Subject: How to manage the access to a Db on an unstable connection
Posted by Giorgio on Tue, 16 Oct 2018 09:51:06 GMT
View Forum Message <> Reply to Message

Hi there,
I have developed an application used in the shop floor of a manufacturing company. Some clients collect production data and save them on a PostGresql DB. Initially, clients were wired to the LAN and there were no performance issues. The connection to the DB is managed in GUI_APP_MAIN: the connection is established and then I use the global SqlSession variable SQL throughout the whole application. Recently, some clients were added and connected through a wireless connection, as it was difficult to have cables running all over the shop floor. The wireless connection is unstable by definition, so it happens frequently that the connection is lost. Basically, there are two scenarios, when the connection is lost:

1. The user is just on a "view" status (a SqlTable filled with data) without issuing any sql statement.
2. The user is performing an action that requires an insert/update/select statement to be issued.

In the first scenario, if the client can restore the connection in a reasonable time, the user does not notice anything; otherwise everything on the SqlTable disappears: the user can cleanly exit the application and after restarting it usually everything works.

In the second scenario the application seems "freezed" (it is waiting for the DB server to respond, so it "hangs"). Sometime the "freeze" lasts only few seconds so the user just waits. In other cases the "freeze" lasts for more time (either because the connection with the db is completely lost, or because it takes sometime to the client to get back online), so the user kill the application.

For the above reasons, I have to rethink the way my application connects to the database, but I am a bit confused because is the first time I am in such a situation.

Basically I think the flow should be as follows:
-the application starts and connect to the db;
-after the initial select statements (to fill in all the necessary data) the application disconnects from the db;
-any time an operation that requires a db connection is performed (e.g. adding new data, when the SqlTable runs the Query() method, etc.) the application connects and - after the operation is performed - it disconnects;
-during each connection, an appropriate timeout is set and if the connection is failed the user is informed (so the user does not thing that he has to kill the application);
-if possible, the global SQL variable is used in the connection/disconnection actions.

So that is my idea, but I would be really happy if anyone want to share is ideas and experiences in developing a DB application in an environment where the connection is unstable.

Regards,
gio

Subject: Re: How to manage the access to a Db on an unstable connection
Posted by Zbych on Tue, 16 Oct 2018 19:24:49 GMT
View Forum Message <> Reply to Message

There is no need to open/close connection every time you want to fetch some data or update the database. You simple need to handle database errors correctly. When connection is lost you should catch exception, analyse error and try to reconnect if connection is lost. Also all inserts/updates can be surrounded by begin/commit/rollback and if you loose connection postgress will automatically drop unfinished inserts/updates.
Just check functions: ConnectionOK, ReOpen and callback WhenReconnect

https://github.com/ultimatepp/mirror/blob/master/uppsrc/Post greSQL/PostgreSQL.h

Subject: Re: How to manage the access to a Db on an unstable connection
Posted by Giorgio on Tue, 23 Oct 2018 08:46:36 GMT
View Forum Message <> Reply to Message

Hi Zbych,
thank you for your suggestions.

My routines that deal with the database have all the same structure:

```
int SaveData::TotalDrums(String c)
{
 int retval;
    SQL.ClearError();
 try {
  retval = SQL % Select(SqlSum(DRUMS))
   .From(MYTABLE)
    .Where( MYFIELD == AsString(c) );
 } catch(SqlExc) {
  retval = -1;
  ErrorOK(t_("Error: ") + SQL.GetLastError());
 }
 return retval;
}
```

Do you think I should catch also something else?
Is there on the upp site some documentation on ConnectionOK etcetera? I found almost nothing.
Regards,
gio