

Киллер фичи Эрланг

Юра Жлоба для FuncBy

Потоки в Эрланг, немного бенчмарков и цифр

Надежность, уровни изоляции ошибок

Отладка работающей ноды в продакшене

И пару слов о памяти и сборке мусора

Потоки в Эрланг

Эрланг – это много, много, много
легковестных потоков. Бла-бла-бла.

“Много” – это сколько?

“Легковестный” – это что значит?

Потоки в Эрланг

“Много” – это сколько?

Документация к erl
раздел Emulator Flags
Флаг **+P**

1024 (2^{10}) – 134,217,727 (2^{27})

262,144 (2^{18})

Потоки в Эрланг

Вы можете создать 134,217,727 потоков

(если у вас хватит памяти :)

Потоки в Эрланг

“Легковестный” – это что значит?

Сделаем пару бенчмарков:

Как быстро потоки стартуют

Сколько памяти получают на старте

Потоки в Эрланг

```
emacs@yura-ThinkPad-T520
1 -module(test).
2 -export([test/1, start/1, do_some_work/1]).
3
4 test(Num) ->
5     {Time, ok} = timer:tc(?MODULE, start, [Num]),
6     NumProcesses = erlang:system_info(process_count),
7     {Time, NumProcesses}.
8
9 start(Num) ->
10     [spawn_link(?MODULE, do_some_work, [N]) || N <- lists:seq(0, Num)],
11     ok.
12
13 do_some_work(_N) ->
14     timer:sleep(20000),
15     done.
```

Потоки в Эрланг

erl +P 1048576

Увеличиваем лимит потоков до 1М

Потоки в Эрланг

Thinkpad T520

4 ядра Intel Core i5, 2.50GHz

4Gb оперативной памяти
из которых свободны 2Gb

Потоки в Эрланг

Число потоков	Время запуска
200K	0.7 секунды
400K	2 секунды
600K	2.8 секунды
800K	11.7 секунд
1M	20 секунд

Потоки в Эрланг

200-600К потоков стартуют
за 3-5 **микросекунд**

800К-1М потоков –
проявляется нехватка памяти,
идет своппинг на диск

Потоки в Эрланг

```
emacs@yura-ThinkPad-T520
1 -module(test2).
2 -export([start/1, do_some_work/1]).
3
4 start(Num) ->
5     [spawn_link(?MODULE, do_some_work, [N]) || N <- lists:seq(0, Num)],
6     ok.
7
8 do_some_work(_N) ->
9     Info = process_info(self(), [total_heap_size, heap_size, stack_size, memory]),
10    io:format("~p ~p~n", [self(), Info]),
11    done.
```

Потоки в Эрланг

```
[{total_heap_size, 233},  
 {heap_size, 233},  
 {stack_size, 1},  
 {memory, 2696}]
```

Потоки в Эрланг

Куча: 233 машинных слова

Стек: 1 машинное слово

+ внутренние структуры данных

Всего: 2696 байт

Потоки в Эрланг

Легковесный поток:

Стартует за 3-5 микросекунд

Занимает 2696 байт памяти

(1М потоков зайнимают 2.5Gb памяти)

Потоки в Эрланг

Планировщики:

Их несколько, по числу ядер процессора

Балансируют нагрузку, перебрасывая
потоки между собой

Уровни изоляции ошибок

Try...catch

Правильно расставь

Правильно обработай

Уровни изоляции ошибок

Try...catch

В Эрланге есть, но используется редко

потому что ...

Уровни изоляции ошибок

Supervisor – специальный поток,
наблюдает за

Worker – рабочий поток

Уровни изоляции ошибок

Если в worker-потоке
возникает ошибка, то:

Запись ошибки и стэктрейса в лог

Рестарт потока с начальным состоянием

Уровни изоляции ошибок

Супервизоры объединены в дерево,
где worker-потоки являются листьями

Можно рестартовать лист,
или ветку, или все дерево

Уровни изоляции ошибок

Объединение нод в кластер

Резервные ноды, failover механизм

Уровни изоляции ошибок

Try...catch

Дерево супервизоров

Кластер

Отладка в продакшене

Подключиться к удаленной ноде

Выполнить любой код

Заменить код

Отладка в продакшене

Механизм трассировки

Встроен в виртуальную машину
на низком уровне

Мало влияет на производительность

Отладка в продакшене

Можно получать информацию:

О жизненном цикле потоков

Об отправке и получении сообщений

Вызовы функций

Состояние потоков

Отладка в продакшене

Можно получать информацию:

Работа планировщика

Потребление памяти

Работа сборщика мусора

Отладка в продакшене

Данные можно:

Наблюдать в консоли

Направить в файл

Обработать в кастомной функции

Обработать на другой ноде

Отладка в продакшене

Теоретически
можно узнать почти обо всем,
что происходит в ноде

Практически главная проблема
найти нужную информацию
во всем этом массиве данных

Сборка мусора

В функциональных языках проще
благодаря неизменяемым переменным

Сборка мусора

2 поколения объектов:

Молодые (короткоживущие)

Старые (долгоживущие)

Сборка мусора

Отдельный сборщик мусора
для каждого потока

Сборка мусора

Нет эффекта Stop World

Короткоживущие потоки
обходятся без сборки мусора

Долгоживущие с малым расходом
памяти тоже

Сборка мусора

Оказывает мало влияния
на производительность системы

Вопросы?