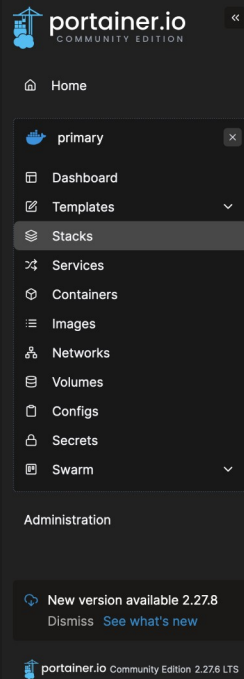


A Distributed Agata Emulator



replay_local_initializergitlab-registry.in2p3.fr/ip2igamma/docker_images:prodreplicated1 / 1Scale-2025-07-01 10:44:30

StatusFilterTaskActionsSlotNodeLast Update

runningtux501nqeallrbftt08mizljn1agata-analysis-62025-07-01 10:44:30

replay_local_processorgitlab-registry.in2p3.fr/ip2igamma/docker_images:prodreplicated11 / 11Scale-2025-07-01 10:44:40

StatusFilterTaskActionsSlotNodeLast Update

running1w0mlb4etmynn2zv6stwyvuv6agata-analysis-62025-07-01 10:44:40

running2apgy1qxulpkdoktfywybyfgo9agata-analysis-82025-07-01 10:44:40

running8oncw3mya4jvblxolgtkezy2b2agata-analysis-82025-07-01 10:44:40

runningafmcv3i7ihz35flqphm97k68210agata-analysis-82025-07-01 10:44:40

runningcn8f4dsqpm6cna2ktmg9lgkrf5agata-analysis-72025-07-01 10:44:40

runningj7jzdpd2fz14ui2amd7acefq1agata-analysis-62025-07-01 10:44:40

runningtkokogplj5gg1gw1qm2p4535z4agata-analysis-72025-07-01 10:44:40

runningw4rx96do7xnu1fe3hgnxd1ccp11agata-analysis-72025-07-01 10:44:40

runningxdxmvj081e3yw6w73nj6bc3wc8agata-analysis-62025-07-01 10:44:40

runningxqjjwoveqwoi4f1ajvlt71vn3agata-analysis-82025-07-01 10:44:40



Status at LNL
Features in preprod

G. Baulieu



AGATA Week - 10/09/2026

Features

- Agapro emulator - topology dynamically built from a configuration file
- Can dispatch processing on a cluster
 - Goal is to be scalable to handle growth in crystal number

Features

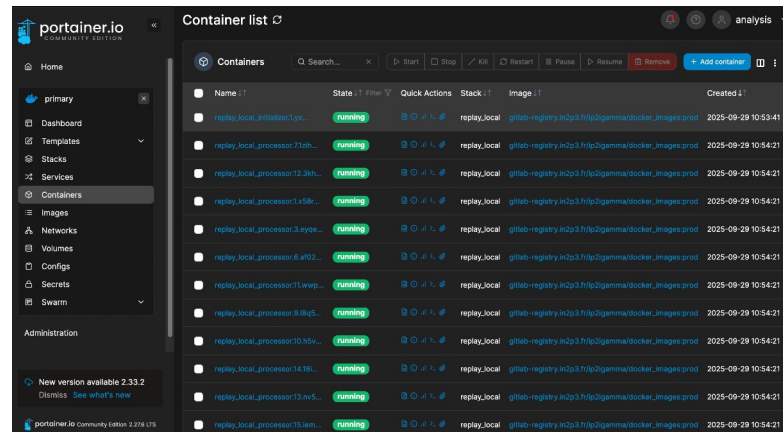
- Agapro emulator - topology dynamically built from a configuration file
- Can dispatch processing on a cluster
 - Goal is to be scalable to handle growth in crystal number
- **Local level** - we can have up to N processes for N crystals (dispatched on up to N servers)

Features

- Agapro emulator - topology dynamically built from a configuration file
- Can dispatch processing on a cluster
 - Goal is to be scalable to handle growth in crystal number
- **Local level** - we can have up to N processes for N crystals (dispatched on up to N servers)
- **Global level** - we have 4 steps :
 - Data loading
 - Event building
 - Tracking
 - Tree building
 - Can use between 1 and 4 servers to distribute processing

DAEmule at Legnaro

- Presentation of the tool on 26th January 2026
- Used for calibration since February (Julgen Pellumaj)
- Deployed on the analysis cluster (4 servers)
- Processes deployed in Docker containers
- Portainer Web interface



- Custom command line daemule_stack.py

Available documentation

- New actors : <https://gbaulieu.pages.in2p3.fr/handbook-dev/actors/AgataEventBuilder/>
- DAEmule configuration : <https://gbaulieu.pages.in2p3.fr/handbook-dev/DAEmule/DAEmule/>
- Portainer usage : <https://gbaulieu.pages.in2p3.fr/handbook-dev/DAEmule/Deployment/>
- Command line : https://gbaulieu.pages.in2p3.fr/handbook-dev/DAEmule/Deployment/#daemule_stack_management_with_daemule_stackpy
- Tutorial : <https://box.in2p3.fr/s/xK2nNc52aBRYEEi>

Links

- Agapro production Docker image : gitlab-registry.in2p3.fr/ip2igamma/docker_images:prod
- Portainer on analysis cluster : <https://172.16.1.224:9443>
- Grafana : <http://192.135.30.136:12888>

Coming next (1/2)

- Support for additional producers/filters/consumers
- Free topology description
- Currently :

```
"producer": "BasicAFP",  
"apply_preprocessing": false,  
"apply_psa": false,  
"apply_postpsa": true,  
"consumer": "BasicAFC"
```



- Can be overwritten :

```
"producer": "BasicAFP",  
"filters": ["PostPSAFilter", "ADFFrameHistogramer"],  
"consumer": "BasicAFC"
```



Coming next (2/2)

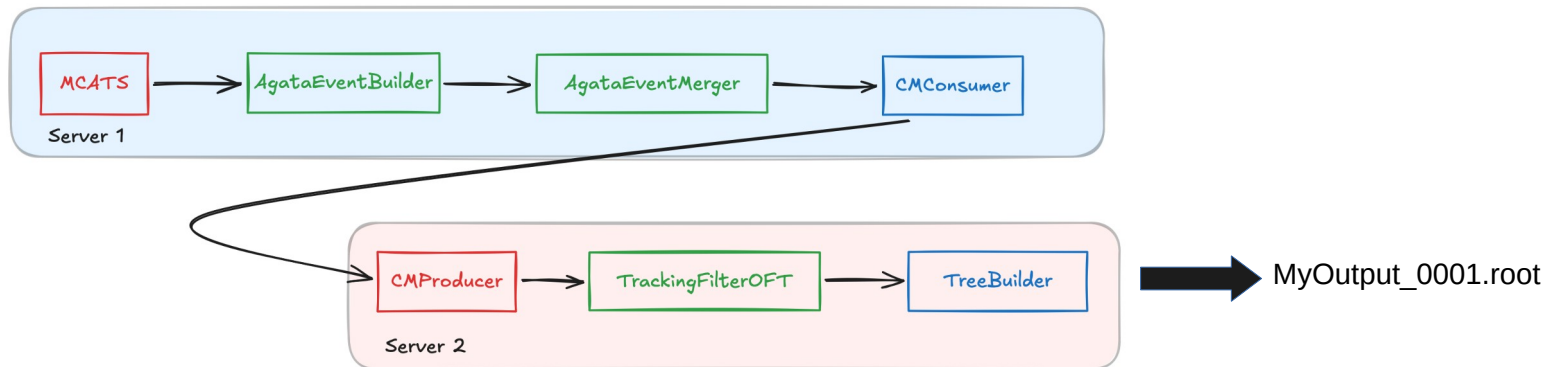
- Better logs handling

Coming next (2/2)

- Better logs handling
- MCATS optimization (Time Sorter) – tested over 2 MHz with ~30 entries

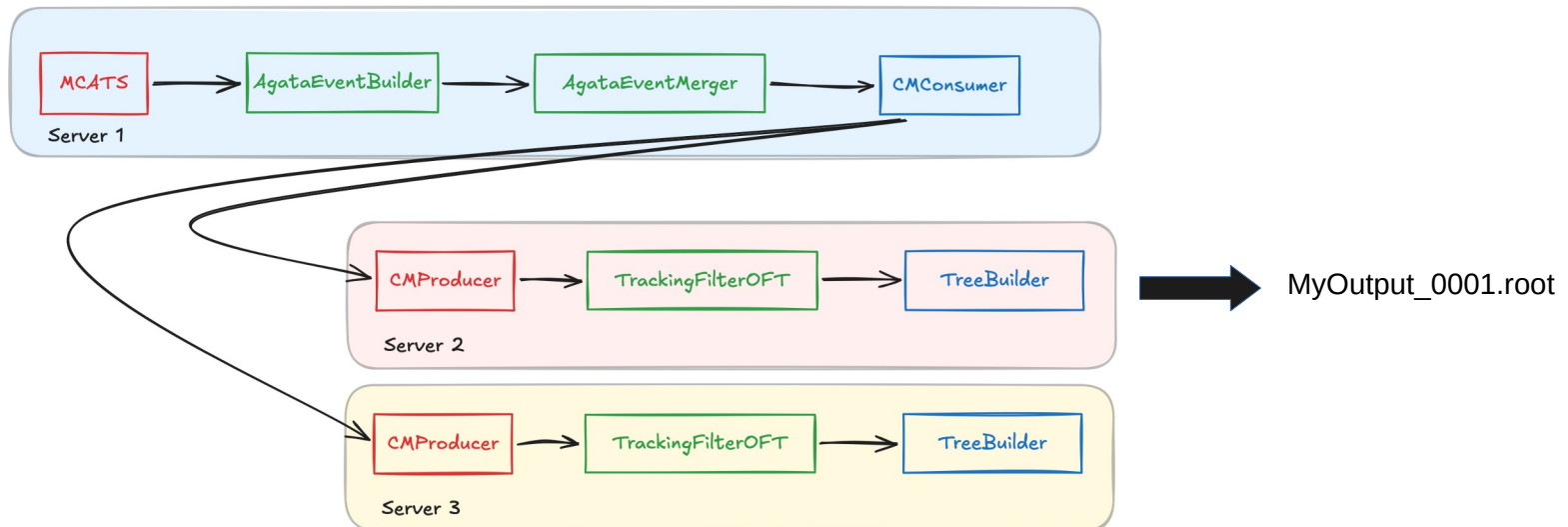
Coming next (2/2)

- Better logs handling
- MCATS optimization (Time Sorter) – tested over 2 MHz with ~30 entries
- Possibility to parallelize the Tracking+TreeBuilding step :



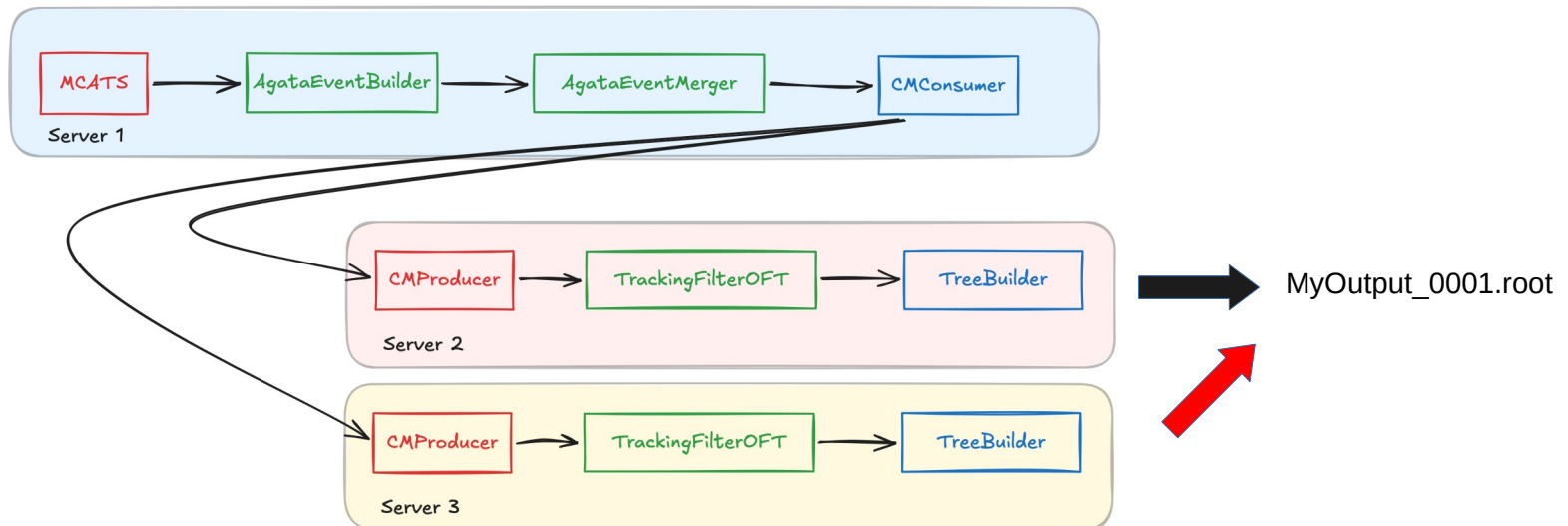
Coming next (2/2)

- Better logs handling
- MCATS optimization (Time Sorter) – tested over 2 MHz with ~30 entries
- Possibility to parallelize the Tracking+TreeBuilding step :



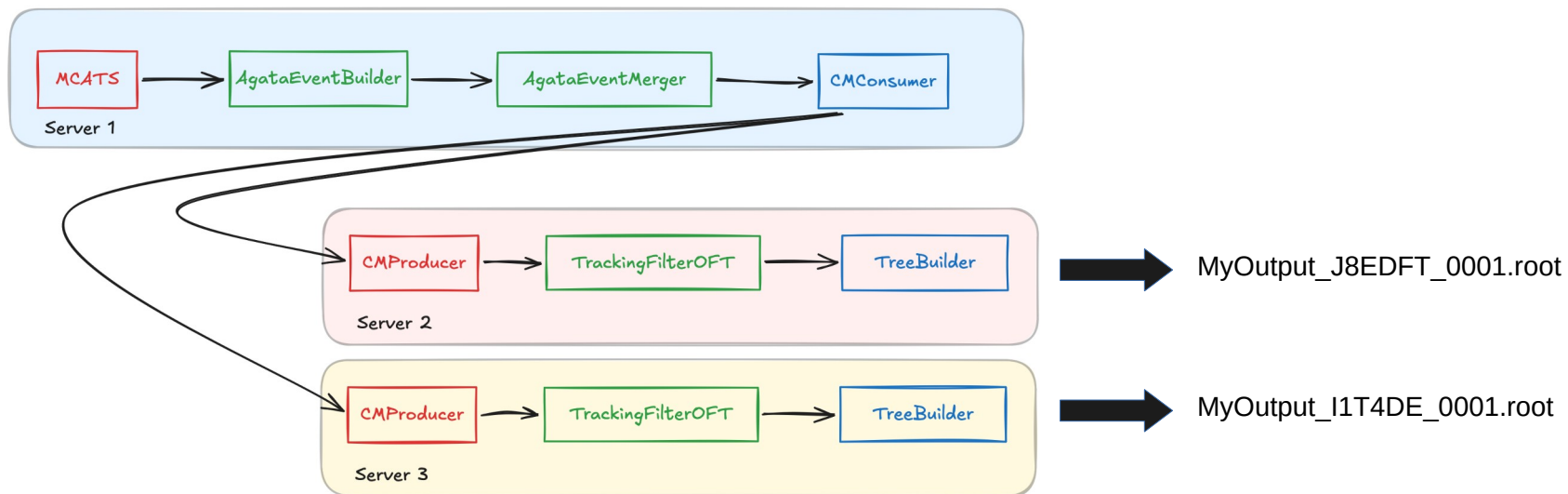
Coming next (2/2)

- Better logs handling
- MCATS optimization (Time Sorter) – tested over 2 MHz with ~30 entries
- Possibility to parallelize the Tracking+TreeBuilding step :



Coming next (2/2)

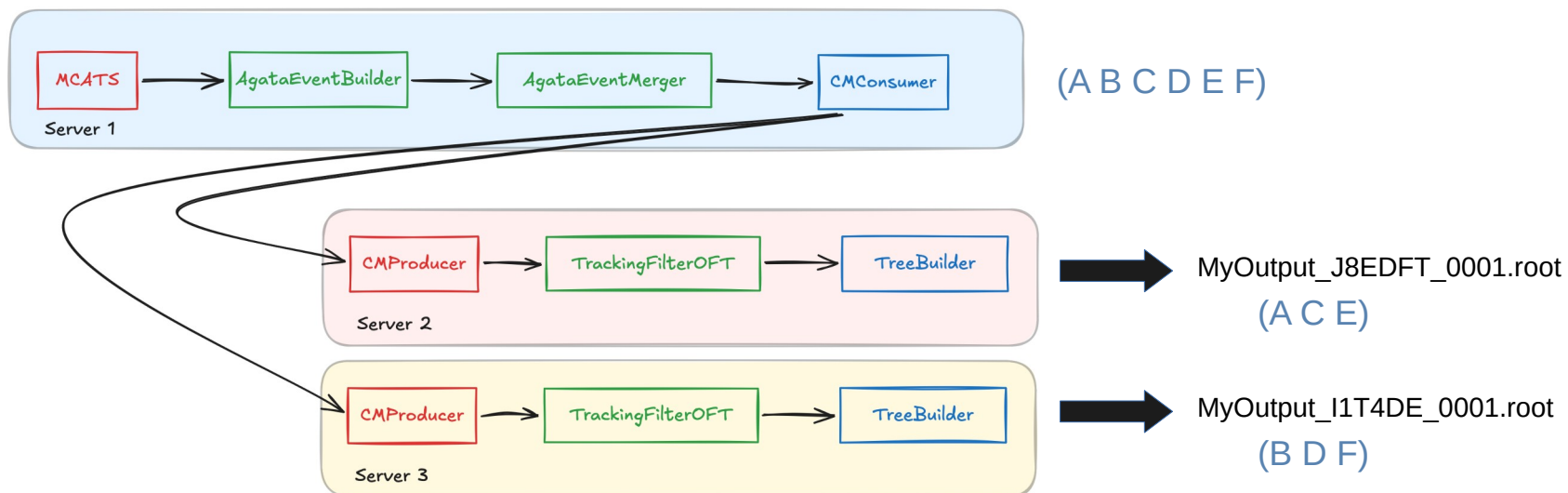
- Better logs handling
- MCATS optimization (Time Sorter) – tested over 2 MHz with ~30 entries
- Possibility to parallelize the Tracking+TreeBuilding step :



Add parameter *ParallelMode* in `TreeBuilder` → add a UID in the output file name
→ will create one set of ROOT files per `TreeBuilder`

Coming next (2/2)

- Better logs handling
- MCATS optimization (Time Sorter) – tested over 2 MHz with ~30 entries
- Possibility to parallelize the Tracking+TreeBuilding step :



Add parameter *ParallelMode* in `TreeBuilder` → add a UID in the output file name
→ will create one set of ROOT files per `TreeBuilder`

Sum up

- In production at Legnaro since February for calibration
- Running on analysis cluster (4 servers) in Docker containers
- Can be started using :
 - Portainer web interface
 - Command line `daemule_stack.py`
- Coming soon
 - Free topology description
 - Optimizations
 - Tracking + TreeBuilding parallelization