Why is Parallel Programming Hard?

Richy Delaney (08479950)

January 31, 2011

Thinking in parallel

Parallels first stumbling block is that the ordinary sequential programming which programmers are so accustomed to and program in every day, becomes a multi-thread concurrent operation, and a different mindset must be adopted. Apart from embarrassingly parallel algorithms, the other popular and often implemented algorithms which don't lend themselves easily to parallelising, must be rewritten with parallel capabilities built in. These sort of rewrites are tricky and error-prone and if a efficient solution is not found the parallel choice may not even produce a performance increase.

Although it is relatively easy to place more and more cores on a chip, The transition from sequential programming to concurrent is one that is very much a paradigm shift, even for seasoned programmers. As there is more than one task being run at once, it makes understanding the flow of algorithms far more complex.

Debugging

Finding problems and bugs in your code when in parallel programming is much harder as code is run concurrently producing indefinite values depending on the how the threads run. Although it is possible to code very defensively when programming in parallel, simple variable collisions and ambiguous behaviour can make it very hard to program without errors.

Depending on the language used to program in parallel, this problem can have varying significance, for example, Threads in Java often are easier to debug and get around problems such as this as they very much help the user not to trip over themselves, while in other more powerful languages, threads are far more dangerous in this regard, in C for example the user must take care of all variables and must account for the state of variables at each stage of an algorithm.

Resources needed

As a programmer, I know the value of good documentation and an active community around a language and how much this helps to write good and effective code in the language as well as to get help when I run into an obstacle. Another big problem space in parallel programming is in the area of resources for learning and for getting help. So much of the help that can be found online is for sequential programming, and problems with parallel programs can be much harder to debug when you have no point of reference.

The fact that this area is a fast moving and very much in development part of programming makes it hard but also makes it vital, as this seems to be the obvious place where computing and programming is headed. Many new and modern languages or extensions to languages (Open MP) try to make parallelising programs as pain-free as possible, but there are still a lot of stumbling blocks.

I think its an area that is difficult due to the huge shift in design and flow of the program you have to write. Although optimally the performance increase can be incredible, often after a lot of development time, the performance increase is negligible or not worth the time.

Sources used for reference in this essay.

http://www.thinkingparallel.com/2007/08/06/what-makes-parallel-programming-hard/

http://blogs.intel.com/research/2007/08/what_makes_parallel_programmin.php