PROCEEDINGS OF SCIENCE

A transient detection and monitoring pipeline for LOFAR

John Swinbank[a], This Coenen[a], Casey Law[a], Joseph Masters[a], James Miller-Jones[b], Bart Scheers[a], Hanno Spreeuw[a], Ben Stappers[c], Ralph Wijers[a] and Michael Wise[ad] on behalf of the LOFAR Transients Key Project

[a] *Astronomical Institute 'Anton Pannekoek', University of Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, The Netherlands*

[b] *National Radio Astronomy Observatory, 520 Edgemont Road, Charlottesville, VA 22903, USA*

[c] *Jodrell Bank Centre for Astrophysics, University of Manchester, Manchester M13 9PL, UK*

[d] *Stichting ASTRON, Postbus 2, 7990 AA Dwingeloo, The Netherlands*

E-mail: swinbank@science.uva.nl, tcoenen@science.uva.nl, claw@science.uva.nl, jmasters@science.uva.nl, jmiller@nrao.edu, bscheers@science.uva.nl, hspreeuw@science.uva.nl, Ben.Stappers@manchester.ac.uk, rwijers@science.uva.nl, wise@science.uva.nl

When LOFAR, the LOw Frequency ARray, comes online over the next few years, it will provide an unprecedented sensitivity and field of view in the low-frequency radio regime. The Transients Key Project aims to exploit these capabilities by constructing an automatic system for monitoring the sky for transient sources with LOFAR. Such a project requires automatic processing and archiving of a huge volume of data, presenting a software design and engineering challenge. Here, we present an overview of some of the technical issues faced, as well as outlining the approaches adopted to address them.

Bursts, Pulses and Flickering: Wide-field monitoring of the dynamic radio sky
June 12-15 2007
Kerastari, Tripolis, Greece

PoS(Dynamic2007)044

1. Introduction

LOFAR, the LOw Frequency ARray, is a new "software" radio telescope currently under construction in the Netherlands and elsewhere. Operating between 30 and 240 MHz (split into "low" 30–80 MHz and "high" 120-240 MHz bands), the instrument is comprised of a large number of individual dipoles, each sensitive to the entire sky above it. By dividing up the total bandwidth and electronically inserting delays in the signal path, multiple beams may be created imaging several different parts of the sky simultaneously. "Core Station 1", the first Dutch LOFAR station, currently has equipment in the field, and is being used for testing development; see Figure 1. The full array will contain of tens of similar stations, each containing both low and high band antennae, split between a compact core, with baselines of a few kilometres, and an extended array providing baselines up to 100 km. It is scheduled for completion in 2009.

Figure 1: Low frequency antennae in the field at LOFAR's 'Core Station 1'.

The compact central core is designed to provide LOFAR with a very wide field of view. It is therefore ideal for observing large areas of the sky at high sensitivity. The Transients Key Project[5] (TKP) aims to exploit this unique capability to investigate transient phenomena. The key technology is the Radio Sky Monitor, or RSM. Multiple beams from the LOFAR core will tile out a wide area of sky, while others can simultaneously carry out independent observing programmes: see Figure 2. Focusing on the zenith and galactic plane, it is anticipated that 25% of the visible sky can be regularly monitored in this way.

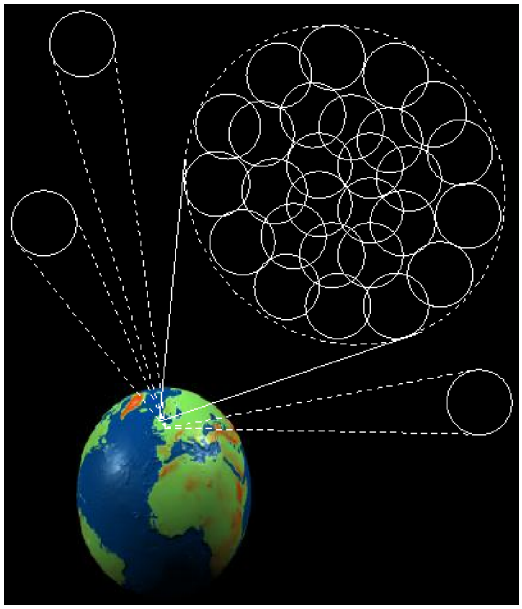

Figure 2: Radio Sky Monitor concept: multiple beams tile out a large area on the sky, while others are used for targeted observations.

The sky will be searched for transients on a logarithmically spaced range of timescales between 1 and 10^4 seconds. The aim is to respond to transients in real-time: for example, to provide notifications of detections to other facilities or to reconfigure the telescope to best investigate the phenomenon detected. This means that all data processing must be fully automated and able to keep pace with the incoming data stream—no more than a second is available to process the shortest

timescale maps.

A software pipeline for processing the data is currently under development by members of the Transients Key Project in Amsterdam. Here, we outline some of the many technical challenges faces, along with the approaches being taken to address them.

2. Pipeline Overview

The four main goals of the TKP pipeline are: to detect new transient sources in real time; to monitor the locations of previous detected transients for activity; to respond intelligently to the results of both of the previous; and to update an archive database of lightcurves with details of all detected objects.

The pipeline developed by the TKP is designed to receive data that has already been processed by the main LOFAR pipeline. This includes facilities for correlation, RFI detection and flagging, calibration and imaging. These aspects are not considered further here.

The TKP pipeline can be broadly broken down into components as shown in Figure 3 at right. In software terms, each of these is developed as a separate module, with inputs and outputs designed to facilitate communication with the other components. The rest of this document will discuss each of the blocks in more detail.

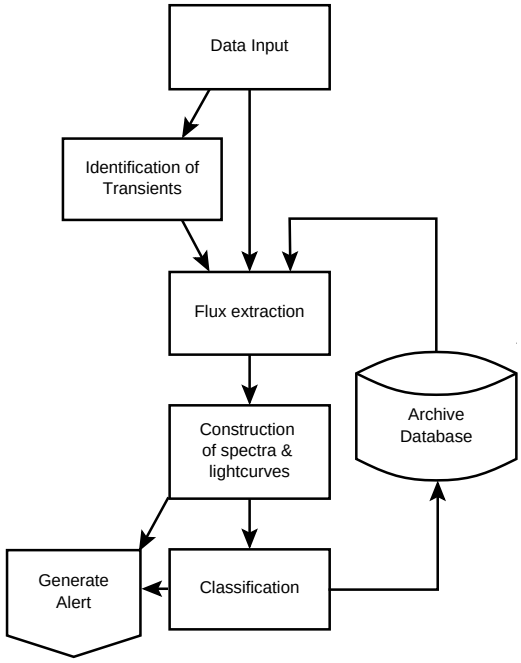

Figure 3: A schematic outline of the main components of the TKP pipeline.

In the prototyping stages, as much as possible of the pipeline is being constructed using *Python*[8], a dynamic, interpreted, object-oriented language which excels in terms of flexibility and rapid development. After a complete pipeline has been implemented, tested and profiled, it may be necessary to re-implement specific components in *C++*[9] for performance reasons.

3. Data Input

One of the design requirements of the pipeline is that it should be flexible about its data source. As well as a version implemented directly on the real-time LOFAR processing cluster, it should also be possible to use it for off-line analysis as and when required. The main body of the pipeline, therefore, does not make assumptions about the format of data presented to it. Instead, an input module normalizes the data to a consistent format. Currently, basic input modules have been written for *uv*-data in *AIPS* MeasurementSet format and image data in FITS format.

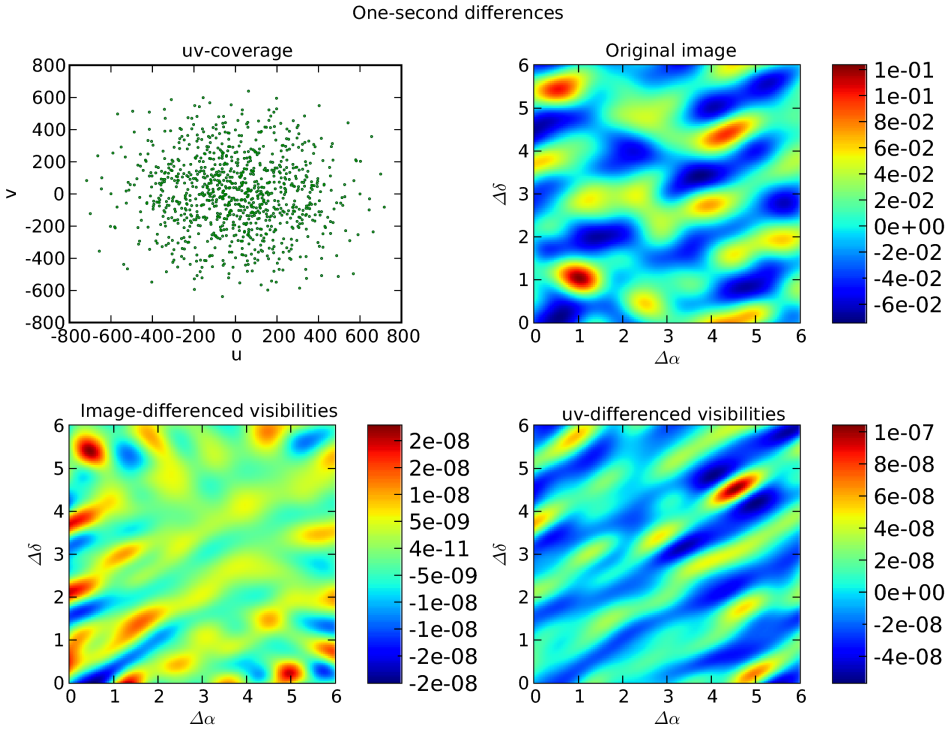

Figure 4: An example of simulations carried out to investigate the best methods of image differencing. At top left, the *uv* coverage of the simulated data is shown; in this case, since the images are separated by only one second, this does not vary significantly from image to image. The other panels show the noise distribution: note this is significantly lower in the image-differenced case.

A Data Access Layer is in development by the LOFAR Software Group which will provide a uniform interface to data stored in HDF5, FITS and MeasurementSet formats. The pipeline will make the transition to this system as soon as it becomes stable.

Internally, the data is stored as an array of numerical pixel values with associated metadata. This is made possible by the *NumPy*[7] extension to Python, which adds efficient support for large, multidimensional arrays and provides a library of high-level mathematical functions which operate on them. The *NumPy* system and its associated modules are used extensively throughout the pipeline.

4. Transient Identification

Transients are identified in the data by a process of differencing consecutive datasets and identifying any objects that do not appear in both or vary between the two. This differencing may be performed in either the *uv* or the image plane. Simulations were carried out to determine the most effective procedure: datasets were generated containing both point and extended sources as well as random noise and various differencing algorithms were applied. An example is shown in Figure 4, which compares three point image and *uv* plane differencing on a one second timescale

for a field containing extended sources. The best results—both in this example and in general—were obtained by three point differencing in the image plane. That is, a 'difference' image is generated from a particular input by subtracting the mean of its predecessor and successor (i.e. $I_t - 0.5 \times [I_{t+\delta t} + I_{t-\delta t}]$). This process is sensitive to changes in the derivative of flux; a new transient would be detected as the flux starts to increase, but not in successive difference images if the rate of increase remains constant.

After differencing has been carried out, only transient sources should be seen in the resultant images. A variety of packages were evaluated for searching the data for sources; the primary criteria being speed and accuracy. *SExtractor*[1] is a clear winner for the former; however, it has some limitations. For example, it is not capable of fitting Gaussians to detected objects, so its measurements are limited in terms of accuracy.

The *stsci_python*[1] package provides more flexibility: it integrates with *NumPy*, enhancing the latter's capabilities with a convenient set of astronomical analysis tools. Some testing demonstrated that it is relatively straightforward to duplicate *SExtractor*'s results using *stsci_python*, and that it could be easily extended to performing fitting or whatever other analysis was required.

5. Lightcurve Construction

In each dataset, flux and position measurements are made both of all newly detected transients and of all those objects flagged for monitoring in the archive database. These monitoring targets are, in general, the positions of previously detected transients (as discussed in section 2). The measurement process is closely linked to the source detection procedure described in Section 4: the combination of the *NumPy* and *stsci_python* packages are used. We fit an elliptical Gaussian to each detected object, to accurately measure its flux, shape and position.

After the fluxes have been measured in each frequency band, they are combined to make a light curve for each object at the current timestep. Measurements at different frequencies are associated by position. Associating sources is made tractable by use of the the Hierarchical Triangular Mesh (HTM)[2] system, developed for the Sloan Digital Sky Survey. This recursively decomposes the sky into a series of interlocking triangles, thereby providing a rich capability for rapid indexing and comparison of positions on the sky. The use of the HTM system makes it straightforward to query a list of sources for any that could, within a given uncertainty, lie on the same place in the sky: an operation which would be computationally intensive were right ascension and declination to be compared directly.

6. Source Classification

Source classification is important for two reasons: intelligent real-time response to observations (§7) and effective long term data mining applications (§8).

A classification system is under design to automatically identify (as far as is possible) every object stored in the archive. This process must happen asynchronously from the main pipeline: an effective classification might depend on temporal information, but it is important not to delay

[1] <URL:http://www.stsci.edu/resources/software_hardware/pyraf/stsci_python>

[2] <URL:http://skyserver.org/HTM/index.html>

information from entering the archive while a time series is collected. Therefore, the classifier will run as a separate 'daemon' process, which is notified whenever the main body of the pipeline updates the database.

A prototype classifier has been constructed based on a 'decision tree' system[3]. Initially, this will be primed with data based on the expected properties of the various sources we expect to observe; once the telescope is operational, data from real observations will be used to refine the decision trees for greater accuracy.

7. Alerts and Triggers

It is important to be able to respond intelligently to the data received. To this end, the pipeline can send alerts to destinations both within LOFAR and outside. These may be instantaneous, based on e.g. a sudden increase in brightness, or longer-term decisions based on classification results.

Internally, for example, the detection of a new transient could be used to trigger an array reconfiguration. Further, LOFAR offers a powerful 'transient buffer' facility. This enables the recording of the raw data from each antenna for a short period. On receipt of a trigger, it would be possible to 'freeze' the buffer: the contents may then be used to, for example, examine the onset of the transient event or probe it on shorter timescales.

It is also anticipated that triggers will be sent to external facilities. The prototype pipeline is already capable of sending SMS alerts; in the future, we expect both direct collaborations with other groups and to submit events to public notification systems such as *VOEventNet*[4].

8. Archive Database

An archive of lightcurves for all transients detected by the system will be kept. This serves two roles: it will both be useful for data-mining, but also for recording the positions of monitoring targets for each telescope pointing.

The data volumes to be stored in the archive are significant. The TKP team extrapolated the source counts given by Huynh et al.[6] to lower frequencies and combined them with likely parameters of the complete LOFAR telescope to estimate the number of sources which might be observed in a field of view for a given frequency and integration time. These range from tens of sources in a one second integration at 200 MHz to tens of thousands in 10^4 seconds at 30 MHz.

It is estimated that 379 sources will be observed in 1 second at 30 MHz assuming a 10^{-7} false detection rate ratio and 4 MHz bandwidth. If the TKP is allocated 20% of LOFAR observation time and 50% of that allocation is used in the low band, such observations will be made around 3.2×10^6 times in one year. Given 24 beams, we would therefore expect around 3×10^{10} catalogue entries per year. The current database structure calls for around 1 kB per entry, resulting in a total of 28 TB per year of operation. This is the single largest part of the catalogue; including the high band and other timescales will increase the database size to several tens of terabytes.

The large volume of data to be stored and queried pushes conventional database systems to their limits. The TKP team is cooperating with the Centrum voor Wiskunde en Informatica in Amsterdam in overcoming these limits using their high-performance *MonetDB*[2] system, an inno-

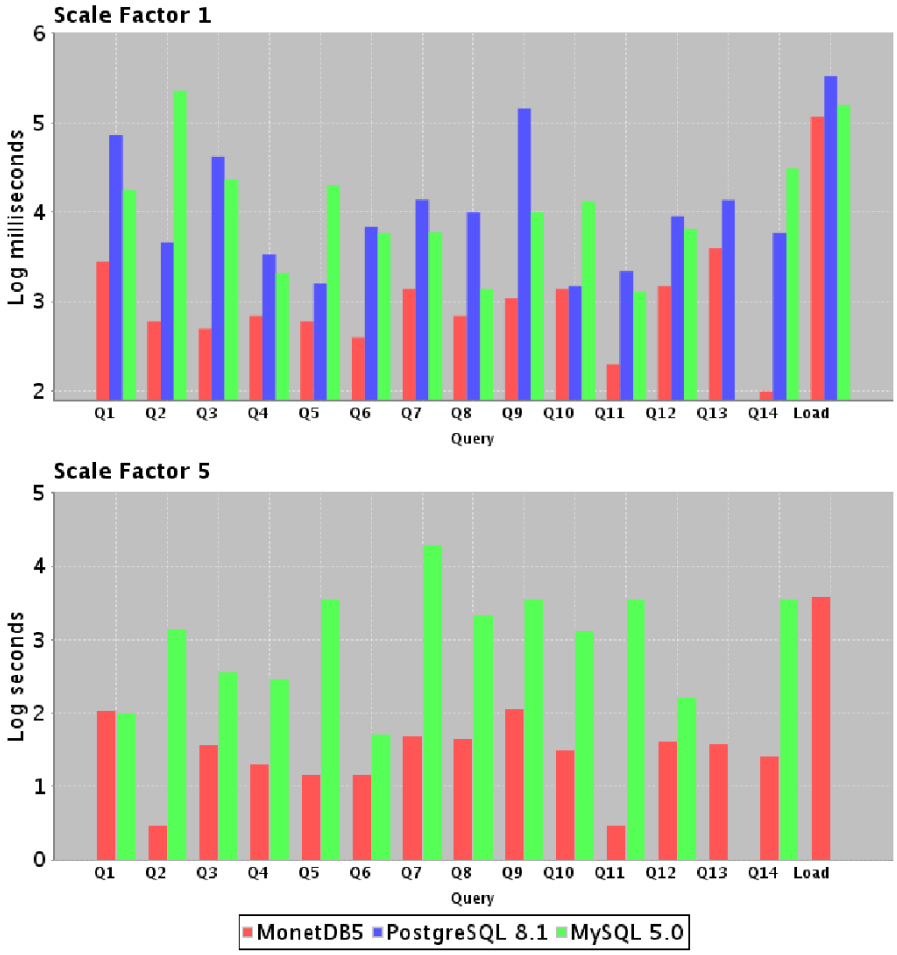

Figure 5: The TPC Benchmark H[10] consists of a standardized series of queries and data modifications. Here, results results comparing *PostgreSQL* 8.1, *MySQL* 5.0 and *MonetDB/SQL*, each with default settings, are shown. Each column represents the execution time of one of the standardized TPC-H tests. At scale factor 1, all the data fits within main memory; at scale factor 5, it does not, with a consequent dramatic increase in I/O requirements. At a scale factor of 5, PostgreSQL failed to execute the benchmark.

vative column-based database designed for extreme performance demands. Figure 5 shows a brief comparison of *MonetDB* performance compared to leading open-source solutions.

9. Conclusions

Development and construction of a transient detection and monitoring pipeline for LOFAR is now well underway. Considerable thought has been given to the various challenges such a pipeline must meet, and a variety of solutions have emerged. These should begin to bear fruit over the coming year as the telescope is commissioned.

References

[1] E. Bertin and S. Arnouts, *SExtractor: Software for Source Extraction*, *A&AS* **117**, 393–404, 2006.

[2] P.A. Boncz and M.L. Kersten, *Monet: An Impressionist Sketch of an Advanced Database System* in proceedings of *The Basque International Workshop on Information Technology*, 1995.

[3] L. Breiman, J. Freidman, C.J. Stone and R.A. Olshen, *Classification and Regression Trees*, Chapman & Hall, 1984.

[4] A.J. Drake et. al., *VOEventNet: Event Messaging for Astronomy*, *Bulletin of the American Astronomical Society* **38**, 1002, 2006.

[5] R.P. Fender et. al., *The LOFAR Transients Key Project* in proceedings of *The VI Microquasar Workshop: Microquasars and Beyond*, 2006.

[6] M.T. Huynh, C.A. Jackson, R.P. Norris and I. Prandoni, *Radio Observations of the Hubble Deep Field-South Region. II. The 1.4 GHz Catalog and Source Counts*, *AJ*, **130**, 1373–1388, 2005.

[7] T.E. Oliphant, *Guide to NumPy*, <URL:http://www.tramy.us/guidetoscipy.html>, 2006.

[8] G. van Rossum and F.L. Drake, *Python Reference Manual*, Python Software Foundation, 2006.

[9] B. Stroustrup, *The C++ Programming Language, Special Edition*, Addison Wesley, 2000.

[10] Transaction Processing Performance Council, *TPC Benchmark H (Decision Support) Standard Specification*, revision 2.6.1, <URL:http://www.tpc.org/tpch/>, 2007.