[erlang-questions] Pid after node restart

Fred Hebert <mononcqc@ferd.ca>
Tue Sep 30 17:58:05 CEST 2014

- Previous message: [erlang-questions] Pid after node restart
- Next message: [erlang-questions] [ANN] Rockse: A Facebook Rocksdb Erlang binding
- **Messages sorted by:** [date] [thread] [subject] [author]

On 09/29, Pavel Baturko wrote:
> *My case is:*
> *I have 2 connected nodes.*
>
> *[snip]*
>
> *Could someone explain this behavior?*
> *What is changing in internal pid representation after node restart so*
> *comparison in step 10 returns false? And why comparison in step 11 returns*
> *true?*

The visual pid representation you see isn't the full thing.

Interesting for this is to look at the external term format:
http://erlang.org/doc/apps/erts/erl_ext_dist.html#id88514

```
    1       N       4   4       1
    103     Node    ID  Serial  Creation

    Encode a process identifier object (obtained from spawn/3 or friends).
    The ID and Creation fields works just like in REFERENCE_EXT, while the
    Serial field is used to improve safety. In ID, only 15 bits are
    significant; the rest should be 0.

    [..]

    Creation is a byte containing a node serial number that makes it
    possible to separate old (crashed) nodes from a new one.
```

What this tells us is that each pid you send around contains the full
node name (internally you may have a hash, which is X in <X.Y.Z>).
Locally, 'X' is always printed as '0' to be easy to recognize.

Y and Z are then two counters you have in there, and hidden in it is the

'Creation'

Looking at your list:

> *1. On node 1 I generate a pid P (<0.X.0>)*
> *2. P is transfered to node 2 (on node 2 I receive it as <Y.X.0>)*
> *3. P is transfered back from node 2 to node 1 (I'm receiving it as P2)*
> *4. P2 is recognized as local pid (<0.X.0>)*

This is covered by the node ID and local/remote printing of pids

> *5. I can compare P and P2 on node 1 and they are identical*

Covered by internal pid structure.

> *6. I'm restarting node 1*
> *7. On node 1 I generate the same pid P as in step 1 (<0.X.0>)*

The Creation id is likely set propelry there.

> *8. I repeat step 3, again receive initial pid as P2*
> *9. P2 is recognized as local pid (<0.X.0>) - as before*
> *10. But now P == P2 returns false.*
> *11. Also P == list_to_pid(pid_to_list(P2)) returns true*

The creation byte is likely what's at play there. You can
term_to_binary(P) and term_to_binary(P2) to confirm the difference in
there.

list_to_pid does some fancy pants evaluation, but term_to_binary and
binary_to_term will do the full form lossless serialization.

Regards,
Fred.

- Previous message: [erlang-questions] Pid after node restart
- Next message: [erlang-questions] [ANN] Rockse: A Facebook Rocksdb Erlang binding
- **Messages sorted by:** [date] [thread] [subject] [author]

More information about the erlang-questions mailing list