Actinia: Cloud based geoprocessing

Neteler, M., Gebbert, S., Tawalika, C., Bettge, A., Benelcadi, H., Löw, F., Adams, T., Paulsen, H.

Session 3 - Interactive processing and visualisation
Team of mundialis

- Founded in 2015, Bonn based
- Focus on geospatial data analysis and Earth observation
- Open Source developers

Dr. Fabian Löw (EO-Optical)  Hinrich Paulsen (Management & Finance)
Till Adams (Community & Sales)
Jan Orzekowski (IT)  Sören Gebbert (Cloud & Development)
Stefan Cerfontaine (Design)  Charlotte Eberz (Project management)
Anika Bettge (Development)  Dr. Hajar Benecaldi (EO-Radar)
Dr. Markus Metz (BigData & Algorithms)  Carmen Tawalika (Development)
Dr. Markus Neteler (BigData & Research)
Our toolbox – open source stack
open source geo-processing
(actinia – geoprocessing Engine

- cloud based geoprocessing API & engine
- focus on spatio-temporal analysis
- scalable (docker, Openstack, OpenShift, ...)
- open documented API (openAPI)
- data catalog
- user-owned data + EO archives (e.g. Sentinel, Landsat, ...)
- job management, Access Control Layer (auth)
- quota management

https://actinia.mundialis.de/
https://github.com/mundialis/actinia_core/
actinia – supported algorithms

- most GRASS GIS functionality can be covered
- wrappers for software like SNAP (GPT or SNAPPY)
- Python libraries like scikit-learn
- … (your choice)
- User Defined Functions (UDF)
architecture
Cloud based processing with actinia

- **Components**
  - actinia-core (on github)
  - actinia-GDI (yet unpublished)

- **Connectors**
  - actinia-GDI ↔ Geonetwork Open Source
  - actinia-core ↔ Geoserver (OGC services)
  - Object-storage and Shared File System

- **Plugins**
  - image classification
  - fibre optic cable planning
SentinelINDVIResponseModel: object

DESCRIPTION
The response of the Sentinel2A vegetation index computation

PROPERTIES
status: string
The status of the response
user_id: string
The id of the user that issued a request
resource_id: string
The unique resource id
process_log: Array<ProcessLogModel>
A list of ProcessLogModels
process_chain_list: Array<GrassModule>
The list of GRASS modules that were used in the processing

EXAMPLE
{
"message": "Processing successfully finished",
"process_chain_list": [
  {
    "flags": "g",
    "inputs": {
      "map": "ndvi"
    },
    "module": "r.univar",
    "outputs": {}
  },
  {
    "flags": "n",
    "inputs": {
      "map": "ndvi"
    },
    "module": "d.rast"
  }
],
"status": "2018-05-30 12:20:11.800060",
"http_code": 200
}
applications
Integration of external data sources

- Field data / measuring sites
- GIS layers
- EO layers
- SOS
- Time series
- Raster/Vector
- Forest change analysis
- Flooding SAR + DEM
  Source: ITZI

Classification

Time series processing

- OGC: WMST, WFS, ...

Reporting

Web Services

Cloud API

actinia overview
**actinia processing chains: Sentinel-1**

**Pre-processing of GRD mode:**

- `r.s1.grd.orb` (orbit)
- `r.s1.grd.rc` (radiometric correction)
- `r.s1.grd.SpFilter` (Speckle filter)
- `r.s1.grd.tc` (terrain correction)
- `r.s1.grd.tnc` (thermal noise removal)
- `r.s1.grd.bandmath`

**Processing INSAR (ongoing):**

- coherence, interferogram and all related preprocessing

**Pre-processing of SLC mode:**

- `r.s1.grd.tss` (split subswath of Sentinel-1 SLC mode)
actinia processing chains: Sentinel-2

Pre-processing:
- atmospheric correction of L1C data:
  - i.atcorr (6S based, GRASS GIS)
  - ARCSI (6S based)
  - sen2cor
- optionally reprojection

Processing (ongoing):
- vegetation indices
- time series analysis (gap-filling, anomalies, …)
- classification
- change detection

Integration with geospatial data (examples)
- zonal statistics
- fragmentation analysis
- ...

Optical image, Sentinel-2, 10m Spatial Resolution, natural colors
Station Nour, Drâa Valley – Morocco
actinia processing chains: topological space-time algebra

Available data types (each map is time-stamped):

- Space-Time Raster Dataset (STRDS) – e.g. daily climatic raster data or Sentinel bands
- Space-Time Raster-3D Dataset (STR3DS) – soil or atmospheric volumes
- Space-Time Vector Dataset (STVDS) – land cover/land use time series

Temporal relations:

<table>
<thead>
<tr>
<th>A in relation to B</th>
<th>B in relation to A</th>
</tr>
</thead>
<tbody>
<tr>
<td>A equivalent B</td>
<td>B equivalent A</td>
</tr>
<tr>
<td>A follows/adjacent B</td>
<td>B precedes/adjacent A</td>
</tr>
<tr>
<td>A overlaps B</td>
<td>B overlapped A</td>
</tr>
<tr>
<td>A after B</td>
<td>B before A</td>
</tr>
<tr>
<td>A during B</td>
<td>B contains A</td>
</tr>
<tr>
<td>A starts B</td>
<td>B started A</td>
</tr>
<tr>
<td>A finishes B</td>
<td>B finished A</td>
</tr>
</tbody>
</table>

=> Topology based spatio-temporal map algebra
actinia processing chains: topological space-time algebra

- new spatio-temporal topological operators (Gebbert et al, submitted)
- implementation in GRASS GIS, exposed through the REST API of actinia:

**t.rast.algebra – a topology based spatio-temporal map algebra**

- spatially: it uses the smallest common resolution
- temporally: it uses temporal-topological relations (instances can be buffered in time)
  - smallest granularity is 1 sec
  - common extent is calculated from temporal topological relations

→ it allows the application of algebraic expressions to time series of globally scattered satellite images

Future: support of image collections
actinia processing chains: topological space-time algebra

NDVI example: compute the NDVI of an “arbitrary” area on 8 dedicated CPU cores:

t.rast.algebra basename=ndvi -s nprocs=8 \
  expression="NDVI=(S2A_B08{-,equal|equivalent,1}S2A_B04) \
  {/,equal|equivalent,1} \n  (S2A_B08{+,equal|equivalent,1}S2A_B04)"

Computational time needed by t.rast.algebra to compute the NDVI from 100 Sentinel-2 scenes using 1, 2, 4 and 8 CPUs

Gebbert et al. submitted
Compute annual hydro-thermal coefficients (HTC) from 60 years of daily climate data

$$HTC = \frac{\sum P_{(T>10^\circ C)}}{\sum T_{(T>10^\circ C)} \cdot \frac{1}{10}}$$

T := daily temperatures,  
P := daily precipitation

```
t.rast.algebra "HTC = (D {+,contains,1} if(T >= 10, P, 0)) / (D {+,contains,1} if(T >= 10, T / 10, 0))"
```

HTC for 2003 and 2007

HTC of extreme events for droughts (HTC < 1) in red and humid years (HTC > 1.7) in blue

Gebbert et al. (submitted)
actinia shell: interactive cloud programming

ace - actinia command execution

The **ace** tool allows on an **actinia REST service** (e.g. https://actinia.mundialis.de/):

- execution of a single command, or a list of commands
- job management, ACL
- map layer query, creation and deletion of data

- processing in ephemeral and persistent databases
- generated outputs becomes available as a REST resource (URL)

**Tutorial:** https://github.com/mundialis/actinia_core/tree/master/scripts
actinia shell: interactive cloud programming

ace - actinia command execution

GRASS 7.7.svn (sentinel2):~/bin > ace --script ./ace_segmentation.sh
Resource status accepted
Polling: https://actinia.mundialis.de/api/v1/resources/markus/resource_id-a036fabe-a669-4799-97bd-f0e5bfb669e0
Resource poll status: running
Checking access to URL:
https://apps.mundialis.de/sentinel_2/IMG_DATA/R10m/T34TDR_20180919T093029_AOT_10m.tif
Resource poll status: running
... Resource poll status: running
Running executable i.segment with parameters ['group=T34TDR_20180919T093029_AOT_10m', 'threshold=0.25', 'radius=1.5' ... 2010_segment_25', 'goodness=T34TDR_20180919T093029_AOT_10m_seg_25_fit'] for 5.01212 seconds
... Resource poll status: running
Running executable i.segment with parameters ['group=T34TDR_20180919T093029_AOT_10m', 'threshold=0.25', 'radius=1.5' ... 2010_segment_25', 'goodness=T34TDR_20180919T093029_AOT_10m_seg_25_fit'] for 10.0254 seconds
Resource poll status: running
Export vector layer <T34TDR_20180919T093029_AOT_10m_segment_25> with format GeoJSON
Resource poll status: finished
Processing successfully finished
First Adopters
WANDEL: Water resources as a major driver of energy transformation at local and global level

WANDEL users can define their own workspace and process data
WANDEL: processing of S1 data in the cloud
The openEO H2020 project

**openEO** - a common, open source interface between Earth Observation data infrastructures and front-end applications

We use actinia as one of the backends of openEO

Source code on github

https://github.com/Open-EO/openeo-grassgis-driver
outlook & conclusions
Conclusions and what’s next

- actinia: a new proposed cloud based geoprocessing API & engine is available
- deployments
  - initial deployment in Deutsche Telekom cloud running
  - ongoing discussions with CODE-DE (for new BMVI mFund project “incora”)
  - relevant for DIAS?
- interfaces
  - REST API is online at actinia.mundialis.de (demo user)
  - Web: SHOGun framework
  - QGIS plugin planned