

VisIt - Bug # 1193: VisIt can't connect client/server to sims when ssh tunneling is on.

Status:	Pending	Priority:	Normal
Author:	Brad Whitlock	Category:	
Created:	09/06/2012	Assigned to:	
Updated:	01/22/2013	Due date:	
Likelihood:	3 - Occasional		
Severity:	3 - Major Irritation		
Found in Version:	2.5.2		
OS:	All		
Support Group:	Any		
Subject:	VisIt can't connect client/server to sims when ssh tunneling is on.		
Description:	VisIt isn't successfully connecting client/server to sims when ssh tunneling is on. It works when ssh tunneling is off. However, when tunneling is on, the connection seems to be made but the type representations read in the sim contains all zeros. Maybe the socket bridge in vcl is to blame? This is happening on the trunk but I think I have also seen it in released versions of VisIt.		

History

09/26/2012 07:56 pm - Eric Brugger

- Status changed from New to Pending

10/11/2012 02:08 pm - Eric Brugger

- Target version deleted (2.6)

01/22/2013 06:46 pm - Brad Whitlock

We've recently upgraded from VisIt 2.5.2 to 2.6.0, and are having some trouble connecting to instrumented simulations. We're able to reproduce the problem with a simple test program (included below, with log from -debug 5 run), and it looks like a socket issue. We've already done all of the troubleshooting we could think of on our firewall settings, and hope that maybe someone could help us here.

Here's a walk through of what we see: We launch the simulation (compiled from the code below) on a linux node, and it starts normally. It creates a sim2 file in ~/.visit/simulations with what I would expect are proper settings, the name of the node, the port, etc. And we have it set to blocking, so it waits for an incoming connection.

At this point, we start VisIt 2.6.0 on Windows (with -debug 5 to get some debug info) click open, select the linux host entry in the connection window, which starts qssh (prompts me for my password), which launches the metadata server. I browse to the file location (~/.visit/simulation/*.sim2), and open the file.

On the linux side, the libsim V2 code attempts to create 1 `read` and 3 `write` sockets, with the first two connections working, and the next two connections refused. It then uses the first (`read`) socket to query the version of the windows client. It's at this point that the `recv` function fails (returns -1), but no error condition is generated yet. Shortly after, the code attempts to get the version of the windows client from the still-empty buffer, and this evaluates to a blank version string. When it's compared to `2.6.0` local (linux) version string, it causes the

version check to fail and an exception to be generated. The resulting error is written to the log `³A.visit.5.vlog²` and the program exits with:

Versions are `()`,2.6.0

Exception: (IncompatibleVersionException)

On the desktop side, the "compute engine launch progress" windows is still waiting, and has to be cancelled.

Some questions we have are: what are the 1 'read' and '3' write sockets used for? Also, why would some of the sockets (first read and first write) connect, while the others have connection refused? It looks like even the connected read socket has a problem, as the 'recv' called on it returns -1, leaving the zeroed-out buffer unaltered.

The comments in the logs says that 9/10 of the issues are with firewall and host file config. What is the ideal setup in the `/etc/hosts` file? (should the hostname be listed with the 127.0.0.1 address? should it resolve itself to the actual IP of the first interface?)

Here is the sample code we used (condensed purposely):

```
#include "VisItControllInterface_V2.h"

#include <iostream>
#include <stdlib.h>

int main( int argc, char* argv[] ) {

    VisItSetParallel( false );

    char* env = NULL;
    env = VisItGetEnvironment();

    VisItSetupEnvironment2( env );
    if ( env != NULL ) {
        free( env );
    }

    VisItInitializeSocketAndDumpSimFile( "simulation", "Simulation", "",
    NULL, NULL, NULL );
    VisItOpenTraceFile( "libsim_trace.log" );

    std::cout << "Waiting for connection from VisIt" << std::endl;
    int blocking = 1;
    int visitState = VisItDetectInput( blocking, -1 );

    switch ( visitState ) {
        case 0:
            break;
        case 1:
            if ( VisItAttemptToCompleteConnection() == VISIT_OKAY ) {
```

```

std::cout << "VisIt connected to simulation" << std::endl;
}
else {
std::cout << "VisIt did not connect to simulation" << std::endl;
return 1;
Break;
}
case 2:
break;
default:
std::cout << "Unknown VisIt state" << std::endl;
return 1;
break;
}

VisItCloseTraceFile();

return 0;
}

```

And the associated log:

```

0visit -v 2.6 -engine -debuglaunch -nolookback -host node-01 -port 39879
-key 89ea67c1a29831b2aa9d

```

ENGINE started. pid: 17154

```

ParentProcess::Connect: Called with (numRead=1, numWrite=3, argc=14,
argv=(visit, -v, 2.6, -engine, -debuglaunch, -nolookback, -host, node-01,
-port, 39879, -key, 89ea67c1a29831b2aa9d, -key, 89ea67c1a29831b2aa9d))
ParentProcess::Connect: hostName = node-01

```

```

ParentProcess::GetHostInfo: Calling gethostbyname("node-01")

```

```

ParentProcess::Connect: port = 39879

```

```

ParentProcess::Connect: securityKey = 89ea67c1a29831b2aa9d

```

```

ParentProcess::Connect: securityKey = 89ea67c1a29831b2aa9d

```

```

ParentProcess::Connect: Creating sockets

```

```

ParentProcess::Connect: Creating read sockets

```

```

ParentProcess::GetClientSocketDescriptor: Set up using port 39879

```

```

ParentProcess::GetClientSocketDescriptor: Creating a socket

```

```

ParentProcess::GetClientSocketDescriptor: Setting socket options

```

```

ParentProcess::GetClientSocketDescriptor: Calling connect

```

(If you see no messages after this one, VisIt was not able to connect to the client machine. Nine times out of ten, this is a firewall issue on the client machine.

It could also mean that VisIt was unable to resolve the IP address for the client machine. You may need to verify the contents of /etc/hosts.)

```

ParentProcess::GetClientSocketDescriptor: Connected socket

```

```

ParentProcess::Connect: Creating write sockets

```

```

ParentProcess::GetClientSocketDescriptor: Set up using port 39879

```

```

ParentProcess::GetClientSocketDescriptor: Creating a socket

```

```

ParentProcess::GetClientSocketDescriptor: Setting socket options

```

ParentProcess::GetClientSocketDescriptor: Calling connect
(If you see no messages after this one, VisIt was not able to connect to the client machine. Nine times out of ten, this is a firewall issue on the client machine. It could also mean that VisIt was unable to resolve the IP address for the client machine. You may need to verify the contents of /etc/hosts.)

ParentProcess::GetClientSocketDescriptor: Connected socket

ParentProcess::GetClientSocketDescriptor: Set up using port 39879

ParentProcess::GetClientSocketDescriptor: Creating a socket

ParentProcess::GetClientSocketDescriptor: Setting socket options

ParentProcess::GetClientSocketDescriptor: Calling connect
(If you see no messages after this one, VisIt was not able to connect to the client machine. Nine times out of ten, this is a firewall issue on the client machine. It could also mean that VisIt was unable to resolve the IP address for the client machine. You may need to verify the contents of /etc/hosts.)

ParentProcess::GetClientSocketDescriptor: Could not connect! (error=111: Connection refused)

ParentProcess::GetClientSocketDescriptor: Set up using port 39879

ParentProcess::GetClientSocketDescriptor: Creating a socket

ParentProcess::GetClientSocketDescriptor: Setting socket options

ParentProcess::GetClientSocketDescriptor: Calling connect
(If you see no messages after this one, VisIt was not able to connect to the client machine. Nine times out of ten, this is a firewall issue on the client machine. It could also mean that VisIt was unable to resolve the IP address for the client machine. You may need to verify the contents of /etc/hosts.)

ParentProcess::GetClientSocketDescriptor: Could not connect! (error=111: Connection refused)

ParentProcess::Connect: Exchanging type representations.
Versions are (),2.6.0

Exception: (IncompatibleVersionException)
/root/visit_2_6_0/visit2.6.0/src/common/comm/CommunicationHeader.C, line 270: <The reason for the exception was not described>
catch(IncompatibleVersionException)
/root/visit_2_6_0/visit2.6.0/src/common/comm/ParentProcess.C:477

Exception: (IncompatibleVersionException)
/root/visit_2_6_0/visit2.6.0/src/common/comm/ParentProcess.C, line 510:
<The reason for the exception was not described>
catch(IncompatibleVersionException)
/root/visit_2_6_0/visit2.6.0/src/engine/main/Engine.C:1354

The engine has a different version than its client.