GRASS GIS in the sky

GRASS GIS as high-performance remote sensing toolbox

Markus Neteler, Markus Metz, Moritz Lennert

https://grass.osgeo.org/
GRASS GIS Intro

- Geographic Resources Analysis Support System
- Open Source GIS
- developed since 1984, since 1999 GNU GPL
- Portable code (many operating systems, 32/64bit)
- Your GIS backbone – linkable to:

  ● Geographic Resources Analysis Support System
  ● Open Source GIS
  ● developed since 1984, since 1999 GNU GPL
  ● Portable code (many operating systems, 32/64bit)
  ● Your GIS backbone – linkable to:
GRASS GIS and Python

Using GRASS GIS from “outside” through “grass-session”

Finally easy use of GRASS GIS as a processing backend in Python!

Combine with GDAL, OTB, ...

```
#!/usr/bin/env python
# filename: test_session.py

from grass_session import Session
from grass.script import core as gcore

# create a new location from EPSG code (can also be a GeoTIFF or SHP or ... file)
with Session(gisdb="/tmp", location="location",
            create_opts="EPSG:4326"):
    # do something in permanent
    print(gcore.parse_command("g.gisenv", flags="s"))
    # {'GISDBASE': u'"/tmp"'},
    # 'LOCATION_NAME': u'"epsg3035"'},
    # 'MAPSET': u'"PERMANENT"'},

# create a new mapset in an existing location
with Session(gisdb="/tmp", location="location", mapset="test",
            create_opts=""):
    # do something in the test mapset.
    print(gcore.parse_command("g.gisenv", flags="s"))
    # {'GISDBASE': u'"/tmp"'},
    # 'LOCATION_NAME': u'"epsg3035"'},
    # 'MAPSET': u'"test"'},
```

pip install git+https://github.com/zarch/grass-session.git

# in future, the stable "grass-session" release
# will be available with:
# pip install grass-session
GRASS GIS 7.4 Release
https://trac.osgeo.org/grass/wiki/Grass7/NewFeatures74

- New: Get demo data at start screen
- GUI: data catalog improved
- Most of ortho-rectification brought back
- r.in.gdal + r.external: provide support for import of raster maps exceeding 90N or 90S or with an EW extent larger than 360 degrees
- r.out.gdal: possibility to create overviews which enhances the compatibility with other GIS software packages
- v.clip added for easy vector clipping
- ... (480 fixes and improvements with respect to 7.2.0)
GRASS GIS 7.4 Release

- Atmospheric correction updated with new satellites
- MODIS product processing made easy
Remote sensing in GRASS GIS: a long history

- Imagery submodule already available for GRASS 1.1 in 1986
- i.* modules fully integrated in version 3.0 in 1988
- Steady improvement and additions ever since
- From simple text terminal to X-monitors to a modern GUI
- Memory efficient modules
Remote sensing in GRASS GIS: pixel-based techniques

- Pixel-based tools for satellite and aerial imagery
- Most state-of-the-art methods implemented
- Complete toolchain from preprocessing to classification
- Many highly specialized tools
Remote sensing in GRASS GIS: object-based image analysis

- Complete toolchain from segmentation to classification
- Including
  - unsupervised segmentation parameter optimization
  - high performance object statistics calculation
  - module-level parallelization
- Recently created module for SLIC superpixel creation

Source: http://dx.doi.org/10.3390/rs9040358
Remote sensing in GRASS GIS: plus so much more!

- Suite of LiDAR data tools
- Suite for creation of orthophotos
- Current developments:
  - convolutional neural networks
  - cutlines for semantically sensitive tiling
  - etc, etc
- Constantly growing list of extensions
- Permanent work on performance improvements
High-performance computing

Example: NDVI time series

Harmonic Analysis of Time Series (HANTS)
High-performance computing

Components
- master with job/queue manager
- compute nodes
  - disk space per CPU core
  - RAM per CPU core
  - ideally one SSD per node

Parallelization:
several GRASS commands running at the same time
High-performance computing

Chunks for parallel processing

Temporal processing

- spatial chunks need to be mosaicked at the end
- or
- temporal chunks need overlap
High-performance computing

Spatial chunks

- computational region
  - North, South, West, East
  - rows, columns
- pre-defined regions,
on one for each chunk (tile)

*Alternative*

- create X tiles from one raster map (r.tile)
High-performance computing

Chunks for parallel processing

Spatial processing

- each time step as one chunk
  or
- spatial chunks
  not recommended
  $\rightarrow$ spatial discontinuities
High-performance computing

GRASS installation setup
- environmental variables
- paths

GRASS session setup
- variable GISRC for rc file
  - GRASS database
  - location
  - mapset

**script 1**
GRASS commands

**script 2**
1. create unique GISRC, unique mapset
2. run *script 1*
3. copy results
4. delete GISRC, mapset

**script 3**
job manager settings
run *script 2*

*always check return codes*
High-performance computing
temporary GRASS GIS session

script 2

Arguments: first and last time step

1. create temporary GISRC, mapset
   with unique names, using a pre-defined
   mapset template

2. run script 1 in temporary mapset

3. copy results, one for each time step, from
   temporary mapset to final mapset

4. delete temporary GISRC, mapset
   rm -rf $GISRC /path/to/temp_mapset

GISRC: name of file with GRASS variables
GISRC contents:
   GISDBASE: name
   LOCATION_NAME: name
   MAPSET: name

   directories

LOCATION_NAME: sub-directory of GISDBASE
MAPSET: sub-directory of LOCATION_NAME

export GISRC=/path/to/tmpgisrc
Ideally all on a SSD scratch disk

http://grass.osgeo.org/wiki
High-performance computing

job/queue manager

- Select / create a queue
- submit a job to a queue
- start a job when hardware resources are available
- redirect stdout and stderr to unique files

job with task(s), here task = script 2

job 1 : running
job 2 : running
job 3 : running
job 4 : running
job 5 : waiting
job 6 : waiting
job 7 : waiting
job 8 : waiting

waiting : no hardware resources available
High-performance computing

Collect results

- Copy results to one common GRASS mapset
  → this is the I/O bottleneck
  try nice / ionice

\textit{cluster file systems}
High-performance computing

MODIS Land Surface Temperature

temporal + spatial processing
High-performance computing

Most important

Have a good admin that fixes the system after you broke it