
AMD DEVELOPER INSIDE TRACK

INTRODUCTION TO THE AMD CORE MATH LIBRARY



Get a first hand introduction of the AMD Core Math Library from Chip Freitag, a 17 year veteran, currently working on the ACML Team. He talks about what ACML is, how it differs from ACML-GPU and who could really take advantage of it. He also gives some insight into what the team is working on now to improve ACML and when we will have access to those new features. (7:36)

TRANSCRIPTS

Chip Freitag, SMTS Software Development Engineer on the AMD Core Math Library Team

Sharon: AMD Developer Inside Track is here with Chip Freitag from the ACML team. Chip, why don't you introduce yourself.

Chip: I'm Chip Freitag. I'm currently a project engineer for the AMD Core Math Library. I've been with AMD 17 years. I started back in the 29k days. I've been a marketing engineer and a software developer for much of that time.

Sharon: Wow, another 17 year veteran that is awesome. So, as you mentioned the AMD Core Math Library is the team you're managing now and, I'm just curious, can you tell the audience, what is ACML?

Chip: ACML is a collection of optimized math library routines typically used for scientific and high performance computing. It consists of a BLAS, basic linear algebra sub programs, a LAPAC, a collection of FFTs, random number generators and a small collection of optimized math transcendental functions.

Sharon: Great, and who should be interested in this? Who are your main users?

Chip: Anybody with an AMD computer who's doing any kind of scientific or high performance computing programming would be interested in ACML. If you're developing applications such as finite application analysis, or for oil and gas, or arrow dynamics, or crash simulation, those types of applications, you're probably calling functions that ACML has optimized for AMD platforms.

Sharon: Right, so very high computationally intensive programs, it sounds like.

Chip: That's right.

Sharon: So, I'm also wondering, I found that we have an ACML-GPU version of ACML, can you tell us the differences?

Chip: Sure. ACML-GPU is an offshoot of ACML. It's a separate download. It includes all the same functions that are in ACML and we've optimized key ones of those functions for operation on the GPU. Specifically, we've optimized SGEM and DGEM. We're coming out with a new version soon that will also include ZedGEM and CGEM. Since those are the key drivers of performance for many of the other functions in ACML, for certain applications, most of the other functions in ACML will be able to take advantage of these optimized GPU functions.

Sharon: Ok, and what sort of hardware would they need?

Chip: Currently we support ACML-GPU only with ATI graphics cards. You'll get the best performance when you're running with a higher end graphics card such as a 4870 or the new generation, 5870, card. It also runs on other ATI GPU's as well.

Sharon: Ok, and how about the regular ACML, does that only work on AMD processors?

Chip: The regular ACML runs on AMD processors. It also runs very well on non-AMD processors. We've taken steps to ensure that we've done optimization for other processors as well. So, you should give it a try even if you don't have AMD processors.

Sharon: So, it's safe to say that ACML is a cross compatible platform and actually even cross optimized.

Chip: Yes, and we actually get a lot of feedback from our customers that that's a feature they like to take advantage of.

Sharon: That's always good to know because I know developers often have a hard time knowing, do they do one optimization path for AMD only or do they have to do two if they have to optimize for non-AMD? But, in this case they can use ACML and it will run well on both.

Chip: That's right. And we've made those decisions inside the library so that when it detects that it's running on an AMD processor it will run the AMD kernels and then in other instances it will choose another appropriate kernels.

Sharon: Good to know. So, for someone getting started with ACML, what sort of resources are available?

Chip: So, starting when you download the package, you'll see in addition to the help files and the user guide that are included with the package, we have a set of examples that we include. These are fairly simple examples of how to call various functions with AMD within ACML and how to compile your application to take advantage of ACML and how to link ACML with your application. We provide support for a variety of compilers including gFortran, PGI, Open64, and the Intel I4 compiler in addition to the Microsoft Visual Studio. Other help resources include our online resources, including our ACML forums on (AMD) Dev Central. We have a collection of knowledge base articles. We have an online user guide. And we also have an email support on (AMD) Dev Central and we get those emails directly when you send those email help requests.

Sharon: That's great. So, can you tell me? What's the development team working on now, and what's coming down the line?

Chip: So, we just released ACML 4.4 in late February and since then we've shifted focus to start working on the next generation of AMD processors. We're heads down developing the assembly kernels to support the next generation AMD processors. We'll be working on that for the balance of the year and we'll have a beta release in the, towards the end of the year. In addition we'll be adding some other features such as LAPAC 3.2 and some enhancements to the calling interfaces for FFT's including 2D and 3D real and complex FFT's with some new storage formats for complex FFT's.

Sharon: Excellent. Well thanks so much for your time, Chip. This has been really helpful for my understanding of ACML and hopefully helpful for our developer audience as well.

Chip: Sure.