Henrik Swahn, Software Engineering
hesw91@gmail.com/h_swahn@hotmail.com
31 januari 2016

Pthreads and OpenMP, a comparison

Introduction

In this study i want to explore two different approaches to working with threads in C++. At the very low level we have technique/API called POSIX threads[1], at the other end of the spectrum we have openMP which is a very high level technique/API. OpenMP uses compiler directives, library routines and environment variables to work with threads[2]. I want to look into the development effort and compare it to the performance gain of the techniques.

Target area

The main target for this study is the performance of pthreads and openMP. I want to look into the execution speed for programs which uses the threading libraries to implement parallelism. I want to examine the speed up that program gets when implementing parallelism with the libraries and compare them to each other. The last thing i want to examine is the effort for required to develop a program with the libraries and take that into an account. I want to use C++ and Linux as my development and experiment environment.

Motive

When the increase in CPU speed was halted because the power consumption and heat became to high some other technique had to be developed in order to increase performance, multi-core processors was a solution to this[3]. So today almost every system uses a multi-core processor or a CPU with threading support which means that the software has to take advantage of this technique for it to be worth something. For writing software that can take advantage of this C++ is one of the most used language on the market and that is why i think it is very interesting to take a look at two of the available threading models for C++[4].

Because so many systems uses a multicore processor today it's important to utilize parallelism to it's best capabilities and that where i hope to add something.

Value

This is an important topic because many programs are written in C++ and it is important to select a threading model that is appropriate for the goal. To gain understanding of how both the techniques works, how much effort required compared to the performance gain can help programmer make the right choice which in order can improve the development process.

It's also a very interesting to make this comparison because pthread represents a low level approach to threading and openMP represent a high level API to threading. The interesting here is to see that if we can get similar execution time from two very different approaches, or if we get very different result that can also be very interesting.

News worth

When researching this area i came across some studies that brings up just the execution speed of different models and approaches. What i want to add with my study is to also take the development process into an account when looking att the performance. Because this aspect is not researched in the same extend as just execution speed.

Preliminary Goals

- Find interesting algorithms to perform experiments on
- Define research questions
- Gain further understanding how the different models work

References

1. Linux Manual. 2015. PThreads, POSIX Threads Manual. http://man7.org/linux/man-pages/man7/pthreads.7.html
2. OpenMP Architecture Review Board. 2013. OpenMP API Version 4.0. http://www.openmp.org/mp-documents/OpenMP4.0.0.pdf
3. Balaji Venu. Multi-core processors - An overview. http://arxiv.org/pdf/1110.3535.pdf
4. Tiobe Software. 2016. Top programming language of 2015. http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html