



WHAT'S NEW AND WHAT'S COOKING IN OTP

SERVICE RELEASE R15B01 4:TH APRIL

MAJOR RELEASE R16B

SERVICE RELEASE R15B01

- › Mainly bug fixes
- › 146 distinct fixes
 - 25 directly from Open Source users
 - A lot of fixes are adopted Open Source patches
- › A slight dip in number of open source patches
 - Less than half the number of Open Source patches as in R14B04
 - › Some patches now go into master/master-pu branch
 - › “Natural” fluctuations
 - › More adoptions than in R14B04
- › The VM team is changing ways of working
 - Will result in better flow in patch reviews eventually
 - May have caused the opposite during the past months

SERVICE RELEASE R15B01 HIGHLIGHTS

- › DTrace build option adopted and integrated
- › Clean (flushing) termination of escripts (erlang:halt/1 updated)
- › Common test - a lot of enhancements (25 integrated branches)
- › Reltool - Large rewrite and bugfixes, specific enhancements targeted at rebar (work in progress)
- › Observer - new monitoring tool
- › New application eldap (!)
- › Ssh rewritten to use public_key, solves a lot of problems
- › Ssl stabilized and enhanced regarding SSL/TLS distribution
- › Parallell make enhanced, works for all -j variants...
- › erlang:statistics(scheduler_wall_time) to accurately measure system load

SERVICE RELEASE R15B01 CONTINUED

› DTrace (Thanks to Scott Lystig Fritchie):

- Configure VM with `--with-dynamic-trace=dtrace`
- Inserts probes for `efile_drv`, messages, function calls, gc etc
- Minimal impact on performance when no probes are enabled
- Example D scripts in `runtime_tools/examples`
- More probes to come, may even be default on OsX and others in the future
- Not supported (yet), use for debugging, not production
- Read `$ERL_TOP/README.dtrace(.md)` and the documentation on “dyntrace” in “runtime_tools”

› Systemtap

- Configure VM with `--with-dynamic-trace=systemtap`
- Works mostly like dtrace
- Examples in same directory as for DTrace
- Still very much a work in progress, patches are welcome

WHAT THE FUTURE PROBABLY WILL HOLD... (R16)

- › Code loading in parallel (also tracing manipulation without stopping all schedulers)
- › General enhancements to the many core case
 - A lot more lock-free data structures
 - Enhancements to scheduling
 - Lower latency in network communication
- › Type-specs and documentation more tightly integrated
 - Regular type specs for built in functions
 - BIF's will appear in modules with NIF-like stubs
- › Faster dialyzer
 - Both in single and multicore case
- › Cross-compile support for testsuites

WHAT THE FUTURE PROBABLY WILL HOLD... (R16) CONTINUED



- › More dynamic tracing
 - More probes
 - LTTng UST support
 - Default on some platforms
- › More enhancements to reltool
- › Expanded eldap application
- › SSL
 - Certificate Revocation List Verification
 - Next protocol negotiation (User contribution)
 - TLS 1.1 and 1.2 (user contributed prototype)
 - Distribution configuration enhancements
- › Performance enhancements in protocol implementations
 - Joint effort between VM and application developers

WHAT THE FUTURE MIGHT HOLD

- › NUMA enhancements
 - Researched at the moment
- › JIT compiler
 - Researched in cooperation with SICS
 - Do listen to Lukas later today if you are interested in details
- › Enhancements to message passing
 - Sharing preservation, internal/external messages
 - RELEASE spinoff
- › General string module
- › New built in data structures
 - Replace records
 - Replace 'dict'

WHAT THE FUTURE MIGHT HOLD CONTINUED

- › More many-core enhancements
 - More RELEASE related/spinoffs
- › Dirty schedulers
 - Replace async for non CPU-bound jobs
 - Handle CPU intensive non Erlang jobs
- › Native processes, the next step from NIF's
- › Enhanced I/O
- › SCTP enhancements
- › Continued Unicode work
 - Source code in UTF-8
 - Atoms
 - Scanner/Parser
- › Many, many more high quality user contributions!
- › ...



ERICSSON