Database Remote-Copy Tool For SQLite

1. Usage

To cause the SQLite database named by REPLICA to be a copy of the current content of the ORIGIN database, run this command:

```
$ sqlite3-rsync ORIGIN REPLICA ?OPTIONS?
```

Use the `--help` or `-?` flag to see the complete list of options. Option flags may appear before, after, or between the ORIGIN and REPLICA arguments.

Add the `-v` option to see more output, in a format similar to "rsync".

2. Features

1. One or the other of ORIGIN or REPLICA may be of the form "USER@HOST:PATH". The other is just a simple PATH. This utility causes REPLICA to be a copy of ORIGIN.

    a. If REPLICA does not already exist, it is created.

    b. ssh is used for communication, so "USER@HOST" may be an SSH alias.

2. Both databases may be "live" while this utility is running. Other programs can have active connections to the databases on either end while this utility is running. Other programs can write to ORIGIN and can read from REPLICA while this utility runs.

    REPLICA becomes a copy of a snapshot of ORIGIN as it existed when the sqlite3-rsync command started. If other processes change the content of ORIGIN while this command is running, those changes will be applied to ORIGIN, but they are not transferred to REPLICA Thus, REPLICA ends up as a fully-consistent snapshot of ORIGINAL at an instant in time.

3. The synchronization uses a bandwidth-efficient protocol, similar to rsync (from which its name is derived).

3. Limitations

1. The database files must both be in WAL mode, and must have the same page-size.

2. While sqlite3-rsync is running, REPLICA is read-only. Queries can be run against REPLICA while this utility is running, just not write transactions.

3. Only a single database is synchornized for each invocation of this utility. It is not (yet) possible to synchronous many different databases using wildcards, as it is with standard "rsync".

4. At least one of ORIGIN or REPLICA must be on the local machine. They cannot both be databases on other machines.

5. On the remote system, this utility must be installed in one of the directories in the default $PATH for SSH. The `/usr/local/bin` directory is often a good choice. Alternately, the `--exe NAME` flag may be used to specify a remote location for the binary, e.g. `--exe /opt/bin/sqlite3-rsync`.

6. The replica will be a very close copy of the origin, but not an exact copy. All of the table (and index) content will be byte-for-byte identical in the replica. However, there can be some minor changes in the database header. In particular, the replica will have the following differences from the origin:

    a. The change counter in bytes 24 through 27 of the database header might be incremented in the replica.

    b. The version-valid-for number in bytes in 96 through 99 of the database header will be the SQLite version number of the sqlite3-rsync program that made the copy, not the version number of the last writer to the origin database.

7. On Windows, a single-letter HOST without a USER@ prefix will be interpreted as a Windows drive-letter, not as a hostname.

4. Network Bandwidth

The protocol is for the replica to send a cryptographic hash of each of its pages over to the origin side, then the origin sends back the complete content of any page for which the hash does not match.

Suppose the replica contains R pages. If the replica and the origin are already identical, then about R*20 bytes of content is sent from the replica to the origin, and apart from some trivial overhead and housekeeping traffic, nothing else moves over the wire. So for databases with a 4096-byte page size, the minimum bandwidth required to run this utility is equivalent to about 0.5% of the database. The worst case synchronization occurs if they replica and origin or completely different and have no pages in common. In that case, about total network traffic is about 100.5% of the database size.

The calculations in the previous paragraph do not consider the compression that SSH implements. Most SQLite databases are compressible and so the bandwidth cost of a complete synchronization is probably less than 100.5% of the database size. However, the cryptographic page hashes are not compressible, so the best case will never be better than about 0.5% of the database size, for a 4096-byte page size. Minimum bandwidth required goes down for larger page sizes. The best case is about 0.03% of the database size for databases with a 65,536-byte page size.

This page last modified on 2024-09-18 15:57:24 UTC

*** DRAFT ***