ВАК 2024
MODERN ASTRONOMY:
FROM THE EARLY UNIVERSE TO EXOPLANETS AND BLACK HOLES
300 years Russian Academy of Sciences

Testing the DiFX VLBI correlator in a cloud service infrastructure

A. Kononov, I. Bezrukov, and A. Melnikov

Institute of Applied Astronomy of the Russian Academy of Sciences,
Saint Petersburg, 191187 Russia

Abstract. This study explores the possibility of using cloud infrastructure to deploy the DiFX software correlator for creating a backup system in case of emergencies. The proposed approach involves running DiFX on virtual servers provided by a cloud service, allowing the reduction of financial and time resources. The DiFX software correlator is a primary tool for processing data obtained through Very Long Baseline Interferometry (VLBI). Running such software on cloud provider servers will enable the scaling of the data correlation system. For the experiment, the automation tools Ansible (written in Python) and Terraform were used. Ansible was utilized to deploy the necessary software on a pre-created virtual machine (VM) with Linux, while Terraform managed the provisioning of the virtual machine in the cloud environment. The study analyzes the performance and scalability of DiFX in the cloud. Additionally, the advantages and limitations of deploying DiFX in the cloud in terms of cost, flexibility, and ease of use were evaluated. This paper presents the results of testing and using cloud technologies for running DiFX and justifies the relevance of developing such software.

Keywords: instrumentation: interferometers; techniques: interferometric; virtual observatory tools

DOI: 10.26119/VAK2024.173

SAO RAS, Nizhny Arkhyz, Russia 2024 https://vak2024.ru/

1 Introduction

This study explores the use of cloud infrastructure to deploy the DiFX[1] software correlator as a backup system. Running DiFX on cloud servers can reduce financial and time resources. DiFX is essential for processing Very Long Baseline Interferometry (VLBI) data, and cloud deployment allows scaling the data correlation system.

2 Experiment setup

The experiment was conducted in several stages.

Creating a virtual machine with the required configuration. Using Terraform,[2] a request was generated to provide a VM with specified parameters from a cloud provider. Virtual machines were created on the Yandex Cloud platform with Ubuntu 22.04 and 8, 16, and 32 computational cores. Time taken: 1 minute.

Automated software setup. To automate the software installation on the virtual machine, an Ansible[3] script was developed. Key components installed were the Intel IPP library and the DiFX VLBI correlator. Time taken: 20 minutes.

Transferring data for correlation processing. Files m5b with digitized radio signals were transferred from the Institute of Applied Astronomy (IAA RAS) to the virtual machine in Yandex Cloud using jive5ab/m5copy in four parallel streams. The peak speed was 600 Mbps, with an average of 200 Mbps. Each 2 GB file was transferred in 10 seconds.

Correlation and processing. Using the DiFX software correlator, correlation processing of the received VLBI data was performed.

Uploading results. Correlation processing results were uploaded from the virtual machine in Yandex Cloud back to the IAA RAS data storage system for further analysis.

Deleting the virtual machine. After the experiment, the virtual machine was deleted using Terraform to optimize resource usage and reduce costs.

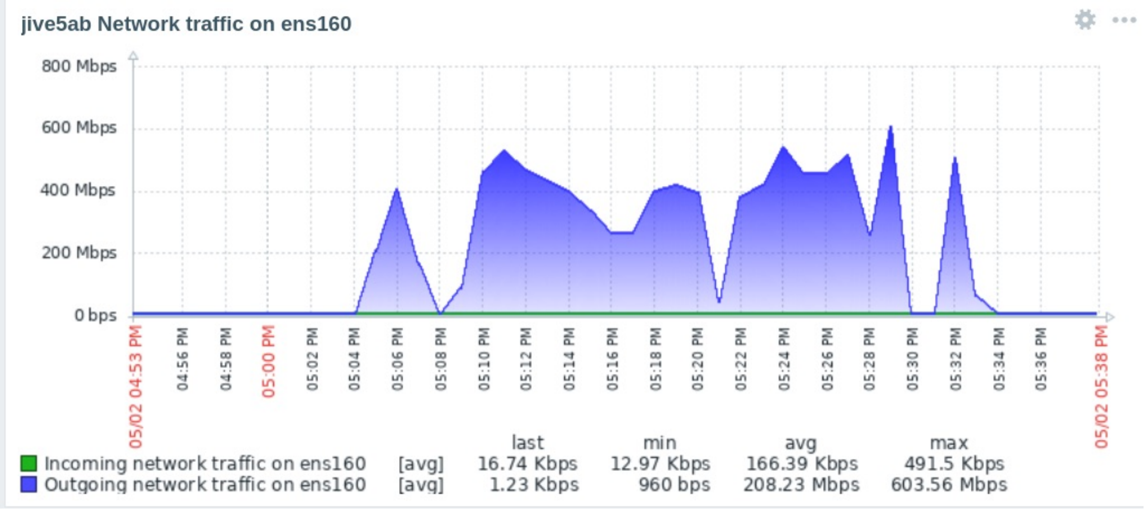
3 Data transfer speed testing

To assess the feasibility of using cloud infrastructure for data correlation, the channel bandwidth between a Yandex Cloud virtual machine and the IAA RAS server was tested. Data were transferred in four streams using jive5ab/m5copy. The test results with the UDT protocol are shown in Fig. 1.

[1] https://www.atnf.csiro.au/people/Tasso.Tzioumis/vlbi/dokuwiki/doku.php/difx/documentation/

[2] https://terraform-provider.yandexcloud.net/

[3] https://docs.ansible.com/



		last	min	avg	max
🟩 Incoming network traffic on ens160	[avg]	16.74 Kbps	12.97 Kbps	166.39 Kbps	491.5 Kbps
🟦 Outgoing network traffic on ens160	[avg]	1.23 Kbps	960 bps	208.23 Mbps	603.56 Mbps

Fig. 1. Data transfer speed to a virtual machine in Yandex Cloud.

Also, iPerf3 was used to test the channel speed between the Yandex Cloud VM and the IAA RAS data storage. Each session lasted 10 seconds. From these data it has been concluded that the average transfer speed per channel is 660 Mbps, slightly higher than the peak value obtained by jive5ab.

4 Session transfer and processing

To test the DiFX correlator, the observing session ri4518 was processed. This is a 1-hour session obtained with two 32-meter VLBI telescopes ("Badary" and "Zelenchukskaya") of the IAA RAS. Data transfer was made using jive5ab/m5copy in four streams, and processing was conducted on VMs with 8, 16, and 32 cores (Table 1). An analysis has shown that data transfer time is about 35 minutes and is independent of the number of virtual machine cores. Similarly, data processing time does not depend on the number of computational node cores.

Table 1. Session transfer and processing Times

No.	Number of cores	RAM	Data transfer time	Session processing time
1	8	8 GB	34 min 48 s	38 min 48 s
2	16	16 GB	34 min 45 s	34 min 18 s
3	32	32 GB	34 min 17 s	33 min 17 s

5 Virtual machine operation

An external system monitoring tool from Yandex Cloud was used to assess the load on the virtual machine. It monitored the processor cores, disk subsystem, and network channel during three stages: software installation, data loading, and session processing. Each stage took about 1/3 of the total operating time. Monitoring results for the 32-core VM are shown in Fig. 2.

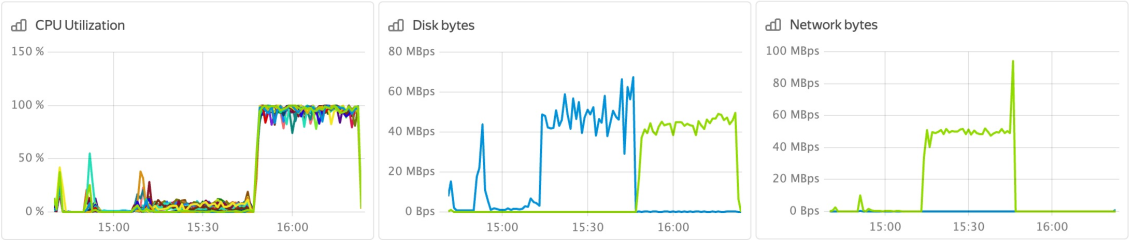


Fig. 2. Monitoring of a 32-core virtual machine.

6 Conclusions

- Software was written to automate the deployment of the DiFX software correlator environment on a virtual machine in the Yandex Cloud service.
- The file transfer speed between the virtual machine in Yandex Cloud and the data storage system of the IAA RAS was measured.
- A test transfer and correlation of a one-hour session recorded by two antennas were carried out.
- It was noted that the data transfer and processing time did not change when increasing the number of cores of the virtual machine from 8 to 16 and 32. This is a very interesting result. The reason for this can be investigated in future works.
- The solution described in the article can be used for data correlation in case of unforeseen situations.

7 Future plans

To increase session processing speed, the following options can be considered:
- using a preconfigured virtual machine image to reduce initialization time;
- shutting down rather than deinitializing the VM after processing to save time;
- parallel processing of sessions by assigning specific cores;
- processing sessions in parallel with simultaneous data transfer to the VM.