

1. 如何动态的添加一个节点到 Mnesia cluster 中?

用到的 Mnesia APIs:

`mnesia:info()`

`mnesia:create_schema(DiscNodes).`

(参数是一个节点列表, 很多文件会在每一个节点的 mnesia directory 中创建, 并且每个 node 的 directory 必须唯一)

`mnesia:start()`

`mnesia:create_table(Name, TabDef).`

`mnesia:change_config(extra_db_nodes, NodeList)`

(参数是新的 node 的节点, 如果成功, 返回{ok, ResNodeList}, 其中 ResNodeList 是已经加到 mnesia cluster 中的节点)

`mnesia:change_table_copy_type(Table, Node, Type)`

(这个函数也可以用于 schema 表, schema 表的 type 只能是 ram_copies 或者 disc_copies, 如果 schema 的类型是 ram_copies, 这个节点上的别的表都不能存储在磁盘上)

`mnesia:add_table_copy(Tab, Node, Type)`

<1> 启动两个 Erlang 节点:

```
werl.exe -setcookie testcookie -sname node1 -mnesia dir "c:/home/mnesia/node1"
```

```
werl.exe -setcookie testcookie -sname node2 -mnesia dir "c:/home/mnesia/node2"
```

<2> 在 node1 上创建 schema, 启动 mnesia, 创建两个测试表格: user1, user2

`mnesia:create_schema([node()]).`

`mnesia:start().`

`mnesia:create_table(user1, [{disc_copies, [node()]}]).`

`mnesia:create_table(user2, [{disc_copies, [node()]}]).`

在 node1 上调用 `mnesia:info()` 查看信息

```
running db nodes = ['node1@liqiang-tfs']
```

```
stopped db nodes = []
```

```
master node tables = []
```

```
remote = []
```

```
ram_copies = []
```

```
disc_copies = [schema,user1,user2] %%本地的三张磁盘表
```

```
disc_only_copies = []
```

<2> 在 node2 上启动 mnesia

`mnesia:start()`

在 node2 上调用 `mnesia:info()` 查看信息

```
running db nodes = ['node2@liqiang-tfs']
```

```
stopped db nodes = []
```

```
master node tables = []
```

```
remote = []
```

```
ram_copies = [schema] %% 注意这个在 ram 中的 schema
```

```
disc_copies = []
```

```
disc_only_copies = []
```

<3> 在 node1 上调用

```
mnesia:change_config(extra_db_nodes, ['node2@liqiang-tfs']).
```

```
{ok,['node2@liqiang-tfs']}
```

在 node2 上调用 mnesia:info()查看信息

```
running db nodes = ['node1@liqiang-tfs','node2@liqiang-tfs'] %% 已经连接到 mnesia cluster
```

```
stopped db nodes = []
```

```
master node tables = []
```

```
remote = [user1,user2] %% 两张远程的表格
```

```
ram_copies = [schema] %% 本地 ram 的 schema 表
```

```
disc_copies = []
```

```
disc_only_copies = []
```

注意:

这个操作会让 node1 尝试连接 node2 到 mnesia cluster.参数是尝试连接的 node 的列表, 结果是已经连接上的 node 列表.

这个操作完成后 node2 已经连接到了 mnesia cluster, 但是只有一个 schema 表的 ram copy.(可以查看 node2 的存放路径, 里面没有任何数据内容)

<4> 让 node2 具备存储磁盘表的能力, 在 node2 上运行:

```
mnesia:change_table_copy_type(schema, node(), disc_copies).
```

在 node2 上调用 mnesia:info()查看信息

```
running db nodes = ['node1@liqiang-tfs','node2@liqiang-tfs']
```

```
stopped db nodes = []
```

```
master node tables = []
```

```
remote = [user1,user2]
```

```
ram_copies = []
```

```
disc_copies = [schema] %% 本地 disc 的 schema 表
```

```
disc_only_copies = []
```

此刻 node2 上只包含了一个在磁盘上存储的 schema 表, 没有其它任何内容.(可以查看 node2 的存放路径, 里面包含了 schema 表的信息)

<5> 尝试把 node1 上的表格(user1, user2)以及内容复制到 node2 上, 在 node2 上运行:

```
mnesia:add_table_copy(user1, node(), disc_copies).
```

```
mnesia:add_table_copy(user2, node(), disc_copies).
```

在 node2 上调用 mnesia:info()查看信息

```
running db nodes = ['node1@liqiang-tfs','node2@liqiang-tfs']
```

```
stopped db nodes = []
```

```
master node tables = []
```

```
remote = []
```

```
ram_copies = []
```

```
disc_copies = [schema,user1,user2] %% user1 和 user2 都复制到了 node2 上
```

```
disc_only_copies = []
```

此刻 node2 在 mnesia cluster 中是 node1 的一个备份了, 即使 node1 失效,

原先 node1 上所有的数据都可以在 node2 上读取.

2. 如何动态的从 mnesia cluster 中删除一个节点?

用到的 Mnesia APIs:

mnesia:info()

mnesia:del_table_copy(Tab, Node)

(这个函数在 Node 上删除 Tab 表格的备份, 如果这个表格的最后一个备份被删除, 这个表也就被删除了, 这个函数还可以用来删除 schema, 如果删除 schema, 这个 node 将在 mnesia cluster 中被移除, 调用之前需要在这个 node 上停掉 mnesia)

mnesia:stop()

mnesia:delete_schema(DiscNodes)

(彻底的在这些 node 上删除 mnesia 的数据)

如果一个集群运行在三个节点上: node1, node2, node3, 这三个节点上都有 user1 和 user2 表格的 disc_copies 备份:

<1> 在 node3 上运行 mnesia:info()查看信息.

```
running db nodes = ['node1@liqiang-tfs','node2@liqiang-tfs','node3@liqiang-tfs'] %% mnesia cluster
```

```
stopped db nodes = []
```

```
master node tables = []
```

```
remote = []
```

```
ram_copies = []
```

```
disc_copies = [schema,user1,user2]
```

```
disc_only_copies = []
```

```
[{'node1@liqiang-tfs',disc_copies},
```

```
{ 'node2@liqiang-tfs',disc_copies},
```

```
{ 'node3@liqiang-tfs',disc_copies}] = [schema,user1,user2]
```

<2> 停止 node2 节点, 在 node2 节点上运行:

```
mnesia:stop()
```

<3> 在 node3 上运行:

```
mnesia:del_table_copy(schema, 'node2@liqiang-tfs').
```

(注意: 在删除某个节点上的 schema 表的时候, 该节点上的 mnesia 必须停止, 否则出错)

<4> 在 node3 上调用 mnesia:info()查看信息:

```
running db nodes = ['node1@liqiang-tfs','node3@liqiang-tfs'] %% node2 已经从 mnesia cluster 中移除
```

```
stopped db nodes = []
```

```
master node tables = []
```

```
remote = []
```

```
ram_copies = []
```

```
disc_copies = [schema,user1,user2]
```

```
disc_only_copies = []
```

```
[{'node1@liqiang-tfs',disc_copies},{ 'node3@liqiang-tfs',disc_copies}] = [schema,
```

```
user2,
```

```
user1]
```

<5> 在 node2 上运行下面命令, 彻底删除 node2 的 mnesia dir 下面的数据.

```
mnesia:delete_schema([node()]).
```