



UNIVERSITY OF TWENTE.

ITC GEOSPATIAL COMPUTING PLATFORM

dr.ing. Serkan Girgin MSc
s.girgin@utwente.nl

Center of Expertise in Big Geodata Science (CRIB) is a *horizontal facility* established in **March 2020** to **enable** the better use of **big geodata technology** in *education, research, and institutional strengthening* activities at **ITC**

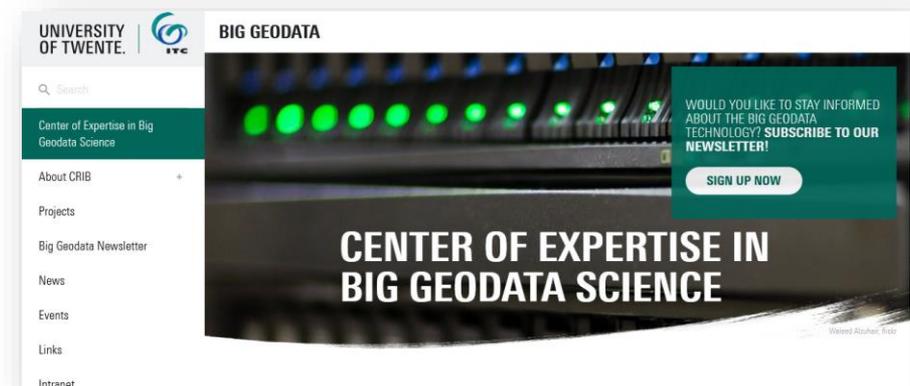
Mission

Collect, develop, and share **operational know-how** on big data technology to solve large-scale geospatial problems

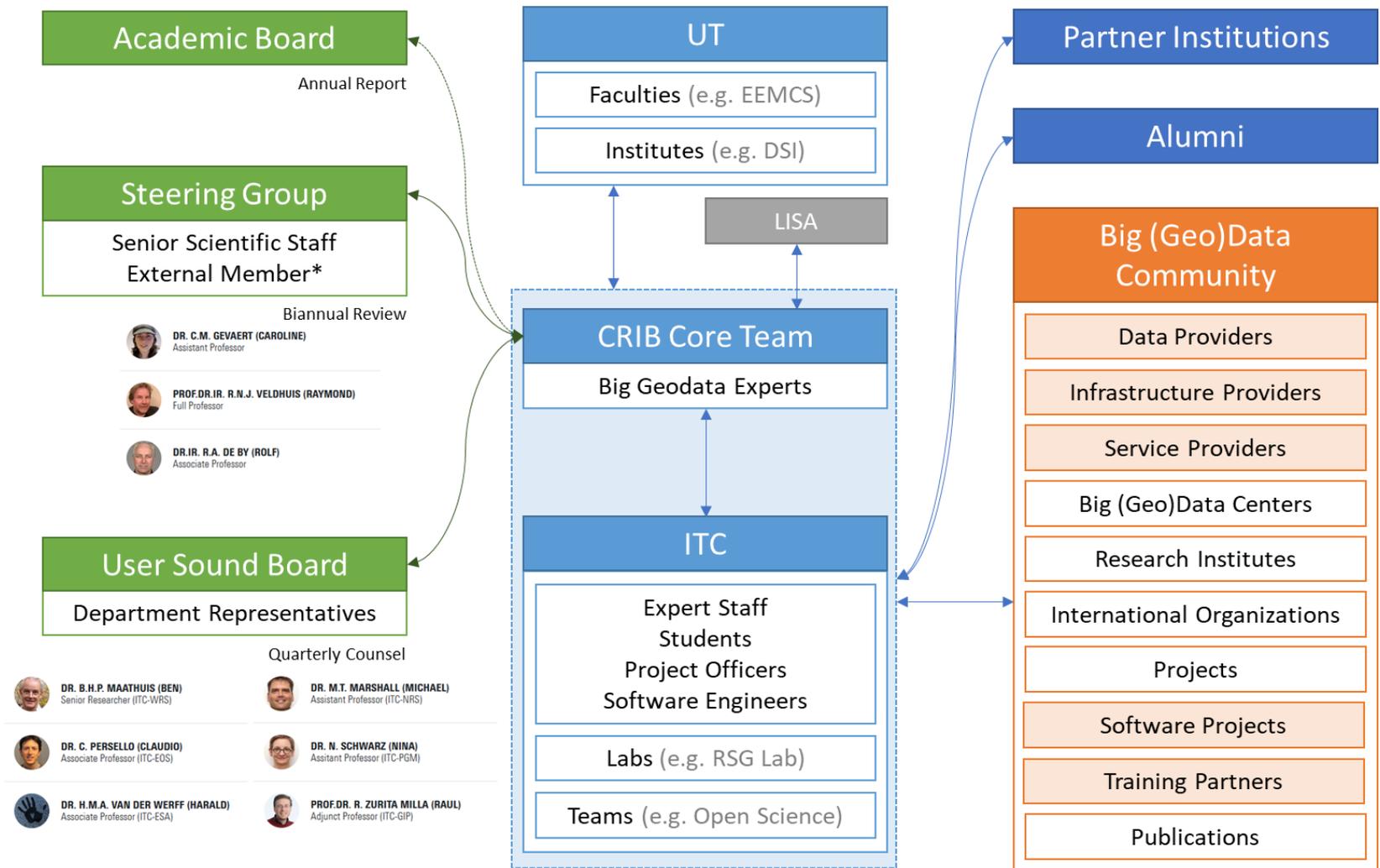
Vision

Position UT/ITC as a *globally renowned* center of excellence in **geospatial big data** science.

<https://itc.nl/big-geodata>



Structure



Activities

- Capacity and Knowledge Development
 - Improving expert knowledge, providing theoretical and hands-on training, facilitating community
- Infrastructure Development
 - Local and cloud-based infrastructure for big geodata computing
- Project Services
 - Consultancy and advisory services for integration and better use of big data technology, active support to research projects
- Monitoring and Networking
 - Monitoring recent developments in big (geo)data technology, networking with data providers, developers, research institutions
- Visibility
 - Ensuring high visibility of big-data related activities

Big Data Needs Assessment

- Key Findings
 - **Information** on big data technology should be actively communicated to the staff and students
 - **Proficiency** of the staff and students should be improved
 - Easy-to-use **computing infrastructure** should be made available
 - **Research projects** should be enhanced and improved with big data technology
 - Big geospatial data know-how should be transferred to **alumni and partners**

Modern computing infrastructure is necessary not only for big geospatial data analysis, but also for **geospatial data analysis** in general.

ITC Geospatial Computing Platform

- Designed to serve primary activities identified by the needs assessment:
 - **Self-learning**
 - **Exploratory research**
 - **Education**
- Provides **highly-available, easy-to-use** environment with good performance
 - **User-friendly** interface for data analysis and visualization
 - **Ready-to-use** scientific and geospatial analysis software
 - **Parallel and distributed computing** by using high-level frameworks
 - Computing by using special processing units (e.g., **GPU**)

Resources

- **16 x NVIDIA Jetson AGX Xavier** computing units (128 cores, 512 GB)
 - **8-core CPU** (NVIDIA Carmel ARMv8.2, 2.26 GHz)
 - **512-core GPU** (Volta architecture with 64 Tensor Cores)
 - **32GB memory** (DDR4x, 137 GB/s)
 - **500 GB – 1 TB local storage** (NVMe SSD, 3 GB/s)
- **Big data computing unit**
 - **2 x 8-core CPU** (Intel Xeon E5-2640, 32 threads, 2.60 GHz)
 - **24 TB local storage** (20 x 1.2 TB 2.5" 10K SAS 12 GB/s HDD, RAID 20+2)
 - **768 GB memory**
- **Hub server** (6-core, 192 GB)
- **200 TB storage** (0.2 PB)

We upgrade and repurpose **idle** resources and make them available on the platform for **common use**.



Architecture

<https://crib.utwente.nl>

- Based on open-source software (Ubuntu, Docker, JupyterHub, ...)
- Accessible through a **web browser** (No software installation is required)
- **No registration** is required (Login with UT credentials)
- Each user has an individual and isolated **working environment**
- Each user has access to all available* **unit resources**, including **GPU**
- Each user has access to all available* **cluster resources**
- **Replicated storage** with minimum two copies (Hardware failure protection)
- **Distributed storage** for big data processing (HDFS)
- Low energy footprint (10-30W per unit)

Key Features

<https://crib.utwente.nl>

- **Interactive notebook, terminal and remote desktop** access are available
- Multiple interactive languages are supported (Python, R, Julia, Octave, Go, ...)
- **Up-to-date and optimized software packages** are **ready to use** (No setup required)
- Users can install additional packages (e.g., Python, R packages)
- Distributed computing clusters are **ready to use** (Dask, Apache Spark)
- **Public** assets are shared by all users
- **Shared workspaces** allow assets to be shared by selected users
- Access can be granted to **external users**
- **User support** is available*
- Provided and maintained by **CRIB** at no extra cost (i.e., free PaaS)

Available Software

<https://crib.utwente.nl>



and hundreds more...

Additional Services

<https://crib.utwente.nl>



GeoServer

Open source server for sharing
geospatial data



MapServer

Open source platform for
publishing spatial data



PostgreSQL

Open source relational database



MariaDB

Open source relational database



GeoNode

Open source geospatial content
management system



Dataverse

Open source research data
repository software



Gitea

Open source lightweight code
hosting solution

In cooperation with ITC Research Data Team

Incubating! - BETA

Potential Use Cases

<https://crib.utwente.nl>

- Education
 - Computation platform for **courses** (Shared course workspaces)
- Research
 - **M.Sc.** / Ph.D. thesis studies
 - Collaborative (big) data analysis and visualization
 - Strengthen project proposals (Reduced budget needs for small projects, e.g., 50-100K EUR)
- Capacity Development
 - **Self-learning** (Cloud computing, distributed computing, GPU computing, Machine Learning, ...)
 - Computation platform for **training activities** (e.g., workshops)

Current Usage

- Operational since **January 2021**
- **271** registered users
- **5-20** concurrent users at a time
- Provided approximately **15,000** hours of multi-core/GPU computation
- Overall, quite **positive feedback** from a wide-range of use cases
- Several **courses** started to use the platform
- Several **projects** started to use or will use the platform (e.g., IDEAMAPS, EO Africa R&D, TMT+ Bangladesh)
- Several **project proposals** consider to utilize the platform
- Other **UT units** (e.g., DCC, BDSI) are interested in having similar platforms
- **LISA** decided to build a similar platform for UT-wide use

Quick Demo

<https://crib.utwente.nl>

The screenshot displays a JupyterLab environment with the following components:

- Left Panel:** A file browser showing a directory of Jupyter notebooks, including 'dask-pipeline-3-dask.ipynb' which is currently selected.
- Top Panel:** A code editor for 'dask-pipeline-3-dask.ipynb' containing the following code:

```
[3]: from dask.distributed import fire_and_forget

[4]: %%time
for i in range(100):
    img = client.submit(gather_from_detector, pure=False)
    img = client.submit(smooth, img)
    img = client.submit(np.fft.fftf2, img)
    future = client.submit(save, img, "file-" + str(i))
    fire_and_forget(future)
```

Below the code, a 'Dask Dashboard' window shows a 'Task Stream' visualization. The x-axis represents time from -59:40 to :01:40. The y-axis represents task IDs from 0 to 100. The visualization shows a dense stream of blue bars representing tasks, with some red and yellow bars indicating errors or warnings.
- Right Panel:** A code editor for 'cartopy.ipynb' containing the following code:

```
import matplotlib.pyplot as plt
import numpy as np

import cartopy.crs as ccrs
import cartopy.feature as cfeature

[2]: fig = plt.figure(figsize=[10, 5])
ax1 = fig.add_subplot(1, 2, 1, projection=ccrs.SouthPole)
ax2 = fig.add_subplot(1, 2, 2, projection=ccrs.SouthPole,
                      sharex=ax1, sharey=ax1)
fig.subplots_adjust(bottom=0.05, top=0.95,
                    left=0.04, right=0.95, wspace=0.02)

# Limit the map to -60 degrees Latitude and below.
ax1.set_extent([-180, 180, -90, -60], ccrs.PlateCarree)

ax1.add_feature(cfeature.LAND)
ax1.add_feature(cfeature.OCEAN)

ax1.gridlines()
ax2.gridlines()

ax2.add_feature(cfeature.LAND)
ax2.add_feature(cfeature.OCEAN)

# Compute a circle in axes coordinates, which we can use
# for the map. We can pan/zoom as much as we like - this
# is permanently circular.
theta = np.linspace(0, 2*np.pi, 100)
center, radius = [0.5, 0.5], 0.5
verts = np.vstack([np.sin(theta), np.cos(theta)].T)
circle = mpath.Path(verts * radius + center)

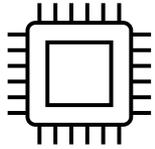
ax2.set_boundary(circle, transform=ax2.transAxes)

plt.show()
```

Below the code, two maps are displayed side-by-side. The left map shows the South Pole region with a grid and a circle. The right map shows the same region with a different projection and a circle.
- Bottom Panel:** A status bar showing system information: 'Mem: 371.13 / 32768.00 MB', 'Saving completed', 'Mode: Command', 'Ln 13, Col 1', 'English (American)', and 'cartopy.ipynb'.

Available on the platform at [public/platform/demo](https://crib.utwente.nl/public/platform/demo)

Contact



<https://crib.utwente.nl>



<https://itc.nl/big-geodata>



crib-itc@utwente.nl



[@BigGeodata](https://twitter.com/BigGeodata)



[Big Geodata Newsletter](#)

