Accelerated processing for modern radio telescopes using a combination of KNL and FPGA

Sajish Chandrababu, Yogindra S. Abhyankar, Sarun O. S.
Centre for Development of Advanced Computing
Pune - 411 008
India
sajishc@cdac.in,yogindra@cdac.in,sarunn@cdac.in

C. R. Subrahmanya
Raman Research Institute
Bengaluru - 560 080
India
crs@rri.res.in

ABSTRACT

The Ooty Radio Telescope (ORT) is a cylindrical paraboloid of reflecting surface, 530 m long and 30 m wide. It operates at a frequency of 326.5 MHz with a maximum bandwidth of 15 MHz at the front-end. The ORT was commissioned long back but continues to be one of the sensitive radio telescopes in the world. There is a plan to expand the antennas (1056 dipoles along the feed array) of this telescope in order to increase the front-end bandwidth. Computational setup of the order of petaflop will be required just for the software correlation of this planned system.

A common feature of any modern radio telescopes is the several hundreds of antenna elements and wide operational bandwidths from several hundred MHz to a couple of GHz. The real-time signal processing involves fusing the responses of these elements in many different ways, both spatially and temporally. The most common fusion operation is the cross-correlation of responses from every element with itself and with every other element. For a variety of reasons, many observational characteristics as well as instrumental response are frequency dependent. Hence, the broad band responses of individual elements are channelized into many subbands and fusion operations are performed independently for each subband. From a computational point of view, this leads to an attractive possibility of distributed processing using a Single Program Multiple Data (SPMD) algorithm. For realization of real time processing using a High Performance Computing (HPC) cluster, a convenient option is to use the coarse level subbands for allocating load to different nodes of the cluster, while the fine grain channelization can be used to distribute the load among different threads on a given node.

Due to the nature of the cross-correlation operation and to maximize the power efficiency, it makes sense to use many core processors like the latest Intel Knights Landing for doing the software correlation. The Knights Landing (KNL) processor has 72 cores each, with two 512-bit vector processing units and each core capable of running four threads on it. Each thread within the KNL processor performs cross-correlation of the entire array for a single spectral channel leading to simultaneous processing of as many spectral channels as the number of available hardware threads in the KNL processor.

The biggest challenge in having such a software correlator on a HPC cluster with a number of HPC nodes is the communication bandwidth required for all-to-all connectivity. This interconnectivity is required as at any given time, signals from all the antennas should be forwarded to a single HPC node. When the signals from all the antennas are to be forwarded to a single HPC node, the standard Ethernet switch port connected to the HPC node becomes the bottleneck. To meet the computational requirement in an efficient way KNL processor is used for software correlation; however this in turn further increases the communication bottleneck making the utilization of KNL less efficient. In order to overcome the bottleneck, in this work we are using FPGA boards with designed functionality to segregate the communication between the ORT antennas and the HPC nodes. All the FPGA boards are connected to the switch and an index based approach is used for doing communication between all the FPGA boards to all the HPC nodes.

The digitized data from the antenna is forwarded to the FPGA board using JESD204B serial protocol. Each FPGA board will support four JESD204B serial interfaces each working at 5 Gbps. FPGA board uses a large on-board memory to buffer the incoming data. This data corresponding to a number of subbands are first stored in the memory and then the stored data corresponding to a subband is transferred to a HPC node. After transferring data to the HPC node, the data for the next node is transferred and same goes on sequentially for all the nodes. To transfer the data to HPC nodes, each FPGA board will have four 10 Gigabit Ethernet interfaces. To avoid switch level congestion, different FPGAs are sequenced in such a way, that no two FPGA boards are sending data to the same HPC node at the same time. This is achieved by making sure that each FPGA board uses a different index-value, to decide from which of the HPC node it should start the sequencing.

One of the major barriers for using an FPGA is simply that these devices are difficult to program. Traditional FPGA based design via hand coding may take few months to develop. Recently FPGA manufactures have started supporting high level languages like C, OpenCL and other languages, to reduce this time. We are using Xilinx HLS tool for most of our designs to reduce the development time. An added advantage of this is that the researchers having no knowledge of HDL can also do modifications in the design to perform different experimentations.

By addressing the communication bottleneck, our proposed system can efficiently use KNL and the combination of KNL and FPGA boards will be able to meet the computing requirements of the expanded ORT. In the next phase of our work, we are implementing data pre-processing part inside the FPGA.