Processing LOFAR Telescope Data in Real Time on a Blue Gene/P Supercomputer

John W. Romein, *Jan David Mol*, *Rob V. van Nieuwpoort*, and *P. Chris Broekema*
Stichting ASTRON (Netherlands Foundation for Research in Astronomy)
Oude Hoogeveensedijk 4, 7991 PD Dwingeloo, The Netherlands
{romein,mol,nieuwpoort,broekema}@astron.nl

Abstract

This paper gives an overview of the LOFAR correlator. Unlike traditional telescopes, the correlator is implemented in *software*, yielding a very flexible and reconfigurable instrument. The term "correlator" understates its capabilities: it filters, corrects, coherently or incoherently beam forms, dedisperses, and transforms the data as well. It supports several observation modes, even simultaneously. The high data rates and processing requirements compel the use of a supercomputer; we use a Blue Gene/P. The software is highly optimized and achieves extremely good computational performance and bandwidths, increasing the performance of the *entire* LOFAR telescope.

1 Introduction

LOFAR, an acronym for ***LO****w* ***F****requency* ***AR****ray*, is the first of a new generation of aperture array radio telescopes [1, 2]. Rather than using expensive dishes, it forms a distributed sensor network that combines the signals from tens of thousands of simple antennas. Its revolutionary design allows observations in the 10–250 MHz range, a range that has hardly been studied before. Another novel feature of LOFAR is the elaborate use of *software* to process data, where traditional telescopes use customized hardware. This dramatically increases flexibility and substantially reduces costs, but the high processing and bandwidth requirements compel the use of a supercomputer.

The LOFAR antennas are grouped in *stations*, and their signals are digitized, filtered, and beam-formed in the field, using FPGAs [3]. The FPGAs send their output over wide-area networks to the central supercomputer, that combines the data in multiple ways, in real time. We support several observation modes (e.g., imaging, known pulsar, unknown pulsar, cosmic rays), each with its own processing pipeline. The data products are stored on disk for further processing after the observation has finished, depending on the observation mode. For example, in the imaging mode, the data is then disposed of RFI, calibrated, and imaged on a PC cluster [4], while in the unknown pulsar mode, data is dedispersed and Fourier transformed, to find repeating peaks [5].

This paper gives an overview of the real time, central processing of LOFAR station data (more details can be found elsewhere [6, 7, 8]). For this, we use an IBM Blue Gene/P (BG/P) [9], an energy-efficient supercomputer with good support for operations on complex numbers, necessary for efficient signal processing. Specialized types of internal networks support different types of collective operations at hundreds of gigabits of bisectional bandwidth. The system consists of compute nodes that provide processing power and I/O nodes that transparently perform external I/O operations for the compute nodes. Although I/O nodes are not supposed to be programmed by end users, we recognized that running part of the application there would greatly improve performance and flexibility [10], and we do some simple, I/O-intensive tasks there, as explained below.

The software supports independent, simultaneous observations by partitioning the stations or the observation bandwidth. In addition, the software supports *piggy-back observations*, where multiple pipelines process the same input data, e.g., the data is correlated in an imaging observation, while the same data is beam formed to search for unknown pulsars. Also, the software is extremely optimized. The net result is that the observation bandwidth was increased from 32 MHz to 48 MHz (the maximum that the station FPGAs can produce), improving the efficiency of the *entire* telescope by 50%, while using only half the planned amount of computational resources.

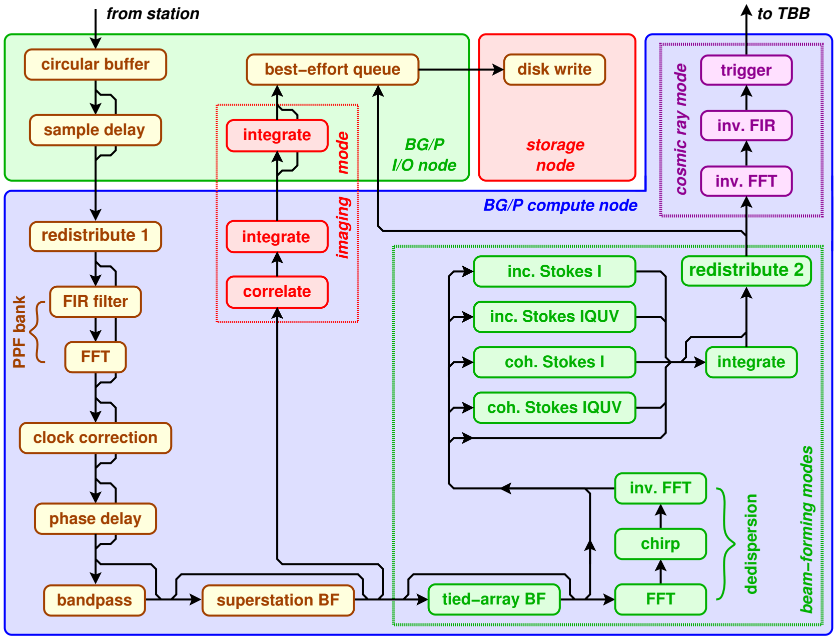

Figure 1: Overview of the pipeline components.

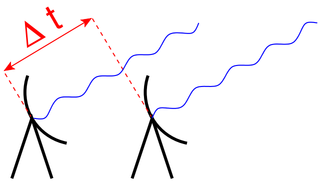

Figure 2: The left antenna receives the wave later.

2 Signal processing on the Blue Gene/P

Figure 1 shows an overview of the pipelines. Most pipeline components run on the BG/P compute nodes, but some run on the BG/P I/O nodes or on external storage nodes. The figure identifies groups of components for specific observation modes, such as the imaging mode, the beam-forming (pulsar) modes, and cosmic ray mode. We briefly describe the pipeline components below.

The I/O nodes receive up to 200 Gb/s of data from the LOFAR stations. The data is stored in circular buffers for about three seconds. The buffers are used to compensate for differences in travel times over the wide-area network links, to handle small hiccups in the remainder of the processing pipelines, and to do *delay compensation*. The latter is done to track the observed source, of which the wave front hits the stations at different times (see Figure 2). To align the wave fronts, we delay the data stream from each station by an integer amount of samples (the remaining fraction is corrected later). This delay depends on the position of the source with respect to the stations, and continuously changes due to earth rotation. Unlike single-pixel feed, dish-based telescopes, we can observe multiple sources simultaneously.

The I/O nodes send their data to the compute nodes. Each input stream contains all subbands from a single station, but to combine the data later, a compute node needs a single subband from all stations. Hence, we redistribute the data across the compute nodes using the fast three-dimensional torus network inside the BG/P. Then, a *PolyPhase Filter* (PPF) bank splits each 195 KHz subband into 16–4096 channels. Next, the phases of the signals are corrected for differences in the lengths of the cables that provide a shared clock to the central core stations. Then, the remaining fraction of the delay compensation is applied by another frequency-dependent phase rotation. We do this after the PPF bank, since we have better frequency resolution here. Subsequently, we apply a *bandpass* correction: we flatten a ripple that is caused by the first PPF bank at the LOFAR stations [11]. Next, the *superstation beam former* optionally combines the data from multiple stations; this will typically be used in long-baseline observations to combine some of the closely-located core stations prior to correlation, significantly reducing the output data rate.

From that point, the pipelines divert, possibly running simultaneously. The imaging pipeline correlates and integrates the data [8], after which the resulting visibilities are sent to the I/O nodes. After possible extra integration, the data is stored in a best-effort buffer, that tries to send the data to the storage nodes. If this fails (e.g., due to disk or network failure) it drops the data that cannot be written, to prevent the correlator from blocking entirely and losing *all* data. Normally, the visibility data is written to disk, and will be calibrated and imaged afterwards.

Additionally, the *beam-forming* modes are a collection of pipelines that are used for pulsar and cosmic ray observations [6]. The *tied-array beam former* creates tens or hundreds of narrow beams within the station beam; we are the first to support this. The resulting complex voltages can optionally be *dedispersed* coherently, i.e. within channels, by a forward FFT that creates 3 Hz subchannels, applying a chirp function that corrects the phases and doing a backward FFT. This allows us to observe millisecond pulsars at low frequencies. Then, coherently beam-formed data is converted to Stokes IQUV, Stokes I only, or not converted at all. Incoherently beam-formed data can also be output as Stokes IQUV or Stokes I only. Optionally, data can be integrated to reduce the output data rate. Then, the data is redistributed again across the three-dimensional torus network: each input data stream contains a few channels from all beams, polarizations, and Stokes components, while each output stream contains all frequencies of a single beam, polarization, and Stokes component. The data is then sent to the I/O node and stored in the best-effort buffer, or used as input of the cosmic ray processing pipeline.

The cosmic ray pipeline is currently in development. A backward FFT and inverse PPF undoes the PPF bank at the stations, to regain the 5 ns. time-series data. A trigger algorithm will try to distinguish transient events from RFI (a spike in all beams is generally caused by RFI while a spike in a few adjacent beams is typically caused by a transient event). The trigger algorithm then freezes the *Transient Buffer Boards* (TBB) at the stations, that hold the digitized, unprocessed samples from the individual dipoles or tiles for about four seconds. The contents of the TBBs are then frozen and dumped for accurate analysis afterwards.

The application is highly optimized. Most compute-intensive kernels are written in assembly code (the low-level, native computer language), and are often an order of magnitude faster than equivalent C++ code. For example, the correlations are computed at 96% of the floating-point peak performance [8]. In the imaging mode, the computationally most-challenging components are the correlator and the PPF bank, responsible for up to 71% resp. 20% of the system load [8]. In the coherent beam-forming mode, computing the complex voltages and computing Stokes values are the most time-intensive tasks, depending on the number of stations used and the number of beams created [6]. We also optimized for high I/O bandwidths, and developed a new network protocol for data transport between the I/O nodes and compute nodes, that is three times as fast as the standard system software [7]. Also, part of the operating system was modified to work around a limitation in the memory-management unit of the hardware [12].

The scheduling of work across the compute nodes is very complicated. This is partially due to the slow (but energy-efficient) processor cores that need up to sixteen seconds to process one second of subband data (hence multiple cores are needed to process one subband of data), partially due to constraints in the network topology between I/O nodes and compute nodes, and partially to obtain high bandwidths between the compute nodes [8].

3 Conclusions and future work

The elaborate use of *software* turns LOFAR into a very flexible and reconfigurable radio telescope. We use a Blue Gene/P supercomputer to correlate, filter, beam form, and dedisperse data. Multiple pipelines can run simultaneously on the same data, allowing multiple types of observations at the same time. Due to the highly optimized program code, we can observe at 50% more bandwidth than the original specifications required, increasing the performance of the entire telescope.

Future work focusses on even more functionality, and includes a trigger algorithm for the cosmic ray pipeline. Also, we want to dedisperse beams at many dispersion measures at the same time, to find fastrotating pulsars at low frequencies. Both extensions are computationally challenging and require many (inverse) FFTs to be done: the cosmic ray pipeline on roughly fifty beams and the pulsar search pipeline on as many dispersion measures as can be tried simultaneously. Thanks to the flexibility of a software approach, we can add new functionality with modest effort.

References

[1] H.R. Butcher. LOFAR: First of a New Generation of Radio Telescopes. *Proceedings of the SPIE*, 5489:537–544, October 2004.

[2] M. de Vos, A.W. Gunst, and R. Nijboer. The LOFAR Telescope: System Architecture and Signal Processing. *Proceedings of the IEEE*, 97(8):1431–1437, August 2009.

[3] A.W. Gunst and M.J. Bentum. Signal Processing Aspects of the Low Frequency Array. In *IEEE International Conference on Signal Processing and Communications*, pages 600–603, Dubai, United Arab Emirates, November 2007.

[4] R. J. Nijboer and J. E. Noordam. LOFAR Calibration. In R. A. Shaw, F. Hill, and D. J. Bell, editors, *Astronomical Data Analysis Software and Systems (ADASS XVII)*, number 376 in ASP Conference Series, pages 237–240, Kensington, UK, September 2007.

[5] B. Stappers et. al. Observing Pulsars and Fast Transients with LOFAR, 2011. Under review.

[6] J. D. Mol and J. W. Romein. The LOFAR Beam Former: Implementation and Performance Analysis, 2011. Under review.

[7] J.W. Romein. FCNP: Fast I/O on the Blue Gene/P. In *Parallel and Distributed Processing Techniques and Applications (PDPTA'09)*, pages 225–231, Las Vegas, NV, July 2009.

[8] J.W. Romein, P.C. Broekema, J.D. Mol, and R.V. van Nieuwpoort. The LOFAR Correlator: Implementation and Performance Analysis. In *ACM SIGPLAN Symposium on Principles and Practice on Parallel Programming (PPoPP'10)*, pages 169–178, Bangalore, India, January 2010.

[9] IBM Blue Gene team. Overview of the IBM Blue Gene/P project. *IBM Journal of Research and Development*, 52(1/2):199–220, January/March 2008.

[10] K. Iskra, J.W. Romein, K. Yoshii, and P. Beckman. ZOID: I/O-Forwarding Infrastructure for Petascale Architectures. In *ACM SIGPLAN Symposium on Principles and Practice on Parallel Programming (PPoPP'08)*, pages 153–162, Salt Lake City, UT, February 2008.

[11] J.W. Romein. Bandpass Correction in LOFAR. Technical report, ASTRON, August 2008. http://www.astron.nl/∼romein/papers/BandPass/bandpass.pdf.

[12] K. Yoshii, K. Iskra, H. Naik, P. Beckman, and P.C. Broekema. Performance and Scalability Evaluation of "Big Memory" on Blue Gene Linux. *International Journal of High Performance Computing Applications*. To appear.