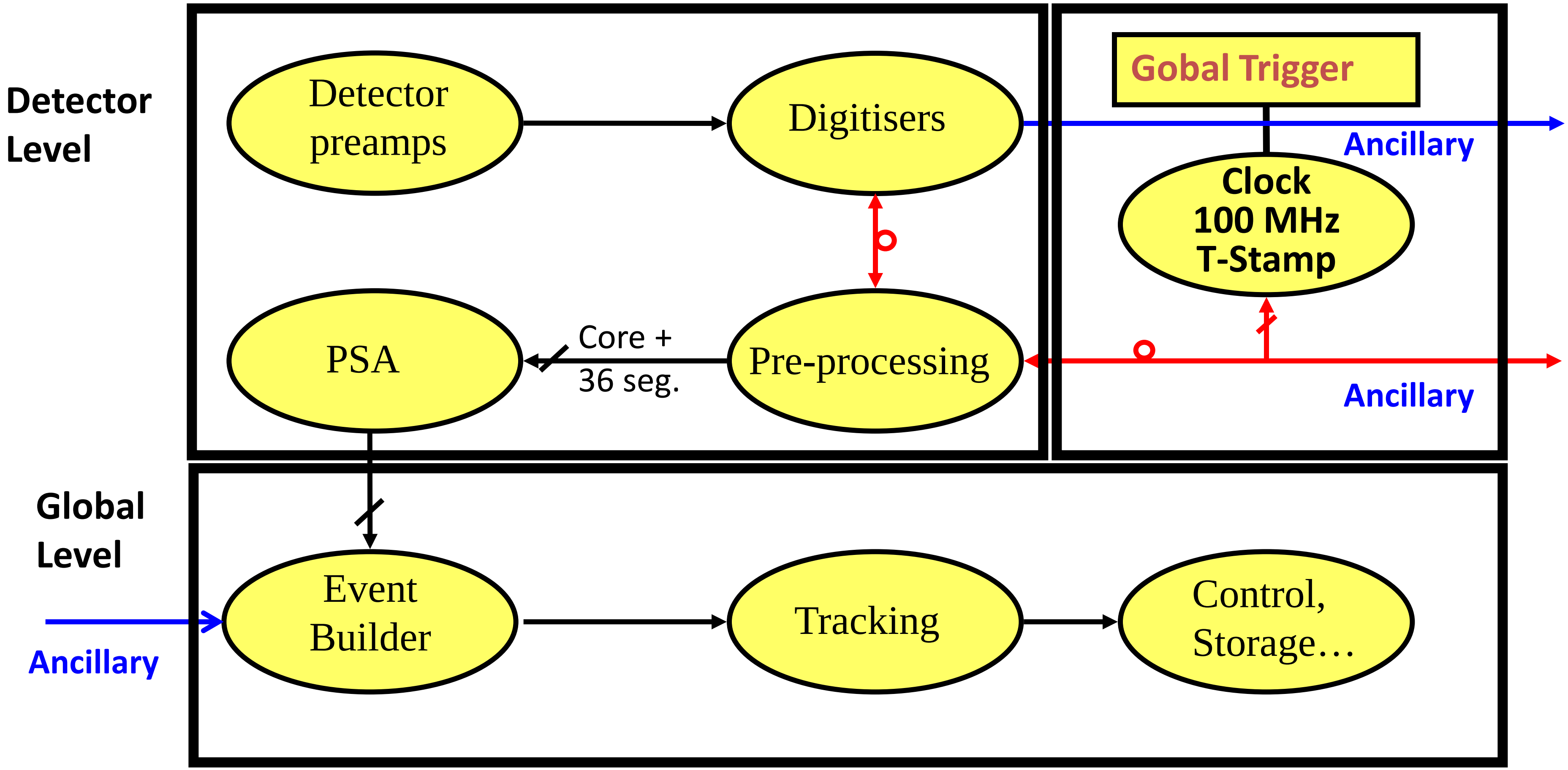
Progress in online and offline data ~~Analysis~~ Processing

Dino Bazzacco
INFN Padova

- Structure of the system
- Configuration of the analysis
- Detail and info of the various actors
- Suggestion on how to proceed with the replay of raw data

10th AGATA Week, IPNL-LYON 22-26 November 2010

Structure of EDAQ



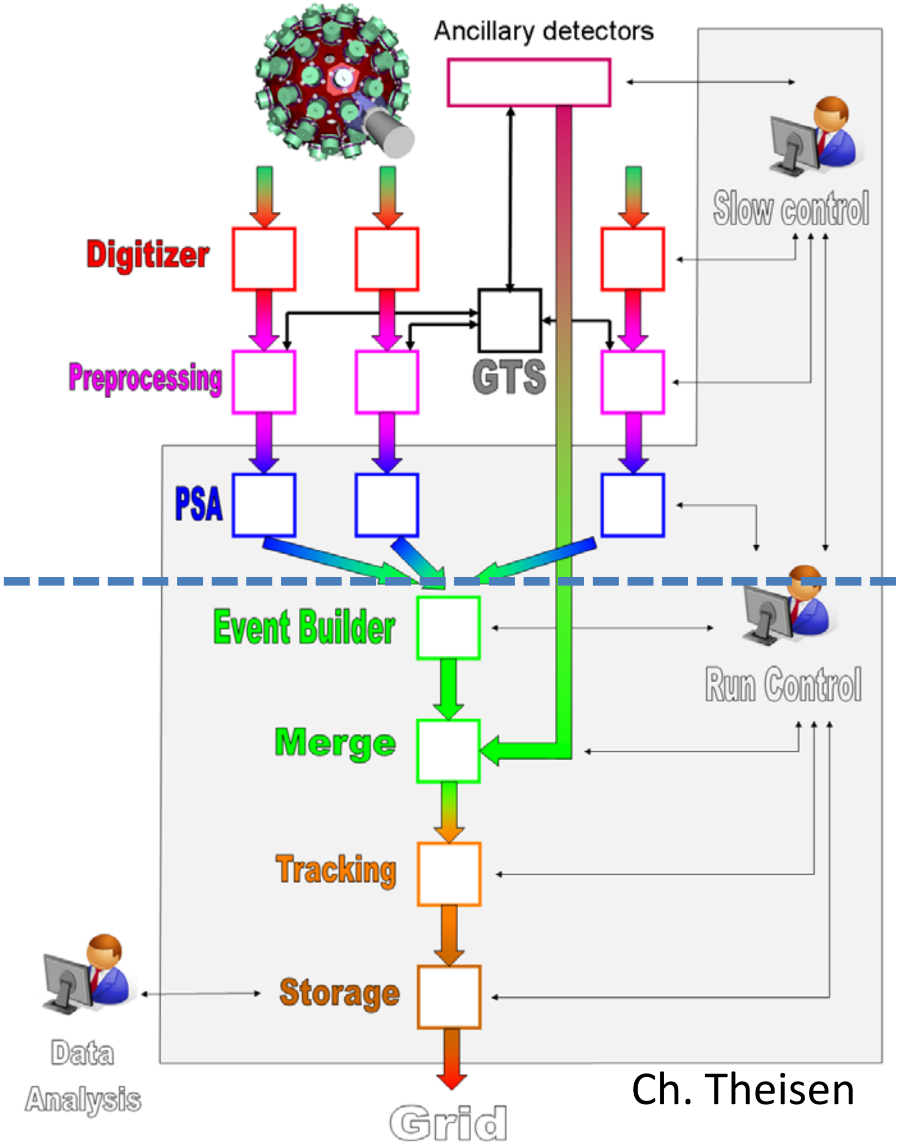
Ancillary detectors

- interface to GTS via mezzanine
- merge time-stamped data into event builder
- prompt local trigger available from digitisers

Structure of Data Processing

- Local level processing
  - AGATA
    - Readout
    - Pre-processing
    - PSA
  - Ancillaries
    - Readout
    - Pre-processing

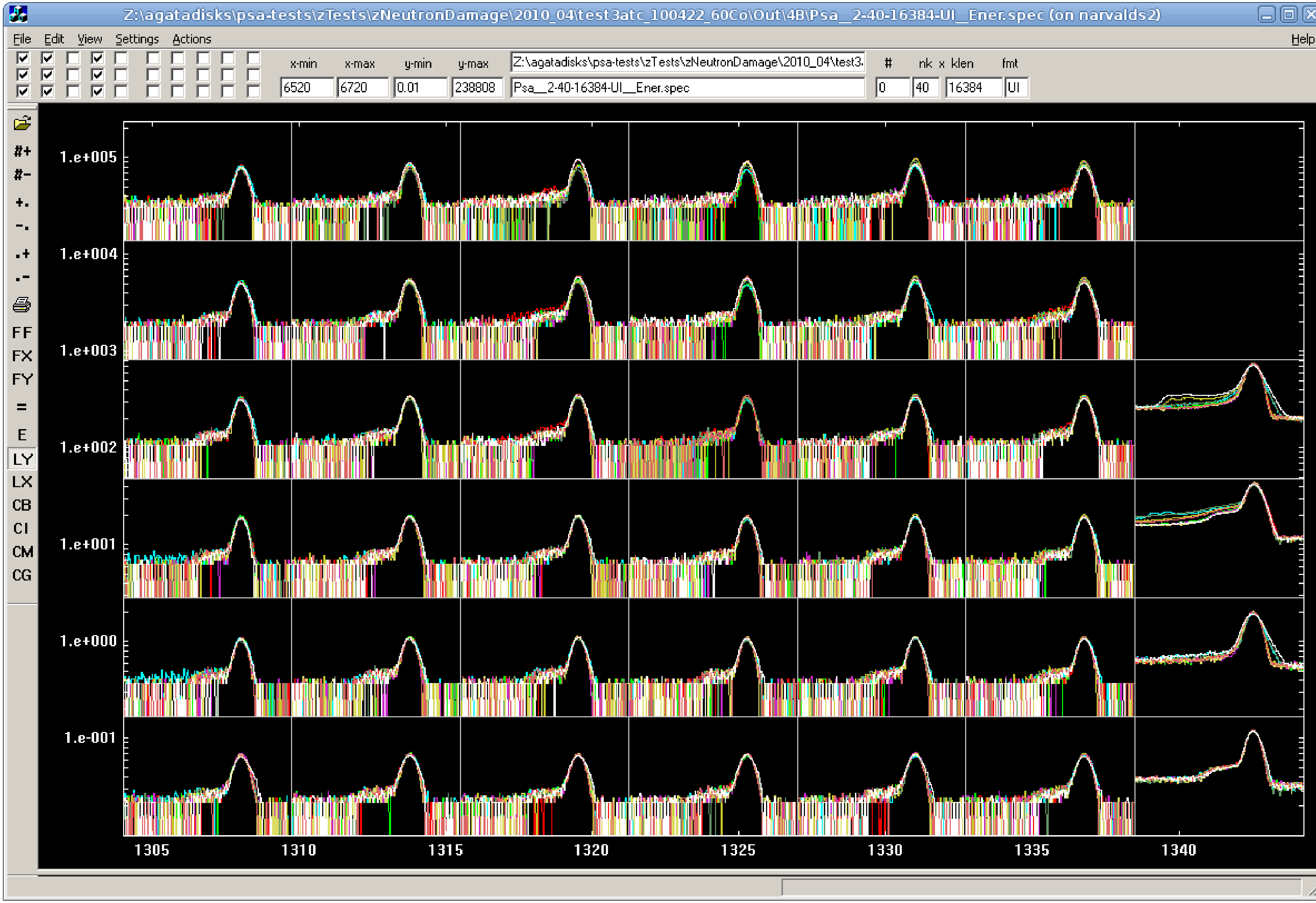
- Global Level processing
  - Event builder
  - Event merger
  - Pre-processing
  - Tracking
  - Post-Processing

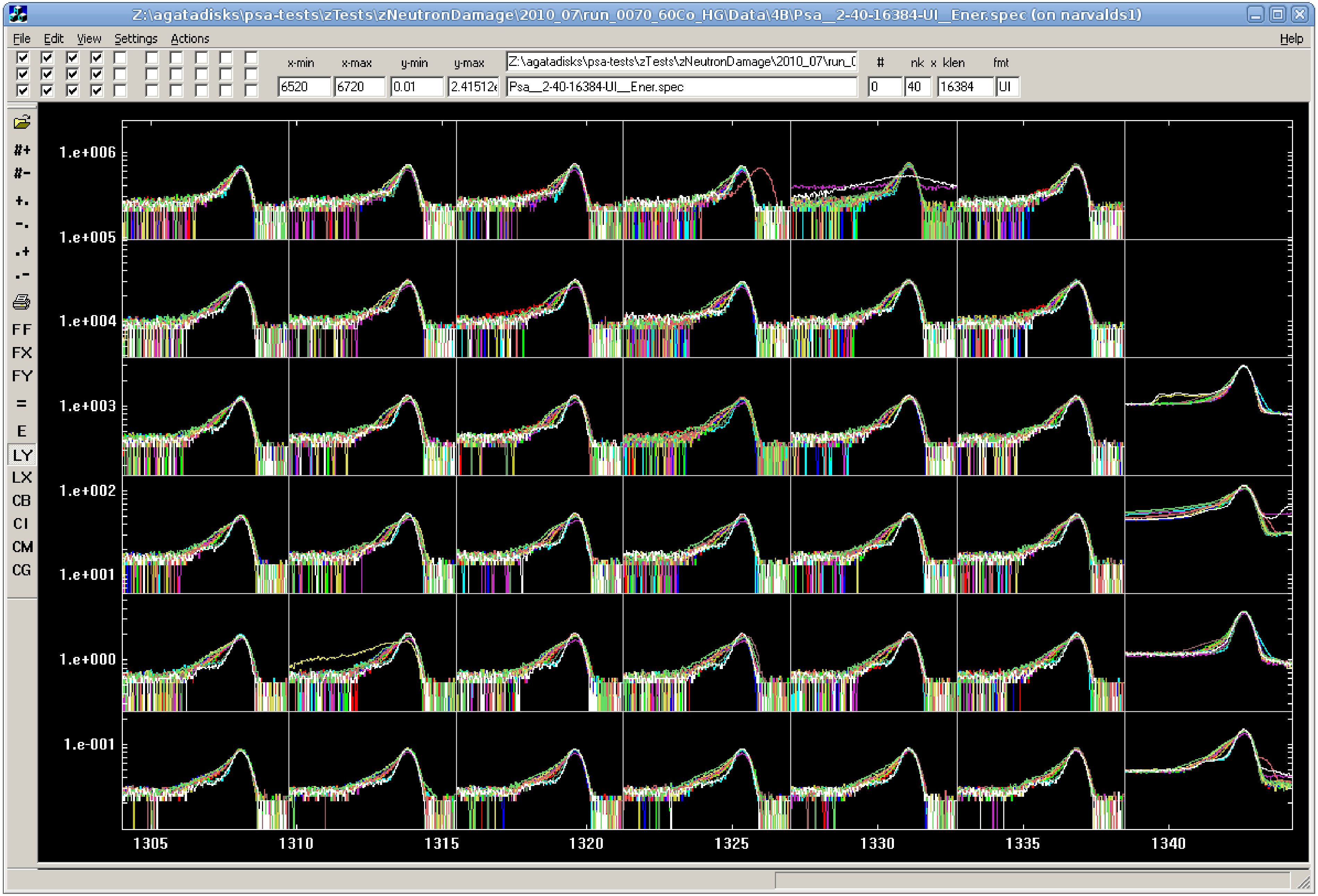


Ch. Theisen

Data processing and replay

- Online = Offline
- A series of programs organized in the style of Narval actors
- Director of the actors
  - **Online** : **Narval**
  - **Offline** : **Narval** or narval emulator(s) → WinCtest → Emulator
    - Depending on the available computing resources
    - Single computer
    - Farm with distributed processing
- The system is complex and “difficult” to manage
  - Very large number of detectors NumberOfCrystals*38 → 5 ATC = 555
  - No chance to take care of them individually
  - Rely on automatic procedures
  - Hope that the system is stable
  - Result depends on average performance


Z:\agatadisks\psa-tests\zTests\zNeutronDamage\2010_04\test3atc_100422_60Co\Out\4B\Psa__2-40-16384-UI__Ener.spec (on narvalds2)
File Edit View Settings Actions
Help
x-min x-max y-min y-max
6520 6720 0.01 238808
Z:\agatadisks\psa-tests\zTests\zNeutronDamage\2010_04\test3.
Psa__2-40-16384-UI__Ener.spec
nk x klen fmt
0 40 16384 UI
#+
#-
+.
-.
.+
.-
FF
FX
FY
=
E
LY
LX
CB
CI
CM
CG
1.e+005
1.e+004
1.e+003
1.e+002
1.e+001
1.e+000
1.e-001
1305
1310
1315
1320
1325
1330
1335
1340


Z:\agatadisks\psa-tests\zTests\zNeutronDamage\2010_07\run_0070_60Co_HG\Data\4B\Psa__2-40-16384-UI__Ener.spec (on narvalds1)
File Edit View Settings Actions Help
x-min x-max y-min y-max
6520 6720 0.01 2.41512e
Z:\agatadisks\psa-tests\zTests\zNeutronDamage\2010_07\run_0
Psa__2-40-16384-UI__Ener.spec
nk x klen fmt
0 40 16384 UI
#+ #- +. -. .+ .- FF FX FY = E LY LX CB CI CM CG
1.e+006
1.e+005
1.e+004
1.e+003
1.e+002
1.e+001
1.e+000
1.e-001
1305 1310 1315 1320 1325 1330 1335 1340

Data processing and replay

- Online = Offline
- A series of programs organized in the style of Narval actors
- Director of the actors
  - **Online** : **Narval**
  - **Offline** : **Narval** or narval emulator(s) → WinCtest → Emulator
    - Depending on the available computing resources
    - Single computer
    - Farm with distributed processing
- The system is complex and "difficult" to manage
  - Very large number of detectors NumberOfCrystals*38 → 5 ATC = 555
  - No chance to take care of them individually
  - Rely on automatic procedures
  - Hope that the system is stable
  - Result depends on average performance

- **We need GUIs, Histogram Viewers and and scripts to manage the work**

- Processing/Analysis model developed with contributions of several people
  - Xavier Grave Olivier Stezowski and their groups for the architecture and the basic tools
  - Joa Ljungvall, Daniele Mengoni, Enrico Calore, Francesco Recchia
  - The users and their feedback

Structure of analysis directories

- ANADIR

  /agatadisks/data/2010_week07/zCurrentNarvalDataDir
  /agatadisks/data/2010_week29/zCurrentReplay

  containing some standard subdirectories
  - Conf → contains configuration of actors, calibrations, ... for each detector

    1R 1G 1B 2R ... 5B Ancillary Global ← new

    with minimal differences between online and offline
  - Data → contains the data produced during the experiment

    1R 1G 1B 2R ... 5B Ancillary Global
    Online writes data here
    Offline takes data from here
  - Out → where we write the offline data

    1R 1G 1B 2R ... 5B Ancillary Global
    Offline writes data here

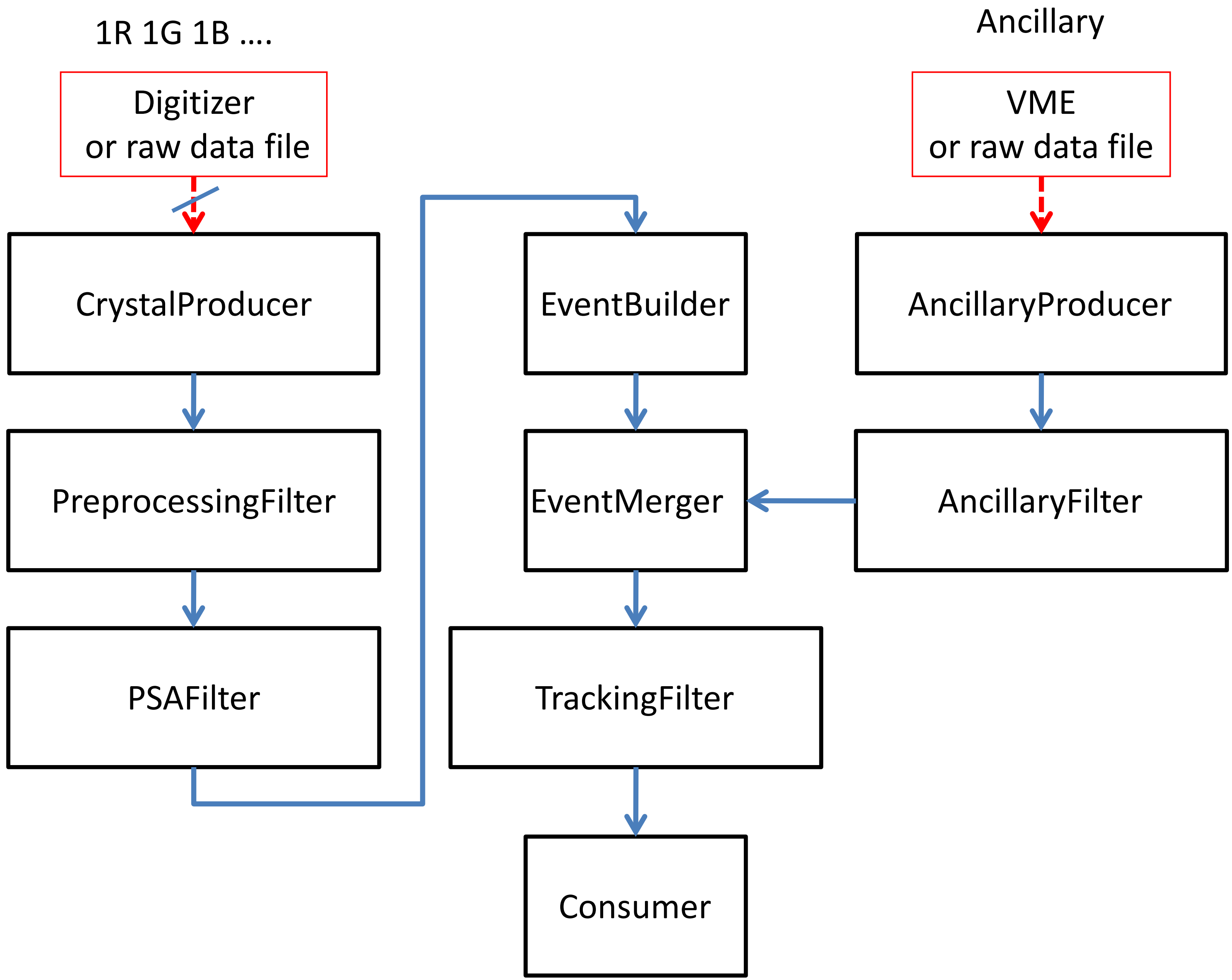
Conf, Data and Out are often symbolic links to actual directories
`$CONFDIR $READDIR $SAVEDIR` in the following examples

Details on the configuration files

- The Conf directories contain, for each used actor, one file with "keyword value" pairs to control the details of the actual analysis.
- The name of the configuration file is always **actorName.conf**
- If the actor is organized as a mother-class/daughter-class the daughter can also have its configuration files (e.g for energy calibrations, xTalk corrections, description of ATCA and VME hardware ...), but these are specific and, unfortunately, the names are not standardized
- The configuration files for the crystals are very similar; the changes to adapt them to the specific experiment/analysis have been done often using tools like sed → error prone
- Clearly, we need some grafic interface
- As a temporary solution there is now a bash script "gen_conf.sh" which should simplify the job
- For energy calibrations, xTalks, n-damage, ..., there are (or there should be) dedicated programs which should produce and copy to the target directories the proper files
- The Narval API puts constraints on how some of the parameters are passed to the system

The actors

- Producers
  - CrystalProducer / CrystalProducerATCA
  - AncillaryProducerTCP
  - BasicAFP
- Filters
  - PreprocessingFilter / PreprocessingFilterPSA
  - PSAFilter / PSAFilterGridSearch
  - TrackingFilter / TrackingFilterOFT
- Consumers
  - BasicAFC
- Builders
  - EventBuilder/EventMerger
    - ADA version for the online
    - C++ version in the WinCtest emulator


1R 1G 1B ....
Ancillary
Digitizer
or raw data file
VME
or raw data file
CrystalProducer
EventBuilder
AncillaryProducer
PreprocessingFilter
EventMerger
AncillaryFilter
PSAFilter
TrackingFilter
Consumer

CrystalProducer

- Reads the data from:
  - The PCI express driver connected to the ATCA electronics
  - Raw data files (event_mezzdata.bdat) for the offline
- Acts as a local event builder to check and merge the sub-events coming from the ATCA readout (or from the raw data file)
  - For the online has to work in nonblocking-mode to avoid blocking the ACQ
  - This feature was implemented with boost::threads
  - Damiano has now implemented the “select” and threads have been removed
  - The local event builder and the management of the mezzanine data should be completely rewritten …
- Can write the original data with the format of the ATCA electronics, and various other formats (e.g. only the energies for calibrations)
- Prepares data:crystal frames, without using adf (version using adf exists, maybe slower)

CrystalProducer.conf

```
file=$CONFDIR/$CRYSTAL/CrystalProducer.conf

ActualClass         CrystalProducerATCA
CrystalID           $CRYSTAL_ID       # position of detector in the AGATA frame
ReadDataDir         $READDIR/$CRYSTAL
SaveDataDir         $SAVEDIR/$CRYSTAL
WriteTraces         1000     # write traces the first N events as plain binary
TraceLength         100      # this is needed because the default is 160
WriteDataMask       10       # 0=none,2=event_mezzdata,8=event_energy,16=event_core
#WriteCompressed             # WriteCompressed is the default
#WriteUnCompressed
#TimeStep           10       # max delay to produce output (10 s default)
#WriteDataThreshold 5000     # CC amplitude in channels
#WriteDataSplit     10000    # split raw data after 100000 events
#NumInputFiles      100      # max number of files ( e.g. event_mezzdata_0000.bdat)
#StopErrorCode      100      # < 100 to avoid stopping all at the first input error
#TstampCorrection   0        # used in case of problems
#MaxTstampSeconds   3200     # stop analysis after N elapsed timestamp seconds
#Verbose
```

Database of crystal positions and PSA signal bases

```
#### These value correspond to AGATA_LUT4.xls as of 2010-05-12
CRYSTAL_ID=
SIGNAL_BASIS=
SIGNAL_BASEDIR=D:/NarvalTests/bases/ADL
CrystalData()
{
case "$1" in
   1R) CRYSTAL_ID=0;  SIGNAL_BASIS=$SIGNAL_BASEDIR/LibTrap_A001.dat ;;
   1G) CRYSTAL_ID=1;  SIGNAL_BASIS=$SIGNAL_BASEDIR/LibTrap_B002.dat ;;
   1B) CRYSTAL_ID=2;  SIGNAL_BASIS=$SIGNAL_BASEDIR/LibTrap_C002.dat ;;

   2R) CRYSTAL_ID=6;  SIGNAL_BASIS=$SIGNAL_BASEDIR/LibTrap_A003.dat ;;
   2G) CRYSTAL_ID=7;  SIGNAL_BASIS=$SIGNAL_BASEDIR/LibTrap_B003.dat ;;
   2B) CRYSTAL_ID=8;  SIGNAL_BASIS=$SIGNAL_BASEDIR/LibTrap_C005.dat ;;

   3R) CRYSTAL_ID=3;  SIGNAL_BASIS=$SIGNAL_BASEDIR/LibTrap_A002.dat ;;
   3G) CRYSTAL_ID=4;  SIGNAL_BASIS=$SIGNAL_BASEDIR/LibTrap_B005.dat ;;
   3B) CRYSTAL_ID=5;  SIGNAL_BASIS=$SIGNAL_BASEDIR/LibTrap_C006.dat ;;

   4R) CRYSTAL_ID=12; SIGNAL_BASIS=$SIGNAL_BASEDIR/LibTrap_A005.dat ;;
   4G) CRYSTAL_ID=13; SIGNAL_BASIS=$SIGNAL_BASEDIR/LibTrap_B001.dat ;;
   4B) CRYSTAL_ID=14; SIGNAL_BASIS=$SIGNAL_BASEDIR/LibTrap_C001.dat ;;
esac
}
```

BasicAFP & BasicAFC

- The basic producer that comes with the adf library
  reads the .adf files that have usually been written with
- The basic consumer that comes with the adf library
- These files are usually named
  AFC_XXX_nnnn.adf
  have a maximum size of 4 GiB and are identified by the numerical sequence nnnn

- Typical configuration files for BasicAFC/BasicAFP

```
file=$CONFDIR/$SAVEDIR/$CRYSTAL/BasicAFC.conf
file=$CONFDIR/$READDIR/$CRYSTAL/BasicAFP.conf

$SAVEDIR/$CRYSTAL   AFC_${CRYSTAL}_ 0
$SAVEDIR/Global     AFC_tracked_ 0
```

PreprocesssingFilter

- Performs
  - Energy calibrations and xTalk corrections
  - Analysis of traces
    - Calculation of T0 from core (from a linear fit of the first samples)
    - Time calibrations and shifts
    - Vertical normalization of traces
    - Define the net-charge segments
  - Reformats the data as data:ccrystal frames

- The calibration files are produced by external programs as part of the calibration procedures; these programs work on more compact data files written by the CrystalProducer.

PreprocessingFilter.conf

```
file=$CONFDIR/$CRYSTAL/PreprocessingFilter.conf

ActualClass        PreprocessingFilterPSA
SaveDataDir        $SAVEDIR/$CRYSTAL
EnergyGain         5                 # chan/keV of online "LOCALSPECTRA"
XtalkFile          xinv_1325-1340.cal
WriteTraces        1000
#CoreEnergyGate    20 40000
#SegmentFoldGate 1 3 10.
#TraceLengthRaw    160               # needed only for data before 02-2010
#SegmentEnergyThreshold 10           # 10 keV is the default
#HandShift         1.5               # extra shift when aligning traces
#Verbose
```

PSAFilter

- Signal decomposition
- Implemented algorithm is the Grid Search
  - As a full grid search
  - As a coarse/fine search (AGS)
- Reduces size of data by factor 20
- Provides the parameters for the correction of neutron damage (can also perform it)
- Must be expanded to improve timing
- **Takes ~95 % of total CPU time**
- Is the critical point for the processing speed of online and offline analyses

PSAFilter.conf

file=$CONFDIR/$CRYSTAL/PSAFilter.conf

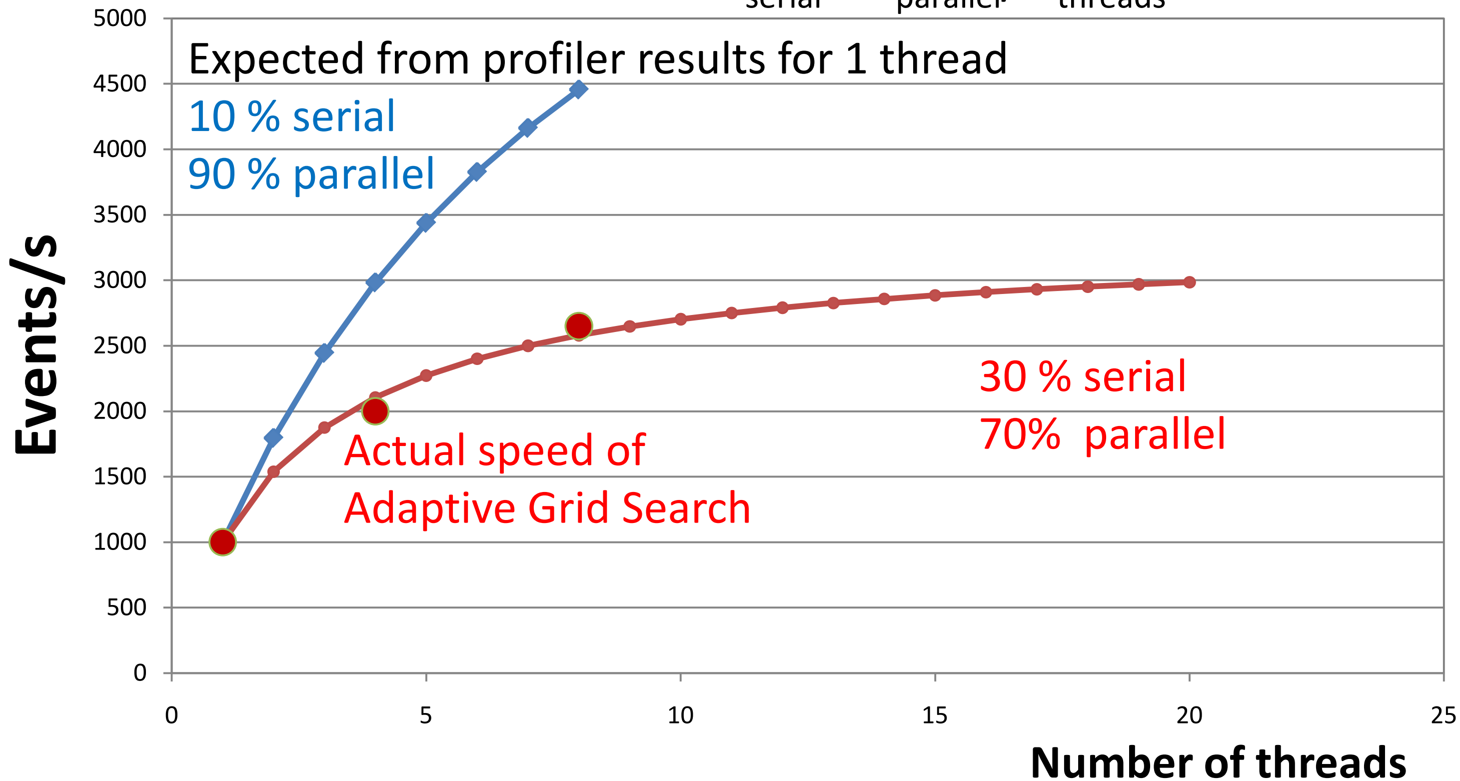
```
ActualClass       PSAFilterGridSearch
BasisFile         $SIGNAL_BASIS
SaveDataDir       $SAVEDIR/$CRYSTAL
EnergyGain        5
WriteTraces       1000
Threads           3 100    # number of threads, number of events/thread
#PlaceAtSegCenter          # the simplest PSA with the hit placed
                           # at the segment center
#SegmentFoldGate  1 3
#XtalkFile        xinv_1325-1340.cal
#TshiftFile       tshifts.cal
#WritePsaHits                      # file to calibrate neutron damage
#LambdaE                500
#LambdaH                60
#ForceSegmentsToCore
#Verbose
```

The Grid Search algorithm
implemented in Narval as **PSAFilterGridSearch**

- Signal decomposition **assumes one interaction per segment**
- The decomposition uses the transients and a differentiated version of the net charge pulse
- Proportional and differential cross-talk are included using the xTalk coefficients of the preprocessing.
- The minimum energy of the “hit” segments is a parameter in the PreprocessingFilter → 10 keV
- No limit to the number of fired segments (i.e. up to 36)
- The number of used neighbours is a compile time parameter (usually 2 as Manhattan distance)
- The algorithm cycles through the segments in order of decreasing energy;
the result of the decomposition is removed from the remaining signal
→ subtraction method at detector level
- Presently using ADL bases (Bart Bruyneel) with the neutron-damage correction model
(the n-damage correction is not yet fully implemented in this actor)
- Using 2 mm grids → ~48000 grid points in a crystal; 700-2000 points/segments
- Speed is **➔ 150 events/s/core for the Full Grid Search**
➔ 1000 events/s/core for the Adaptive Grid Search
- To speedup execution, parallelism has been implemented (using boost:threads) with blocks of ~100 events passed to N parallel threads of execution

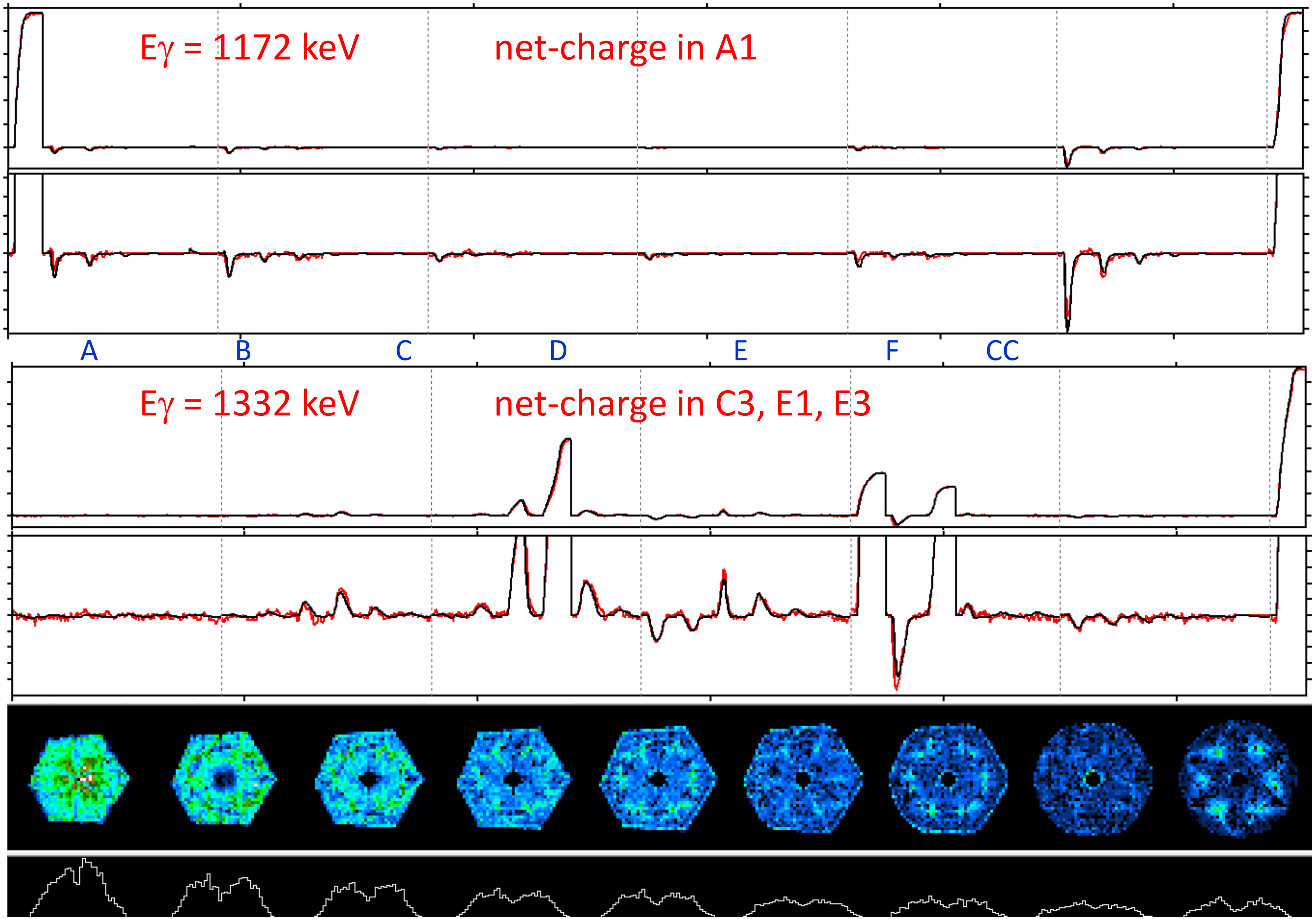
Speed does not increase linearly with the number of threads

Amdahl's law $T = T_{serial} + T_{parallel}/N_{threads}$



- The Grid search algorithm has been ported to a GPU **NVIDIA Quadro FX 1700** by **Enrico Calore** using OpenCL to reduce the modifications of the c++ code.
- Speed gain (with respect to the threaded version) is ~3.5 for the full Grid Search and 1.5 for the Adaptive (as the coarse search is still done in the CPU)

Examples of PSA signal decomposition



AncillaryProducerTCP

- Written by Enrico Calore as a tcp server
- Is not an ADF::NarvalProducer (but, for the sake of the Emulator can inherit from it)
- Reads data via tcp
- Decodes the AGAVA structure and VME data
- Formats the data as data:ranc0, not using adf
- Can write the input data as raw binary files event_vmedata.bdat

AncillaryProducerTCP

```
file=$CONFDIR/Ancillary/AncillaryProducerTCP.conf

ActualClass        AncillaryProducerTCP
ReadDataDir        $READDIR/$DIRE
SaveDataDir        $SAVEDIR/$DIRE
DetectorID         99      # 99   is default
PortTCP            9999    # 9999 is default
TcpLittleEndian            # default is big endian
#TcpBigEndian
#EndEventTag               # remove the EOE 0xffff word (Nicola's format)
#InputDataFile     event_vmedata.bdat # only for replay
#WriteDataMask     1               # input data not saved if 0 or not given
#AdfKey            data:ranc0      # ranc0 (default) or ranc1 or ranc2
#Verbose
```

AncillaryFilter

- Analysis of Ancillary data
- Changes a lot from experiment to experiment
  - PRISMA
  - Dante
  - Silicon strip detectors
  - Scintillators
- Almost impossible to set int up properly during the experiment
- In the last experiments, people gave up trying to analyse them on-line and usually opt for root-like methods
- What remains is simply raw or calibrated projections of VME data

AncillaryFilterTCP.conf

```
file=$CONFDIR/Ancillary/AncillaryFilter.conf

ActualClass       AncillaryFilterVME       #
SaveDataDir       $SAVEDIR/$DIRE           #
ReferenceTDC      3  17  550               # module index offset
#PrismaTDC        3  17  550               # module index offset
#PRISMALUTFile    $CONFDIR/$DIRE/PRISMA/lutPRISMA_WEEK49.txt
#PRISMAManager    $CONFDIR/$DIRE/PRISMA/manager.conf
#DanteChan        20 21 22                 # T X Y
#DanteTDC         3  17  550               # module index offset
#DantePos         77 4 -62 -30 -56
#DanteBeta        0.06
#DanteCalX        100 .2
#DanteCalY        -50 .5
#AllPairsMat                               # generate a big matrix of all
                                             pairs of VME signals

#Verbose
```

EventBuilder/EventMerger

- Work on data filtered by the online digital trigger by Luciano Berti
  - Can process up to 40 end-nodes
    - Agata Detectors are nodes, VME/AGAVA is a node
  - Can partition the end nodes in groups which trigger on multiplicity
  - Has a second level of prompt or delayed coincidences between partitions
- The software builders use the timestamp of the data coming from the detectors
  - Could also reconstruct using the eventnumber but this is given with only 24 bits and rolls over every 16M events → modulo electronics
- The online Narval builders had a hard life in keeping up with the variability of data flow and sometimes lost synchronization.
  - A solution was to have very large timeouts in the reconstruction algorithms but this slowed too much down the stop procedures
  - The latest improvements by Xavier Grave have not yet been tested
- The offline builders have a simpler life because data flow from files is (?) more smooth (and the timeouts in the reconstruction algorithms can be very large)

EventBuilder/EventMerger config

file=$CONFDIR/Global/EventBuilder.conf

```
ActualClass          EventBuilder
SaveDataDir          $SAVEDIR/$DIRE
BuilderType          TimeStamp  200        # EventNumber also possible but not working well
KeyIn                data:psa              # key of 1st queue default is data:psa
KeyOut               event:data:psa        # key of the output frame; default is event:data:psa
MinFold              1          # minimum Ge fold
#AddMerger                       # obsolete in the new version where the Merger
#TimestampCorrect    0  -128    # indexed by the input queue number !!
#TimestampCorrect    3  -128    # given for each channel to correct
Verbose
```

file=$CONFDIR/$DIRE/EventMerger.conf

```
ActualClass          EventMerger
SaveDataDir          $SAVEDIR/Global
BuilderType          TimeStamp 200         # EventNumber also possible but not working well
keyIn                data:ranc1            # key of 1st queue data:ranc1
keyIn                event:data:psa        # key of 2nd queue default is event:data:psa
keyOut               event:data            # key of the output frame default is event:data
MinFold    2                    # 2 means both inputs must be present
#TimestampCorrect    0  -128    # indexed by the input queue number !!
```

TrackingFilter

- This is the first point where we can check integrity of global events and analyse them
  - As AGATA only
    - Build global spectra: energy, time and $T\gamma\gamma$
    - Adjust gain of CC (and SG, not implemented) → should be moved to postPSA
    - Force sum energy of segments to energy of CC → small loss of resolution but removes extra left tail
  - As AGATA + Ancillary
    - Spectra of Ge in coincidence with ancillary
    - Gate on T_{Ge}-T_{anc} (still timestamp only) to select the data for tracking
- Input of tracking can be written in mgt-style for offline tracking with OFT or MGT
- **Tacking algorihm**
  - TrackingFilterOFT
  - Waely's, parameter-free cluster tracking algo
  - Has been a bottleneck, but now can achieve 20 kHz
  - Problem: Timing not reported well
- Output of Tracking written as adf or root tree
  - Both outputs keep the input hits (and the Ancillary data if present), adding the tracked gammas and the Ancillary filter
  - Timing is not yet reported properly

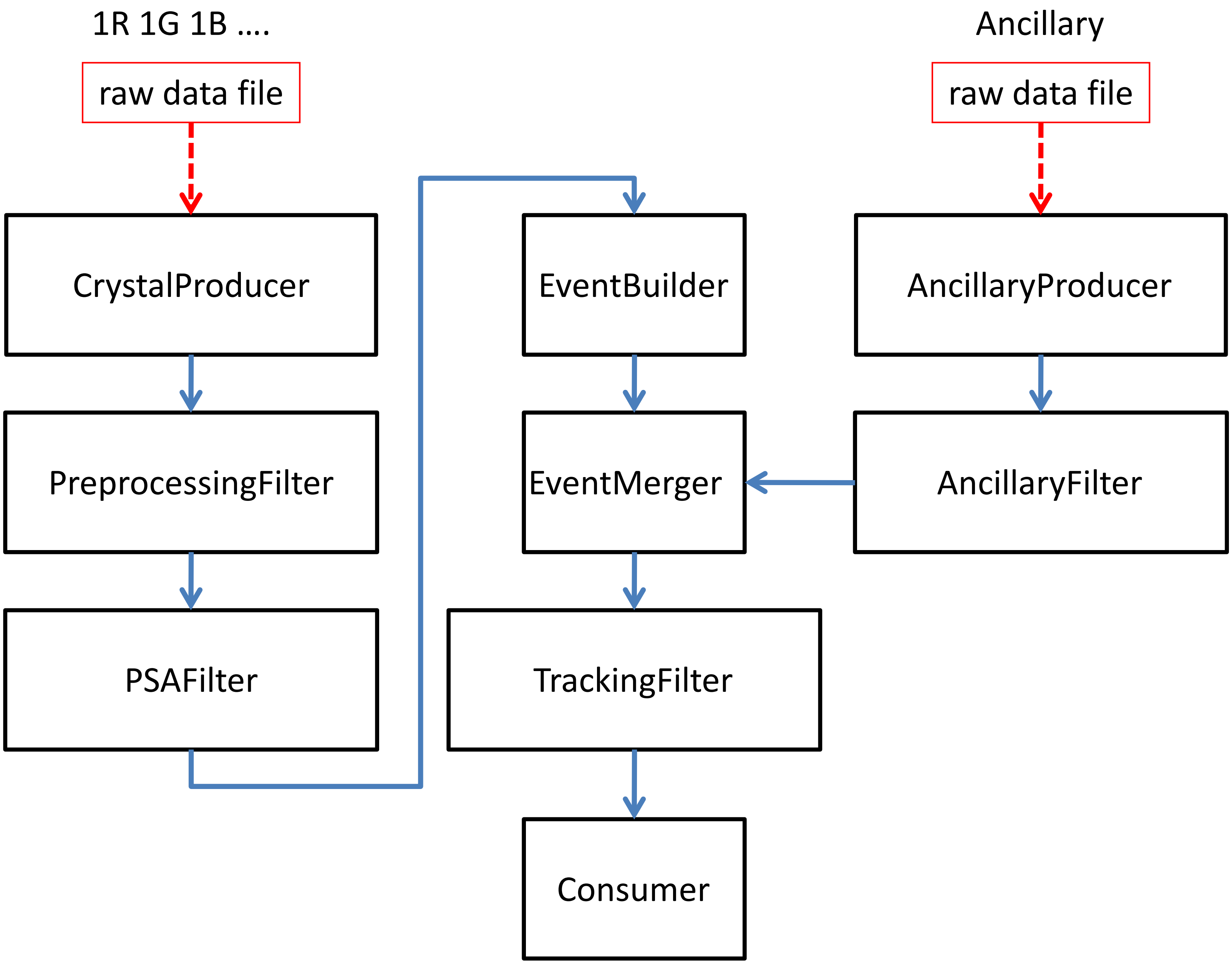
TrackingFilter.conf

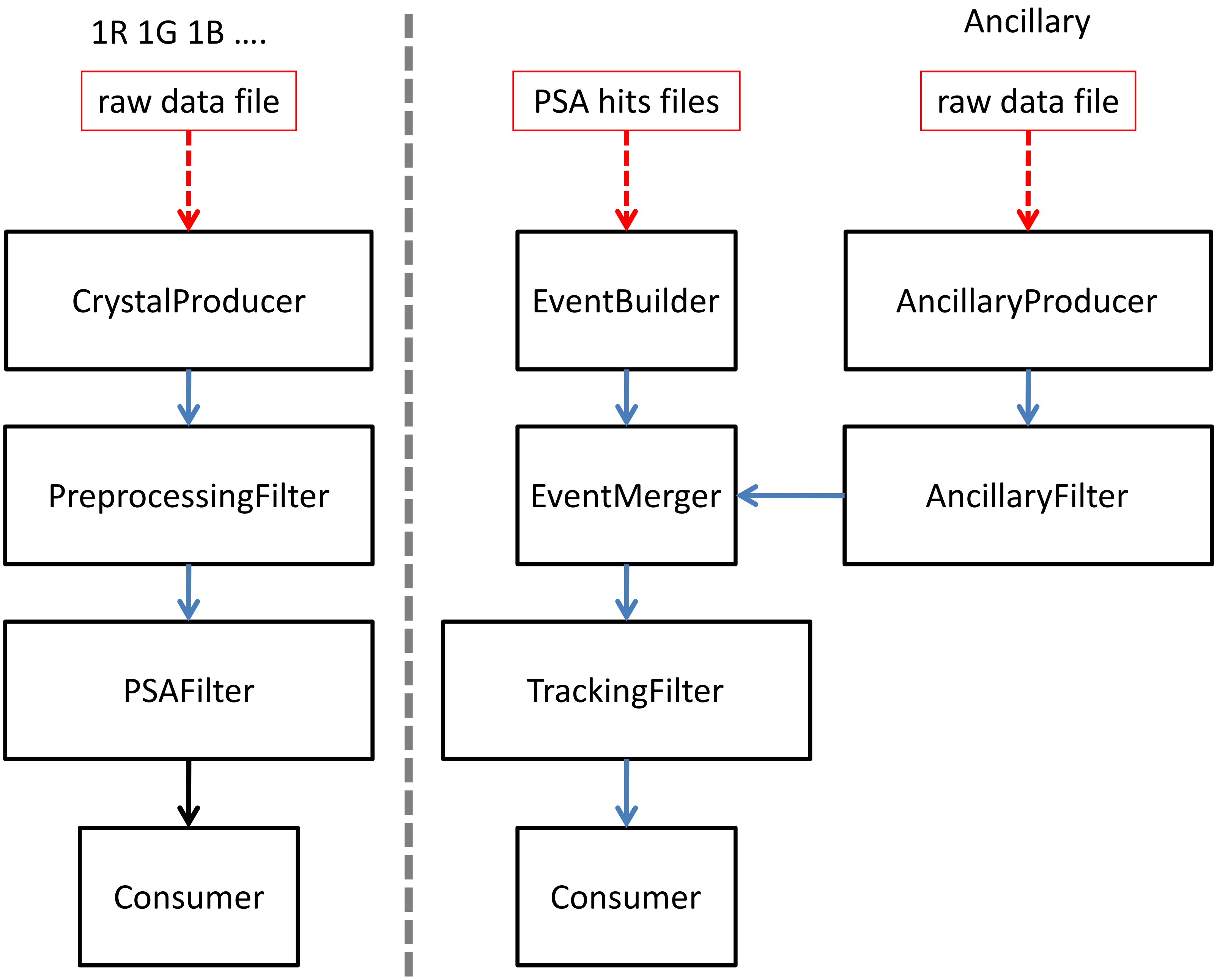
```
file=$CONFDIR/Global/TrackingFilter.conf

ActualClass              TrackingFilterOFT
SaveDataDir              $SAVEDIR/Global
EnergyGain               5
Ancillary                                       # tracking only if ancillary data must be present
Recoiling                                       # Doppler shift data expected from the ancillary frame
#RotoTranslations        CrystalPositionLookUpTable        # this is the default
#SourcePosition          0 0 54
#RecoilDirection         0 0 1
#RecoilBeta              0.053
#KeepEmpty                                      # writeout also event with zero tracked gammas
#DiscardEmpty
#OutputModel             kSafe                  # kStrict (default)  kSafe  kGrowing
RecalFile                recalfile.cal          # recalibration of CC and SumSG at tracking level
#RescaleHits             RescaleHits.dat        # same as the previous one ??
ForceSegmentsToCore                             # as a workaround of the neutron damage correction
#MinEnergyCC             1000                   # in keV, to select high energy events
#WriteInputHits                                 # write Mgt_Hits.txt
#TimeWindowGeAnc         1110 1130              # limits taken from Oft__**__TA.spec
#TimeWindowGeAncillary 1110 1130                # variant of the previous
#Verbose
#Debug                                          # to get a detailed printout of the input frames
```

Conclusion and a suggestion on how to replay the raw data

- There should be a good reason to do it
- Speed not higher than 3000 events/s on a machine like the AGATA pizza box (2 CPUs, 8 cores, 4.5 kHz, 16 GiB memory, ...)
  - a 10 TB (non compressed) data set has 10^9 events
  - 4 days of computing time if the well organized→ 1 or 2 weeks is more realistic
  - The most variable part of the analysis is at the global level (pre-tracking, correlation to ancillaries)
  - The probability of accidents along the way is proportional to the complexity.
- Two scenarios
  - **Simultaneous reprocessing of all detectors like during the experiment**
    → use Narval on a farm with several machines and saving the final tracked data
  - **Independent reprocessing of the local level part for each germanium detector**
    → use the Emulator (on your machines or on the Grid) and save the hits on adf files.
    → then, run the global level replay (including ancillary), which is much faster and can be repeated as many time as needed at a marginal cost
- In the next experiments we will probably save the PSA hits to allow for the second scenario without the need replaying the raw germanium data


1R 1G 1B ....
raw data file
Ancillary
raw data file
CrystalProducer
EventBuilder
AncillaryProducer
PreprocessingFilter
EventMerger
AncillaryFilter
PSAFilter
TrackingFilter
Consumer


1R 1G 1B ....
raw data file
CrystalProducer
PreprocessingFilter
PSAFilter
Consumer
Ancillary
PSA hits files
raw data file
EventBuilder
AncillaryProducer
EventMerger
AncillaryFilter
TrackingFilter
Consumer

Experiments replayed following this model

- ✓ 2010_week29 → complete
  - Lifetime measurement of the 6.792MeV state in ^{15}O
- ✓ 2020_week29 → complete
  - Order-to-chaos transition in warm rotating ^{174}W nuclei
- 2010_week25 → just started
  - Lifetime measurements of the neutron-rich Cr isotopes
- 2010_week24 → just started
  - Lifetime measurement in neutron-rich Ni, Cu and Zn isotopes
- ✓ 2010_week21 → complete
  - Inelastic scattering as a tool to search for highly excited states up to the region of the Giant Quadrupole Resonance
- 2010_week19
  - Neutron-rich nuclei in the vicinity of ^{208}Pb
- 2010_week07
  - Coulomb Excitation of the Presumably Super-Deformed Band in ^{42}Ca

- **Experience, status and results will be reported in the second part of the meeting by the persons taking care of the analysis**

Further details of the GS

- No limit to the number of fired segments (36)
- The number of used neighbours is a (compile time) parameter and usually is 2 (as Manhattan distance)

6				x		
5			x	x	x	
4		x	x	@	x	x
3			x	x	x	
2				x		
1						
	A	B	C	D	E	F

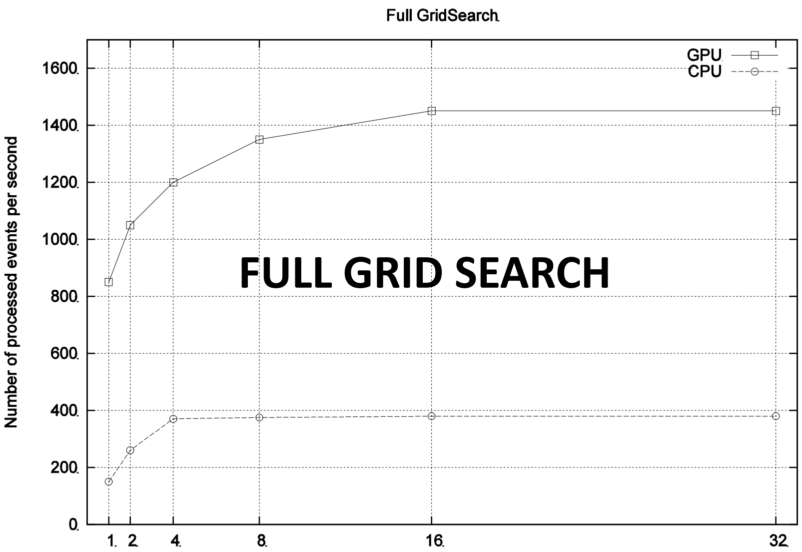
6						
5						
4						
3				x		
2			x	x	x	
1	x	x	x	@	x	x
	A	B	C	D	E	F

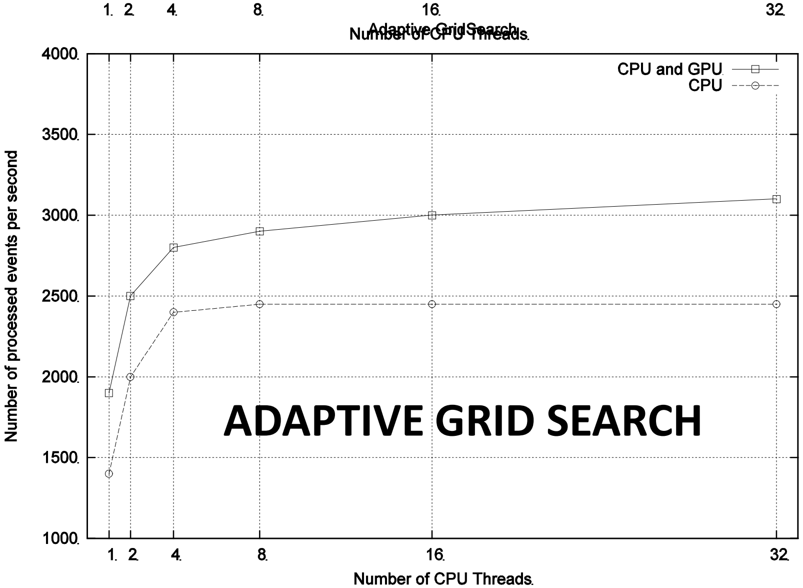
6				y		
5			xy	y	y	
4		xy	xy	@	y	y
3	x	x	@	xy	xy	
2		x	x	xy		
1			x			
	A	B	C	D	E	F

- Neighbours of the segment being analysed are not considered if they have a net energy release; no attemtp is done to remove second neighbours

Grid Search implemented on GPU

Enrico Calore





- The Grid search algorithm ported to a GPU **NVIDIA Quadro FX 1700**
- Using OpenCL and taking care of minimizing the changes to the c++ code
- To have a significant gain one needs to adapt to the architecture of the GPU
- The limiting factor is the memory bandwidth and the relatively small size of the catch

Performed experiments (so far)

- Coulomb Excitation of the Presumably Super-Deformed Band in ^{42}Ca (A.Maj, F.Azaiez, P.Napiórkowski)
- Neutron-rich nuclei in the vicinity of ^{208}Pb (Zs.Podolyák)
- Inelastic scattering as a tool to search for highly excited states up to the region of the Giant Quadrupole Resonance (R.Nicolini, D.Mengoni)
- Lifetime measurement in neutron-rich Ni, Cu and Zn isotopes (E.Sahin, M.Doncel, A.Görgen)
- Lifetime measurements of the neutron-rich Cr isotopes (J.J.Valiente-Dobón)
- Order-to-chaos transition in warm rotating ^{174}W nuclei (V.Vandone)
- Lifetime measurement of the 6.792MeV state in ^{15}O (R.Menegazzo)