****

**Student ID: C00131026**

**Student Name: Guanting Su**

**Supervisor: Joseph Kehoe**

**Date of submission: 8.4.2011**

**Project Report**

**<STM Lua>**

**(CW228)**

Content

[1. Introduction 1](#_Toc290046607)

[2. Data structures 2](#_Toc290046608)

[3. Module descriptions 4](#_Toc290046609)

[1) Create Transaction 4](#_Toc290046610)

[2) Add to create a table 4](#_Toc290046611)

[3) Read 4](#_Toc290046612)

[4) Write 4](#_Toc290046613)

[5) Commit transaction 4](#_Toc290046614)

[6) Create thread 5](#_Toc290046615)

[7) Call function 5](#_Toc290046616)

[8) Print global table 5](#_Toc290046617)

[4. Update earlier documents 5](#_Toc290046618)

[5. Problems and Solutions 6](#_Toc290046619)

[5.1. New Language and C language 6](#_Toc290046625)

[5.2. Need achieve Goto in Lua 7](#_Toc290046626)

[5.3. Create thread and lock 7](#_Toc290046627)

[5.4. Lua call function from c 8](#_Toc290046628)

[5.5. C call function from Lua 10](#_Toc290046629)

[5.6. Merge them together 12](#_Toc290046630)

[5.7. Fail to create thread 12](#_Toc290046631)

[6. What I achieved & not achieve 13](#_Toc290046632)

[7. Testing 13](#_Toc290046633)

[8. What I learned 16](#_Toc290046634)

[9. What I would do differently if starting again 17](#_Toc290046635)

[10. Conclusion 18](#_Toc290046636)

# Introduction

My project is developed a concurrency control into a library of Lua. There are many techniques to achieve concurrency control, such as Locking, STM (Software Transaction Memory) and so on. And my project is implemented by STM and Lua. In my project use STM instead of locking to control parallel programming.

STM there are two types, one is lock-free and the other one is based-lock. But in STM, the locking only need a few time. For example, only after process, it need lock to let resource cannot be available to others. If it finds other one use it, it will restart process the data. Compare STM with Locking, the STM can avoid dead lock and live lock and most of time is more efficient than Locking.

Lua is a powerful, fast, lightweight, simple, embeddable scripting language. Lua can run on all kinds of UNIX and Windows, and mobile devices as well. I build my project to a Lua library, and then any one use Lua can require this library to achieve concurrency.

To let my project support locking and create multithreads, I need C language to call system help. In the c code write down the functions, and let Lua call function from C. Therefore my project is include dll file is written by C in Visual Studio 2010 environment. And the Lua part is written in Lua for windows environment. In Lua for windows, there are many libraries and compiler let me coding Lua code earlier and more convenient. Because usually the Lua install, only Lua standard libraries and command line control. But my project doesn’t use the third part libraries.

In Lua the most popular data structure is table. In Lua, table can simulate array, link-list, map and so on. Lua is not OO language and it just is script language. But Lua can use table achieve class and object. Metatable can overwrite the table operations.

So far there are createmetatable, createtransaction, paraDo, callfunc, createT, printGtable 5 global functions. There 6 functions in a class called “x”, they operation of STM. And in c code, there are 7 functions; they create thread, call function from Lua and setlock and so on. In the module section I will descript more details.

Testing is always a big part of the project. For this project, test is more important. Because my project supports multithread, so synchronization is big problem, it need time and test to make sure it works. There are some simple examples which used to test this program showed in the testing part.

# Data structures

In my project, the data structure is the most widely used is table in Lua. In Lua, array is table, the class and object also use table. For example, this is a empty table a={}, create a new array, a={1,3,5,7}, from “Programming in Lua” book, I know the table can simulate different data structure, matrix, queue, link-list, set, map and so on. Because Lua is script language not OO language, but Lua can use table simulate class and object operations. For example,

x={}

function x.add()

function x.read()

function x.write()

function x.commit()

The function add, read, write and commit are elements of table x. Hence if set y=x, y.add() is the same as x.add(). Also can create class like this, is the same.

x={

 add=function()

…………………

}

Except table is used in my project, my project also use metatable. Metatables allow us to change the behavior of a value when confronted with an undefined operation. For instance, using metatables \_add field, we can define how Lua computes the expression a+b, where a and b are tables. If you want to read table also can using metatbale \_index return what you define return value. For example z={a={value=1,version=2}}, usually can print out like print(z[“a”].value), but if set table to metatable setmetatable(z) and define \_index return t[key].value, now you can print out like print(z[“a”]). These two ways get same answer. Sometimes use metatable can let coding more efficient.

The next data structure is stack. But this stack is use to achieve Lua Interacts with C. C and Lua transform value each other by stack.

For example, Lua call function from C. coding this function in C has a standard type:

static int lua\_test(Lua State \*L){……..}

in Lua call function

lua\_test(2,4)

In here, the stack will save 2 in index 1 and 4 in index 2. Now the function get value, in the C code will process as below show:

static int lua\_test(Lua State \*L){

 int n = lua\_gettop(L); //get the size of statck

 if(n!=2){ //means not transform two value

 return 0; //end

 }

 //check index 1 and index 2 is digital or not

 if(!lua\_isnumber(L,1)||!lua\_isnumber(L,2)){

 //push error message in stack

 luaL\_error(L,"please input number!!");

 }else{

 //push the result into stack

lua\_pushnumber(L, lua\_tonumber(L,1) + lua\_tonumber(L,2));

 }

 return 1; //end

 }

# Module descriptions

The module is similar with what is described in design document. only make my project more reasonable.



## Create Transaction

In this function create a new empty table to store the users need process data. Because the table except data also has many functions to achieve STM control.

## Add to create a table

The programmer can use it to add variables into a table. The table will save value of valuable, initial a version number and lock.

## Read

Read function is get the value, version and lock of variables from global table.

## Write

Update new value, version number and lock into the global table.

## Commit transaction

Before the transaction end, in the commit function, the transaction will lock the valuable and check version and then update version and valuable.

## Create thread

Call C function create a thread to run function op1. This one requires the project support synchronization, so need locking, make sure get the right answer.

## Call function

Run function op1 without thread. Doesn’t care the synchronization, the transactions will be running in turns.

## Print global table

Print out the value of variables in global table one by one.

# Update earlier documents

In my design document, I introduced my data design and architectural design. But in the coding, I modified the data and architectural became better and fit for my project.

When user creates transaction, my project will create a new table to store variables. In the table, every variable also use table to store value, version and lock of variable.



Global table stores variable and variable also is a table

And the architectural design also change, in the coding, after user create transaction, can call add() function to add variables into table. For example, y=createtransaction() y.add(“a”,8). And then user can process variable directory, such as y.a=y.b+y.c. at the end of transaction, user can call commit function to update the variable to global table. At last call creatT(functioname) or callfunc(functioname) to run the functions.



New architectural

Others only change function name and add some new function into my project. And before I didn’t know I need create thread in C code. Now I use C code to call system to create a new thread. Because this reason, I research how to create thread in Lua.

At the beginning, I think I can use coroutines or lanes to create threads. The coroutines doesn’t create thread, just use stack to save states to simulate multithreads. When I ask my supervisor, he said this is one is not create thread, so I ask him how about the lanes. The lanes is a library support creates threads. But I don’t why, I create thread by lanes to run the function, there is error return my function is nil value. And supervisor need me create thread by myself. Therefore I research how to create thread in Lua, and I found create thread and lock need call system to create. Now I call windows API to create thread and lock.

# Problems and Solutions

1.
2.
3.
4.
5.

## New Language and C language

Before I chose the project, I never heard Lua, and didn’t what Lua is. I learn Lua from “Programming in Lua” book and some website. I code some examples. Then start to develop my project, when I was in trouble, I would find out the solution in the book or in the Internet. And my supervisor also helps me and gives me some suggestions.

The C language, I learn it in the last year. But usually I rarely use C and I understand C. this time I need create a C code let Lua can call function to create thread to run the Lua function what is define by users. There is a little different from normal C code. I also found out solution from the book and the Internet.

From these problems, I understand learn very new language need practices more and more. And when be in trouble, need know why the errors occur and find the solution in the book or Internet, or ask the supervisor.

## Need achieve Goto in Lua

At the beginning, supervisor and I decided the transaction should be like this:

begin()

………….

commit()

When fail to commit, the transaction need go to begin and restart. In other language, I just need set a signal in begin function, and the when fail to commit, use goto language come back the begin function and then restart.

But the Lua there is no goto, so I need consider how to solve. First I think create a C function in C to achieve this purpose. Unfortunately, I didn’t find out the solution. At last my supervisor gives me an example STM what is written in Java. In the example, it use while loop to achieve when fail to commit.

While done==false do

………………………

done=commit()

end

Now require commit return a Boolean value. If commit is successful, then return true and end the loop, if not return false and keep looping.

## Create thread and lock

I tried to use coroutines and lanes to create thread. Because supervisor requires my project need create thread by self, so I need find out how to create it in Lua. Hence I research in the Internet. After research, the create thread and lock need call system to create. Because my system is windows, so I search how to create them in windows. At last I found windows API, and some examples. Following is simple sample:

int time = 0;

HANDLE Mutex;

void main()

{

 HANDLE thread1,thread2;

 thread1 = CreateThread(NULL,0,Fun1Proc,NULL,0,NULL);

 thread2 = CreateThread(NULL,0,Fun2Proc,NULL,0,NULL);

 CloseHandle(thread1);

 CloseHandle(thread2);

 Mutex = CreateMutex(NULL,FALSE,NULL);

 cout << "main thread is running" << endl;

 Sleep(4000);

 system("pause");

}

DWORD WINAPI Fun1Proc(LPVOID param)

{

 ………………………….

 return 0;

}

DWORD WINAPI Fun2Proc(LPVOID param)

{

 ………………………………….

 return 0;

}

This sample shows me how to create thread, set a lock and release lock. Although this sample doesn’t fit for my project, but give me idea to develop my project.

When I research more I found, create thread in C code not only one ways. Except Createthread from Windows API, there are others. They are \_beginthread, \_beginthreadex and pthread\_create. The \_beginthread, \_beginthreadex belong to c library. It is also call Createthread from windows API, but they better than directory use Createthread to create thread. pthread\_create is come from pthread.h in Linux.

## Lua call function from c

Because thread only can be created in C code, so I need develop a C function to be called by Lua. But the C code not just create .c file. From I research in the Internet, the C code need to develop into a dll file. In the C code, I create function style is followed the standard style that is define by Lua. For example:

typedef int (\*lua\_CFunction) (lua\_State \*L)

I also found some examples to let test and found out the solution to solve problems. Below is a sample:

//connect the lib of Lua

#pragma comment(lib,"../lib/lua5.1.lib")

#include "lua.h"

#include "lauxlib.h"

#include "lualib.h"

//function add, get sum of two values

static int Lua\_add(lua\_State \*L)

{

 int n = lua\_gettop(L);

 if(n!=2){

 return 0;

 }

 if(!lua\_isnumber(L,1)||!lua\_isnumber(L,2)){

 luaL\_error(L,"please input number!!");

 }else{

lua\_pushnumber(L, lua\_tonumber(L,1) + lua\_tonumber(L,2));

 }

 return 1;

}

//register function into library mylibs

static const struct luaL\_reg mylibs [] = {

 {"add", Lua\_add},

 {NULL, NULL} /\* sentinel \*/

};

// register a library into Lua

\_\_declspec(dllexport) int functionExport(lua\_State \*L)

{

 //luaL\_openlib(L, "\_G", mylibs, 0);

 luaL\_register(L,"mylib",mylibs);

 return 1;

}

In Lua file, I just need type:

mylib = loadlib("fullname-of-your-library","luaopen\_mylib")

mylib()

print(mylib.add(3,4))

If register library into \_G of Lua, then you just need print(add(3,4)), because it is global function. Run the Lua code, it will print out 10.

From this example, I know how code in C. I in the Lua reference manual there are a lot of C API, let me know how to develop my application. For example, upstairs show lua\_isnumber(L,1), in my application the transform value is string, so I change to lua\_isstring(L,1). And use a lot of samples to understand how to build the C function can be called by Lua and achieve my reqirment.

## C call function from Lua

After I solve the problem is how Lua calls function from C to create thread. And then I meet a new problem, how I let the thread running the transactions what are code by users. When I couldn’t find the solution, I ask my supervisor for help. He gave me some suggestions. Store function name what is running a transaction and transform to C code and let C process.

To achieve the purpose, I read Lua reference manual and some website. At last I found some examples are very useful.

int use\_lua\_paraDo(const char \*func\_name,int x, int y)

{

 int sum=0;

 lua\_getglobal(L, func\_name); /\* function to be called \*/

 if (!lua\_isfunction(L, -1)){

 luaL\_error (L,"%s is not a function",func\_name);

 return -1;

 }

 lua\_pushnumber(L, x); /\* push 1st argument \*/

 lua\_pushnumber(L, y); /\* push 2nd argument \*/

 /\* do the call (2 arguments, 1 result) \*/

 if (lua\_pcall(L, 2, 1, 0) != 0){

 printf("fail to call function");

 return -1;

 }

 sum = (int)lua\_tointeger(L, -1); //get result

 lua\_pop(L, 1);

 return sum;

}

//create thread

void createThread(lua\_State \*L,unsigned ( \_\_stdcall \*Threadfunc )( void \* )){

 HANDLE newThread;

 newThread=(HANDLE)\_beginthreadex(NULL,0,Threadfunc,NULL, 0, NULL) ;

 CloseHandle(newThread);

}

//let thread running the function from Lua

void ThreadProc1 (void \* pParam){

 int a=0;

 a=use\_lua\_paraDo("add",4,6);

 \_endthread();

}

int main()

{

 int sum = 0;

 L= lua\_open(); //connect to Lua

 InitLuaState(L);

 //load and run test15.lua file

 luaL\_dofile(L, "../test15.lua");

 createThread(L,threadFunc);

 lua\_close(L); //close

 return 0;

}

Above example is integrated my idea and example of Internet. In this example I create a thread to running the function what is in Lua through the function name. In general it works, but sometimes the memory lead will occur. This problem lead to this example will random occur errors.

I also try to code can call function from Lua to run the functions what is defined by users.

int use\_lua\_paraDo1(const char \*func\_name,const char \*a)

{

 int sum=0;

/\* function to be called \*/

lua\_getglobal(L, func\_name);

if (!lua\_isfunction(L, -1)){

 printf("%s is not a function",func\_name);

 return -1;

 }

/\* push 1st argument \*/

 lua\_pushstring(L, a);

/\* do the call (1 arguments, 1 result) \*/

 if (lua\_pcall(L, 1, 1, 0) != 0){

 printf("fail to call function");

 return -1;

 }

 //lua\_call(L, 2, 1);

 sum = (int)lua\_tointeger(L, -1);

 lua\_pop(L, 1);

 return sum;

}

In this example, in Lua code there is function f(parameter), parameter is string type. Therefore lua\_getglobal(L,”f”) means call function f, and lua\_pushstring(L,”a”) means call function f(“a”).

## Merge them together

When I try to merge Lua call C functions and C call Lua functions together. There is one problem, this is Lua state. They are share the same Lua state or not. I try both of them. The first one, in C code I set a global lua state \*mainstate. When Lua call C functions, C code get state \*L and set mainstate=L. And when C call Lua function, also use this mainstate, such as lua\_getglobal(mainstate, func\_name). The other one, Lua call C functions is the same, but when C call Lua functions, will create a new state \*L, initial L, lua\_dofile(L,”XX.lua”).

From I test, the first one is work well, the second so far won’t work.

## Fail to create thread

At the 5.4 create thread; I introduced Createthread, \_beginthread and \_beginthreadex. Almost every tutorial told me doesn’t use Createthread, should use \_beginthread and \_beginthreadex, they are better. But they also said \_beginthread and \_beginthreadex belong to C runtime library, and then invoke Createthread to create new thread. That is let me confuse. When I read more information, then I know why. If I directory use Createthread, it is easy to lead to memory leak. Because the C there are a lot of global variable, if use Createthread will set wrong value to the variables and then lead to memory errors. But \_beginthread and \_beginthreadex belong to C runtime library, so they will create private variables to store value, this avoid to lead to memory errors occur, and then can safety to call Createthread to create new thread. And avoid memory leak, also doesn’t use will terminal thread functions. We should let thread finish as nature. And when I test the thread is successfully created in my test code. But when I test with my Lua code, there are random errors occurring. When I can’t figure out any error in C code, I try to test my Lua code only. I found my Lua code has some logical error, when I fixed that the application can create thread successfully.

# What I achieved & not achieve

So far I have done:

* STM basic control
* Create lock
* Create multithreads

Not achieve:

* Support more STM control and done more parallel work.
* Support more operations of thread.
* Testing more

# Testing

Create two transactions, add same variables into table.

require "stm"

function op1()

 local done=false

 y=createTransaction() //create transaction

 y.add("a",4) // add a into table y

 y.add("b",5)

 y.add("c",7)

 while done==false do

 y.a=y.a+y.b //process data y.a=4+5

 y.b=y.c-y.b

 y.c=y.c+1

 done=y.commit() //commit thransaction update

 end

end

function op2()

 local done=false

 z=createTransaction()

 z.add("a",6)

 z.add("b",8)

 z.add("c",9)

 while done==false do

 z.a=z.a+z.b

 z.b=z.c-z.b

 z.c=z.c+1

 done=z.commit()

 end

end

callfunc("op1") //run function op1

callfunc("op2")

printGtable()

At last the result is:

a 9

c 9

b 6

Because the global table there is variable a and its value is 2, so the result is right.

Now I change callfunc() to creatT().

createT("op1")

createT("op2")

Also get a same answer.

Create two transactions, add different variables into table.

function op1()

 local done=false

 y=createTransaction()

 y.add("a",4)

 y.add("b",5)

 y.add("c",7)

 while done==false do

 y.a=y.a+y.b

 y.b=y.c-y.b

 y.c=y.c+1

 done=y.commit()

 end

end

function op2()

 local done=false

 z=createTransaction()

 z.add("d",6)

 z.add("e",8)

 z.add("f",9)

 while done==false do

 z.d=z.d+z.e

 z.e=z.f-z.d

 z.f=z.f+1

 done=z.commit()

 end

end

if use callfunc():

a 7

c 8

b 2

e -5

d 14

f 10

If use createT() also get the same result.

a 7

c 8

b 2

e -5

d 14

f 10

Test synchronization:

function op1()

 local done=false

 y=createTransaction()

 y.add("b",1)

 while done==false do

 y.b=y.b+1

 done=y.commit()

 end

end

function op2()

 local done=false

 z=createTransaction()

 z.add("b",1)

 while done==false do

 if z.b>1 then

 z.b=z.b-1

 end

 done=z.commit()

 end

end

i=1

while i<=3 do

 createT("op1")

 createT("op2")

 i=i+1

end

The result is:

b=2

b=1

b=2

b=1

b=2

b=1

# What I learned

In this time, I learn much new knowledge. I think this time developing will let me know more about the process of project develop, and time control is very important as well.

In the writing part, I have deeper understand the documents is very important to develop application. From the documents, I can get ideas to build my application, and know what techniques should be implemented.

In the technical part, I learn new language-Lua, and how to develop an application with windows API. There is a deeper understand C language. But so far I am still feeling C language is more difficult to understand than Java, perhaps I doesn’t enough understand C language. Special the memory error, it is very hard fix. From this time, I need keep learning more about C and C++. In the hand, from the developing I know how to embedded Lua into C and learn how to call C function from C in Lua. And if you need write parallel programs, the thread and lock should be created by systems. And know concurrency control not only locking, there are other techniques to instead of it, such as STM (Software Transaction Memory).

For my personal part, I learned how to use well the search engine. In the Internet there are a lot of resource to help me solve problems and study. I feel time is running faster than I imagine. Let me know time control is important and time is important as well. In this time I wasted a lot of time do nothing, lead to I far behind in my schedule. And the research and design document is very important. Research helps me understand what my project is, what technique need to be used and so on. And the design document can help me get the ideas to develop my project and decide use what technique to develop. Listen supervisor’s suggestions also are very helpful to develop project. It will help you get new ideas, right direction to develop and save time.

# What I would do differently if starting again

If I have chance start develops my project again. I will notice following points:

In the research, I will know more about what my project is and if I need develop my project, what technique will be need. Get clear what my project should be developed to. In the design document, I will have main idea to develop the project. And integrates supervisor’s suggestions, I think I can develop project more efficient.

In this time, supervisor helps me very much. Without my supervisor suggestion, I think I can’t develop my project. Therefore if start again, also needs a good communicate with supervisor.

The schedule is so important. If start again, I will resolutely follow the schedule that is made by myself. If I can in front of my schedule also is good. Because I have more time to test and debug to let my project become more perfect.

Take more time to learn Lua and C language. Learn more about what metatable is and how it works. Get more experience about Lua and C coding.

# Conclusion

Through this time done the project, I learn much knowledge that I never know and even heard before. And I found my some disadvantages in developing. The developing project experience will be very helpful for my life. May be not only for coding, I think do anything before should need some research what I need to prepare, have a plan, and should be good at time control. When be in trouble, don’t be afraid, upset and give up easily and earlier, we need clam down and patiently to find out the reason. Sometimes the problems you can’t find solution completely, do not upset and discouraged. But your friends, classmates or teachers may can help you or give you some suggestions. People can get new knowledge in difficult.