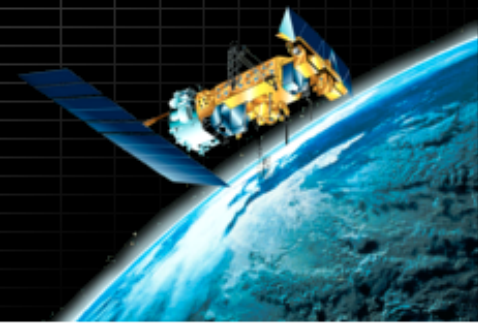


PPS v2014

Running in real-time

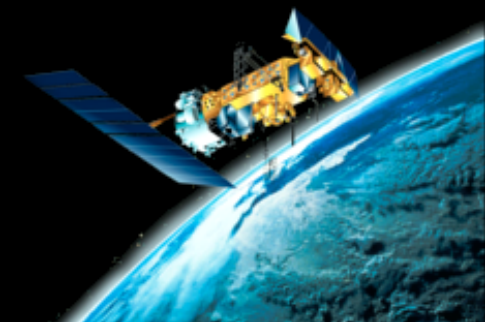


Adam Dybbroe



Outline

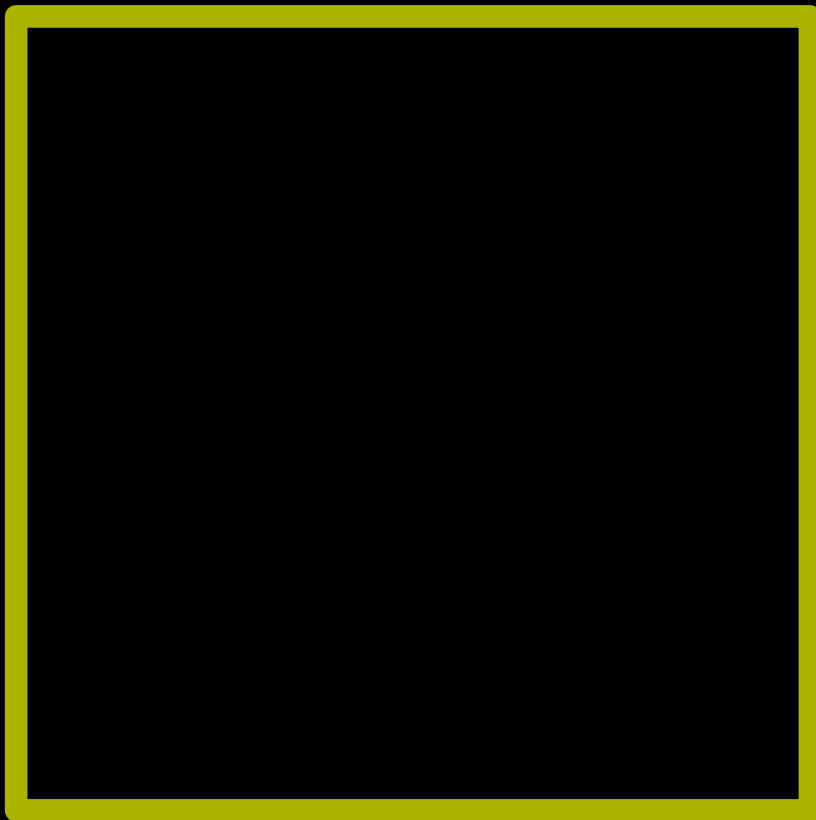
- Direct readout
- A simple setup using Inotify
- Real time processing at SMHI



Do you run PPS in real time today?

Yes

No



Local data - direct readout

- PPS runs on level 1 data as provided by
 - AAPP (AVHRR - NOAA and Metop)
 - CSPP (VIIRS - Suomi NPP)



Local data - direct readout

- Data needs to be located in a sub-directory containing only one swath:
 - `{sat}_{yyymmdd}_{hhmm}_{nnnnn}`
- The environment variable `SM_AAPP_DATA_DIR` is crucial here. Ex:

```
SM_AAPP_DATA_DIR=${DATA_DIR}/import/PPS_data/source
```



- Changing SM_AAPP_DATA_DIR on run time allows to store the level-1 data at different places:

```
$ ls -lt /san1/pps/import/PPS_data/source/ | more
```

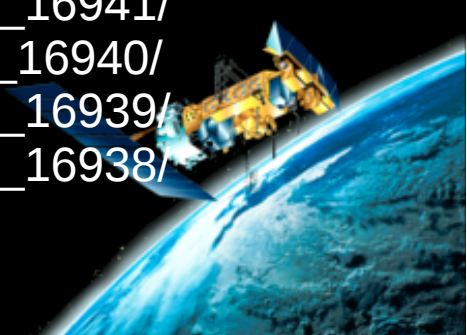
```
total 384
```

```
drwxrwxr-x 2 safusr.t safusr.t 4096 3 feb 21.30 metop01_20150203_2117_12348/  
drwxrwxr-x 2 safusr.t safusr.t 4096 3 feb 19.50 metop01_20150203_1935_12347/  
drwxrwxr-x 2 safusr.t safusr.t 4096 3 feb 18.10 metop01_20150203_1755_12346/  
drwxrwxr-x 2 safusr.t safusr.t 4096 3 feb 18.00 noaa15_20150203_1748_86983/  
drwxrwxr-x 2 safusr.t safusr.t 4096 3 feb 17.42 noaa18_20150203_1726_50024/
```

```
$ ls -lt /san1/polar_in/direct_readout/npp/lvl1/ | more
```

```
total 400
```

```
drwxrwxr-x 2 safusr.t safusr.t 40960 3 feb 12.50 npp_20150203_1229_16941/  
drwxrwxr-x 2 safusr.t safusr.t 36864 3 feb 11.11 npp_20150203_1048_16940/  
drwxrwxr-x 2 safusr.t safusr.t 36864 3 feb 09.31 npp_20150203_0910_16939/  
drwxrwxr-x 2 safusr.t safusr.t 24576 3 feb 07.51 npp_20150203_0733_16938/
```



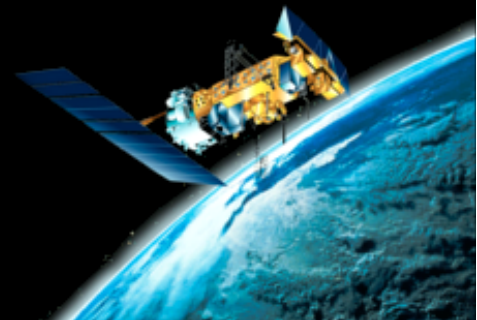
Running with Inotify - example

- Using `pyinotify` to detect incoming level-1 files on a directory (no NFS!)

pyinotify: <https://github.com/seb-m/pyinotify>

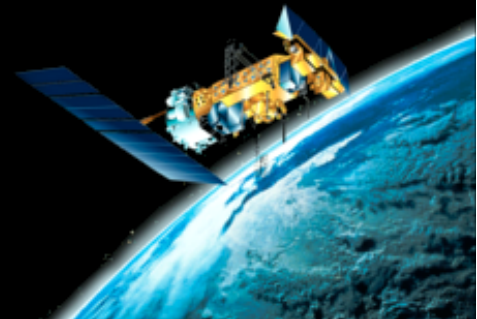
trollsift: <https://github.com/pnuu/trollsift>

PyTROLL



Running with Inotify - example

- Run pps via `ppsRunAllParallel.py` using `Popen` and a thin shell script to set environments



Running with Inotify - example

pps_run.sh

```
#!/bin/bash

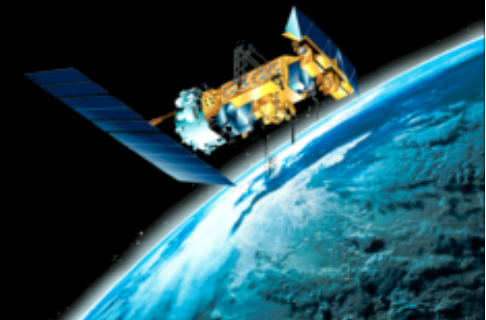
# $1 = Satellite
# $2 = Orbit number
# $3 = lvl1_dir
# $4 = lvl2_dir

. /local_disk/opt/acpg/cfg/.profile_pps

if [ "x$3" != "x" ]; then
    SM_AAPP_DATA_DIR=$3
    export SM_AAPP_DATA_DIR
    echo "Level 1 dir = $SM_AAPP_DATA_DIR"
fi

if [ "x$4" != "x" ]; then
    SM_PRODUCT_DIR=$4
    export SM_PRODUCT_DIR
    echo "Level 2 dir = $SM_PRODUCT_DIR"
fi

python /local_disk/opt/acpg/scr/ppsRunAllParallel.py -p $1 $2
```



Running with Inotify

pps_inotify.py – main part

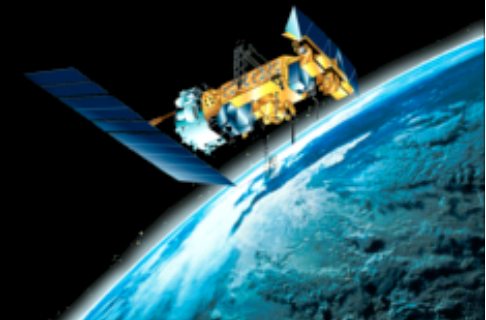
```
import pyinotify
from trollsift.parser import Parser

wm = pyinotify.WatchManager()
mask = pyinotify.IN_CLOSE_WRITE | pyinotify.IN_CREATE

class EventHandler(pyinotify.ProcessEvent):

    def process_IN_CLOSE_WRITE(self, event):
        print "Detected file:", event.pathname
        p = Parser(LVL1DIR +
                  "{directory}/hrpt_{platform}_{time:%Y%m%d_%H%M}_{orbit:05d}.11b")

        if p.validate(event.pathname):
            data = p.parse(event.pathname)
            run(data)
        else:
            print "Not a valid filename"
```

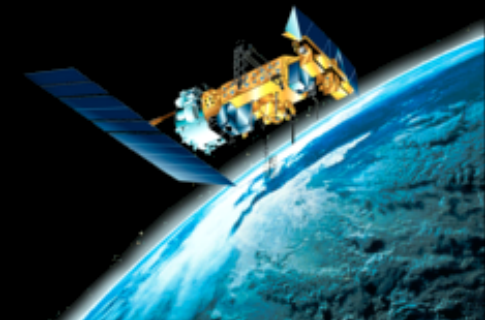


Running with Inotify

pps_inotify.py – main part

```
handler = EventHandler()
notifier = pyinotify.Notifier(wm, handler)
wdd = wm.add_watch(LVL1DIR, mask, rec=True, auto_add=True)

notifier.loop()
```



Running with Inotify

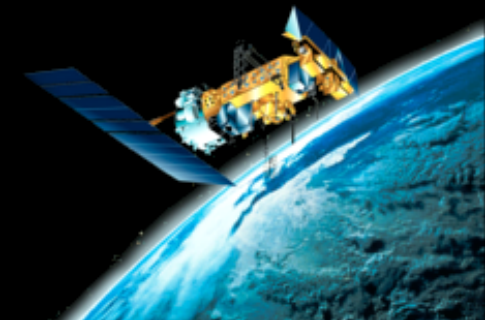
pps_inotify.py – pps-run part

```
PPS_SCRIPT = "./pps_run.sh"
from subprocess import Popen, PIPE, STDOUT
LVL1DIR = "/tmp/lvl1"
LVL2DIR = "/tmp/lvl2"

def run(scene):
    """Run the PPS main script"""

    cmdstr = "%s %s %s %s %s" % (PPS_SCRIPT, scene['platform'],
                                scene['orbit'], LVL1DIR, LVL2DIR)

    pps_proc = Popen(cmdstr, shell=True, stderr=PIPE, stdout=PIPE)
```



Running with Inotify

pps_inotify.py – pps-run part

```
while True:
    line = pps_proc.stdout.readline()
    if not line:
        break
    print(line)

while True:
    errline = pps_proc.stderr.readline()
    if not errline:
        break
    print(errline)

pps_proc.wait()

print("Ready with PPS level-2 processing on scene: " + str(scene))
```

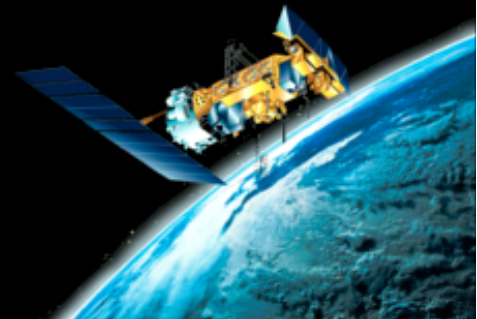


A more advanced setup

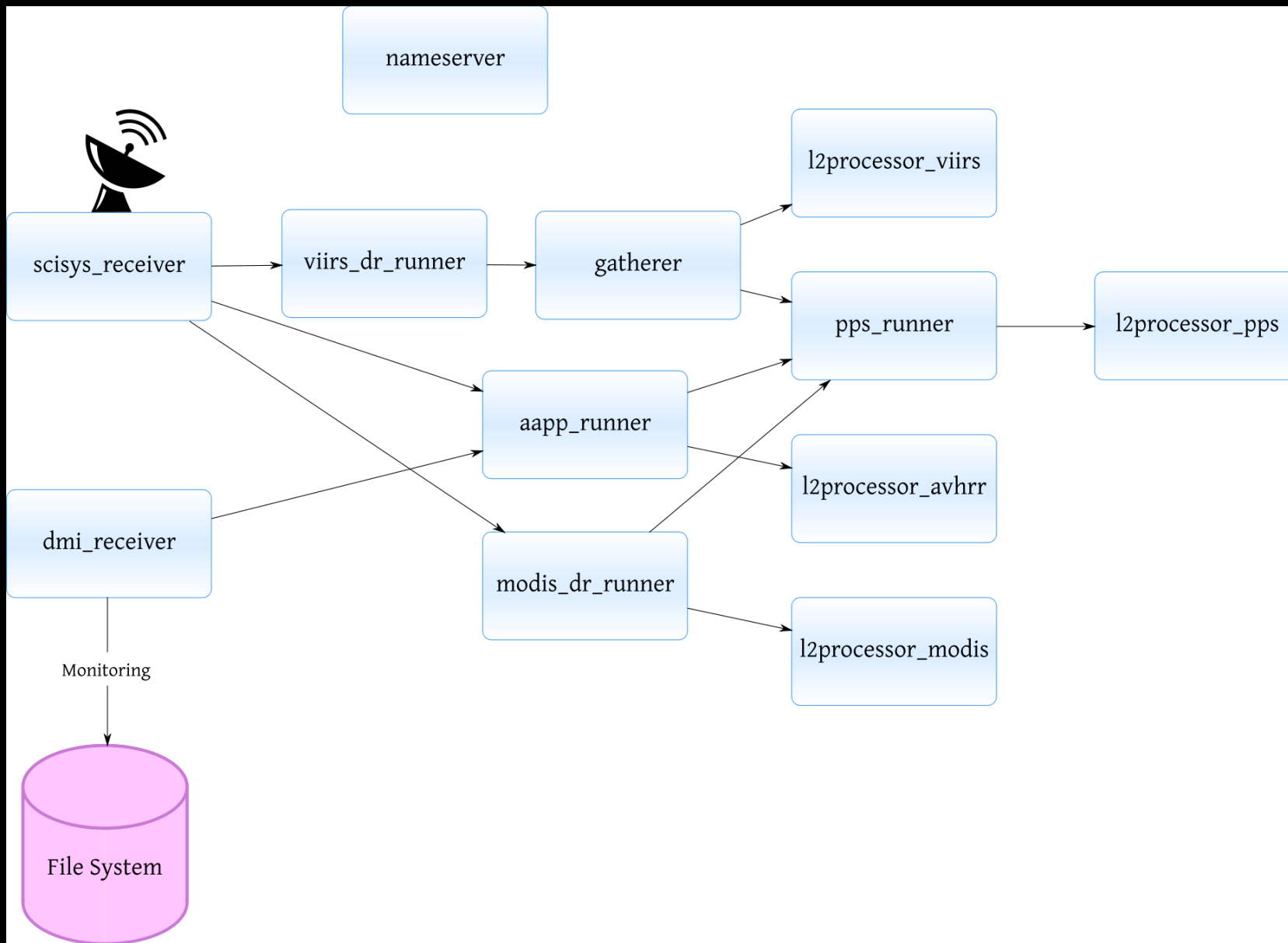
- Using messages to trigger processing
- Pytroll
- Supervisor

Pytroll: <http://pytroll.org/>

Supervisor: <http://supervisord.org/>



A more advanced setup



A more advanced setup

Supervisor status

Page refreshed at Wed Feb 4 14:15:44 2015

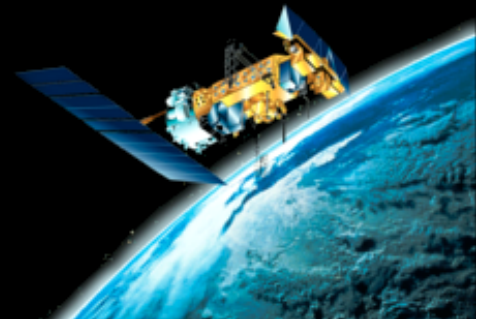
REFRESH
RESTART ALL
STOP ALL

State	Description	Name	Action
running	pid 10546, uptime 23 days, 4:42:41	aapp_runner	Restart Stop Clear Log Tail -f
running	pid 16371, uptime 31 days, 18:47:28	crashmailbatch	Restart Stop Clear Log Tail -f
running	pid 28855, uptime 25 days, 23:28:09	dmi_receiver	Restart Stop Clear Log Tail -f
running	pid 16380, uptime 31 days, 18:47:25	fatalmailbatch	Restart Stop Clear Log Tail -f
running	pid 27084, uptime 23 days, 22:33:26	gatherer	Restart Stop Clear Log Tail -f
running	pid 14224, uptime 16:44:48	l2processor_avhrr	Restart Stop Clear Log Tail -f
running	pid 11513, uptime 17:09:59	l2processor_modis	Restart Stop Clear Log Tail -f
running	pid 14736, uptime 16:38:22	l2processor_pps	Restart Stop Clear Log Tail -f
running	pid 7226, uptime 1 day, 2:29:09	l2processor_viirs	Restart Stop Clear Log Tail -f
running	pid 16142, uptime 18 days, 20:31:40	modis_dr_runner	Restart Stop Clear Log Tail -f
running	pid 8096, uptime 27 days, 6:04:21	nameserver	Restart Stop Clear Log Tail -f
running	pid 20148, uptime 16 days, 4:15:03	pps_runner	Restart Stop Clear Log Tail -f
running	pid 12395, uptime 9 days, 5:29:00	scisys_receiver	Restart Stop Clear Log Tail -f
running	pid 7227, uptime 21 days, 4:38:56	viirs_dr_runner	Restart Stop Clear Log Tail -f



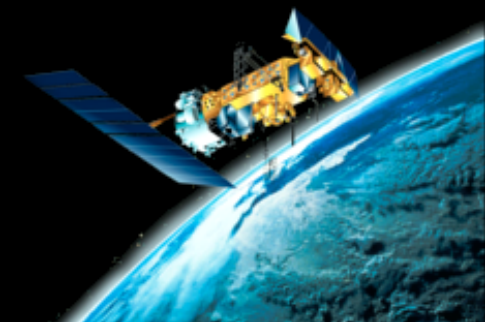
Summary

- PPS is a library, including high and low level scripts
- No Task Manager anymore
- A simple example running PPS in real time using Inotify has been shown
- A more advanced setup based on **PyTROLL** is used at SMHI and has been briefly sketched



Summary

- PPS can be run on Metop GDS in the same way as shown here on Direct Readout data



Questions?

