



Erlang Solutions Ltd.

Resolving Mnesia Netsplits (in 10 minutes)

Unsplit-Brain

Ulf Wiger

Erlang Solutions Ltd

Erlang Factory, San Francisco, 25 Mar 2010

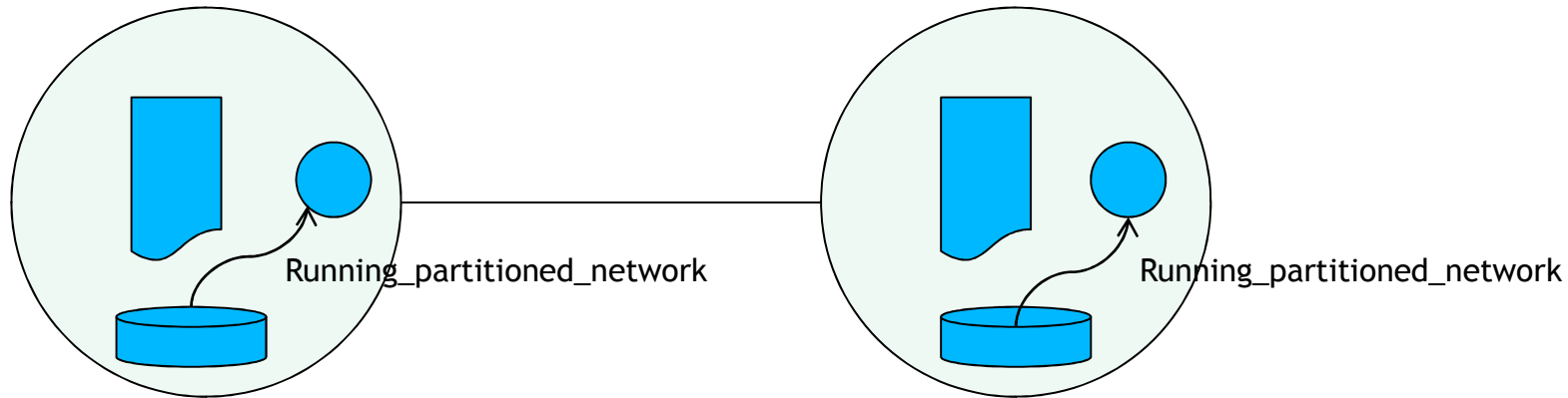
Initial Problem

- **Mnesia is a wonderful DBMS**
 - Tight integration with the language
 - Fast in-memory tables
 - Robust, battle-tested persistent storage
- **It has some weaknesses**
- **Our focus: Recovery from partitioned networks**

Split-brain

- Network failure is indistinguishable from normal nodedowns
- When nodes are reconnected, database can be inconsistent
- Pathological problem in general
- Mnesia detects the condition
 - Issues a “running partitioned network” event
 - Refuses to merge the tables
- Only remedy offered:
 - Call `mnesia:set_master_nodes([N])` on one side
 - Unconditionally load data from N
 - Data loss is very likely

The 'unsplit' Application



- Install an event handler on each node (automatic)
- When triggered, grab a global lock (`global:trans/2`)
 - The one who wins, resolves the conflict
- Merge the schema, lock tables, and merge in one operation
 - Requires a mnesia patch (will submit soon)

How to merge

```
mnesia:create_table(test, [{ram_copies, [n1@debian, n2@debian]},  
                           {attributes, record_info(fields, test)},  
                           {user_properties,  
                            [{unsplit_method, {unsplit_lib, vclock,  
                                                [#test.vclock]}}]}  
                           ]}).
```

- Vector clock implementation borrowed from Riak
- Other methods possible
 - Predefined methods: last_modified, bag, ...
- The unsplit_reporter module can be used to report inconsistencies
 - Sends “summary alarm” to alarm_handler in SASL
 - Collects conflicting records in a temp table for inspection

Automatic updating of Vector Clocks

- `mnesia:activity(transaction, Fun, my_mnesia_cb)`
- Make a hook function for `write(Tid, Ts, Tab, Rec, LockKind)`
- Suggestion: `exprecs` for generic record attribute access:

```
-module(my_mnesia_cb).  
-include("table_defs.hrl").  
-export_records([....]).  
  
write(Tid, Ts, Tab, Rec, LockKind) ->  
    Rec1 = try Old = '#get-'(Rec, [vclock]),  
            '#set-'(Rec, [{vclock, unsplit_vclock:increment(node(), Old)}])  
        catch  
            error:badarg ->  
                Rec  
        end,  
    mnesia:write(Tid, Ts, Tab, Rec1, LockKind).
```

That's it! 😊

- <http://github.com/uwiger/unsplit>
- http://github.com/uwiger/parse_trans (for exprecs)
- http://github.com/uwiger/mnesia_merge (the mnesia patch)
- Possibly vie for inclusion into OTP
- NOTE! Problem is still very hard
- You need to plan your data model
- Prepare for inconsistencies
- Split happens - this might at least give you a chance to cope