News From the OTP TEAM

Kenneth Lundin, Raimo Niskanen, Erlang/OTP, Ericsson
Code Beam Conference, San Francisco, March 2018

Outline

- Planned Releases
- Short about 20.3
- Preliminary about 21.0
- Longer term

Planned Releases

- 1 major release per year (20, 21, …)
- 3 planned “correction packages” like 20.1, 20.2,...
- Emergency packages as needed (20.1.3)
- Maint branch = 20.X
- Master branch = next major (21.0)

OTP 20.3 March 14

OTP 21.0-rc1 May 02

OTP 21.0-rc2 May 30

OTP 21.0 June 20

“Next” minor (maint branch)

- httpc requests over Unix Domain Sockets?
- More crash dump info such as: process binary virtual heap stats, full info for process causing out-of-mem during GC, more port related info, and dirty scheduler info.
- SSH: default exec is disabled when a user-defined shell is enabled.
- 18 number of (external) contributions since 20.2

20.3

[March 14, 2018]

Next major
(master branch)

Erlang VM

- Enhanced IO scalability (and less latency)
- Support for usage of **distribution controller processes** (previously only ports). For impl. of alternative transports, routing etc.
- Compact instructions on 64 bit systems that load program code below 4 Gbyte. **20% less memory for loaded code!**

21

[June, 2018]

- ☐ OTP 21.0-rc1 May 02
- ☐ OTP 21.0-rc2 May 30

Next major
(master branch)

Erlang VM ...

- Rewrite of the efile-driver with NIFs and “dirty schedulers” (**= faster file operations**)
- It is now possible to open device files and FIFOs with file:open/2.
- `ets:whereis(TableName) -> TableRef`
- non-smp VM removed
- `link` and `monitor` optimized for scalability (used in `gen_server:call`)

21

[June, 2018]

- OTP 21.0-rc1 May 02
- OTP 21.0-rc2 May 30

Next major
(master branch)

Compiler

- code such as
  ```
  f({ok, Val}) ->
    {ok, Val}.
  ```
  is now automatically rewritten to
  ```
  f({ok,Val} = Tuple) ->
    Tuple.
  ```
  reduces code size, execution time, and removes GC pressure.
- Misc compiler optimizations including contributions from the Elixir team
  (**10%** performance improvement in benchmarks)

21

[June, 2018]

- [ ] OTP 21.0-rc1 May
- [ ] OTP 21.0-rc2 May

Next major
(master branch)

Compiler ...

- Arguments and operator now part of the stacktrace when an exception is thrown from the operators `band, bor, bsl, bsr, bxor, div, rem, +, -, * , /.`
- New syntax:
  `try … catch`
  `  C:R:StackTrace -> …`
  Now deprecated:
  `erlang:get_stacktrace/0`
- Creation of small maps with literal keys has been optimized; faster and less memory

21

[June, 2018]

- ☐ OTP 21.0-rc1 May
- ☐ OTP 21.0-rc2 May

Next major
(master branch)

Security
(ssl, ssh, crypto, public_key)

- DTLS supported (part of SSL application)
- Enhanced support for distribution over TLS.
- Remove “unsecure” chipers from defaults in SSL and SSH, e.g. RSA key exchange

21

[June, 2018]

- ☐ OTP 21.0-rc1 May
- ☐ OTP 21.0-rc2 May

Next major
(master branch)

Standard libraries (stdlib, kernel)

- `logger` NEW API for logging
- New `uri_string` module which parses URIs according to “the standards”
- New function

```
lists:search(List,fun/1)
  -> {ok,Value} | false
```

21

[June, 2018]

- ☐ OTP 21.0-rc1 May
- ☐ OTP 21.0-rc2 May

OTP 21 changes in repos and applications

- Move out to separate repo called Orber
  - `Orber, CosTime, CosNotification, …` *8 apps*
- Move out to separate repo
  - `otp_mibs`
- Split **`inets`** into separate applications
  - move out `ftp, tftp`
  - keep `httpc,httpd` in `inets` for now
  - remove dependencies towards `inets` as far as possible
  - new `uri_string` in `stdlib` is one of the steps

Longer term

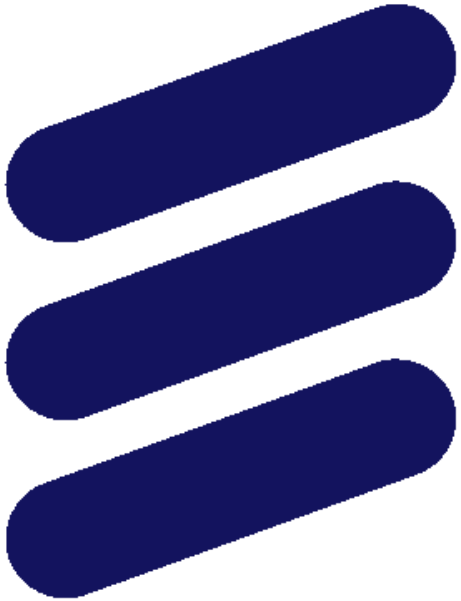
- More compiler optimizations using SSA (Single Static Assignment) representation
- Rewrite inet-driver (TCP/UDP/SCTP) as NIFs using `enif_select`
- Profiling tools improvements `cprof/eprof/fprof ...`

Longer term, continued

- Heterogenous distribution (TLS TCP)
- Replace Global?
- Scalable up to +100 nodes?
- Super efficient counters
- JIT
- Erlang Language Server (Language Server Protocol)
- TLS 1.3

Anniversary Erlang 20 years as Open Source 2018

	R6B 1999	20.2 2018
Compressed tar with source	6.7 Mbytes	83 Mbytes
Erlang code (lines)	213571	1756751
C code (lines)	199655	322227
Applications	33	45
Modules	738	1831



ERICSSON