Subject: Application Crashes Due to Duplicated Concurrent INSERT Transactions
Posted by kong on Tue, 20 Sep 2016 05:25:55 GMT
View Forum Message <> Reply to Message

This is further on the issue that was brought up here http://radicore.org/fud/index.php?t=msg&th=2283&goto=5883&a mp;a mp;a mp;a mp;#msg_5883

Multiple submits of the same ADD1 forms will lead to concurrent INSERT transactions in the database. If the table uses non auto-increment primary keys or has other unique key constraints, this could result in fatal error thrown by the database (in this case MySQL) and application crash.

The checks built into the framework as described here http://www.tonymarston.net/php-mysql/functions-and-variables.html#n otes._dml_insertrecord should have prevented such crash and presented the user with user-friendly error messages instead. However, under certain race conditions which are hard to replicate, multiple submits of the same form in rapid succession could result in concurrent transactions invoked by calling startTransaction() function followed by insertRecord() method in the controller script for every click on the submit button. If you are unlucky, you could end up with these concurrent insertRecord transactions all working off the same pre-transaction snapshot of the database table and hence each transaction passing the above mentioned framework built-in checks without raising any errors, until one transaction has been committed successfully but then is followed by another transaction's INSERT which in turn makes the database throw a Duplicate Key Fatal Error, crashing the application.

To solve this problem, there are several options I can think of:
1) Explicitly set database table lock before the transaction starts by customizing _cm_getDatabaseLock() for every table with non auto-increment primary key or other unique keys.
2) Use a hidden field token to guard against multiple submissions of the same form, for example as shown in http://docstore.mik.ua/orelly/webprog/pcook/ch09_06.htm and http://phpsense.com/2006/prevent-duplicate-form-submission/.
3) Since we are relying on the framework for validations and unique key constraint checking anyway, we might as well use a different SQL insert command that does not throw Unique Key Fatal Errors on insert. For example in MySQL use INSERT IGNORE instead of INSERT.
4) Temporary lower the Isolation Level for Transactions in the database before the transaction begins. For example, in MySQL, lower from default level 3 'REPEATABLE READ' to level 1 'READ UNCOMMITTED'.

In my case I am using option 3 as that is least amount of work. But for the long run and improving robustness of the framework I think the 2nd option could be worth some effort.

Subject: Re: Application Crashes Due to Duplicated Concurrent INSERT Transactions
Posted by AJM on Tue, 20 Sep 2016 09:59:14 GMT
View Forum Message <> Reply to Message

Firstly, you should train your users not to keep repeatedly pressing the submit button in the hope

that it will make things go faster. This issue is not limited to the Radicore framework as it is eactly the same for *ALL* web applications.

Secondly, you are missing an option. I have seen some websites which, when a submit button is pressed, use javascript to disable the button. The button is not enabled until the operation started by that button is completed and the form is refreshed.

Thirdly, using an option which requires comparing a value in the form with a value stored in the session data would still not be foolproof as the second submission could read the session data before the first submission has updated it.

In all the years that the Radicore framework has been used this issue has never appeared. It sounds like the only sure-fire option would be to lock the entire table before each unsert so that concurrent inserts are then impossible. However, the side effect of this would be to slow down the system for all those users who are adding records to the same table.

Subject: jQuery Solution: Application Crashes Due to Duplicated Concurrent INSERT Transactions
Posted by kong on Sat, 01 Oct 2016 07:08:47 GMT
View Forum Message <> Reply to Message

Using jQuery it is quite effective to disable a button after the first click and show a spinner to indicate that the application is processing the request.

For those interested, let me summarize:

1) Get an image file for the spinner you like to use and save to radicore/css/ folder. Attached is the image I used (spin-red.svg).

2) Download jQuery file from https://jquery.com/download/ and save to radicore/javascript/ folder. Attached is the jQuery file I used (jquery-3.1.1.min.js).

3) For each module that you would like to implement this solution for, add the following to the module's style_custom.css file:

```
.loadingspinner
{
    background:
        url("../css/spin-red.svg")
        no-repeat
        center center;
}
```

4) Add the following to the *.class.inc files of the tables you need this solution for (Also check you have fixed http:// radicore.org/fud/index.php?t=msg&th=2299&start=0&amp ;)

```
    function _cm_setJavaScript ($javascript)
        // insert any javascript to be included in the <HEAD> or <BODY> elements.
    {
        $javascript['head'][]['file'] = '../javascript/jquery-3.1.1.min.js';
```

```
$javascript['foot'][]['code'] =  "
    $('form :submit').click(
        function () {
            \$(this).
            addClass('loadingspinner').
            prop('disabled', true).
            closest('form').
            append(\$('<input/>', {
                type: 'hidden',
                name: this.name,
                value: this.value
            })).
            submit();
        }
    );";

return $javascript;

} // _cm_setJavaScript
```

That's all, enjoy!

References:
http:// stackoverflow.com/questions/22509343/prevent-double-form-sub mission
http:// stackoverflow.com/questions/12837335/how-to-display-loading-spinner-in-a-textbox-on-clicking-of-a-button

File Attachments

1) jquery_solution.7z, downloaded 1639 times

Subject: Re: jQuery Solution: Application Crashes Due to Duplicated Concurrent INSERT Transactions
Posted by AJM on Sat, 01 Oct 2016 08:20:13 GMT
View Forum Message <> Reply to Message

That looks very interesting. When I get the time I will check it out.

Subject: Re: jQuery Solution: Application Crashes Due to Duplicated Concurrent INSERT Transactions
Posted by AJM on Wed, 05 Oct 2016 17:09:52 GMT
View Forum Message <> Reply to Message

Attached is a file that will make it possible to define in one file any javascript which is to be applied

on every form and optionally for every field. The zip file contains the following:

custom_javascript.class.inc - to be placed in your 'includes' directory
include.xml.php5.inc - modified to read this file (if it exists)

I have tried this, and although it works OK in Firefox it doesn't work in Internet Explorer.

File Attachments

1) custom_javascript.class.zip, downloaded 1674 times

Subject: Re: jQuery Solution: Application Crashes Due to Duplicated Concurrent INSERT Transactions
Posted by kong on Wed, 05 Oct 2016 23:32:16 GMT
View Forum Message <> Reply to Message

Made some changes for IE, please update $javascript['foot'][]['code'] to the following:

```
$javascript['foot'][]['code'] =  ''
            $('form :submit').click(function () {
                \$(this).
                prop('disabled', true).
                closest('form').
                append(\$('<input/>', {
                    type: 'hidden',
                    name: this.name,
                    value: this.value
                })).
                submit();
                \$(this).addClass('loadingspinner');
                event.preventDefault();
            });'';
```

IE11 does not support svg file, so I attached a gif file for the spinner. Extract and save spin-red.gif file to radicore/css folder and update default.css or whatever css file you use:

```
.loadingspinner
{
  background:
        url("spin-red.gif")
        no-repeat
        center center;
}
```

I have tested this successfully on:
Chrome 53
Firefox 49
Waterfox 48
IE 11

File Attachments
1) spin-red.7z, downloaded 1659 times

Subject: Re: jQuery Solution: Application Crashes Due to Duplicated Concurrent INSERT Transactions
Posted by AJM on Thu, 06 Oct 2016 09:29:25 GMT
View Forum Message <> Reply to Message

That works fine in IE now. Thank you for your efforts. I shall incorporate this into the standard product and include it in the next release. I shall mention this contribution on my "Credits" page at http://www.radicore.org/credits.php

Subject: Re: jQuery Solution: Application Crashes Due to Duplicated Concurrent INSERT Transactions
Posted by AJM on Thu, 06 Oct 2016 15:35:20 GMT
View Forum Message <> Reply to Message

I have found a slight problem with this approach - if a LIST page has a navigation button which sends back a CSV file to the client, the navigation button is disabled with the spnning image, but when the CSV file is closed the LIST page is not refreshed and the spinner does not disappear. This also means that the navigation button is disabled and cannot be pressed again. The only way around this is to click on another button or hyperlink.

Subject: Re: jQuery Solution: Application Crashes Due to Duplicated Concurrent INSERT Transactions
Posted by kong on Fri, 07 Oct 2016 07:17:05 GMT
View Forum Message <> Reply to Message

As reference, here is a good description of the cause of the problem related to downloads in general: http:// gruffcode.com/2010/10/28/detecting-the-file-download-dialog- in-the-browser/

In short: For any actions that do not result in a page refresh or redirect, we need to make use of a temporary cookie, which when detected in the browser will trigger jQuery to clean up the screen (meaning get rid of previous spinners and re-enable buttons).

Here are the steps to implement this update:

1) Download and extract 2 files from the attachment. An updated custom_javascript.class.inc and a new file jquery.cookie.min.js which should be saved in the radicore/javascript folder.

2) In std.output1.inc and std.output4.inc, find this line of code:header("Content-type: text/comma-separated-values");and change it to:setcookie('clean_up_screen_' . session_name(), $GLOBALS['task_id'], time()+30);
header("Content-type: text/comma-separated-values"); When these transactions are activated

they will set a 30 seconds cookie by session_name and by controller path upon start of download, which is the signal for the browser to clean up the screen without the need for screen refresh. The cookie will be deleted after each clean up. So far, I have only identified these 2 patterns that need this fix. Please let me know of other transaction patterns that also do not result in page reloads or redirects, as those will need similar fix as well.

I have checked this on
Chrome 53
Firefox 49
Waterfox 48
IE 11

File Attachments

1) fix_csv.7z, downloaded 1588 times

Subject: Re: jQuery Solution: Application Crashes Due to Duplicated Concurrent INSERT Transactions
Posted by AJM on Fri, 07 Oct 2016 09:48:48 GMT
View Forum Message <> Reply to Message

That works brilliantly! I cannot see any other patterns that would need this change. The patterns for PDF files use a different mechanism so do not have the same problem.
