

Table of Contents

Modifications and version.....	1
Abstract.....	1
Technical details.....	1
Areas of applications.....	2
SQL Databases.....	2
Custom JVM.....	2
Scalable HTTP inspecting proxy that adds security features.	2
Software development as a services.....	2
Development plan.....	2
Resources.....	3
Due dilligence.....	3
References.....	3

Modifications and version

Version	Date	Description of change
0.1	11/20/01	Initial version

About the author

I am Michael Moser, you can contact me at moser.michael@gmail.com or by phone +972544642802 (local phone 0544642802)

Abstract

This project proposes application of a innovative method of implementing high-performance servers. The technique can be easily integrated with existing system software.

The technique makes it possible to port an existing applications written in single threaded or connection per thread process model and turn them into a multiple connections per thread model.

The business proposal is to use this technique in order to augment existing open source system software / middleware software; or to offer porting services to closed source comercial applications.

Several comercial areas of application will be explored in this document.

Technical details

Presently, the EVCOROTHERAD [2] open source library for co-routines has been written in C; this library has been integrated with the libevent – event notification library; As a proof of concept, this library has been used to implement the WEBBY [1] http server.

A co-routine [4] is a cooperatively scheduled user mode thread ; it is similar to the go-routine that is part of the go programming language;

The EVCOROTHREAD library introduces a co-routine thread per connection process model; Each tcp/ip connection is attached to one co-routine thread; One of the design goals of this library was to allow for easy to integration with existing software applications.

Areas of applications

The method can be integrated with existing system software;

This technique can be used to to augment existing open source system software:

SQL Databases

High performance SQL server on the basis of PostgreSQL; presently the open source PostgreSQL database is build as a process per database connection model; this process model is not scalable and therefore prevents this database from being used in an environment that demands high performance; The proposal is to port PostgreSQL to co-routine per database connection model; this of course has to be done as an open source project; The comercial application is to sell support licenses to corporate customers.

As a by product, pursuit of this application area would enable the PostgreSQL database to be seen as a viable choice for comercial customers; There are customers that seek a an SQL server vendor that is not dependend on the whims of big database vendors like Oracle, IBM or Microsoft; this segment of customers has traditionally been the clients of the PostgreSQL product.

Custom JVM

A custom JVM can be created that will introduce the concept of co-routines into the JDK; The open source OpenJDK project can be used as a starting point for this effort;

The project would add additional Java classes that create a co-routine thread; the challenge is to augment/modify existing java IO libraries of the JDK so that they cooperate with the user mode threading library; also multi-threaded locking primitives have to be adjusted to the new threading model.

As in the previous case; this custom JVM can generate revenue from support contracts.

This product adds value by creating additional multipliers – JVM languages different from Java can use this facility to great profit.

Scalable HTTP inspecting proxy that adds security features.

A scalable http proxy that adds deep inspection services ; like parsing of HTTP request / prevention of SQL injections ; application of customs

This path has less novelty value, as there are already a large number of applications that incorporate this functionality.

Software development as a services

It is possible to sell the service of porting existing applications to the new threading library.

Development plan

- Detailed study of available options (one – three months) ; this milestone includes a detailed appraisal of the risk involved and will produce a detailed design;
- (In parallel with the previous task) detailed performance benchmarks of the EVCOROTHEAD library, and WEBBY http server. (about one month)
- Prototype implementation of the solution (three months)
- Stabilising the solution (six months)

The development plan can be implemented by one experienced developer (me) in the time frame of one year;

Resources

The effort would require an estimated 10000-15000\$ in equipment; for test server and network setup.

The effort would require the work of one very experienced programmer (i.e. The modest author of this proposal) during 10 – 12 months; this would cost an estimated 110000\$

Due diligence

The author, Michael Moser has been working as a software developer for the last 20 years; he has accumulated a vast experience in implementing high performance server applications, all levels of networking and low level programming; The author has been part of the development team of several open source projects

The CV of Michael Moser is attached and part of this documentation; references are available upon request.

References

- [1] - WEBBY – an HTTP server with co-routines
<http://mosermichael.github.com/cstuff/all/projects/2011/10/25/http-server-with-coroutines.html>
- [2] - Non blocking IO with co-routines
<http://mosermichael.github.com/cstuff/all/projects/2011/10/09/Non-Blocking-IO-with-coroutines.html>
- [3] - Simple co-routine package in C -
<http://mosermichael.github.com/cstuff/all/projects/2011/10/05/Co-Routines.html>
- [4] – Wikipedia article on co-routines <http://en.wikipedia.org/wiki/Coroutine>