# The ATLAS High Level Trigger Steering

Royal Holloway
University of London

Simon George
Royal Holloway,
University of London

# Authors

N. Berger (LAPP Annecy)
T. Bold (University of California, Irvine)
T. Eifert (Universite de Geneve)
G. Fischer (Humboldt-Universitaet zu Berlin)
S. George (Royal Holloway, University of London)
J. Haller (Universitaet Hamburg)
A. Hoecker (CERN)
J. Masik (University of Manchester)
M. zur Nedden (Humboldt-Universitaet zu Berlin)
V. Perez Reale (CERN)
C. Risler (Humboldt-Universitaet zu Berlin)
C. Schiavi (INFN, Sezione di Genova)
J. Stelzer (CERN)
X. Wu (Universite de Geneve)

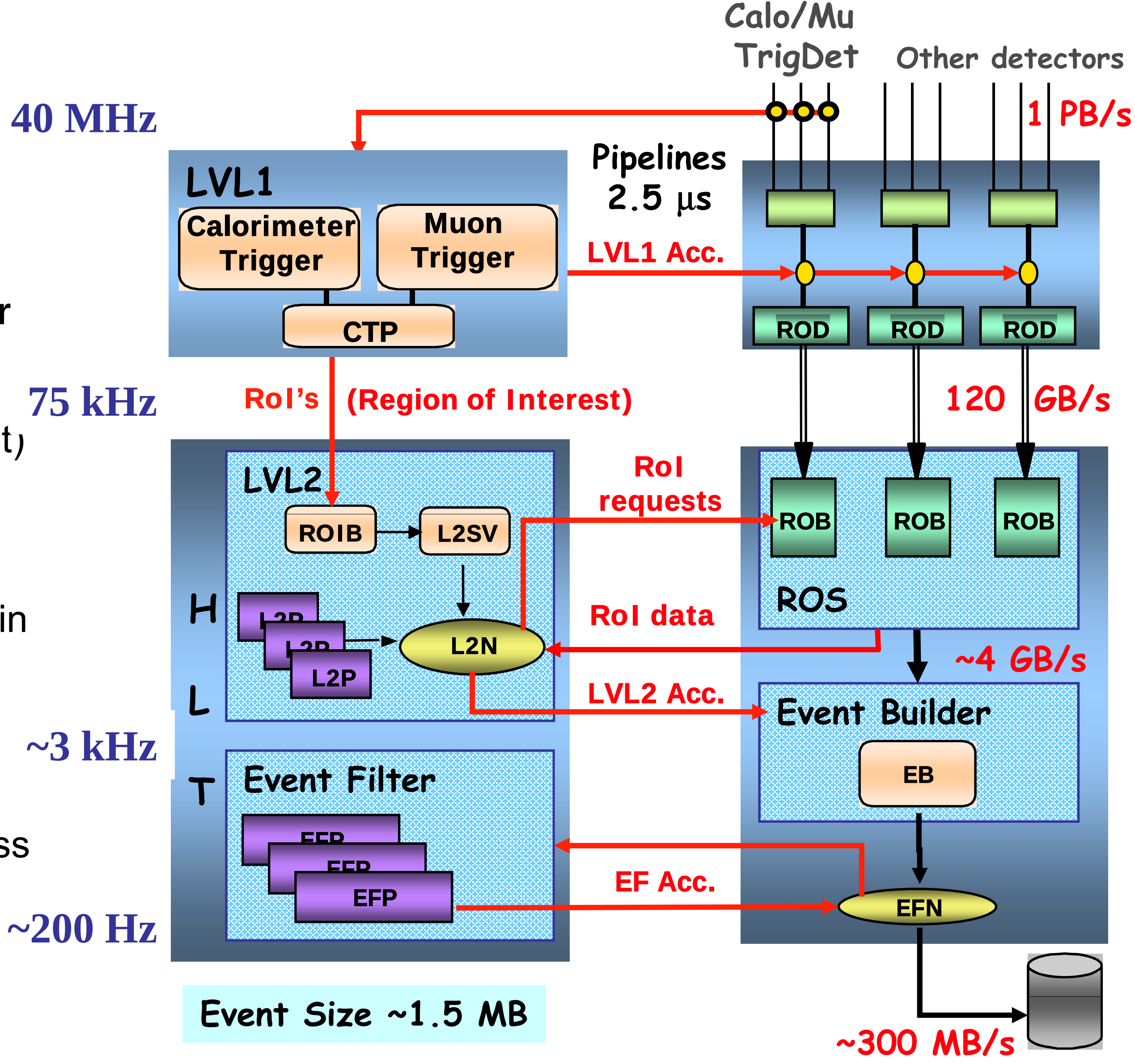
With acknowledgement to ATLAS T/DAQ.

# Outline

- Setting the scene: ATLAS trigger
- Motivation: HLT strategy
- Solution: HLT Steering
- Steering details:
  - Trigger Menu
  - Configuration
  - Algorithms
  - Logic
  - Caching
- Performance
- Monitoring
- Conclusions

# The ATLAS Trigger System

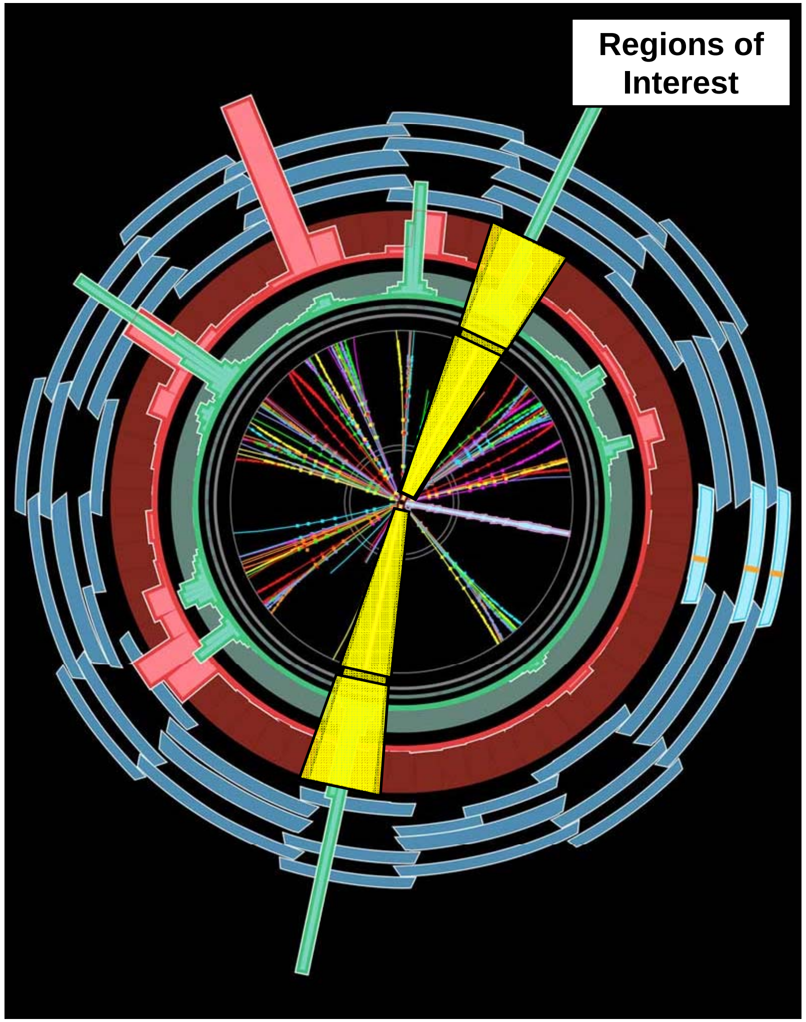
- Level 1
  - Hardware based
  - Coarse granularity calorimeter and muons only
- High Level Trigger (HLT)
  - **Level 2** and **Event Filter**
  - Software based
  - Mostly commodity hardware (PC + Ethernet)
- **Level 2** (L2)
  - Data requested from ROBs over network
  - Full detector granularity in RoIs
  - Special fast algorithms
- **Event Filter** (EF)
  - Seeded by L2
  - Potential full event access
  - Full detector granularity
  - Offline algorithms



**Simon George**

Royal Holloway University of London

# Key features of ATLAS trigger strategy



- HLT uses Regions of Interest
  - Based on L1 triggers
  - Reduce data bandwidth at L2
  - Reduce processing time
- Early rejection
  - Three level trigger
  - **Steps** within L2 and EF
  - Reduce processing time
  - Reduce decision latency

# HLT steering - in a nutshell

- RoI mechanism
  - Each trigger level or sub-**step** is seeded by the result of the previous one
- Early rejection
  - Drop event as soon as it cannot pass the trigger
  - Minimise average processing time
- Fast
  - Leave most time for event-selection algorithms
- Flexible
  - Enable/disable triggers
  - Construct complex menus from simple building blocks
- Instrumented for monitoring
- Work in both online and offline s/w environments
  - Online data taking
  - Offline development/debugging and simulation

# Sample trigger menu

| **Generic name** | e5 | e5_PT | e10 | g10 | 2e10 | e20_XE12 | XE12 |
|---|---|---|---|---|---|---|---|
| **LVL1** | EM3<br>PS | | EM8 | | 2EM8 | EM18_XE12 | XE12<br>PS |
| **LVL2** | e5 | PT | e10 | g10 | 2e10 | e20_xe12 | PT |
| **EF** | e5 | PT | e10 | g10 | 2e10 | e20_xe12 | PT |

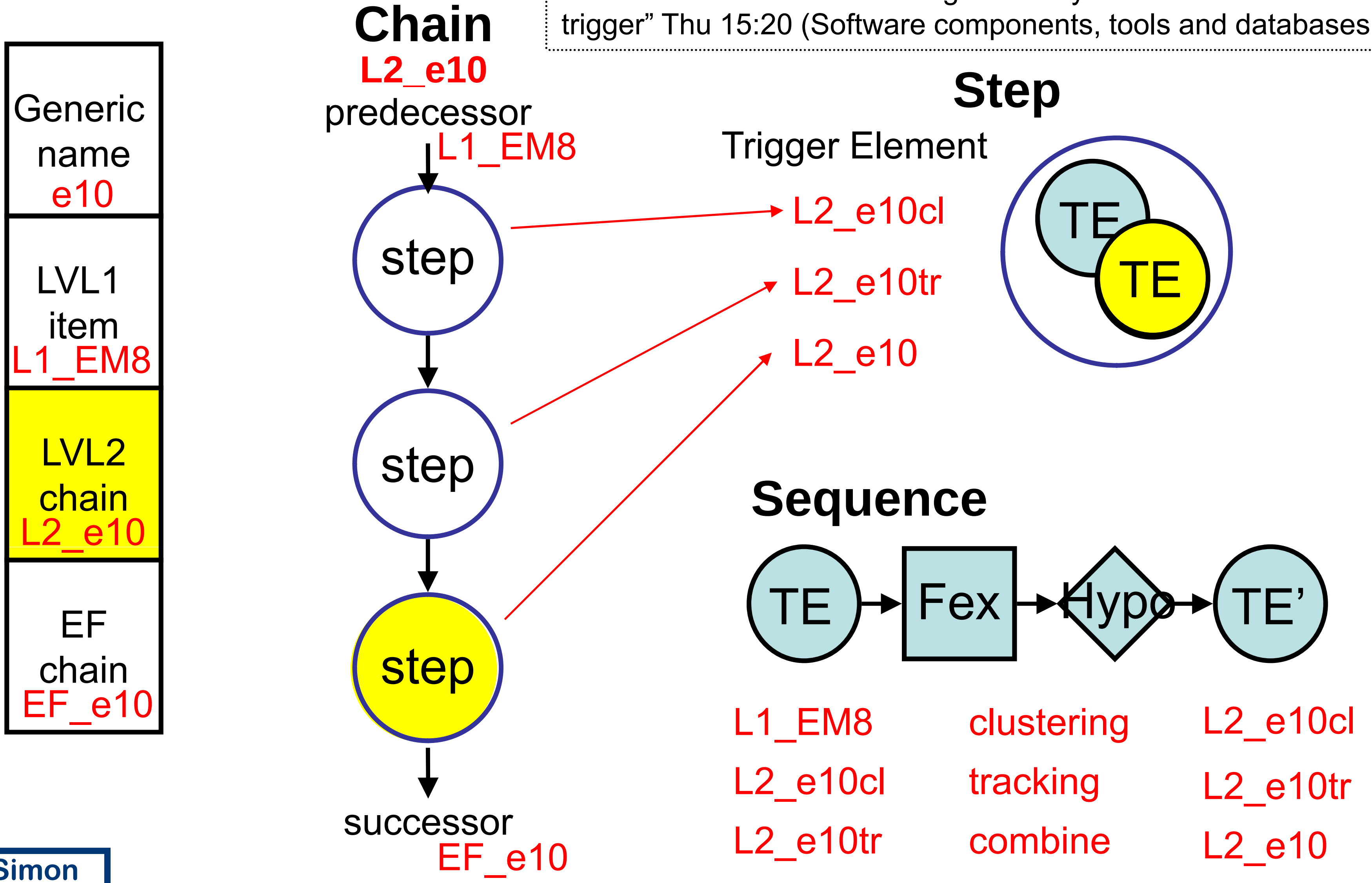
**Chain**:
represents several steps, several algorithms at each step

**Fr**

It c multiple and combined triggers, var (PS) and/or pass through (PT)

# Steering concepts

See also: J Stelzer “The configuration system of the ATLAS trigger” Thu 15:20 (Software components, tools and databases)



Generic name
e10

LVL1 item
L1_EM8

LVL2 chain
L2_e10

EF chain
EF_e10

**Chain**
**L2_e10**

predecessor
L1_EM8

step

step

step

successor
EF_e10

**Step**

Trigger Element
L2_e10cl
L2_e10tr
L2_e10

TE
TE

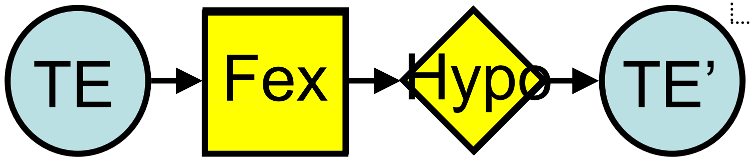
**Sequence**

TE → Fex → Hypo → TE’

| TE | Fex | TE’ |
|---|---|---|
| L1_EM8 | clustering | L2_e10cl |
| L2_e10cl | tracking | L2_e10tr |
| L2_e10tr | combine | L2_e10 |

# Algorithms

## Sequence



See also: T Fonseca Martin “Event reconstruction algorithms for the ATLAS trigger” Mon 17:55 (Online Computing)

Typical feature extraction (Fex) algorithm:

- Seeded by previous step or RoI
- Retrieves detector data
- Finds “feature” e.g. cluster, track
- Updates RoI position
- Runs once per RoI

Typical hypothesis (Hypo) algorithm:

- Follows Fex algorithm
- Compares features to hypothesis
- Marks TE’ as valid or not
- Runs once per threshold

Example Hypotheses:

Calo cluster: Cut on cluster shape parameters

Electron: cut on cluster-track matching variables

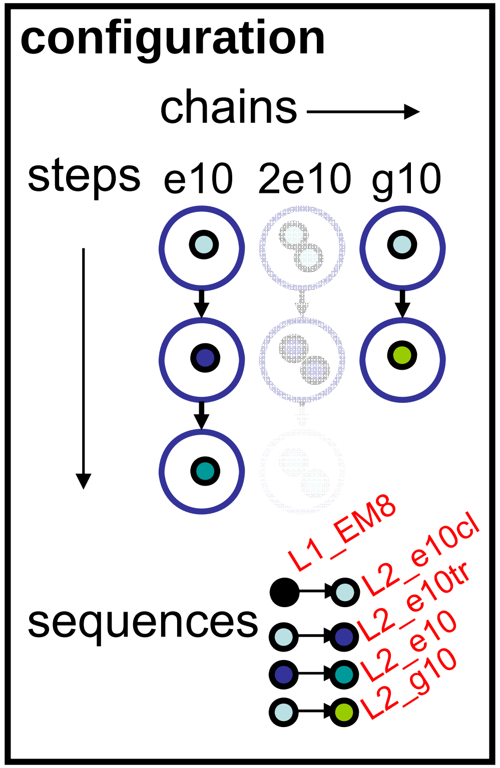
Most cases: apply $E_T$ or $p_T$ threshold

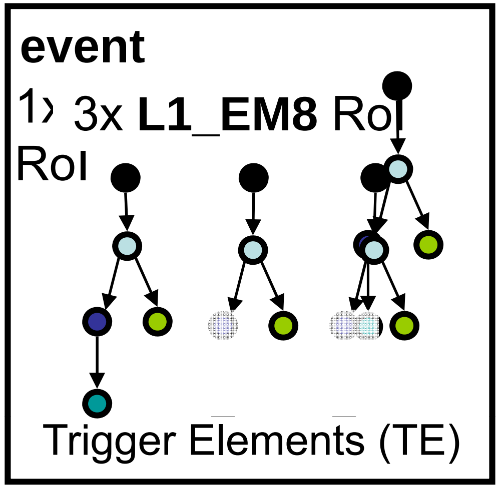
Other types of algorithm available for more complex logic.

# Steering logic

- Static configuration + dynamic event state

- First create initial TEs from L1 RoIs
  - One per threshold per RoI
  - At EF, from L2 output instead
- Activate relevant chains
- Loop over steps
  - Loop over active chains
    - Loop over TEs (event) that match step requirements (config)
      - Run sequence that links TE from prev. step in chain to required TE
      - Result (depends on algorithms): TE is active or not
    - If insufficient active TEs remain, deactivate chain
  - If no active chains remain, end loop
- Apply pre-scale, pass-through and reject/accept event

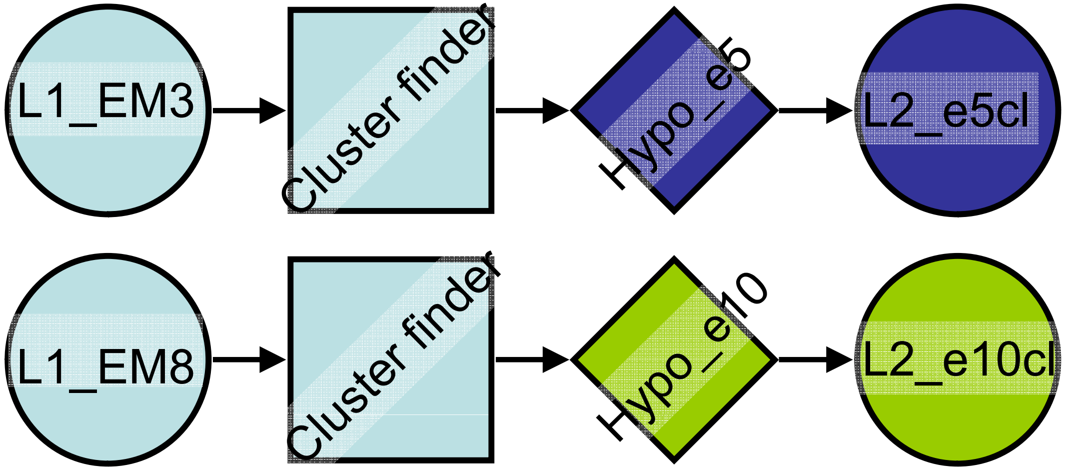




Simon George
Royal Holloway University of London

# Caching

Two sequences: same fex, different hypo:



1) Steering will run fex only once;

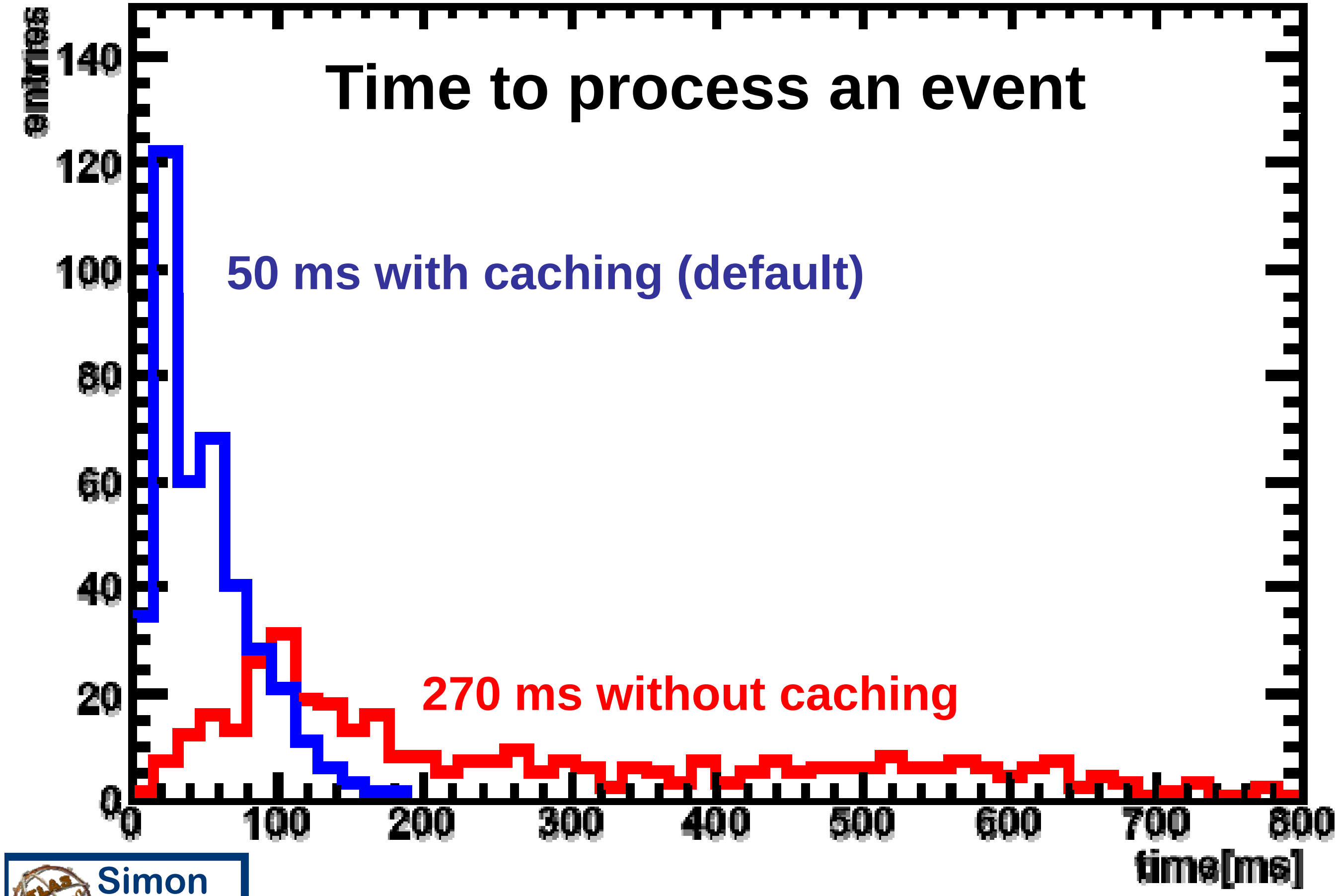
Second time, results are taken from cache

2) Same sequence in different chains is also cached

⇒ **Implicit caching** when same item appears multiple times in configuration

# Benefits of caching

- Useful gain in speed
- Simplifies configuration
- Times consistent with target



*3 GHz Xeon CPU*

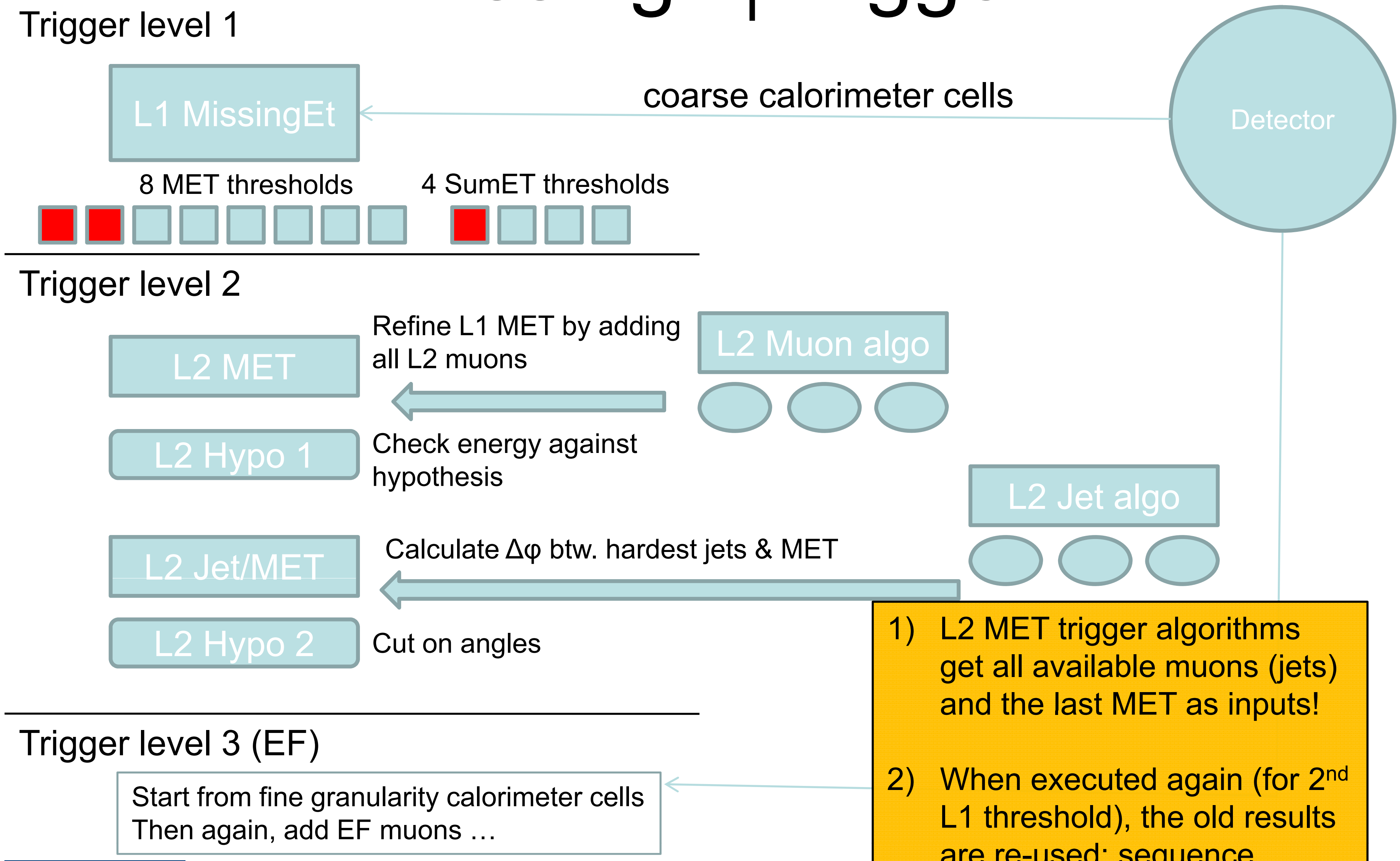
*Code is not yet optimised: expect to reduce steering time.*

*No rejection: worst case scenario.*

*Inclusive e/$\gamma$ menu for very low lumi.*

*400 top events: atypically busy; average 16 RoIs (~4 EM) per event rather than the usual ~2.*

# MissingE$_T$ trigger



Trigger level 1

L1 MissingEt

coarse calorimeter cells

Detector

8 MET thresholds

🟥 🟥 ☐ ☐ ☐ ☐ ☐ ☐

4 SumET thresholds

🟥 ☐ ☐ ☐

Trigger level 2

L2 MET

Refine L1 MET by adding all L2 muons

L2 Muon algo

L2 Hypo 1

Check energy against hypothesis

L2 Jet algo

L2 Jet/MET

Calculate Δφ btw. hardest jets & MET

L2 Hypo 2

Cut on angles

1) L2 MET trigger algorithms get all available muons (jets) and the last MET as inputs!

2) When executed again (for 2nd L1 threshold), the old results are re-used: sequence caching

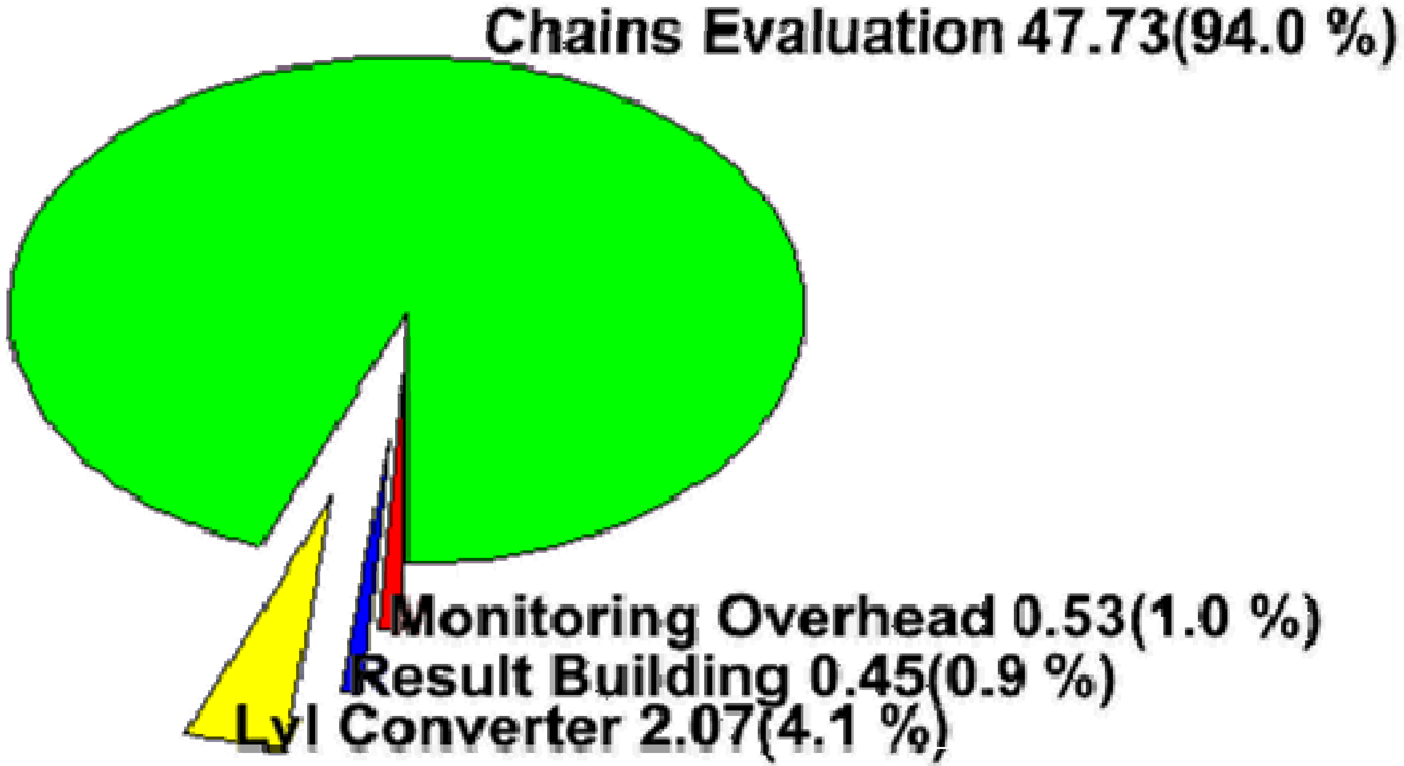
Trigger level 3 (EF)

Start from fine granularity calorimeter cells
Then again, add EF muons …

# HLT steering - overhead time

- Framework is only small fraction of total
- Times consistent with target

**Average time to process an event**



*3 GHz Xeon CPU*

*Code is not yet optimised: expect to reduce steering time.*

*No rejection: worst case scenario.*

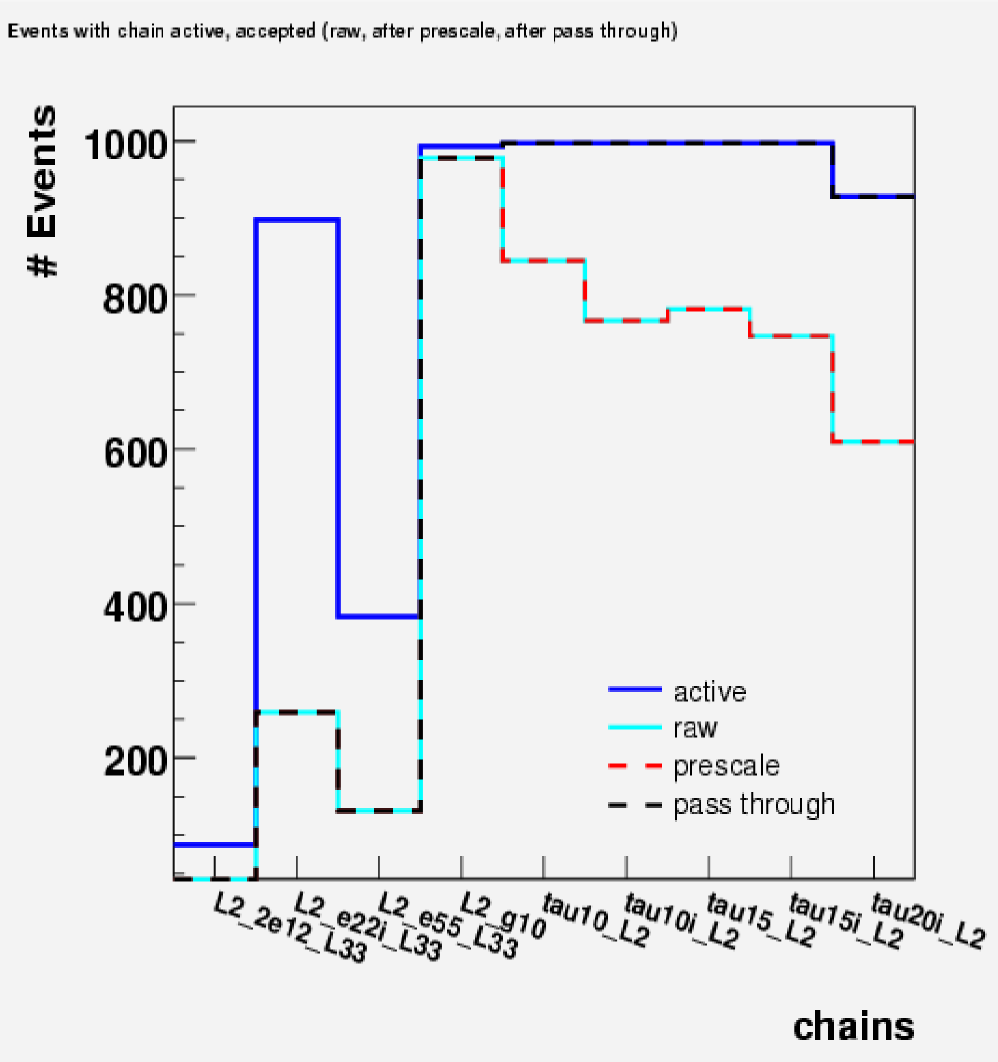
*Inclusive e/$\gamma$ menu for very low lumi.*

*400 top events: atypically busy; average 16 RoIs (~4 EM) per event rather than the usual ~2.*

**Steering has been run live online during cosmic data runs**

# Monitoring

- Vital for online operation
- Steering instrumented to count events
  - By chain and by step
  - before & after pre-scale and pass-through
- Also monitoring RoI position, TE abundances, errors, times
- Histograms gathered and merged for online monitoring display
  - Time stamps allow conversion to rates
- IMonitoredAlgorithm interface allows algorithms to declare variables for histogramming.
  - These are also gathered, merged and available for online display.
  - Same histos used for validation.



# Conclusions

- The HLT steering implements the key features of the HLT event selection strategy:
    - Region of Interest/seeding
    - Steps/early rejection
- Complex menus have been built up
- Caching saves time and simplifies config
- Time overhead is modest
- Well instrumented for monitoring
- Already used in cosmic runs and tech. runs
- More information in the paper

# Other ATLAS T/DAQ presentations at CHEP07

- *Monday, 03 September 2007*
  - [421] **Implementation and Performance of the ATLAS Second Level Jet Trigger**
    - Patricia CONDE MUíñO
    - Poster 1 board 19
  - [437] **Remote Management of nodes in the ATLAS Online Processing Farms**
    - Marc DOBSON
    - Poster 1 board 21
  - [143] **The ATLAS High Level Trigger Steering**
    - by Simon GEORGE (Royal Holloway)
    - (Oak Bay: 16:50 - 17:05)
  - [339] **Event reconstruction algorithms for the ATLAS trigger**
    - by Teresa Maria FONSECA MARTIN (CERN)
    - (Oak Bay: 17:55 - 18:10)
  - [378] **Trigger Selection Software for Beauty physics in ATLAS**
    - by Dmitry EMELIYANOV (RAL)
    - (Oak Bay: 18:10 - 18:25)
- *Wednesday, 05 September 2007*
  - [112] **A software framework for Data Quality Monitoring in ATLAS**
    - by Serguei KOLOS (University of California Irvine)
    - (Oak Bay: 14:35 - 14:50)
  - [332] **The ATLAS DAQ System Online Configurations Database Service Challenge**
    - by Igor SOLOVIEV (CERN/PNPI)
    - (Oak Bay: 15:05 - 15:20)
  - [45] **The ATLAS Trigger: Commissioning with cosmic-rays**
    - by Jamie BOYD (CERN)
    - (Lecture Theatre: 17:30 - 17:45)
- *Thursday, 06 September 2007*
  - [28] **Integration of the Trigger and Data Acquisition Systems in ATLAS**
    - by Benedetto GORINI (CERN)
    - (Oak Bay: 14:50 - 15:05)
  - [50] **The configuration system of the ATLAS Trigger**
    - by Jörg STELZER (CERN, Switzerland)
    - (Lecture Theatre: 15:20 - 15:40)

# Backup slides

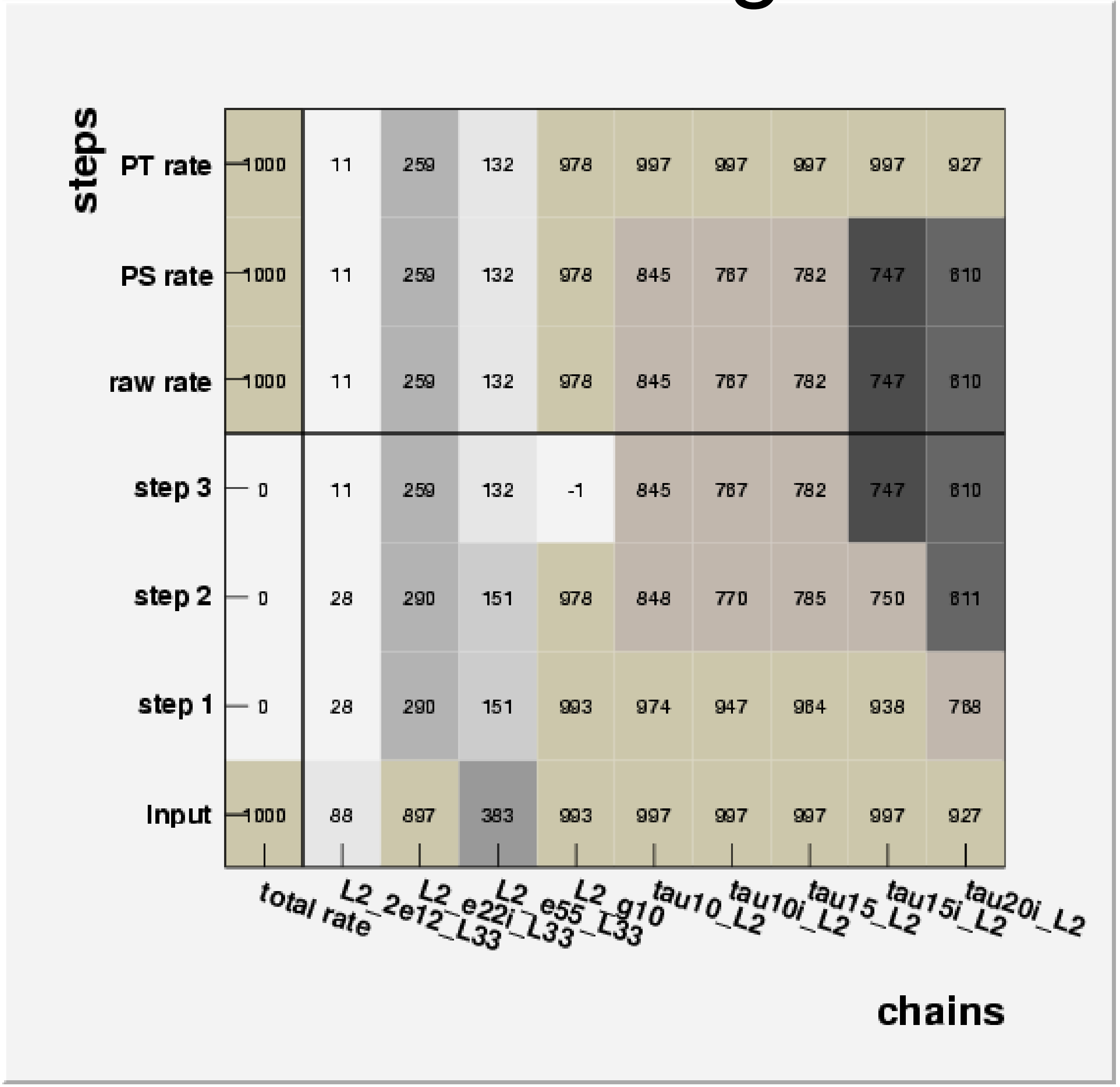
# Time constraints on HLT Algorithms

- LVL2 timing requirement
  - Need to absorb up to 75 kHz LVL1 rate (upgradeable to 100 kHz)
    - Processing of a new event is initiated every 10 μs
  - ~500 1U slots allocated to the LVL2 farm
    - Current baseline: 500 quad-core dual CPU (>=2 GHz), one event per core
      - ⇒ ~40 ms average processing time per event (includes data access & processing time)
- EF timing requirement
  - Need to absorb up to ~3 kHz LVL2 rate
    - Processing of a new event is initiated every ~300 μs
  - ~1800 1U slots for the EF farm
    - Current baseline: 1800 quad-core dual CPU (>=2 GHz), one EF process per core
      - ~4s average processing time per event (includes data access & processing time)
- Aggregate processing power of current baseline HLT farms is consistent with assumption in TDR based on 8 GHz single core dual CPU
- Relative allocation of LVL2 & EF processors is configurable

# Time measurements

- Top events
  - 400 events
  - 0.6 MU, **3.6 EM**, 5.0 TAU, 6.8 J RoIs
  - i.e. tot 16 per event
  - Not typical!
- Menu
  - LVL2 e/g with very low thresholds for commissioning at low lumi
  - 47 chains.
- CPU
  - 3 GHz Xeon
- Time
  - Caching saves ~80% and is on by default
  - Steering overhead (with caching) ~6%
- Compared to realistic setup
  - Expect to use few hundred chains in full menu
  - Expect average ~2 RoIs per event with high lumi thresholds and typical events
  - Online technical runs show times in more realistic setup are compatible (or close) with no. of available processors

# Monitoring



| steps | total rate | L2_2e12_L33 | L2_e22i_L33 | L2_e55_L33 | L2_g10 | tau10_L2 | tau10i_L2 | tau15_L2 | tau15i_L2 | tau20i_L2 |
|---|---|---|---|---|---|---|---|---|---|---|
| PT rate | 1000 | 11 | 259 | 132 | 978 | 997 | 997 | 997 | 997 | 927 |
| PS rate | 1000 | 11 | 259 | 132 | 978 | 845 | 787 | 782 | 747 | 810 |
| raw rate | 1000 | 11 | 259 | 132 | 978 | 845 | 787 | 782 | 747 | 810 |
| step 3 | 0 | 11 | 259 | 132 | -1 | 845 | 787 | 782 | 747 | 810 |
| step 2 | 0 | 28 | 290 | 151 | 978 | 848 | 770 | 785 | 750 | 811 |
| step 1 | 0 | 28 | 290 | 151 | 993 | 974 | 947 | 964 | 938 | 768 |
| Input | 1000 | 88 | 897 | 383 | 993 | 997 | 997 | 997 | 997 | 927 |