

Use Cases of CNAF Farm



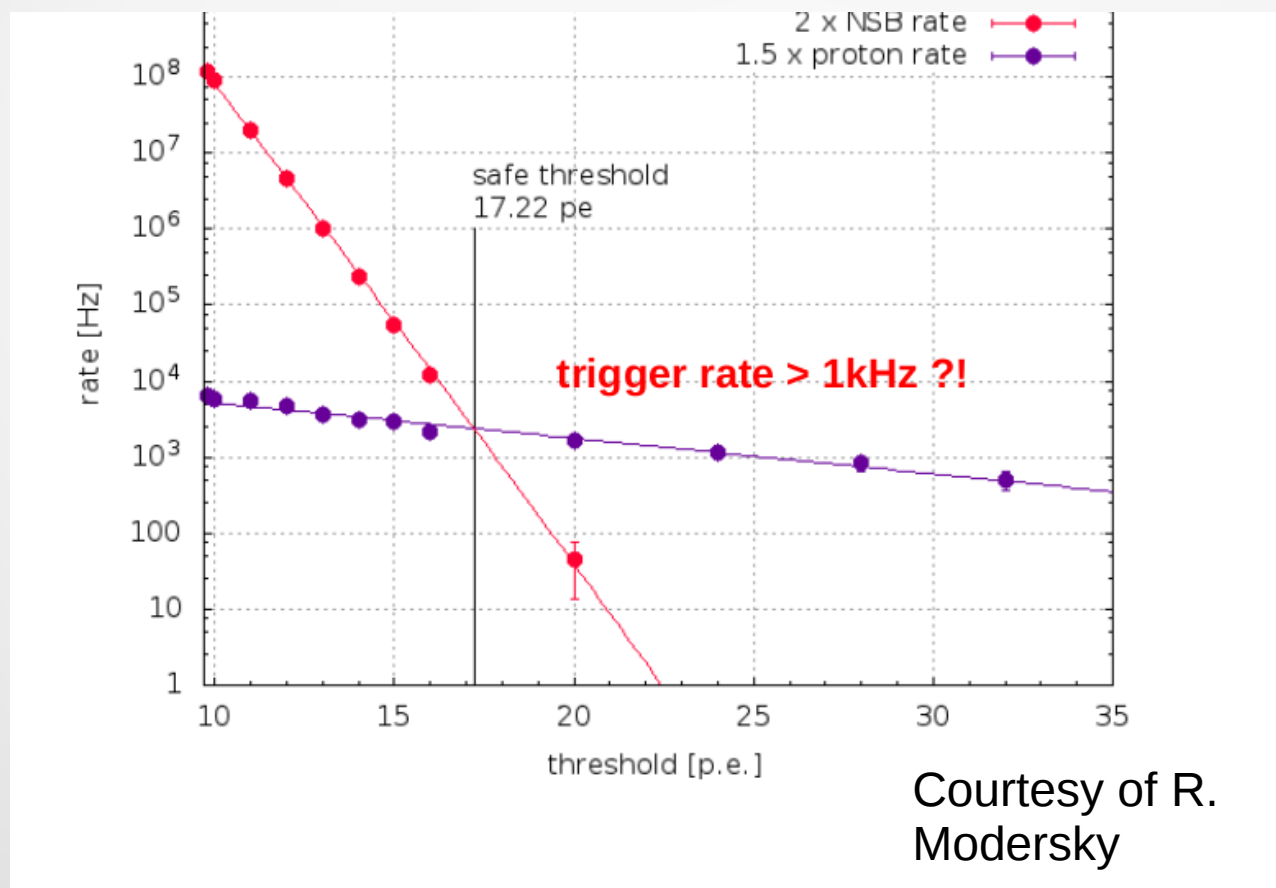
Ciro Bigongiari
INAF - OATo
bigongiari@oato.inaf.it

Topics

- **Discriminator Threshold Optimization (for ASTRI)**
- **Optimization of Calibration Data simulation (Ongoing)**

Discriminator Threshold optimization

- Safe trigger:
 - $\text{NSB rate} \times 2.0 = \text{Proton rate} \times 1.5$
 - Trigger rate below the DAQ maximum rate



Discriminator Threshold optimization

So we need:

- To simulate the detector response for many different discriminator threshold both for:
 - Pure NSB events
 - Pure proton events
 - Proton + NSB events – Crosscheck

Unfortunately `sim_telarray` allows only one discriminator threshold for each run

- Many runs
- Quite some events for each run due to the high rejection power
 - Different NSB levels
 - Different trigger strategies (Majority 4 & Majority 5)
- Highly parallelizable task
 - Use CNAF farm

CNAF

- Login to CNAF
 - `ssh -Y -l bigongia bastion.cnaf.infn.it`
 - You can't do nearly anything here
- Move to the UI
 - `ssh -Y -l bigongia ui-tier1.cr.cnaf.infn.it`
 - Not so much space in your home directory
 - Move to local storage area
 - `cd /storage/gpfs_data/ctalocal/bigongia/`

Pure NSB events

- Go to the directory where all the relevant stuff is:
 - `cd cta_meeting`
- All the scripts are readable (You can copy them if you want :))
- Have a look at the script:
 - `nsb_example_job.sh`
 - Three input parameters:
 - Single pixel NSB photo-electron rate [GHz]
 - Discriminator threshold [pe]
 - Working directory
- Copy it to your preferred directory
 - Try it:
 - `./nsb_example_job.sh 0.046 1.0 /storage/gpfs_data/ctalocal/bigongia/cta_meeting/`

The script

- The script creates some sub-directories if needed
- Copy relevant files
- Write another script
- Submit it to the queue system
- `JOBNAME=`bsub -q $QUEUE $QOPTION < $SCRIPTDIR/job_script_dir/$TMPSCRIPT.lsf``

```
\${SIMTELSYS}/sim_telarray \${extraopt} -c "\${cfgfile}" \
-C DISCRIMINATOR_THRESHOLD=\${DISCRIMINATOR_THRESHOLD} \
-C NIGHTSKY_BACKGROUND=" all:\${NIGHTSKY_BACKGROUND}" \
-C OUTPUT_FORMAT="0" \
-C SAVE_PHOTONS="2" \
-C "MAXIMUM_EVENTS=1000" \
-C "histogram_file=\${hdata_file}" \
-o \${output_file} \
-i \${input_file} >& output_${OUTPUTID}.dat
```

#

Monitor your job

- Monitor your jobs with:

- **bjobs**

- | JOBID | USER | STAT | QUEUE | FROM_HOST | EXEC_HOST | JOB_NAME | SUBMIT_TIME |
|----------|---------|------|-------|-----------|-------------|------------|--------------|
| 81740244 | bigongi | RUN | cta | ui-tier1 | wn-206-04-2 | *4_1.0.lsf | Jun 22 15:13 |

- Another useful command is:

- **busers**

- | USER/GROUP | JL/P | MAX | NJOBS | PEND | RUN | SSUSP | USUSP | RSV |
|------------|------|-----|-------|------|-----|-------|-------|-----|
| bigongia | - | - | 1 | 1 | 0 | 0 | 0 | 0 |

- You can check the status of the queues with:

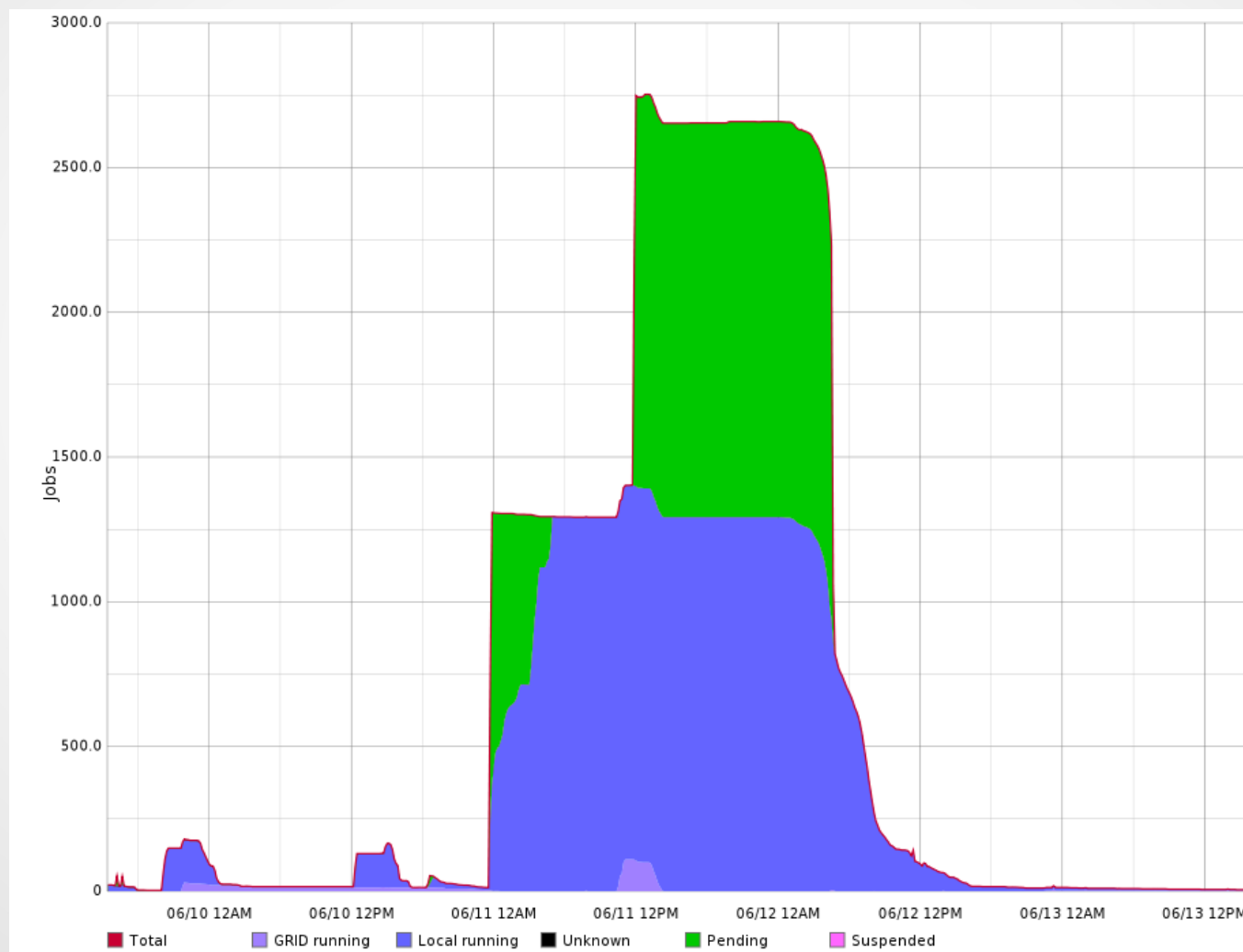
- **bqueues**

- | | | | | | | | | | | |
|-------|----|-------------|------|---|---|---|-----|-----|-----|---|
| lhcf | 40 | Open:Active | 400 | - | - | - | 100 | 0 | 100 | 0 |
| cta | 40 | Open:Active | 2000 | - | - | - | 1 | 0 | 1 | 0 |
| belle | 40 | Open:Active | 1600 | - | - | - | 783 | 245 | 538 | 0 |
| panda | 40 | Open:Active | 300 | - | - | - | 0 | 0 | 0 | 0 |

- Otherwise you can use the Web interface at:

- <http://tier1.cnaf.infn.it/monitor/index.html>

Monitoring Example



Counting the triggered events

- Go to your working directory
 - `cd /storage/gpfs_data/ctalocal/bigongia/cta_meeting/`
 - `cd output_files`
- You can simply ignore .hdata.gz and .simtel.gz file
- Have a look at .log file
 - There is one line with “has triggered” for each triggered event
 - There is one line with “Event end “ for each simulated event
- You can count the triggered events and showers with:
 - `grep -c “has triggered” MyFile.log`
 - `grep -c “Event end” MyFile.log`

Scan Discriminator Threshold

- We need to repeat the same procedure for many different discriminator thresholds
- Have a look at the script:
 - `nsb_threshold_scan.sh`
- Launch many jobs, one for each discriminator threshold
- Run it :
 - `nsb_threshold_scan.sh 0.046 /storage/gpfs_data/ctalocal/bigongia/cta_meeting/`

bjobs output example

```
[bigongia@ui-tier1 cta_meeting]$ bjobs
```

JOBID	USER	STAT	QUEUE	FROM_HOST	EXEC_HOST	JOB_NAME	SUBMIT_TIME
81760058	bigongi	RUN	cta	ui-tier1	wn-206-04-2	*046_1.lsf	Jun 22 17:40
81760061	bigongi	RUN	cta	ui-tier1	wn-206-07-2	*6_1.5.lsf	Jun 22 17:40
81760062	bigongi	RUN	cta	ui-tier1	wn-201-01-3	*046_2.lsf	Jun 22 17:40
81760064	bigongi	RUN	cta	ui-tier1	wn-200-03-1	*6_2.5.lsf	Jun 22 17:40
81760066	bigongi	RUN	cta	ui-tier1	wn-201-07-3	*046_3.lsf	Jun 22 17:40
81760068	bigongi	RUN	cta	ui-tier1	wn-200-08-0	*6_3.5.lsf	Jun 22 17:40
81760070	bigongi	RUN	cta	ui-tier1	wn-200-08-0	*046_4.lsf	Jun 22 17:40
81760072	bigongi	RUN	cta	ui-tier1	wn-201-07-1	*6_4.5.lsf	Jun 22 17:40
81760075	bigongi	RUN	cta	ui-tier1	wn-201-08-0	*046_5.lsf	Jun 22 17:40
81760076	bigongi	RUN	cta	ui-tier1	wn-201-03-0	*6_5.5.lsf	Jun 22 17:40
81760077	bigongi	RUN	cta	ui-tier1	wn-206-04-2	*046_6.lsf	Jun 22 17:40
81760078	bigongi	RUN	cta	ui-tier1	wn-205-01-1	*6_6.5.lsf	Jun 22 17:40
81760081	bigongi	RUN	cta	ui-tier1	wn-205-01-1	*046_7.lsf	Jun 22 17:40
81760083	bigongi	RUN	cta	ui-tier1	wn-205-01-1	*6_7.5.lsf	Jun 22 17:40
81760085	bigongi	RUN	cta	ui-tier1	wn-205-03-2	*046_8.lsf	Jun 22 17:40
81760086	bigongi	RUN	cta	ui-tier1	wn-205-03-0	*6_8.5.lsf	Jun 22 17:40
81760087	bigongi	RUN	cta	ui-tier1	wn-205-01-3	*046_9.lsf	Jun 22 17:40

Count triggered events – smart way

- Many jobs → Many log files
- Count manually triggered and generated events is boring and prone to errors
 - → Use a script
- Have a look at **CountTriggeredEvents.csh** script
 - Input:
 - The directory where the log files are
 - Output:
 - An ASCII file named CountTriggeredEvents.log with 4 columns:
 - NSB_level Discriminator_Threshold Generated_Events Triggered_Events
 - One line for each run

NSB Triggered events

```
[bigongia@ui-tier1 cta_meeting]$ more CountTriggeredEvents.log
```

0.046	1.00	1000	1000
0.046	1.50	1000	1000
0.046	2.00	1000	1000
0.046	2.50	1000	900
0.046	3.00	1000	149
0.046	3.50	1000	9
0.046	4.00	1000	0
0.046	4.50	1000	0
0.046	5.00	1000	0
0.046	5.50	1000	0
0.046	6.00	1000	0
0.046	6.50	1000	0
0.046	7.00	1000	0
0.046	7.50	1000	0
0.046	8.00	1000	0
0.046	8.50	1000	0
0.046	9.00	1000	0
0.046	9.50	1000	0
0.046	10.00	1000	0

NSB Trigger rate

- We simulated
 - $N_{\text{simulated}}$ events
 - For each event we simulated 120 time slices
 - The sampling frequency is 500 MHz
- Total simulated time
 - $T = N_{\text{simulated}} * 120/500\text{MHz}$
- $\text{NSB_Rate} = N_{\text{triggered}}/T = 120/500\text{MHz} * N_{\text{triggered}}/N_{\text{simulated}}$

Pure proton events

- Have a look at the script:
 - `proton_example_job.sh`
 - Four input parameters:
 - Single pixel NSB photo-electron rate [GHz] = 0
 - Discriminator threshold [pe]
 - Corsika input file
 - Working directory
- Copy it to your preferred directory
 - Try it:
 - `./proton_example_job.sh 0.0 1.0 proton_20.0_180.0_alt1740.0_run001183.corsika.gz /storage/gpfs_data/ctalocal/bigongia/cta_meeting/`

Scan Discriminator Threshold for protons

- Have a look at the script:
 - `proton_threshold_scan.sh`
- Launch many jobs, one for each discriminator threshold
- Run it :
 - `proton_threshold_scan.sh 0.046`
`proton_20.0_180.0_alt1740.0_run001183.corsika.gz`
`/storage/gpfs_data/ctalocal/bigongia/cta_meeting/`
- Wait for job completion
- Count triggered events in the usual way

Proton Trigger Rate

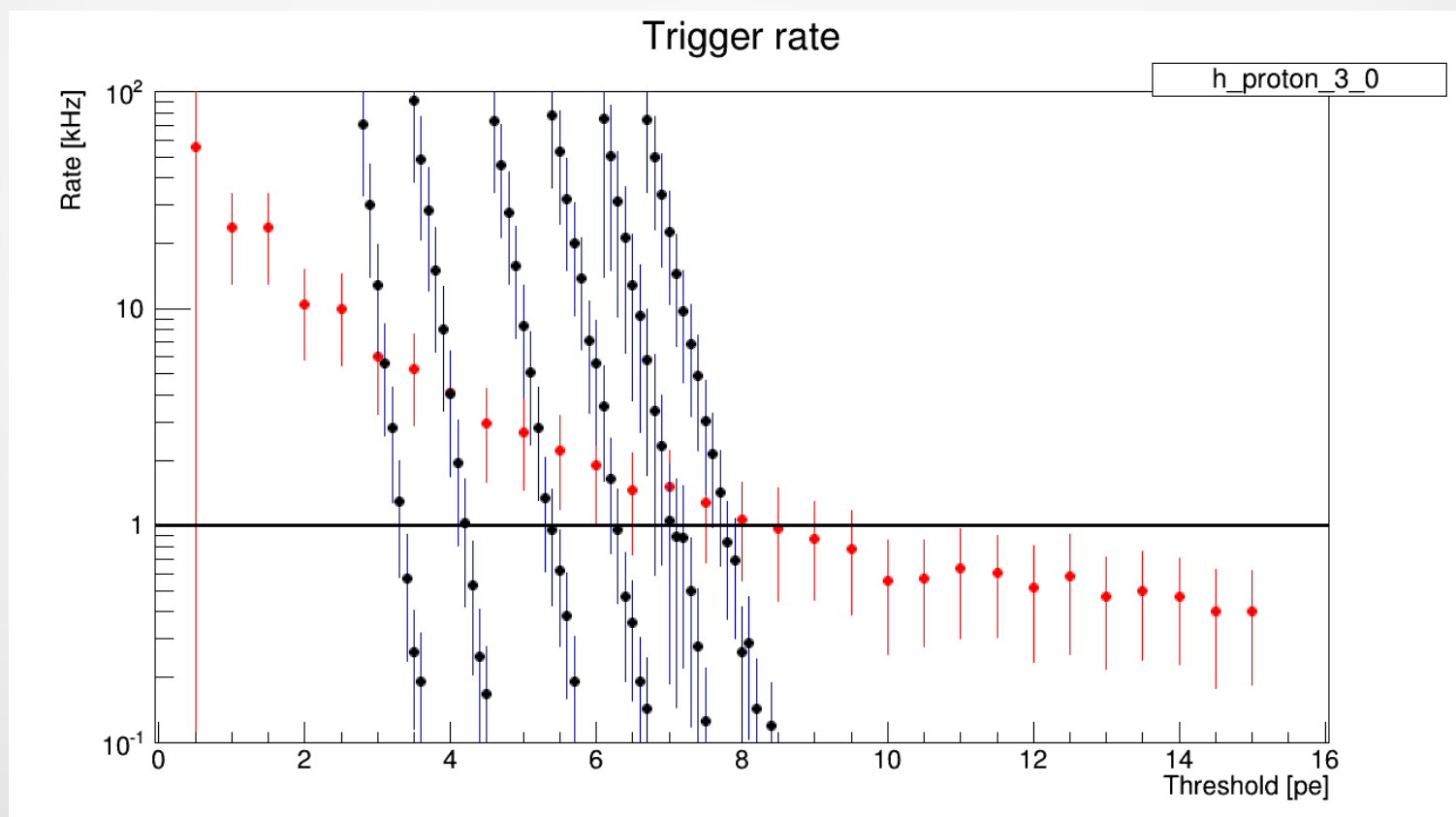
- $E_{min} = 100 \text{ GeV}$ $E_{max} = 600 \text{ TeV}$
- Spectral Slope 2.7
- $R_{max} = 1500 \text{ m}$
- $\Theta_{max} = 10 \text{ degs}$

$$A = \pi \cdot R_{max}^2 \quad \omega = 2 \cdot \pi \cdot (1 - \cos(\theta)) \quad MaxRate = A * \Omega * \int_{E_{min}}^{E_{max}} Flux$$

$$MaxRate = 2.22 \text{ MHz} \quad Rate = MaxRate \cdot \frac{N_{triggered}}{N_{simulated}}$$

Putting all together

- Many many jobs later:



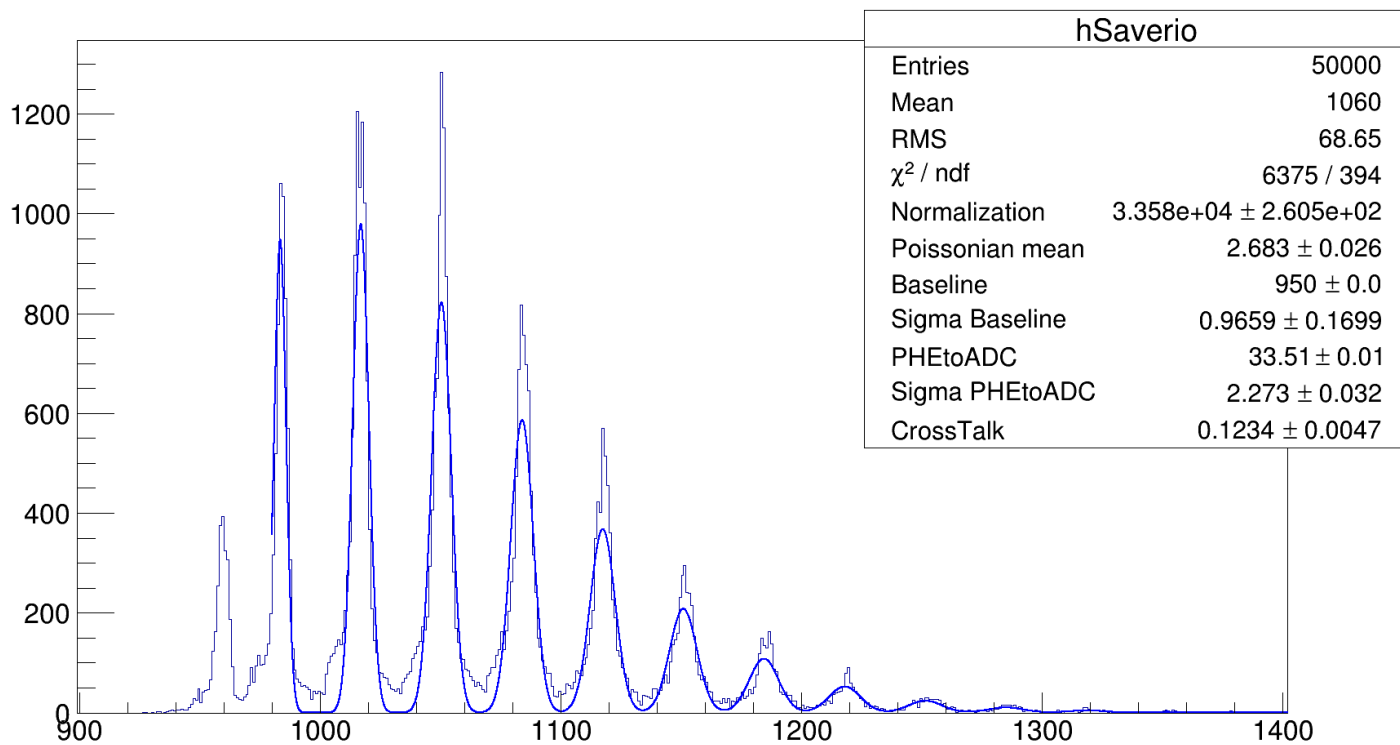
Topics

- **Discriminator Threshold Optimization (for ASTRI)**
- **Optimization of Calibration Data simulation (Ongoing)**

Simulation tune-up

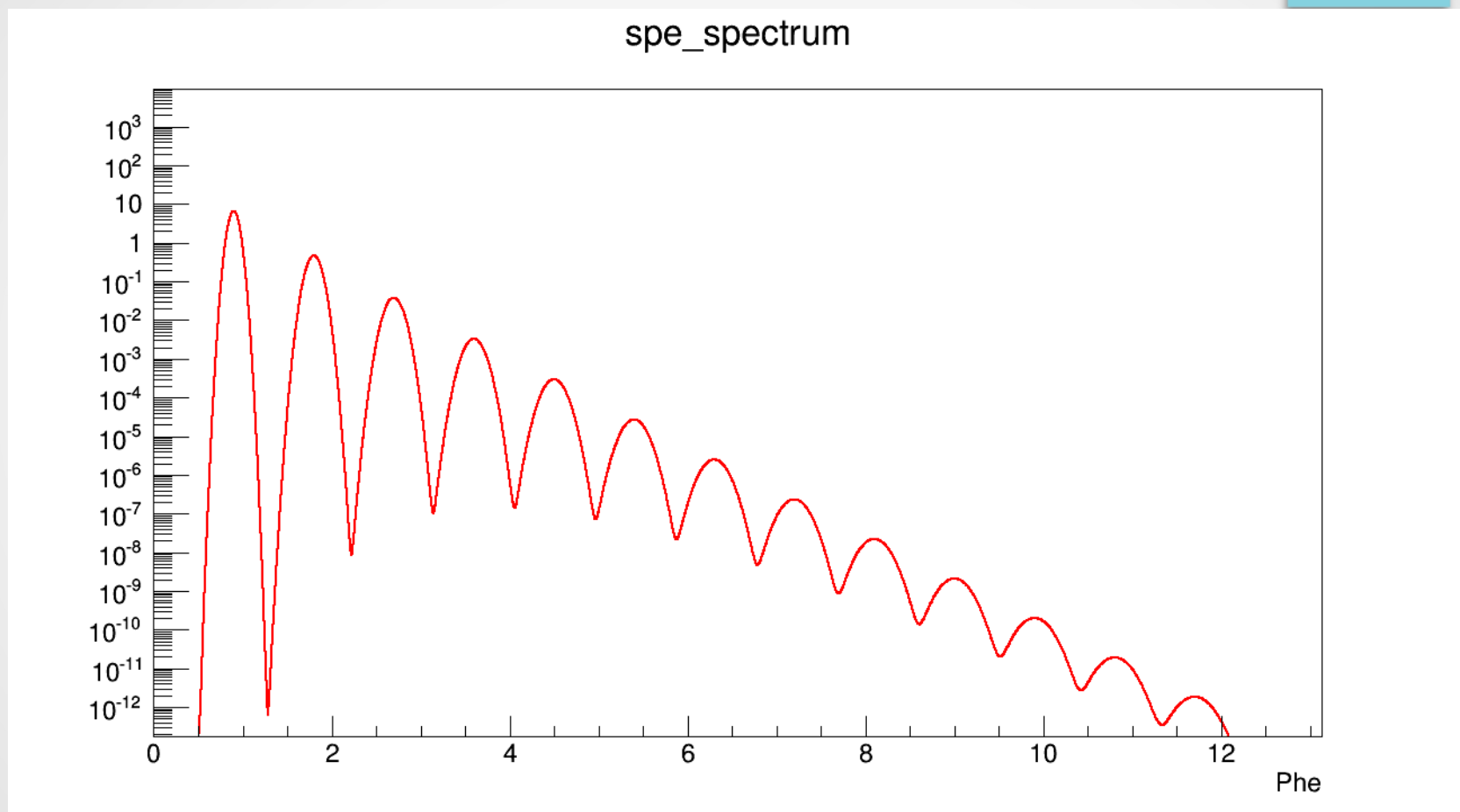
- We have already real data
 - Calibration data taken in the lab
- Try to reproduce them with our simulation code

Analytical Fit



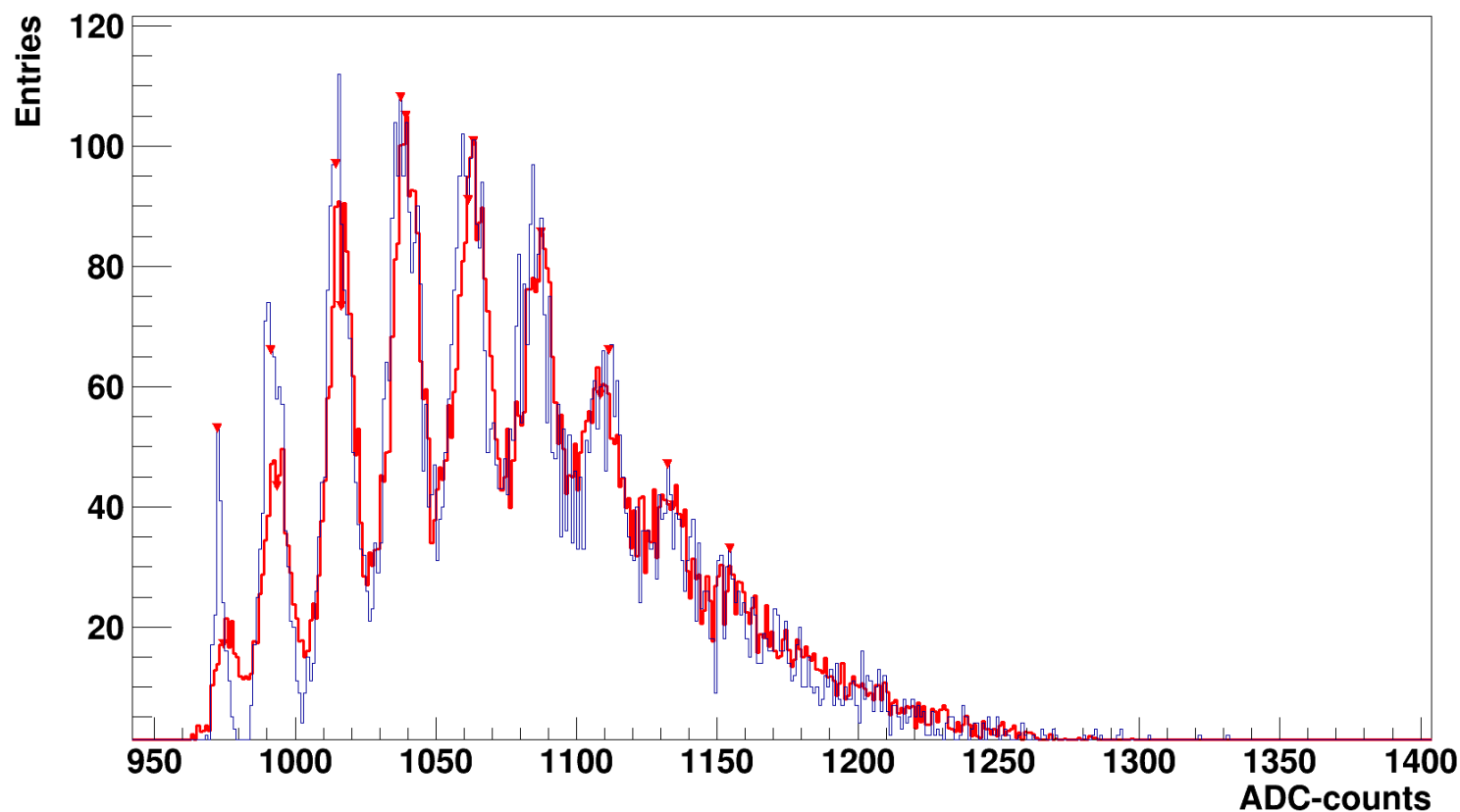
7 free parameters including the normalization

SPE spectrum



This curve depends on SigmaPheToADC and CrossTalk
The other fit parameters go into the sim_telarray configuration file

Data MC comparison



Be aware: different data sample !
Can we improve the Data-MC agreement ?

Try to improve Data-MC agreement

- Keep fixed the PheToADCcounts parameter
- Keep fixed the Baseline (Can be changed later)
- For each of the other parameters consider 6 values around the one determined with the fit
- $6 * 6 * 6 * 6 = 1296$ combinations
- Prepare 1296 configuration files (Too many parameters to change)
- Run one full simulation for each configuration file
- Compare the results with Data (Chi2 & Kolmogorov)
- Look for the best input parameter combination
- Iterate ?

Full simulation

- Simulate calibration events `sim_telarray`
- Extract calibration events `extract_hess`
- Convert the output file to ROOT `CTA.convert_hessio_to_VDST`

- Three steps → Three scripts
 - `led_simtel_job.sh`
 - `led_extractor_job.sh`
 - `led_converter_job.sh`
- To loop over configuration files
 - `led_configuration_scan.sh`

Status

- sim_telarray jobs completed
- extract_hess completed
- CTA.convert_hessio_to_VDST completed
 - 1296 root files available :)
- Loop over all the root files and compare the simulated distribution with the real data one
 - DataManineMcManineComparison.C
 - ROOT macro :(
 - Running interactively at CNAF :(

Output

ROOT file and ASCII file

In the ASCII file there is one line for each configuration file with the Chi2 and Kolmogorov values

```
>>>>> Input file LedEvents_2.550_950.0_1.00_33.51_2.250_0.100_37.5_27034_31515.simtel_6708_32520.extracted_26458_4519 Shift 0 Chi2 117888.98 Kolm 0.000000000
>>>>> Input file LedEvents_2.550_950.0_0.40_33.51_2.300_0.105_37.5_25491_28770.simtel_5175_11592.extracted_25006_9078 Shift 0 Chi2 117702.27 Kolm 0.000000000
>>>>> Input file LedEvents_2.550_950.0_1.20_33.51_2.350_0.100_37.5_27754_5725.simtel_7629_19079.extracted_27133_21363 Shift 0 Chi2 116383.42 Kolm 0.000000000
>>>>> Input file LedEvents_2.550_950.0_1.00_33.51_2.200_0.100_37.5_26941_8181.simtel_6617_11577.extracted_26369_12676 Shift 2 Chi2 115368.87 Kolm 0.000000000
>>>>> Input file LedEvents_2.550_950.0_0.40_33.51_2.350_0.125_37.5_25641_1371.simtel_5325_5972.extracted_25152_1298 Shift 1 Chi2 115115.54 Kolm 0.000000000
>>>>> Input file LedEvents_2.550_950.0_0.40_33.51_2.300_0.100_37.5_25475_31847.simtel_5160_23331.extracted_24992_19917 Shift 0 Chi2 114093.85 Kolm 0.000000000
>>>>> Input file LedEvents_2.550_950.0_1.00_33.51_2.300_0.110_37.5_27154_6619.simtel_6828_5770.extracted_26570_8432 Shift 0 Chi2 113713.66 Kolm 0.000000000
>>>>> Input file LedEvents_2.550_950.0_1.00_33.51_2.200_0.100_37.5_26941_8181.simtel_6617_11577.extracted_26369_12676 Shift 1 Chi2 110791.88 Kolm 0.000000000
>>>>> Input file LedEvents_2.550_950.0_0.40_33.51_2.350_0.125_37.5_25641_1371.simtel_5325_5972.extracted_25152_1298 Shift 0 Chi2 110659.36 Kolm 0.000000000
>>>>> Input file LedEvents_2.550_950.0_1.00_33.51_2.200_0.100_37.5_26941_8181.simtel_6617_11577.extracted_26369_12676 Shift 0 Chi2 106511.64 Kolm 0.000000000
```

Macro stopped working at about 800 files

.....