Item 28

git_comments:

git_commits:

1. **summary:** TServerSocket.cpp: Ensure the server is really listening (via helper variable listening_)
   message: TServerSocket.cpp: Ensure the server is really listening (via helper variable listening_) Client: cpp Patch: Mario Emmenlauer This closes #2232

github_issues:

github_issues_comments:

github_pulls:

1. **title:** TServerSocket.cpp: Ensure the server is really listening
   body: Currently the C++ server does not ensure that `listening_` is `true` in `isOpen()`. I think this is incorrect because in multi-threaded environments, the socket may not be open or listening yet. Also, `listening_` should only be set to `true` when the socket is fully set up and listening. This PR adds a minor change: it respects `listening_` for `isOpen()`, and sets `listening_` to `true` only at the end of spcket initialization. In heavy stress tests, we found this makes the server method `isOpen()` more reliable. Thanks for consideration. - [ ] Did you create an [Apache Jira](https://issues.apache.org/jira/projects/THRIFT/issues/) ticket? (not required for trivial changes) - [ ] If a ticket exists: Does your pull request title follow the pattern "THRIFT-NNNN: describe my issue"? - [x] Did you squash your changes to a single commit? (not required, but preferred) - [x] Did you do your best to avoid breaking changes? If one was needed, did you label the Jira ticket with "Breaking-Change"? - [x] If your change does not involve any code, include `[skip ci]` anywhere in the commit message to free up build resources.

github_pulls_comments:

1. > LGTM — eventhough, I think, people should pick the `TNonBlockingServer` for multithreaded environments, as it has at least some synchronization and is in general more suitable for threadpools and the like. I'm very curious about this! We did not try it yet, was it a simple drop-in replacement for you? It uses libevent, correct? Did you compare it with the new fbthrift c++ server?
2. Checked, the build errors in Travis are all unrelated. Ready for review and merge...
3. > was it a simple drop-in replacement for you? Not sure about this... > It uses libevent, correct? Better to check the source, but I think not. It just expects to be used from multiple threads, and does appropriate locking. > Did you compare it with the new fbthrift c++ server? No; what's that?
4. > > Did you compare it with the new fbthrift c++ server? > > No; what's that? Apparently Facebook has continued development on thrift in a separate project. I think it has less features and supports less languages, but it seems to have a cool new async c++ server. I've always wondered if this server could not be backported to Apache thrift, but that seems to involve complicated discussion of licenses. But for end-users, things may be simpler: possibly the server could just be used with Apache thrift if one would manage to take it out and build it against Apache thrift. No idea if this is possible, though :-) See https://github.com/facebook/fbthrift/blob/master/thrift/doc/Cpp2.md for more details.

github_pulls_reviews:

jira_issues:

jira_issues_comments: