Erlang, the big switch in social games

Paolo Negri @hungryblank

About this talk

Erlang adoption @

how we did it

and what we found

Social Games

- Run in the browser
- Published on social networks
- Popularity measured in millions of daily users

Social Games

Flash client (game)

HTTP API

wooga
world of gaming

Social Games

Flash client

- Game actions need to be persisted and validated
- 1 API call every 2 secs

wooga
world of gaming

Social Games

HTTP API

- @ 1 000 000 daily users
- 5000 HTTP reqs/sec
- more than 90% writes

wooga
world of gaming

Social Games @wooga

HTTP API

- up to 1 300 000 daily users on a single game
- team of 2 people
- develop from scratch
- AND do deployments + live operations
- Releasing on weekly schedule


The hard nut
http://www.flickr.com/photos/mukluk/315409445/

The hard nut

60K queries/sec

Read/Write ~ 1

@ 1mln daily users your SQL / NOSQL data layer is the main technical challenge

The hard nut roots

Stateless app servers

- very easy to scale (add one more)
- easy to develop
- Plenty of ready frameworks

wooga
world of gaming

The hard nut roots

Stateless app servers

- high pressure on data layer
- extreme query/req optimization means heavy refactoring
- query/req of 2 best possible ratio

wooga
world of gaming

The case for erlang

"A gaming session is a stream of close requests altering repeatedly the same game state"

The case for erlang

"What about application servers that understand and represent efficiently the concept of a gaming session?"

The case for erlang

"Which language or framework for:"

- Long lived state
- Safe handling of a multitude of independent, different states
- All the above on a big cluster of machines

The case for erlang

- gen_server, gen_fsm good approach to state handling
- Processes are very good at isolating failures
- clustering is a concept in the language itself

Pitching the idea

"This architecture let us leverage factors we can't currently leverage in order to solve the scaling challenge"

"By the way erlang seems to be a very good match for this architecture"

Pitch won!

CTO says:

- Develop proof of concept

Works?

- Hire 1 developer with erlang experience
- Go on and develop the backend for our new game

Pitch won!

CTO says:

- Develop proof of concept

Works? Yes!

- Hire 1 developer with erlang experience
- Go on and develop the backend for our new game

How to hire erlang devs?

Hiring

- mailing list
- Linkedin
- No agents
- No paid ads

Hiring

- 2 months
- Handful of applications with the right skills
- Hired one dev

Hiring

We also got quite a few...

"I'm really experienced in _something else_ but I would like to learn and work with erlang"

From people that are *good* for real

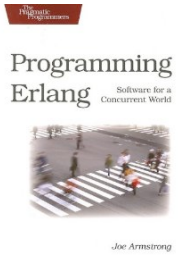
How to

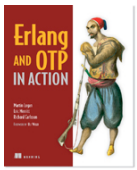
With ERLANG

http://www.flickr.com/photos/jakeandlindsay/5524669257/

getting started

Nice books around





wooga
world of gaming

getting started

http://www.erlang.org/doc

```
keytake(Key, N, TupleList1) -> {value, Tuple, TupleList2} | false
```

Types:

```
Key = term()
N = 1..tuple_size(Tuple)
TupleList1 = TupleList2 = [Tuple]
Tuple = tuple()
```

Docs are complete, well written but sometime missing examples

getting started

gen_server, gen_tcp, gen_*

Very well documented and awesome by default

getting started

The rest of OTP was black magic and foggy until the book was published

Still official howtos and complete docs would be nice to have

getting started

Learn you some erlang[1]

- Hands on book
- Plenty of small examples
- Written with the beginners in mind

[1] http://learnyousomeerlang.com

http://www.flickr.com/photos/loonatic/3871627355

tools we're using

rebar

- Dependency management
- Eunit integration
- Release packaging
- Enforces OTP standard

https://github.com/basho/rebar

tools we're using

eunit

- not tested =:= not done
- Eunit is a complete unit testing framework
- No guidelines/tools for testing gen_servers

http://www.erlang.org/doc/apps/eunit/chapter.html

tools we're using

erlymock

- mocking framework
- Mock all module or nothing
- Works, but very limited

https://github.com/nialscorva/erlymock

tools we're using

misultin

- library for building lightweight HTTP servers
- Tiny amount of code
- Performs well
- Good match for minimalist REST API

https://github.com/ostinelli/misultin

tools we're using

gproc

- Extended process registry for Erlang
- Used to map active users/sessions -> pids
- Works fine at 100K processes/node
- We use it only in local (single node) mode

https://github.com/esl/gproc

tools we're using

tsung

- Multi protocol (HTTP, XMPP) benchmarking tool
- Synthetic load only way to "scale proof" new project
- Able to test non trivial call sequences
- Can actually simulate game play

http://tsung.erlang-projects.org/

tools we're using

tsung

- Generates ~ 2500 reqs/sec on AWS m1.large
- Flexible but hard to extend
- Code base rather obscure, not hosted on github

http://tsung.erlang-projects.org/

Erlang brought with it few benefits we weren't explicitly looking for...

Freebies

Transactions

```
handle_call(Request, From, State) ->
    NewState = mylib:myfun(State),
    {reply, ok, NewState};
```

Immutability -> last sane state is always known and easily available

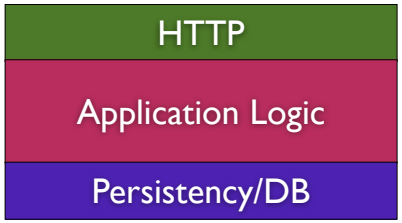
Freebies

Transactions

- Immutability (erlang)
- Functional approach (erlang)
- Careful persistency side effects handling (you)

Freebies

Decoupling



Freebies

Decoupling

- Switching from HTTP to sockets would take a day
- Changing persistency solution would take a day (data migration excluded)
- Both changes would leave application logic untouched

Freebies

Test a cluster on a box

- Multiple erlang nodes on a single machine
- Continuous integration for clustering logic!
- End to end test on a single box

Freebies

New point of view

- From the erlang point of view problems look different
- Approaches used in this project were ported to ruby projects
- Erlang makes for a good thinking exsercise


What we
LIKE
of
ERLANG

what we like

Low level runtime infos

- erlang:statistics
- erlang:system_monitor
- erlang:process_info

what we like

Performance tweaks

- Efficiency guide[1]
- Control on initial process heap
- Garbage collection can be measured

[1] http://www.erlang.org/doc/efficiency_guide/introduction.html

what we like

“erlang-questions” mailing list

- Very active and responsive
- “Big guns” open to help
- Threads go in depth on interesting topics

what we like

Github

- Thanks for sharing your code on github!
- Lots of answers to “how is that done?”
- Growing collection of erlang libs

what we like

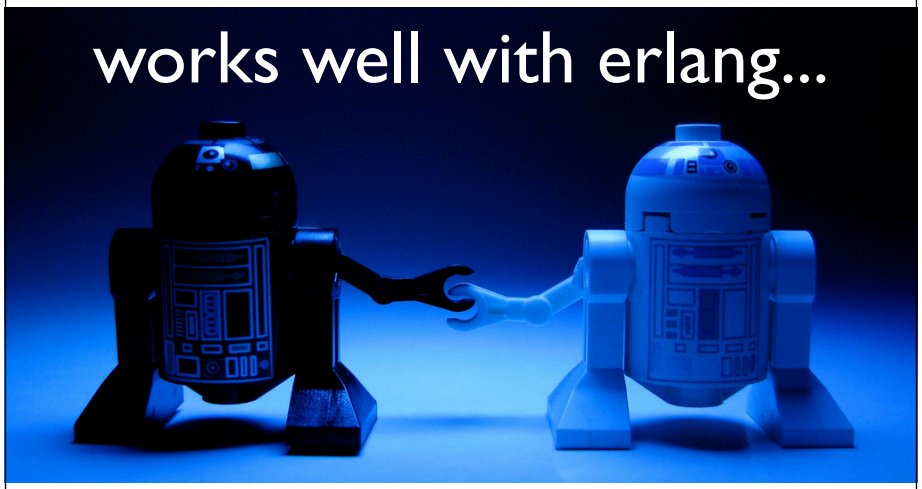
Github Tags Appeal

Please use tags on your github projects...

So we can bind rebar deps to a tag...

```
git push --tags
```

works well with erlang...



http://www.flickr.com/photos/legofenris/4563478042

works well with 

Rake!

- Integration test tasks
- Continuous integration setup/teardown
- Other tasks specific to the project

works well with 

Ruby test frameworks

- Flexible assertions
- Rapid modeling of API
- Low maintenance overhead
- Easy fixturing/mocking

works well with 

Redis

"An open source, advanced key-value store"

- In memory key value store
- Optional on disk persistency
- Optional replication

works well with 

Redis

Binary safe protocol

+

Binary safe datatypes

works well with 

Redis

200K ops/sec on a single thread

Serialization?

`term_to_binary/1`

`binary_to_term/1`

works well with 

Cloud (AWS)

- Erlang understands clustering out of the box
- Nodes joining/leaving handling
- Async I/O to compensate high latency
- Cloud works better if you're parallel


We're missing...

we're missing

HashMaps

```
user[:inventory][:food][:apples]
                => 1
User#user{inventory = Inventory}
                ...
```

we're missing

packages à la CPAN
rubygems...

we're missing

also...
github is great but which
one of 5 forks is the
"authoritative" one?

we're missing

will agner
A Giant Nebula of Erlang
Repositories be the
answer?

http://erlagner.org

we're missing

Live environment performance
monitoring S.A.S.

ruby/java/php
have http://newrelic.com
erlang?

how is it going?

- On time so far
- Benchmarks look good
- Stay tuned... launch date approaching!

Thanks!

http://www.wooga.com/jobs

paolo.negri@wooga.com

@hungryblank