Adding tables to a cluster

This document describes how to add a table to an existing Bruce replication cluster[1]. An familiarity with administering PostgreSQL is presumed of the reader.

The simple case: A brand new table

- Create the table on the master database, using psql and 'create table' statements, including indexes, constraints, primary key, etc.
- Create the table on all the slaves. One way to do this is[2]:

```
pg_dump -n <table schema name> -t <table name> <master database> | psql -a <slave database>
```

- Create slave triggers on all slaves:

```
psql <slave database>
create trigger <table name>_deny
BEFORE INSERT OR DELETE OR UPDATE ON <fully qualified table name>
FOR EACH ROW EXECUTE PROCEDURE bruce.denyaccesstrigger();
\q
```

- Create master triggers:

```
psql <master database>
create trigger <table name>_sn
BEFORE INSERT OR DELETE OR UPDATE ON <fully
qualified table name>
FOR EACH STATEMENT EXECUTE PROCEDURE
bruce.logsnapshottrigger();
create trigger <table name>_tx
AFTER INSERT OR DELETE OR UPDATE ON <fully
qualified tables name>
FOR EACH ROW EXECUTE PROCEDURE
bruce.logtransactiontrigger();
\q
```

- Table is now being replicated, no daemon restart is necessary.

The not so simple case: An already populated table

- Lock the cluster.

```
psql <master database>
begin work;
lock table bruce.currentlog in access
exclusive mode;
--- Keep this transaction open for now
```

- In another window: create table on all slaves, including table data:

```
pg_dump -h <master host> -n <table schema
name> -t <table name> <master database name>
|
psql -h <slave hostname> <slave database
name>
```

- Create slave triggers on all slaves:

```
psql <slave database>
create trigger <table name>_deny
BEFORE INSERT OR DELETE OR UPDATE ON <fully
qualified table name>
FOR EACH ROW EXECUTE PROCEDURE
bruce.denyaccesstrigger();
\q
```

- Back to the open transaction on the master, create master triggers:

```
create trigger <table name>_sn
BEFORE INSERT OR DELETE OR UPDATE ON <fully
qualified table name>
FOR EACH STATEMENT EXECUTE PROCEDURE
bruce.logsnapshottrigger();
create trigger <table name>_tx
AFTER INSERT OR DELETE OR UPDATE ON <fully
qualified tables name>
FOR EACH ROW EXECUTE PROCEDURE
bruce.logtransactiontrigger();
```

- Close transaction:

```
commit;
\q
```

- Table is being replicated. No daemon restart is nessasary

Footnotes

[1]. For now. In a future version of Bruce, a method will be provided to ease this process.
[2]. Replace text within <> with appropriate values.