

Erlang port on FreeBSD

or

how Erlang runs similarly or differently on FreeBSD
from Linux and people don't care much about it

Kenji Rikitake (力武 健次)

NSIRG, NICT, Japan

第3回Erlang分散システム勉強会

3-JUL-2009

Twitter: @jj1bdx (ja_JP) / @kenji_rikita (en_US)

Before we begin...

Slides are written in English

- でも発表は日本語で行います

Questions are welcome at any time

- 質問は発表中でも遠慮なくどうぞ

I'm new to Erlang, so what I show
you might be wrong (let me know)

- 内容が間違っていたら教えてくださいね

How I got the Erlang bug

Have you read the TWO books?

Joe Armstrong's **Programming Erlang**

Francesco Cesarini's **Erlang Programming**

My speciality is on Internetworking

- Erlang looks promising as a testing tool

I like Erlang as a restrictive language

- The language itself assists the writers of the code to think concurrently

What this presentation is about

Porting Erlang on FreeBSD

- FreeBSD-specific issues and non-issues

How Erlang works similarly (and differently) between FreeBSD and Ubuntu

- Benchmarking with ejabberd and tsung

Future plans and works

- wishing Erlang to be less Linux-centric

Why FreeBSD for Erlang?

Good SMP performance since 7.x

- Steady improvement by removing the **giant locks** from the kernel

Safely operating services in Jail

- Generic chroot'ing = virtualization
- Separate userland/address for each Jail

... and **BSD IS THE UNIX**. Period.

- 4.3BSD, BSDI, and FreeBSD since 1997

FreeBSD ports and packages

Source code and binaries integrated

- pre-built binaries = packages
- source code sets for packages = ports

Dependencies between programs are resolved in the ports distribution

- No need to worry about compatibility

You can modify and learn from ports

- and rebuild modified code on your own

Erlang port status on FreeBSD

Master port: **lang/erlang**

- Maintained by Giacomo "Jimmy" Olgeni
- R13B01 is available now

Other related ports

- lang/erlang-lite: w/o Java, X11, ODBC
- Available applications and tools
couchdb, distel, ejabberd, rabbitmq, wings,
yaws / esdl, mysql, py_otp_interface, xmlrpc

What does the Erlang port do?

Solving directory structure difference

- FreeBSD-specific structure applied
for the online manuals and libraries

Registration for (un-)installation

- deinstall-ation erases all the old files

Various patches to Erlang code

- Many applicable to master source tree
- and a few FreeBSD-specific patches

FreeBSD directory structure

All 3rd-party things under **/usr/local**

- Largely different from Linux
- Erlang to: **/usr/local/lib/erlang/**

Caution: port maintainer may specify completely different tree structure

- Read the Makefile carefully

For some Jails, **/usr is not writable**

- In my case, only **/usr/local** is writable

(Past) glitches on FreeBSD

TAI (leap second) issue

- fixed in R13A (included my patch)
- calendar module cannot handle TAI yet

R12B5 killed the whole OS

- reproducible; fixed in R13A

HiPE doesn't work well on gcc 4.x

- stock gcc: 4.2.1 (on 7.x) shows this
- gcc-3.4.6 compiles the code OK

Benchmarking method with ejabberd

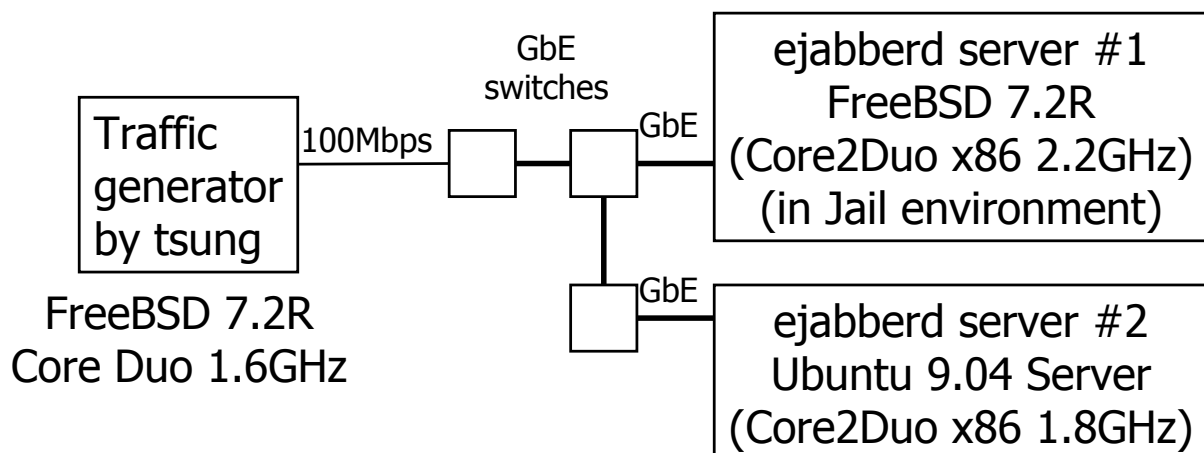
Stress-testing ejabberd

- on FreeBSD 7.2R and Ubuntu 9.04
- network parameter tuning performed
e.g. maxconns > 30000, large mbuf mem

Test traffic from tsung

- running on FreeBSD 7.2R
- Jabber in-band register (XEP-0077)
- 100~200 new users/sec

Test equipments and network



Generated test traffic from tsung and took the statistics of the response

ejabberd installation (on FreeBSD)

Installed from the latest trunk (2.10a)

- Configured with `-prefix=/var/ejabberd`

exmpp (0.91a) also needed

- erlang.m4 in autoconf-2.62 patched

- installed under Erlang lib directory

No big difference from Ubuntu, but...

- used **gmake**, not BSD make (= pmake)
(Erlang is GNU-make centric)

tsung is installed as well

Installed from the trunk (1.3.1a)

- Perl Template toolkit needed

<http://www.template-toolkit.org/>

- gnuplot needed for drawing stats

Tsung code is installed under Erlang lib directory as default

- test examples (HTTP, Jabber, LDAP)

- stat drawing tools in Perl and Python

ejabberd benchmarking results

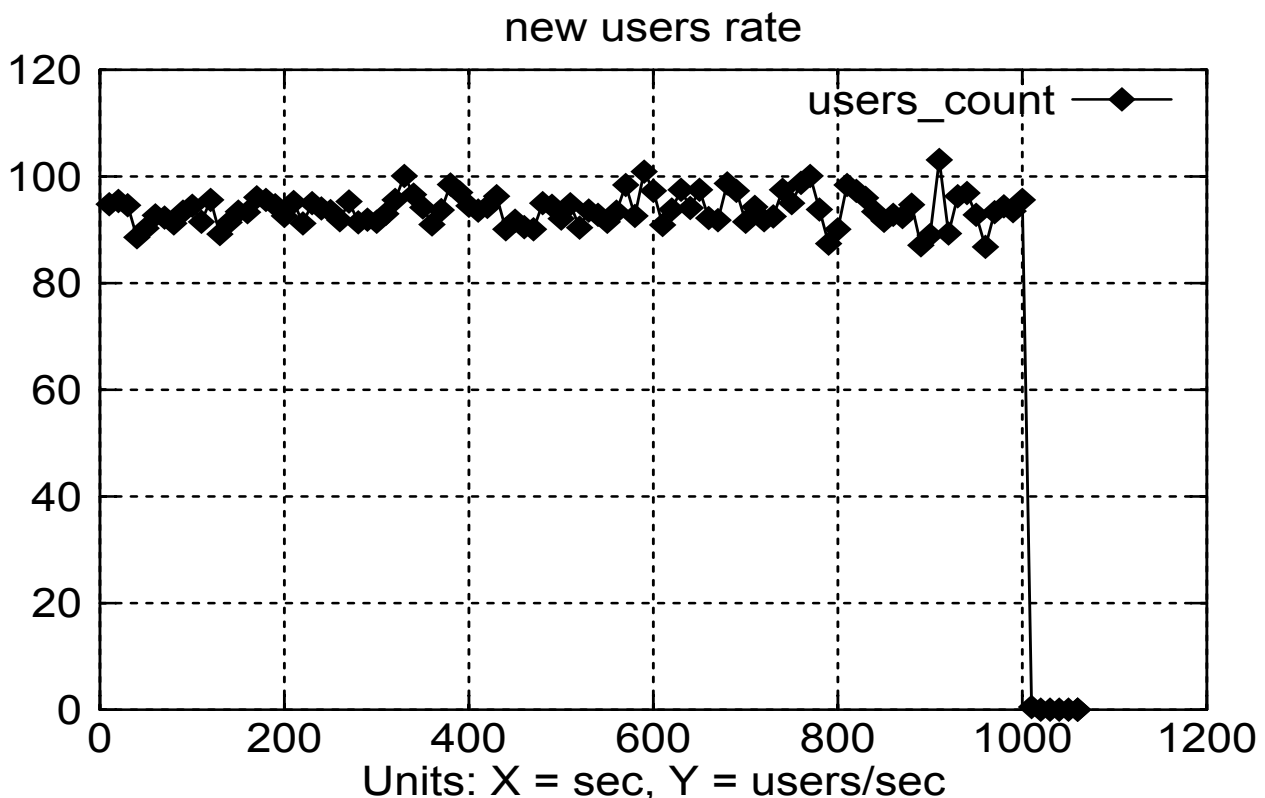
FreeBSD and Ubuntu behaved almost the same in manageable workload

- well-handled 100 new users/s

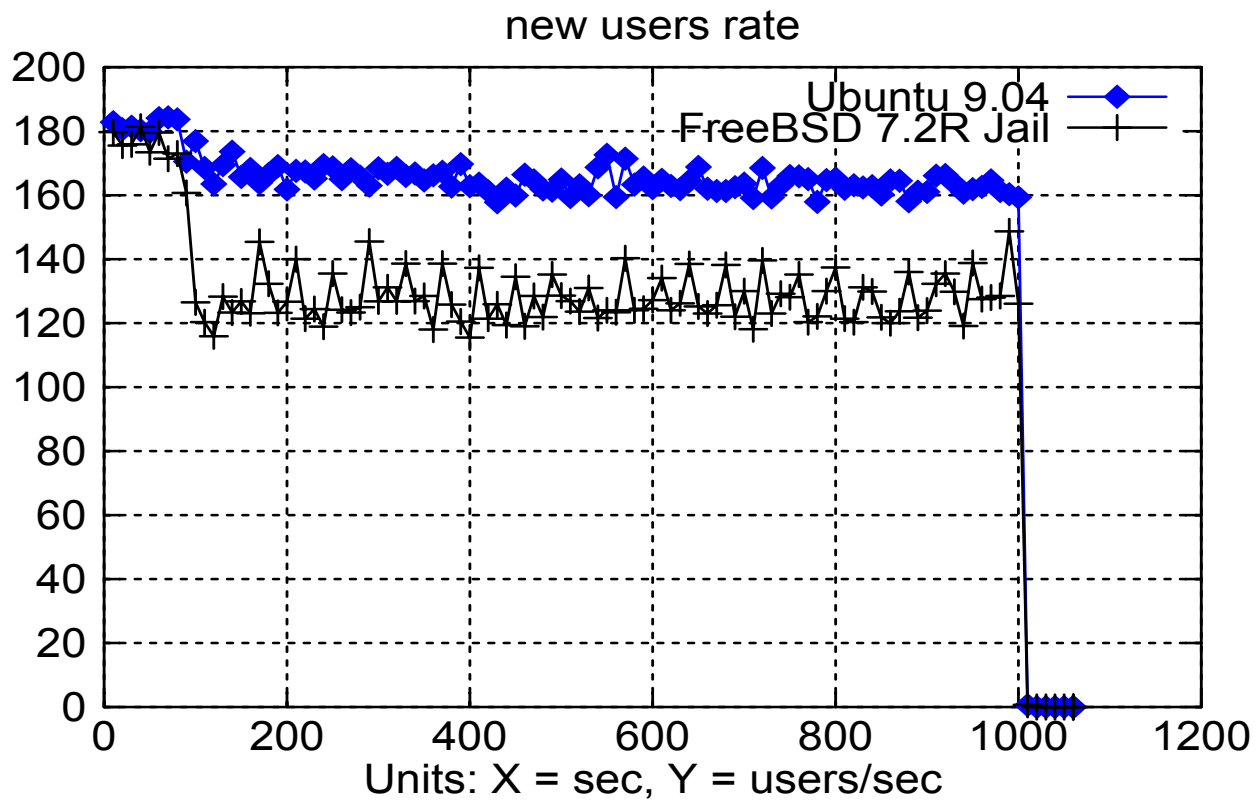
Ubuntu was faster than FreeBSD, when heavily stressed

- 200 users/s: Ubuntu ~165, FBSD ~130
- even in slower CPU clock freq
- Running in Jail may affect the results

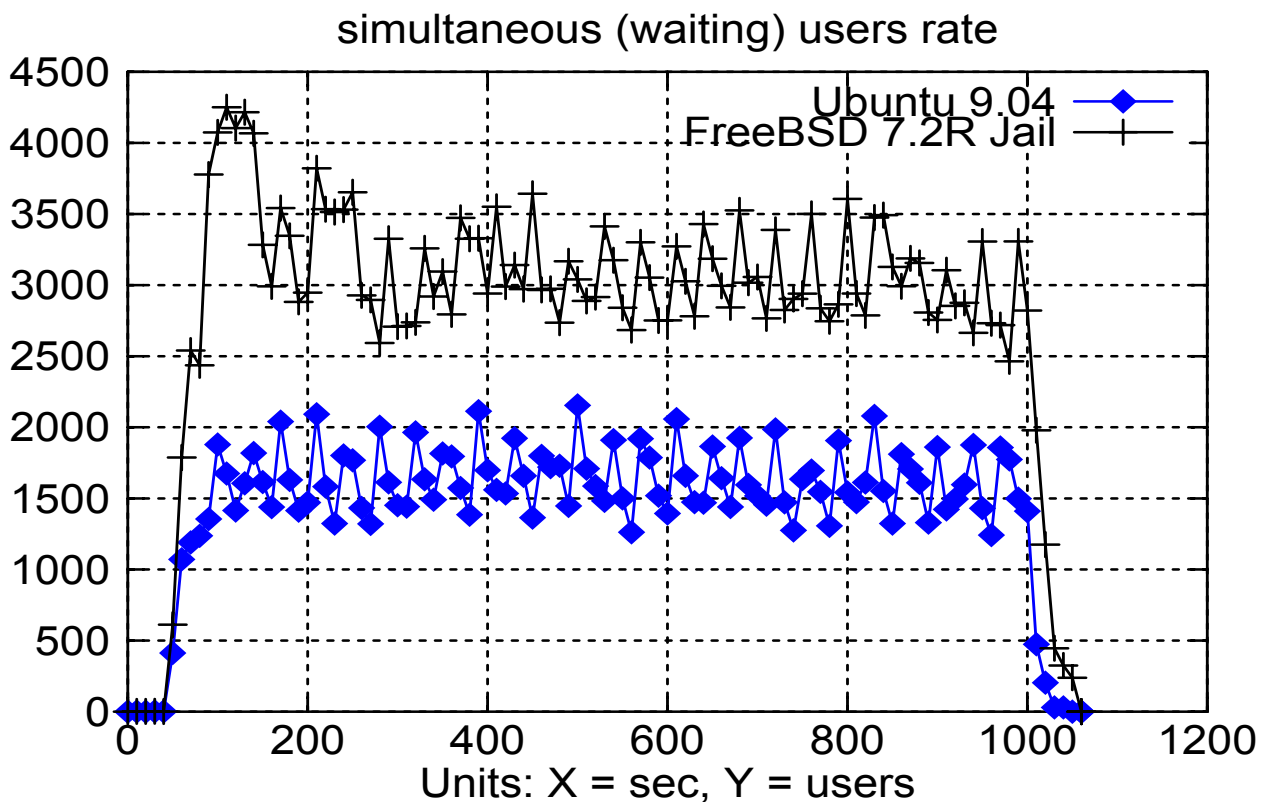
100 new users/s, 30000 users



200 new users/s, 200000 users



200 new users/s: waiting users



Future plans and works

Detailed benchmarking and analysis

- which is the slower module in FreeBSD?
TCP/IP code? DNS resolver? Mnesia?

Testing newer protocols and features

- IPv6, SCTP, SSL, SSH
- Secure RPC compatible with rpc module
- Random number drivers (e.g., SFMT)
- ... and making Erlang less Linux-centric

Thanks for listening!

Any questions?