Real-Time Pulsars Pipeline Using Many-Cores

Authors: Alessio Sclocco (Vrije Universiteit Amsterdam, Netherlands); Rob V. van Nieuwpoort (Netherlands eScience Center, Netherlands); Henri E. Bal (Vrije Universiteit Amsterdam, Netherlands)

Published in:

Exascale Radio Astronomy, AAS Topical Conference Series Vol. 2. Proceedings of the conference held 30 March - 4 April, 2014 in Monterey, California. Bulletin of the American Astronomical Society, Vol. 46, #3, #203.01

Publication Date: 04/2014

Abstract

Exascale radio astronomy presents challenges to both astronomers and computer scientists. One of these challenges is processing the enormous amount of data that will be produced by exascale instruments, like the Square Kilometer Array (SKA). Traditional approaches, based on storing data to process them off-line, are common nowadays, but are unfeasible in the exascale era due to the high bandwidths. We investigate the use of many-core accelerators as a way to achieve real-time performance without exceeding cost and power constraints. In our current research, we aim at accelerating the pulsar searching process, and produce a real-time and scalable software pipeline for the exascale era. Our pipeline consists of three main steps: dedispersion, folding and signal-to-noise ratio computation. It is open source and implemented using the Open Computing Language (OpenCL). To achieve our goals of real-time performance, scalability and portability, we applied three different techniques. First, we designed all steps of the pulsars pipeline to run on many-core accelerators, even the less computational intensive. This way, communication between host and accelerator is minimized, avoiding a common bottleneck of many-core accelerated computing. Second, we parallelized the pipeline with a fine-grained approach. Because of this parallelization strategy, it is not only possible to distribute the input beams to different computation nodes, but also to define which part of the search space is explored by any node. This completely avoids inter-node communication, and scalability of the pipeline can simply be achieved by adding more machines. Third, we use extensive auto-tuning for both the single processing kernels and the pipeline as a whole. By using auto-tuning, we do not simply find the best possible parameter configuration, thus obtaining high-performance, but also make the pipeline portable among different computing devices, and adaptable to different telescopes and observational setups. We tested our pipeline with different accelerators, like GPUs from AMD and NVIDIA and the Intel Xeon Phi, and were able to obtain real-time performance in different operational scenarios for different telescopes.