Noob to Production

in two months...

BitLove

jamesgolick.com

an blog.

BitLove

github.com/jamesgolick

code.

BitLove

twitter.com/jamesgolick

shit talk

BitLove

bitlove.co

company.

BitLove

fetlife.com

product.

BitLove

fetlife.com

- ~ 375 million pageviews / mo
- ~ 900 web req/s peak
- > 1 million users
- Mandatory, site-wide SSL
- profitable (*gasp*)

BitLove

Noob to Production

in two months...

BitLove

Requirements

- IM chat exposed over web UI, similar to Facebook chat.
- Authentication through our main web app.
- Rosters managed by and pulled from the web app.
- Security: only friends on site can chat with each other.

BitLove

xmpp

ejabberd

BitLove

erlang

http://www.erlang.org/download/getting_started-5.4.pdf

BitLove

```
-module(ejabberd_auth_fetlife).

-export([start/1,
    set_password/3,
    check_password/3,
    check_password/5,
    try_register/3,
    dirty_get_registered_users/0,
    get_vh_registered_users/1,
    get_vh_registered_users/2,
    get_vh_registered_users_number/1,
    get_vh_registered_users_number/2,
    get_password/2,
    get_password_s/2,
    is_user_exists/2,
    remove_user/2,
    remove_user/3,
    plain_password_required/0
  ]).
```

```
-module(ejabberd_auth_fetlife).

-export([start/1,
    set_password/3,
    check_password/3,
    check_password/5,
    try_register/3,
    dirty_get_registered_users/0,
    get_vh_registered_users/1,
    get_vh_registered_users/2,
    get_vh_registered_users_number/1,
    get_vh_registered_users_number/2,
    get_password/2,
    get_password_s/2,
    is_user_exists/2,
    remove_user/2,
    remove_user/3,
    plain_password_required/0
   ]).
```

```
-module(mod_roster_fetlife).

-export([start/2, stop/1,
         process_iq/3,
         process_local_iq/3,
         get_user_roster/2,
         get_subscription_lists/3,
         get_in_pending_subscriptions/3,
         in_subscription/6,
         out_subscription/4,
         set_items/3,
         remove_user/2,
         get_jid_info/4,
         item_to_xml/1,
         webadmin_page/3,
         webadmin_user/4,
         get_versioning_feature/2,
         roster_versioning_enabled/1,
         roster_version/2]).
```

```
-module(mod_filter_fetlife).

filter_packet(drop) ->
  drop;
filter_packet(Packet = {#jid{user=FromUser}, #jid{user=FromUser},_}) ->
  Packet;
filter_packet(Packet = {#jid{user=""}, _, _}) ->
  Packet;
filter_packet(Packet = {#jid{user=[]}, _, _}) ->
  Packet;
filter_packet(Packet = {#jid{user=FromUser},
#jid{lserver=LServer,server=Server,user=ToUser}, _}) ->
  case get_friendship_status(FromUser, ToUser) of
    true ->
      Packet;
    false ->
      drop
  end.
```

IM Bridge

IM Bridge

- Written in erlang.
- Bridge between web and ejabberd.
- Speaks jsonp to web and XMPP protocol to ejabberd.
- Keeps buddy list / conversation state in memory and relays updates to front end.

BitLove

IM Bridge

- Multiplexes XMPP session across multiple browser tabs.
- Erlang XMPP client sorta works ish.
- Prototype deployed to ~20k alpha testers.

BitLove

ejabberd revisited

- 59kLoC of erlang.
- Completely incomprehensible logging.
- ejabberd_c2s.erl has several functions that are more than 2 screenfuls on my Macbook Air.
- Fuck ejabberd.

BitLove

IM Bridge.erl **v2**

- What if we wrote our own presence implementation?
- How hard could it possibly be to write a router?
- UI state persistence.
- Conversation history persistence.

BitLove

IM Bridge.erl **v2**

- Written, deployed, and fully rolled out to 1.5M users in about a month.
- Maxing out around 50k sessions with 200k open tabs.
- ~10k "packets" per second.
- 20 messages per second.
- Running on one machine.

BitLove

Hardware Specs

- Dual E5520s.
- 12GB Memory.
- 2 x Micron 100GB gen3 SSDs in RAID1.
- CentOS 5.4.
- Linux 2.6.39.

BitLove

Optimizations

- Remove JSON calls in favour of speaking directly to the production database.
- Remove local roster cache.
- Gutted the router to increase efficiency.
- ~50 deployments in 3 weeks.
- **0 process restarts.**

BitLove

Major Bug

- Tons of sessions timing out at once.
- Guess: they're all waiting for some resource.
- Deadlock?
- Chat still appears working despite all this because of supervision.

BitLove

Major Bug

- Need more data.
- Catch timeout exceptions from key gen_server:call invocation and log stack traces.
- Lager trace files.

BitLove

```
lager:debug([{user_id, User#user.id}],
            "Something incredibly relevant to your debugging efforts happened!",
            []).
```

```
lager:debug([{user_id, User#user.id}],
            "Something incredibly relevant to your debugging efforts happened!",
            []).
```

```
lager:trace_file("/var/log/myservice/user-1.debug.log",
                 [{user_id, 1}],
                 debug).
```

```
lager:trace_file("/var/log/myservice/user-1.debug.log",
                 [{user_id, 1}],
                 debug).
```

```
lager:trace_file("/var/log/myservice/user-1.debug.log",
                 [{user_id, 1}],
                 debug).
```

```
lager:trace_file("/var/log/myservice/user-1.debug.log",
                 [{user_id, 1}],
                 debug).
```

Major Bug

- Turns out all the dead sessions were waiting on MySQL queries.
- Open MySQL driver sources and immediately puke all over myself.
- Looks like the MySQL driver's pool only maxes out at one connection unless you specifically issue mysql:connect/**8**.

BitLove

Major Bug

- Fork of MySQL driver that makes it more OTP-ish @ jamesgolick/erlang-mysql-driver.
- Uses poolboy for connection pooling.
- Bug fixed.

BitLove

Conclusions

- Code reloading and supervision actually work.
- BEAM actually does make operations easier.
- But that doesn't mean that all erlang software is easy to operate.
- Erlang OSS world is a bit of a mess.
- BitLove is standardizing on Erlang.

BitLove

The End.