ADMIN ENTERPRISE HPC

AMD

Exploring AMD's
Ambitious ROCm Initiative

Get Rockin'

ROCm ON

DISCOVERING ROCm

AMD's ROCm platform brings new freedom and portability to the GPU space. By Joe Casad

Three years ago, AMD released the innovative ROCm hardware-accelerated, parallel-computing environment [1] [2]. Since then, the company has continued to refine its bold vision for an open source, multiplatform, high-performance computing (HPC) environment. Over the past three years, ROCm developers have contributed many new features and components to the ROCm open software platform.

ROCm is a universal platform for GPU-accelerated computing. A modular design lets any hardware vendor build drivers that support the ROCm stack [3]. ROCm also integrates multiple programming languages and makes it easy to add support for other languages. ROCm even provides tools for porting vendor-specific CUDA code into a vendor-neutral ROCm format, which makes the massive body of source code written for CUDA available to AMD hardware and other hardware environments.

What is ROCm, and why is it poised to shake up the whole HPC industry? The best way to get familiar is to look inside.

Big Picture

General programming languages like C++ were developed before the birth of GPU-based parallel computing, and the standard forms of the language do not have the features necessary to capitalize on all the benefits of today's high-performance computers. In the past, GPU vendors developed their own dialects and drivers to activate GPU-based optimizations for their own hardware. The result was a tangle of proprietary specs and incompatible languages. The absence of a unifying, open source platform cost untold hours of development time and left coders with few options. Code was written for a single hardware platform, and it required an enormous investment of time and expense to make the code ready for a different environment. This vendor lock-in caused inefficient programming practices and limited the organization's ability to seek a long-term, cost-efficient solution.

The ROCm developers knew the HPC industry needed a universal solution that would end the problem of proprietary specs and incompatible languages, so they built ROCm as a universal open platform that allows the developer to write the code once and compile it for multiple environments. ROCm supports a number of programming languages and is flexible enough to interface with different GPU-based hardware environments (Figure 1).

At the center of the ROCm environment is a technology known as Heterogeneous-Compute Interface for Portability (HIP) [4]. HIP lets you create code that is ready to compile for either the AMD or CUDA/NVIDIA GPU environment.

AMD maintains a special version of the open source Clang compiler for preparing and compiling HIP code. The HIP/Clang compiler is available for free download at GitHub. The Clang developers are currently working on porting the HIP extensions upstream to the mainline

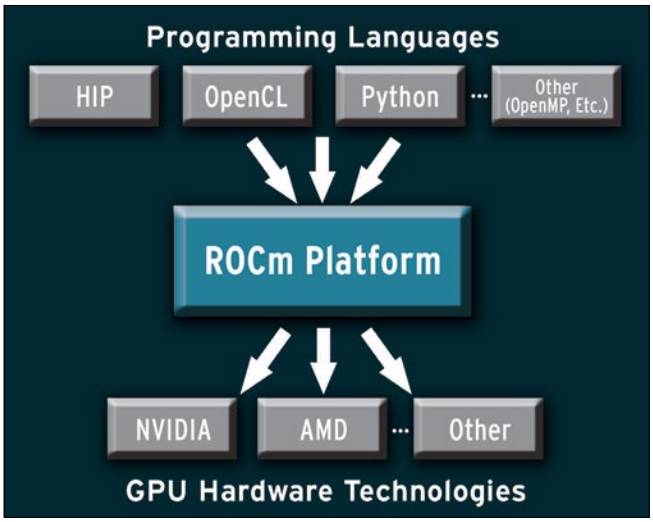

Figure 1: ROCm is designed as a universal platform, supporting multiple languages and GPU technologies.

Clang compiler. Once this upstream effort is complete, a separate HIP/Clang compiler won't be necessary, and HIP will simply be a compiler option within the standard Clang environment.

What About CUDA?

CUDA is an example of a proprietary language designed to work with only one hardware vendor. HIP and the ROCm environment eliminate the need for single-vendor languages like CUDA. However, the ROCm developers are well aware that lots of CUDA code is already out there in the world, so ROCm provides an easy path for porting CUDA code to the vendor-neutral HIP format automatically. Once the code is converted to HIP, you can compile it for AMD hardware using the HIP/Clang compiler. As shown in Figure 2, a CUDA header is all that is needed to prepare the HIP code for the NVIDIA tool chain and the NVCC compiler.

ROCm provides two different alternatives for converting CUDA code to HIP format:

- hipify-perl – a Perl script that you can run on the source code to convert a CUDA program to equivalent HIP code.
- hipify-clang – a preprocessor included with the HIP/Clang compiler that performs the conversion automatically as a standalone preprocessing stage of the compiler process.

The best option for your project will depend upon the details. The Perl script is often easier to use, especially for smaller jobs. The preprocessor gives more extensive hints and error messages and is therefore better suited for large and complex projects.

As a proof of concept, the ROCm team ported the whole Caffe machine-learning framework (with around 55,000 lines of code) from CUDA to HIP: 99.6 percent of the code went unmodified or was automatically converted, and the remaining 0.4 percent took less than a week of developer time to tie up loose ends. The preprocessor can often perform the conversion without *any* cleanup, for some programs; however, the script provides greater flexibility.

The power to integrate previously written CUDA code with the all-open ROCm makes ROCm a truly universal platform. Near-automatic conversion options eliminate time barriers for CUDA shops looking for a more flexible and a less restrictive solution.

Support for Other Languages

The versatile LLVM/Clang compiler infrastructure means ROCm supports a wide range of programmer preferences within the C/C++ language family, from standard C, to standard C++, to STL parallel extensions, to the turbo-charged GPU-based features embodied in C++ AMP. HIP and the HIP conversion options bring CUDA into the mix. Beyond C and C++, the ROCm platform also supports Python Anaconda. Anaconda is a specialized version of Python tailored for scientific computing and large-scale data processing. ROCm also provides native support for the OpenCL (Open Compute Language) framework. OpenCL

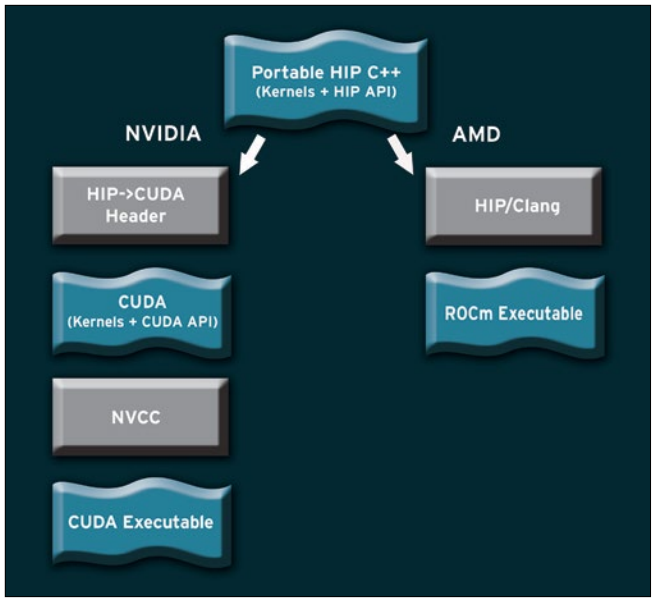

Figure 2: ROCm's HIP format lets the vendor write the code once and compile it for different hardware environments.

is an open standard maintained by the nonprofit Khronos group that was originally envisioned as a heterogeneous framework for supporting CPU- and GPU-based computing in parallel programming environments. In other words, OpenCL has some goals that are very similar to ROCm. AMD is a member of the Khronos group and has invested heavily over the years in OpenCL as a framework for GPU-accelerated programming.

The OpenMP parallel programming API supports offloading to Radeon GPUs through Clang, so developers can access the advanced capabilities of Radeon GPUs from within OpenMP.

HSA-Compliant System Runtime

The foundation for the ROCm environment is the ROCm kernel driver and system runtime stack. The ROCr system runtime API, which resides above the kernel driver, is a language-independent runtime that complies with Heterogeneous System Architecture (HSA) specifications. HSA is an industry standard designed to support the integration of GPUs and CPUs with shared tasks and scheduling.

The modular form of the ROCm system runtime stack means the system runtime could one day support additional programming languages and additional hardware acceleration devices. In the true spirit of heterogeneous computing, the kernel layer below is also implemented as a separate module to facilitate easy porting to other kernel environments.

ROCm also supports a number of powerful APIs and libraries to optimize performance for GPU-based HPC scenarios. For instance, the RCCL (pronounced "rickle") collective library is a powerful tool for supporting multiple GPUs on a single node in single- and multi-process operations, as well as extending the environment to include multi-node communication.

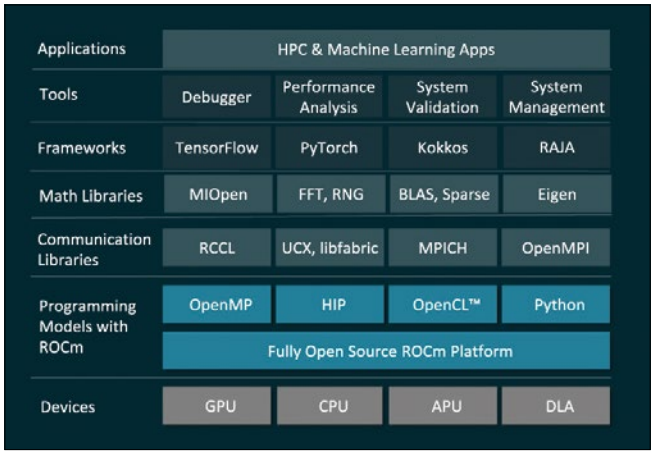

Figure 3: The versatile ROCm provides support for several popular libraries and machine-learning frameworks.

Machine Learning and AI

ROCm is built to accommodate future technologies, and the future is machine learning and artificial intelligence. Many of the high-performance computer systems that depend on AMD's GPU-based computing environment are used with AI research and development, and ROCm is designed from the ground up as a versatile and complete platform for machine learning and AI (Figure 3).

Recent versions of the TensorFlow and PyTorch machine-learning frameworks provide native ROCm support. ROCm also supports the MIOpen machine-learning library [8]. MIOpen serves as a middleware layer between AI frameworks and the ROCm platform, supporting both OpenCL and HIP-based programming environments. The latest version of MIOpen includes optimizations of new workloads for Recurrent Neural Networks and Reinforcement Learning, as well as Convolution Neural Network acceleration.

The latest version of ROCm adds enhanced support for distributed training and the deep learning model. Many example codes have already been validated on AMD Radeon Instinct GPUs for TensorFlow on ROCm.

Containers and ROCm

A recent trend in HPC is increased reliance on containers. Containers are easy to extend and adapt to specific situations, and a containerized solution is an efficient option for many HPC configurations and workloads. The ROCm developers have continued to improve and expand ROCm support for container technologies, such as Docker and Singularity.

The latest version of ROCm includes support for Kubernetes, Slurm, OpenShift, and other important tools for managing and deploying container environments. Also, the *ROCm-docker* repository contains a framework for building the ROCm software layers into portable Docker container images. If you work within a containerized environment, ROCm's Docker tools will allow you to integrate ROCm easily into your existing organizational structure.

Other Tools

The ROCm ecosystem is envisioned as a complete open development environment that includes a comprehensive toolkit of developer utilities. ROCm comes with a collection of debugging tools, including a HIP debugger and ROCm-GDB, a version of the GDB debugger modified for the ROCm platform. The ROC Profiler and ROC Tracer utilities provide performance analysis for programs writ-

ten in C/C++, Python, and Fortran. ROCm also supports the Tau performance system, a portable profiling and tracing toolkit for performance analysis of parallel programs written in Fortran, C, C++, UPC, Java, and Python. Beyond the Tau tools, which are available now, AMD continues to work on expanding support for other tools and performance profilers for large systems, such as PAPI and the HPCToolkit, which will be openly available in the future.

A system management interface (ROCm-SMI) supports a number of functions related to system time and temperature settings. In GPU environments, clock speed is an important consideration, and AMD GPUs can operate at a variety of different clock levels to optimize speed and energy usage. As with all high-performance environments, the clock speed has an effect on energy use, which has an effect on the temperature of the system. ROCm-SMI has options for measuring temperature, controlling voltage, and managing the fan speed. You can integrate the commands of the system management interface into programs and scripts to build speed and temperature controls directly into the programming environment. See the ROCm documentation for more information on ROCm management and development tools.

New Generation

Free software happens in communities, and the free ROCm platform has already unleashed a flurry of community development. GitHub is home to a number of open source projects that extend and expand the ROCm ecosystem for HPC, including the NWChem computational chemistry toolkit, as well as the LAMMPS, NAMD, and Gromacs molecular dynamics simulators.

The latest ROCm update occurs as AMD continues to build on its new generation of hardware for machine learning and HPC. The new Vega 7nm technology-based product line includes the Radeon Instinct™ MI50 16GB and 32GB GPUs [9], which operate at 26.8 (FP16), 13.3 (FP32), and 6.6 (FP64) TFLOPS peak performance, have up to 1 TB/s memory bandwidth and are, according to AMD, the world's first PCIe® Gen 4 accelerators. The new Infinity Fabric™ link technology delivers up to 184 GB/s peer-to-peer bandwidth – up to 4.75 times faster than PCIe 3.0 alone. These hardware innovations let you group GPUs together into "hives," which could further boost performance for some configuration scenarios.

The latest ROCm release is designed to exploit the powerful possibilities of AMD's advanced GPU-based HPC products, with built-in switches and optimizations that will bring this next-generation GPU hardware to its fullest potential. ROCm 3.0 supports the new 2nd Gen AMD EPYC™ processor series [10], which comes with up to 64 cores and 128 threads and has broken the barrier on CPU performance with 100 performance world records [11]. The new ROCm release also includes support for the Bfloat16 floating-point math format.

Conclusion

AMD's ROCm platform is a bold step toward portability and heterogeneous computing in the HPC space. AMD's GPU product line now has an equivalent to the benefits available with NVIDIA's GPUs through the CUDA framework, but ROCm goes a step further by creating a complete language- and hardware-independent path for GPU-accelerated programming. A developer can write the code once and then compile it for either the CUDA/NVIDIA or the ROCm/AMD environment. Upstream Support for ROCm-enabled GPUs in machine-learning frameworks like TensorFlow and PyTorch/Caffe2 ensures immediate relevance for projects that depend on these tools.

AMD's vision for a GPU-based, all-open software programming stack is disrupting the whole HPC industry. The open and modular architecture means other vendors can integrate their own technologies into the ROCm stack, and the easy path for porting existing languages and frameworks to ROCm's neutral format will simplify the learning curve for programmers who want to stay within their preferred coding environment. ■

Resources

ROCm:

[1] [rocm.github.io]

[2] [amd.com/ROCm]

[3] ROCm documentation: [http://rocm-documentation.readthedocs.io/]

[4] HIP: [https://github.com/ROCm-Developer-Tools/HIP]

HPC:

[5] [amd.com/ROCm/HPC]

[6] [amd.com/HPC]

Machine Learning:

[7] [amd.com/ROCm/ML]

[8] MIOpen: [https://rocmsoftwareplatform.github.io/MIOpen/doc/html/index.html]

Products:

[9] Radeon Instinct GPUs: [amd.com/INSTINCT]

[10] 2nd Gen EPYC CPUs: [amd.com/EPYC]

[11] [amd.com/WorldRecords]