

Tussle – новый формат бенчмаркинга рельсных приложений



Aleksey Ignatenko
Ruslan Scherbakov

Azul

Joker<?>

Зачем нам бенчмаркинг реальных приложений?

- Есть стандартные бенчмарки, например SPEC
- Есть jmh
- ...

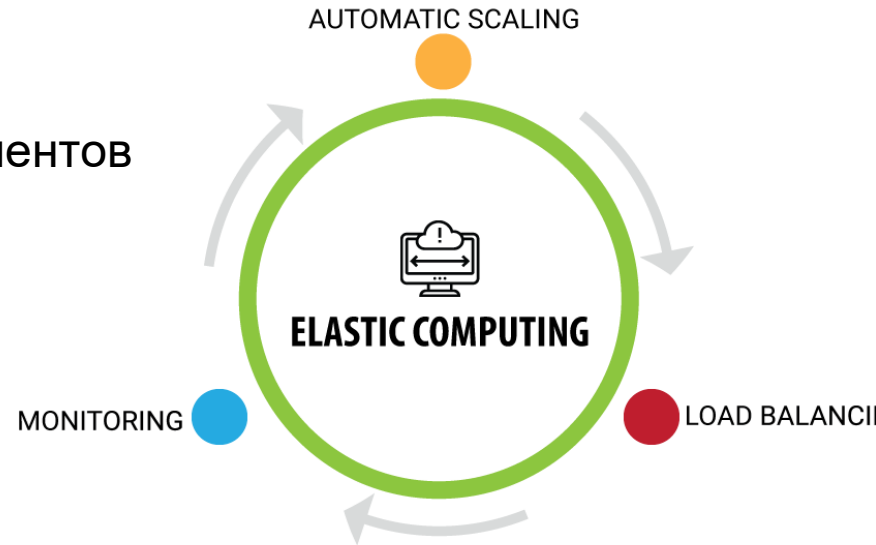
Зачем нам бенчмаркинг реальных приложений?

- Посмотрим на простом примере
- Написали приложение (пусть будет интернет магазин)
- Задеплоили в клауде
- Сотня клиентов, все летает
- Косты на Cloud разумный

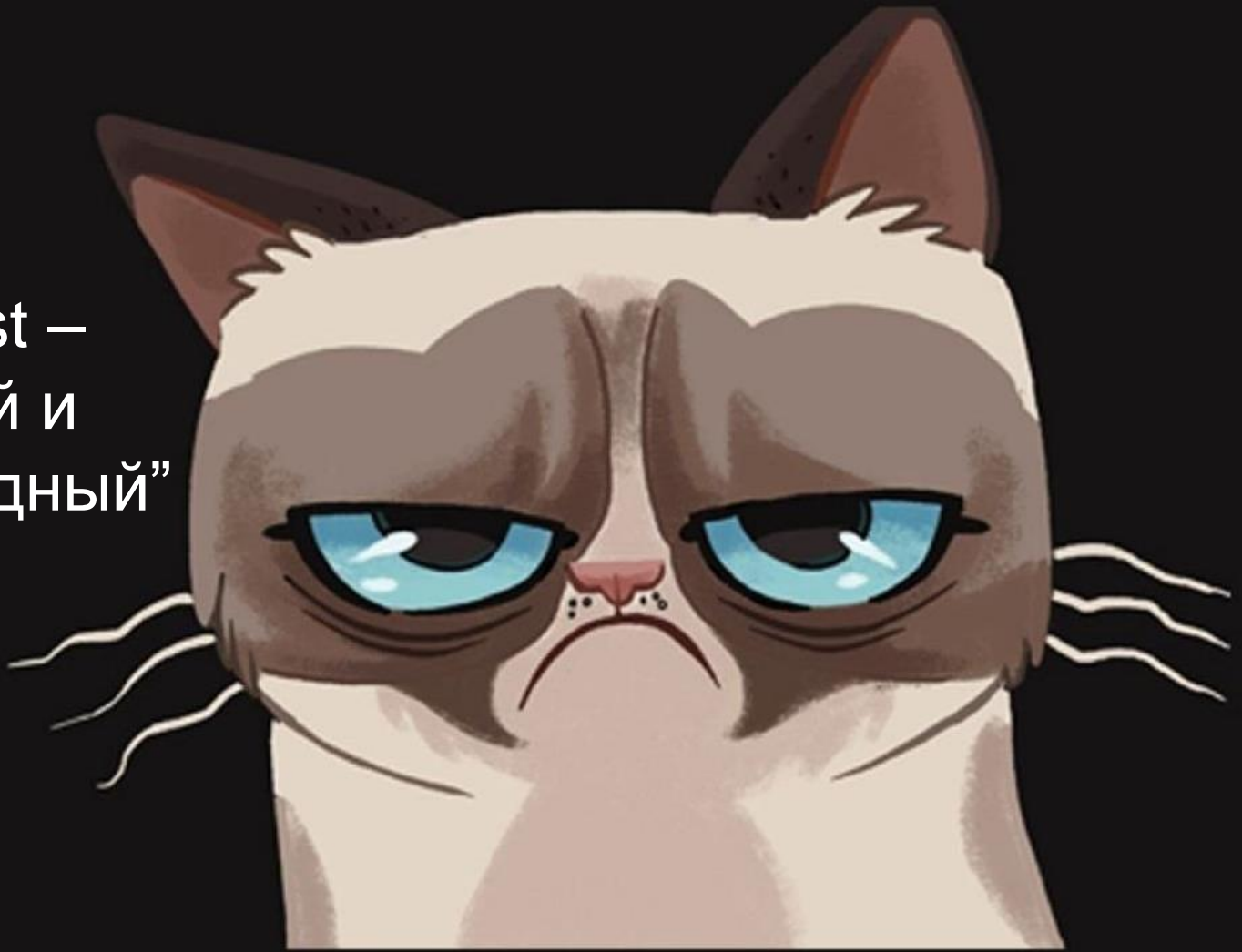
Зачем нам бенчмаркинг реальных приложений?

- Стали популярными
- В день получили от 10 000 до 100 000 клиентов
- Kubernetes для автоскейлинга
- Жизнь удалась?

HOW ELASTIC COMPUTING WORKS



Cloud cost –
“жестокый и
беспощадный”



Зачем нам бенчмаркинг реальных приложений?

- Load generator
- Понять сколько нам стоит один запрос
- Профилировать
- Оптимизировать



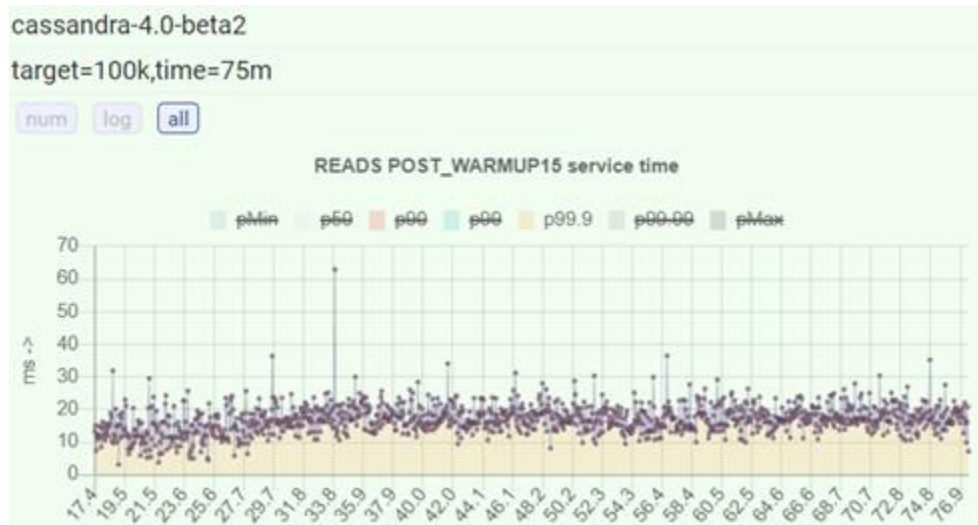
БЕНЧМАРКИНГ

Меню на Сегодня

- Посмотрим на одну очень известную Enterprise бенчмарку
- Научимся правильно измерять производительность
- Поймем плюсы и минусы готового решения
- Ответим Чемберлену или создадим свою бенчмарку
- Покажем Демо

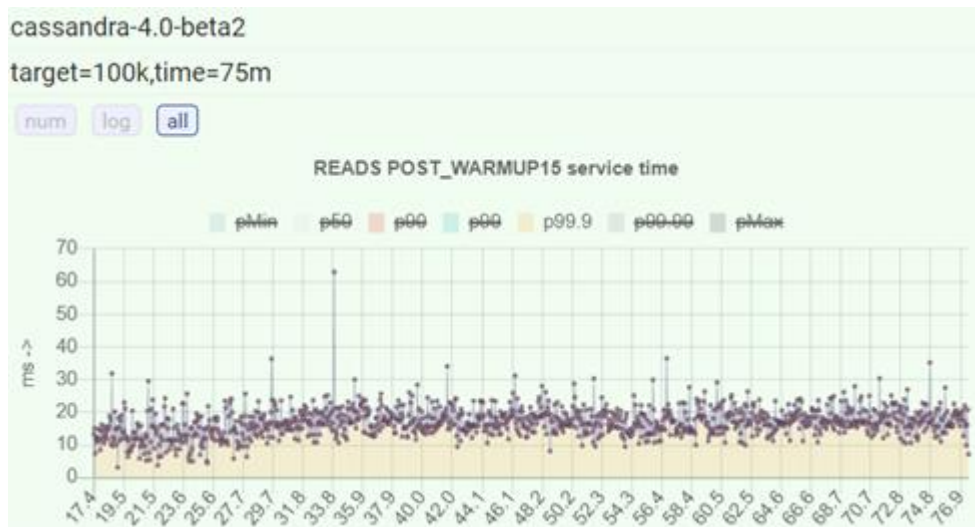
Сначала термины

- Benchmark
- Throughput
- Latency
- Персентили
- SLA

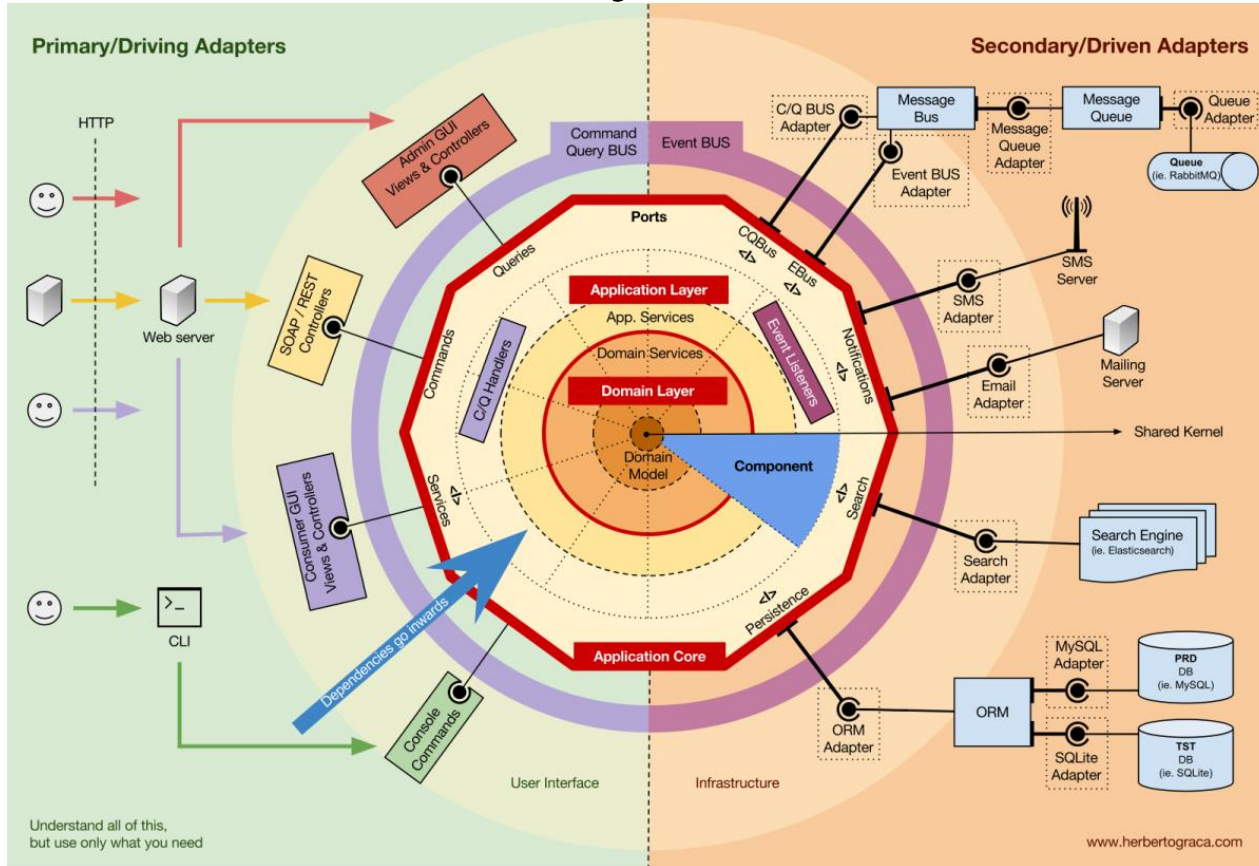


Сначала термины

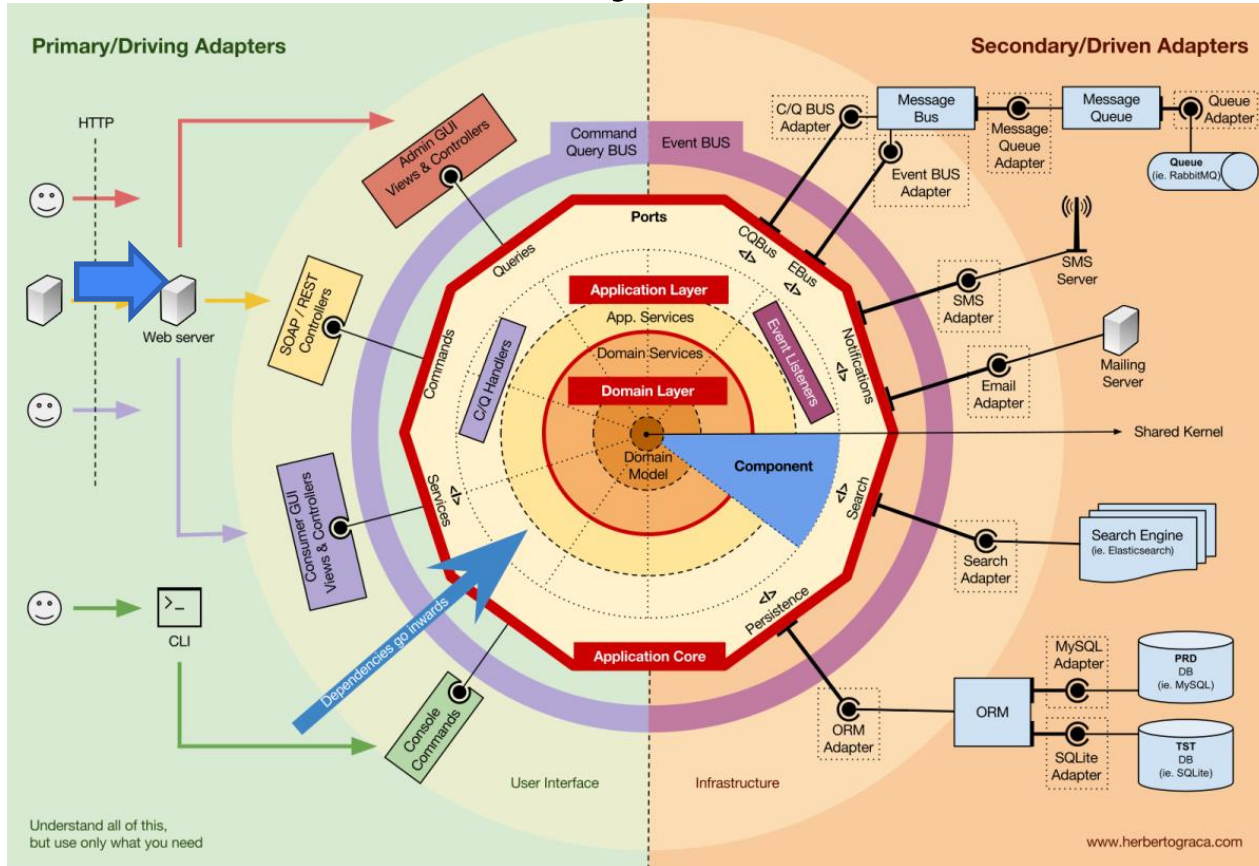
- Benchmark
- Throughput
- Latency
- Персентили
- SLA (пример):
 - 50% percentile < 50ms -> median
 - 99% percentile < 150ms
 - 100% percentile < 2s



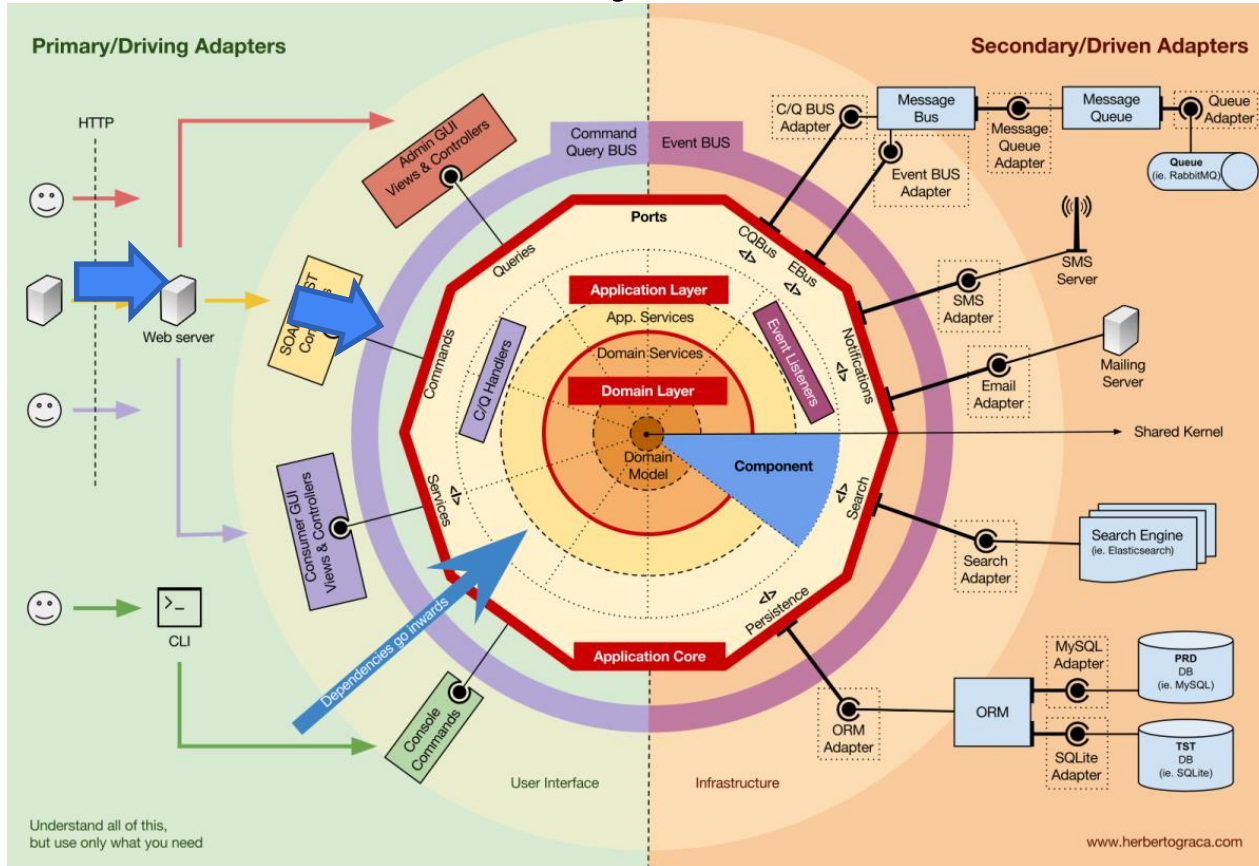
SLA - почему это важно?



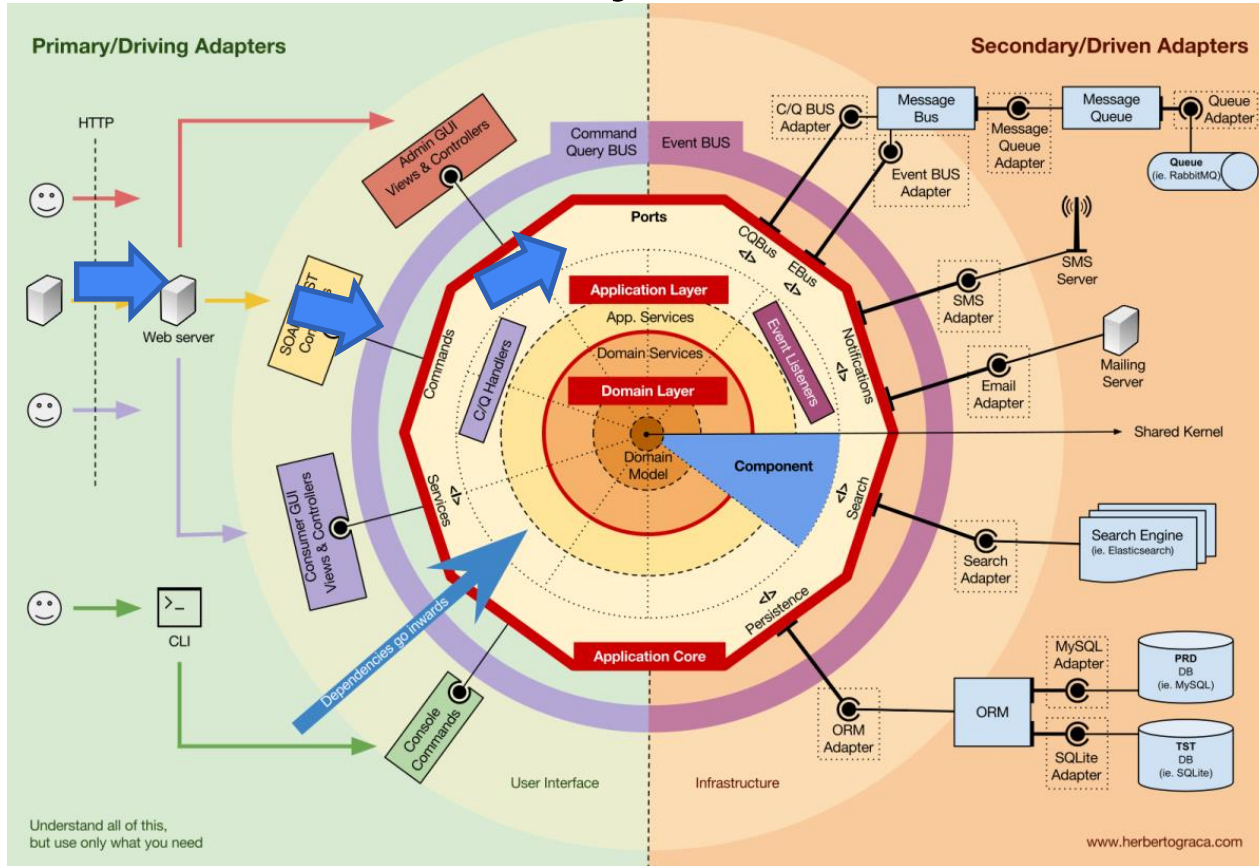
SLA - почему это важно?



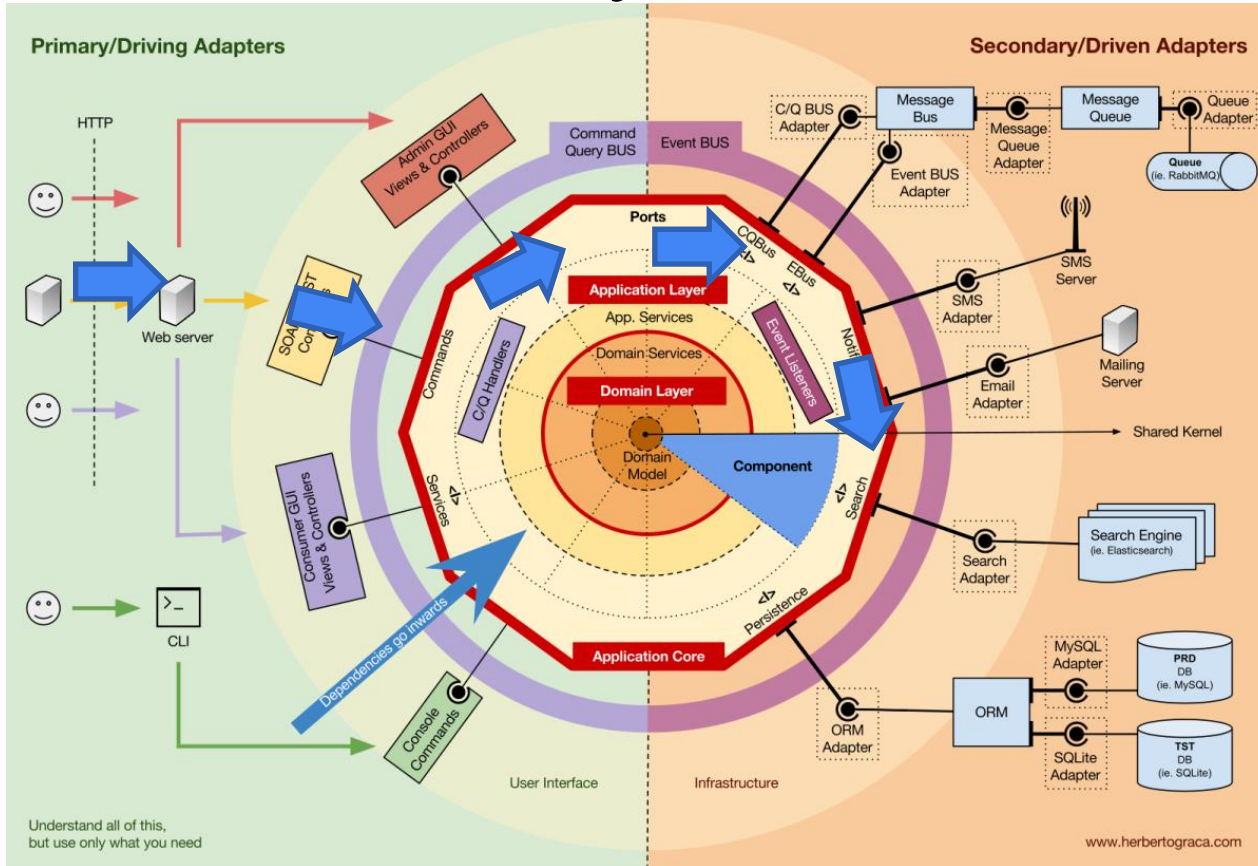
SLA - почему это важно?



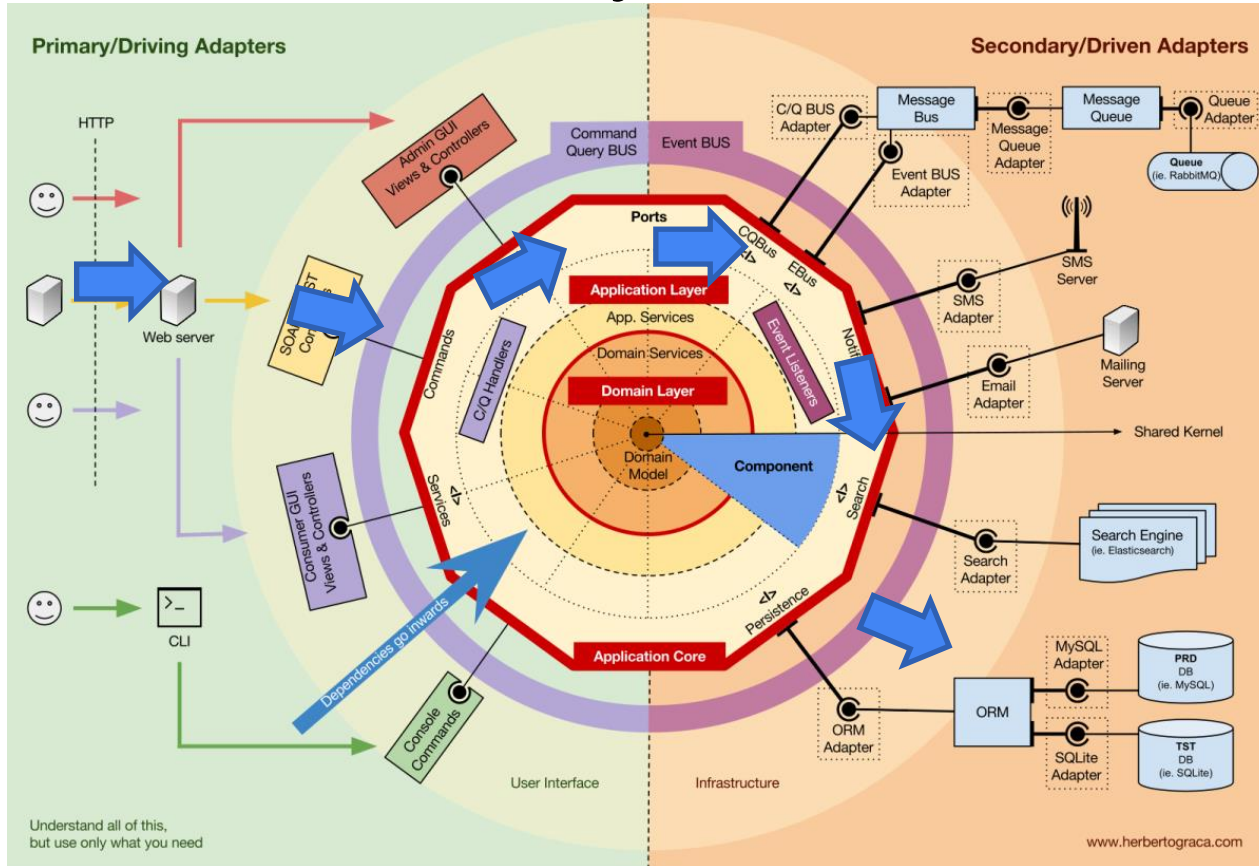
SLA - почему это важно?



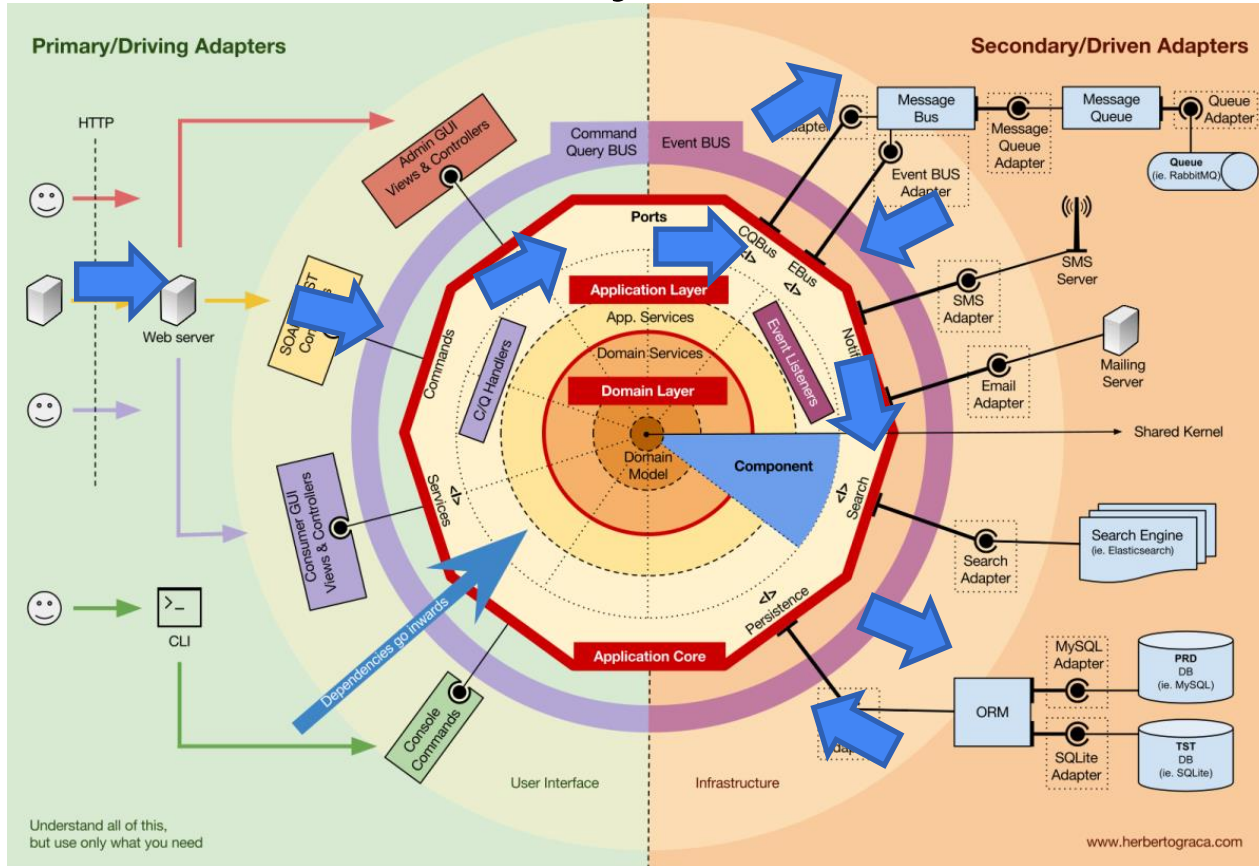
SLA - почему это важно?



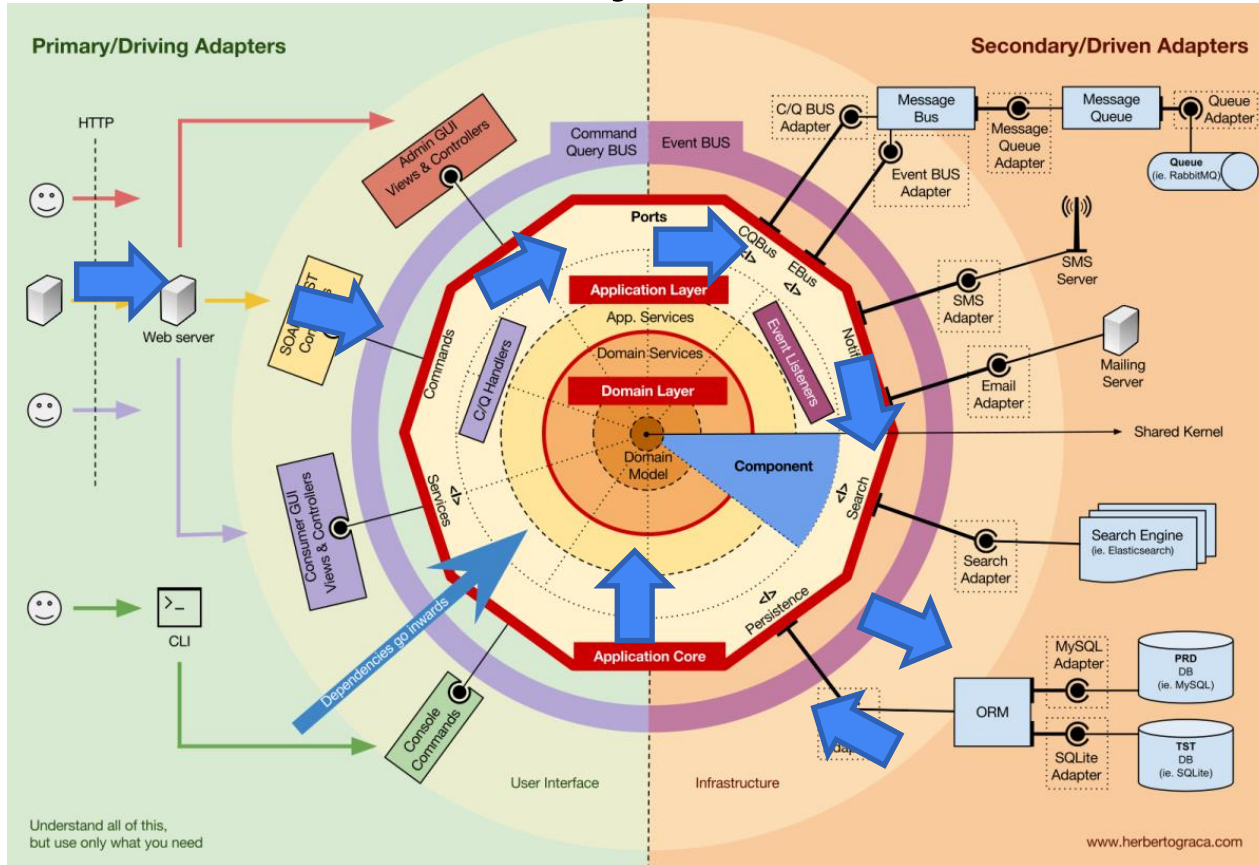
SLA - почему это важно?



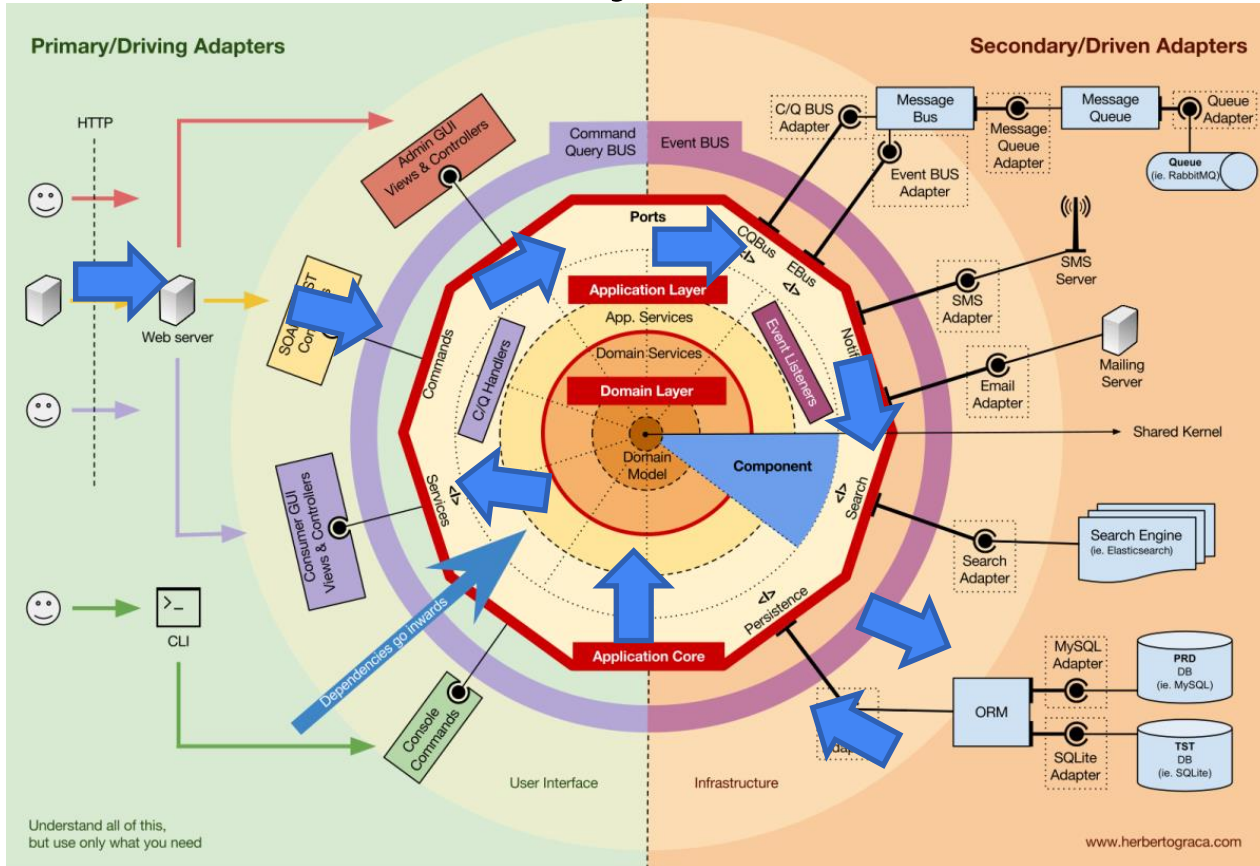
SLA - почему это важно?



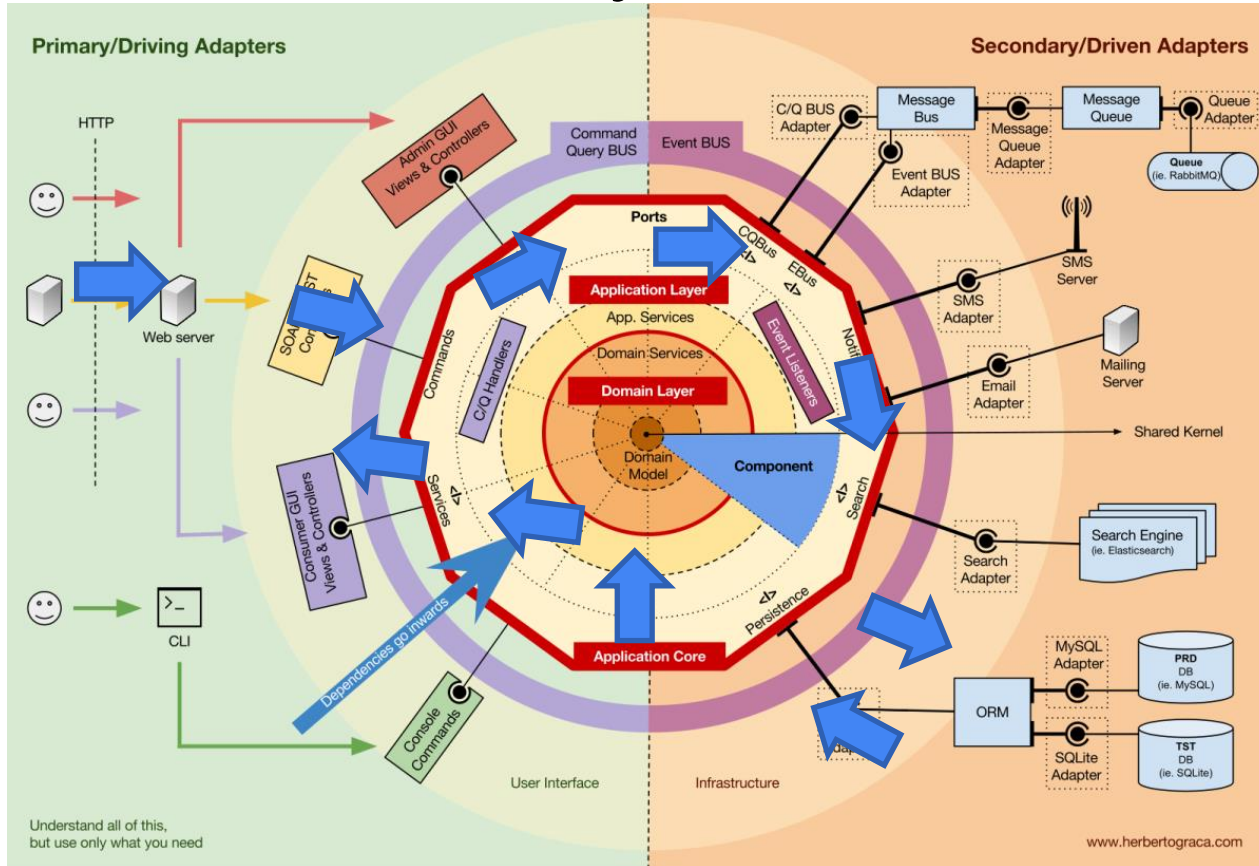
SLA - почему это важно?



SLA - почему это важно?

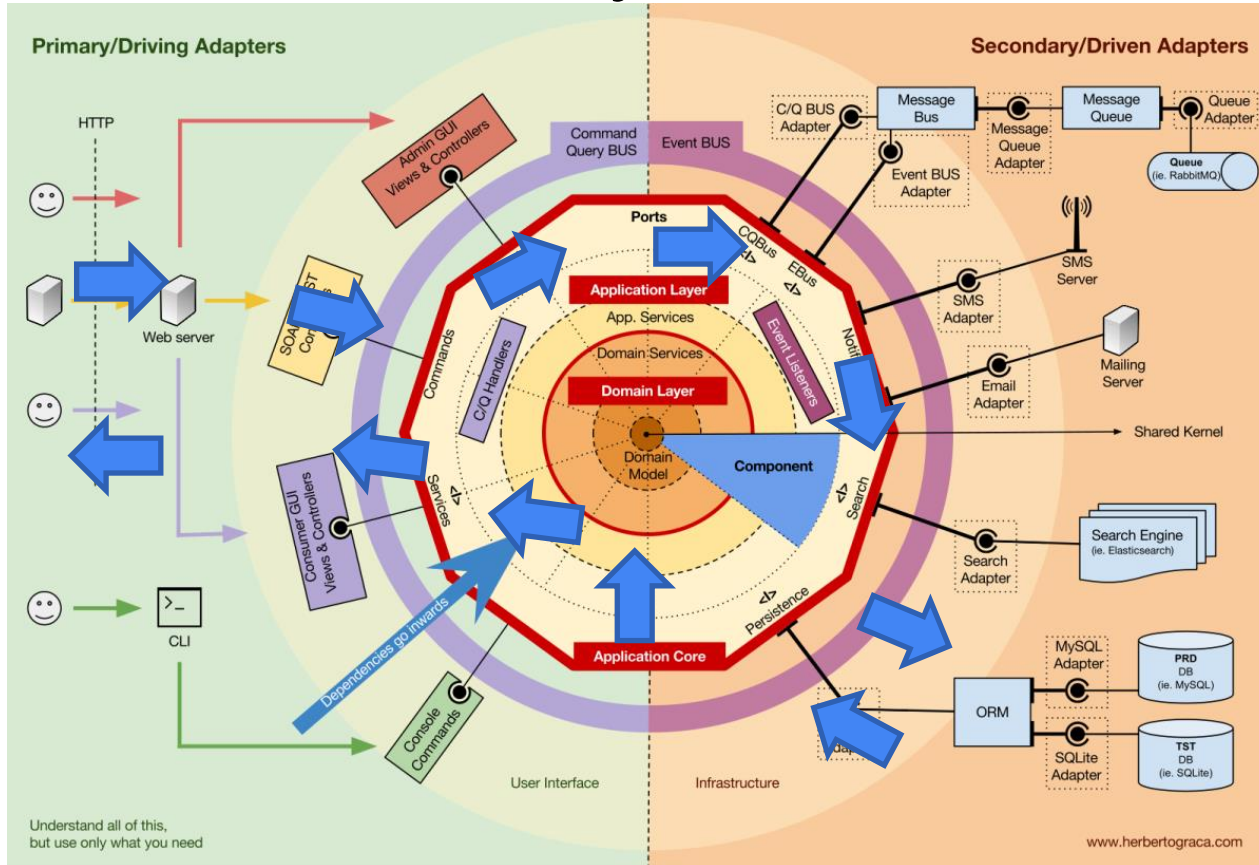


SLA - почему это важно?

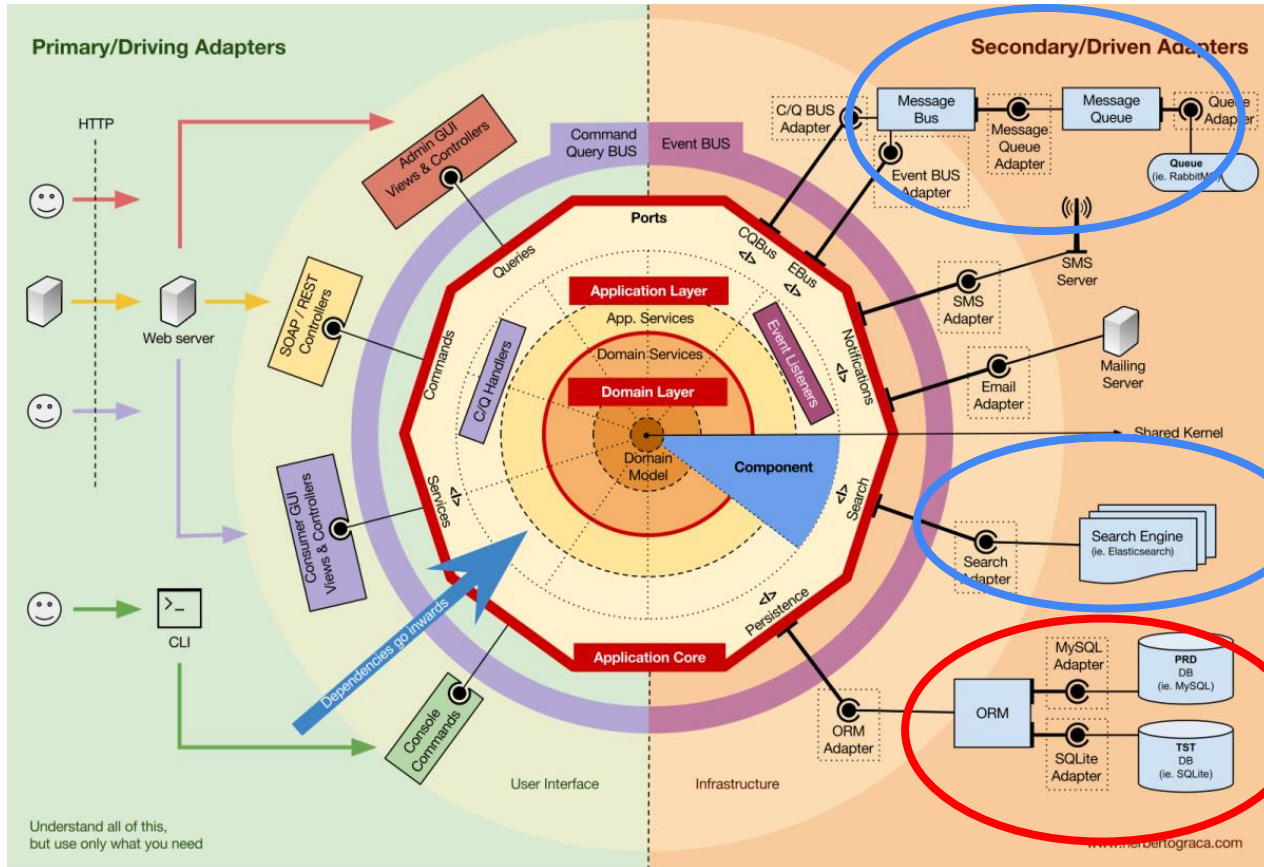


SLA - почему это важно?

2сек
Pmax!



SLA для каждой компоненты



А какие SLA в Twitter?

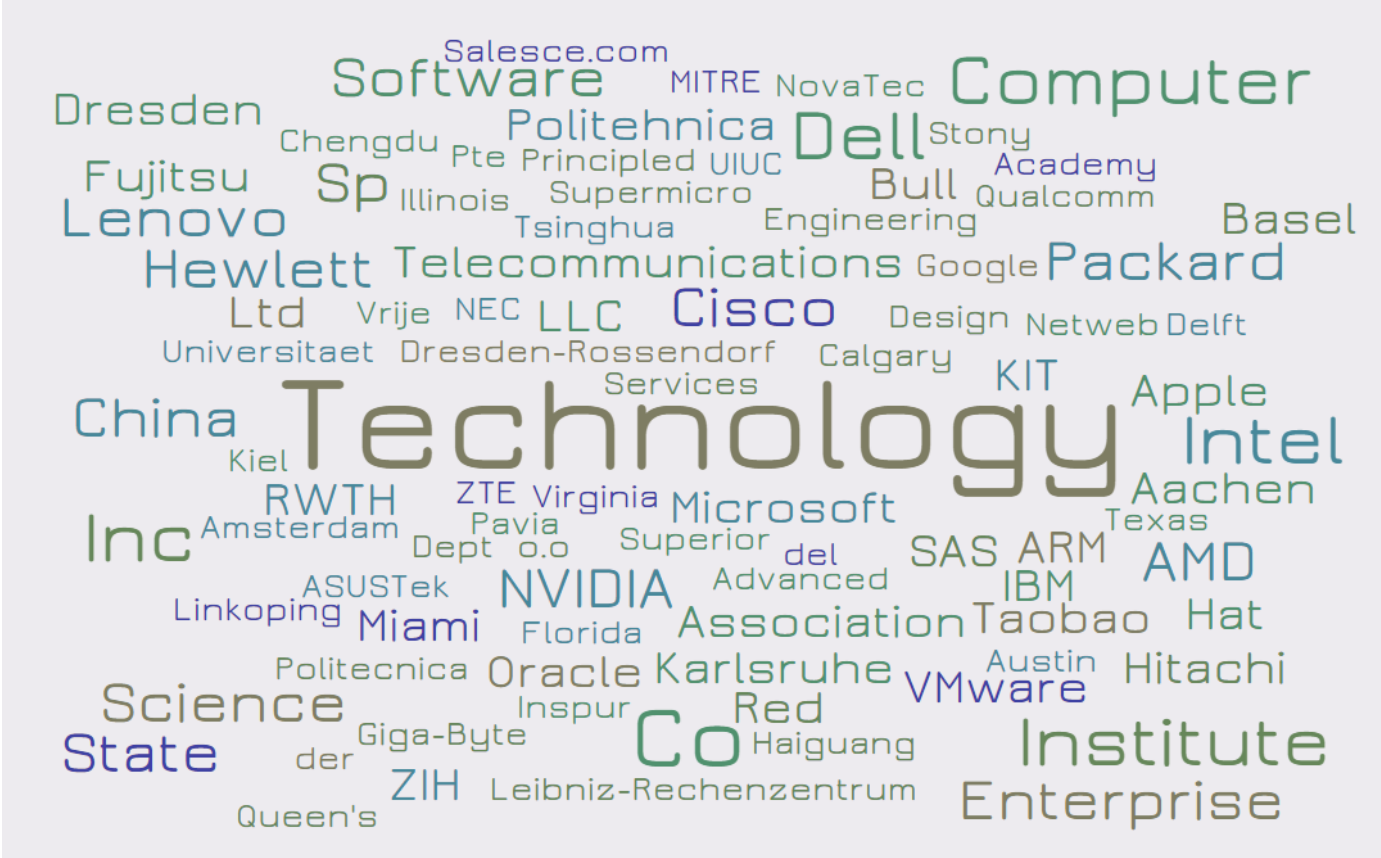


SPECjbb2015

<https://www.spec.org/jbb2015/>

- Основная Java Enterprise Бенчмарка на сегодняшний день
- Работа основана на базе модели всемирного супермаркета с IT инфраструктурой, которая обрабатывает запросы на продажу, онлайн заказы и data-mining операции
- Разработана консорциумом SPEC

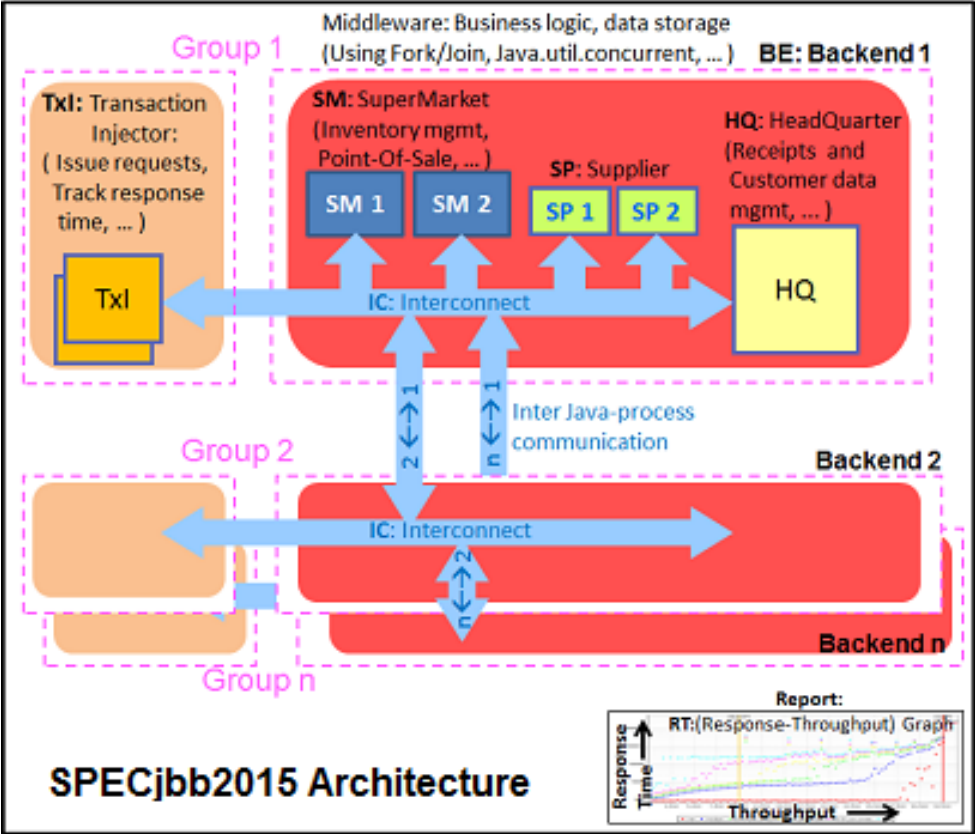
SPEC.org - The Standard Performance Evaluation Corporation



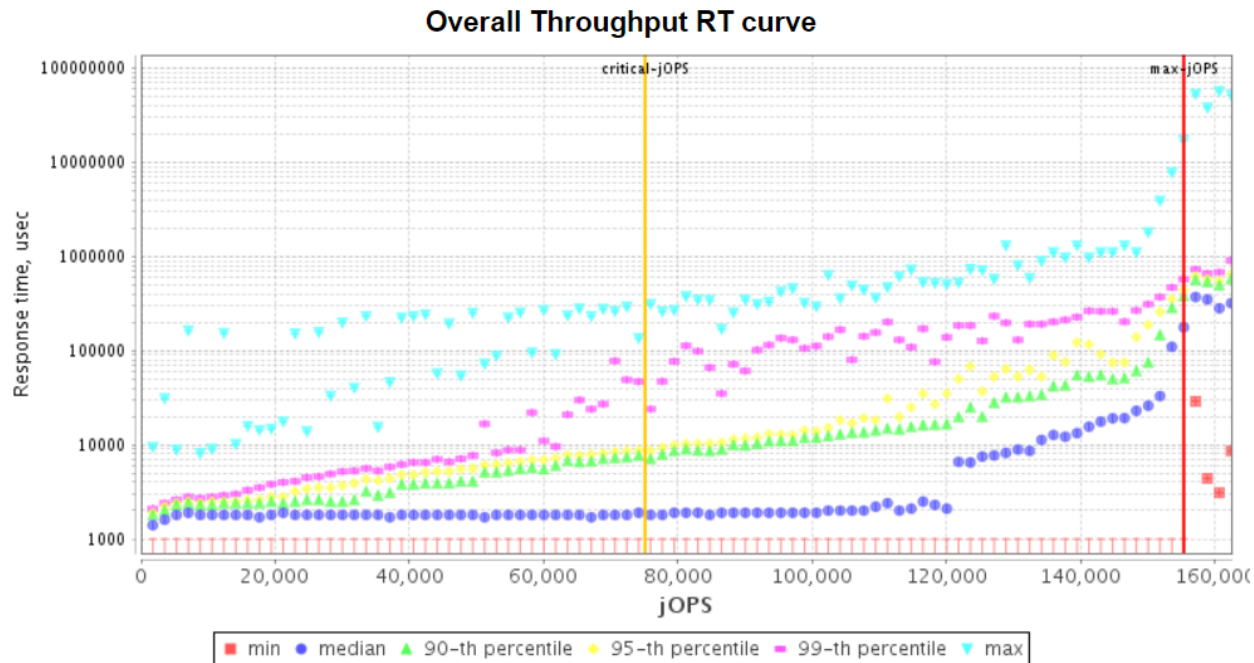
SPECjbb2015

- Основная метрика - пропускная способность *:
 - Пиковая (Max jOps)
 - В пределах допустимого SLA (Critical jOps)
** где время отклика варьируется от 10 до 100 мс*
- Поддерживаются разные конфигурации прогонов
- Цель: анализ проблем производительности на разных уровнях системного стека, включая железо, ОС, JVM и уровни приложения
- Поддержка виртуализации в клауде
- И тд

SPECjbb2015

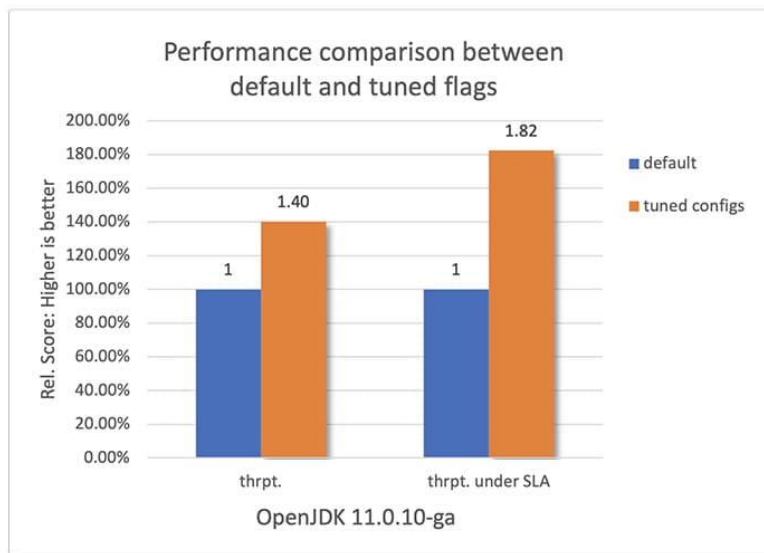


SPECjbb2015 или принцип работы «синхрофазатрона» или Throughput Under SLA в действии



SPECjbb2015 – где используется?

- Например для определения производительности нового инстанса в клауде <https://blogs.oracle.com/javamagazine/post/java-arm-runtime-switches-benchmarks>



Опции JVM, которые используются в публикациях

- пример:
- `-server -Xms330g -Xmx330g -Xmn310g -XX:+AggressiveOpts -XX:+UseLargePages -XX:-UsePerfData -XX:-UseAdaptiveSizePolicy -XX:SurvivorRatio=48 -XX:AllocatePrefetchInstr=2 -XX:TargetSurvivorRatio=95 -XX:MaxTenuringThreshold=15 -XX:LargePageSizeInBytes=2m -XX:+UseParallelGC -XX:+UseNUMA -XX:+AlwaysPreTouch`

Опции JVM, которые используются в публикациях

- пример:
- `-server -Xms330g -Xmx330g -Xmn310g -XX:+AggressiveOpts -XX:+UseLargePages -XX:-UsePerfData -XX:-UseAdaptiveSizePolicy -XX:SurvivorRatio=48 -XX:AllocatePrefetchInstr=2 -XX:TargetSurvivorRatio=95 -XX:MaxTenuringThreshold=15 -XX:LargePageSizeInBytes=2m -XX:+UseParallelGC -XX:+UseNUMA -XX:+AlwaysPreTouch`



Stop-The-World
Garbage Collector
обеспечивает лучшее
latency на
SPECjbb2015!!!



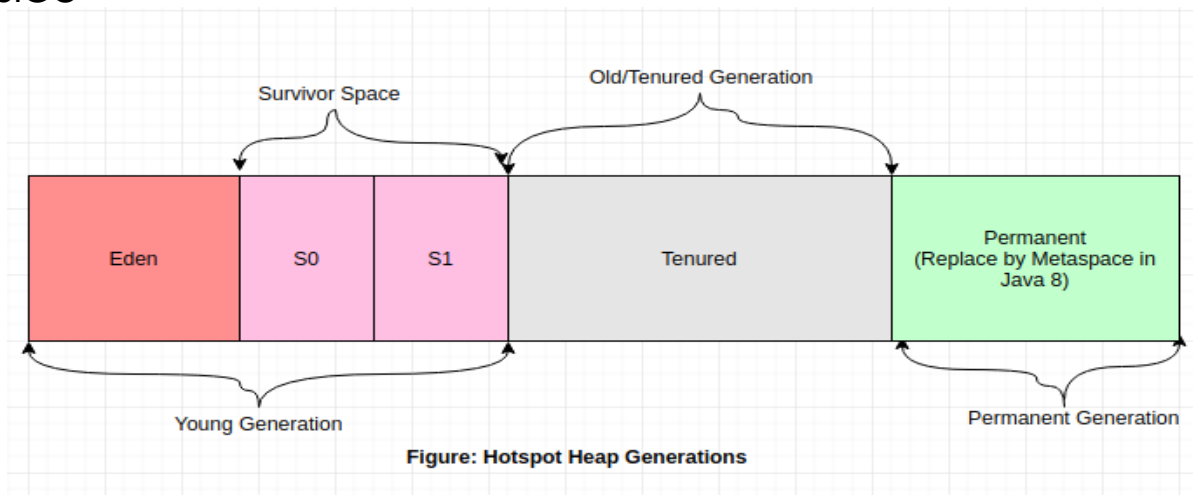
Хорошая Latency != STW GC

В мире много concurrent GC

- CMS
- G1
- Azul C4
- ZGC
- Shenandoah

Почему так происходит?

ParallelGC



Почему так происходит?

-Xms330g -Xmx330g -Xmn310g



Почему так происходит?

-Xms330g -Xmx330g -Xmn310g

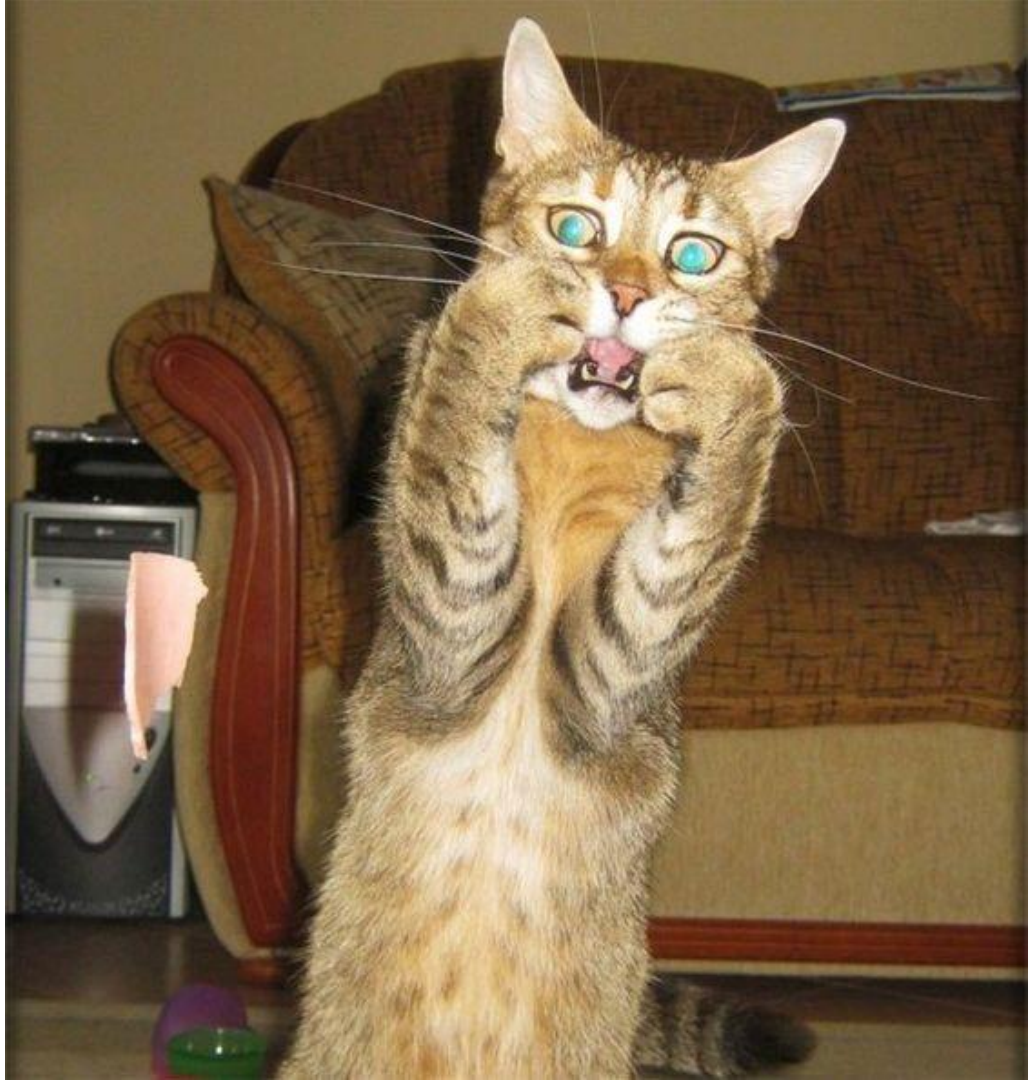


Подобраны магические значения не имеющие практической применимости!

Как в анекдоте



Enterprise Java Benchmarking



SPECjbb2015: Что в итоге?

Плюсы

- Алгоритм тестирования производительности SPECjbb2015 эффективен
- Дает объемлющие данные по поведению системы под разной нагрузкой
- Он позволяет посмотреть на проблему производительности системно

Минусы

- Реализация функционирования супермаркета хромает
- Нельзя тестировать реальные приложения
- Вы где-нибудь видели продуктовую систему, в которой большинство аллоцируемых объектов «die young»?
- Продуктовая система ориентированная на response time никогда не работает на ParallelGC!

Будет ли Community ответ?

Будет ли Community ответ?

Нужен инструмент для тестирования реальных приложений современным способом!

Throughput Under Service Level Agreement

TUSSLE

(пер. борьба, драка)

by Azul

Методология Tussle

- Схожа со SPECjbb2015
- Увеличиваем target load до момента нахождения Max Throughput
- Далее снова повторяем цикл увеличения target load до момента, когда мы ломаем SLA
- Используем в Azul
- <https://github.com/AzulSystems/tussle-framework>
- Есть ключевые отличия

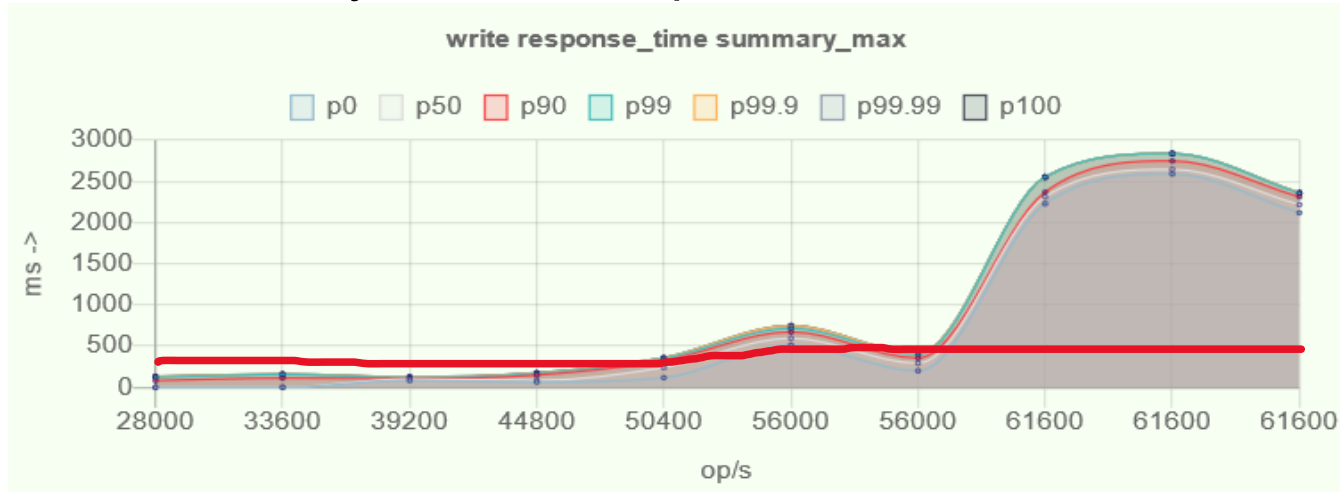
2 ключевых отличия Tussle от SPECjbb2015

- Для тестирования используем реальные приложения
- Такие продукты на рынке как:
 - Kafka
 - Cassandra
 - Solr
 - Hazelcast, ...
 - Реальные приложения
- Плавающие «окна»

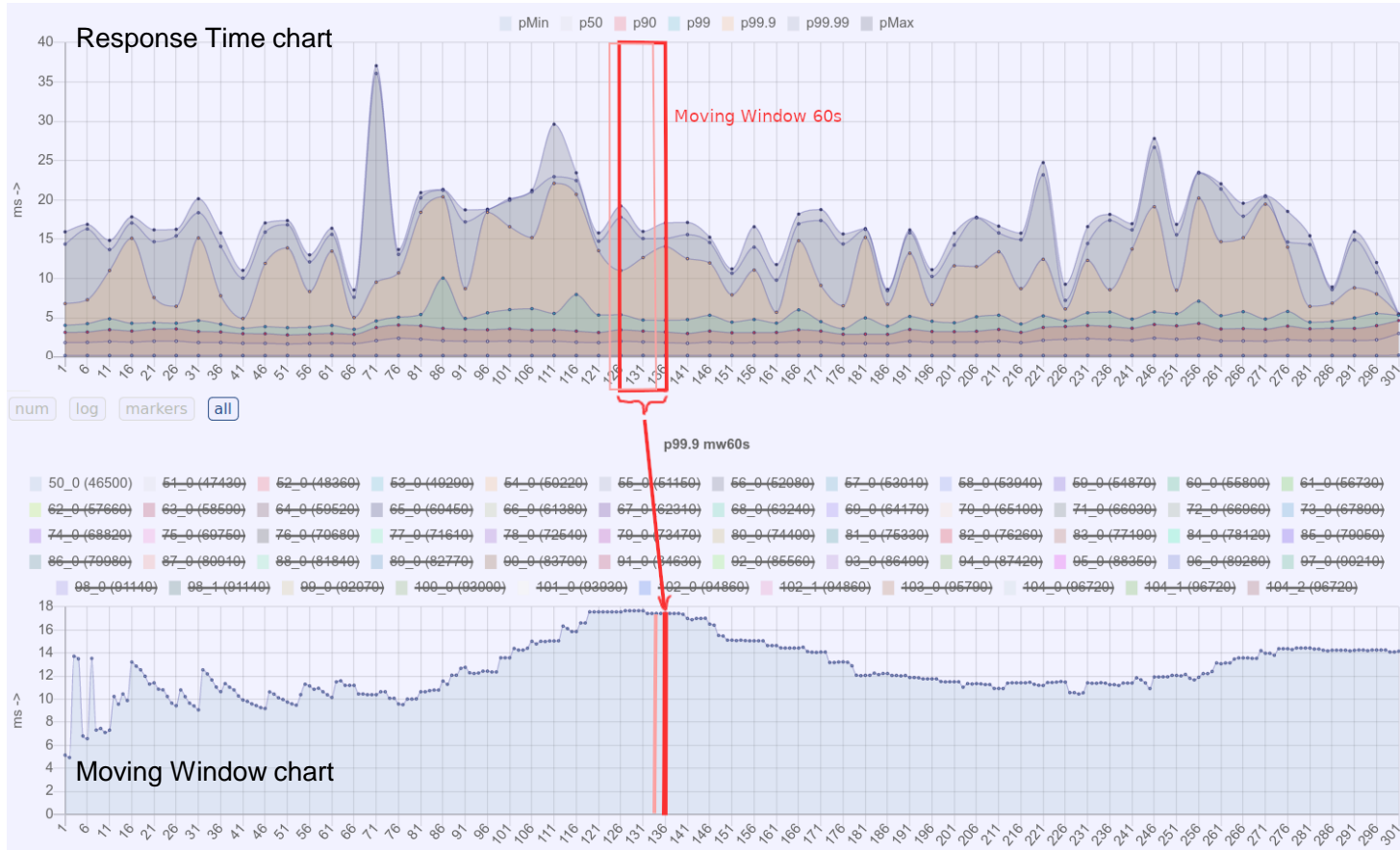
Tussle Concepts

- Service Level Expectation

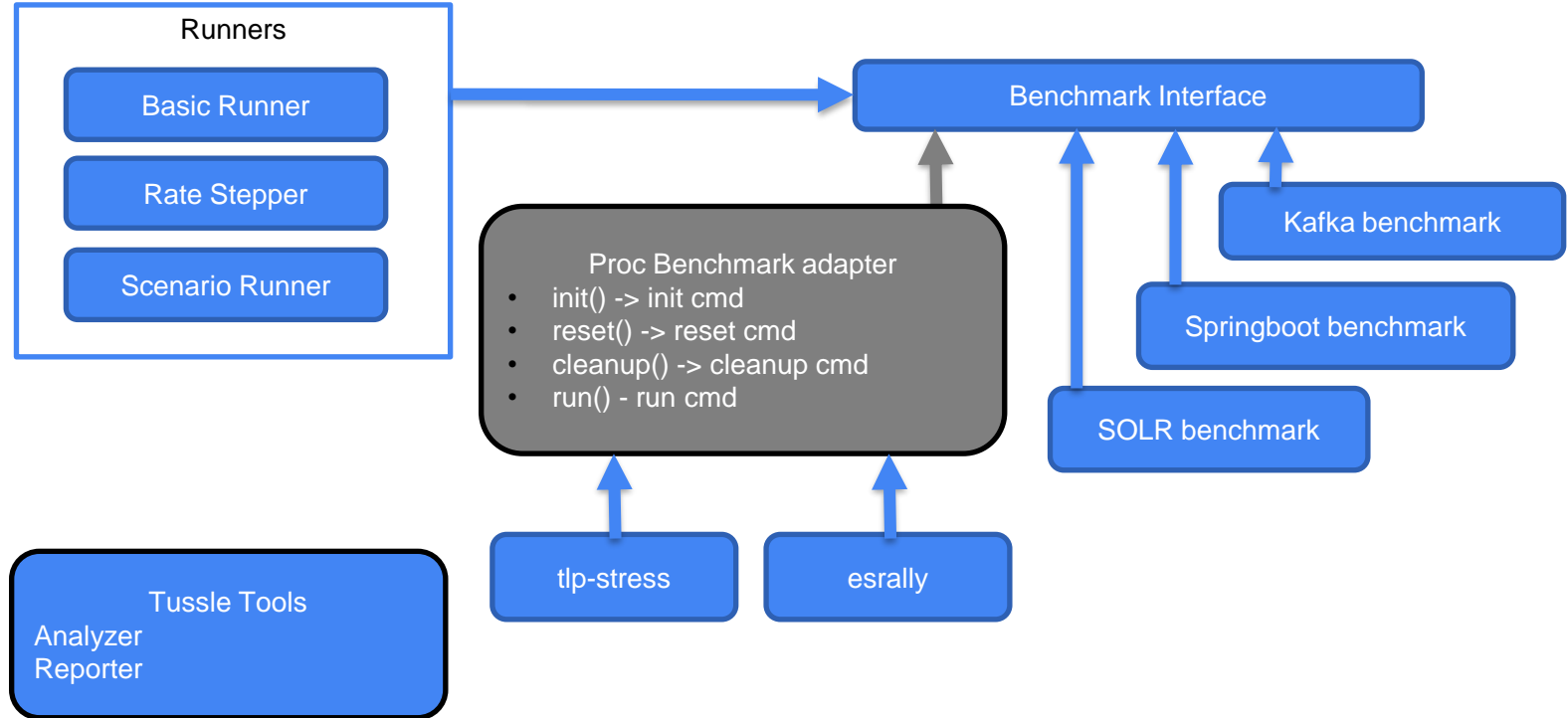
- SLE == SLA
- pass/fail (met or not)
- может быть несколько SLEs
- пример: 99% перцентиль времени отклика в течении любого 10ти секундного интервала должен быть < 20мс



Tussle - Moving Window



Tussle Framework



ДЕМО

Пример, где Tussle помогает на примере Kafka

OpenJDK



VS



Azul Platform Prime



<https://www.azul.com/blog/the-practical-roi-of-running-kafka-on-azul-platform-prime/>

Что в итоге

- Можно ли верить современным Enterprise бенчмаркам?
- Фиксированного target load-а недостаточно
- Tussle фреймворк предоставляет современную методологию тестирования Enterprise приложений
- Хотите затащить Tussle методологию в популярный тулинг – you're Welcome!

Thank you!