Counters

Maxim Kharchenko, Cloudozer LLP

10/02/2014

1 Overview

The document describes an extension to the Erlang runtime - counters. Counters are variables that hold a single usigned integer value. The variable can be updated (incremented) and read by any process that knows its reference.

2 Interface

The counters interface resembles that of timers:

```
erlang:new_counter(Bits) -> Ref
erlang:new_counter() -> Ref
```

The call allocates a new counter and returns its reference. The initial value of the counter is zero. ***Bits*** is the number of bits in the counter value. When the counter overflows it goes back to zero. `erlang:new_counter()` is equivalent to `erlang:new_counter(64)`.

```
erlang:read_counter(Ref) -> integer() >= 0 | false
```

The function retrieves the current value of the counter. It returns false if the counter was released or Ref was never a counter.

```
erlang:update_counter(Ref, Incr) -> true
erlang:update_counter(Ref) -> true
```

The call adds ***Incr*** to the counter value. `erlang:update_counter(Ref)` is equivalent to `erlang:update_counter(Ref, 1)`. A badarg exception is raised if Ref is not a counter.

```
erlang:release_counter(Ref) -> true
```

The function destroys the counter. If the counter does not exist, a badarg exception is raised.