Accelerating the ATLAS Trigger System with Graphical Processing Units

8th IDPASC Student Workshop
16/10/2024

Nuno dos Santos Fernandes

CERN
REPÚBLICA PORTUGUESA
FCT Fundação para a Ciência e a Tecnologia
TÉCNICO LISBOA
LIP LABORATÓRIO DE INSTRUMENTAÇÃO E FÍSICA EXPERIMENTAL DE PARTÍCULAS
ATLAS EXPERIMENT

ATLAS & Trigger

ATLAS and its Trigger System

- **A** **T**oroidal **L**HC **A**pparatu**s**: one of the two **general-purpose detectors** at the LHC
- The **ATLAS Trigger** is used to **filter the detected events** to ensure a **manageable output rate**
  - **Two stages**:
    - **Hardware-based** (**Level 1**/Level 0)
    - **Software-based** (**High-Level Trigger**/Event Filter)
- The **High-Luminosity LHC Upgrade** will **increase the luminosity**, making **event reconstruction more computationally demanding**
- The **Phase II ATLAS upgrade** needed for the High-Luminosity LHC increases event rate at the software-based stage from **1 kHz to 10 kHz**
- This **higher computational load** requires **more computing power** and/or **better optimization**
- Alternative: **hardware acceleration**
  - Ongoing studies for both FPGA and **GPU acceleration**

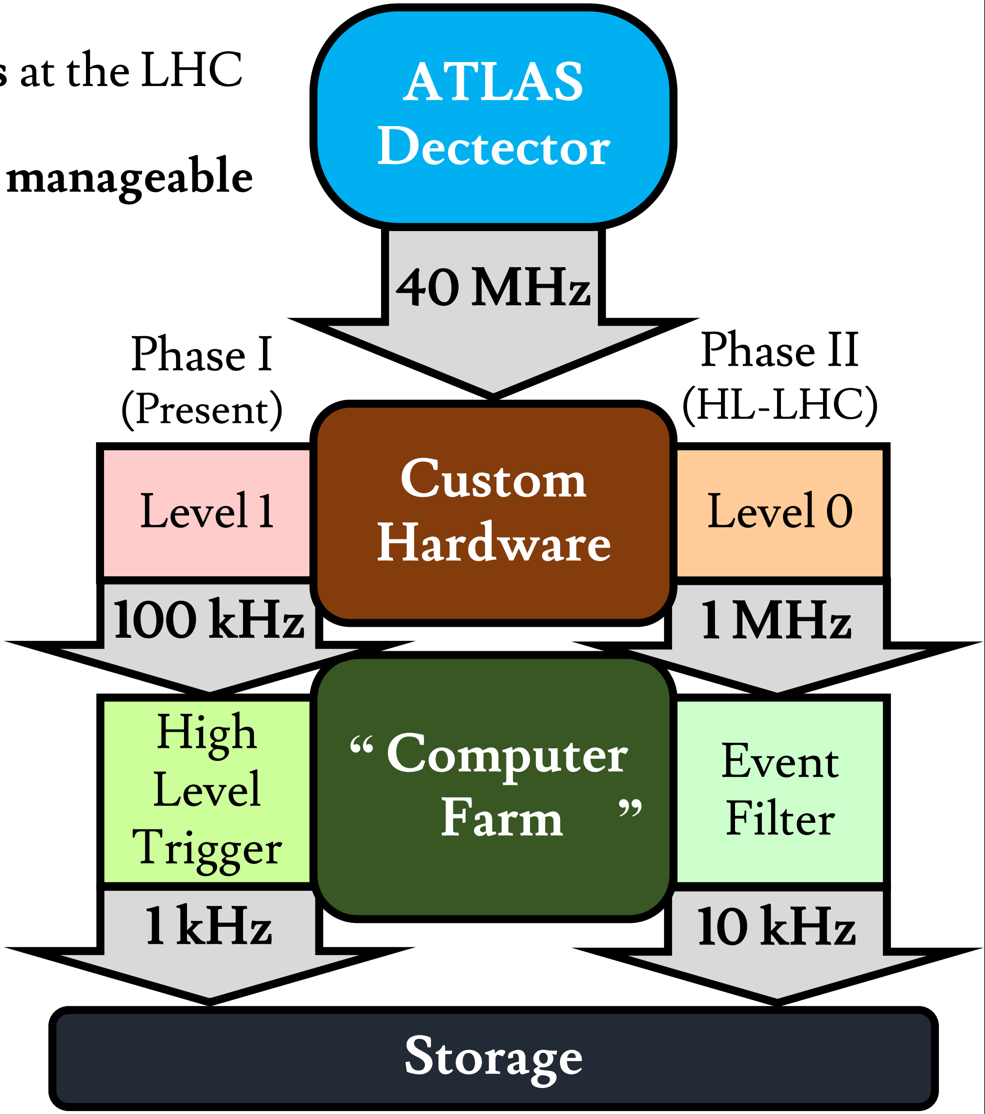
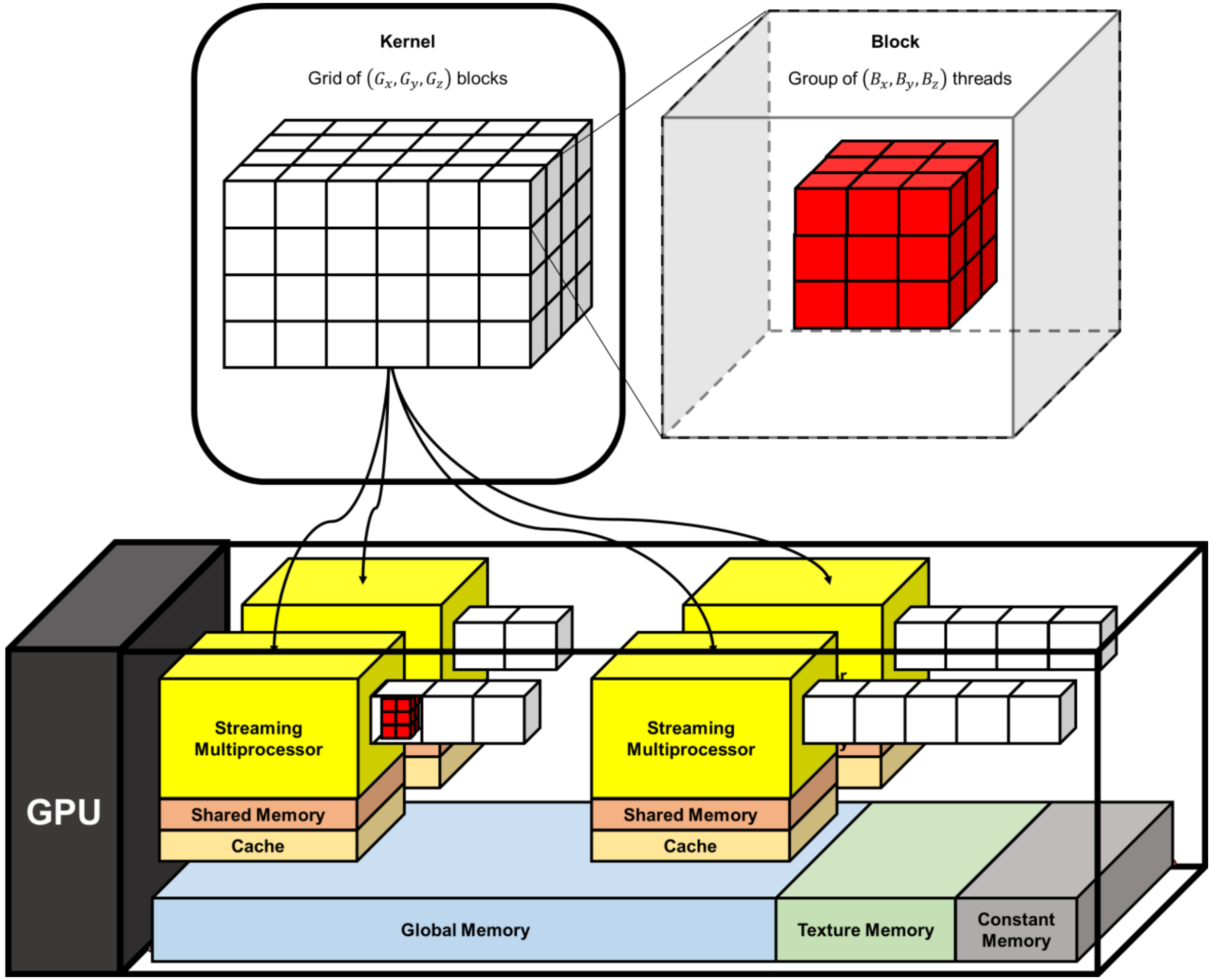


Diagram of the ATLAS Trigger System

GPU Programming

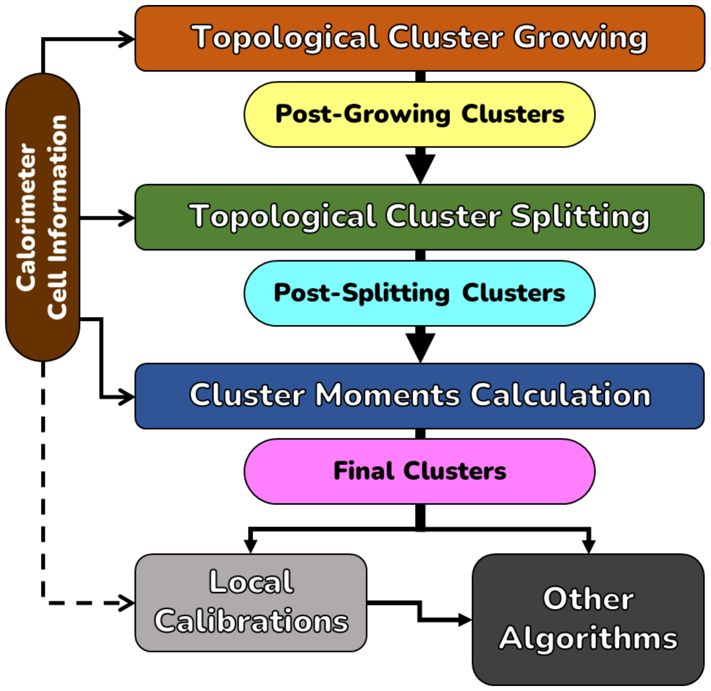
- **<u>G</u>**raphical **<u>P</u>**rocessing **<u>U</u>**nits
- Developed and designed to render 3D graphics
- Highly parallel operations → highly parallel design
- "SIMT": *Single Instruction – Multiple Threads*
- Branching is problematic
- Memory access patterns must be carefully considered

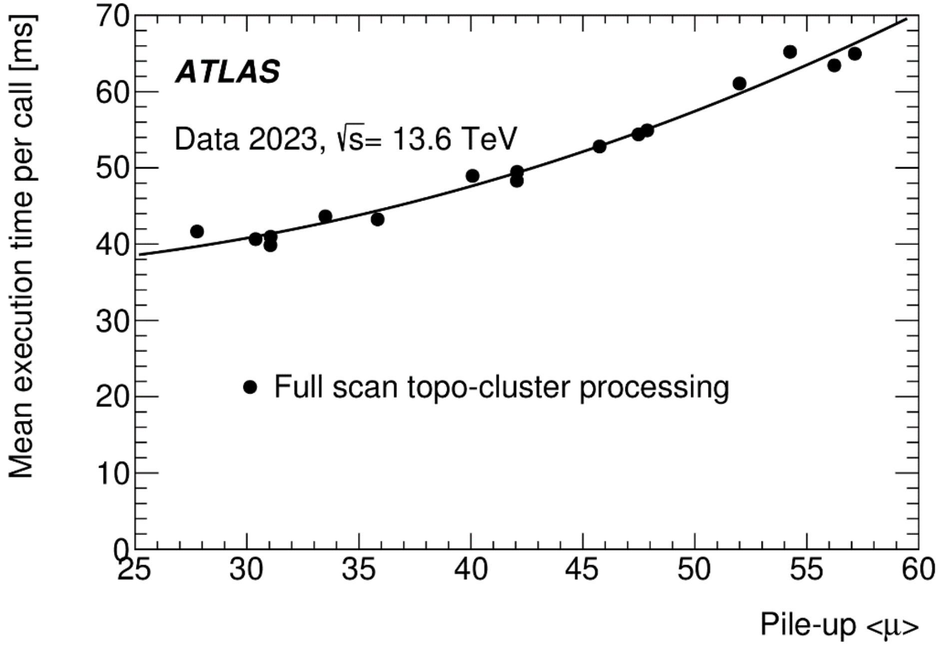


Calorimeter Clustering Algorithms

Topological Clustering

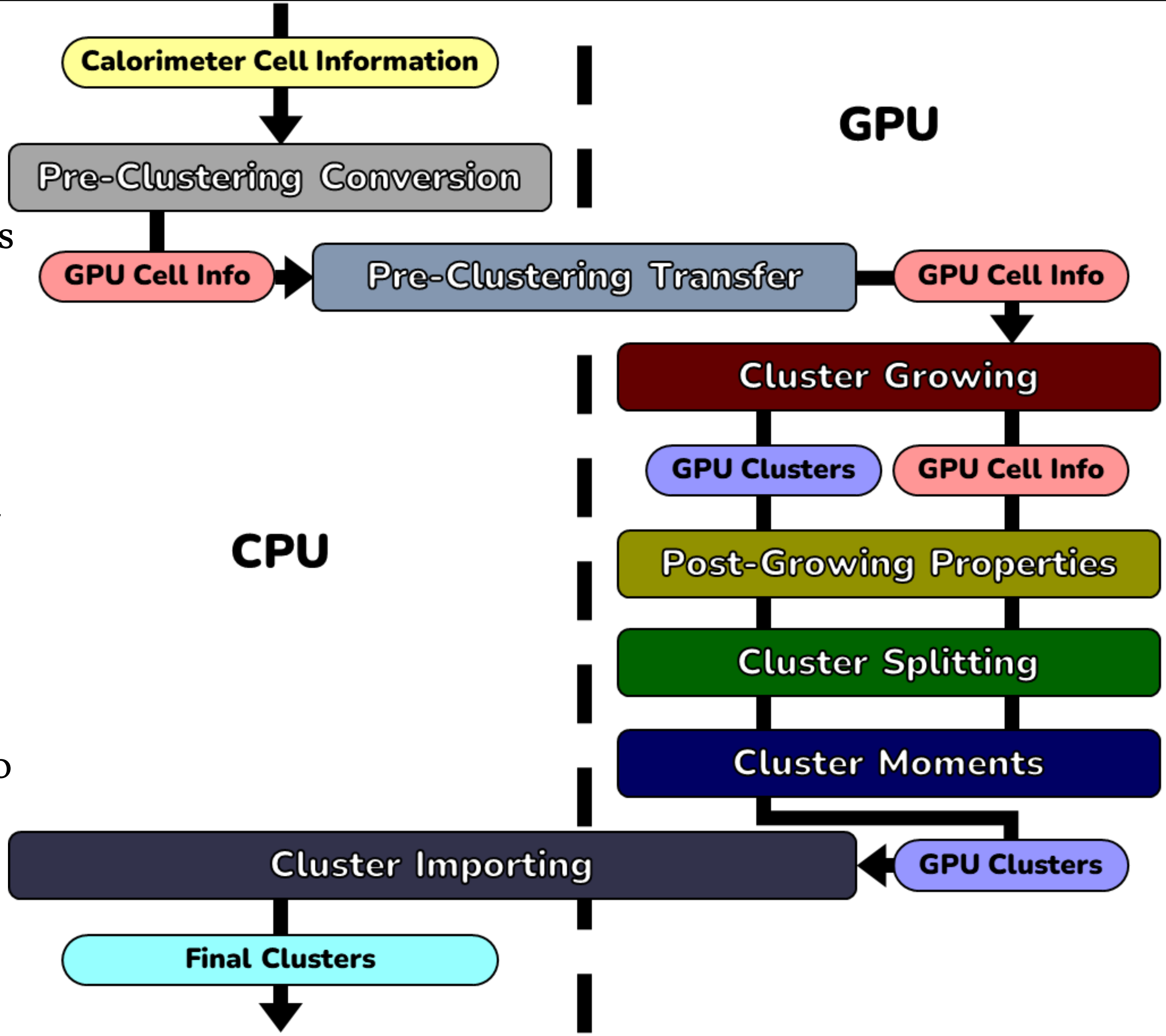
- **Topological Clustering** is the currently used approach for **calorimeter reconstruction** in ATLAS
  - Among the **top 20th most computationally demanding algorithms** within the ATLAS trigger
- Three main steps: **cluster growing, cluster splitting, cluster moments calculation**
- Clustering typically groups up **several tens of calorimeter cells, some clusters** may be **significantly larger**
- **Several hundred to a few thousand clusters** per event, depending on the **physical process**
- Significant **dependence on the number of collisions per bunch crossing** (μ) in terms of the **execution time**





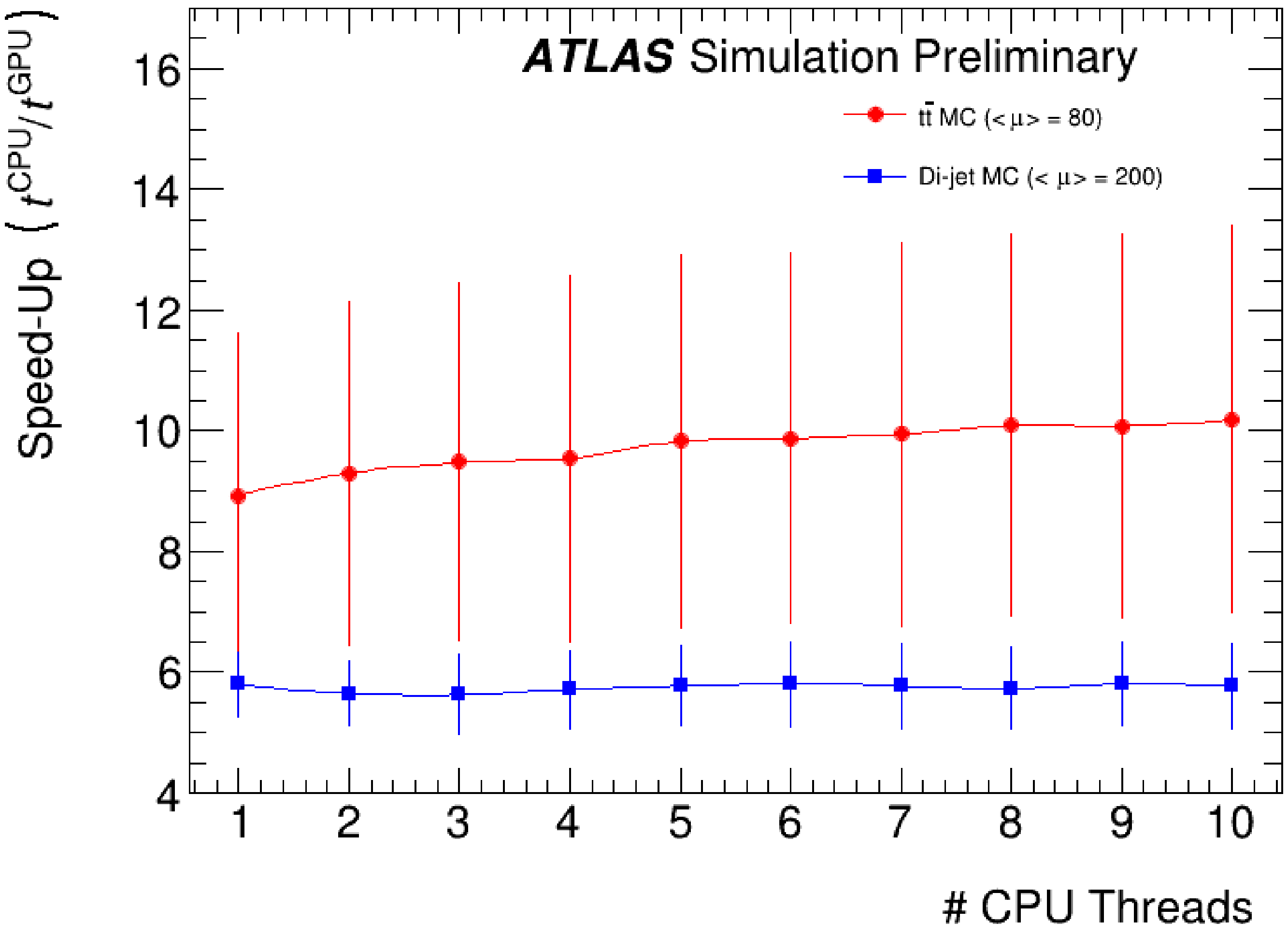
Topo-Automaton Clustering

- Topological clustering is **not accelerator-friendly**: a **different algorithmic approach** is needed
  - **Tags** express the cluster assignment of the cells
  - **Tag propagation rules** to grow and split the clusters
- Formally equivalent to a **cellular automaton**, hence **Topo-Automaton Clustering**
- **Fully implemented in the GPU** using **CUDA**
- **100% agreement in cell assignment** can be achieved between CPU and GPU, any **reasons for differences** if certain options are taken are **fully understood**
- **Basic cluster properties** (e. g.: energy, η, φ) yield **similar values** (within floating point accuracy)
- Some **cluster moments** have **greater differences** due to accumulated and **compounded floating point errors**
- The **data structures** used in the CPU part of the code **cannot be used directly** in the GPU, so we need to **convert between the two representations**



Results

Speed-up from GPU Acceleration in Relation to the CPU Implementation



We currently achieve a **speed-up of ~5.9 for di-jets, ~8.9 for** $t\bar{t}$, considering all data conversions and transfers.
The speed-up depends on the complexity of the event (number and size of the clusters), mostly due to CPU scaling.

Breakdown of GPU Execution Times

- **Main bottleneck: converting the GPU data structures** representing the clusters **back to CPU-compatible structures**

Step		t-$\bar{t}$ Events			Jet Events		
		Time (µs)	Fraction of Total Time		Time (µs)	Fraction of Total Time	
Pre-Clustering Conversion		1441 ± 225	$9.24 \pm 1.58\%$		1128 ± 88	$13.24 \pm 1.14\%$	
Pre-Clustering Transfer		266 ± 8	$1.71 \pm 0.18\%$		248 ± 15	$2.92 \pm 0.30\%$	
Growing	Cell Classification	61 ± 2	$15.76 \pm 1.77\%$	$2.53 \pm 0.35\%$	56 ± 3	$20.38 \pm 1.30\%$	$3.24 \pm 0.25\%$
	Neighbour Pair Creation	159 ± 8	$40.68 \pm 3.23\%$		114 ± 6	$41.45 \pm 1.98\%$	
	Tag Propagation	175 ± 46	$43.55 \pm 4.87\%$		106 ± 13	$38.17 \pm 2.59\%$	
	Total	396 ± 53	—		276 ± 16	—	
Post-Growing Property Calculation		74 ± 22	$0.47 \pm 0.14\%$		55 ± 2	$0.65 \pm 0.05\%$	
Splitting	Neighbour Pair Creation	409 ± 28	$29.23 \pm 2.73\%$	$9.02 \pm 1.16\%$	287 ± 14	$33.80 \pm 1.62\%$	$9.97 \pm 0.57\%$
	Local Maxima Identification	88 ± 7	$6.25 \pm 0.53\%$		57 ± 3	$6.77 \pm 0.33\%$	
	Secondary Maxima Exclusion	229 ± 16	$16.48 \pm 2.25\%$		230 ± 14	$27.15 \pm 2.11\%$	
	Main Tag Propagation	642 ± 194	$44.44 \pm 5.46\%$		236 ± 49	$27.53 \pm 3.54\%$	
	Finalization	50 ± 5	$3.60 \pm 0.31\%$		40 ± 3	$4.75 \pm 0.27\%$	
	Total	1417 ± 226	—		851 ± 67	—	
Cluster Moments		1422 ± 134	$9.07 \pm 0.58\%$		889 ± 52	$10.43 \pm 0.51\%$	
Post-Clustering Transfer + Conversion		10679 ± 137	$67.77 \pm 2.59\%$		5094 ± 690	$59.27 \pm 2.26\%$	
Total		15724 ± 1630	—		8565 ± 847	—	

Summary and Future Efforts

Summary and Future Efforts

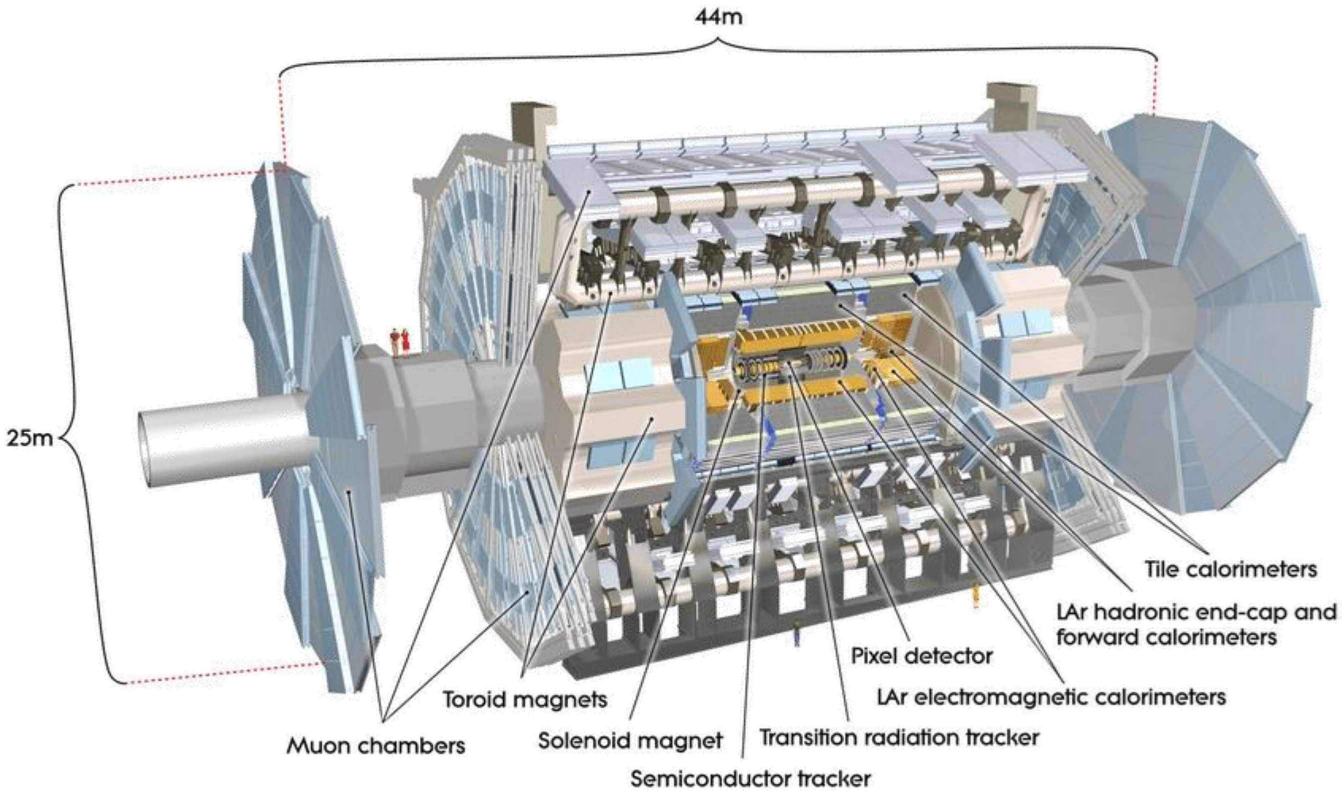
- Topo-Automaton Clustering **fully implemented** and working, for **cluster growing**, **cluster splitting** and **cluster moments calculation**, with configurability on a par with the CPU implementation (essentially, **drop-in replacement**)
  - First completed prototype for ATLAS Phase II, well ahead of schedule
- A very significant **speed-up** was found (factor of **~5.9 for di-jet events, ~8.9 for denser $t\bar{t}$ events**)
  - A significant portion of the GPU event processing time **(60~70%)** is spent in **data conversions**
- **A general solution** to **mitigate the data structure conversion overhead** is under development (Marionette)
  - **Integration** with the **current implementation** of Topo-Automaton Clustering to follow
  - At least a **factor of 2** improvement on the current **speed-up** seems feasible
- **Lessons learned** and **experience gained** from this development have fed back into general hardware acceleration-related development within ATLAS and in particular the ATLAS Trigger
  - Currently co-coordinator of HLT Calo, responsible for the calorimeter reconstruction in the High Level Trigger
- A **final decision** on **using this approach in the ATLAS Trigger** depends on a **general technical assessment** of the feasibility and/or performance of **GPU-accelerated algorithms**, being scheduled for **next year**
  - The approach is **also being considered** for **offline reconstruction** on **grid sites where GPUs are available**

Thank you for your attention!

Backup Slides

The ATLAS Experiment

- **A** **T**oroidal **L**HC **A**pparatu**s**
- One of the two **general-purpose detectors** at the LHC
- Three layers:
  - Inner Detector
  - **Calorimeters**
  - Muon Spectrometers
- 10^8 electronic channels
  - 187652 calorimeter cells with multiple gain paths to optimize resolution *versus* dynamic range of operation



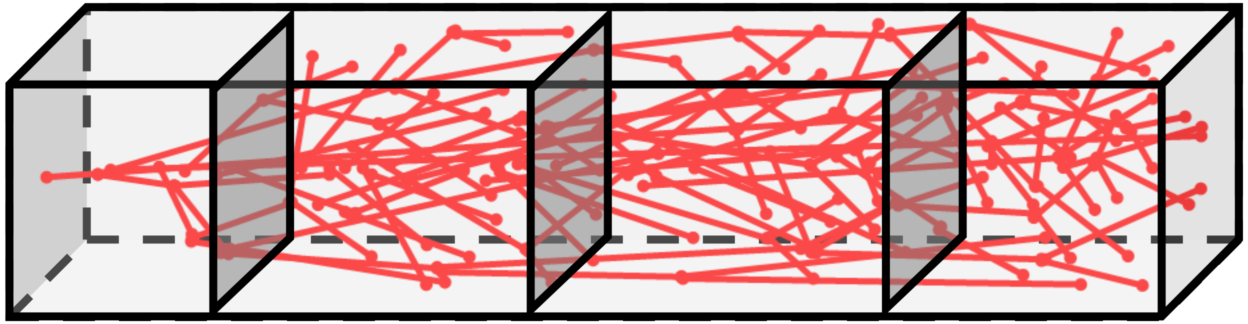
Conditions of the Benchmarking

- Samples correspond to two kinds of **Monte-Carlo simulated** events:
  - $t\bar{t}$ **events**: 3000 events, $\mu = 80$
  - **di-jet events**: 10000 events, $\mu = 200$
- Results were obtained on a remote server provided by the Brookhaven National Laboratory:
  GPU is a Tesla P100, CPU is a Xeon E5-2695 v4
- Time measurements were based on a **per-thread clock**
  - For a single thread, “any clock” would work
    - The CPU – GPU comparison is a bit lopsided, though…
  - For more threads, **timing and speed-up are representative, but throughput is a best-case estimate**
    - Essentially, we are assuming everything is always running in parallel
  - This is due to **several limitations** when trying to **benchmark within the ATLAS software**

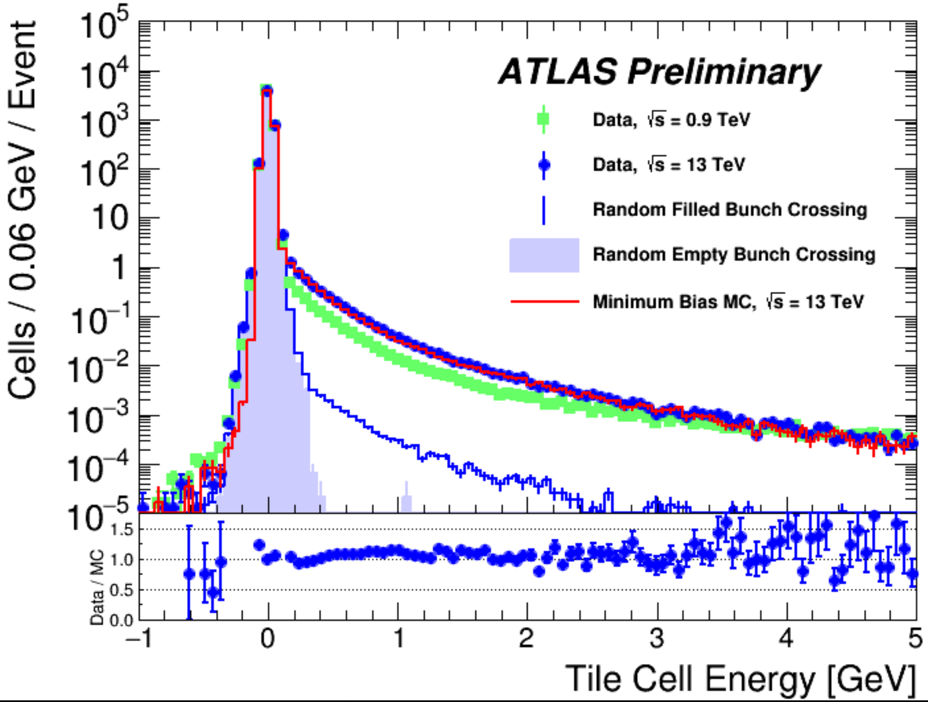
Breakdown of GPU Execution Times

Step		t-$\bar{t}$ Events			Jet Events		
		Time (μs)	Fraction of Total Time		Time (μs)	Fraction of Total Time	
Pre-Clustering Conversion		1441 ± 225	$9.24 \pm 1.58\%$		1128 ± 88	$13.24 \pm 1.14\%$	
Pre-Clustering Transfer		266 ± 8	$1.71 \pm 0.18\%$		248 ± 15	$2.92 \pm 0.30\%$	
Growing	Cell Classification	61 ± 2	$15.76 \pm 1.77\%$	$2.53 \pm 0.35\%$	56 ± 3	$20.38 \pm 1.30\%$	$3.24 \pm 0.25\%$
	Neighbour Pair Creation	159 ± 8	$40.68 \pm 3.23\%$		114 ± 6	$41.45 \pm 1.98\%$	
	Tag Propagation	175 ± 46	$43.55 \pm 4.87\%$		106 ± 13	$38.17 \pm 2.59\%$	
	Total	396 ± 53	—		276 ± 16	—	
Post-Growing Property Calculation		74 ± 22	$0.47 \pm 0.14\%$		55 ± 2	$0.65 \pm 0.05\%$	
Splitting	Neighbour Pair Creation	409 ± 28	$29.23 \pm 2.73\%$	$9.02 \pm 1.16\%$	287 ± 14	$33.80 \pm 1.62\%$	$9.97 \pm 0.57\%$
	Local Maxima Identification	88 ± 7	$6.25 \pm 0.53\%$		57 ± 3	$6.77 \pm 0.33\%$	
	Secondary Maxima Exclusion	229 ± 16	$16.48 \pm 2.25\%$		230 ± 14	$27.15 \pm 2.11\%$	
	Main Tag Propagation	642 ± 194	$44.44 \pm 5.46\%$		236 ± 49	$27.53 \pm 3.54\%$	
	Finalization	50 ± 5	$3.60 \pm 0.31\%$		40 ± 3	$4.75 \pm 0.27\%$	
	Total	1417 ± 226	—		851 ± 67	—	
Cluster Moments		1422 ± 134	$9.07 \pm 0.58\%$		889 ± 52	$10.43 \pm 0.51\%$	
Importing	Cluster Number Transfer	21 ± 1	$0.20 \pm 0.03\%$	$67.77 \pm 2.59\%$	17 ± 1	$0.33 \pm 0.04\%$	$59.27 \pm 2.26\%$
	Cluster Info Transfer	250 ± 39	$2.35 \pm 0.27\%$		102 ± 20	$2.00 \pm 0.21\%$	
	Cluster Creation + Cell Info Transfer	395 ± 83	$3.68 \pm 0.46\%$		147 ± 24	$2.89 \pm 0.20\%$	
	Cell Cycle	2892 ± 444	$27.10 \pm 2.42\%$		1624 ± 117	$32.18 \pm 2.45\%$	
	Cluster Ordering	192 ± 41	$1.79 \pm 0.28\%$		77 ± 15	$1.52 \pm 0.16\%$	
	Cluster Filling	2022 ± 336	$18.88 \pm 1.41\%$		944 ± 165	$18.46 \pm 1.18\%$	
	Moments Transfer	3.7 ± 0.5	$0.04 \pm 0.01\%$		3.8 ± 1.0	$0.07 \pm 0.02\%$	
	Moments Filling	4903 ± 697	$45.96 \pm 3.43\%$		2178 ± 389	$42.54 \pm 2.36\%$	
	Total	10679 ± 1377	—		5094 ± 690	—	
Total		15724 ± 1630	—		8565 ± 847	—	

- Reconstruction of **showers** generated by outgoing particles in the calorimeters of the ATLAS experiment

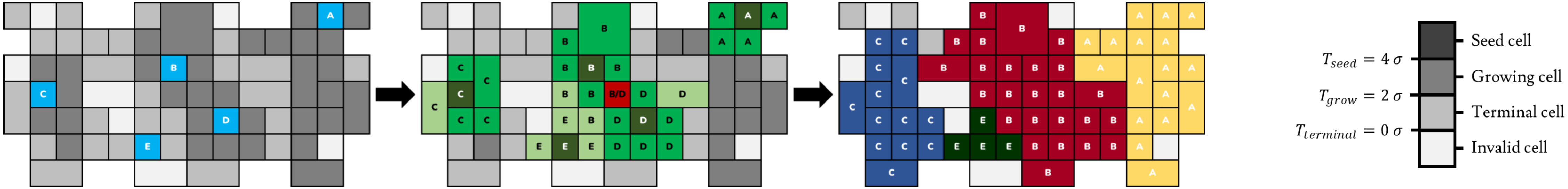


- Showers **deposit their energy** in a finite region of space: a **calorimeter cell**
- Calorimeter cells organized in up to **28 different sampling layers**
- Two main sources of **noise**: electronic read-out and pile-up
  - The **noise estimate** is typically a function of the gain of the cell
  - For the **Tile calorimeter**, the electronic noise can be estimated by a **two-Gaussian model**, which involves more sophisticated computations (inverse error function of error functions)

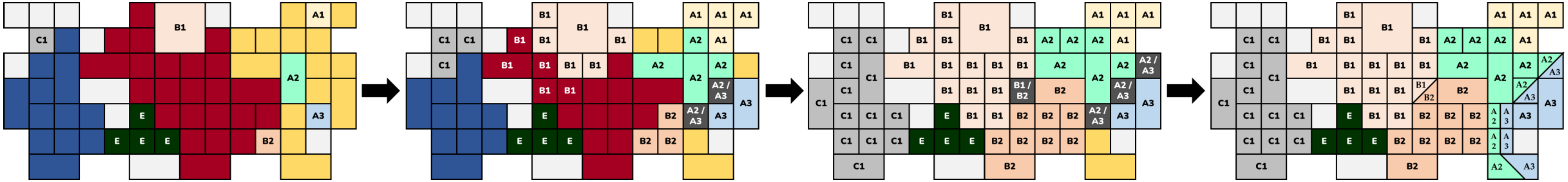


Topological Clustering

- Two main algorithmic stages:
  - **Cluster growing**: iteratively **assign cells** to clusters based on the **SNR** (classify cells as **seed**, **growing** or **terminal**, clusters grow out from the seeds to their neighbouring cells in an order defined by the SNR of the seed, clusters are merged if they touch through growing cells)



  - **Cluster splitting**: split the clusters around **local maxima of the energy** to distinguish different objects travelling in the same direction (identify local maxima, exclude maxima from certain regions of the detector that overlap in certain directions to favour layers with greater radiation depth, start growing the clusters to neighbouring cells in an order defined by the energy of the cells, cells that can belong to more than one maximum are shared, shared cells grow clusters only in the end and are weighted based on the energy and distance to the centroid)



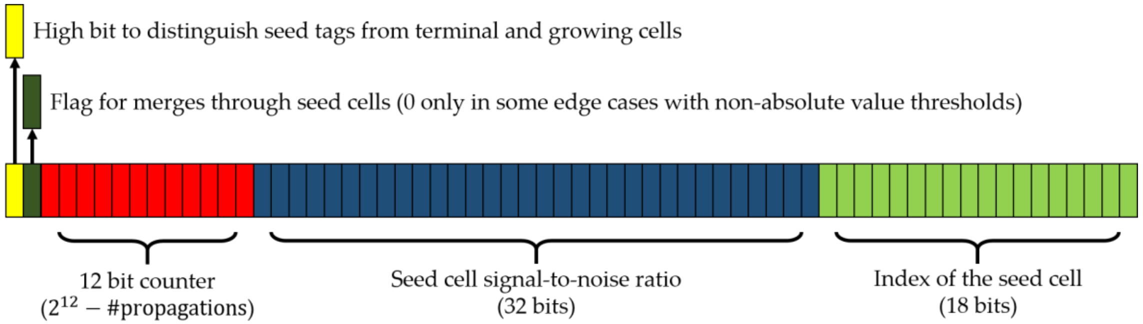
Limitations of Topological Clustering

- **Resizing** a container is **difficult to do in parallel**, and it goes **against the memory model** of both GPUs and FPGAs
- Topological Clustering involves **keeping track of multiple lists**[1] of cells, especially for **cluster splitting**
- The **clusters** themselves are also **expressed as lists**[1] which must be **resized** as we add and remove cells
- For a more **parallel-friendly** implementation, we can instead **mark the cells** that belong to each cluster with a "**tag**"
  - By constructing these **tags** appropriately, the **sorting steps can be skipped** entirely: floating point numbers that follow the **IEEE-754 standard** can be put in a "**total ordering**" where the **bit patterns**, interpreted as integers, are **ordered in the same way** as the original floating point numbers
  - By defining a **set of rules** for how these tags are propagated **from a cell to its neighbours**, one can replicate the entire behaviour of the iterative parts of cluster growing and cluster splitting while only **considering each pair of neighbours independently from each other** (potentially in parallel, as long as tag updates are thread-safe)
- Since we have both a **state** for each cell and can specify the **rules for how that state changes** based on the neighbourhood, this is equivalent to a **cellular automaton**, hence **Topo-Automaton Clustering**

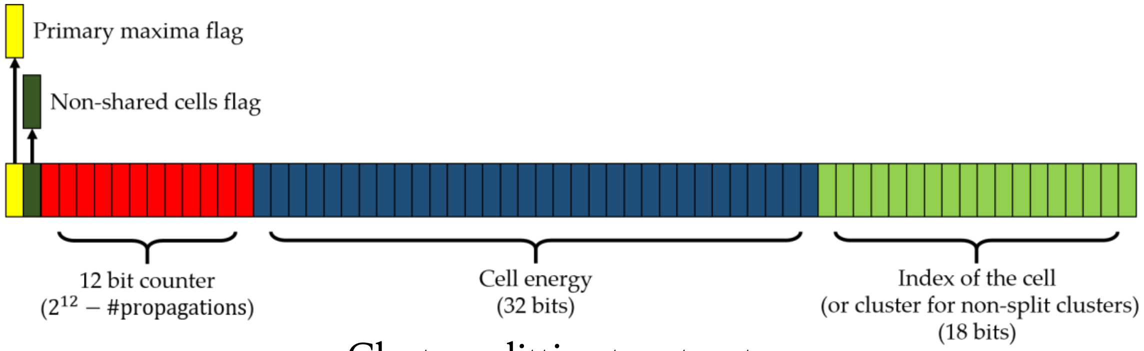
[1] – "List" is used here in the sense of an ordered collection of items; specifically, they correspond to dynamically allocated arrays, or "vectors" in C++.

Topo-Automaton Clustering

- **Cluster tags** are **64-bit integers** with **specific structure**:



Cluster growing tag structure



Cluster splitting tag structure

- The tags are **propagated through pairs of neighbouring cells** satisfying the conditions for clusters to expand
  - We handle each **pair of cells in parallel**, using appropriate **atomic operations** when needed
- **Additional logic** (e. g. keeping a cell to cluster index table) **reduces the number of iterations**
- All necessary **temporary information stored** in the same block of memory meant to hold the **cluster moments** (calculated only at the end), **everything can be pre-allocated**
  - Total **per event memory** footprint is **~80 MB**
  - Cell geometry and neighbourhood relations also need to be stored: **~100 MB** of **constant information**

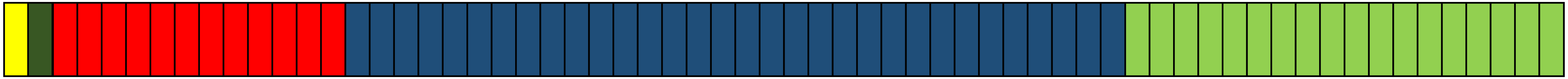
High bit to distinguish valid tags from terminal and growing cells

Flag for preventing merges through seed cells
(1 only in some edge cases with non-absolute value thresholds)

12 bit counter ($2^{12} - 1 - \#\text{propagations}$)

Assumptions:

- Less than $2^{16} = 65536$ clusters
- Less than 2^{12} propagation steps



Signal-to-noise ratio
in total ordering

Index of the seed cell

Topo-Automaton Cluster Splitting – Anatomy of a Tag

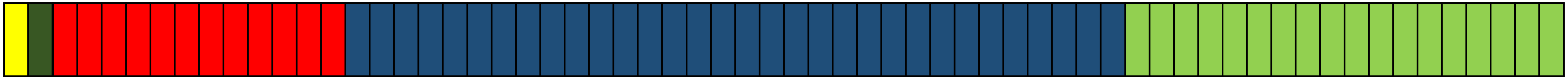
Primary maxima flag

Non-shared cells flag

12 bit counter ($2^{12} - 1 -$ #propagations)

Assumptions:

- Less than $2^{16} = 65536$ clusters
- Less than 2^{12} propagation steps



Cell energy
(all set for original clusters)

Index of the cell (or cluster index for original clusters)