Эрланг и Эликсир
Зачем они вообще нужны?
И зачем они нужны лично вам?

Юра Жлоба

Wargaming.net

Ноябрь 2019

Чуть-чуть истории

Это важно для понимания сути.

Агнер Краруп Эрланг

Датский математик, статистик и инженер,

автор ”Теории массового обслуживания”.

Теория массового обслуживания

1909

Теория очередей, Queueing theory

Математическая модель для оценки пропускной способности телекоммуникационных сетей

Теория массового обслуживания

- не только сетей, но и
- дорог (автомобильных, железнодорожных и т.д.)
- больниц
- складов, магазинов

Эрланг, это

- датский ученый
- единица пропускной способности сети
- язык программирования

highload 80-x

Эрикссон (Ericsson)

телекомуникационное оборудование и услуги.

highload 80-x

- сложное оборудование
- сложный софт
- большой траффик
- жесткие требования по доступности сервиса

Совсем краткая история

- 80-е – разработка языка
- 90-е – использование внутри компании Эрикссон
- 2000-е – выход в мир

2007 год, закон Мура больше не работает

Предел наращивания частот процессоров.

Рост количества процессоров и их ядер.

Необходимость в разработке многопоточных программ.

2007 год, закон Мура больше не работает

Рост интереса к ФП.

Копирование идей ФП в мейнстримовые языки.

2011 год, появление Эликсир

Жозе Валим (José Valim)

Один из основных разработчиков Ruby on Rails.

Erlang VM

Представляет собой операционную систему в миниатюре:

планировщик процессов, управление памятью,
дисковый и сетевой IO.

Классические фичи

- Concurrency
- Fault Tolerance
- Distribution
- Hot Code Upgrade

Не классические, но очень полезные фичи

- Symmetric Multiprocessing
- Actor Model
- Soft Real Time
- Garbage Collection
- Tracing

Concurrency

Процессы являются базовой сущностью языка.

Concurrency

Процессы легковесны,
их можно создавать десятки и сотни тысяч.

1024 - 134,217,727 $(2^{10} - 2^{27})$

дефолтное значение 262,144 (2^{18})

Concurrency

Запуск нового процесса - 3-5 микросекунд.

На старте поток занимает 2696 байт,
включая стек, кучу и память под свои метаданные.

Concurrency

Нет разделяемой области памяти,

каждый процесс имеет свою изолированную память.

Concurrency

Ошибки в процессах также изолированы,

падение одного процесса не влияет на работу остальных.

Fault Tolerance

Три уровня обработки ошибок.

Fault Tolerance

Перехват исключений

Supervisor

Кластер

Distribution

Горизонтальное масштабирование.

Устойчивость в том числе и к аппаратным авариям.

Location Transparency

Процессы общаются отправкой сообщений друг другу.

При этом не важно, находятся ли они на одном узле,

или на разных.

Hot Code Upgrade

VM позволяет загрузить в рантайм новую версию кода модуля,

и переключить выполнение процесса

со старой версии кода на новую,

сохранив состояние памяти процесса.

Symmetric Multiprocessing

Запускается несколько планировщиков,
соответственно количеству процессорных ядер.

Каждый планировщик использует один процесс ОС,
и поверх него запускает эрланговские процессы.

Планировщики умеют балансировать нагрузку,
перераспределяя потоки между собой.

Symmetric Multiprocessing

VM линейно масштабируется на большое количество ядер.

Проверено на практике на машинах с 1024 ядрами.

Actor Model

Система состоит из акторов, которые действуют паралельно
и независимо друг от друга.

Акторы общаются друг с другом
с помощью отправки сообщений (message passing).

Данные копируются, поток не может изменить
данные другого потока.

Отправка сообщений является асинхронной.

Soft Real Time

VM позволяет строить системы реального времени.

То есть, системы, где требуется предсказуемое время ответа.

Soft Real Time

- вытесняющая многозадачность (preemptive scheduling)
- настраиваемый IO
- особенности сборки мусора (garbage collection)

Garbage Collection

Сборка мусора в ФП проще, благодаря иммутабельным данным.

GC использует обычный алгоритм с двумя поколениями данных.

Но есть важная особенность...

Garbage Collection

Отдельная сборка мусора для каждого процесса.

Блокирует только один процесс.

Срабатывает быстро.

Нет эффекта **stop world**, характерного для JVM.

Tracing

- жизненный цикл процессов
- отправка и получение сообщений
- вызовы функций, аргументы, возвращаемые значения
- состояние процессов
- работа планировщика
- потребление памяти
- работа сборщиков мусора

Tracing

Можно узнать почти все о работе ноды.

Сложность в том, чтобы
собрать именно ту информацию, которая нужна.

Языки для Erlang VM

Erlang, Elixir

Joxa, LFE

Alpaca, Gleam

и другие

Эрланг

Чем хорош Эрланг?

Эрланг

Простой язык, который можно быстро освоить.

Консервативный, в него не часто добавляются новые фичи.

При этом быстро развивается виртуальная машина.

(Язык и виртуальную машину развивает одна команда).

Эрланг

Язык водопроводчиков.

Школьные задачи про трубы и бассейны.

В центре внимания: потоки данных и хранилища данных.

RabbitMQ и Riak.

Эрланг

Язык не про то, чтобы создать сложные абстракции,
сложные модели данных,
и сложные взаимодействия между ними.

(Для этого в Эрланг мало выразительных средств).

Язык про то, чтобы эффективно использовать ресурсы.

Эрланг

Основные проблемы, которые решают эрлангисты:

нехватка ресурсов,

не использование имеющихся ресурсов,

обе проблемы одновременно.

Эрланг

Типичная задача:

Пропускная способности системы ниже, чем нужно,
но системе есть свободные ресурсы:
CPU, память, IO.

Найти узкое место.

Устранить его.

Эрланг

Сильная сторона:

способность держать одновременно
много **долгоживущих** соединений/сессий.

Эрланг

Типичный пример: чат-сервер.

Сотни тысяч одновременных подключений.

Длительность сессии – от десятков минут до нескольких часов.

Эрланг и веб

Веб-сервер: Cowboy

Веб-фреймворк: нет

ORM: нет (мы пишем SQL руками)

Эрланг и веб

Бэкенд сервис, API для других сервисов.

Не работает с UI.

Не работает непосредственно с пользователем.

Эликсир

Может делать все, что может Эрланг.

И может больше,
закрывает слабые стороны Эрланг.

Эликсир

Мощный, выразительный язык.

Можно создавать сложные абстракции,
сложные модели данных,
сложные взаимодействия между ними.

Эликсир

Расширяемый,
имеет богатые средства метапрограммирования.

Веб-фреймворк, ORM – в наличии.

Библиотеки (hex), тулинг (iex, mix).

Эликсир

В некотором роде – противоположность Эрланг.

И многие эрлангисты его не любят.

Я тоже принял Эликсир не сразу, долго сопротивлялся.

Вопросы?