



эрливидео
поточный сервер

профилирование Эрливидео

Макс Лапшин
max@erlyvideo.org



Эрливидео

- Потокоеое видео
- Разные протоколы: MPEG-TS, RTMP, ...
- Высокая эффективность



Разные нагрузки

- Тысячи пар: один источник, один клиент
- Один источник, тысячи клиентов



Видео

- Один поток: 50 кадров/сек, 500 КБит/с
- 2000 клиентов — 100 000 кадров/сек.
- Суммарно 1 ГБит/с



RTMP

- У каждого клиента своё
- Каждому клиенту своя копия потока
- Данные общие, заголовки разные



RTMP

- Скорости не хватает — клиент заливает
- Ошибка в тайстемпе — клиент заливает
- Клиент заливает — надо что-то делать



Рост эрливи́део

- Начались проблемы на 800 клиентах
- Неверная архитектура
- Пришлось профилировать



Профилирование

- Девелопмент одно — продакшн другое
- Разные версии клиентов
- Разные скорости каналов
- Разные задержки на каналах



Профилирование

- Профилирование на продакшне
- Не отключая пользователей
- Без деградации системы



Инструменты

- Valgrind и профилировщики тормозят систему
- gdb вносит искажения в код



Инструменты

- Erlang предлагает другие подходы к профилированию



Erlang

- Данные и потоки выполнения объединены, как процессы ОС
- Обмен асинхронными сообщениями
- Message box у каждого процесса



Erlang

- Все данные немутабельные
- Процесс — рекурсивная функция, вызывающая себя с новым значением



Erlang

- Один процесс под каждого клиента
- Один процесс под каждый «объект»
- Вызовы методов строго последовательны



Lowload

- Процесс-«поток» шлет сообщения процессам-«клиентам»
- Клиенты упаковывают RTMP сами
- в системе $FPS * N$ сообщений



Тормоза

- Начинаются «лаги» и «залипание»
- Куда смотреть? Что мерять?



Мониторинг

- Меряем очереди сообщений
- `erlang:process_info(Pid,message_queue_len)`



Мониторинг

- Меряем загрузку CPU
- `etop:start()`
- мониторим общую память процессов



Узкие места

- В узких местах накапливаются сообщения
- Растет CPU, потребляемый процессом
- Накапливается память



Узкие места

- Процессу не хватает одного ядра
- Недообрабатывает сообщения
- Накапливается память
- Встают в ожидании клиентские процессы



Как боролись?

- Уменьшили количество сообщений
- Прямая запись в сокет
- Распараллелили запись в сокеты по ядрам



Результат

- Во много раз увеличили количество одновременных клиентов



Рекомендации

- Планировать количество проходящих сообщений
- Уменьшать работу внутри вызываемого процесса
- Раскладывать работу по процессам —
расползутся по ядрам



Выводы

- Профилировка приложений на erlang возможен без деградации качества обслуживания
- Методы поиска проблем несложны и эффективны



эрливидео
потоковый сервер

Вопросы?

Макс Лапшин
max@erlyvideo.org