SSL over NIO in GNU Classpath

Casey Marshall

This is a proposal for a project to be done as a part of Google's Summer of Code 2006, with the Free Software Foundation (the GNU Project) acting as mentoring organization. The proposal is, in brief, to rewrite the SSL implementation included in GNU Classpath (formerly known as the Jessie project) to use non-blocking IO support, and to bring Classpath's SSL support up to that of the J2SE, version 1.5.

About GNU Classpath

GNU Classpath is an ongoing effort to write a free, clean-room implementation of the Java 2 Standard Edition (J2SE) class library, as published by Sun Microsystems as the core of the Java platform, under the aegis of the GNU project. The current effort aims to be 100% compatible with version 1.4 of the J2SE, and to eventually move towards compatibility with 1.5 and later releases. GNU Classpath currently supports a number of free virtual machines, such as Kaffe, Cacao, JamVM, and GCJ to provide real, usable Java environments that can run many popular programs written in the Java language. GNU Classpath is available on the web from `http://www.gnu.org/software/classpath/`.

About the JSSE

The Java Secure Sockets Extension, the JSSE, is a standard library for using the Secure Sockets Layer (SSL) in Java programs, and provides support for writing both network clients and servers that communicate over SSL. One obvious standard use of this is to write web servers and clients that communicate over HTTPS. In release 1.4 of Java, the JSSE was offered as a standard component, instead of as an add-on package; in release 1.5, the library was greatly enhanced to support the non-blocking IO package, java.nio, whereas the 1.4 release only supported blocking IO.

About Jessie

Jessie is a Java library that I wrote, mostly in 2003. It is a clean-room implementation of the JSSE that was included in the J2SE, version 1.4. That is, Jessie included a clean-room implementation of the `javax.net.ssl` API, and included a provider library that implemented SSL version 3 and TLS version 1.0. In early 2006, Jessie was merged into GNU Classpath, and is now maintained there.

Jessie was written with only the 1.4 API in mind, and as such, only supports a blocking IO model. Supporting the NIO model in the 1.5 API is impossible with the current version of Jessie. Since we want to support the 1.5 API in GNU Classpath, Jessie needs to be rewritten to implement the non-blocking model of the 1.5 API.

My Proposal

My proposal is to rewrite the SSL support in GNU Classpath to fully support the 1.5 API. Since the existing code cannot support a non-blocking IO model, it basically must be rewritten from scratch, and the blocking-IO portion rewritten to use the NIO version (it is possible, and quite easy, to support a blocking IO model on top of a non-blocking model, but not the other way around). Some work has been done to accomplish this; in the jessie-nio branch of GNU Classpath, there is some rudimentary code to handle SSL protocol objects with ByteBuffers instead of with streams, but little else.

This is a difficult task. Writing a stateful SSL protocol handler, that can be updated at any time with partial, or multiple, SSL messages will take a lot of effort and out-of-the-box thinking. That is, reading SSL messages sequentially, in a single thread, is easy; reading them when you may only get pieces of them, or many of them at once, per method call requires a completely different approach. I don't think it would be possible for the average, beginning programmer to complete such a task in a few months, especially if they knew nothing about the SSL protocol. I am in the unique position of being well familiar with SSL — having written the blocking-IO version of Jessie — so I do have a great starting point, and do believe that I can complete this task in the time allotted.

I propose that this project will proceed in the following steps:

1. That the SSL library in GNU Classpath be rewritten to use the NIO model of the JSSE (that is, to write an implementation of the SSLEngine class).

2. That the blocking-IO SSL classes in GNU Classpath be rewritten to use the NIO classes.

3. That a complete unit test suite be written, which exercises the library with both partial and complete sample SSL connections. These tests should be integrated into the Mauve test suite (see http://sourceware.org/mauve/).

4. (Optional, if there is time) That we augment a free Java servlet container (such as GNU Gumdrop) as a test-bed for this library, set up such an instance on a public web server, and run some performance and scalability tests on this server.

The main challenge will be to write an efficient and correct SSL implementation given the complexity of writing non-blocking algorithms. The SSL protocol itself should present no challenge, as I already have a good understanding of it.

About Me

I work for Seagate Technology as a staff software engineer in the Branded Solutions division, formerly a part of Mirra, Inc., creator of the award-winning Mirra Personal Server, until we were acquired by Seagate in 2005. I am attending the Master of Science program in Computer Science at the University of California, Santa Cruz, part time. My academic interests are mostly centered around computer security. I have contributed to GNU Classpath in the past, in the area of Classpath's security and cryptography infrastructure. I like music, beer, and hacking, and I live in Santa Cruz, California, in a big, old house.