Big data ET models & benchmarking with distributed OSGEO tools

Yann Chemin

International Water Management Institute

Contents

CGIAR

Consultative Group for International Agricultural Research
Ratified on October 2nd, 2013
Full Open Access & Open Source research data and publication

- International Public Goods
- Public Domain
- Publications Open Access
- FOSS models and algorithms

2018: all 15 CG centres, already FOSS4G Lab: (gsl.worldagroforestry.org)

Overview

Evapotranspiration is the largest transiting quantity in the daily hydrological cycle along with rain. It is used by scientists and managers in:

- Irrigation systems performance
- Crop water productivity
- Water accounting
- Wetlands-agriculture interface
- Basin water uses quantification
- Climate change on water cycle & users

Overview

There are several types of evapotranspiration modeling methods:

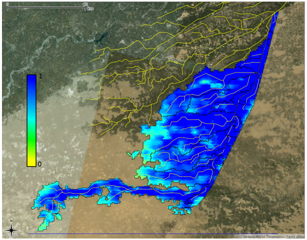
- Reference ET: Hargreaves, Penman-Monteith
- Potential ET: Priestley-Taylor, astronomical
- Actual ET: Thermodynamic/energy balance (mostly)

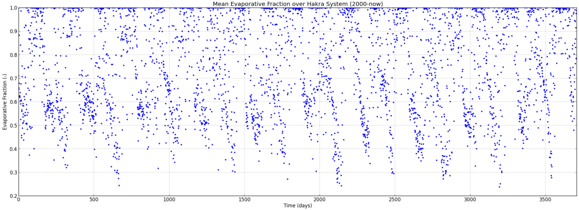
OSGEO tools

Equity of water use in irrigation systems

Irrigation water monitoring & management

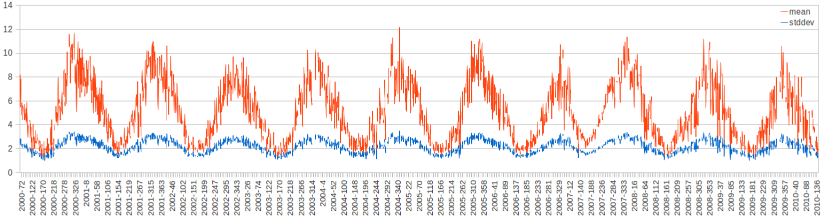
- Map: Uniform colour is equity of water distribution
- Graph: Irrigation system equity in time (mm/d, daily, 12 years)

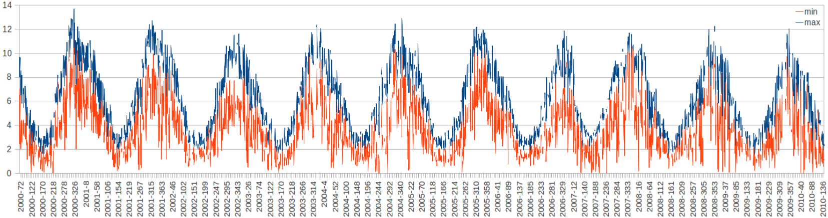




Crop water consumption in irrigation systems

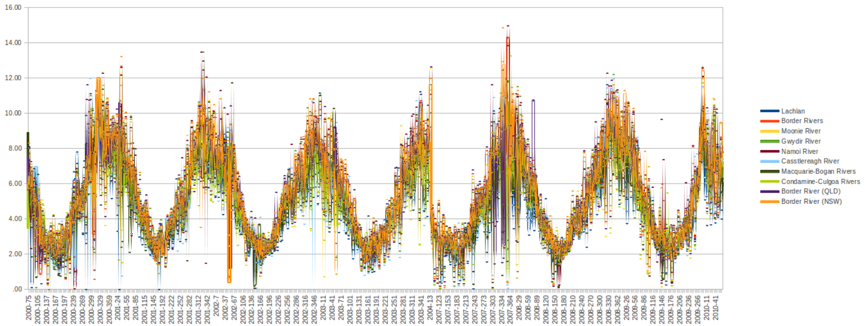
Actual evapotranspiration (mm/d, daily, 11 years)
for agricultural water performance management.





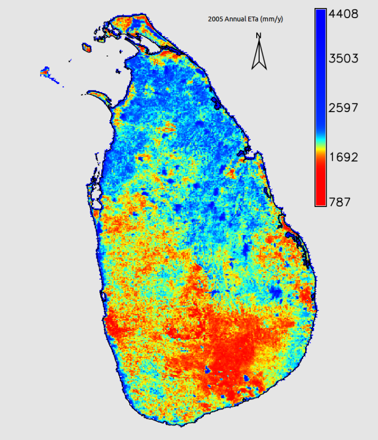
Water depletion in River Basins

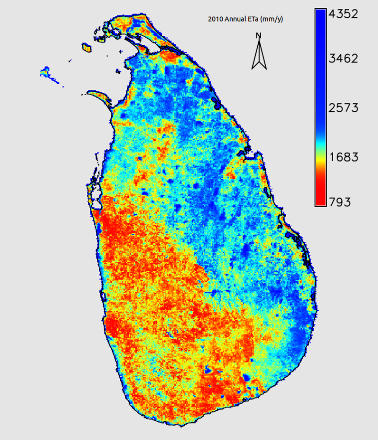
Actual evapotranspiration (mm/d, daily, 11 years)
for the Murray-Darling Basin (1M Km^2)



Evapotranspiration @ country level

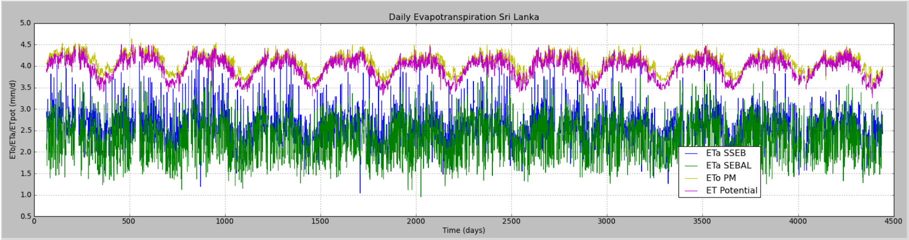
Actual evapotranspiration (365 days integrated)
for water resources monitoring & management.





ET models Benchmarking

Average ET for Sri Lanka, Daily 2000-2012 (mm/d, daily, 12.3 years)



Comparison

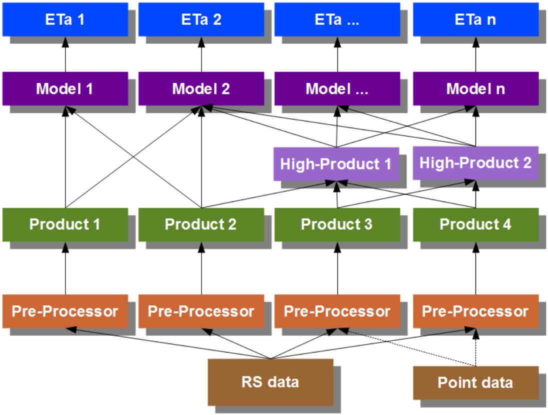
- ETo & ETpot (rad) are similar, expected.
- ETa models are not so similar, expected.
- ETo & ETpot (rad) are higher than ETa models, expected.

Chain processing

Chain processing has a fundamental impact on remote sensing work:

- Standardization limits bugs
- Less prone to human error
- Simpler parameterization access
- Permits to apply any number of modules to all target images
- Ensures maximum quality of generated images

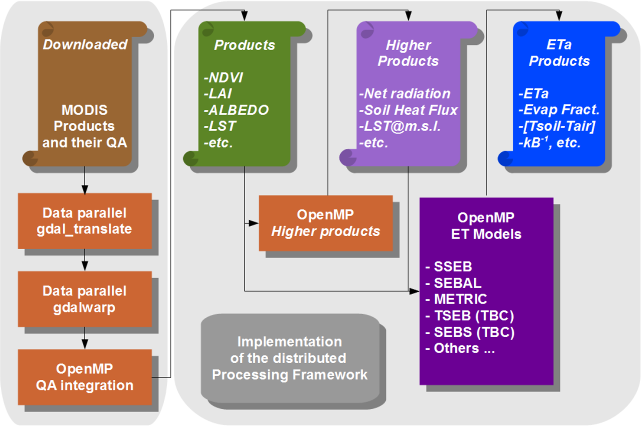
Blueprint



- GDAL+[C+OpenMP]
- GRASSGIS+pyGRASS+[C+OpenMP]

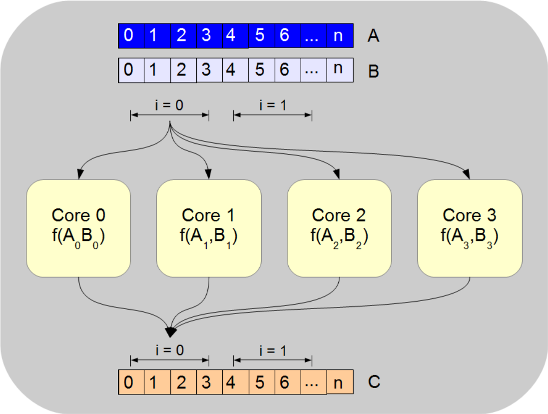
GDAL framework

GDAL+[C+OpenMP]

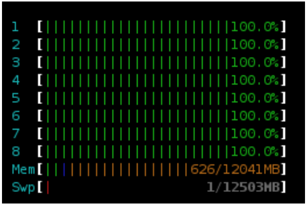


GDAL framework

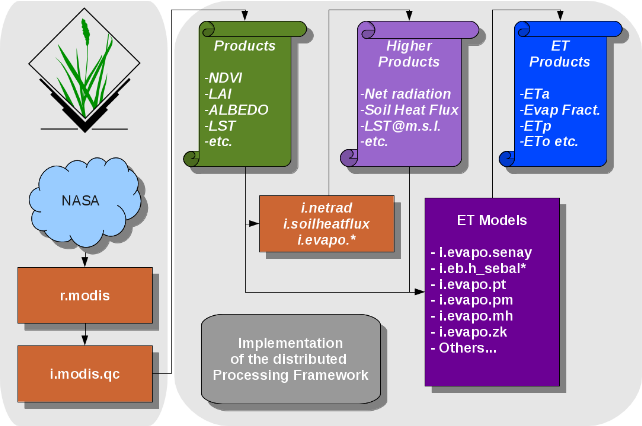
OpenMP Distribution



Cores



GRASS GIS framework



metaModule concept

pyGRASS: vertical integration of GRASS GIS modules

GRASS GIS modules: [C+OpenMP]

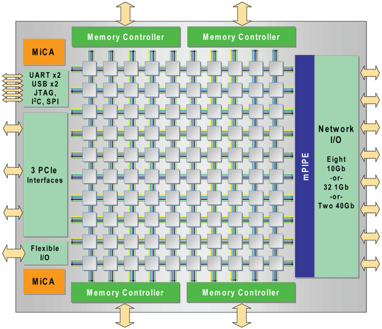
Summary for Landsat 7 pyGRASS MetaModule

```
from grass import script as g
from grass.script import setup as gsetup
gisbase=os.environ['GISBASE']
gsetup.init(gisbase,gisdb,location,mapset)
from grass.pygrass.modules.shortcuts import raster as r
from grass.pygrass.modules.shortcuts import imagery as i
from grass.pygrass.modules.shortcuts import display as d

r.mapcalc(expression="vis=18",overwrite=OVR)
r.in_gdal(input=L7f,output=L7r,flags="e",overwrite=OVR)
r.proj(input="dem",location="Myanmar",memory=10000,resolution=90.0,overwrite=OVR)

i.landsat_toar(input_prefix=pref,output_prefix=outpref,
        metfile=metadata[0],sensor=LSENSOR,quiet=QIET,overwrite=OVR)

i.atcorr(input=b, elevation="dem", visibility="vis", parameters=prm,
        output=b_out, flags="ra", range=[0,1],quiet=QIET,overwrite=OVR)

i.landsat_acca(input_prefix=b_in,output=b_clouds,overwrite=OVR)
r.mask(raster=b_clouds,flags="i",overwrite=True)

i.vi(red=b3,nir=b4,output=b_ndvi,viname="ndvi",quiet=QIET,overwrite=OVR,finish_=False)
i.albedo(input=b_in,output=b_albedo,flags="lc",quiet=QIET,overwrite=OVR,finish_=False)
i.emissivity(input=b_ndvi, output=b_emissivity,quiet=QIET,overwrite=OVR,finish_=False)
```

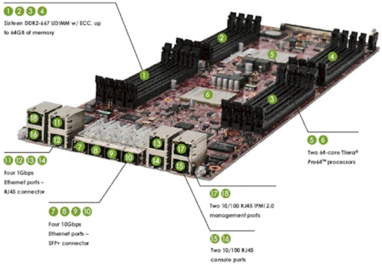
http://grasswiki.osgeo.org/wiki/Python/pygrass

Future: 128-cores from Tile-GX

64-core Tile-GX

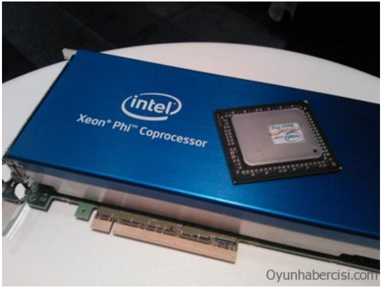


Dual Tile-GX on 1/2 rack board



Future: 120-cores from Xeon Phi

60-core Phi



Dual Phi in Gygabyte cage

Outlooks

- Multi-Cores Hardware (Tile-GX, Xeon Phi)
- Multi-GPU distribution (OpenCL, CUDA)
- Multi-CPU distribution (MPI)
- MODIS and Landsat archives under close pipe distance?
- Online: automatic, PyWPS, SOS/network ?

Conclusions

Distributed ET models benchmarking setup with OSGEO tools

- **GDAL:** C+OpenMP
- **GDAL:** Core-based scaling
- **GRASS GIS:** pyGRASS for metaModule
- **GRASS GIS:** pyGRASS finish_=False
- **GRASS GIS:** C+OpenMP inside modules
- **Targets:** MODIS (Terra/Aqua), Landsat (all), Aster

Thank You

IWMI
International
Water Management
Institute