



UNIVERSITY OF TWENTE.

USING FOSS TO DEVELOP AND OPERATE A 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 **geospatial cloud computing and big data technologies** in education, research, and institutional strengthening activities at **ITC**.

Mission

Collect, develop, and share **operational know-how** on cloud computing and big data technologies to solve large-scale geospatial problems.

Vision

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

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

The logo for CRIB (Center of Expertise in Big Geodata Science) features a stylized, circular emblem composed of overlapping, brush-stroke-like shapes in shades of light blue and teal. Below this emblem, the letters "CRIB" are displayed in a large, light gray, sans-serif font.

CRIB

Motivation

Not all research problems require **cloud computing** and **big data** technologies!

- ITC is heterogeneous with respect to interests and needs, and for some people the topic **is not and will not be** *interesting*.
- Even if there is *no apparent need or interest*, it is **still important** to have at least *a basic understanding* of the topic since it is **becoming** *a key component* of the geo-information and EO domain.
- This is an institutional priority.

Big Data Needs Assessment

- Key findings of the **status quo analysis** and **user needs assessment***
- **Information** on big data technology (BDT) 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 BDT
- BDT 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.

UT does not provide a common computing infrastructure

ITC did not have a common (geospatial) computing infrastructure

ITC departments have their own computing solutions

- Operation and management practices differ
- Usually managed by staff who have other primary roles
- Usually access is restricted

* [Girgin, S. \(2020\) Big Geodata at ITC: Status Quo and Roadmap](#)

User Groups

User Group	Needs	Actions Required
Everybody	<ul style="list-style-type: none"> • Basic information 	<ul style="list-style-type: none"> • Basic training • Periodic updates
Interested students	<ul style="list-style-type: none"> • Detailed education • Access to infrastructure 	<ul style="list-style-type: none"> • Specialized courses • Infrastructure
Interested staff	<ul style="list-style-type: none"> • Detailed information with case studies • Guidance for use • Access to infrastructure 	<ul style="list-style-type: none"> • Technology-specific training • Hands-on practices • Support for use • Infrastructure
Inexperienced staff	<ul style="list-style-type: none"> • Detailed information for problem solving • Guidance for implementation • Access to infrastructure 	<ul style="list-style-type: none"> • Tool-specific training • Hands-on practices • Problem-specific support • Infrastructure
Experienced staff	<ul style="list-style-type: none"> • Advanced training for early adaption and efficiency • Access to advanced infrastructure 	<ul style="list-style-type: none"> • Tool-specific advanced training • Technology-specific support • Advanced infrastructure

ITC Geospatial Computing Platform

- Designed to serve primary activities identified by the needs assessment:

- **Self learning**
- **Exploratory research**
- **Education**

- Design criteria

- **Highly available**
- **Ready to use**
- **User friendly**
- **GPU enabled**
- **Distributed-computing friendly**
- **Low cost**

24/7, no queue

Pre-installed software

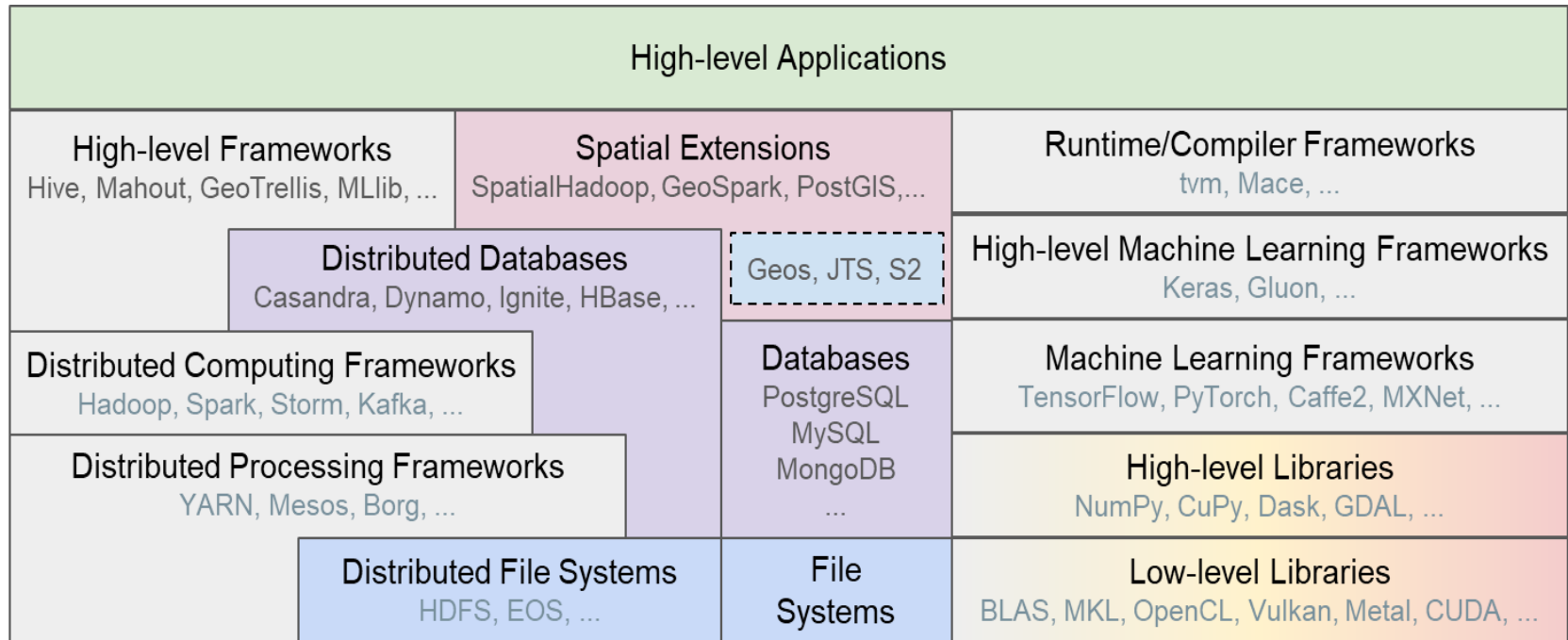
Interactive user interface

GPU for each user

Computing cluster

Feasible investment

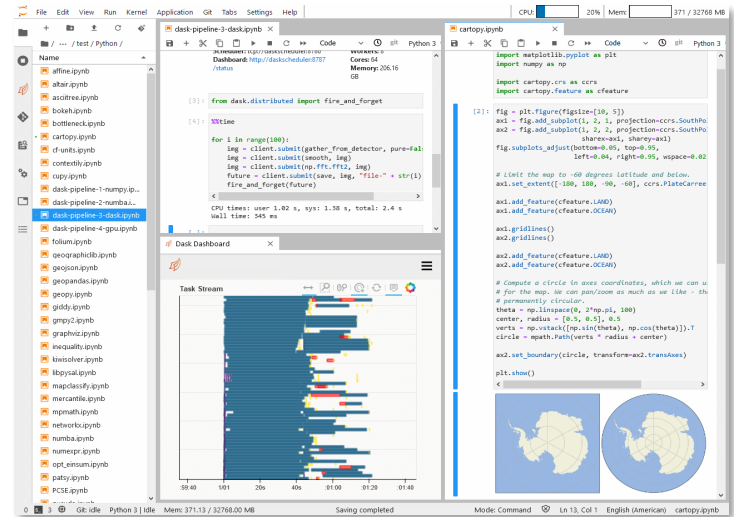
Target Big Geospatial Data Stack



Innovative Solution



NVIDIA Jetson AGX Xavier Cluster



JupyterHub on Docker Swarm

NVIDIA Jetson AGX Xavier

- **8-core CPU**
(NVIDIA Carmel **ARMv8.2**, 2.26GHz, **NVIDIA L4T**)
- **512-core GPU**
(**Volta Architecture** with 64 Tensor Cores)
- **32GB memory**
(256-bit LPDDR4x, 2133MHz, 137GB/s, **Unified**)
- **32GB storage**
(eMMC 5.1)
- **Dual Deep Learning Accelerator***
- **Vision Accelerator***
- **4x 4Kp60 video encoder**
(H.264/H.265)
- **2x 8Kp30 / 6x 4Kp60 video decoder**
(H.265)
- **Gigabit Ethernet**
(RJ45)
- **500 GB / 1 TB M.2 NVMe SSD**
(Samsung EVO 970 Plus, 3GB/s)

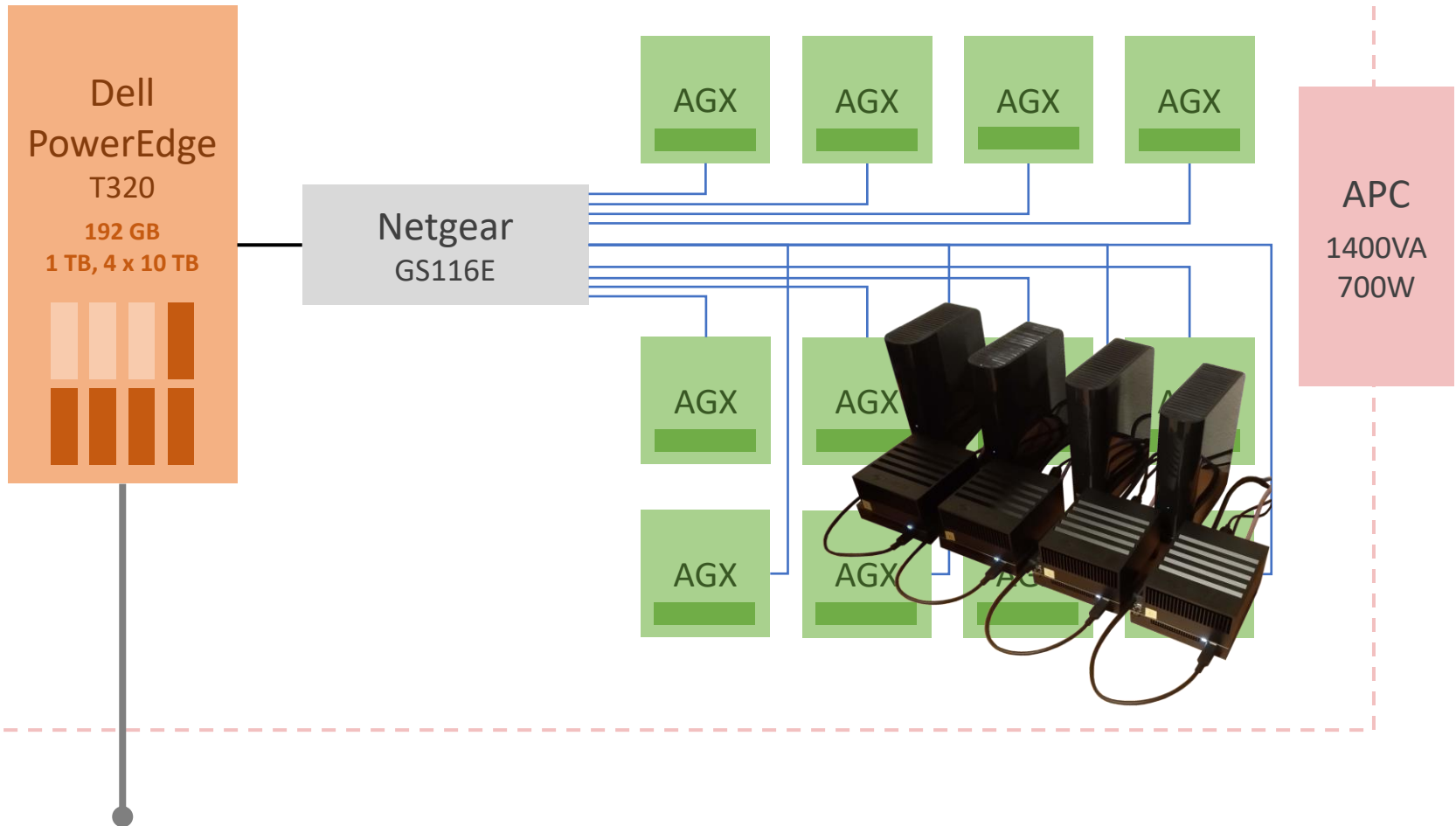


① <https://developer.nvidia.com/embedded/jetson-agx-xavier-developer-kit>

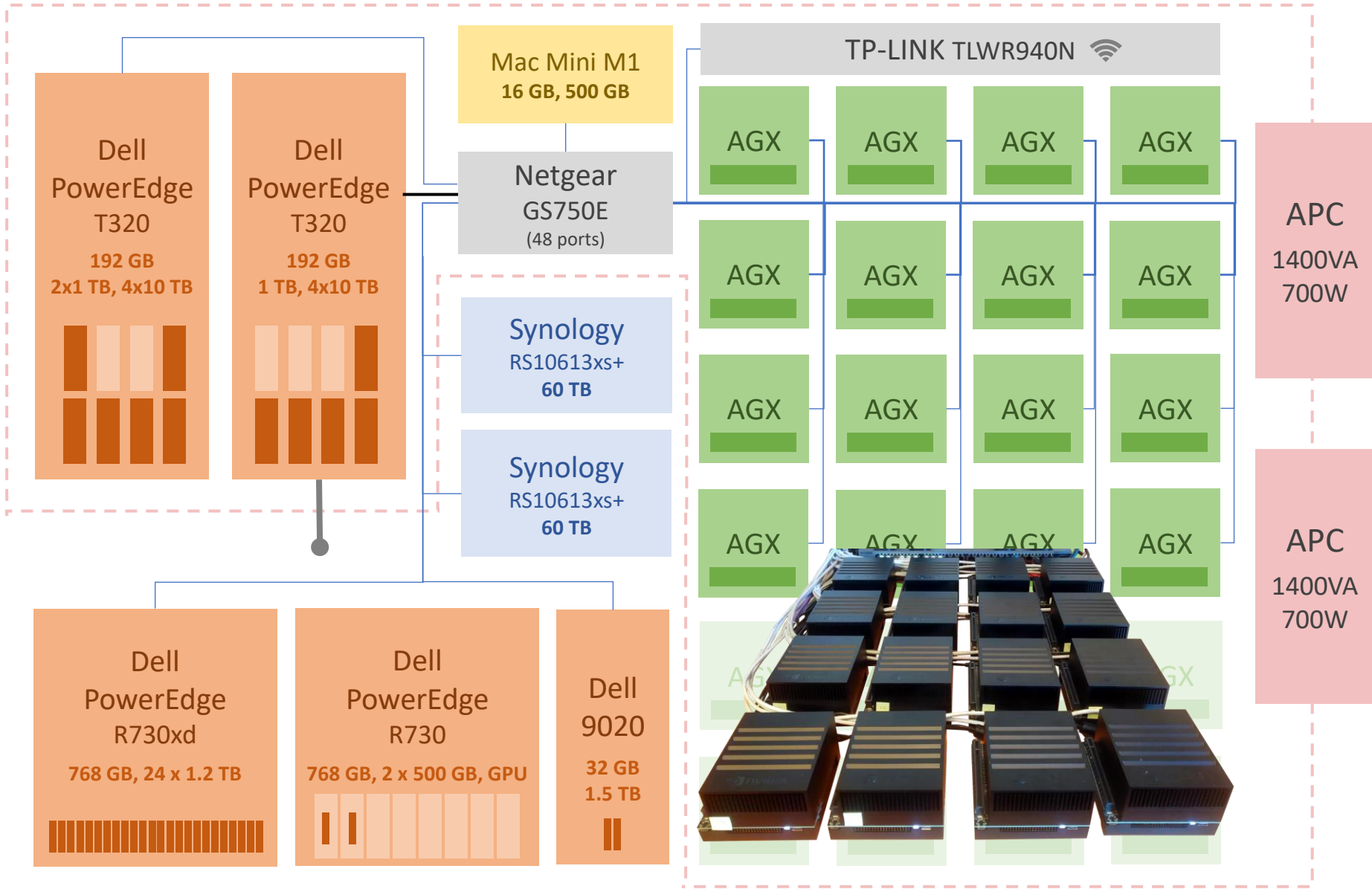
① https://elinux.org/Jetson_AGX_Xavier

Computing Cluster

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



Computing Infrastructure



Current Usage

- Operational since **January 2021**
- **555*** registered users
- **50*** shared workspaces for projects and courses
- **5-25*** concurrent users at a time
- Provided approximately **37,500*** hours of multi-core/GPU computation
- Overall, quite **positive feedback** from a wide-range of use cases
(4.61/5.00 according to the user survey)
- Other **UT units** (e.g., DCC, BDSI) are interested in having similar platforms
- LISA decided to build a similar platform for **UT-wide use**
Co-developed by CRIB, available at <https://jupyter.utsp.utwente.nl> (VPN)
- 4TUResearchData, TU Delft, FAO are interested to **have similar platforms**

* As of 20 October 2021

Platform as a Service

<https://crib.utwente.nl>

- Based on open-source software (Ubuntu, Docker, JupyterHub, JupyterLab, ...)
- Accessible through a **web browser** (No software installation or VPN are 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)
- Automatically scales and **balances workload** among the units
- Low energy footprint (10-30W per unit)

* Resource availability depends on resource usage of other active user



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)
- Different **architectures** and **OS-specific applications** are supported (e.g., Windows)
- Distributed computing clusters are **ready to use** (Dask, Apache Spark)
- **Public** assets are shared by all users (e.g., OSM Planet Data)
- **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)



Interactive Access



The screenshot displays a JupyterLab environment. On the left, a file browser shows a directory structure with files like '1-Hello.ipynb' through '8-dask.ipynb'. The main workspace is divided into three panes. The top pane shows a Python code cell with the following code:

```
[1]: import rasterio
from rasterio.plot import show

dir = '/data/public/GEODATA/Various-Netherlands/Aerial-Photogr...
img = rasterio.open(dir + '253000_470000.tif')
show(img)
```

The middle pane displays an aerial photograph of a residential area with a road and buildings. The axes are labeled with coordinates: x-axis from 253000 to 254000 and y-axis from 470000 to 470500.

The bottom pane shows a histogram titled 'Histogram' with the following code:

```
[2]: from rasterio.plot import show_hist
show_hist(img, bins=50, lw=0.0, stacked=False, alpha=0.3, histt...
```

The histogram shows the frequency distribution of pixel values, with the y-axis labeled 'Frequency' (scaled by 1e6) and the x-axis representing pixel values from 0 to 250.

On the right side, a 'Living Textbook' window is open, displaying the title 'Aerial survey' and a section titled 'Introduction'. The text in the introduction reads:

Aerial photographs are a major source of digital data; soft-copy workstations are used to digitize features directly from stereo pairs of digital photographs. These systems allow data to be captured in two or three dimensions, with elevations measured directly from a stereo pair using the principles of photogrammetry. Analogue aerial photos are often scanned before being entered into a soft-copy system, but with the advance of high-quality digital cameras this step can now be skipped.

In general, the alignment of roads and railways, lakes and water, and shapes of buildings are easily interpreted on aerial photographs - assuming that the scale of the photographs is not too small. Also, constructions such as dikes, bridges, air fields and the main types of vegetation and cultivation are mostly clearly visible. Nevertheless, numerous attribute data related to terrain features cannot be interpreted on aerial photographs: e.g. the administrative qualification of roads, sea and lake depths, functions of buildings, street names, and administrative boundaries. We will have to collect this information in the field or from existing data sets and maps (e.g. road maps, navigational charts or town plans).

Available Software

<https://crib.utwente.nl>



... and many more!

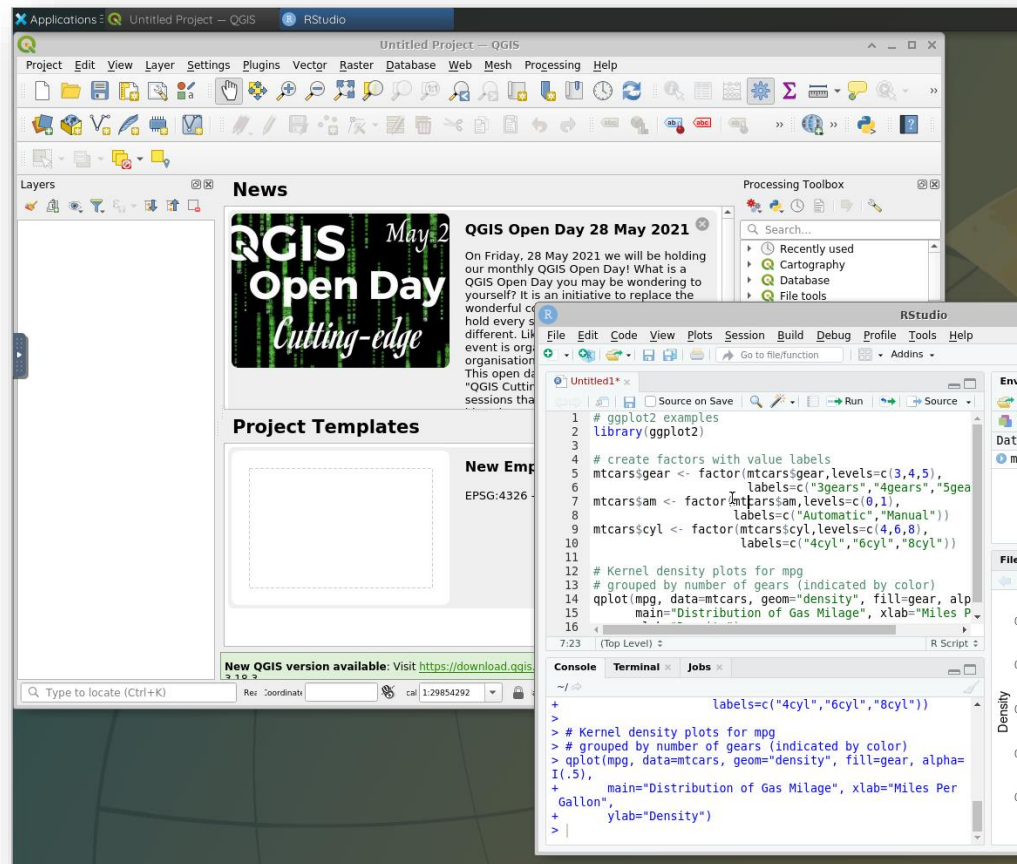
① Complete lists of more than **750+ Python** and **350+ R** packages are available at `/public/platform`

Available Desktop Applications

- QGIS
- GRASS GIS
- SAGA GIS
- SNAP
- **ILWIS 3***
- **ILWIS 4***
- VS Code
- PyCharm
- R Studio
- Netlogo
- GNU Octave
- **MATLAB***
- Glueviz
- Orange Data Mining
- Firefox



TigerVNC



Available 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



Open Data Kit

Open source platform to collect
data quickly, accurately, offline, and
at scale

Support Center

<https://crib.utwente.nl/support/>



[Support Center Home](#)

[Knowledgebase](#)

[Open a New Ticket](#)

[Check Ticket Status](#)

Search

Open a New Ticket

Check Ticket Status

Welcome to the CRIB Support Center!

In order to streamline support requests and better serve you, we utilize a support ticket system. Every support request is assigned a unique ticket number which you can use to track the progress and responses online. For your reference we provide complete archives and history of all your support requests.

Quick Access

- [Report a Problem](#)
- [Shared Workspace Request](#)
- [Course Workspace Registration with Canvas Integration](#)
- [External Account Request](#)
- [Account Removal Request](#)
- [Account Transfer Request](#)
- [Software Request](#)
- [Dataset Request](#)
- [Database Request](#)

Featured Questions

[How can I access to the platform?](#)

[Is it secure?](#)

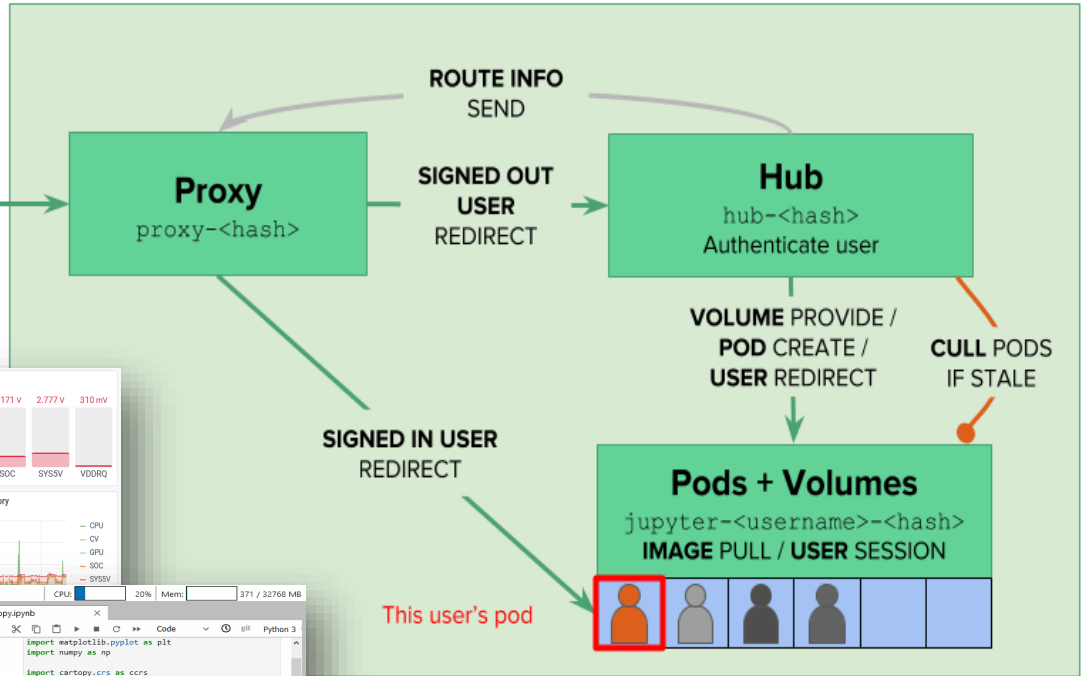
[How can I use the platform?](#)

[Which programming languages are supported on the platform?](#)

[Which libraries and packages are supported by the platform?](#)



User Access



The screenshot displays a JupyterLab environment with several components:

- System Monitoring:** Top row shows CPU Activity (32%, 78%, 3%, 4%, 6%, 28%, 3%, 2%), Temperature (41°C, 41°C, 42°C, 42°C, 41°C, 42°C), and Voltage (2.328V, 0mV, 0mV, 2.171V, 2.777V, 310mV).
- Activity History:** Middle row shows CPU Activity History, Temperature History, and Voltage History.
- File Browser:** Left sidebar shows a list of files and folders, including 'dask-pipeline-3-dask.ipynb' which is selected.
- Task Stream:** Bottom left shows a Gantt chart of task execution.
- Code Editors:**
 - `dask-pipeline-3-dask.ipynb`: Contains code for `from dask.distributed import fire_and_forget` and a loop submitting tasks.
 - `cartopy.ipynb`: Contains code for plotting a map using `cartopy` and `matplotlib`.

This user's pod

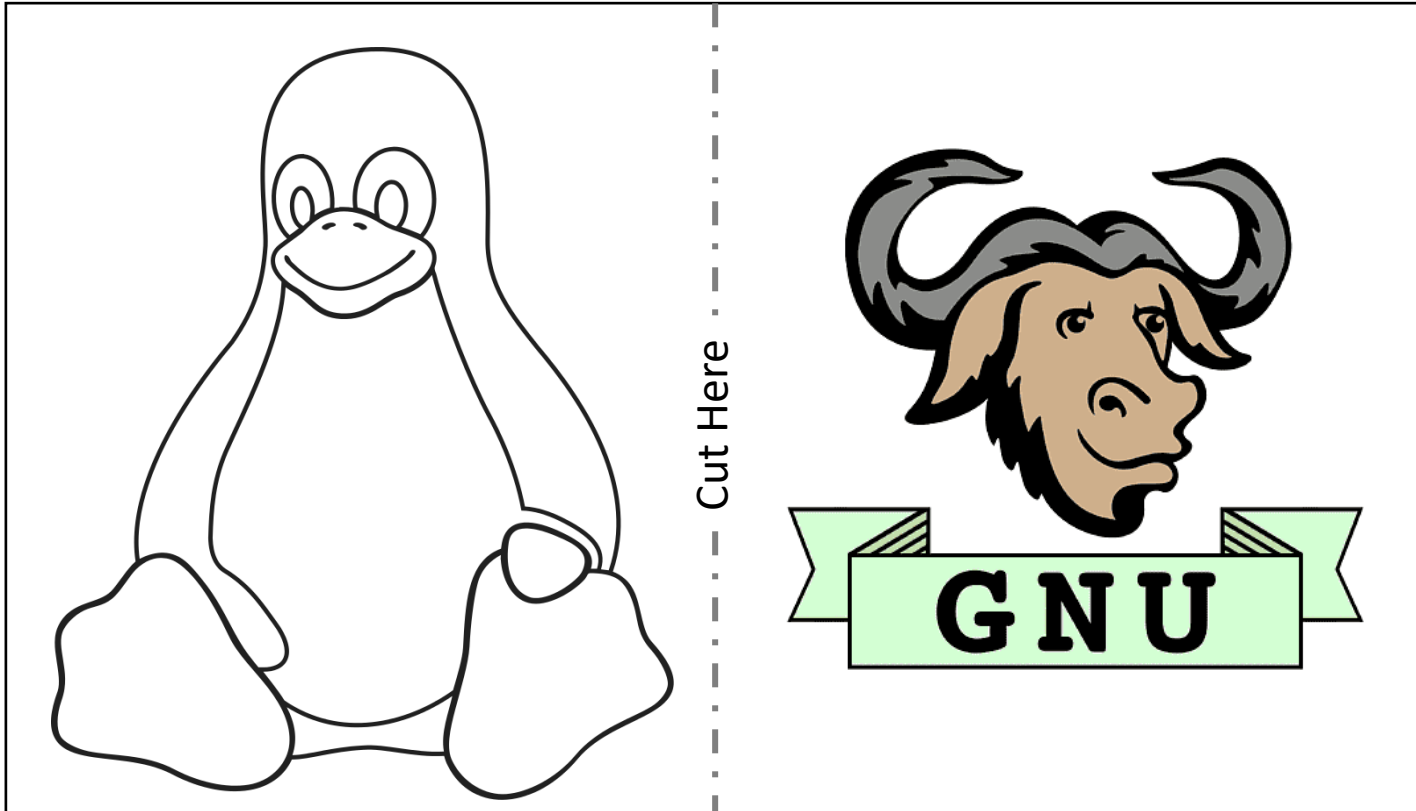


Computing Environment

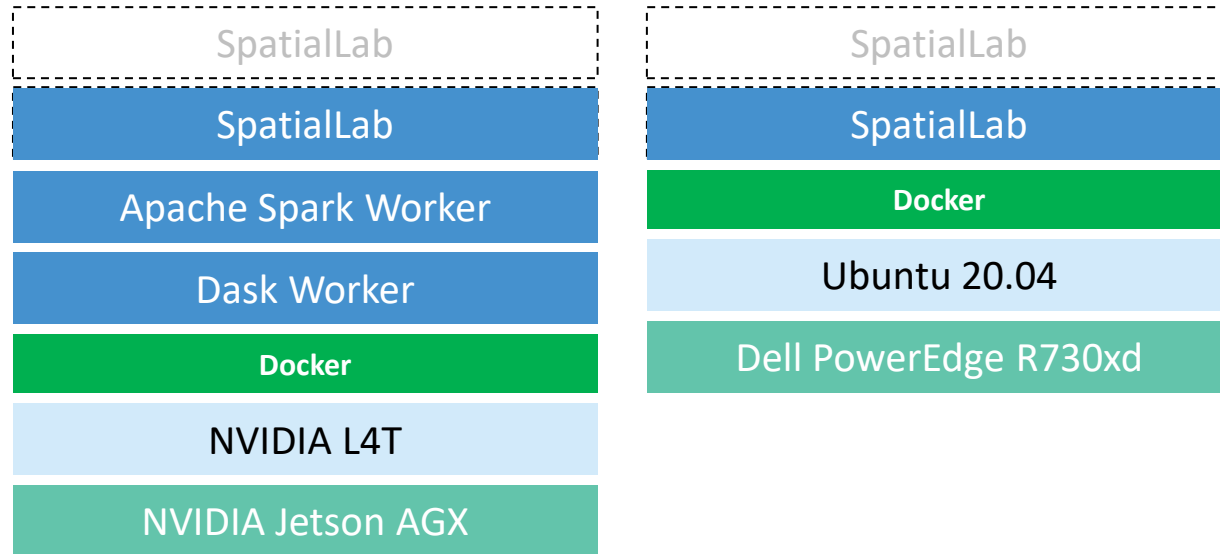


Container Image (common, volatile)

Binded Volumes (personal, permanent)



Container Orchestration



- **Minimum impact** on the host units
- **Custom-built** container images for better performance
- **SpatialLab** image: Ubuntu 20.04, All software
 - NVIDIA Jetson AGX : **21.0 GB**
 - Intel x86-64 : **42.9 GB**
 - Intel x86-64 GPU : **53.9 GB**

Operation and Maintenance

- On-demand and regular (monthly) **rolling updates** (2 restarts in 5 months)
- **Identical** working environment for ARM64 and Intel x86-64 units
- Automated shared workspaces for the **departments**
- Automated shared workspaces for **Canvas courses**
- Automated **notification** to newcomers
- **Daily** check for malicious threads
- **Continuous** resource and performance monitoring



Major Challenges

- Major upgrades
 - Python 3.6 → 3.8
 - JupyterLab 2.x → 3.x
- Major issues
 - Kernel crash with GPU
 - Slow-down under heavy use
- Major requests
 - (Ana)Conda
 - More computing accounts
 - Direct access

Quick Demo

<https://crib.utwente.nl>

The screenshot displays a JupyterLab environment with the following components:

- Left Panel:** A file browser showing a list of Jupyter Notebook files, 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.fft2, img)
    future = client.submit(save, img, "file-" + str(i))
    fire_and_forget(future)
```

Below the code, a 'Dask Dashboard' window is open, showing a 'Task Stream' visualization. The x-axis represents time from 59:40 to 01:40, and the y-axis shows task execution progress with various colored bars.
- 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.SouthPo
ax2 = fig.add_subplot(1, 2, 2, projection=ccrs.SouthPo
      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 u
# for the map. We can pan/zoom as much as we like - th
# 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)

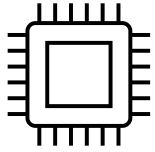
ax2.set_boundary(circle, transform=ax2.transAxes)

plt.show()
```

Below the code, two maps are displayed side-by-side. The left map shows a rectangular projection of the South Pole region, and the right map shows a circular projection of the same region.
- Bottom Panel:** A status bar showing system information: 'Git: idle Python 3 | Idle Mem: 371.13 / 32768.00 MB Saving completed Mode: Command Ln 13, Col 1 English (American) cartopy.ipynb'.

① Available on the platform at public/platform/demo

Contact



<https://crib.utwente.nl>



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



crib-itc@utwente.nl
s.girgin@utwente.nl



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



[Big Geodata Newsletter](#)

