Erlyvideo

success story of erlang usage

Max Lapshin
max@erlyvideo.org

Erlyvideo

How I Learned to Stop Worrying and Love Erlang

Erlyvideo

Question is:
should You choose erlang
for complicated project?

Erlyvideo

Answer for impatient:
YES

Streaming video

Streaming video

Complicated

Streaming video

Very hard and over-engineered protocols

Streaming video

Each client implements 10% of protocol.

You don't know which

Streaming video

Each client implements 10% of protocols with its own bugs.

You don't know which

Streaming video

Business logic is interlaced with infrastructure from top to bottom

Streaming video

Impedance between protocols makes programming hard

My background

- 2M LOC of C++ Corba threaded code

- Ruby. Easy and very cool. And very slow.

- ObjectiveC (C++-done-well). Evented.

Dissatisfaction

easy to start
hard to develop

Dissatisfaction

- hard resource management (leaks)

Dissatisfaction

- hard resource management (leaks)

- obtrusive profiling and introspection

Dissatisfaction

- hard resource management (leaks)

- obtrusive profiling and introspection

- spreading of errors: one error for whole VM

Dissatisfaction

- hard resource management (leaks)

- obtrusive profiling and introspection

- spreading of errors: one error for whole VM

- bad isolation of data

Dissatisfaction

- hard resource management (leaks)

- obtrusive profiling and introspection

- spreading of errors: one error for whole VM

- bad isolation of data

- difference between classes and data classes

Dissatisfaction

100 lines of C++ are fast

100 000 lines are slow

Dissatisfaction

Exponential growth of code connections and complexity

Java/C++ gives you marionette system with single rope

I build system with separate working details

What platform to choose?

About me

- Lazy

About me

- Lazy

- Inattentive

About me

- Lazy

- Inattentive

- Forgetful

About me

C++ isn't for me

Streaming video

I wanted cool tool for network server

Streaming video

- management of long living processes

Streaming video

- management of long living processes

- memory management like ARP

Streaming video

- management of long living processes

- memory management like ARP

- libev efficiency (no time to scale, must work)

Streaming video

- management of long living processes

- memory management like ARP

- libev efficiency (no time to scale, must work)

- bulletproof

Streaming video

- management of long living processes

- memory management like ARP

- libev efficiency (no time to scale, must work)

- bulletproof

- cool

Long-living clients

You have to deal
not only with code
but with errors in it

ERror handling LANGuage

Erlang

It is really working.
Out of the box.

Erlang

Data and threads
are glued in one entity

Erlang

Isolation with messaging simplify divisioning

Erlang

Happy-path programming

Erlang

Paradigm shift to programming <u>active objects</u>

Practice

- 3 months of Erlang development equal to 2 years of Java development

Practice

- 3 months of Erlang development equal to 2 years of Java development

- Linear speed of development

Practice

- 3 months of Erlang development equal to 2 years of Java development

- Linear speed of development

- Production stability: no memory leaks

Practice

- 3 months of Erlang development equal to 2 years of Java development

- Linear speed of development

- Production stability: no memory leaks

- Ease of adding programmers

Problems

- Speed on GBp/s (100 000 frames per second)

Problems

Message passing sometimes doesn't work

Problems

Fixing with architecture change

Problems

- Speed on GBp/s (100 000 frames per second)

- Records and packages

Problems

Hard to deploy plugin architecture

Problems

Fixing with isolating records in modules

Results

2 years of development

Results

Full satisfaction with Erlang

Plans

Move to HTTP

Plans

10 GBp/s

Plans

Erlang will still be required

Questions?

Max Lapshin
max@erlyvideo.org
http://erlyvideo.org