Title: **The Future of Quantitative Computing**

Summary: Numerical Algorithms Group (NAG) recently announced they had been collaborating with Advanced Micro Devices (AMD) to produce math libraries for AMD's next generation microprocessor- a 64-bit high performance chip named Opteron™. This paper includes the interview between Jim Finnegan, editor of Financial Engineering News, and Dr. Ian Reid, vice president of business development at NAG to discuss the significance of this announcement.

Finnegan: So tell me more about your relationship with AMD based on their recent announcements?

Reid: We've been working with AMD to develop the basic math libraries for their new 64-bit chips generically termed AMD64. On the one hand, there is a very basic math library all compilers need and it provides the likes of trigonometric functions (sine, cosine etc). What we've done with AMD is to make these functions very high performance and accurate. And let me stress "accurate" in that statement, because what we've found that some people emphasize computational speed over accuracy. They'll come to us and say 'Well, you are slower than Company X on these speed benchmarks.' But we come back and say, 'Well yes, but we get the right answer and they don't. So, how fast do you want the wrong answer?' It's the combination of speed and accuracy that defines for us the most important metric which is computational efficiency.

Finnegan: So is your basic math library to be used on the new AMD chip, is this something that is part of their new chip architecture?

Reid: No. You could think of it as being one level above the hardware or chip -- a software level which you tend to access via the compilers for basic mathematical calculations such as log, sine, cosine, and so on. We've worked with AMD to produce efficient implementations of these. That math library is often called 'libm'.

Finnegan: So this level of software that you describe as "one level above the chip" is created by AMD?

Reid: Correct. Microprocessor manufacturers, be it AMD, Intel or whoever, tend to produce this level of software. In the case of AMD's latest 64-bit chip, they produced it in association with NAG. For the Linux version of libm we produced, it was decided to put the source code (32-bit and 64-bit) into the public domain in keeping with much basic software for the Linux community, enabling the improvements to be shared by the whole community.

Finnegan: This seems to be a dramatic departure for AMD. I've always perceived them to be a follower- some might even say an imitator- of what Intel does with its chip technology. Hasn't that historically been the case?

Reid: Yes, and this is a very important point. What AMD is doing here in producing their new 64-bit (AMD64) chip is to "walk away" from the "Intel standard". The AMD 32-bit chips are all Intel compatible. So, up until now they never needed to build the software stack I've described in this discussion because they could always use what Intel had done. When Intel introduced their 64-bit chips (IA64; Itanium®), they chose to radically change course from their 32-bit chips (IA32) making them incompatible. To use existing code designed for IA32 you either have to rebuild for IA64 (re-compile etc) or use an emulator (which can be slow). What AMD has done is say 'We don't want to go that way as we introduce our 64-bit chip. We have a 32-bit chip platform, and the instruction set on that platform works just as well in the 64-bit world. So, we are going to move into the 64-bit world in such a way that all of your 32-bit software will still work.'

Finnegan: Sounds like a paid endorsement for AMD over Intel.

Reid: Certainly not! I am to some extent repeating AMD's public position on the advantages of their new strategy for 64-bit computing. But the essence of it is true. People will be able to run their software designed for 32-bit chips on the new 64-bit AMD chips. And, in time, when they are ready, they will be able to migrate with software to 64-bit computing so that their software takes full advantage of 64-bit processing. But they won't be forced to make that migration immediately. Of course, Intel has good reasons for thinking their new design is better – time will tell! I should stress that NAG works closely with both Intel and AMD; being vendor neutral is very important to NAG!

Finnegan: But AMD was historically a follower of Intel, which I believe means that their chips employed the "Wintel" standard- in other words a Windows operating system. Yet you say the new AMD chip will use a Linux operating system. Isn't this going to create problems?

Reid: Actually not. Microsoft has announced that Windows is also available for AMD64, so that's not an issue. You have Windows and Linux on the chip just as you do in the Intel world.

Finnegan: But these 64-bit chips, both Intel and AMD, are primarily oriented towards the server market today, not PC's and personal computing. So is this compatibility with 32-bit software that AMD seems to be offering really such a big deal?

Reid: You're right. 64-bit chips are higher priced microprocessors right now, targeted at the server markets. However, both Intel and AMD want to see these 64-bit chips ultimately become the standard in the PC world too. And that's the big leap or risk that AMD is taking. They are moving away from the Intel roadmap which has lines for IA32 and IA64 way into the future. I suspect that AMD will want to move to a single line as soon as possible – and since they claim that AMD64 chips will run 32-bit code faster than existing AMD32 chips then it's only the pricing issue holding them back.

Finnegan: Tell me about the second level of math library NAG has been working on with AMD. You mentioned the libm library as the most basic one, but what about the other?

Reid: The other library is called the AMD Core Math Library, or ACML for short. And this could be considered a level above libm. It contains Basic Linear Algebra Subroutines (BLAS), which cover the basic vector and matrix calculations; LAPACK which cover higher level matrix calculations such as factorizations; and Fast Fourier Transforms (FFTs). These routines are often key to software developers (such as NAG) who develop high level components or packages – it's critical to have this accurate, high performance set of core routines.

Finnegan: Some of my readers at this point may be asking "That's fine but how does this effect finance?" What is NAG's advantage, if any, here?

Reid: Well, in some sense your readers might just expect this infrastructure to be in place and to be 'correct' – hopefully after reading this article, they will have confidence that NAG is working to help ensure that this is indeed the case. But NAG's real additions to finance lie in the higher-level NAG components available in the various NAG libraries. For instance, popular sections for the finance community involve various statistical areas such as GARCH and Principal Component Analysis for data analysis; optimization and linear algebra for portfolio optimization and index tracking; and partial differential equations and random number generators for derivative pricing etc. Of course, many of these higher level routines rely for performance on the excellent implementations of the core math libraries on the various platforms.

Originally published by Financial Engineering News, July/August 2003 (www.fenews.com).

Numerical Algorithms Group
www.nag.com
infodesk@nag.com

