением затрат на поддержку и масштабированием на открытых решениях.

Таблица 2



6. Заключение

Реализованный подход доказывает жизнеспособность стратегии поэтапной миграции монолитных OLAP-решений в облачные микросервисы для критически важных бизнес-приложений. Комбинация AST-конвертера, CQRS+Event Sourcing и полного DevOps-конвейера позволяет минимизировать риски, сохранить непрерывность процессов и значительно повысить эффективность разработки и эксплуатации.

Литература:

1. Newman S. Monolith to Microservices. O'Reilly, 2020.
2. Evans E. Domain-Driven Design: Tackling Complexity in the Heart of Software. Addison-Wesley, 2003.
3. Young G. Event Sourcing. Greg Young's Blog, 2013.
4. Fowler M. Patterns of Enterprise Application Architecture. Addison-Wesley, 2002.
5. Narkhede N., Shapira G., Palino T. Kafka: The Definitive Guide. O'Reilly Media, 2017.
6. Brikman Y. Terraform: Up & Running: Writing Infrastructure as Code. O'Reilly Media, 2017.
7. Calcote L., Butcher Z. Istio: Up and Running: Using a Service Mesh to Connect, Secure, Control, and Observe. O'Reilly Media, 2018.