Supercomputing Articles

By *Roy Schestowitz*
Created *19/11/2019 - 3:50am*
Submitted by Roy Schestowitz on Tuesday 19th of November 2019 03:50:58 AM Filed under Server [1]

- **Exascale meets hyperscale: How high-performance computing is transitioning to cloud-like environments [2]**

  Twice a year the high-performance computing (HPC) community anxiously awaits the announcement of the latest edition of the Top500 list, cataloging the most powerful computers on the planet. The excitement of a supercomputer breaking the coveted exascale barrier and moving into the top position typically overshadows the question of which country will hold the record. As it turned out, the top 10 systems on the November 2019 Top500 list are unchanged from the previous revision with Summit and Sierra still holding #1 and #2 positions, respectively. Despite the natural uncertainty around the composition of the Top500 list, there is little doubt about software technologies that are helping to reshape the HPC landscape. Starting at the International Supercomputing conference earlier this year, one of the technologies leading this charge is containerization, lending further credence to how traditional enterprise technologies are influencing the next generation of supercomputing applications.

  Containers are borne out of Linux, the operating system underpinning Top500 systems. Because of that, the adoption of container technologies has gained momentum and many supercomputing sites already have some portion of their workflows containerized. As more supercomputers are being used to run artificial intelligence (AI) and machine learning (ML) applications to solve complex problems in science-- including disciplines like astrophysics, materials science, systems biology, weather modeling and cancer research, the focus of the research is transitioning from using purely computational methods to AI-accelerated approaches. This often requires the repackaging of applications and restaging the data for easier consumption, where containerized deployments are becoming more and more important.

- **Exploring AMD?s Ambitious ROCm Initiative [3]**

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.

- **High-Performance Python ? GPUs [4]**

When GPUs became available, C code via CUDA, a parallel computing platform and programming model developed by Nvidia for GPUs, was the logical language of choice. Since then, Python has become the tool of choice for machine learning, deep learning, and, to some degree, scientific code in general.

Not long after the release of CUDA, the Python world quickly created tools for use with GPUs. As with new technologies, a plethora of tools emerged to integrate Python with GPUs. For some time, the tools and libraries were adequate, but soon they started to show their age. The biggest problem was incompatibility.

If you used a tool to write code for the GPU, no other tools could read or use the data on the GPU. After making computations on the GPU with one tool, the data had to be copied back to the CPU. Then a second tool had to copy the data from the CPU to the GPU before commencing its computations. The data movement between the CPU and the GPU really affected overall performance. However, these tools and libraries allowed people to write functions that worked with Python.

In this article, I discuss the Python GPU tools that are being actively developed and, more importantly, likely to interoperate. Some tools don?t need to know CUDA for GPU code, and other tools do need to know CUDA for custom Python kernels.

- **Porting CUDA to HIP [5]**

You?ve invested money and time in writing GPU-optimized software with CUDA, and you?re wondering if your efforts will have a life beyond the narrow, proprietary hardware environment supported by the CUDA language.

Welcome to the world of HIP, the HPC-ready universal language at the core of AMD?s all-open ROCm platform [1]. You can use HIP to write code once and compile it for either the Nvidia or AMD hardware environment. HIP is the native format for AMD?s ROCm platform, and you can compile it seamlessly using the open source HIP/?Clang compiler. Just add CUDA header files, and you can also build the program with CUDA and the NVCC compiler stack (Figure 1).

- **OpenMP ? Coding Habits and GPUs [6]**

  When first using a new programming tool or programming language, it?s always good to develop some good general habits. Everyone who codes with OpenMP directives develops their own habits ? some good and some perhaps not so good. As this three-part OpenMP series finishes, I highlight best practices from the previous articles that can lead to good habits.

  Enamored with new things, especially those that drive performance and scalability, I can?t resist throwing a couple more new directives and clauses into the mix. After covering these new directives and clauses, I will briefly discuss OpenMP and GPUs. This pairing is fairly recent, and compilers are still catching up to the newer OpenMP standards, but it is important for you to understand that you can run OpenMP code on targeted offload devices (e.g., GPUs).

- **News and views on the GPU revolution in HPC and Big Data: [7]**

  Exploring AMD's Ambitious ROCm Initiative
  Porting CUDA to HIP
  Python with GPUs
  OpenMP ? Coding Habits and GPUs

Server

Source URL: http://www.tuxmachines.org/node/130647

Links:
[1] http://www.tuxmachines.org/taxonomy/term/147
[2] https://www.redhat.com/en/blog/exascale-meets-hyperscale-how-high-performance-computing-transitioning-cloud-environments-red-hat
[3] http://www.admin-magazine.com/HPC/Articles/Discovering-ROCm/(language)/eng-US
[4] http://www.admin-magazine.com/HPC/Articles/High-Performance-Python-3
[5] http://www.admin-magazine.com/HPC/Articles/Porting-CUDA-to-HIP
[6] http://www.admin-magazine.com/HPC/Articles/OpenMP-Coding-Habits-and-GPUs
[7] http://www.linux-magazine.com/Online/News/News-and-views-on-the-GPU-revolution-in-HPC-and-Big-Data2