Heterogeneous System Architecture Programming: Today and Tomorrow
-An exclusive interview with Leendert van Doorn, Corporate Fellow and Corporate Vice President of AMD (Advanced Micro Devices)

By Liu Jiang, Lu Dongxiang

Programmer: Could you please briefly introduce to us about what you do?

Leendert：I’m the Corporate Fellow as well as the Corporate vice president of technology in AMD. Specifically speaking, I’m in charge of putting project innovations into practice. Working in the software division of the chip company and cooperating with such world partners in the software industry as Microsoft and Google, etc, we explain to them what we are undertaking and get to know where they are heading to.

Programmer: AMD produces CPU and GPU at the same time. How is it different from Intel in terms of product development?

Leendert：As we develop GPU, we have powerful vector operation technology, and we develop APU based on the integration of CPU and GPU. APU has powerful graphical performance, but we expect to take more advantage of its computing performance. This will be a strategy of “kill two birds with one stone.” In my previous speeches, I’ve already mentioned some applications such as genes study in biology, chemical composition of molecules, drug design and geographical study and so on. All these need powerful computing capabilities. Some other similar applications, especially the ones in the field of Natural User Interface, are also fascinating. What’s more, these are also the areas in which vector operation processor can be better used.

Programmer: Presently, GPU acceleration technology, which was developed by AMD, has already been used in some programs, such as Photoshop CS6 and GIMP, and so on. My question is, in the future years, except the application in graphics, is GPU capable of being used to accelerate more routine work in the operation system?

Leendert：I believe it will. The reason is that there are also some middleware which can exploit the potential of GPU, for instance, in Big Data processing. What I’d like to see is that GPU could be more than peripheral equipment, but a First-class Entity to operation system, so that it can be managed by the operation system. The operation system will decide whether to allocate a type of thread or a task to GPU or CPU to reach optimum operating efficiency. This is also the true Heterogeneous we hope to realize. Within the operation system, GPU is good at both data compression and encryption, which are the basic functions.

Apple has already added OpenCL in OS X, Google pays more attention to WebCL, while Microsoft supports heterogeneous programming among developing tools. All these are still at an initial stage, and there is still a lot of work to do.

Programmer: Right now, OpenCL only supports C-like programming language. Does AMD have any plans to promote it into other programming languages? For example, Erlang, Scala and Clojure had already supported multiple programming quite well. How can programmers benefit from it?

Leendert： Yes. AMD is positively driving such kind of efforts in many aspects. For instance, on the one hand, it engages in making related standards so that it will be more convenient for other language developers, like developing OpenCL with Fortran; on the other hand, we are also seeking DSL approaches. For example, developers can use Java grammar writing program when they are using Aparapi, and it will convert Java byte codes into the codes executable in GPU when it's running. The languages you mentioned above are what we will consider in the future. However, the priority still depends on commercial factors. Besides, we are also on alert of the motions of developers in different fields.

As for most developers, parallel programming is difficult to acquire. I'm convinced that DSL is the perfect solution to help them in GPU programming, because a higher degree of semantic abstraction is capable of covering the details.

HSA is a system architecture. It's not directed in any specific language, neither does it support only OpenCL, although it has been widely used in the business circle, which is exciting to us. Yet, it's only a subversive experiment which is still at its starting stage. We still have a lot of work to do.

Programmer: Do you think that, apart from programming language, other software related to programming, such as, compiler, will also further support heterogeneous programming in the future?

Leendert：I think C++ AMP is a good example. Microsoft began to support it in Visual Studio 2012. It is a powerful tool for Windows developers. On the other hand, concerning OpenCL, we are cooperating with LLVM to develop and improve some open resources. We are also interested in taking HSA as an open standard.

Programmer: Previously, you mentioned that Microsoft supported C++AMP in Visual Studio 2012. Compared to OpenCL, what are their differences?

Leendert：So to speak, can we make choices between C and Fortran? The answer is no. The reason is that they are oriented to different customers and in different fields. Similarly, if users want to accelerate GPU on the platform of OS X, C++ AMP can do nothing about it; but if they want to program in Windows, it will be a wise choice, as Microsoft will surely continue to invest more in the technology of this.

Programmer: In your speech, you demonstrated three computing stages. The first stage was single core computing, the second, multi-core, and the third, HSA that combines CPU and GPU. After that, what will happen to the new computing technology? How will Amdahl's Law influence heterogeneous computing?

Leendert：This is an interesting topic. We are also studying architectures other than heterogeneous computing. Although GPU is the present highlight of heterogeneous computing, yet heterogeneous doesn’t mean GPU alone. For example, programming OpenCL with Fixed-Function Accelerator or FPGA, are also called dynamic computing or reconfigurable computing. These are possible trends in the future, but still difficult to predict right now. However, I believe that heterogeneous will be the major computing manner in the near five years.

Software companies always hope the programs to run as long as possible without much modification. Setting out from this point, what people concern the most about the codes stored in history is not how to make the program parallel. But things might change dramatically in our future life. For example, as we communicate with the internet, dramatic change may also take place in medical field. In the future, the way people interact with computers, and even the definition they give them might be changed as well. These new applications will provide a turning point for an essential change in computing manner, as the current computing can’t meet people’s needs. This is also what Amdahl’s Law describes--if the executing program in CPU has only 10% of parallel programs, the progress of computing architecture cannot solve the problems; Yet it is still another thing with GPU programming. People know from the beginning what its architecture is good at, and if they can parallelize 99% of the programs from the beginning and upgrade the architecture, things will change totally. This is also one of the reasons why the future is so fascinating.

The problems described in Amdahl’s Law needs to be solved with software. For instance, we cannot parallelize the programs whatever computing architecture we apply if we use Bubble sort, but if we use Merge sort, it will be absolutely different. When programming for GPU, programmers need to consider parallelization right from the beginning.

Programmer: To most developers, it is one thing that a new technology has a certain advantage, what really come to attract them is only when a technology has a Killer Application. What’s the Killer Application in HSA? And what suggestions do you have for developers?

Leendert：I think the Natural User Interface will constitute the Killer Application in HSA. Great changes in interaction modes in the future require computers to have more powerful computing capabilities, in which I think heterogeneous computing will be greatly advantageous. I will suggest all the developers at least to try heterogeneous programming. But people work in different fields and have different problems to solve. As we mentioned before, using domain specific languages or domain specific libraries would be a better choice for most people.

Programmer: Linux system gives people an impression of having a video performance that is not as good as that in Windows, game effect, for instance. Will it influence HSA’s functions of high-end workstation and PC clusters?

Leendert：I think people’s lacking confidence in the graphics performance of Linux or Unix is due to X11, which is a rather out-of-date window system. Compared to it, DirectX of Microsoft can make application program directly interact with display card. A more advanced window system is the key to make people get rid of this impression. Speaking of HSA, Kernel-mode drivers only work in end configuration, and don’t have a decisive influence over the function of it.

Programmer: There is a view that GPU is slowly evolving in the direction of a CPU architecture, where instruction latencies are very low, and caches, prefetching and out-of-order execution. On the other hand, some GPU technology will be merged into the CPU, like AVX2. Is there any possibility that the two will eventually become one uniform processor?

Leendert：It's a good question, and a challenge too. Although it seems that GPU brought in CPU technology, it will by no means come out to be CPU. These technologies are only realized in a very simple way when applied in GPU. That's because with the increase of their complexity, their performance will decline, and all the previous advantages of using GPU will also be gone. What we look forward to achieving is, for the sake of programmers, that GPU and CPU could be more and more compatible. But seeing from the complexity of architecture, GPU can never act like CPU. We are also very cautious when we apply AVX in CPU, that's because on the one hand, developers still needs some time for to actually use the instructions, on the other hand, processor X86 is historically loaded. Even if AVX is brought in, it is still unknown to us whether the acquired performance could be promoted. As for the present products, what we have already known is that we still have a lot of upgrading work to do. Therefore, I think it is more feasible to apply HSA rather than adding GPU technology in CPU. The significance of HSA lies in making all computing resources more available to programmers in a more efficient way.