Multi-Core III - a panacea of options

Don MacVittie, 2008-22-09

I find it amazing how many ways we can find to slice a pizza, skin a cat, or solve a technology problem. Truly amazing. Thus far in my multi-core odyssey, I have run into driver-based solutions, library-based solutions, and shim-based solutions to how to maximize use of multi-core. That's not to mention the upcoming version of C++, which is supposed to include the solution in the compiler.

Each of these approaches has its strengths, though I think the best option would be for the OS to resolve the problem directly. It already has a scheduler and mechanisms for semaphores and atomic code. Thought certainly SMP isn't the most optimal solution, it is already in contemporary operating systems and it might be a good first step to extend it to include multi-core.

On my list of people to talk to or products to look at is RogueWave - a company that was near and dear to me ages ago for their libraries, but I haven't heard from in a very long time, and eXludus, a company I had not heard of before a comment to one of the posts in this series. Also on the list is a review of the Cilk++ documentation, which is now available in public beta.

After years of working as a reviewer, I have a tendency to want to compare all of these solutions and give you definitive answers on which is better. I'm not going to do that for several reasons - first is the completely different architectures of some of these solutions, and right after that is the fact that talking about each one's strengths and weaknesses (from a developer perspective and a resource utilization perspective) is more useful to you.

Cilk++, for example, is a library that handles parallelizing the code for you, you just have to make a few calls to make it work from within your code. Excludus claims no changes to source or OS... We don't believe it, something is scheduling CPUs, but we'll give them a chance to prove it. Likely that's market-speak for "we hook the calls and change the processing" which in my book is the same as installing a shim, no?

RogueWave is also kind of sticky about how they handle implementation - they say in their marketure that the Hydra system doesn't require "significant recoding", but the examples in their video all include the phrase "if your developers use Hydra to develop the application..." I guess you're not doing significant recoding if you code it with their stuff from the ground up, heh. Again, I'll talk to them and find out for you.

My reasoning for pursuing this is simple - we can make things faster by pooling your servers to handle heavier load, but the other bit of the equation is making the servers themselves faster, and what impact that has on the network - remember that I tested a box not so long ago (about 2 years) that buried the network cards before maxing out anything else - network was the bottleneck, and when it was removed, disk became the bottleneck. CPU and memory were clipping along without any issues at all. So between our solutions and these solutions, you should be able to uber-optimize your infrastructure... If someone gets multi-core processing right anyway.

I'd also like to point out the article submitted in a comment to my first post in this Multicore series by Dale C. - Some Thoughts on Concurrency it does address the issue of threading pretty well, and talks about the disparity between academia's brainstorming and real-world needs (anyone else remember when ML was going to take over the world? Lori and I were subjected to it during our Masters studies, and the instructor was just gushing with the possibilities (by then ML was 20 years old, but OSS had given it a new lease on life). A fun language to play with, but not much in the productivity department).

There are more out there, I'll keep digging into the topic as long as you all keep telling me it matters to you.

Don.

Share this post :

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc. Corporate Headquarters info@f5.com	F5 Networks Asia-Pacific apacinfo@f5.com	F5 Networks Ltd. Europe/Middle-East/Africa emeainfo@f5.com	F5 Networks Japan K.K. f5j-info@f5.com