Multi-Server Transactions with CFThread

Posted At : February 25, 2009 12:17 PM | Posted By : Jon Hartmann
Related Categories: ColdFusion

I ran into an interesting problem today: I have an application where I need to add a user record to one server, and also conditionally add some records to another server for that user. Problem is that this needs to be inside a transaction, and ColdFusion doesn't allow multiple server connections inside a transaction, even if they are on the same datasource. To get around this, I employeed <cfthread/>. Click "read more" to ready how.

The Setup

Ok, so my basic setup was something like this:

```
<!--- Add User --->
<cftransaction>
    <cfquery name="userInsert" datasource="my_datasource">
        <!--- Insert into table on Server1 --->
    </cfquery>

    <!--- Get the inserted userID back from SQL Server --->
    <cfset userID = userInsert.userID />

    <cfloop list="#form.toolIDs#" index="toolID">
        <cfquery name="insertTool" datasource="my_datasource">
            <!--- Insert into table on Server1 --->
        </cfquery>
    </cfloop>
</cftransaction>
```

But now I need to add records to another table for certain toolID values. I abstract this out into a function:

```
<cffunction name="addTool">
    <cfargument name="userID" />
    <cfargument name="toolID" />

    <cfswitch expression="#arguments.toolID#">
        <cfcase value="2">
            <cfquery name="insert" datasource="my_datasource">
                <!--- Insert into table on Server2 --->
            </cfquery>
        </cfcase>
        <cfdefaultcase>
            <cfquery name="insert" datasource="my_datasource">
                <!--- Insert into table on Server1 --->
            </cfquery>
        </cfdefaultcase>
    </cfswitch>

    <cfreturn />
</cffunction>

<!--- Add User --->
```

```
<cftransaction>
    <cfquery name="userInsert" datasource="my_datasource">
        <!--- Insert into table on Server1 --->
    </cfquery>

    <!--- Get the inserted userID back from SQL Server --->
    <cfset userID = userInsert.userID />

    <cfloop list="#form.toolIDs#" index="toolID">
        <cfset addTool(userID, toolID) />
    </cfloop>
</cftransaction>
```

The Problem

The problem is that I'm now accessing two servers in one transaction, and CF baulks at that. After monkeying around with the code for a bit, I figured out that the only way to escape the transaction is to run the code in another thread:

```
<cffunction name="addTool">
    <cfargument name="userID" />
    <cfargument name="toolID" />

    <cfthread name="addTool">
        <cfswitch expression="#arguments.toolID#">
            <cfcase value="2">
                <cfquery name="insert" datasource="my_datasource">
                    <!--- Insert into table on Server2 --->
                </cfquery>
            </cfcase>
            <cfdefaultcase>
                <cfquery name="insert" datasource="my_datasource">
                    <!--- Insert into table on Server1 --->
                </cfquery>
            </cfdefaultcase>
        </cfswitch>
    </cfthread>

    <cfreturn />
</cffunction>
```

ColdFusion throws a run-time error though, since I'm spawning multiple threads with the same name. Ooops. Lets try that again.

```
<cffunction name="addTool">
    <cfargument name="userID" />
    <cfargument name="toolID" />

    <cfset var thread = "addTool#arguments.tool#to#arguments.userID#" />

    <cfthread name="#thread#">
        <cfswitch expression="#arguments.toolID#">
            <cfcase value="2">
                <cfquery name="insert" datasource="my_datasource">
                    <!--- Insert into table on Server2 --->
                </cfquery>
            </cfcase>
            <cfdefaultcase>
                <cfquery name="insert" datasource="my_datasource">
                    <!--- Insert into table on Server1 --->
                </cfquery>
```

```
            </cfdefaultcase>
        </cfswitch>
    </cfthread>

    <cfreturn />
</cffunction>
```

Thats better, but now I've effectively negated the <cftransaction> tag: any errors generated stay inside the thread and the main thread keeps on chugging along. I could have just moved the loop outside of the transaction if I wanted to do that. I'm in luck though, because if the thread errors out, ColdFusion catches that and makes it available as a variable in the cfthread scope via cfthread["thread-name"].error I can use this to see if something went wrong.

```
<cffunction name="addTool">
    <cfargument name="userID" />
    <cfargument name="toolID" />

    <cfset var thread = "addTool#arguments.tool#to#arguments.userID#" />

    <cfthread name="#thread#">
        <cfswitch expression="#arguments.toolID#">
            <cfcase value="2">
                <cfquery name="insert" datasource="my_datasource">
                    <!--- Insert into table on Server2 --->
                </cfquery>
            </cfcase>
            <cfdefaultcase>
                <cfquery name="insert" datasource="my_datasource">
                    <!--- Insert into table on Server1 --->
                </cfquery>
            </cfdefaultcase>
        </cfswitch>
    </cfthread>

    <cfthread action="join" name="#thread#" timeout="1000" />

    <cfif StructKeyExists(cfthread[thread], "error")>
        <cfthrow message="#cfthread[thread].error.message#" />
    </cfif>

    <cfreturn />
</cffunction>
```

And it worked like a charm. I've even tested this by forcing the system to throw an error when on the second or third toolID, and everything rolls back as you would expect. By the way, as far as I'm aware, you can only use <cfrethrow/> inside <cfcatch/>, so you do have to do it by hand like I did.

The Solution

For those of you that wanted to skip to the end, or want to strip that example down to its bare bones, the code looks like this:

```
<!--- Create Transaction --->
<cftransaction>
    <!--- Add record to Server 1 --->
    <cfquery name="InsertServer1" datasource="my_datasource">
        <!--- Insert into table on Server1 --->
    </cfquery>

    <!--- Create Thread to Escape --->
    <cfthread name="Server2">
        <!--- Insert into Server 2 --->
        <cfquery name="InsertServer2" datasource="my_datasource">
            <!--- Insert into table on Server2 --->
```

```
        </cfquery>
    </cfthread>

    <!--- Join the thread back into the main thread --->
    <cfthread action="join" name="Server2" timeout="1000" />

    <!--- Check for Errors when inserting into Server 2 --->
    <cfif StructKeyExists(cfthread["Server2"], "error")>
        <!--- Throw an error about this error --->
        <cfthrow message="#cfthread['Server2'].error.message#" />
    </cfif>
</cftransaction>
```

Update: I forgot to list the thread join actions on the first pass. Code has been updated to reflect the change.