REAL-TIME SOFTWARE CORRELATION.

Nico Kruithof[1], Damien Marchal[2]
[1] *Joint Institute for VLBI in Europe (JIVE), Postbus 2, 7990 AA Dwingeloo, The Netherlands Kruithof@jive.nl,*
[2] *University of Amsterdam (UvA), Kruislaan 403, 1098 SJ Amsterdam, The Netherlands dmarchal@science.uva.nl*

Abstract In this paper we present the progress of the SCARI*e* project. In which, we investigate the capabilities of a next generation grid based software correlator for VLBI. We will mostly focus on the current design of our software correlator, and on the challenges of running real-time scientific experiments on top of grids infrastructure. This paper also contains experimental results on both software correlation as well our current experiments on the DAS-3 grid and StarPlane its user-controllable dynamic photonic network.

1. INTRODUCTION

Very Long Baseline Interferometry (VLBI) [10] is a type of interferometry used in radio astronomy, in which data received at several telescopes is combined to produce an image with very high angular resolution. This is important as recent astronomical research studies the deep-sky thus requiring high angular resolution to capture all the details of the observed sources. The angular resolution depends on the size of the radio-telescope dish but due to mechanical constraints, it is difficult to build moveable telescope dishes with a size much larger than 100m. With VLBI, it is possible to simulate a dish of a size equivalent to the maximal distance between the farthest telescopes; and so, making virtual telescope with a dish of a size of the Earth. This is done by having several distant telescopes observe the same source, the signal are recorded and sent to a central facility for processing. The central step in processing the data is computing the correlation function between each pair of incoming signals. Within the SCARI*e* project we are developing and analyzing the capabilities of software correlation using grids technologies. Given the

amount of processing power needed for perform software correlation, we are distributing the computation using a master-worker approach.

SCARI*e* is a typical example of a recent trend of the e-Science community in which the worldwide computation centers and the scientific instruments are connected through high-speed networks. We think that to generalize the utilization of such world-size inter-connected facility, the grids and their middleware have to offer services that match the three following important aspects:

- *performance*: solving bigger problem implies the use larger and/or more efficient hardware.
- *isolated environment*: some users require guarantee of isolated execution environment in which the other applications cannot interfere. This is important for real-time application or for benchmarks.
- *scheduling*: it may be needed when the application requires to be synchronized with ”external events”: like a radio-telescope observing at a specific date.

In SCARI*e* we are facing these three challenges, as we want to build a high-performance software correlator able to process in real-time a VLBI experiment. These three aspects are hot-topics in the grid community, especially the networking side; probably because worldwide networking (Internet) has a long history in being a best-effort shared resource. As best-effort is does not provide traffic-isolation we are conducting our experiments using the experimental DAS-3 grids and its an user-controllable dynamic photonic network called StarPlane that could permit us to build, on demand an application specific, isolated virtual-network on top of the complete grid.

The rest of the paper is organized as follows. Section 2 contains a general introduction to VLBI and its recent development called *e*-VLBI. Section 3 describes the architecture of the software correlator. Section 4 describes experiments and benchmarks on executing SCARI*e*on StarPlane DAS-3. Section 5 concludes the article with future work.

2. VLBI

To achieve larger and larger resolution in astronomical imaging, it is necessary to build larger telescopes, or to revert to interferometry. As interferometry combines the measurements of several telescopes to simulate a dish of a size equivalent to the maximal distance between the farthest telescopes on the plane orthogonal to the viewing direction. Numerous arrays (groups of telescopes) use this technique, e.g., the VLA (Very Large Array), Lofar (Low-frequency array), the EVN (European VLBI Network) or the VLBA (Very Large Baseline Array). Interferometry with telescopes that are geographically very far apart

Description	# telescopes	# sub-bands	data-rate (Mb/s)	spect/prod	Tflops
Fabric-demo	4	2	16	32	0.16
1 Gb/s, full array	16	16	1024	16	83.39
future VLBI	32	32	4096	256	˜21457

Table I. Network bandwidths and computing power needed for an *e*-VLBI experiment based on a XF architecture.

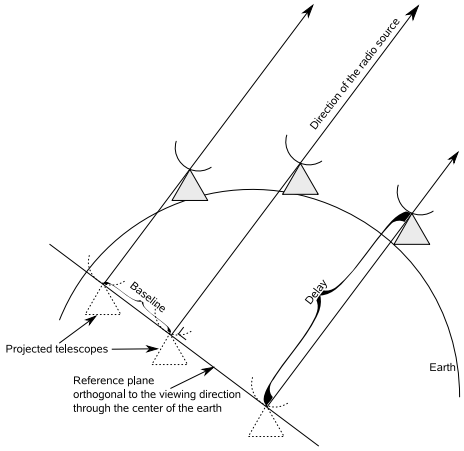

Figure 1. Aligning the signals before correlation.

is refered to as Very Large Baseline Interferometry (VLBI). With VLBI it is possible to build a virtual radio-telescope with a dish of size of the Earth. As the angular resolution of a VLBI experiment depends on the maximal projected distance between two radio-telescopes, VLBI achieves unsurpassed angular resolution with the drawback of a relatively low sensitivity [10]. Another important property is sensitivity, as it allows to detect fainter astronomical objects. Increasing the sensitivity is possible by adding more radio-telescopes or by increasing the sampling rate or resolution. Increasing the sampling rate or resolution increases the data rate per telescope.

In order to get the final picture the signal gathered from the radio-telescopes have to be correlated at a central place, the Joint Institute for VLBI in Europe (JIVE). JIVE is operating a dedicated hardware correlator [8].

The maximal capacity of this hardware correlator is 16 telescopes at a data rate of 1Gbs each. The requirements on both the data streams and the computing power are shown in Table I.

***e*-VLBI.** Traditionally, in VLBI, the data is recorded at the telescopes on disk packs during an experiment. After the experiment the disks are shipped

to a central institute. There can be several weeks between the experiment and the time when the correlated data becomes available.

Currently, JIVE is in the transition phase from traditional VLBI to *e*-VLBI [9]. In an electronic VLBI (*e*-VLBI) experiment, data from the telescopes is transferred directly over the internet to JIVE, where it is streamed into the correlator in real time. The data transport from the telescopes to JIVE goes over several networks like local connections, paths provided by NRENs and the GÉANT backbone in Europe.

Transporting the data over the network has several advantages over a traditional experiment. Obviously, the results of the experiments are almost immediately available. This opens up the possibility to change the course of an experiment based on earlier findings. Also, *e*-VLBI allows for real time analysis of the data and helps to identify and resolve minor technical problems in the data collection during the experiment.

Several experiments in the past have shown that real time *e*-VLBI is possible. The EC funds the EXPReS project [7] which aims at building a production-level *e*-VLBI instrument of upto 16 intercontinental telescopes connected in real-time to JIVE and available to the general astronomy community.

Correlation. Correlation is the process by which data from multiple telescopes is collected and combined to measure the spatial Fourier components of the image of the sky. It consists of two steps: first applying a delay correction to align the signals and secondly computing the correlation function for each pair of telescopes called a baseline.

To align the signals from the different telescopes, we project all the telescopes on the plane through the center of the earth and orthogonal to direction to the source, see Figure 1. We will correlate the signals received by the virtual projected telescopes. To compute these signals, the signals from the real telescopes are delayed with the distance between the telescope and its projection multiplied with the speed of light (the signal travels with the speed of light). Note that the delay changes during the observation because the earth rotates. We conveniently split the delay in an integer number of samples and a remaining fraction. While the integer delay is easily done by an offset in the sample buffer, the fractional bit shift is usually implemented as a phase rotation in the frequency domain. In a final step, called the phase rotation, we change the sample rate to match the rate of the delay function.

After the delay has been applied, the signals are ready for correlation. Correlation [5] is mathematically defined as a function on two signals in which the first signal is delayed with discrete steps and the integral is computed of the delayed signal multiplied with the second signal. The correlation is done for each baseline (pair of telescopes). The correlation is called an auto-correlation if the signal of a station is correlated with itself and a cross-correlation if the

signals are from different stations. Note that the complexity of the correlation is quadratic in the number of telescopes, as it is linear in the number of baselines.

To increase the signal to noise ratio, the correlated signal is averaged over a certain period of time. Typical averaging times lie in the range of $0.25 - 4$ seconds. Finally, the averaged signals are Fourier transformed. Correlation in this order is refered to as XF, because the correlation is done before the Fourier transform. This is how the correlation is implemented in most hardware correlators, as it allows for large parallelisation.

One of the properties of correlation is that the Fourier transform of two correlated signals is equal to multiplying the Fourier transformed signals [3]. This relation is used in most software correlators, where correlation is more expensive than multiplication. After the delay, the signal from each telescope is Fourier transformed and the signals for each baseline are then multiplied elementwise. This implementation is also refered to as FX: the Fourier transform comes before the correlation.

3. SOFTWARE CORRELATOR

If the data in an *e*-VLBI experiment can be streamed over the internet to JIVE, it can also be sent to another correlator. Within SCARI*e*, we are investigating the possibilities of a next generation software correlator using a computing Grid. A similar attempt is presented in [6]. The advantages of a software correlator over a new dedicated hardware correlator lies in its flexibility and accuracy. The main advantage of a dedicated hardware correlator is the greater performance. The flexibility of its architecture allows the software correlator to change with the individual needs of researchers. In fact, the first version of the software correlator was developed to track the Huygens spacecraft during its descent through the atmosphere of Saturn's moon Titan. Due to the nature of this experiment, special requirements are put on the correlator, which the current hardware correlator is not able to provide. Moreover, we expect that the costs of developing a software correlator are much lower than the costs for a hardware correlator.

Currently, the software correlator is used in the production environment for doing ftp-fringe tests for the EVN network. Since the EVN is an ad-hoc array, the EVN telescopes are reconnected before every VLBI-session. In order to test the EVN network, a ftp-fringe test is done in which the telescopes observe a well known source and transmit the data to JIVE where it is processed immediately. The ftp-fringe tests provide quick feedback to the stations on their performance.

Computationally the correlation is relatively inexpensive, in the sense that it requires only few operations per received byte. However, due to the high data rates, the absolute number of flops (approximately 3Tflops) required by the

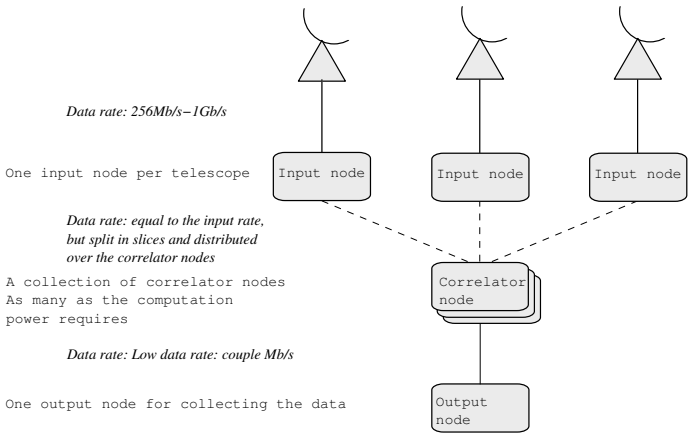

Figure 2. Outline of the network connections between different components in the software correlator.

application is still extremely high for computing grids. Moreover, the problem is quadratic in the number of telescopes participating in the experiment since it is linear in the number of channel pairs that have to be correlated. The huge need for networking and computing power together with it's flexibility, makes a cluster an good platform for this application. Moreover, using grid technology the software correlator can easily be executed on a large number of different clusters.

3.1 Design

In the software correlator, we split the computation in time slices. These time slices are processed in parallel (see Figure 2). The assignment of time slices to computing nodes is done by a unique manager node. For every telescope there is a unique input node that receives the raw stream and converts it into time slices. These time slices are sent to correlator nodes which do the actual correlation. The size of the output of the correlation is much smaller than the input size and is collected and stored by a single output node.

The manager node is the central node that controls the workflow of the entire software correlator. It assigns time slices to available correlator nodes and handles errors.

The input node receives the data from a data stream, which can either be a file, a TCP connection or directly from the dedicated hardware used to record and play back the data. It performs the integer delay correction and then sends the data to the proper correlator node.

The correlator node gets data for the same time slice from every input node. First it, compensates for the fractional delay and performs the phase rotation. These manipulations are not done on the input node because they require floating point samples, hence the data stream expands from 2 bits per sample to 32 bits or even 64 bits per sample, which would require much more network bandwidth. After the fractional delay correction and the phase rotation, the signal is ready to be correlated. The auto and cross correlation are then computed by Fourier transforming the input signal and element-wise multiplying all baselines. These values are accumulated over a certain period of time and the accumulated values are sent to the output node.

The output node will receives the data from the correlator nodes, sorts the data and stores it at a specified location.

4. EXECUTION AND DEPLOYMENT.

In the context of *e*-VLBI the software correlator can be executed in two different operational modes that are: *batch-execution* and *real-time*. Grids have a long history in running batch jobs and grid-middleware is now doing a good job on this task. The execution of real-time applications is much more complex; grid infrastructure and middleware have to provide guarantees on the Quality of the Service to insure successful execution of the job. Quality of Service management in grids is still evolving rapidly. To experiment with these aspects we are running SCARI*e* on a research grid called DAS-3 and its manageable network called StarPlane.

4.1 Real-time and quality of service

The term *real-time* has many definitions in the computer science community, in this paper we will consider that a *real-time* computation is a computation in which: *the amount of buffering for an infinitely long experiment will only require a finite amount of buffers*. This is a formal way to define a process in which the incoming data are ”consumed” by the computation as fast as they are generated. This definition also implies that once the application is started the allocated ”space” on the resources will be maintained during the complete execution.

The main resources SCARI*e* is using are: the network bandwidth, the computation resource and the disk-space. Sharing of the computational resource is now a well understood process and most of the time it is part of the execution service that allocates the requested resources and if all resources are acquired, it deploys and executes the application. The simplest way to offer guaranteed service over a shared resource can be done by restricting the access to only one user at a time. This approach is used in the DAS-3 grid (based on SGE) in which

an allocated node is simply unusable by other users. Same principle could be applied to the complete grid including its networks and other resources. A more flexible approach consists in sharing the resource under the arbitration of a third party that will insure that each application is using only the allocated part of the resource. This is the case with Layer-3 QoS for networks, or the Completely Fair Scheduler (per application CPU time allocation) combined with the Process Containers recently introduced into the Linux Kernel[2]. In a very general point of view all these technologies virtualize the resource thus they permit to build on top of a real grid a complete isolated environment based on user requirements.

4.2 Running SCARI*e* on DAS-3 and Starplane

Networking performance and QoS management is one of the most challenging aspects of SCARI*e*. The regular Internet Layer3 IP routing based on the best-effort policy has a great flexibility but is often slow and unpredictable; on the other hand, we have dedicated *lightpaths* as available in *lambda Grids* [1], with their predictable delays and throughputs offer good performances and a good basis to offer Quality of Service. Giving end users access to dedicated connection has been implemented in many of the current research and education networks. The Dutch National Resarch and Education network SURFnet is one of them. This is used to deliver the data from the radio-telescope to the computation center.

In SCARI*e* having lighpaths between radio-telescopes and the computation center is not sufficient to insure real-time operation. SCARI*e* also uses the network to distribute the correlation. Therefore, a per application manageable network is then required. The DAS-3 supercomputer [4] is composed of fives clusters located in the Netherlands and connected by a photonic network called StarPlane. The StarPlane project manages eight wavelengths with the goal to build a network service that permits *an application-controlled photonic network and node-to-node traffic isolation*. The novelty of the StarPlane project lies in its attempt to build a virtual network service at the lowest possible networking layer: the photonic layer for the optical part and ethernet-layer2 for the connection to the nodes. Another property of StarPlane is that the photonic lightpath can dynamically be reorganized to match the requirements of the user-application. By using StarPlane, a complete virtual network over distributed cluster sites can be build on demand. The lighpaths can also be re-organized at run-time if the network load changes. For SCARI*e*this allows us to distribute the workload over several cluster locations while taking profit of the high-bandwidth with a relative good QoS control over the complete network domain.

4.3 Benchmarks on DAS-3

SCARI*e* and StarPlane have parallel roadmaps. Hence the complete approach cannot be tested yet. We have conducted correlator performance tests using DAS-3. The current software correlator is currently able to perform a 4x256Mbps experiment at 30% of the real-time speed using a total of 15 (quad core 2.0Ghz cpu) nodes. Analysis of the software correlator shows that the bottleneck is in the extraction of the data from the input stream (which we can do at 110Mbps). Hence, we are not able to do real-time correlation, but an optimized prototype shows that we will be able to resolve this bottleneck.

In parallel with the benchmarks of the correlator, we are also testing the capabilities of StarPlane to do high performance traffic isolation that is needed for real-time correlation. At the time of writing StarPlane has implemented the service that allows a program to build and allocate a lighpath between two clusters on demand. We tested this feature by running two client-server applications transmitting data between clusters.

The network traffic of the two applications is depicted in Figure 3. At the start of the application, a request for lighpath allocation is issued (arrow 1). After a while the throughput drops because the second application starts sending data as well (arrow 2). As long as the lighpath is not ready (photonic switching takes 10 minutes) the application is sending the traffic to the default 1Gb/s ethernet route; when the lighpath is ready (arrow 3) the traffic is rerouted to the lightpath.

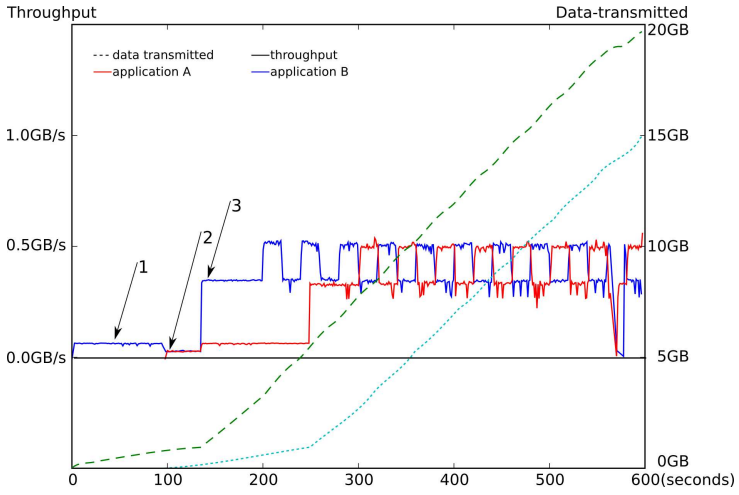

Figure 3. Two application are started at time (arrow 1) and (arrow 2). The two applications have to share the 1Gbps network bandwith. When a lighpath is allocated (arrow 3) the increase in performance is clearly visible.

The results of this experiment are encouraging as they show that good network performance can be obtained between several cluster locations. Lighpath dynamic switching also permits to adapt the photonic part of the network to the software correlation work load and distribution. Neverthless the results of this experiment also rise questions, the secured lightpath is supposed to deliver reliably ”550MB/s” of throughput between a pair of clusters. In Figure 3 we can see a periodic artifact, the traffic falling down to 300MB/s for few seconds, for which we have no explanation. A second issue to investigate is that from time to time the lighpath connectivity disapears entirely (e.g. at second 580). We are currently working with Starplane team to understand and solve these problems.

5. CONCLUSION AND FUTURE WORK

In the last year we laid the foundation for a flexible software correlator based on distributed computing technologies. SCARI*e* is now useable for batch correlation and is used for ftp-fringe tests for the EVN.

In order to reach our real-time correlation goal, we are collaborating intensively with the StarPlane project to test new network architectures for grids with guarenteed Quality of Services.

Future work. Within SCARI*e* and a related project FABRIC, which is a joint research activity in the EXPR*e*S project, we are currently improving and testing the software correlator. We still see possibilities to improve the efficiency of the software correlator, which we would like to investigate further.

On the batch processing aspect of SCARI*e* we want to investigate how resources can be added dynamically at runtime to accelerate the computation. This include node joining and leaving as well as setting up additional lighpaths to the new nodes.

On the real-time processing aspect, more work has to be done on the traffic isolation features of the incoming networks as well as the isolation provided by Starplane.

6. ACKNOWLEDGMENT

SCARI*e* is a joint research project between JIVE, the University of Amsterdam and SARA funded by the Netherlands Organization for Scientific Research (NWO).

References

[1] *Lambda-Grid developments: RDF/NDL, AAA and StarPlane.*, 2007.

[2] Completely Fair Scheduler and Process Containers - `http://en.wikipedia.org/wiki/Completely_Fair_Scheduler`.

[3] Definition of the correlation function - `http://mathworld.wolfram.com/Cross-CorrelationTheorem.html`.

[4] Distributed ASCI Supercomputer - `http://www.cs.vu.nl/das3/`.

[5] Definition of the correlation function - `http://en.wikipedia.org/wiki/Cross-correlation`.

[6] A. T. Deller, S. J. Tingay, M. Bailes, and C. West. Difx: A software correlator for very long baseline interferometry using multi-processor computing environments, 2007.

[7] EXPReS project website `http://expres-eu.org`.

[8] R. T. Schilizzi, W. Aldrich, B. Anderson, A. Bos, R. M. Campbell, J. Canaris, R. Cappallo, J. L. Casse, A. Cattani, J. Goodman, H. J. van Langevelde, A. Maccafferri, R. Millenaar, R. G. Noble, F. Olnon, S. M. Parsley, C. Phillips, S. V. Pogrebenko, D. Smythe, A. Szomoru, H. Verkouter, and A. R. Whitney. The evn-markiv vlbi data processor. *Experimental Astronomy*, 12:49–67, 2001.

[9] A. Szomoru, A. Biggs, M. A. Garrett, H. J. van Langevelde, F. Olnon, Z. Paragi, S. Parsley, S. Pogrebenko, and C. Reynolds. From truck to optical fibre: the coming-of-age of evlbi, 2004.

[10] J. A. Zensus, P. J. Diamond, and P. J. Napier, editors. *Very Long Baseline Interferometry and the VLBA*, 1995.