

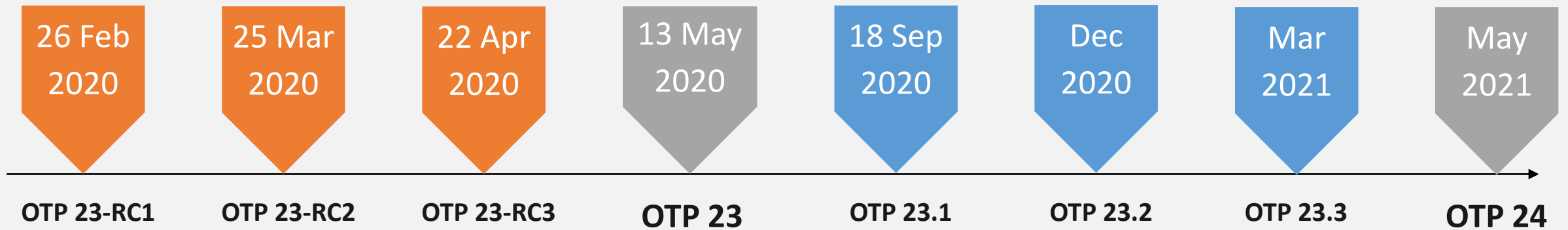
OTP TEAM UPDATE

Péter Dimitrov
Erlang/OTP

Agenda

- Planned Releases
- OTP 23 Highlights
- Long term

Planned Releases



OTP 23

Distributed spawn

ERTS

Distributed spawn

- Scalability and performance improvements
- A distributed `spawn_monitor()` BIF.
`{Pid, MRef} = spawn_monitor(fun client/0)`
- Monitor option in the distributed `spawn_opt()` BIF.
- New `spawn_request()` BIFs for asynchronous spawn of processes.

Distributed spawn cont'd

```
1> spawn_monitor(fun () -> ok end).
```

```
{<0.90.0>, #Ref<0.3028519357.3081764867.128771>}
```

```
2> flush().
```

```
Shell got {'DOWN', #Ref<0.3028519357.3081764867.128771>, process, <0.90.0>,  
          normal}
```

```
ok
```

```
3> spawn_request(fun () -> ok end).
```

```
#Ref<0.3028519357.3081764867.128750>
```

```
4> flush().
```

```
Shell got {spawn_reply, #Ref<0.3028519357.3081764867.128750>, ok, <0.87.0>}
```

```
ok
```

```
5>
```

ETS – decentralized_counters

ERTS, stdlib

ETS – decentralized_counters

- Improved scalability of concurrent calls (all ETS table types):
`ets:insert/2`, `ets:delete/2`
- This may come at the cost of longer latency:
`ets:info(T,size)`, `ets:info(T,memory)`.
- New table option `decentralized_counters` has been added
 - Default **true: ordered_set** with **write_concurrency** enabled
 - False otherwise

New arity-1 functions with utf8

ERTS

New arity-1 functions with utf8

atom_to_binary/1

binary_to_atom /1

binary_to_existing_atom/1

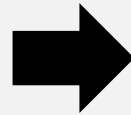
Matching syntax improvements

ERTS, compiler

Matching syntax improvements

```
example1(<<Size:8, Payload: ((Size-1)*8)/binary, Rest/binary>>) ->  
  {Payload, Rest}.
```

```
example2(M, X) ->  
  Key = {tag, X},  
  #{Key := Value} = M,  
  Value.
```



```
example2(M, X) ->  
  #{{tag, X} := Value} = M,  
  Value.
```

```
illegal_example(Key, #{Key := Value}) -> Value.
```

Numeric literals with underscore

compiler

Numeric literals with underscore

1> 123_456_789.

123456789

2> 16#1234_ABCD.

305441741

Distributed Named Process Groups kernel

Distributed Named Process Groups

- Module: pg (pg2 is deprecated)
- A new implementation of distributed named process groups
- Process Groups implement strong eventual consistency
- Process groups can be organised into multiple scopes.
- Scopes are designed to decouple the single mesh into a set of overlay networks.

New erpc module

kernel

New erpc module

- Implements an enhanced subset of the operations provided by the rpc module.
- Now possible to distinguish between returned value, raised exceptions, and other errors.
- Better performance and scalability by utilizing the newly introduced `spawn_request()` BIF.

Help in the shell

kernel

Help in the shell

- EEP 48: Documentation storage and format
- Doc build generates .chunk files for all OTP
- On-line help in the shell
h(Module), **h(Module,Function)**,
h(Module,Function,Arity)
- New module **shell_docs** in `stdlib` with functions
render/{1,2,3}
- New function **code:get_doc(Module)** which returns the doc chunk without loading the module.

Help in the shell cont'd

```
5> h(lists,sort,2).
```

```
-spec sort(Fun, List1) -> List2
```

```
  when
```

```
    Fun :: fun((A :: T, B :: T) -> boolean()),
```

```
    List1 :: [T],
```

```
    List2 :: [T],
```

```
    T :: term().
```

Returns a list containing the sorted elements of List1, according to the

ordering function Fun. Fun(A, B) is to return true if A compares less than or equal to B in the ordering, otherwise false.

Container friendly features

kernel

Container friendly features

- Improved handshake in the Erlang distribution protocol (makes it independent on EPMD)
- CPU quotas are now taken into account when deciding the default number of online schedulers: e.g. docker with the `--cpus` flag.
- New option: `erl_call -address [Host]:Port`
- ...

Build with WSL on Windows

Build with WSL on Windows

- The Windows version is now built with WSL
- Easier to maintain the build “scripts”

SSH/SSL/crypto

SSH/SSL/crypto

- **SSH:**
 - TCP/IP port forwarding aka. tunneling
- **SSL:**
 - TLS 1.3 client (22.1, 23)
 - Session tickets: stateful and stateless (22.3, 23)
 - Key update mechanism (22.3, 23)
 - TLS 1.3 is configured by default* (23)
 - SSLv3 removed!
- **crypto:**
 - New crypto API

Long term

Long term

- Taking active role in EEF workgroups
- Cloud, Container and micro-service friendly
- Replacing inet driver by using socket NIFs
- gen_udp, gen_sctp APIs based on socket NIF
- IDE/Editor support via Language Server (LSP)
- QUIC-TLS, QUIC

