

# Installation and testing the LOFAR software on Prometheus



Elżbieta Kuligowska<sup>1</sup>, Maciej Czuchry<sup>2</sup>, Tomasz Kundera<sup>1</sup>,  
Carole Roskowinski<sup>3</sup>

<sup>1</sup>Astronomical Observatory of Jagiellonian University, Orla 171 Kraków, 30-244 Poland

<sup>2</sup>Academic Computer Centre CYFRONET, Nawojki 11 Kraków, 30-950 Poland

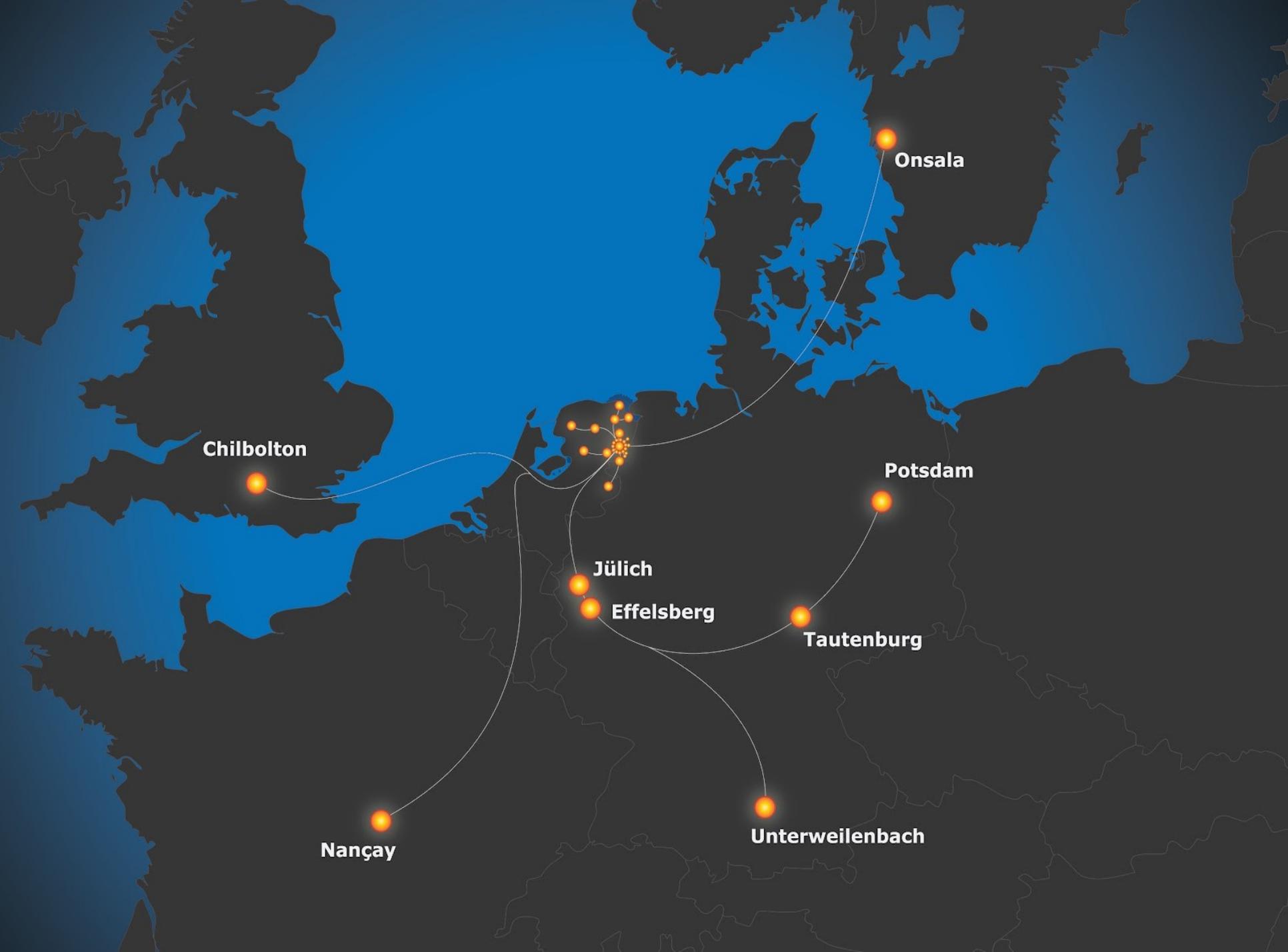
<sup>3</sup>Nicolaus Copernicus University in Toruń, Gagarina 11 Toruń, 87-100 Poland

# LOFAR (Low-Frequency Array)

- Network of radio telescopes completed in 2012 by ASTRON (Netherlands Institute for Radio Astronomy)
- Goal: long interferometry baselines
- Antennas spread out over an area of more than 1000 km in diameter, localized in several European countries including Poland
- Data: radio waves from 10–240 MHz of great resolution, making LOFAR the most sensitive radio observatory at low observing frequencies



▫ [www.lofar.org](http://www.lofar.org)



**Chilbolton**

**Onsala**

**Potsdam**

**Jülich**

**Effelsberg**

**Tautenburg**

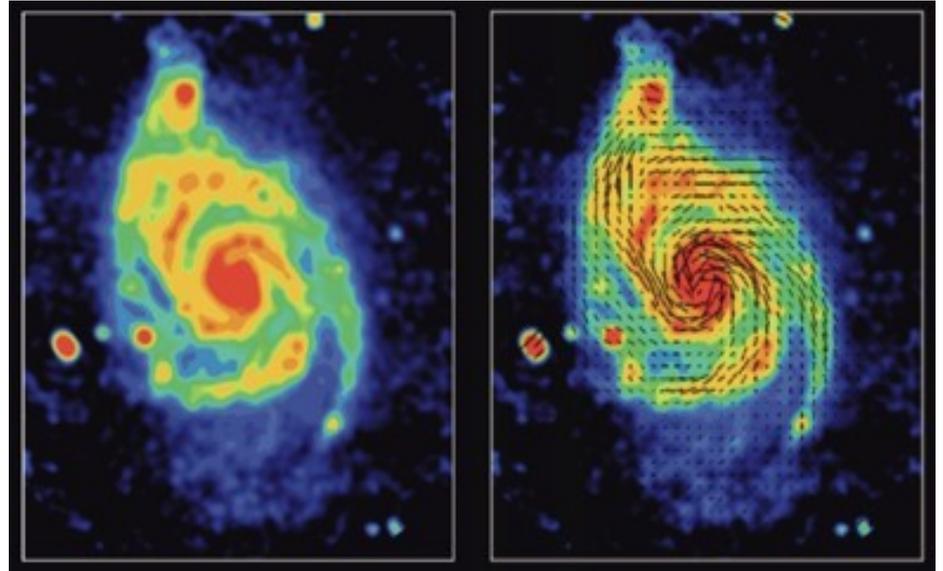
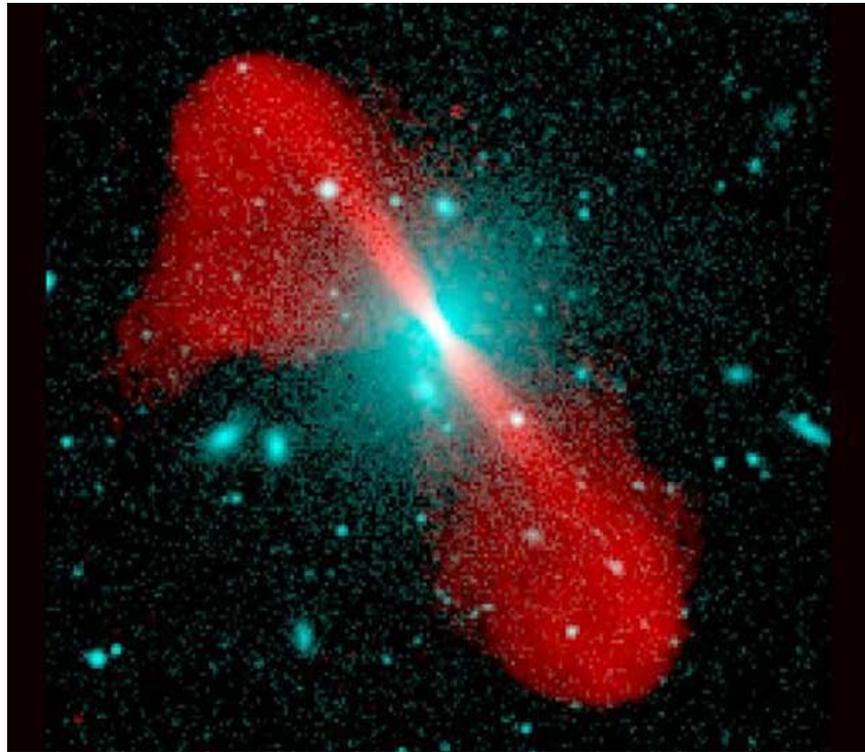
**Nançay**

**Unterweilenbach**

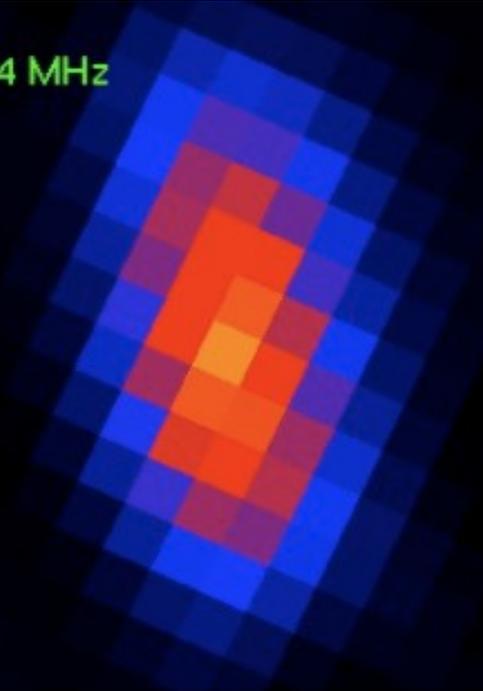
# LOFAR (Low-Frequency Array)



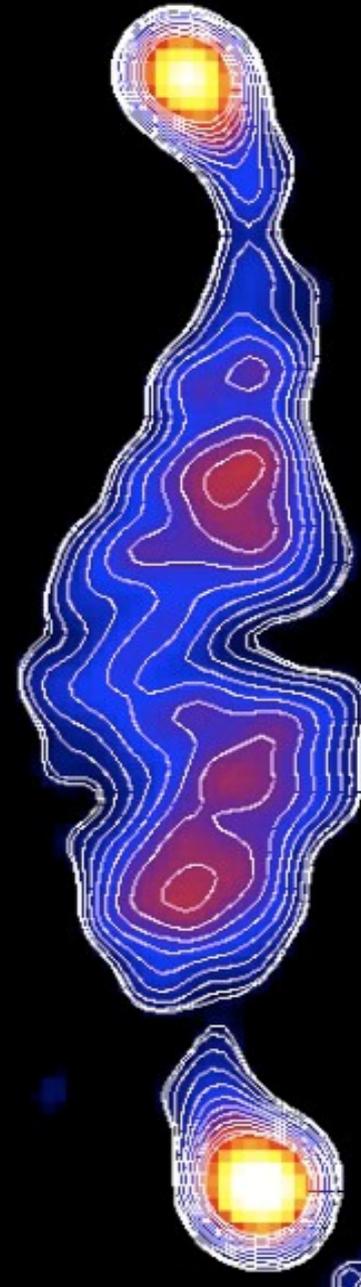
# LOFAR (Low-Frequency Array)



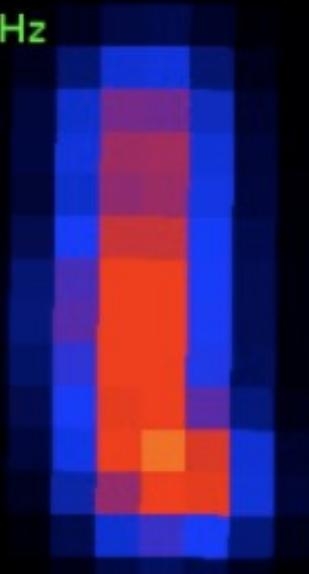
VLSS 74 MHz



LOFAR 173 MHz  
detailed version with contours



WENSS 325 MHz



# LOFAR Software

**Composed of several modules (simulating the steps of data reduction)**

Initial flagging/correcting, removing bad data -> Calibration -> Imaging

- Highly distributed
- Composed of C, C++, FORTRAN and Python components
- Poorly documented (many things unclear)
- Written partly by non-astronomers
- Computation time - in some cases extending up to several months for local computers

# LOFAR Software – installation on Prometheus

- Hybrid: for C and C++ - written components - **GNU GCC** (gcc and g++) compilers used, for Fortran components - **Intel ifort** (due to its compatibility with some libraries previously installed from the sources)
- To achieve the proper functioning of the software on the cluster it was also required to modify the selected Python libraries and adapt them for the SLURM - based environment
- Adding CASA/Casacore
- Tools for visualization
- Adding some missing scripts & building necessary databases not included in the official installation package!
- Defining some user-defined paths (not always described in official documentations)

## **NEXT STEP**

- Adding **Factor** module - a tool for automated producing high-resolution images from LOFAR data

# LOFAR Software – Conclusions

- **Some LOFAR components** are not uniform in the sense of parallel computing capabilities. Running the final data reduction step (AWImager) at cluster does not bring significant time gain, particularly when compared to the previous BBS module
- **None of LOFAR modules** directly uses parallelization based on processes/threads. However, some of them are loaded with the script which run many instances of the given application simultaneously, each of them using one processor
- Installation is **complete and useful** now! (after 5 test grants)
- Achieved speed is not „very, very impressive” but **satisfactory**
- And we are very interested in **disc resources...**
- Compared to non-cluster users: need to use SLURM and module load commands **only!**
- Typical resources 40960 GB of storage, 300 000 h

# LOFAR Software – proposed improvements

- Preparing the configuration files for software installation performed with Intel compilers (GCC, Clang are officially supported) (M. Czuchry, K. Chyży – IT & users)
- Optionally: supporting the accelerative libraries: Intel MKL and future NVIDIA CUDA (GPGPU), Xeon Phi (M. Czuchry)
- Binary packages for the popular Linux distribution (RHEL, Ubuntu, Debian) (M. Czuchry, T. Kundera, E. Kuligowska – IT&users)
- Distributing LOFAR software through CVMFS (possibly in connection to point 2) (M. Czuchry)
- Clear documentation and ownership of the scripts (E. Kuligowska, T. Kundera)

**Finally, regarding the presence of not parallelized parts of the LOFAR pipeline - very interesting to consider its possibly parallelization**

# Thank You!



This research was supported by  
**Polish National Science Centre grant No. 2013/09/B/ST9/00599**  
and  
**PLGrid Plus (Domain - oriented services and resources of the Polish  
Infrastructure for Supporting Computational Science in the European  
Research Space)**