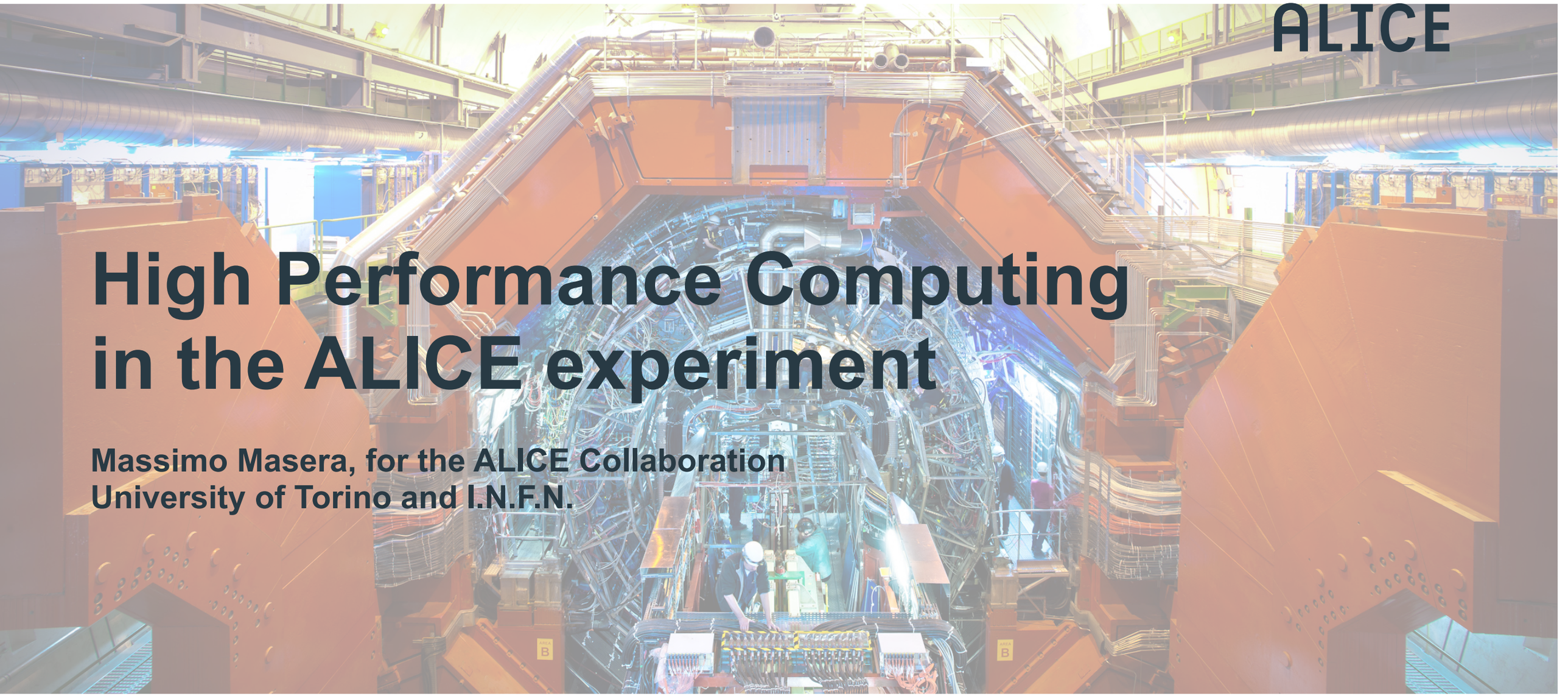
A Large Ion Collider Experiment



High Performance Computing in the ALICE experiment

Massimo Masera, for the ALICE Collaboration
University of Torino and I.N.F.N.

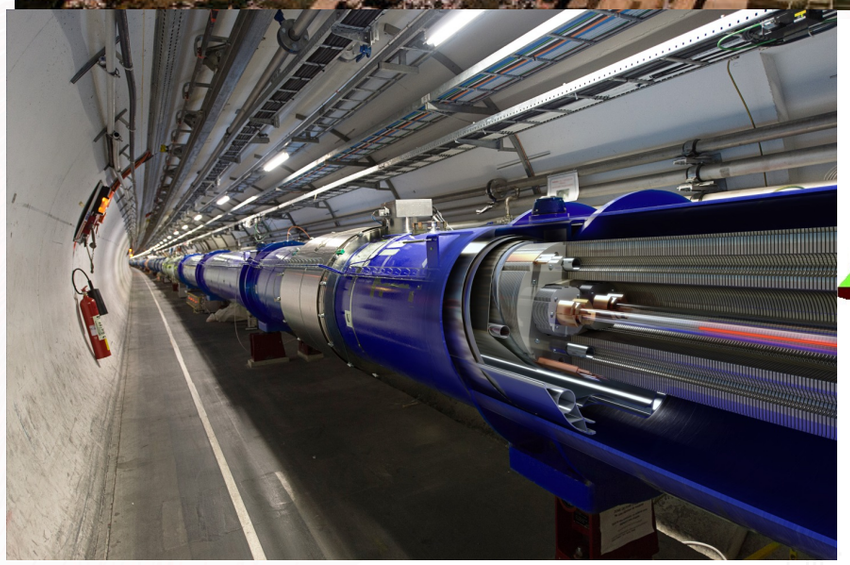
MHPC Workshop on High Performance Computing | Trieste
26/02/2016

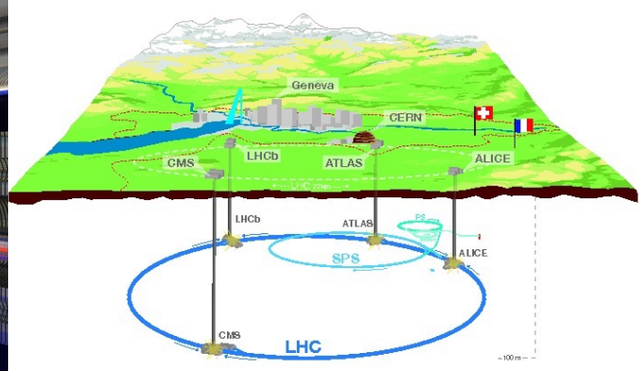
Outline

- The Large Hadron Collider and the ALICE experiment
- Offline computing: embarrassingly parallel approach
- The ALICE High Level Trigger
- Online computing: TPC reconstruction as an example of HPC in ALICE
- The ALICE upgrade for the LHC Run 3: the O^2 system
- The Inner Tracking System standalone tracking
- Conclusions

LHC

- The Large Hadron Collider (LHC) is the largest and most powerful proton and ion collider in the world.
- The present centre-of-mass energy is:
  - »13 TeV for pp collisions
  - »5.02 TeV per nucleon pair for Pb-Pb collisions
- 4 major experiments: ALICE, ATLAS, CMS and LHCb
- ALICE (A Large Ion Collider Experiment) is designed primarily to study nucleus-nucleus collisions.

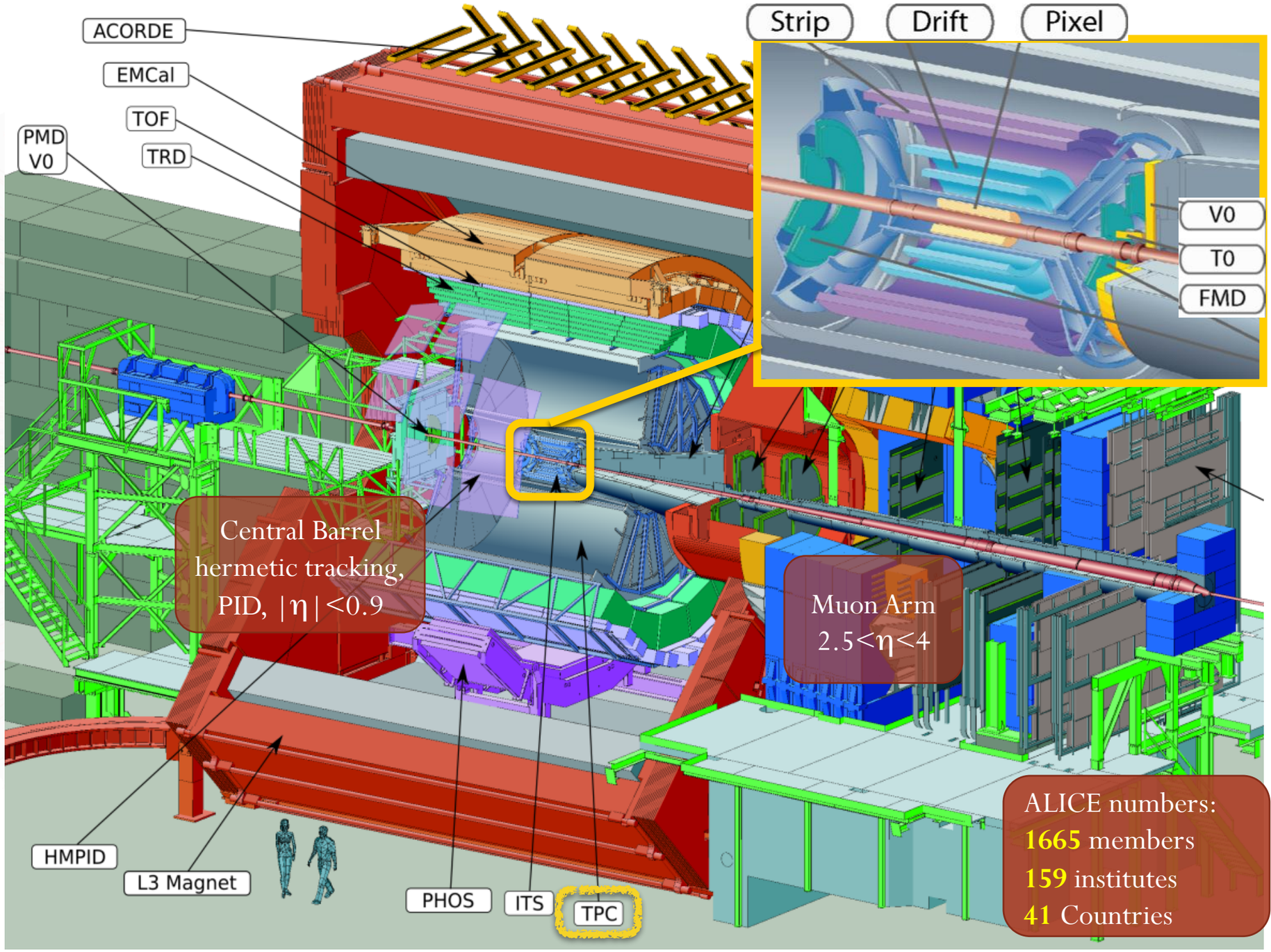




ALICE

- Designed to reconstruct and identify charged particles in a central rapidity window -> central barrel down to low transverse momentum (p_T~100 MeV/c for pions)
- Main vertexing and tracking detectors: ITS and TPC
- Event recording bandwidth: 1.25 GB/s for Pb-Pb events
- Data (raw and reconstructed) on permanent storage: few PB/year. Overall stored data:
  - **tape: ~45 PB**
  - **storage: ~55 PB**
- Reconstruction: almost completely offline

Acronyms:
- ITS - Inner Tracking System
- TPC - Time Projection Chamber



Offline computing in ALICE

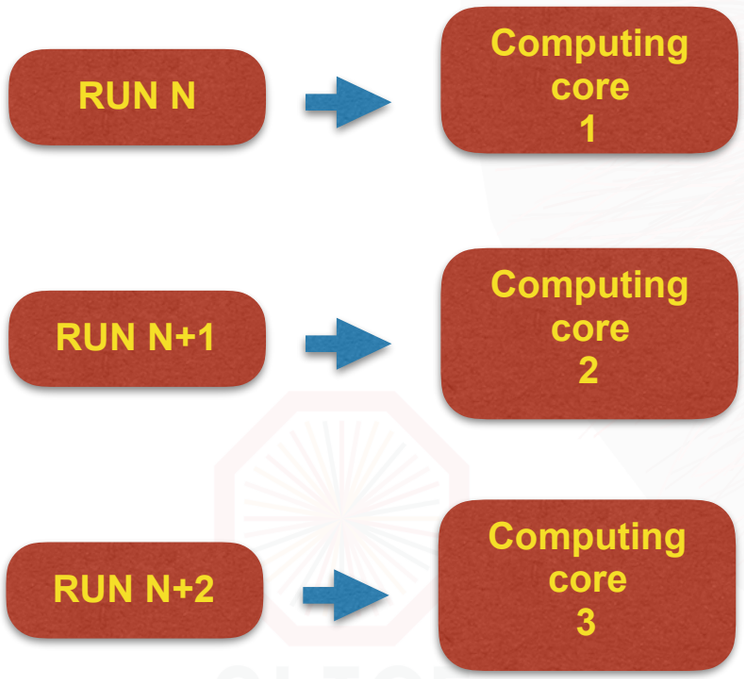
- When a Pb-Pb (pp or p-Pb) collision meets a predefined trigger condition, the corresponding data are collected: these data form an **event**.
- Ideally, an event corresponds to a single Pb-Pb (pp…) collision, even though pile-up of several collisions may occur, depending on the interaction rate.
- The data collected in a continuous data taking period (~ few hours) within one LHC fill, constitute a **run**.
- The events in a run must be processed independently; they share only the same data taking conditions.
- Different runs are processed independently on different computing cores (if no hyper-threading):
  - » at a given moment a computing farm with N cores processes N events in parallel;
  - » embarrassingly parallel computing.
- Presently, no further parallelization within a single event is attempted.

Offline computing in ALICE

- When a Pb-Pb (pp or p-Pb) collision meets a predefined trigger condition, the corresponding data are collected: these data form an **event**.
- Ideally, an event corresponds to a single Pb-Pb (pp…) collision, even though pile-up of several collisions may occur, depending on the interaction rate.
- The data collected in a continuous data taking period (~ few hours) within one LHC fill, constitute a **run**.
- The events in a run must be processed independently; they share only the same data taking conditions.
- Different runs are processed independently on different computing cores (if no hyper-threading):
  - » at a given moment a computing farm with N cores processes N events in parallel;
  - » embarrassingly parallel computing.
- Presently, no further parallelization within a single event is attempted.

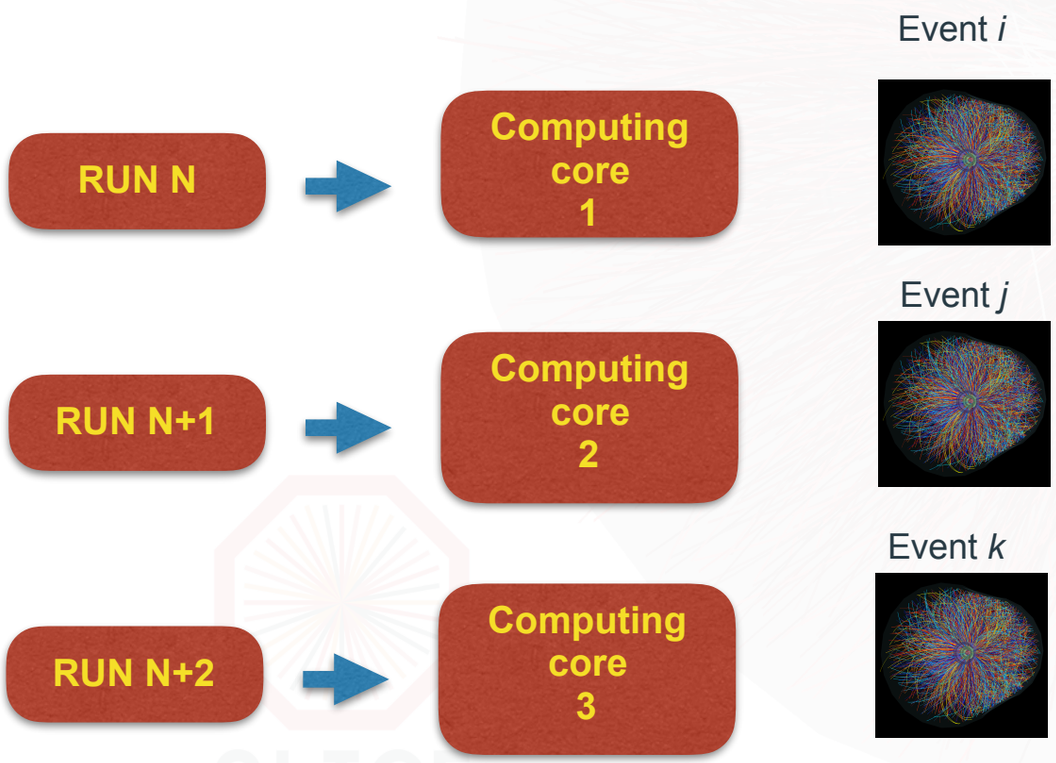
Offline computing in ALICE

- When a Pb-Pb (pp or p-Pb) collision meets a predefined trigger condition, the corresponding data are collected: these data form an **event**.
- Ideally, an event corresponds to a single Pb-Pb (pp…) collision, even though pile-up of several collisions may occur, depending on the interaction rate.
- The data collected in a continuous data taking period (~ few hours) within one LHC fill, constitute a **run**.
- The events in a run must be processed independently; they share only the same data taking conditions.
- Different runs are processed independently on different computing cores (if no hyper-threading):
  - » at a given moment a computing farm with N cores processes N events in parallel;
  - » embarrassingly parallel computing.
- Presently, no further parallelization within a single event is attempted.



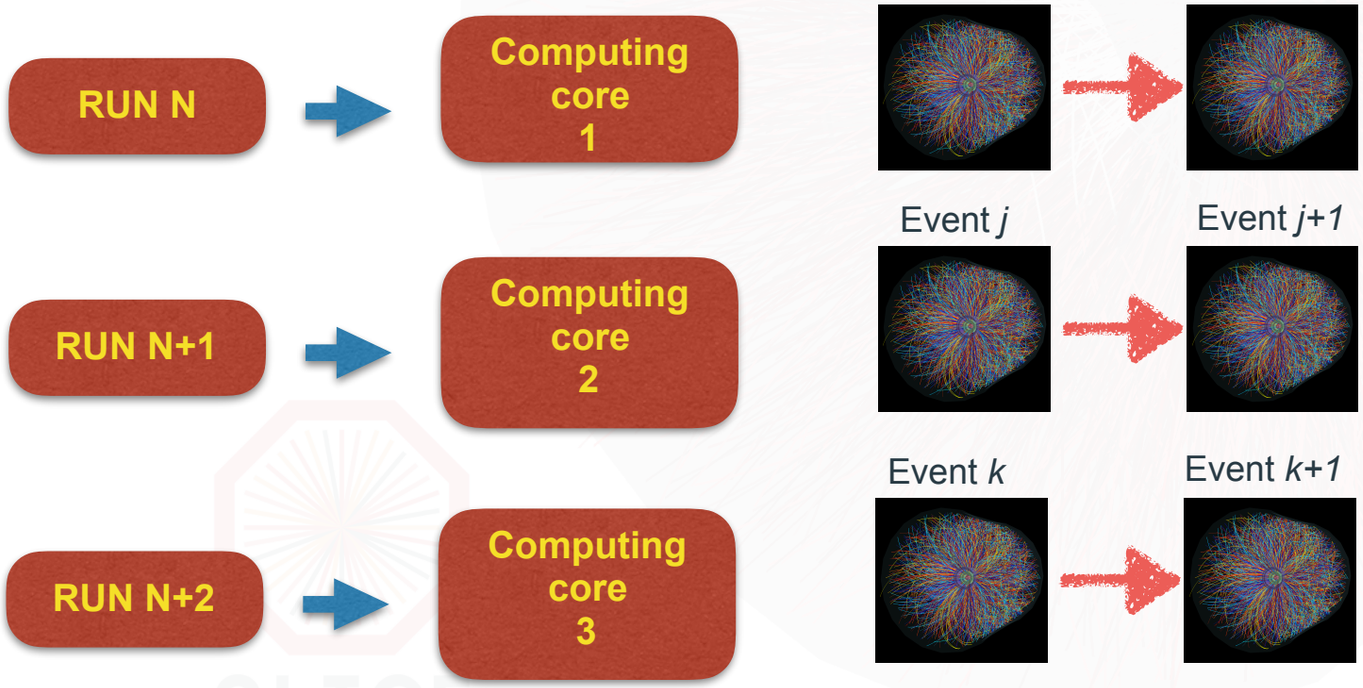
Offline computing in ALICE

- When a Pb-Pb (pp or p-Pb) collision meets a predefined trigger condition, the corresponding data are collected: these data form an **event**.
- Ideally, an event corresponds to a single Pb-Pb (pp…) collision, even though pile-up of several collisions may occur, depending on the interaction rate.
- The data collected in a continuous data taking period (~ few hours) within one LHC fill, constitute a **run**.
- The events in a run must be processed independently; they share only the same data taking conditions.
- Different runs are processed independently on different computing cores (if no hyper-threading):
  - » at a given moment a computing farm with N cores processes N events in parallel;
  - » embarrassingly parallel computing.
- Presently, no further parallelization within a single event is attempted.



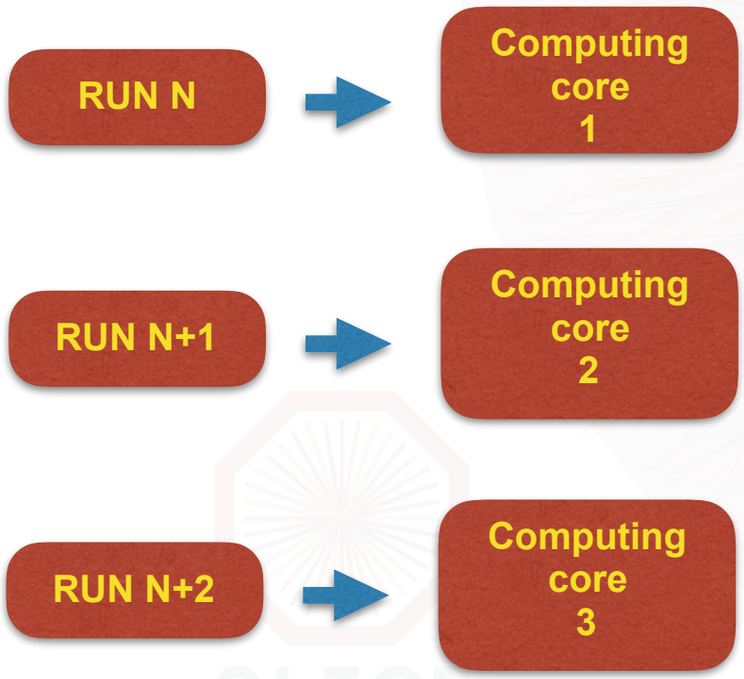
Offline computing in ALICE

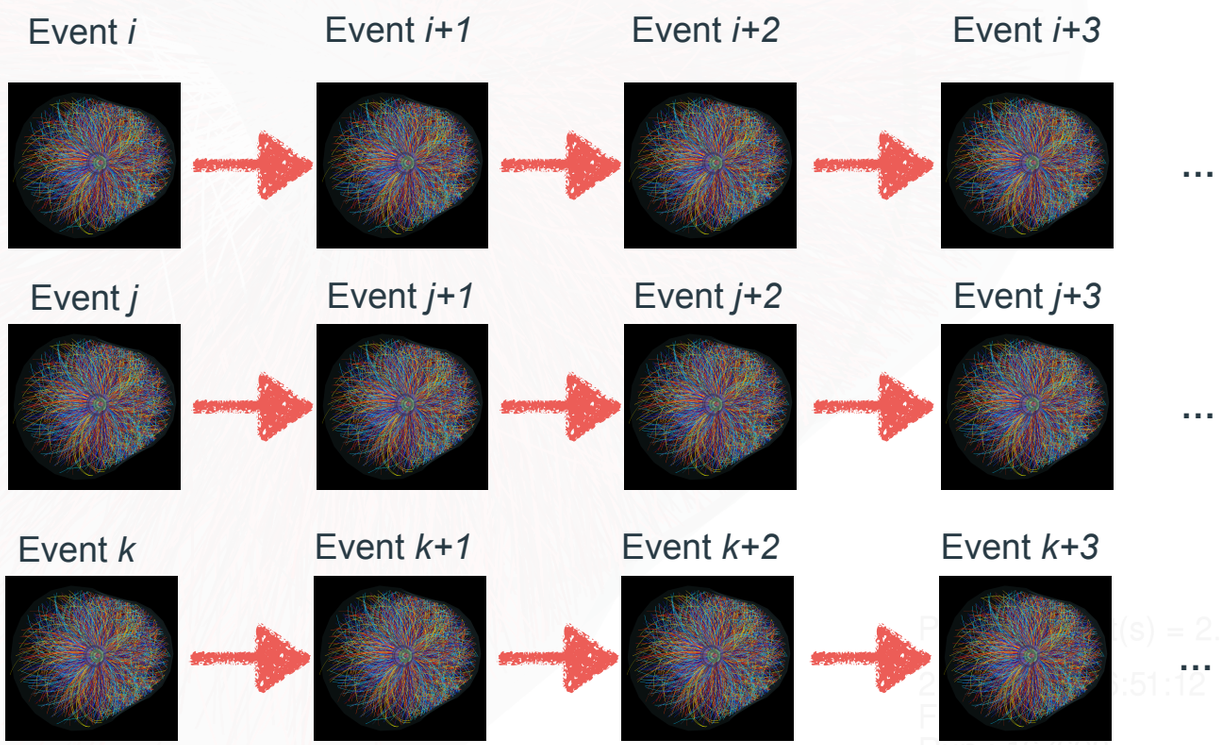
- When a Pb-Pb (pp or p-Pb) collision meets a predefined trigger condition, the corresponding data are collected: these data form an **event**.
- Ideally, an event corresponds to a single Pb-Pb (pp…) collision, even though pile-up of several collisions may occur, depending on the interaction rate.
- The data collected in a continuous data taking period (~ few hours) within one LHC fill, constitute a **run**.
- The events in a run must be processed independently; they share only the same data taking conditions.
- Different runs are processed independently on different computing cores (if no hyper-threading):
  - » at a given moment a computing farm with N cores processes N events in parallel;
  - » embarrassingly parallel computing.
- Presently, no further parallelization within a single event is attempted.



Offline computing in ALICE

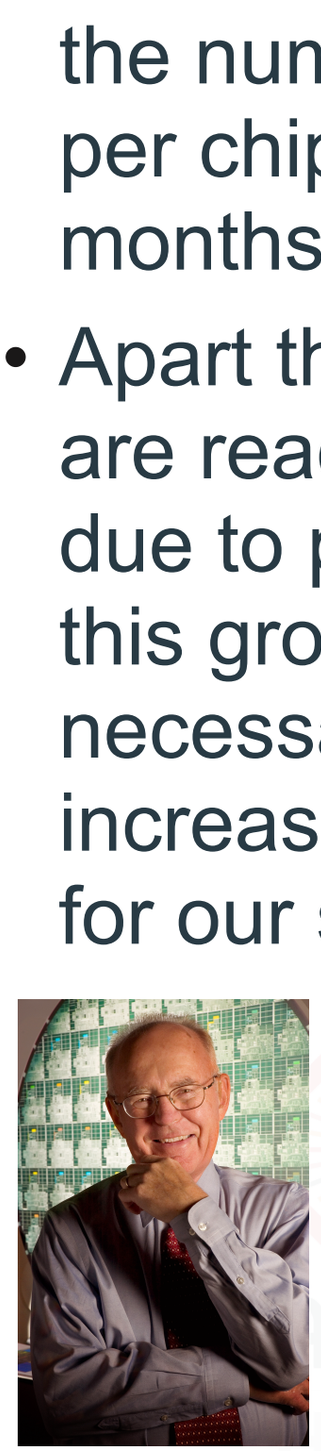
- When a Pb-Pb (pp or p-Pb) collision meets a predefined trigger condition, the corresponding data are collected: these data form an **event**.
- Ideally, an event corresponds to a single Pb-Pb (pp…) collision, even though pile-up of several collisions may occur, depending on the interaction rate.
- The data collected in a continuous data taking period (~ few hours) within one LHC fill, constitute a **run**.
- The events in a run must be processed independently; they share only the same data taking conditions.
- Different runs are processed independently on different computing cores (if no hyper-threading):
  - » at a given moment a computing farm with N cores processes N events in parallel;
  - » embarrassingly parallel computing.
- Presently, no further parallelization within a single event is attempted.

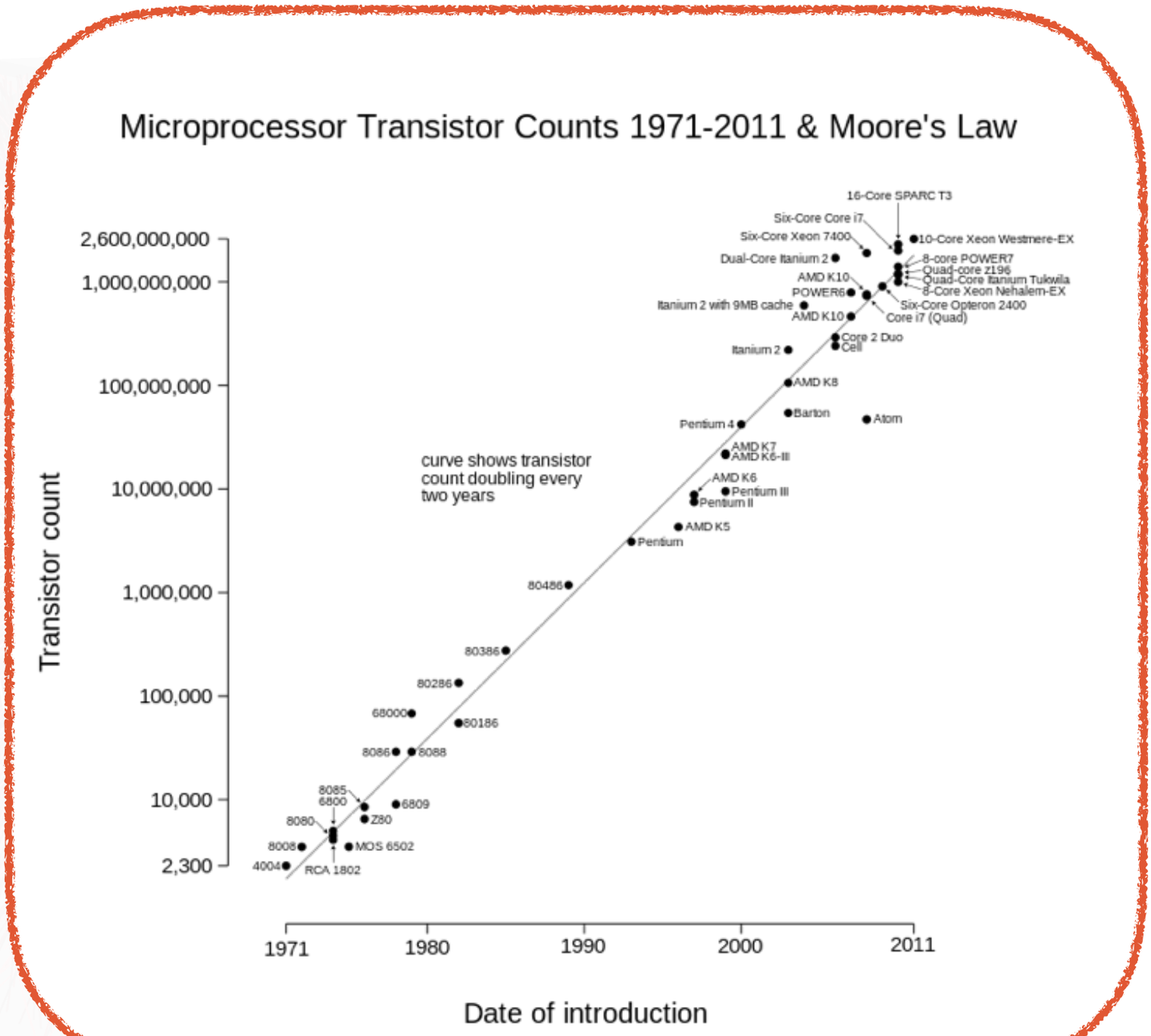




Moore's law

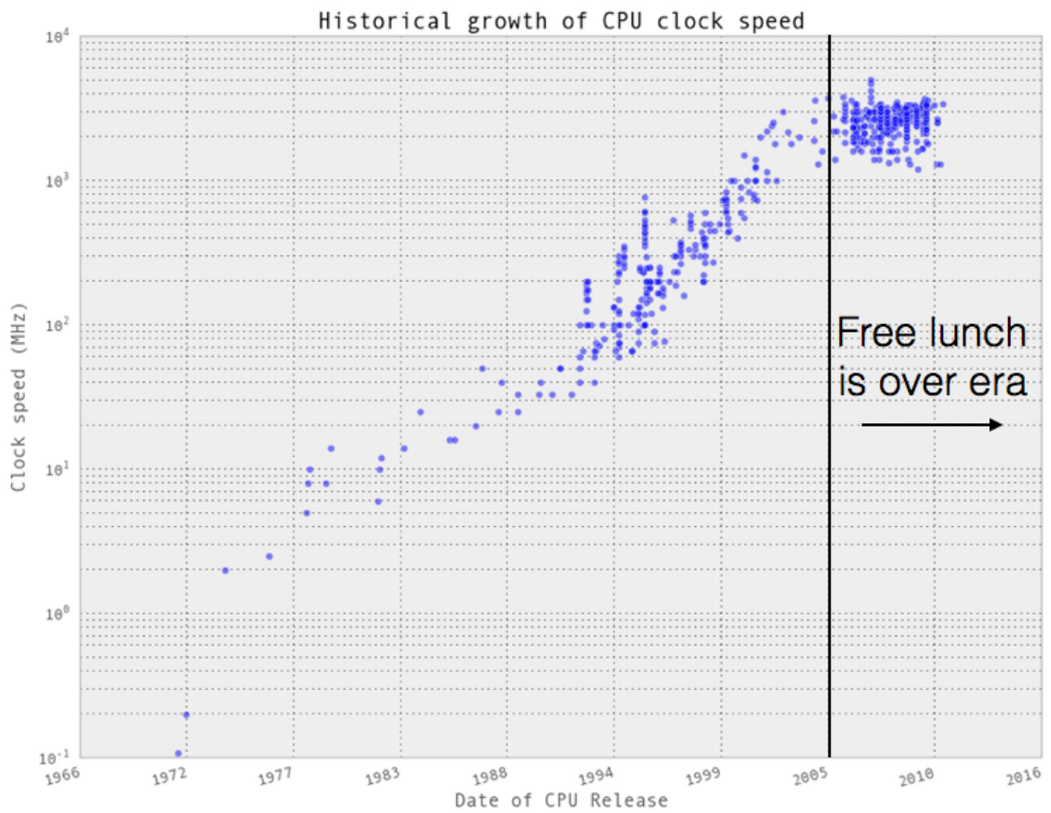
- According to Moore's law the number of transistors per chip doubles in 24 months
- Apart the fact that we are reaching a saturation due to physical limits, this growth does not necessarily imply an increase of performance for our software





Why HPC in High energy Physics?

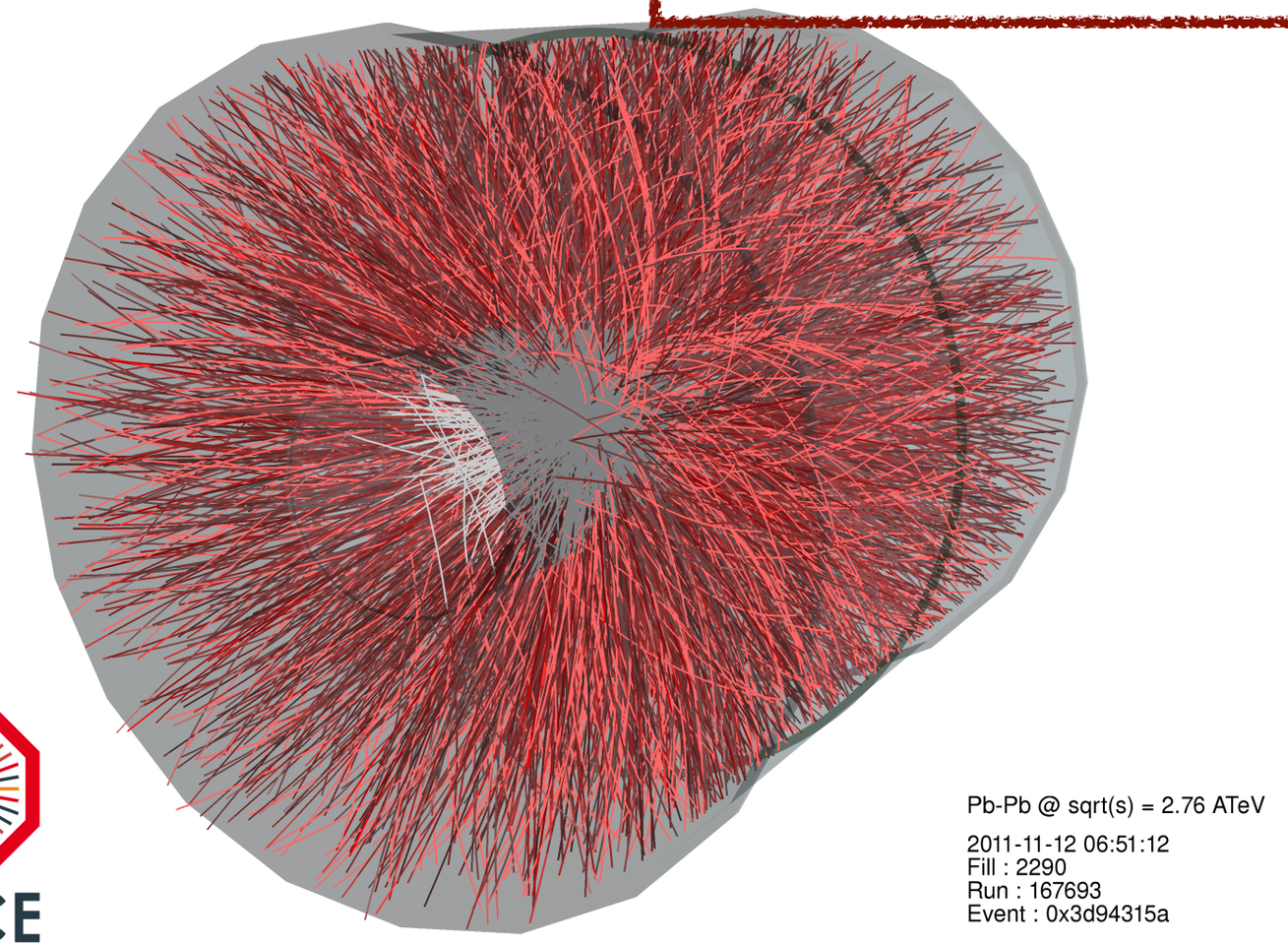
- Our software profited of an exponential growth of the CPU clock frequency.
- This growth ended about 10 years ago.
- CPU performance is still growing since the number of cores/CPU is growing.
- Due to the intrinsic parallelism of our data, we exploited this core growth by increasing the number of jobs/CPU.
- Considerations related to real time needs (online processing) and to budget evaluations (in terms of number of worker nodes and power/cooling costs) are pushing our community towards HPC solutions
- We are just at the beginning!



ALICE - Data reconstruction challenges

- The High Level Trigger (HLT) system is capable to carry out the event reconstruction in real time.
- Designed for online event selection.
- It is currently also used to perform the local reconstruction (clustering) of the TPC -> TPC clusters are stored instead of raw data to reduce the event size
- The track reconstruction in a high multiplicity environment is a challenging issue

Charged particle multiplicity: dN/dy~1400
Number of track in TPC up to 20000
Number of TPC clusters ~10^6

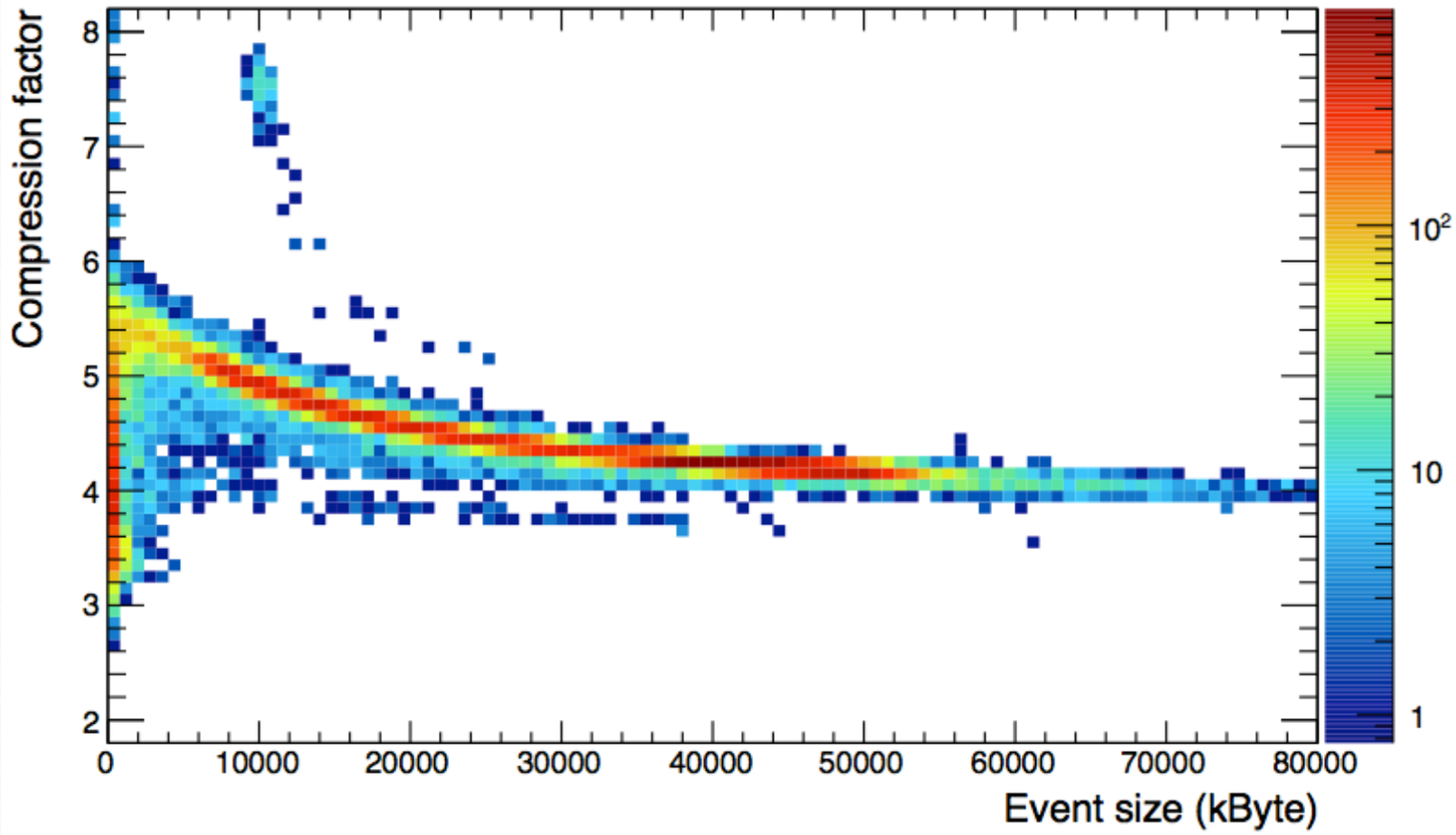


ALICE
A JOURNEY OF DISCOVERY

Reconstructed charged particle trajectories (tracks) in the ITS (white) and TPC detectors for a Pb-Pb event

ALICE - Data reconstruction challenges

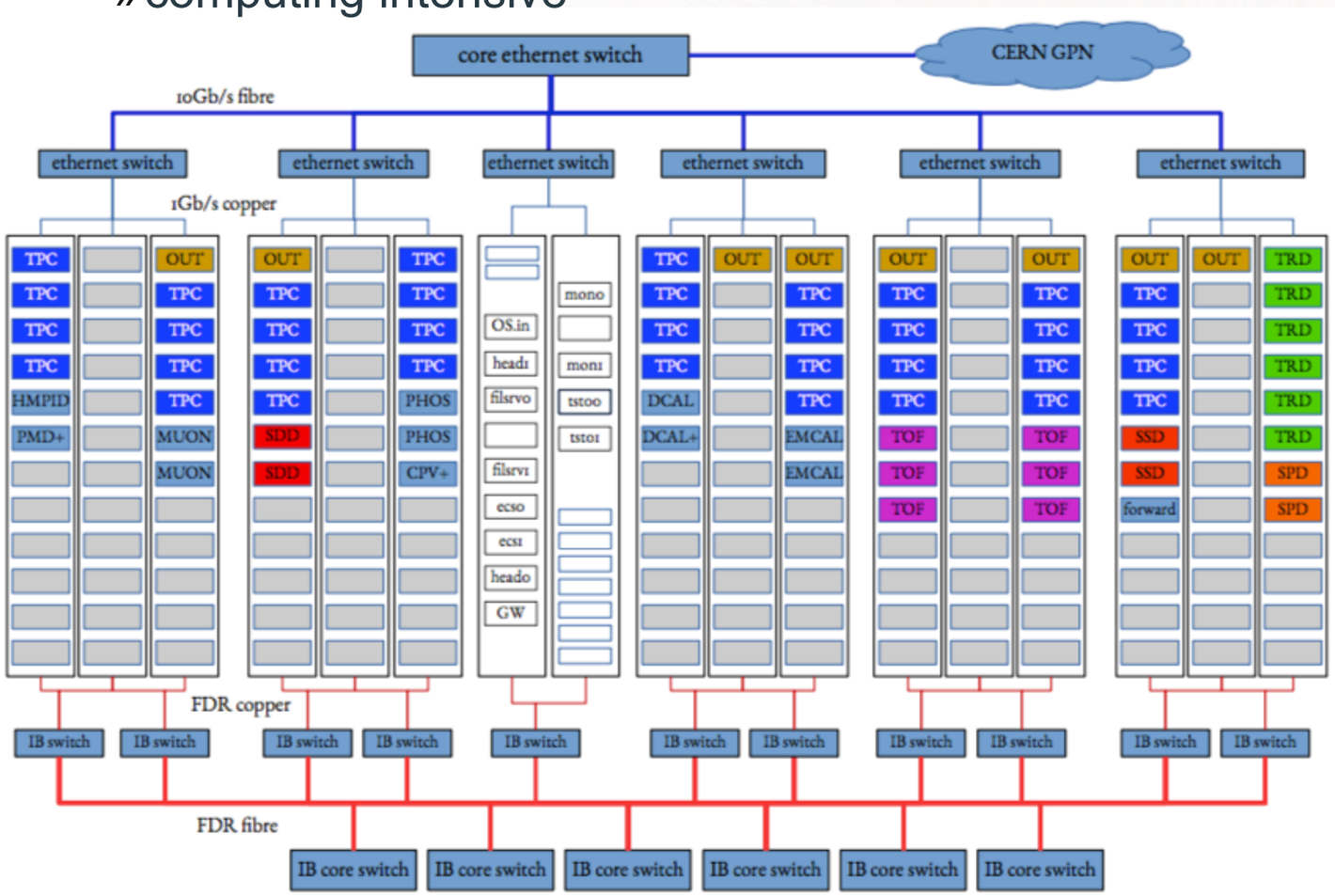
- The High Level Trigger (HLT) system is capable to carry out the event reconstruction in real time.
- Designed for online event selection.
- It is currently also used to perform the local reconstruction (clustering) of the TPC -> TPC clusters are stored instead of raw data to reduce the event size
- The track reconstruction in a high multiplicity environment is a challenging issue

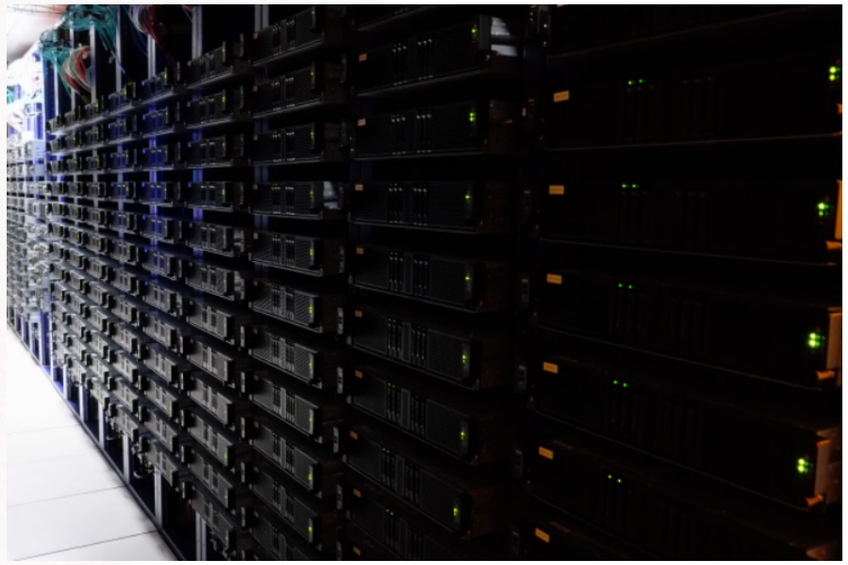


TPC cluster compression factor as a function of the event size (Pb-Pb data - 2011)

Online computing in ALICE: the HLT

- The HLT computing farm has to process the events in real time.
- Event rate 300 Hz Pb-Pb (2 kHz pp)
- Data rate ~30 GB/s
- The cluster finder algorithm for the TPC is coded on FPGAs connected to sub-elements of the detector, i.e. before the event building (in blue)
- Up to 20000 tracks in the TPC/event
- Up to 159 clusters/track
- Tracking: combining clusters to reconstruct particle trajectories
  - » high combinatorics
  - » computing intensive

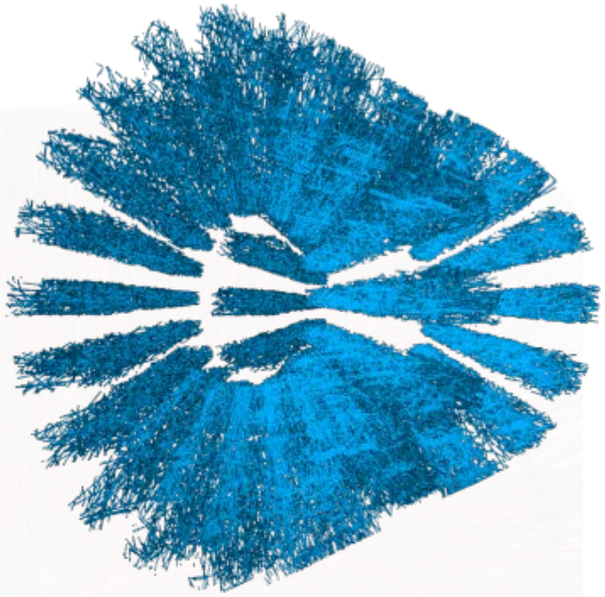


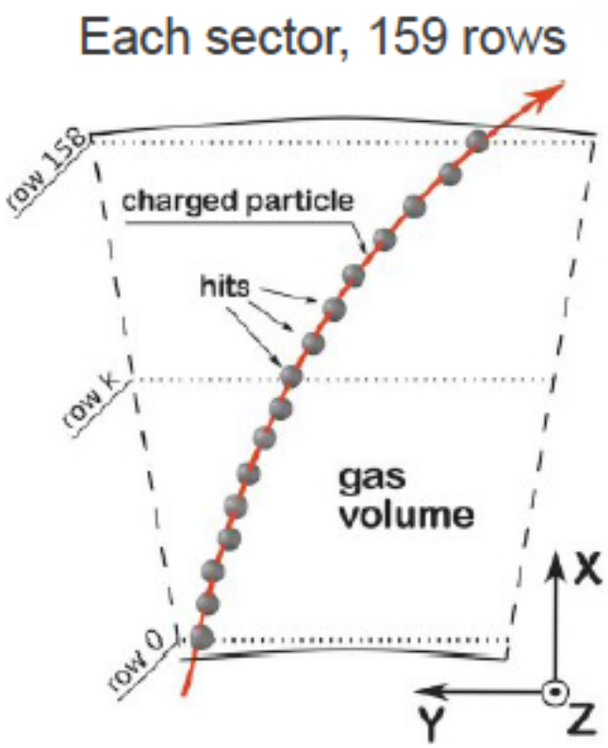


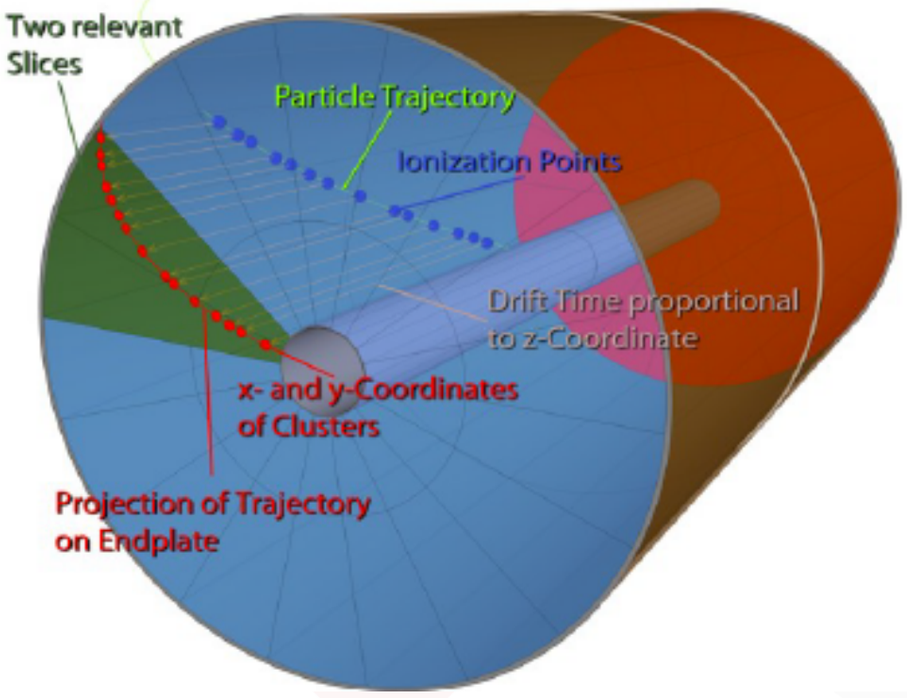
Layout of the HLT farm. Worker nodes are in gray. In colour, the nodes equipped with Readout receiver cards, hosting a Xilinx Virtex FPGA

HLT: TPC tracking

- TPC tracking with the HLT is the most relevant example of HPC in ALICE
- It is the basis for the reconstruction code in the forthcoming LHC Run 3, with an upgraded ALICE apparatus at a much higher interaction rate
- The TPC volume is split in 36 sectors: tracking is done in each sector individually.







- The radial and azimuthal coordinates of the clusters are measured by charge collection in 159 rows.
- Inner radius: 85 cm
- Outer radius: 250 cm
- The coordinate along the beam axis is measured via the drift time

HLT: TPC tracking

Neighbour finder

- For each hit at row k, the best pair of neighbouring hits from row k+1 and k-1 is found (best=straight line)

Evolution

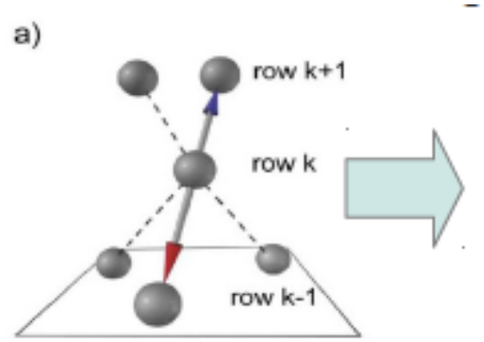
- Reciprocal links are determined and saved

Tracklet reconstruction

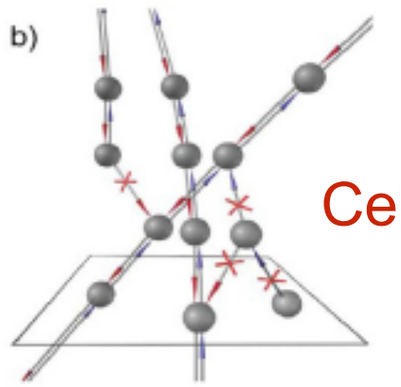
- Tracklets are created following hit-to-hit links; Kalman filter to fit geometrical trajectories

Tracklet selection

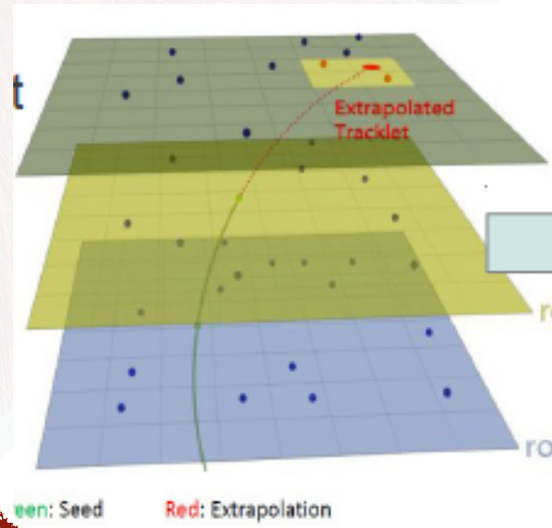
- In case of tracks with overlapping parts, the longest is kept



Every hit in parallel



Cellular automaton



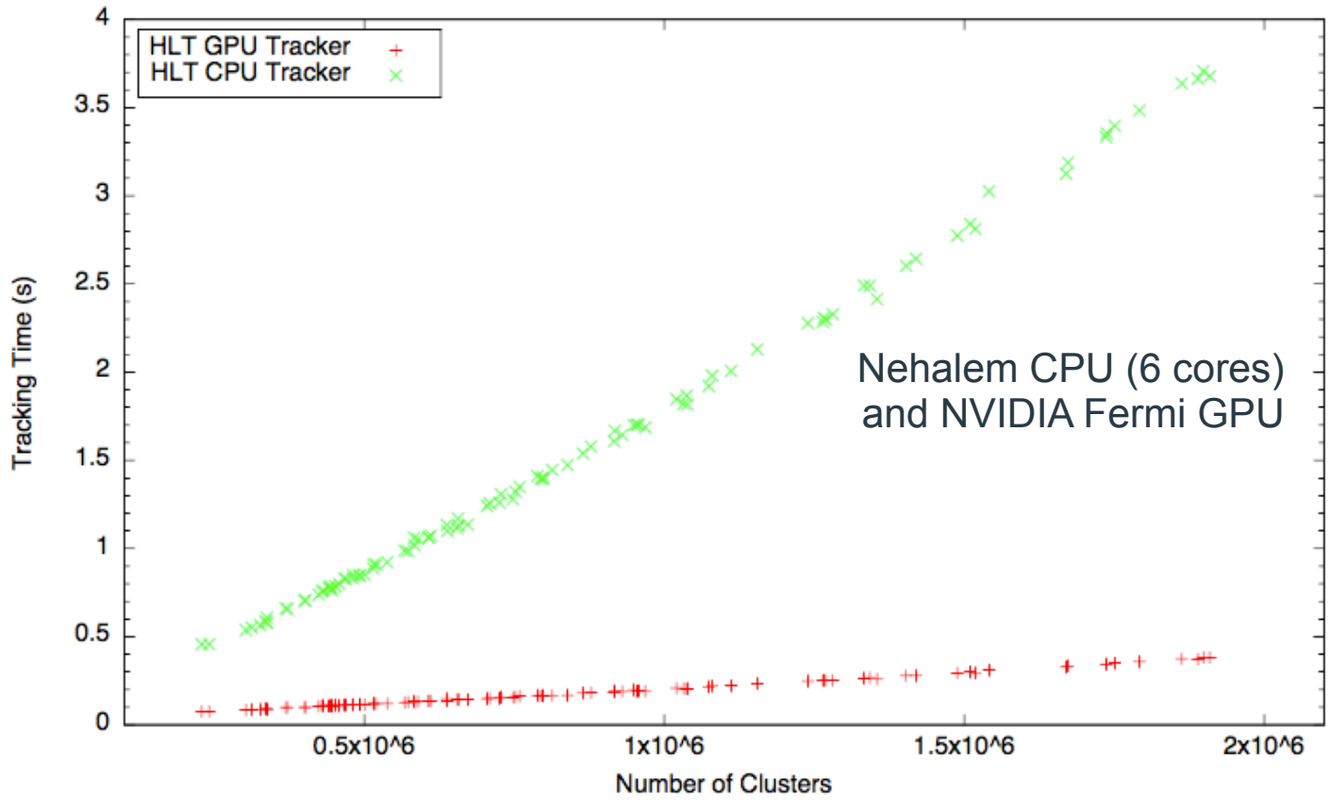
Track following

(50% of the total processing time)

ALICE

HLT: TPC tracking

- GPU tracker is about 10 times faster w.r.t. the CPU version
- Track merging and fitting is done on CPU (data transfer time cancels the GPU speed gain)
- The whole tracking is done with 1 GPU +3 CPU cores
- Performance of 1GPU+3 CPU cores ~ 27 CPU cores



Task	How	Locality	Description	Time	Where
Seeding	Cellular Automaton	Local	Find track candidates (3-10 clusters)	~30%	GPU or CPU
Track following	Kalman filter	Sector	Fit parameters to candidate, find full track segment in one sector via track following with simplified Kalman filter (e.g. constant B-field, y and x uncorrelated)	~60%	
Track merging	Combinatorics	Global	merge track segments within a sector and between sectors	~2%	CPU (GPU version exists - not used)
Track fit	Kalman filter	Globl	Full track fit with full Kalman filter (polynomial approximation of B-field)	~8%	

Code management

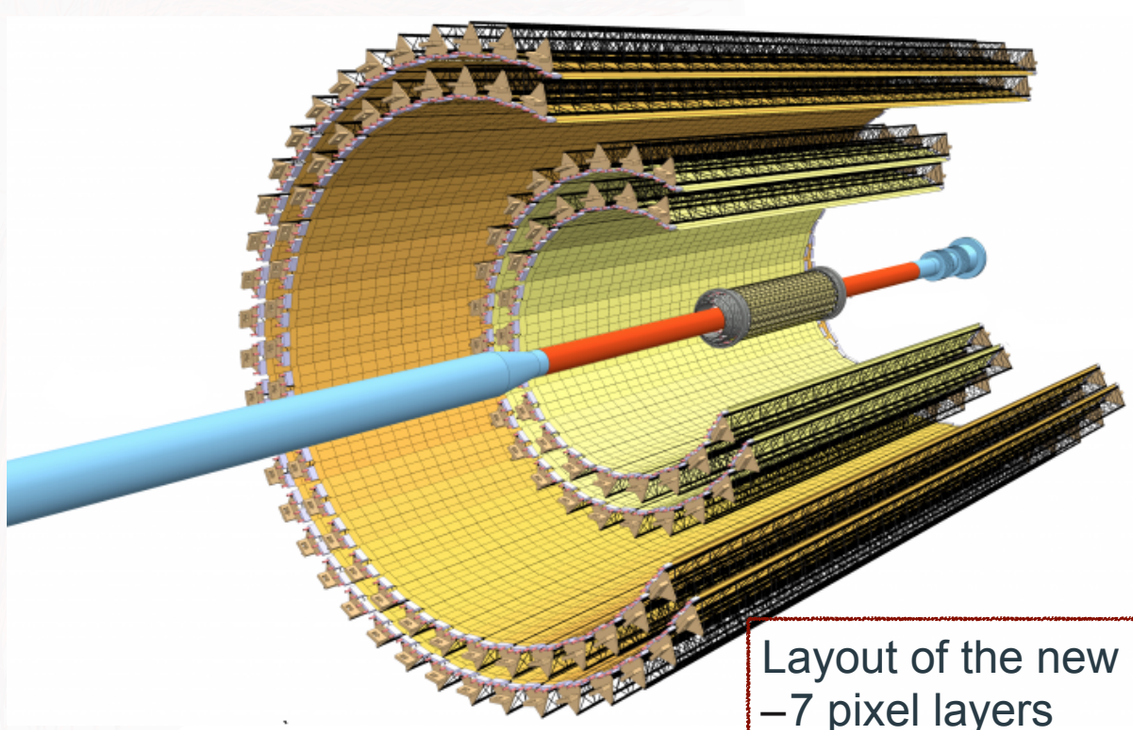
- The HLT farm is heterogeneous: CPU and GPU accelerators are available
- The GPU code may be vendor bound: e.g. Cuda in case of Nvidia cards
- A CPU version of the code must be maintained to be able to run the same reconstruction code on other facilities
  - » on WLCG infrastructure for MC data, for instance.
- For the HLT TPC tracking, CPU and GPU codes share common source files.
- Specialized wrappers for CPU, Cuda and OpenCL are provided. They include the common files.
- The fraction of common source code is **above 90%**
- The experience gained with the HLT is the basis for the new **Online+Offline (O^2)** infrastructure that will be used for LHC Run 3 (starting from 2019)

ALICE Upgrade

- After the second LHC Long Shutdown (2018-19), new conditions are expected for the subsequent Run3:
  - » Expected Pb-Pb peak interaction rate: 50 kHz (now it is 8 kHz)
- Presently ALICE readout rate is limited to ~1 kHz
- Goal for Run3:
  - » no reliable triggering strategies for several physics channels —> increase the readout rate to **50 kHz**
  - » improve pointing resolution both in the barrel (**new ITS**) and in the Muon Arm (new Muon Forward Tracker)

The ALICE upgrade requires major improvements for the TPC and other detectors in order to increase the readout rate

Capability of reducing online the data volume delivered by the detectors, since the expected integrated luminosity is > 10 nb^{-1} for Pb-Pb (x100 w.r.t. Run 1)

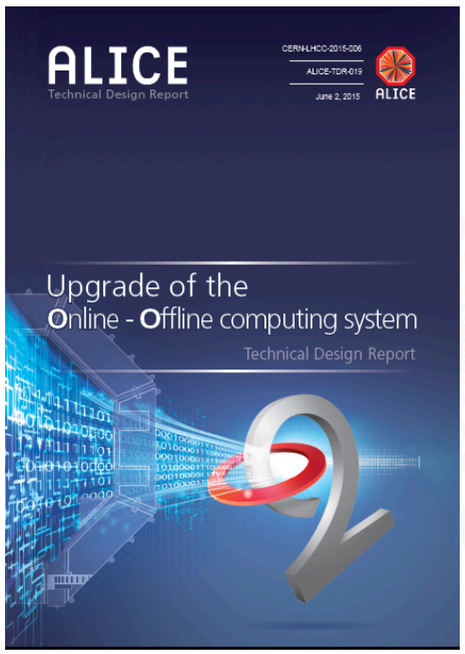


Layout of the new ITS:
– 7 pixel layers
– 10 m^2 of silicon
– 12.5 Gpixel

ALICE Upgrade: the O^2 system

- The expected data rate for Pb-Pb collisions at 50 kHz is ~1.1 TB/s
- The TPC alone accounts for 1 TB/s
- The O^2 project aims to integrate in a single infrastructure the present DAQ, HLT and Offline (for the reconstruction part) systems

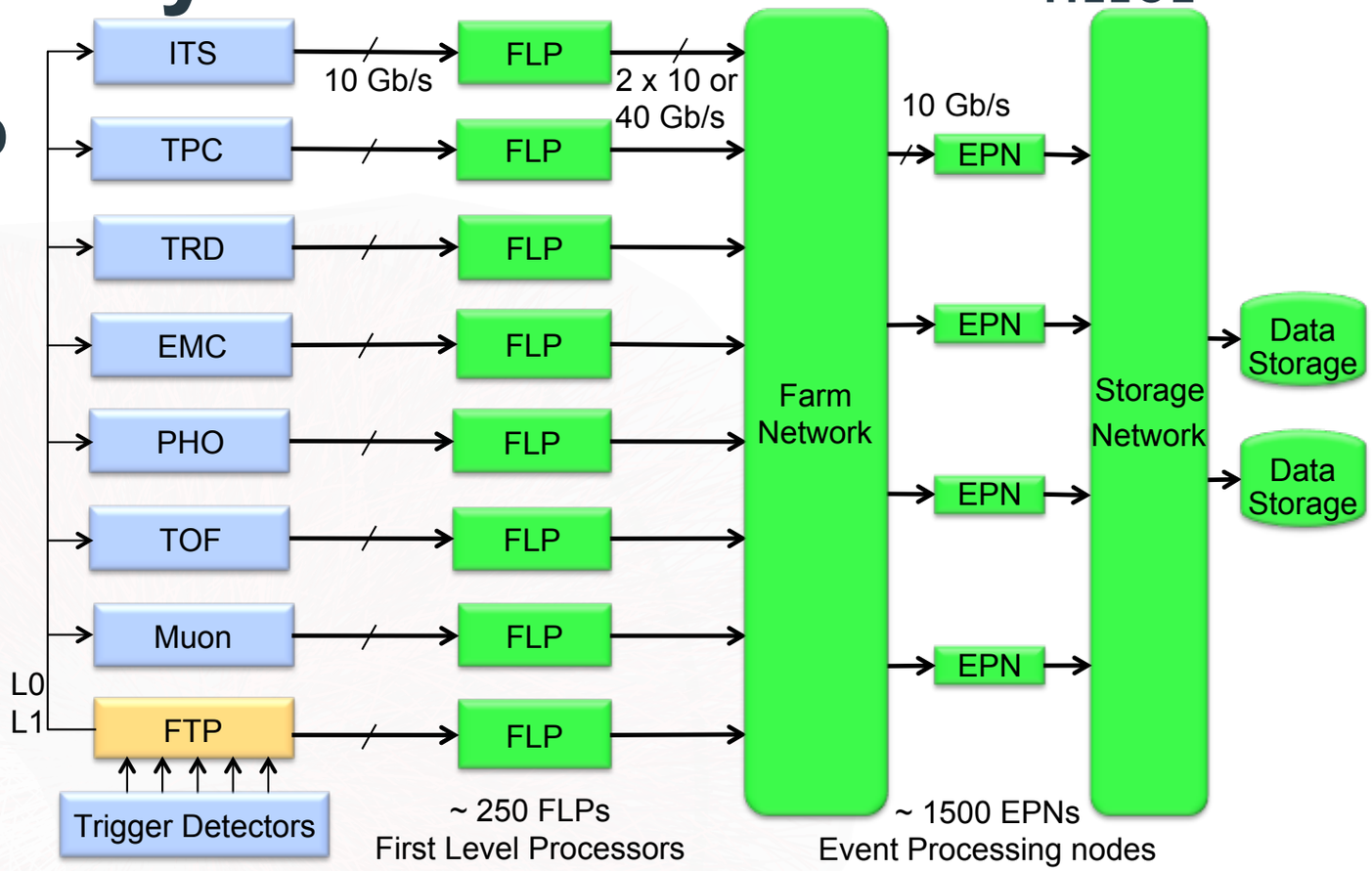
Detector	Average event size (MB)	Data rate for Pb-Pb @ 50 kHz (GB/s)
TPC	20.7	1012
ITS	0.8	40
TRD	0.5	20
MFT	0.2	10
Others	0.3	12.2
Total	**22,5**	**1094,2**

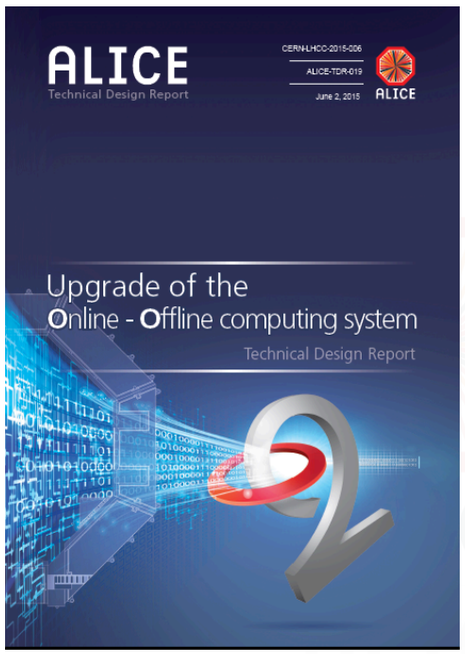


- The data volume coming from the detectors must be substantially reduced before sending the data to the mass storage.
- Online processing is the only option
- The computing strategy must rely on a heterogeneous architecture to match the interaction rate:
  - ~250 FLP worker nodes (First Level Processors) equipped with FPGA
  - ~1500 EPN worker nodes (Event Processing Nodes) equipped with GPU
  - yearly amount of data (2020, 2021): 54 PB

ALICE Upgrade: the O^2 system

- The expected data rate for Pb-Pb collisions at 50 kHz is ~1.1 TB/s
- The TPC alone accounts for 1 TB/s
- The O^2 project aims to integrate in a single infrastructure the present DAQ, HLT and Offline (for the reconstruction part) systems





- The data volume coming from the detectors must be substantially reduced before sending the data to the mass storage.
- Online processing is the only option
- The computing strategy must rely on a heterogeneous architecture to match the interaction rate:
  - »~250 FLP worker nodes (First Level Processors) equipped with FPGA
  - »~1500 EPN worker nodes (Event Processing Nodes) equipped with GPU
  - »yearly amount of data (2020, 2021): 54 PB

Online data volume reduction

- The impressive reduction factor that can be obtained for the TPC is based on:
  - » zero suppression
  - » clustering and compression
  - » removal of clusters non associated to interesting particle tracks (e.g. very low momentum electrons)
  - » data format optimization
- Largely based on the present HLT results

Still uncertainties for the ITS:

» The contribution from noisy clusters is unknown: here a pessimistic estimate of a probability of 10^{-5} per pixel has been made

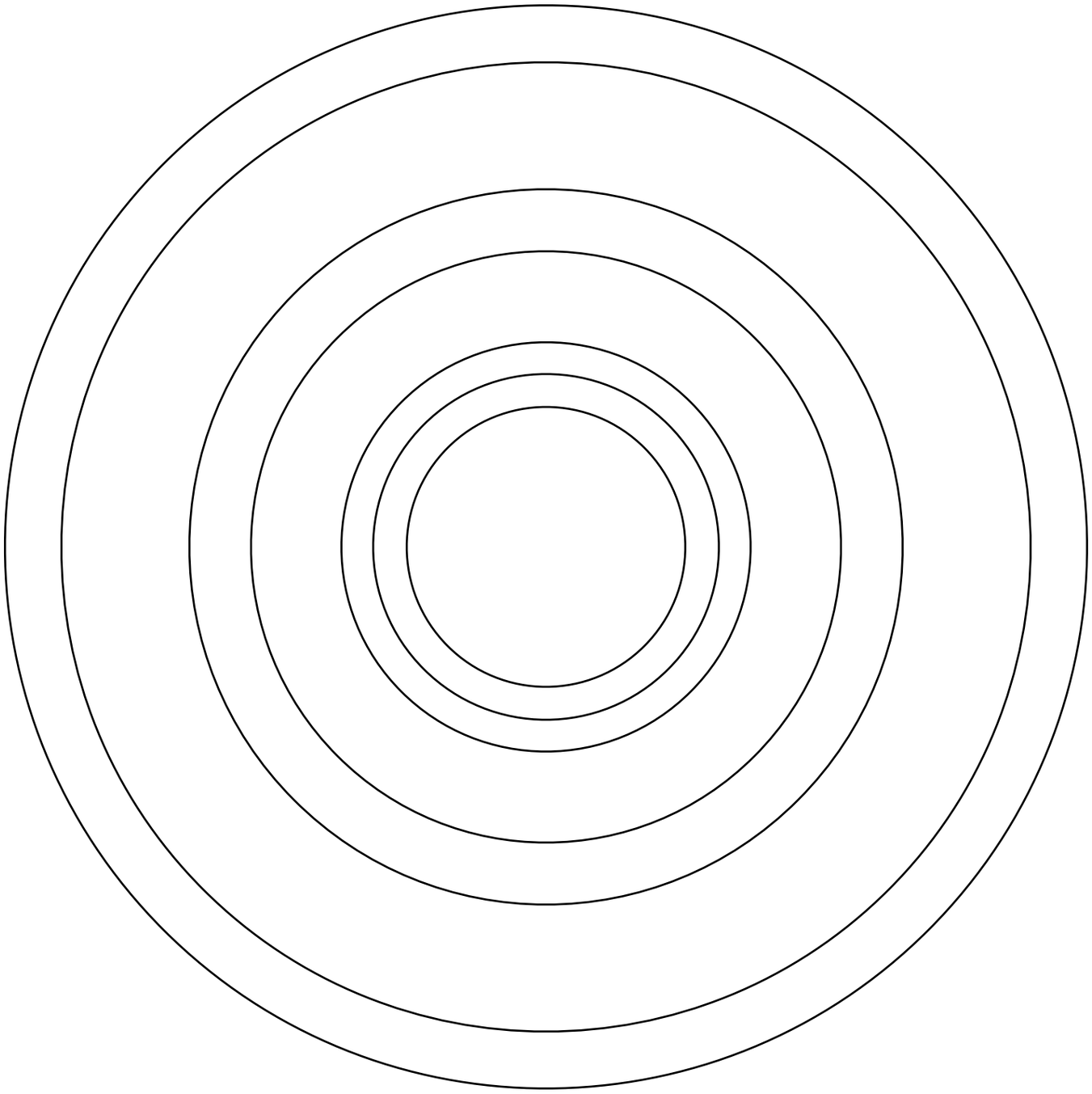
» If full synchronous reconstruction will be feasible a higher reduction factor will be achieved (noise removal)

Detector	Data rate for Pb-Pb @ 50 kHz (GB/s)	Compressed data rate (GB/s)	Data reduction
TPC	1012	50	20.2
ITS	40	26 (8)	1.5 (5)
TRD	20	3	6.7
MFT	10	5	2
Total	**1082**	**84 (66)**	**12.9 (16.4)**

ITS standalone tracking

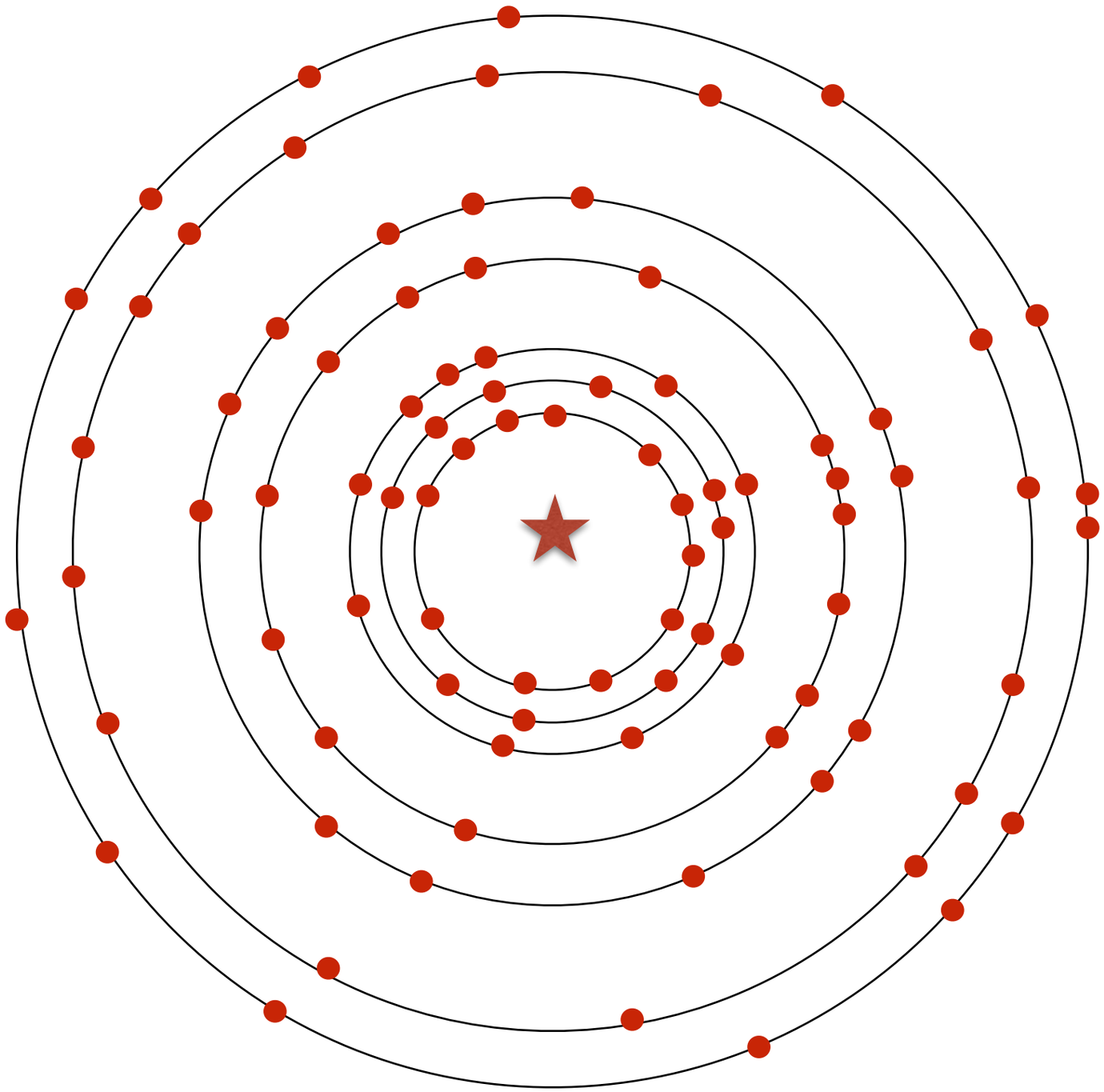
- TPC tracks can be prolonged inwards to the Inner Tracking System.
- However the ITS can be used as a standalone detector and tracks found in the ITS can be prolonged to the TPC.
- Since in Run 3 we will need to calibrate the TPC online, the ITS track seeds will be useful for the TPC calibration
- An ITS Tracker based on a Cellular Automaton has been coded and tested on CPU, within the present ALICE offline framework, AliRoot.
- The next step is to port this code to the new O^2 framework and to a heterogeneous CPU-GPU computing environment
- The goal is to have a demonstrator of TPC+ITS tracking in 2016.

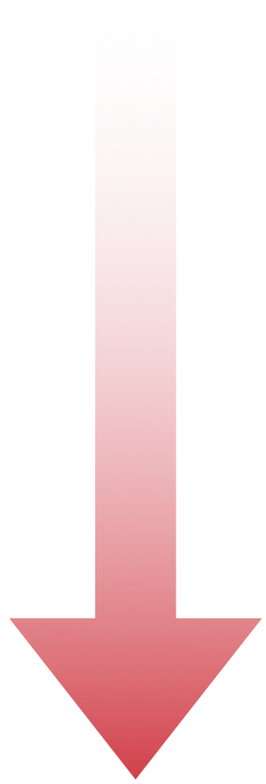
Tracking with ITS Upgrade



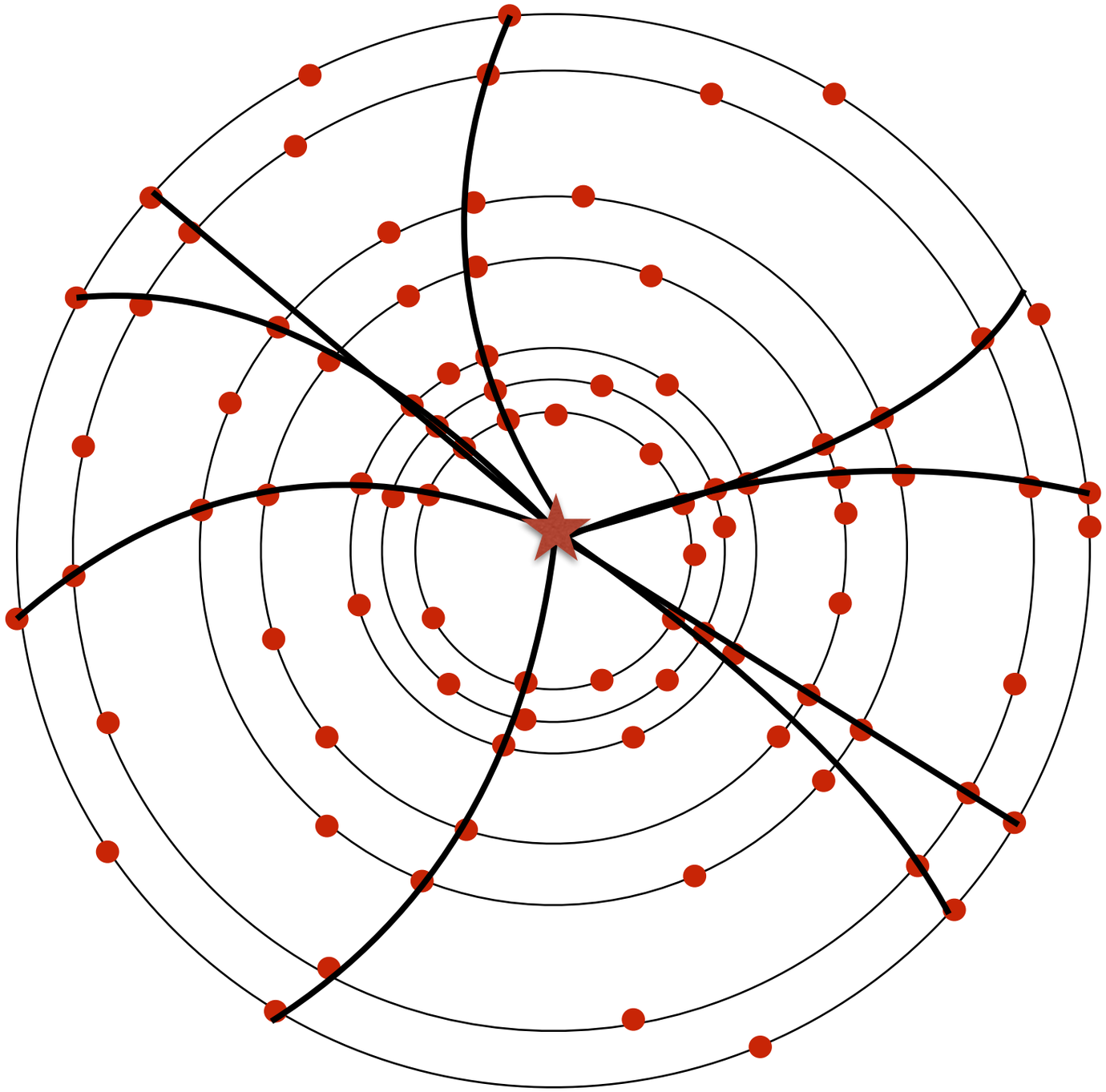
DISCLAIMER:
For easiness in explanations, in the following I will show only the transverse section of ITSU.

Tracking with ITS Upgrade



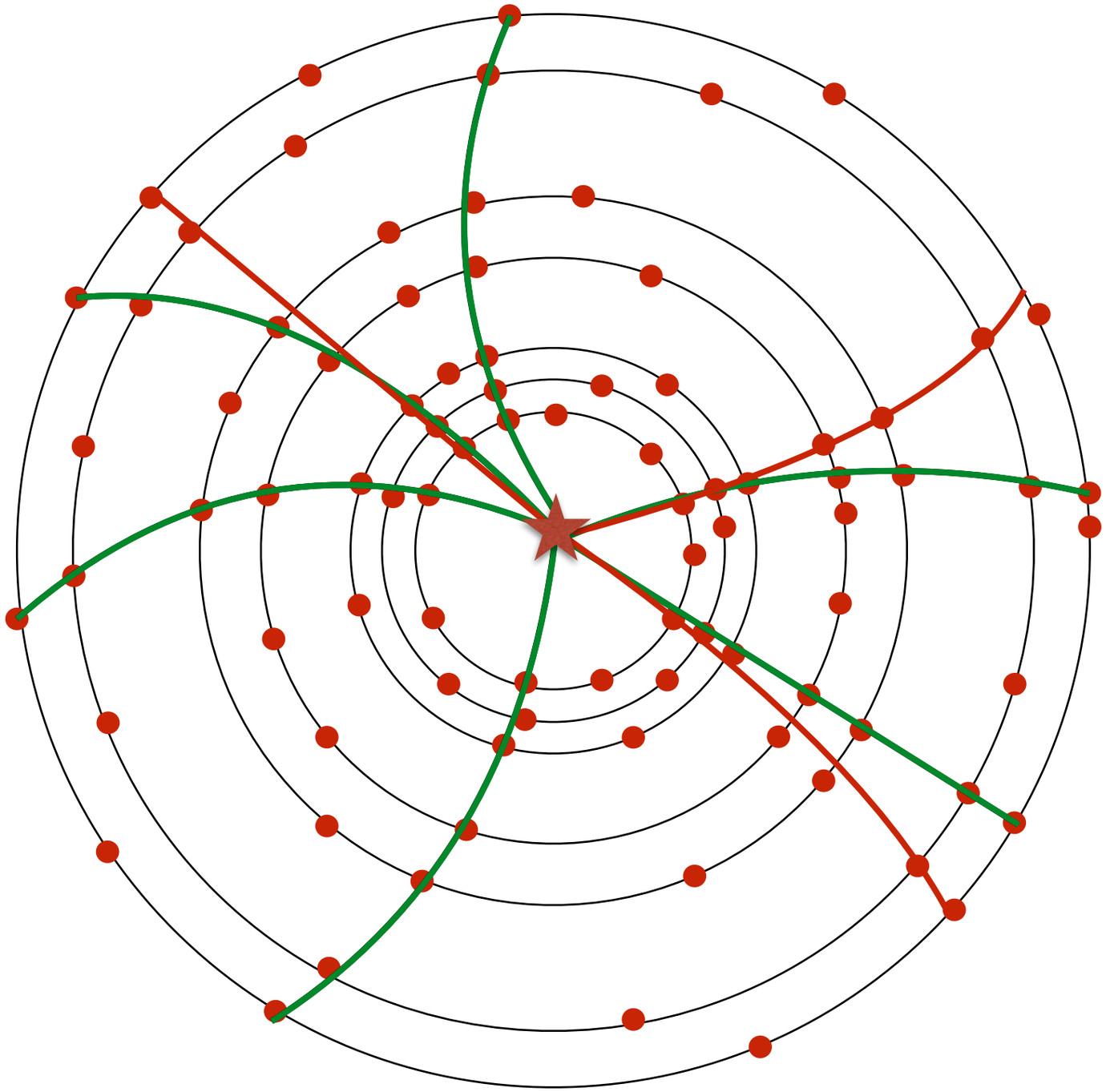


Tracking with ITS Upgrade



Using a pattern recognition method, find track candidates

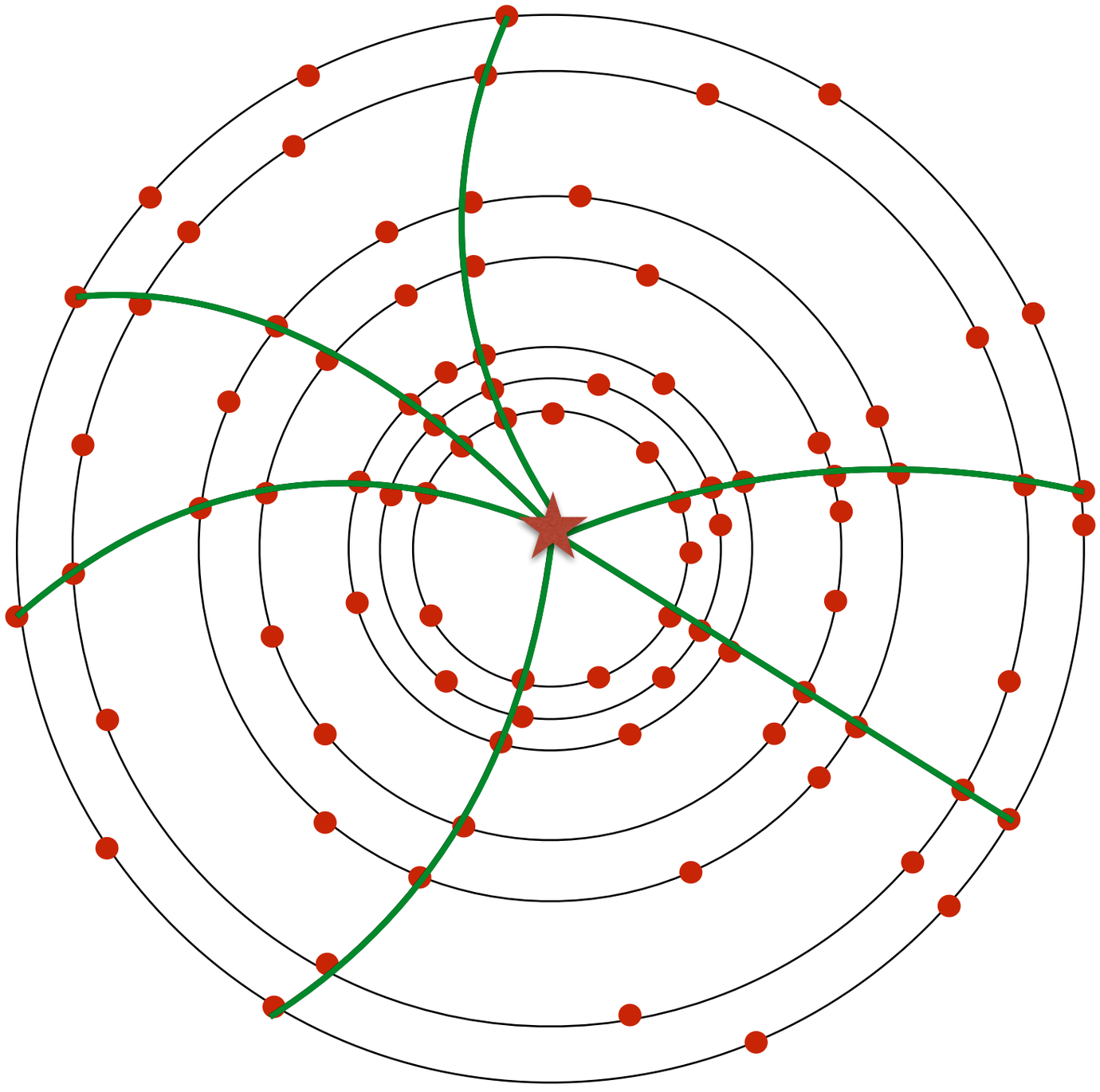
Tracking with ITS Upgrade



Using a pattern recognition method, find track candidates

Fitting of the candidates using Kalman Filter in three passes (inward, outward, inward)

Tracking with ITS Upgrade



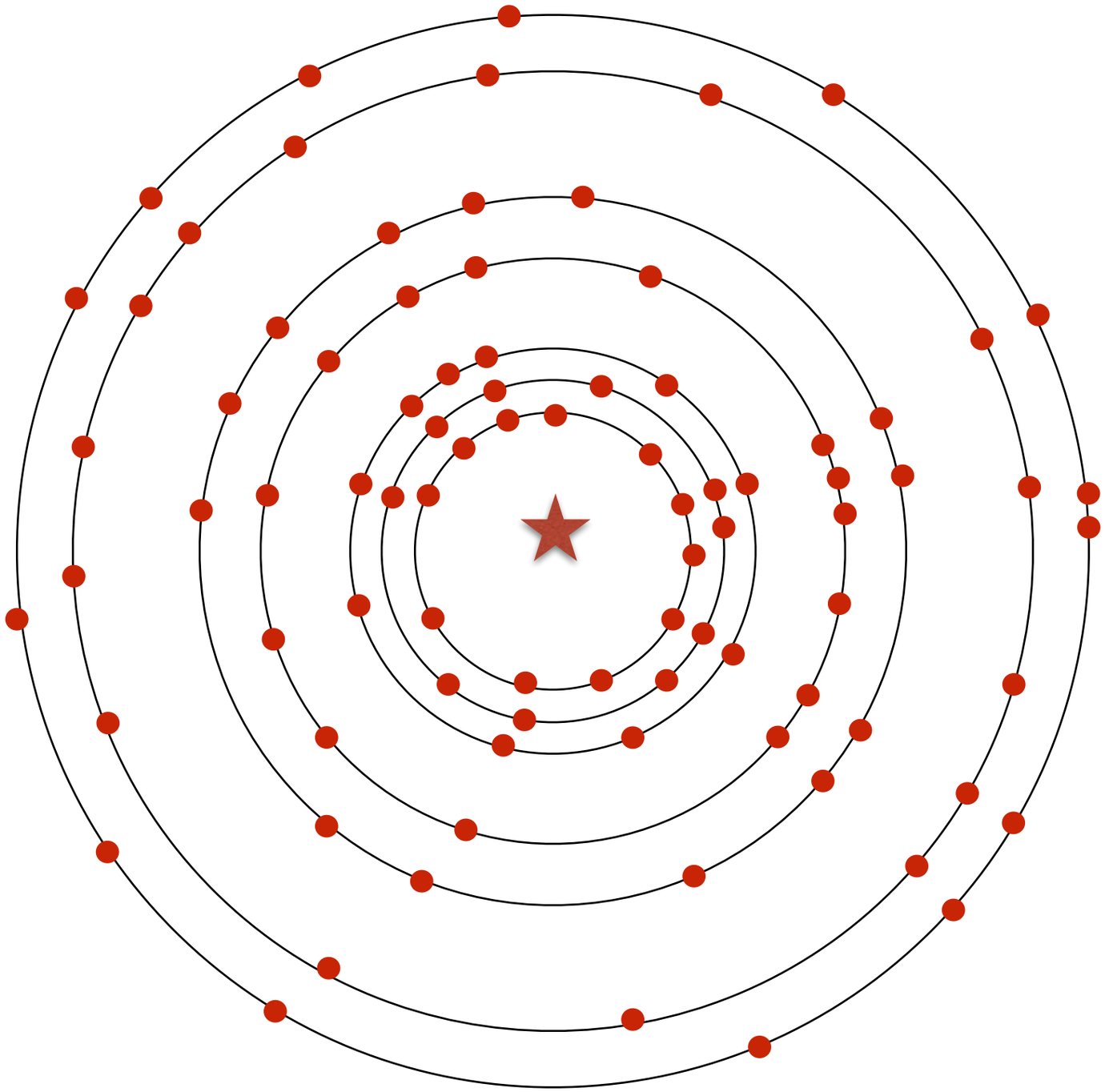
Using a pattern recognition method, find track candidates

Fitting of the candidates using Kalman Filter in three passes (inward, outward, inward)

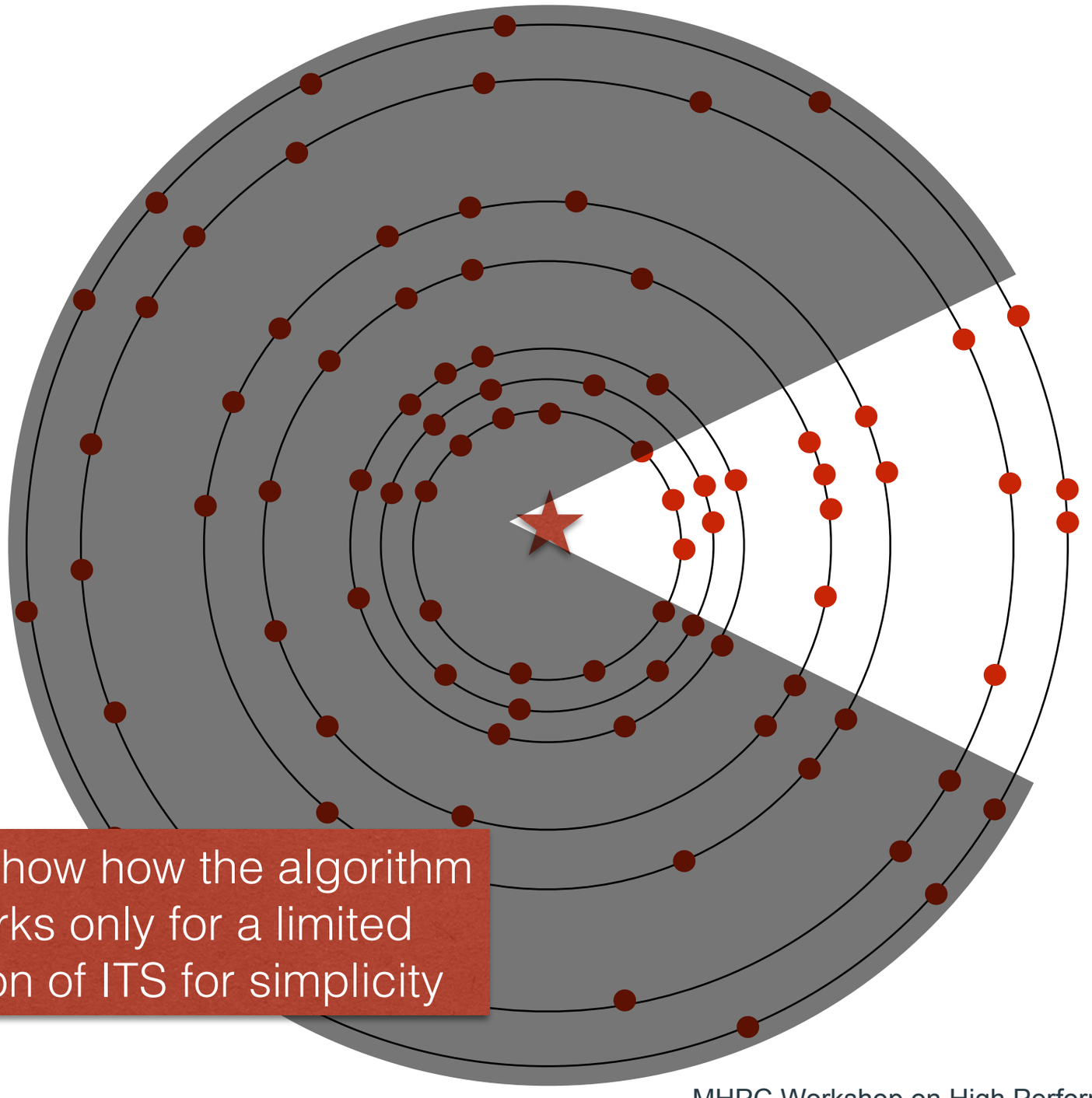
Candidates with the best χ^2 values are stored as reconstructed tracks

Currently two different approaches to the pattern recognition step are implemented for ITS Upgrade

The Cellular Automaton approach

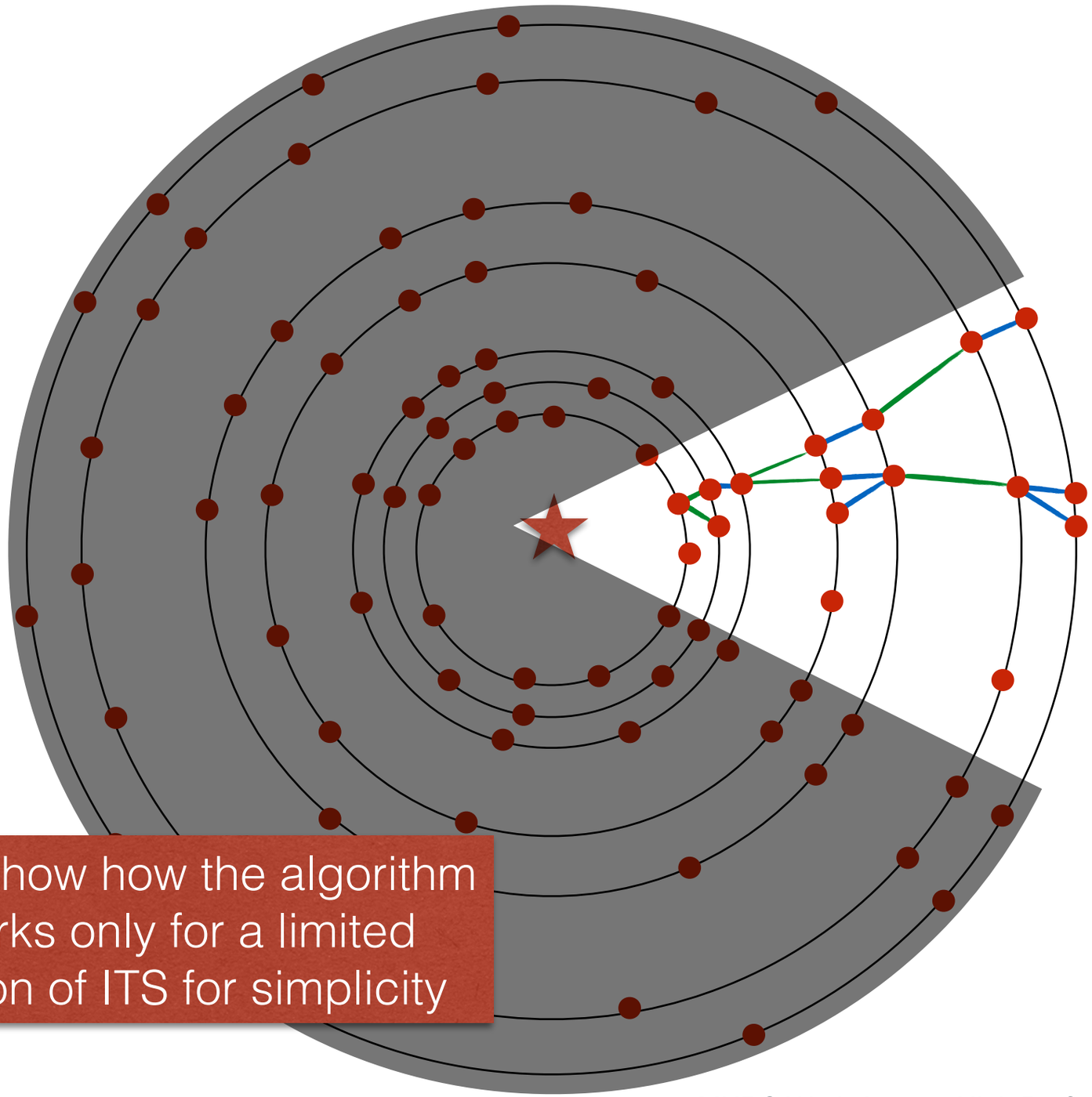


The Cellular Automaton approach



I will show how the algorithm works only for a limited region of ITS for simplicity

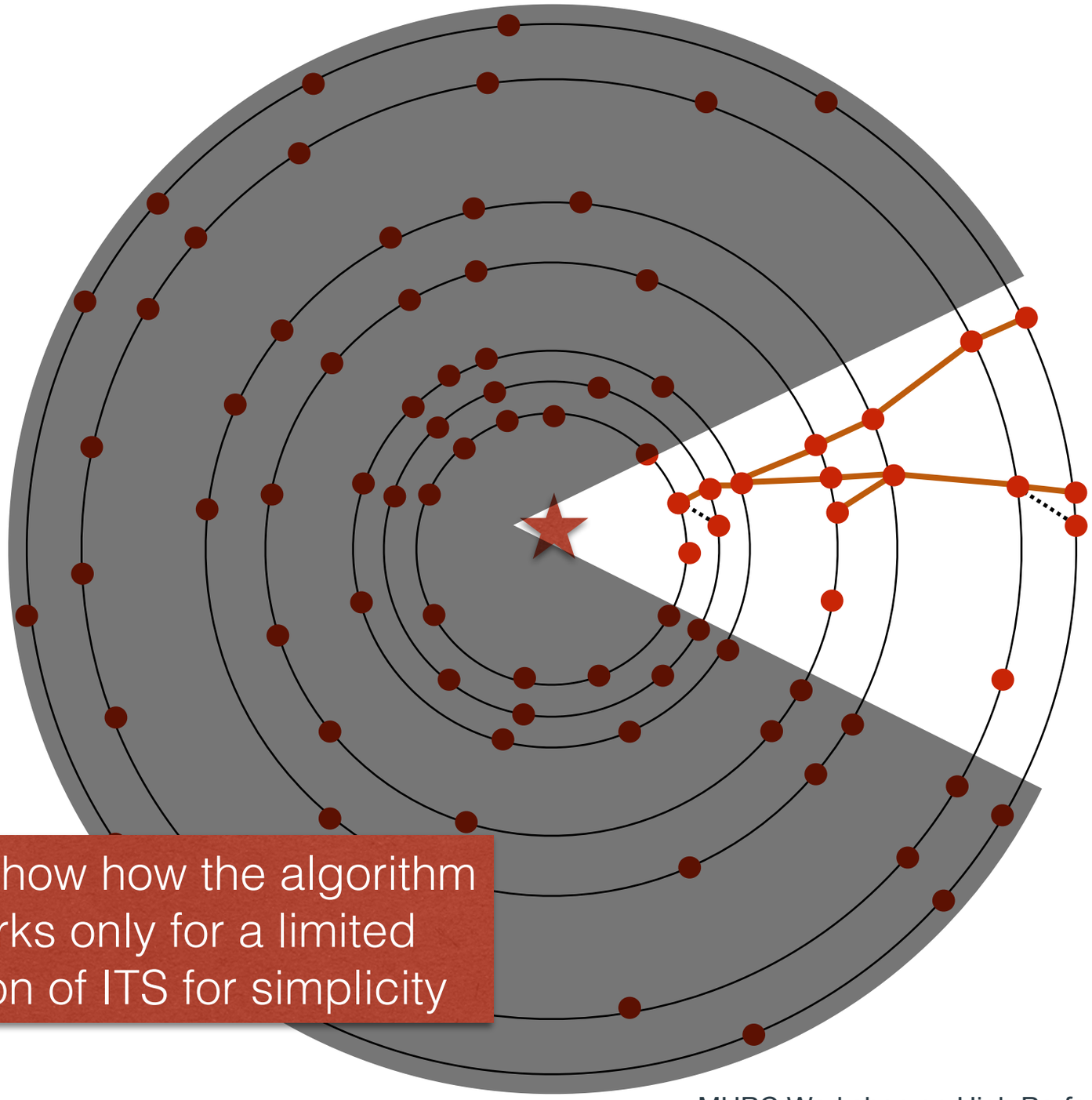
The Cellular Automaton approach



For each cluster on each layer a 2D window is opened. Then the clusters are joined with those on the next layer within the window

I will show how the algorithm works only for a limited region of ITS for simplicity

The Cellular Automaton approach

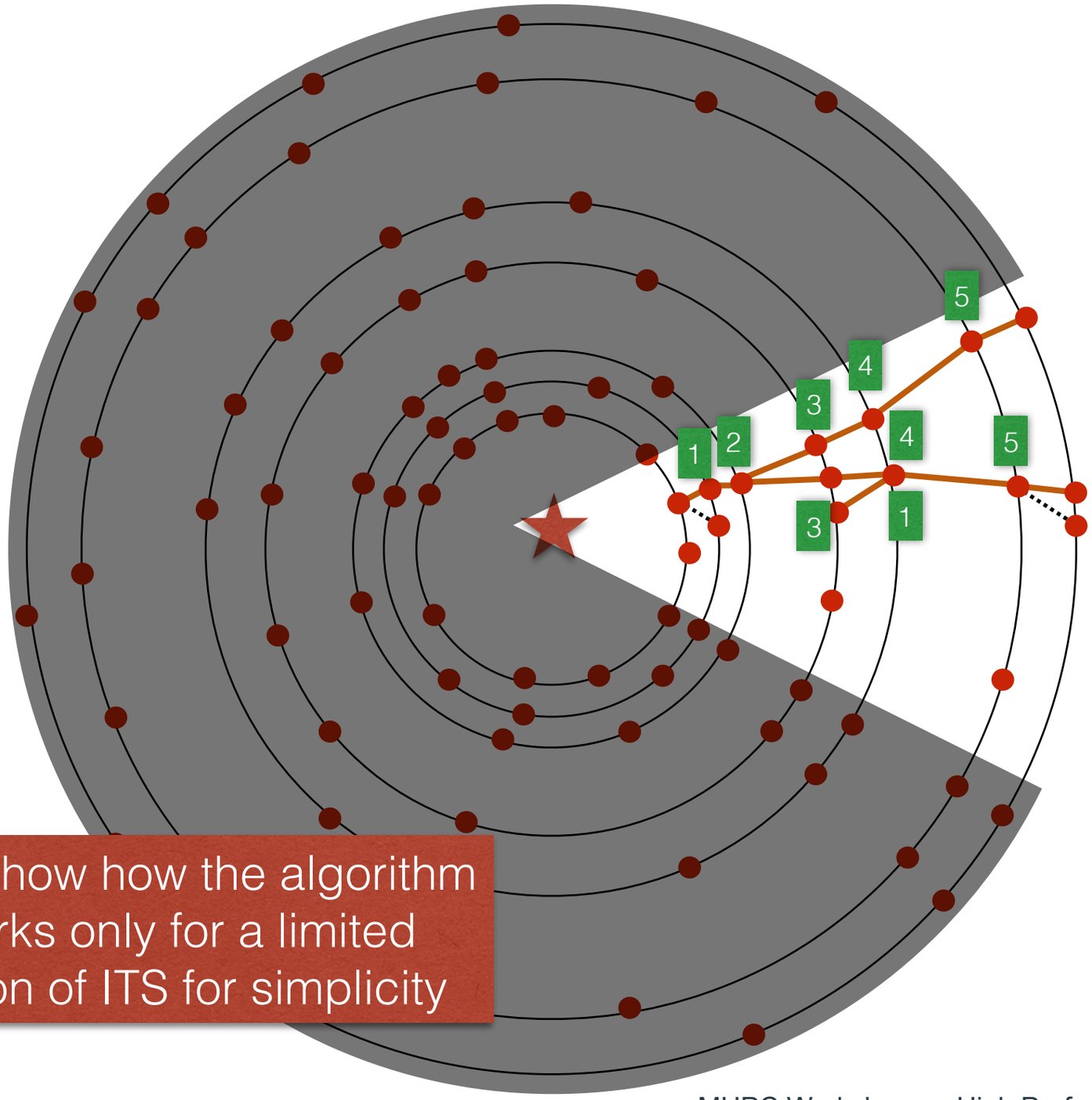


For each cluster on each layer a 2D window is opened. Then the clusters are joined with those on the next layer within the window

Subsequent doublets are combined in cells (3 points seed) and track params are computed

I will show how the algorithm works only for a limited region of ITS for simplicity

The Cellular Automaton approach



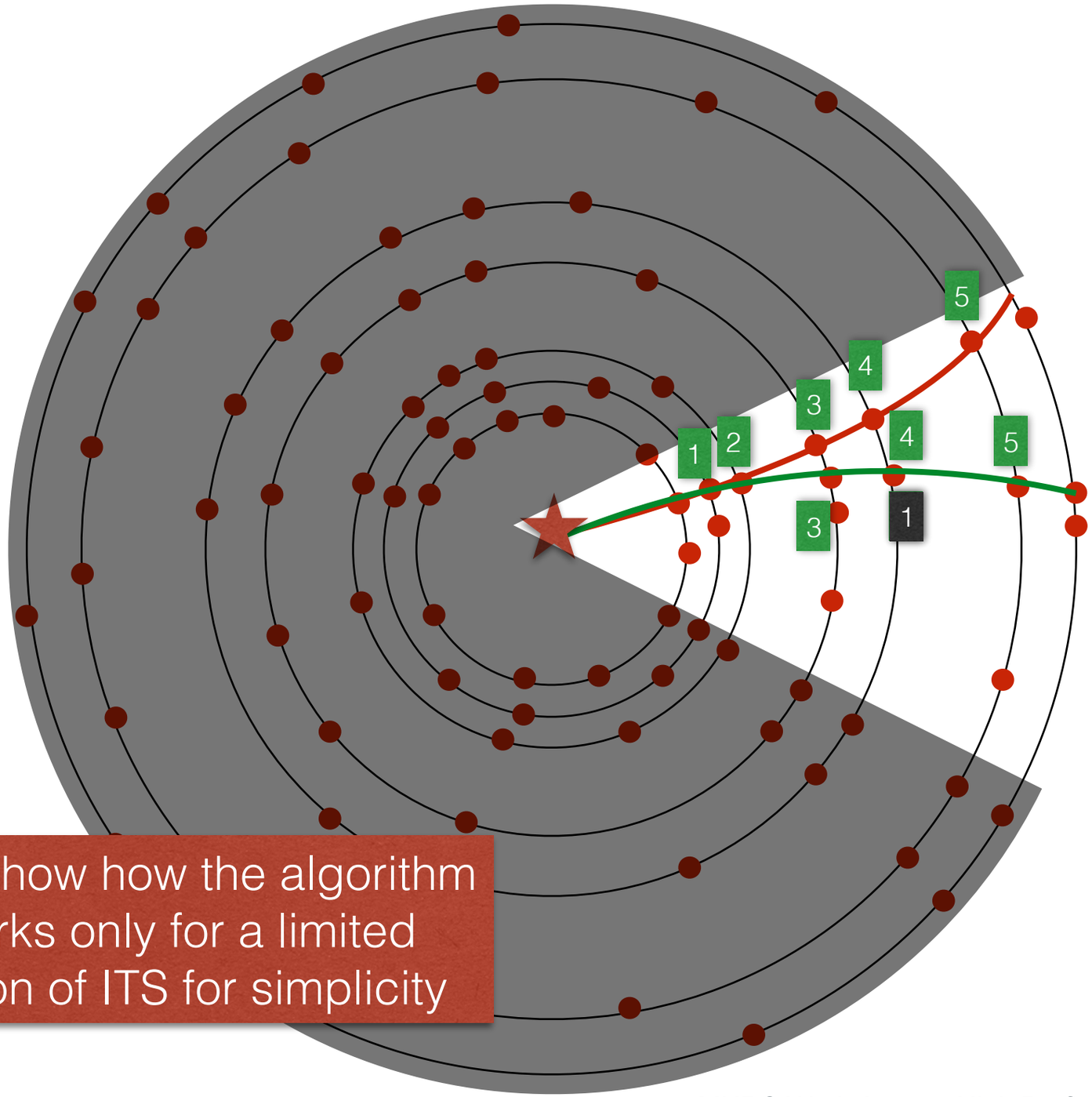
For each cluster on each layer a 2D window is opened. Then the clusters are joined with those on the next layer within the window

Subsequent doublets are combined in cells (3 points seed) and track params are computed

Each cell has an index representing the number of connected inner cells + 1

I will show how the algorithm works only for a limited region of ITS for simplicity

The Cellular Automaton approach



I will show how the algorithm works only for a limited region of ITS for simplicity

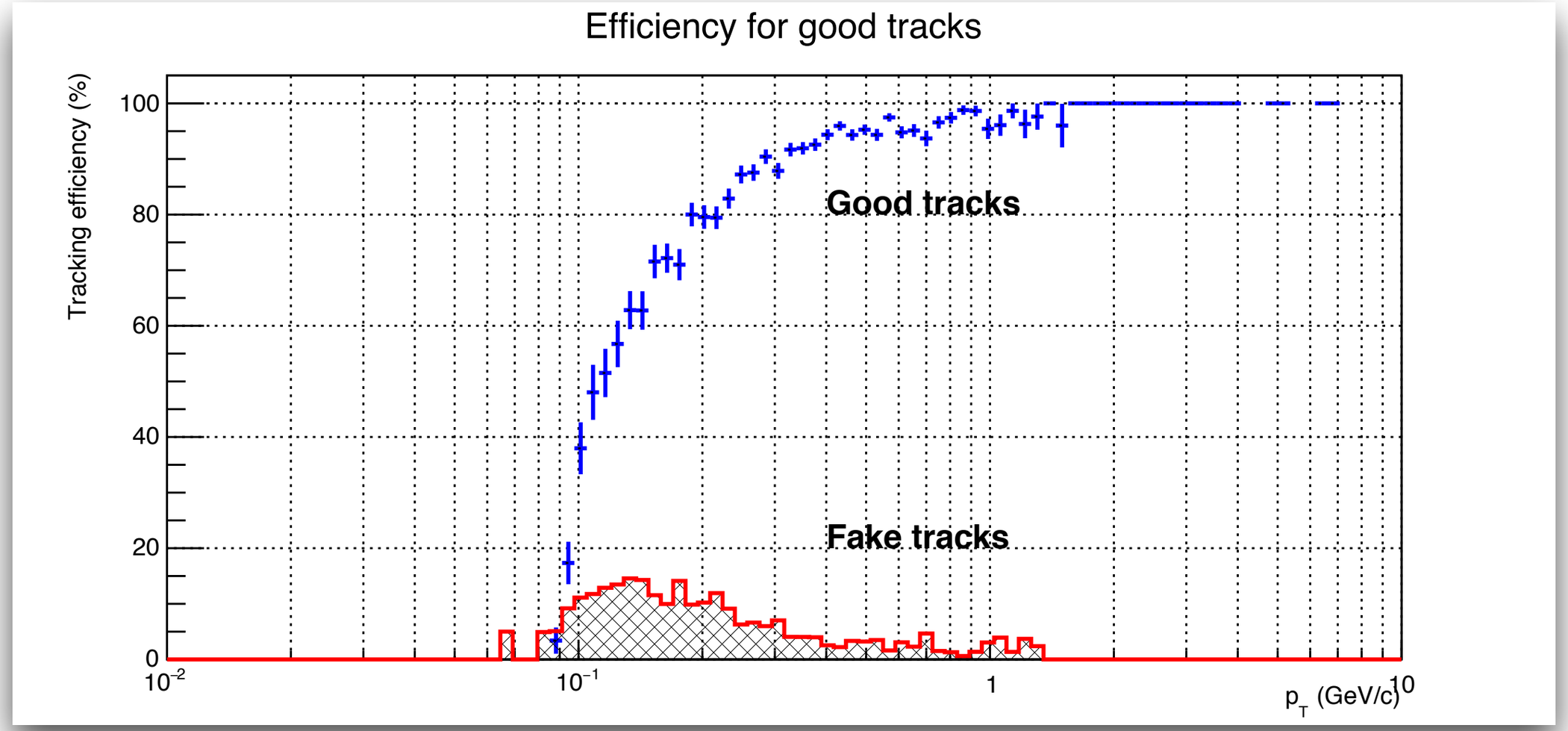
For each cluster on each layer a 2D window is opened. Then the clusters are joined with those on the next layer within the window

Subsequent doublets are combined in cells (3 points seed) and track params are computed

Each cell has an index representing the number of connected inner cells + 1

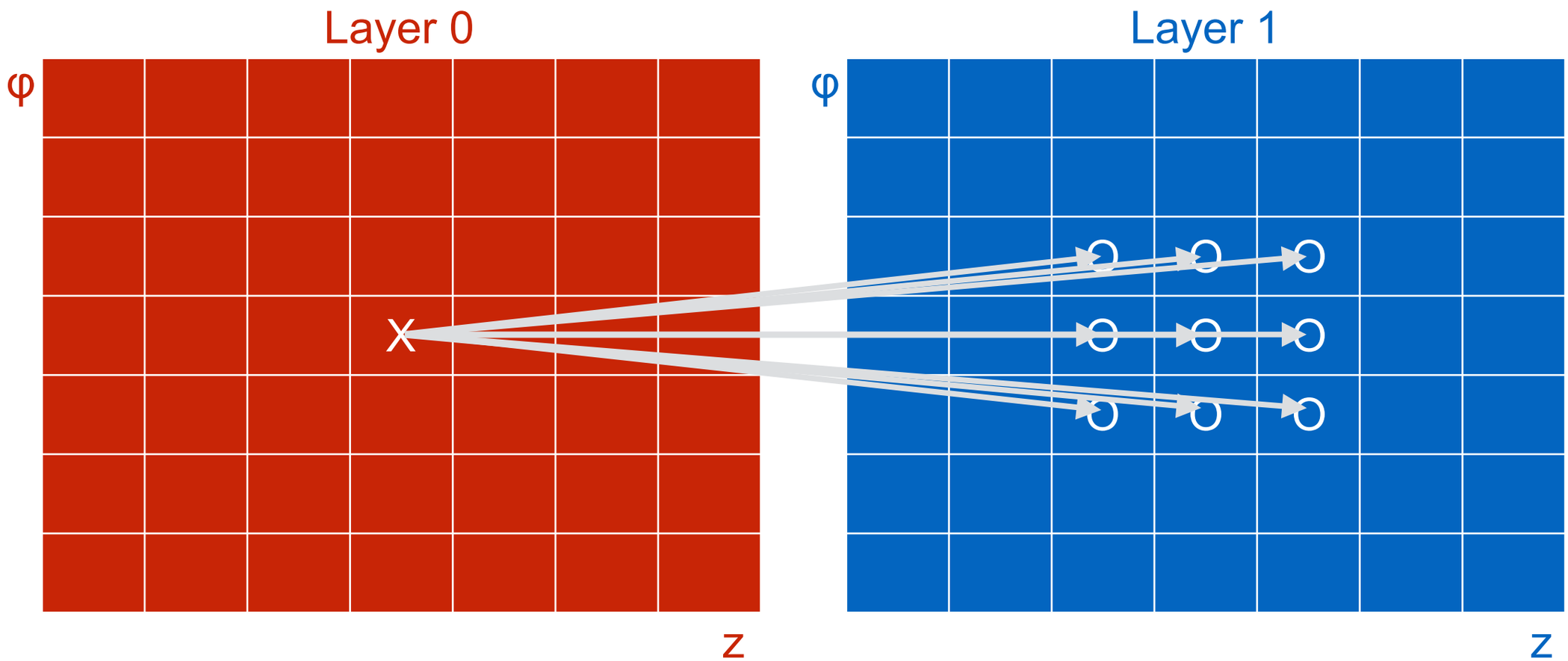
Longest, continuous sequences of indices represent candidates

Tracking performance Pb-Pb



	p_T > 0.6 GeV/c	p_T > 2 GeV/c	Full
Central Pb-Pb	~0.3 s	~0.2 s	~0.7 s
Central Pb-Pb with noise	~1.3 s	~1.0 s	~5 s
p-p with noise	~0.6 s	~0.7 s	~2.3 s

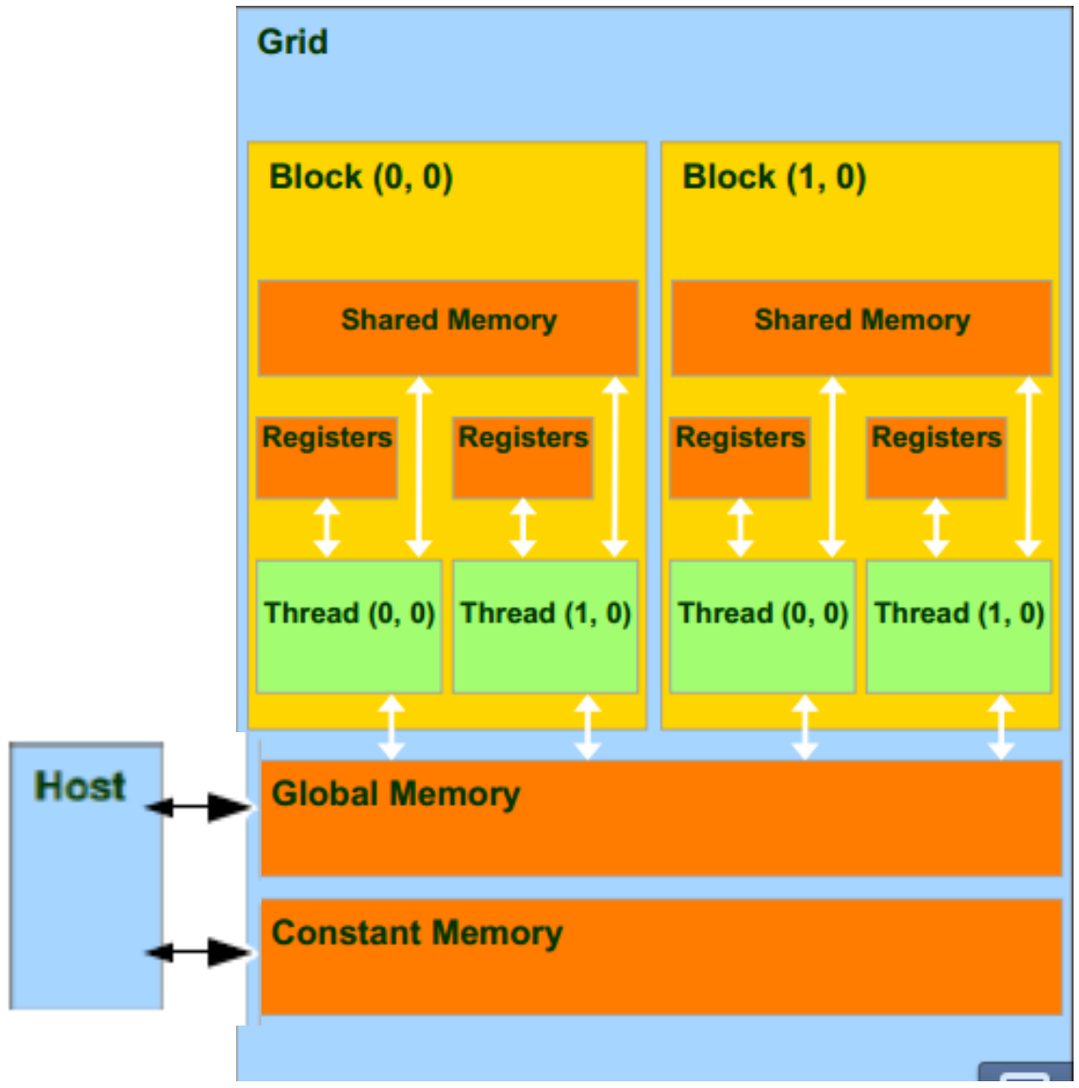
How to speed up: grids to browse data



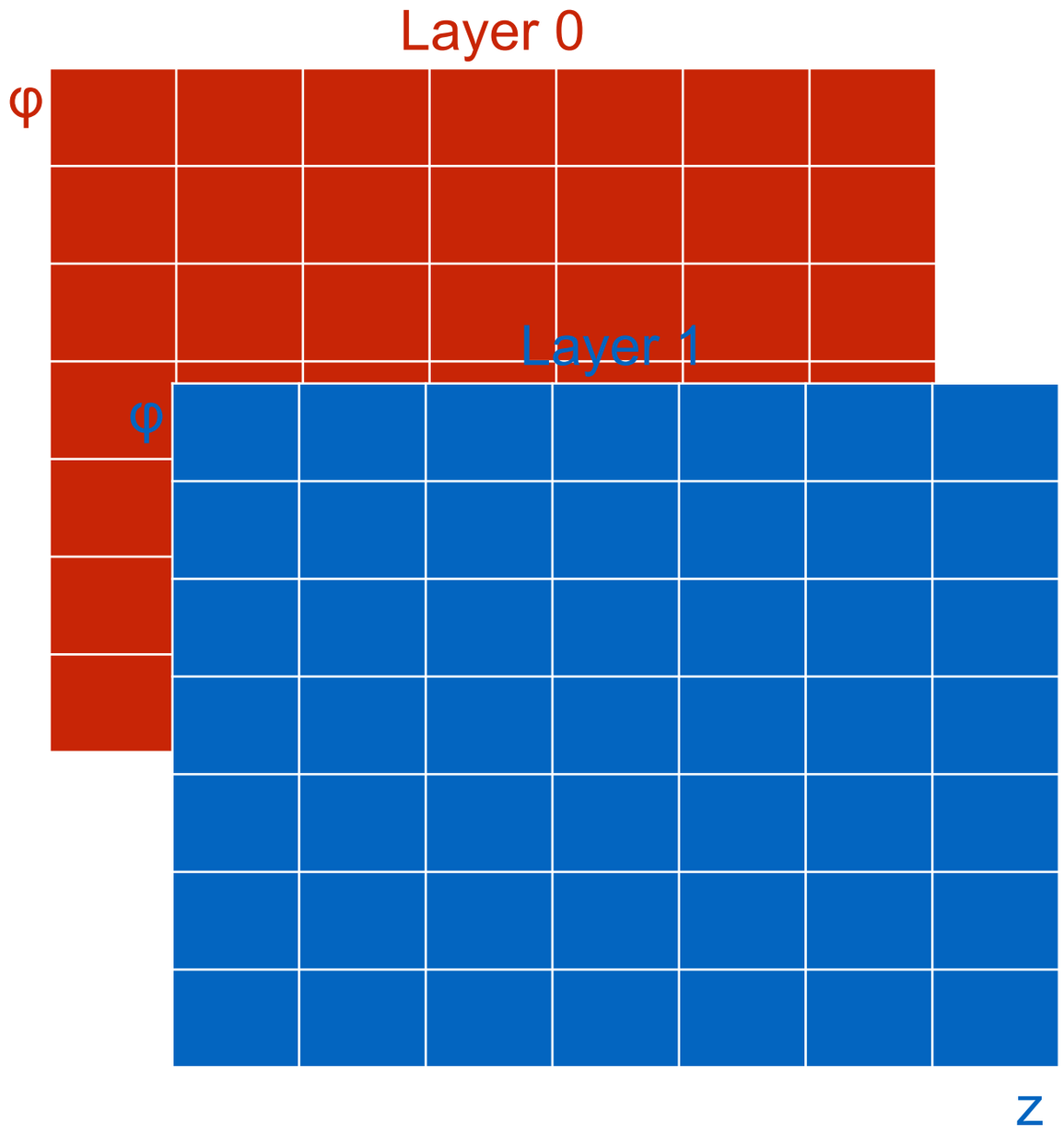
Lookup table approach. It permits:

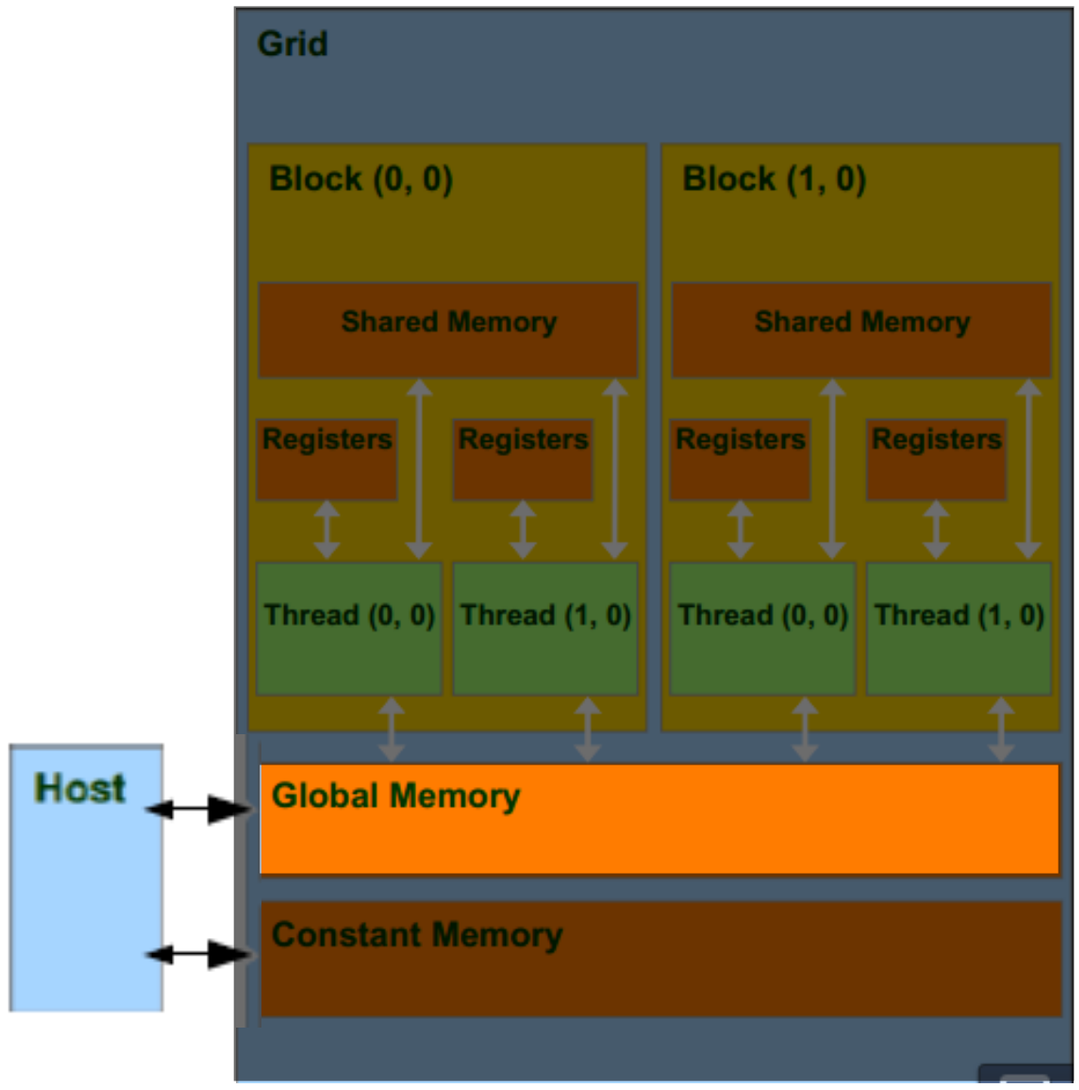
- **Fine grained parallelism**: for each arrow connecting layer0 to layer1 a compute element is used
- **Coarse grained parallelism**: for each cell/row of layer 0 a compute element is used.

Porting on GPU: how to make doublets



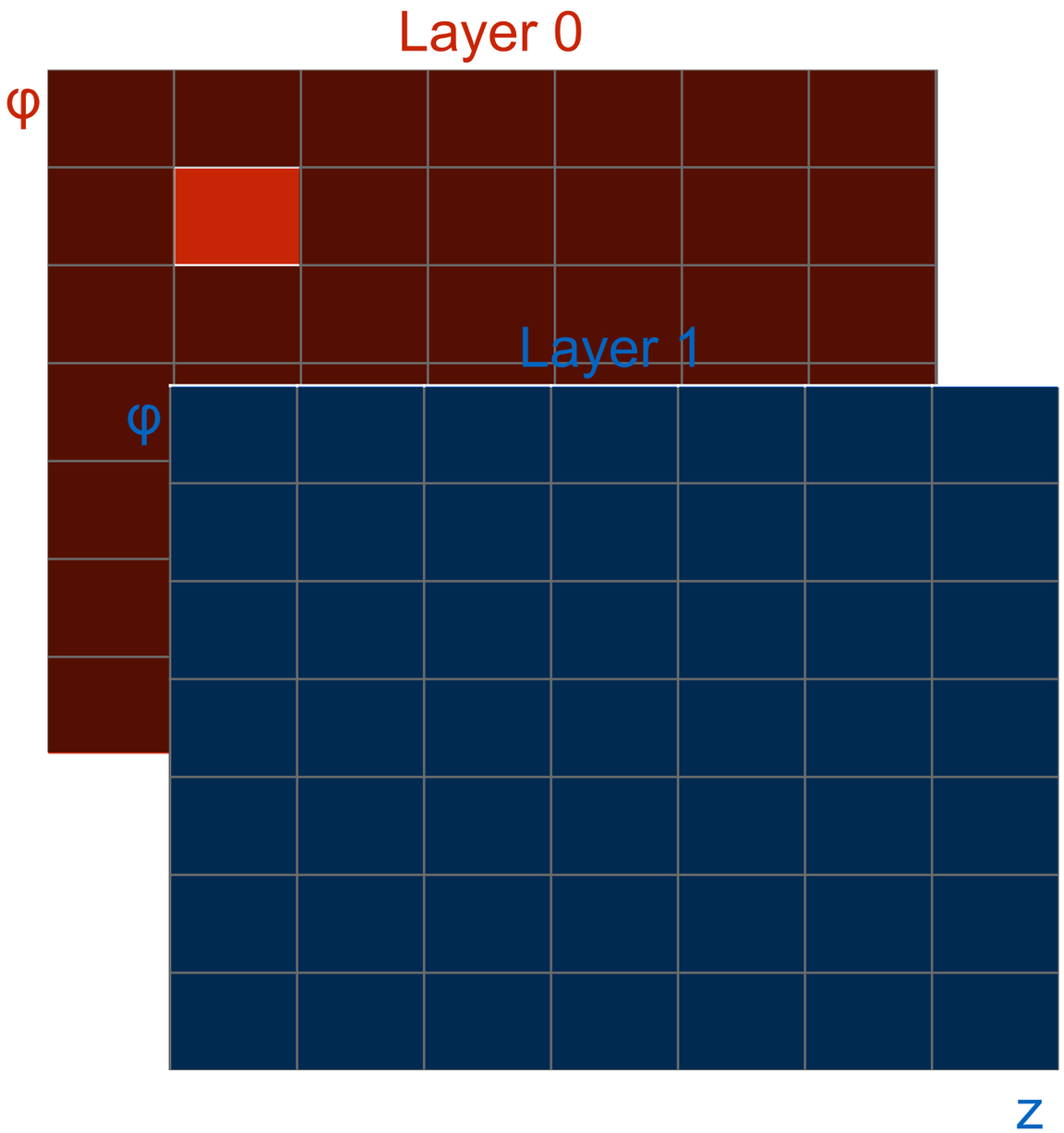
Porting on GPU: how to make doublets

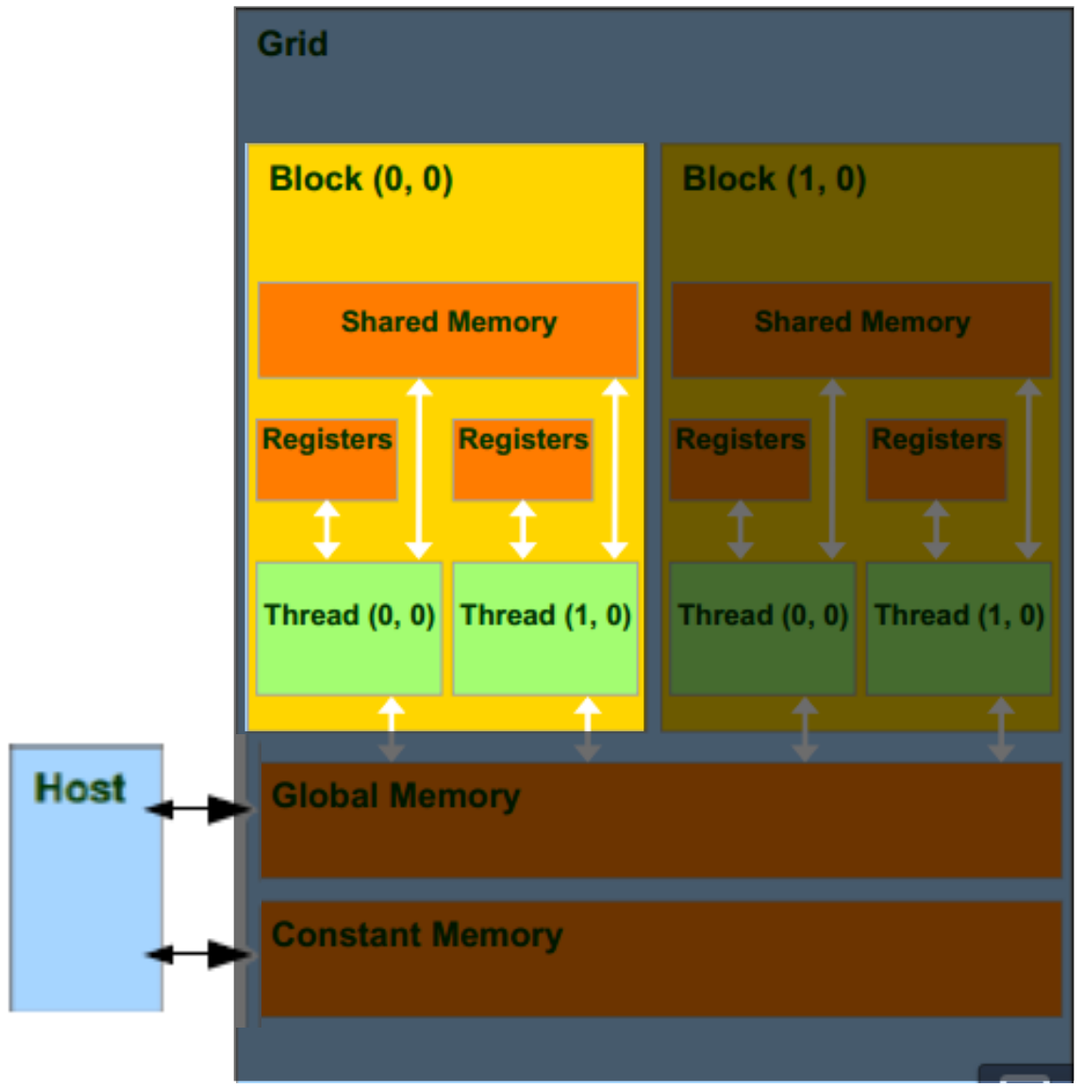




Transfer of the clusters and LUT on the GPU global (constant) memory

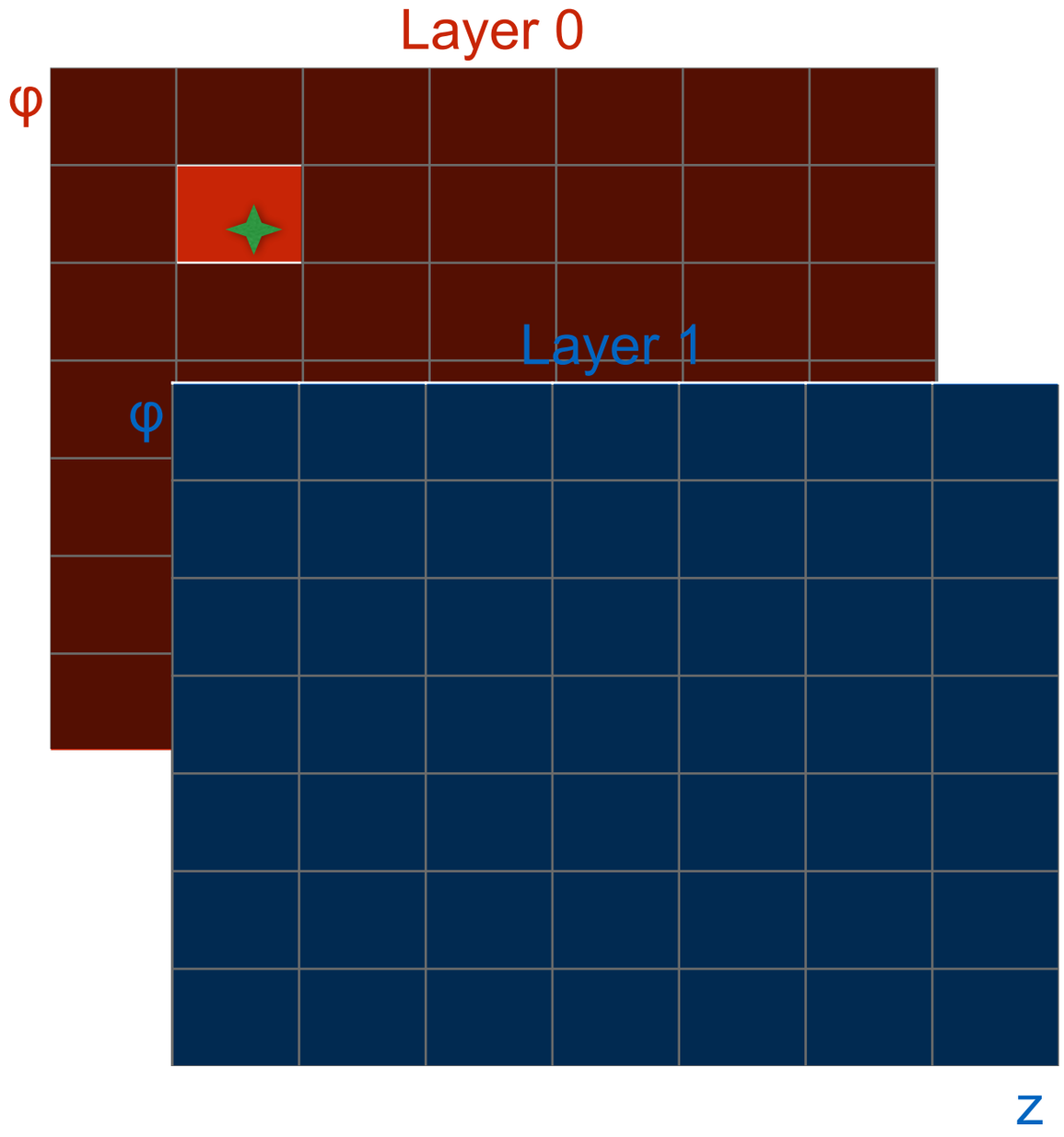
Porting on GPU: how to make doublets

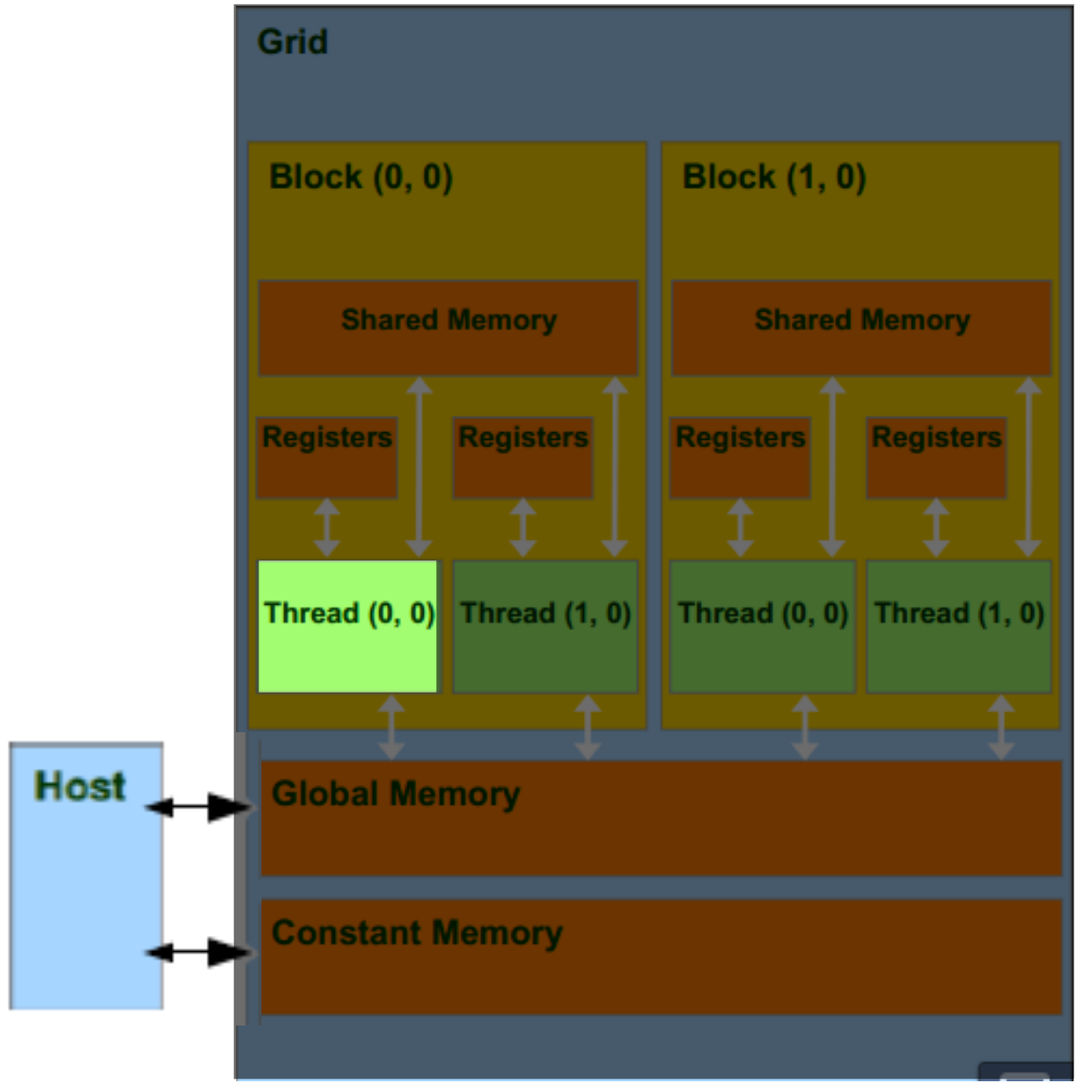




Each block of thread will process one bin on layer 0

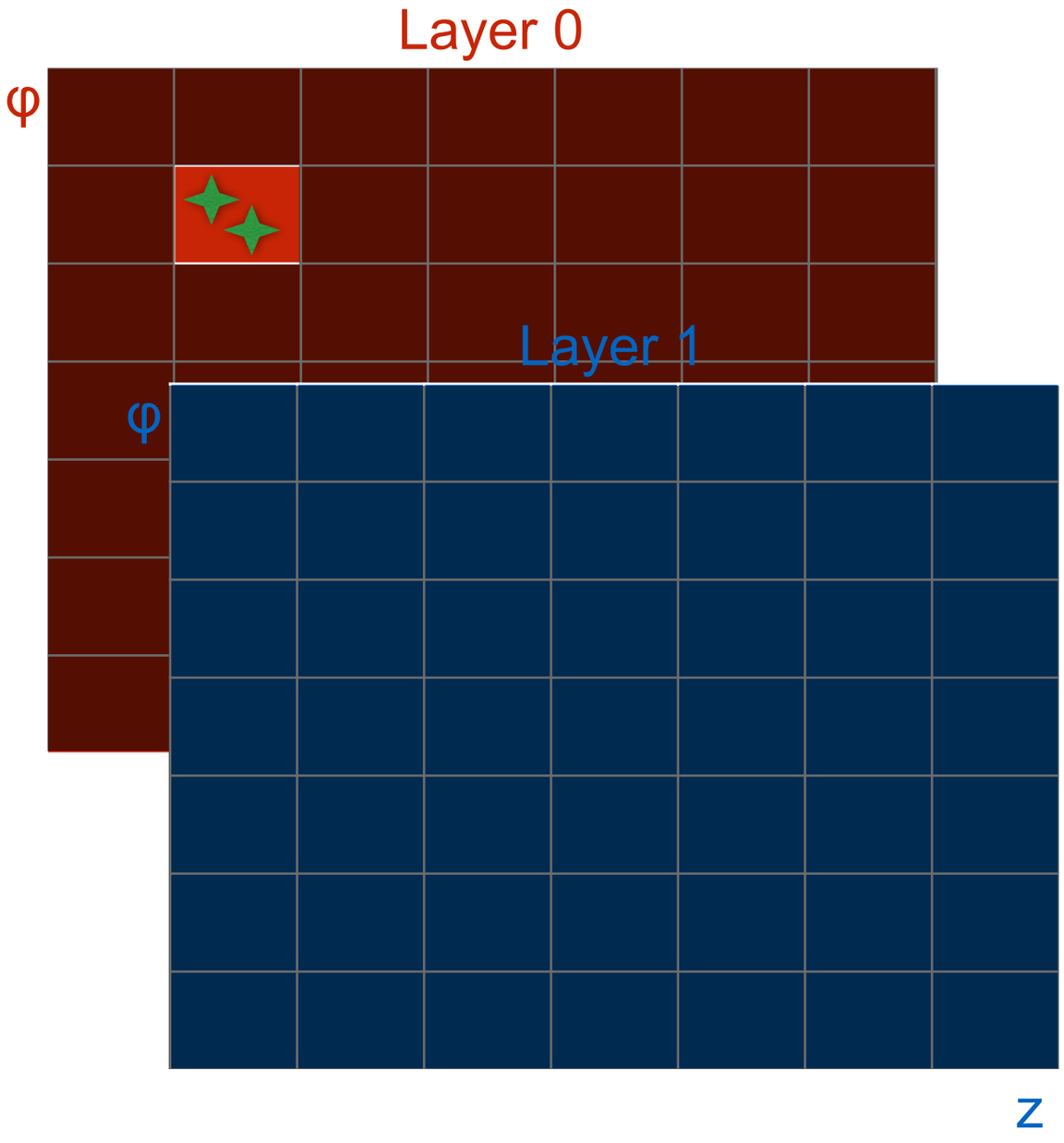
Porting on GPU: how to make doublets

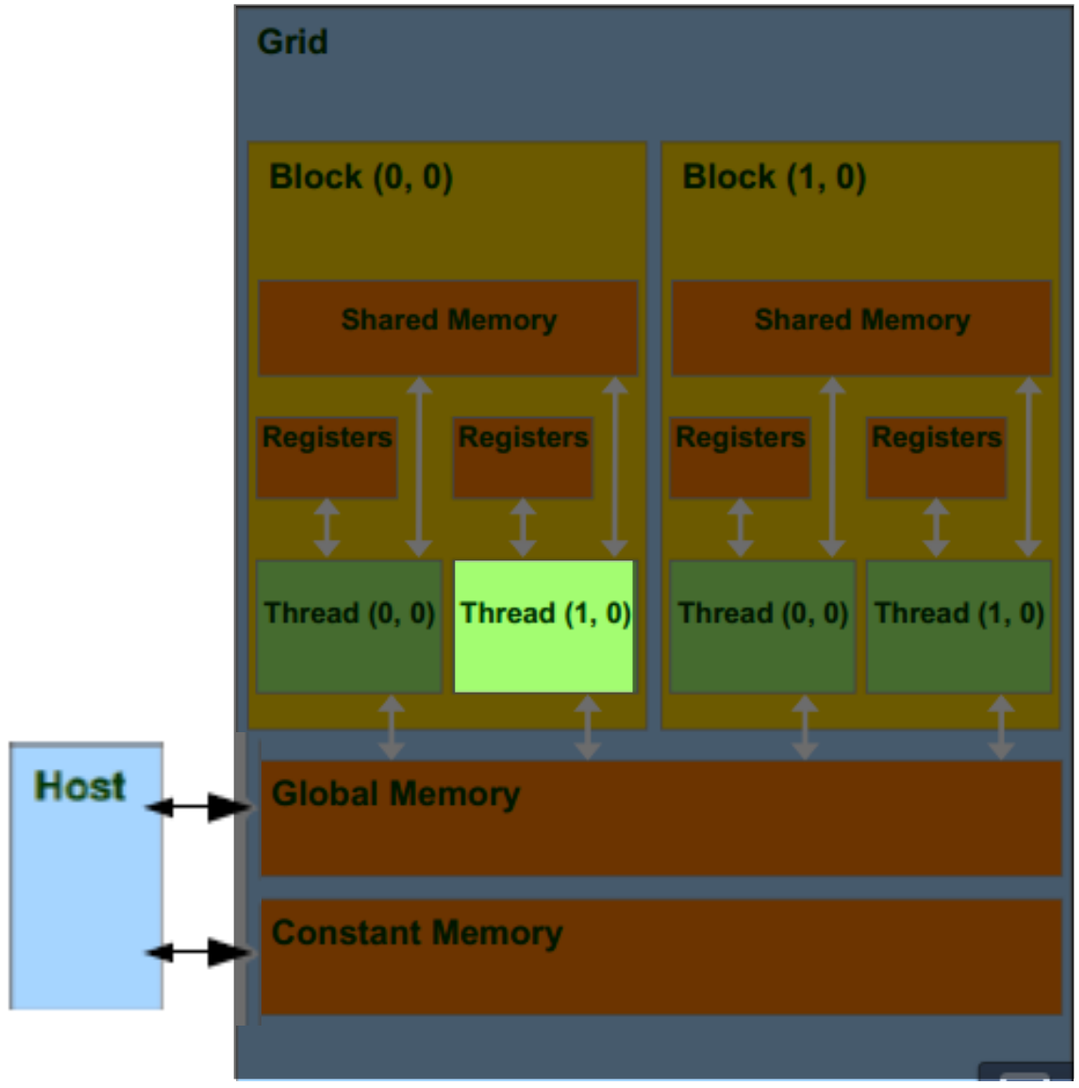




Each thread will take care of one cluster layer 0

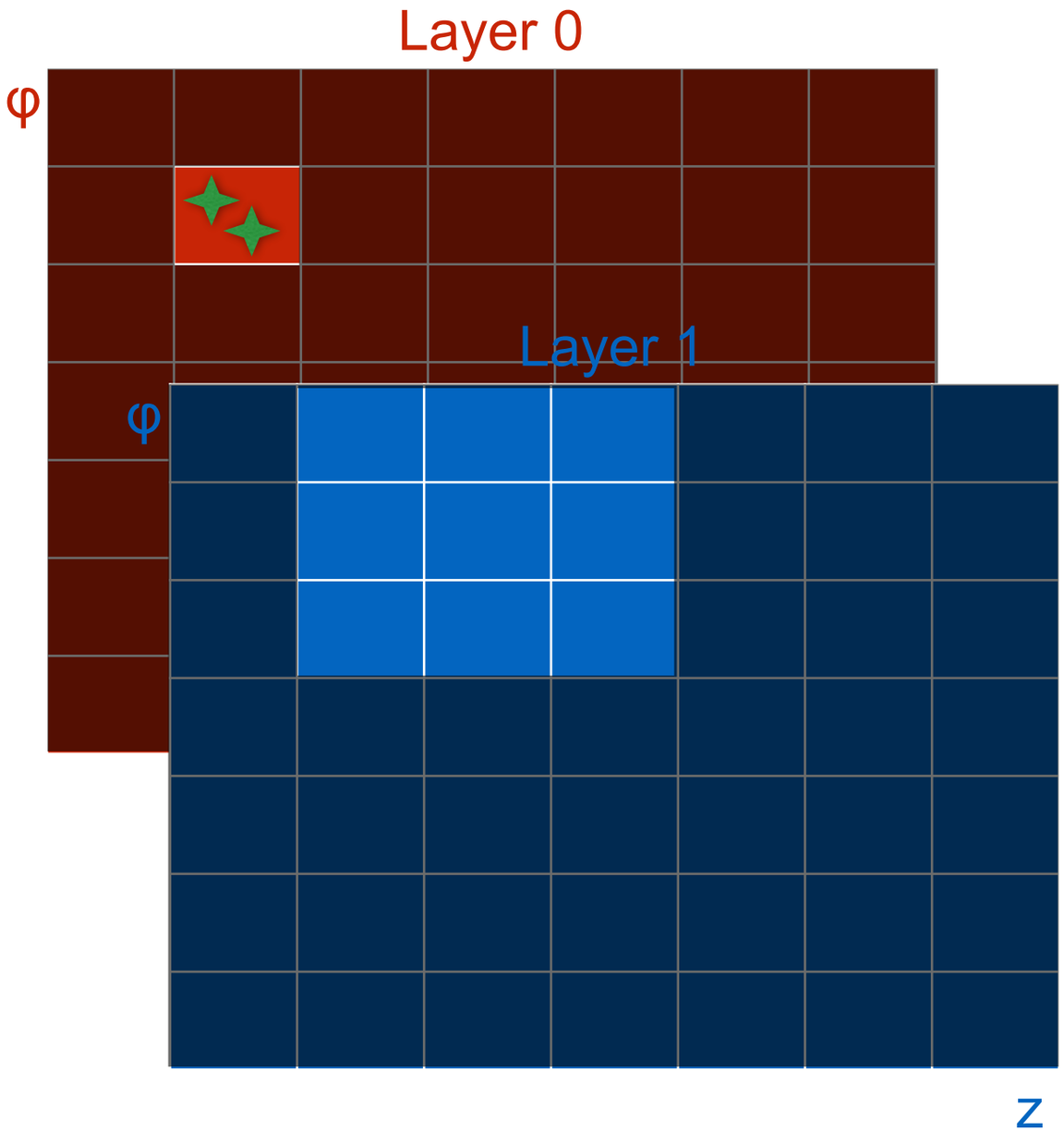
Porting on GPU: how to make doublets

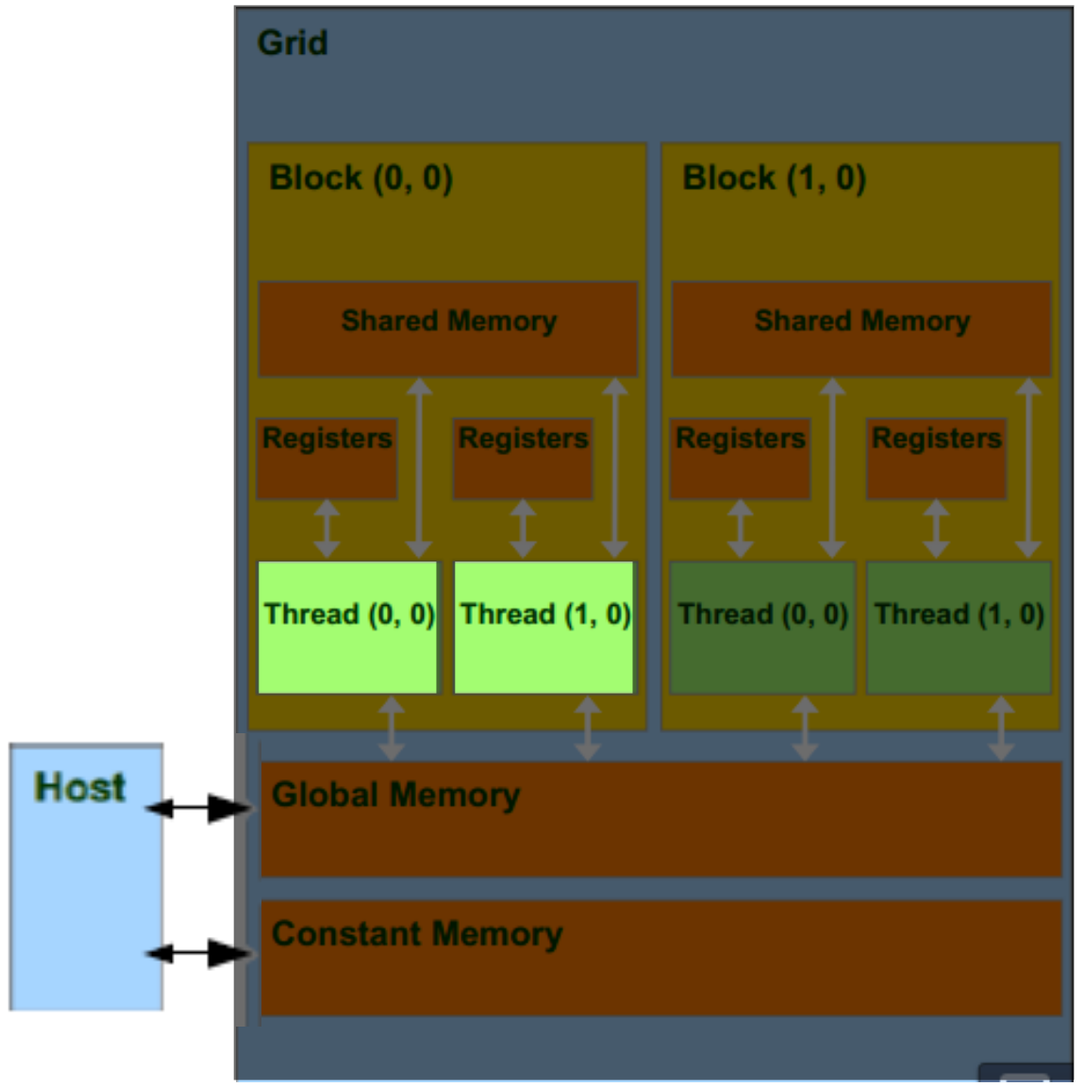




Each thread will take care of one cluster layer 0

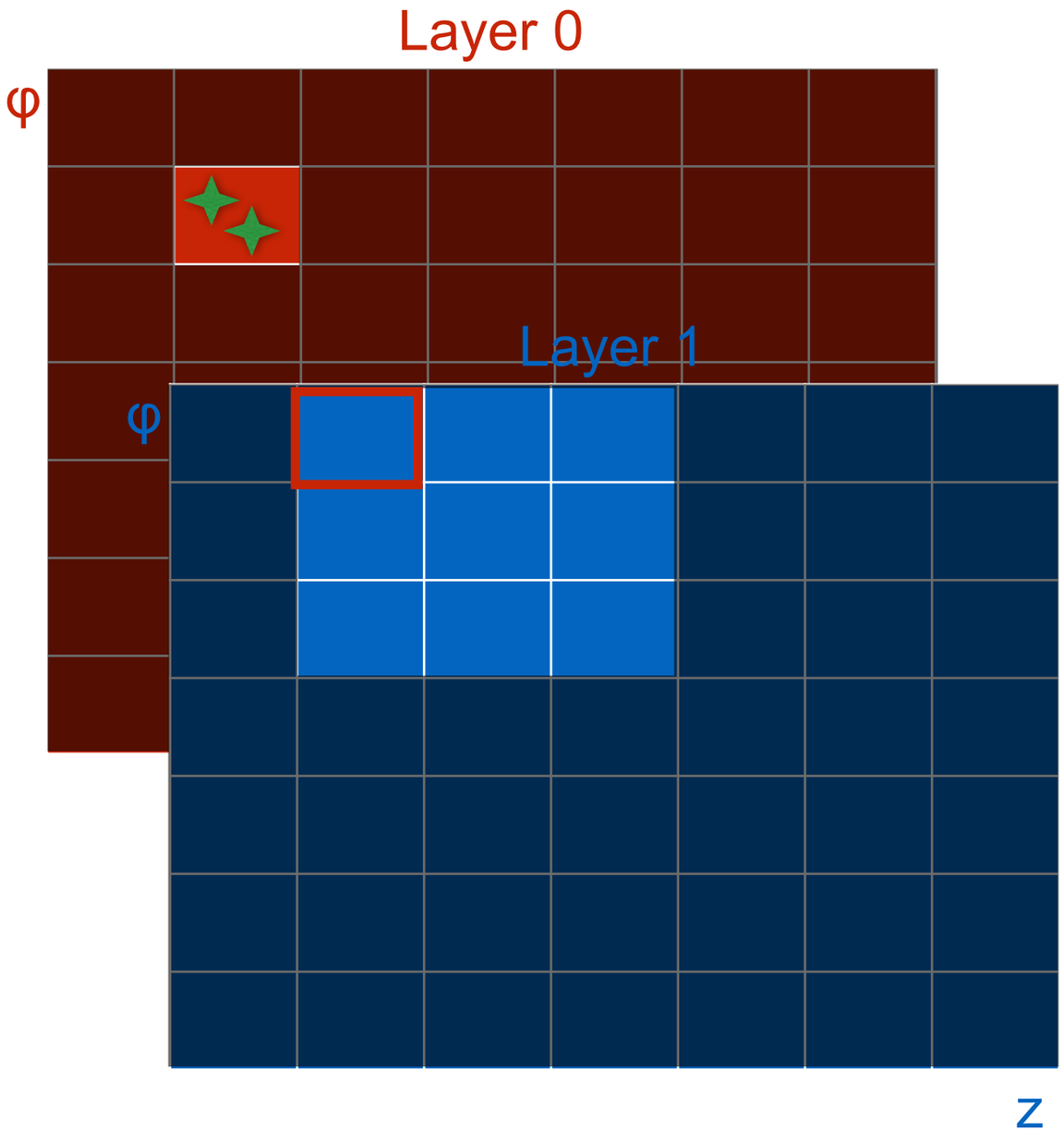
Porting on GPU: how to make doublets

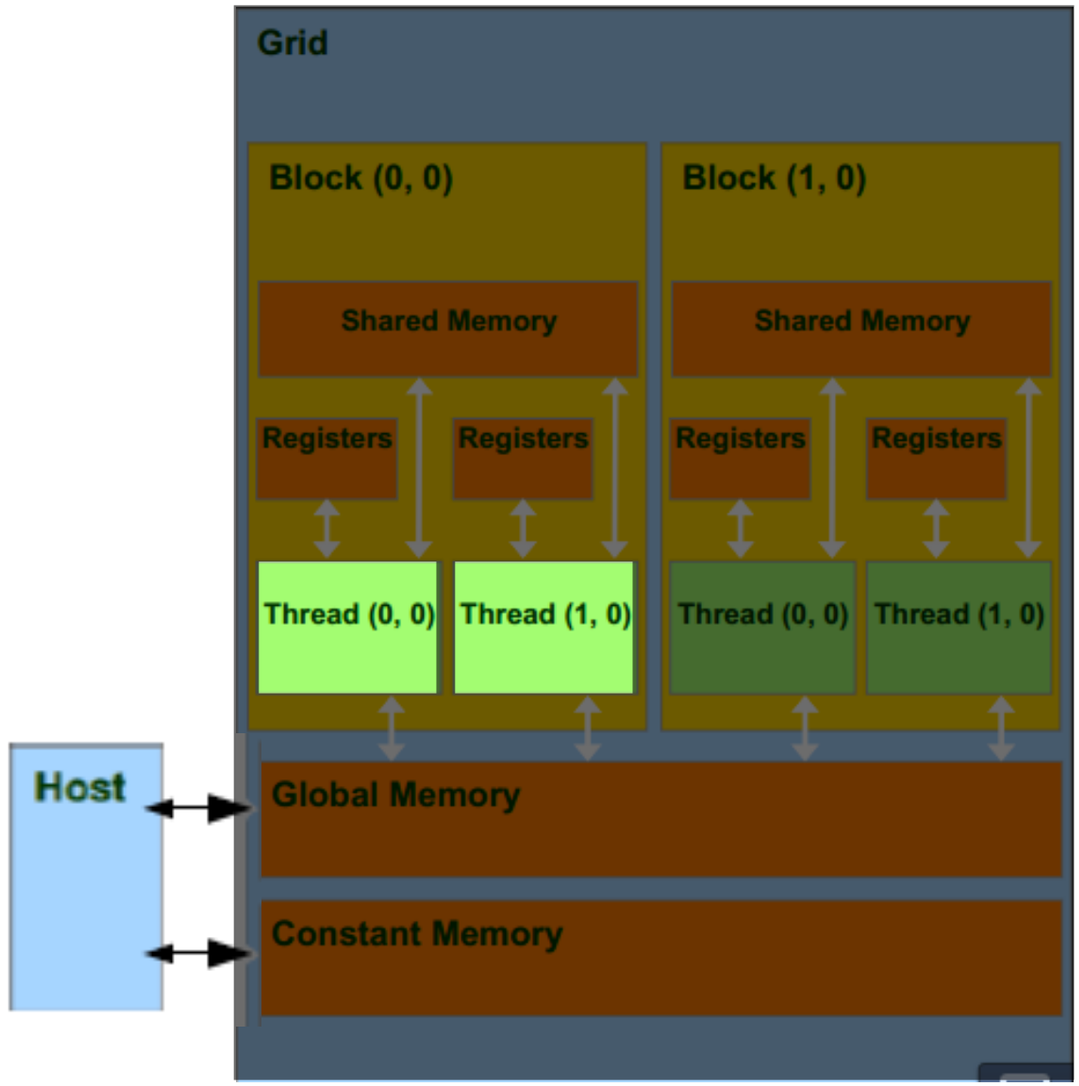




All the clusters of one bin on layer 1 are copied in the shared memory and a for loop starts over these clusters looking for good doublets

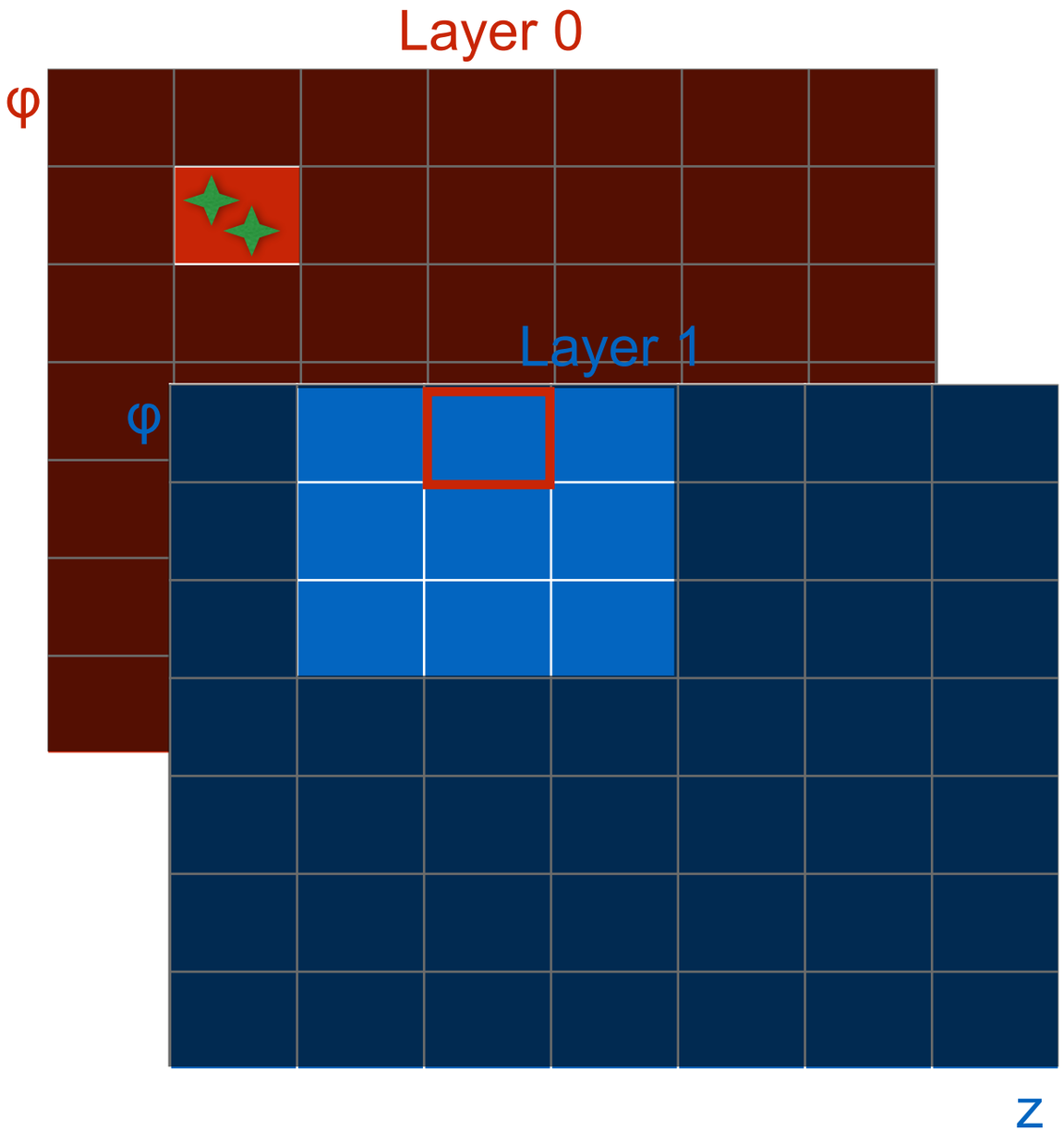
Porting on GPU: how to make doublets

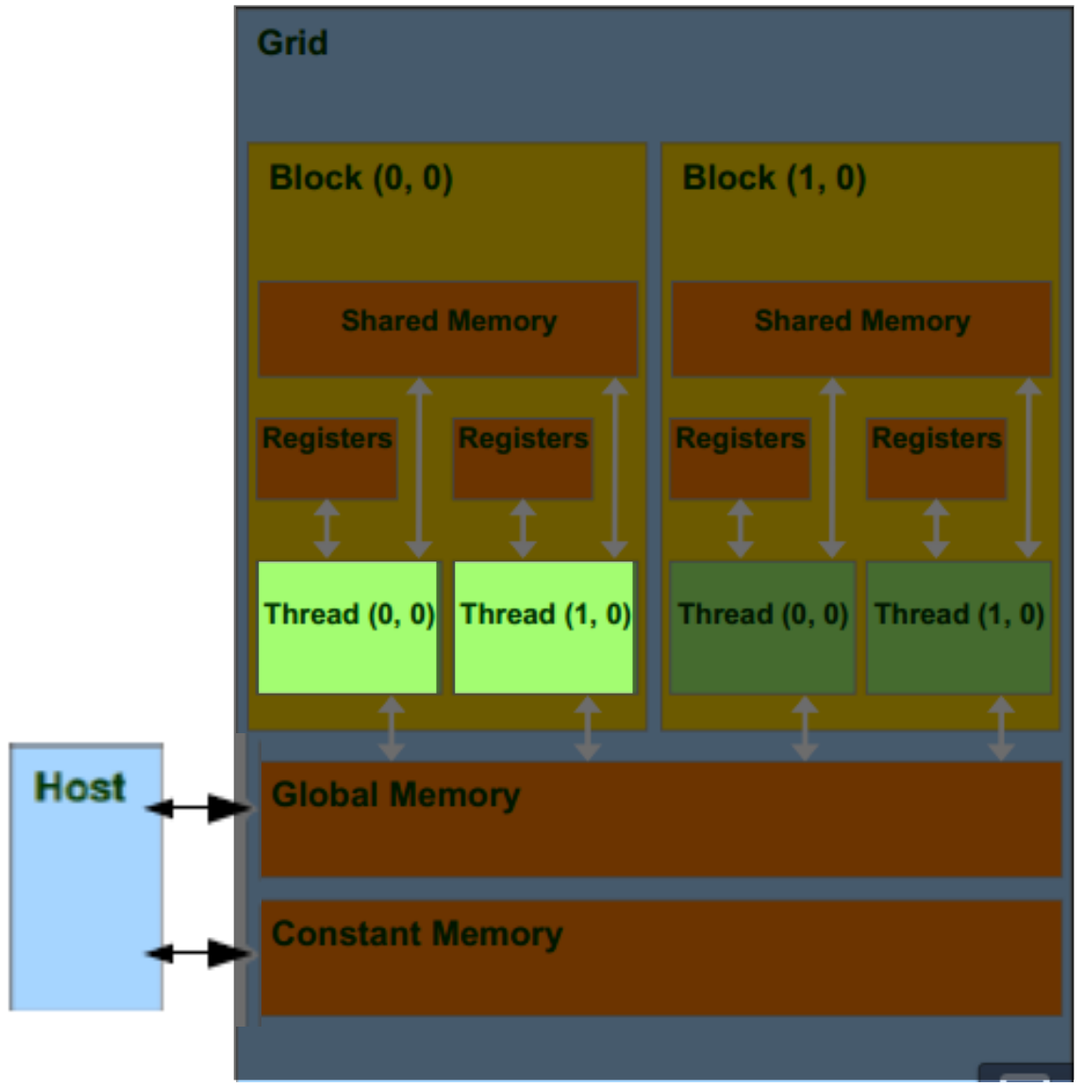




All the clusters of one bin on layer 1 are copied in the shared memory and a for loop starts over these clusters looking for good doublets

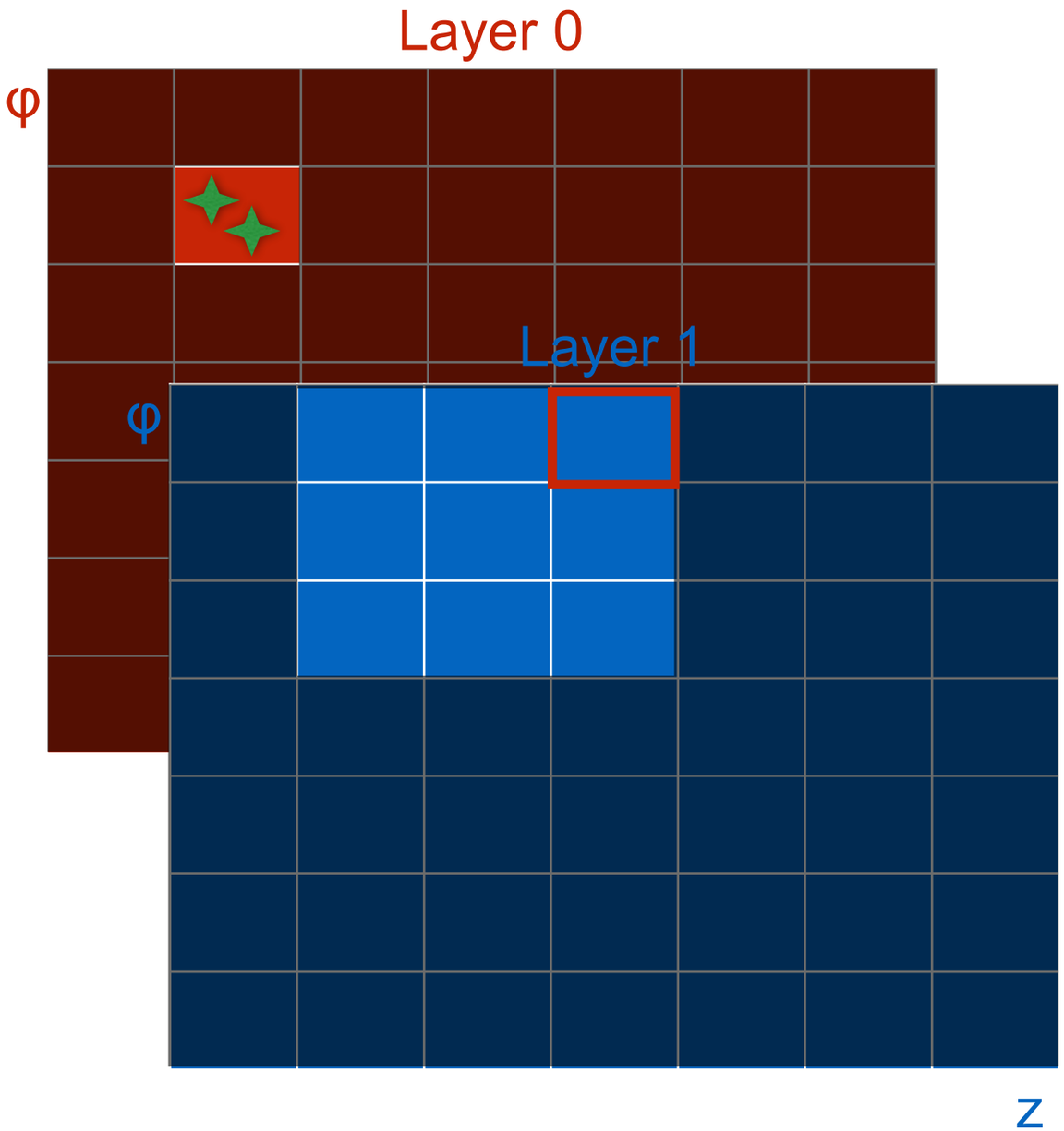
Porting on GPU: how to make doublets

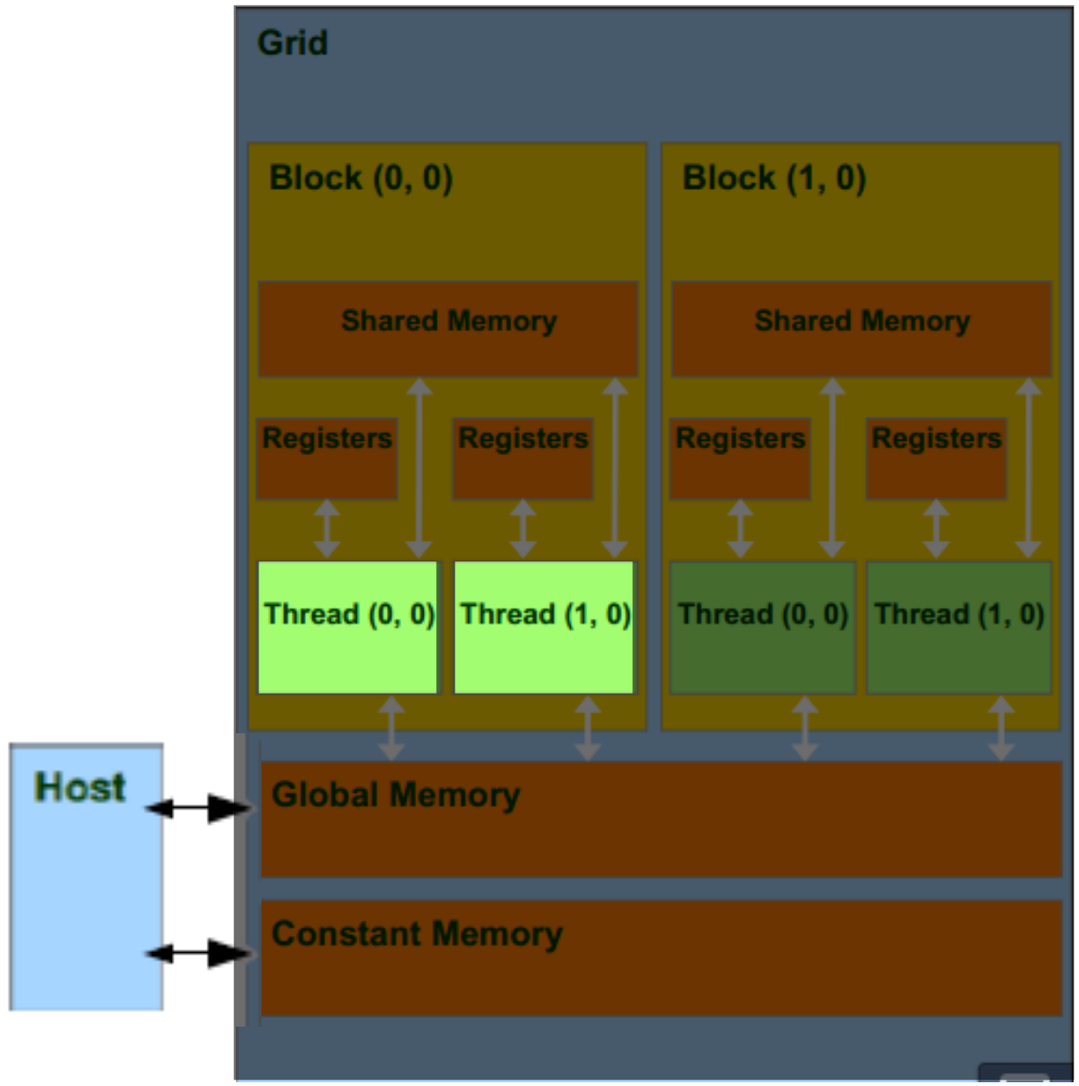




All the clusters of one bin on layer 1 are copied in the shared memory and a for loop starts over these clusters looking for good doublets

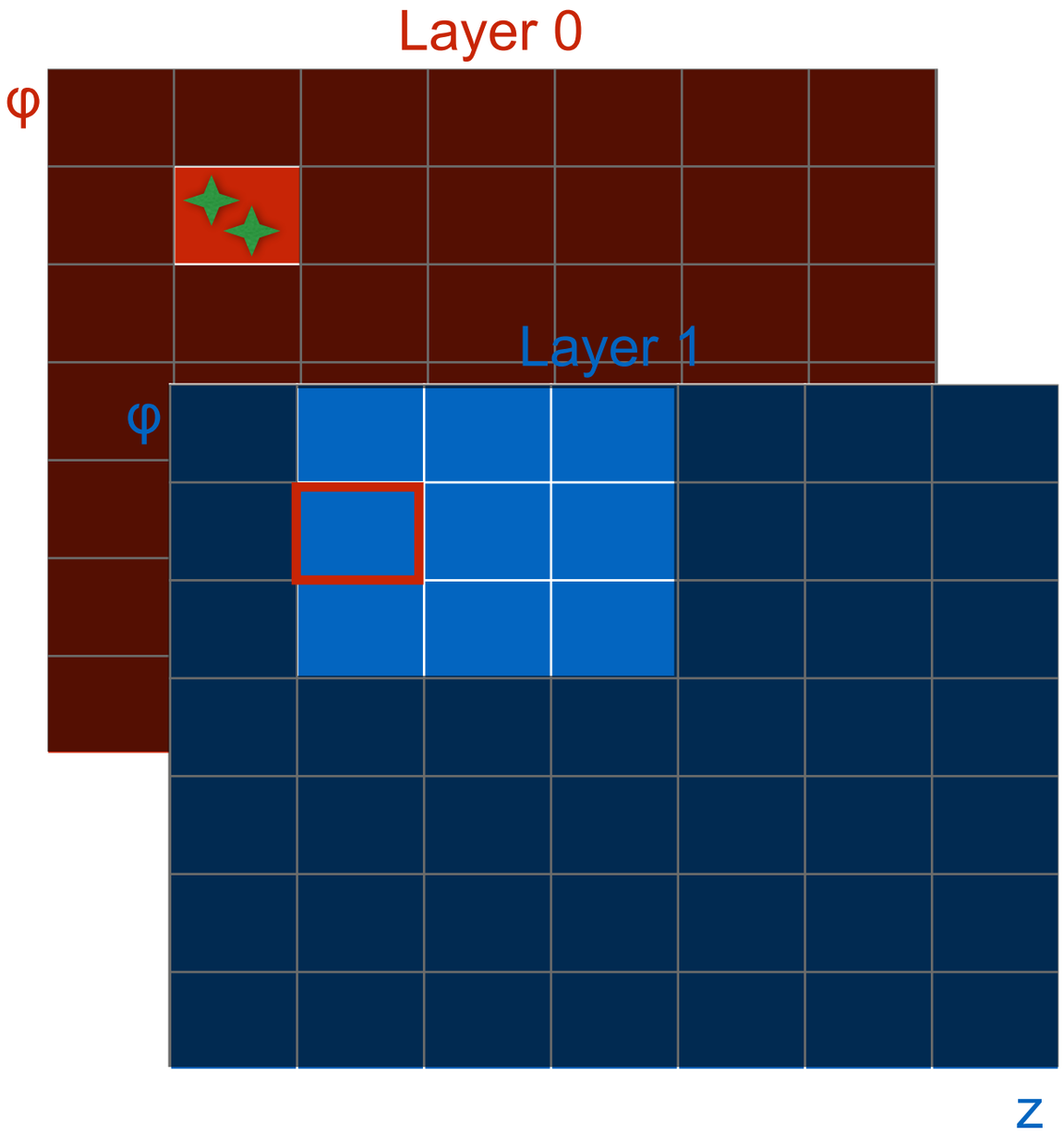
Porting on GPU: how to make doublets

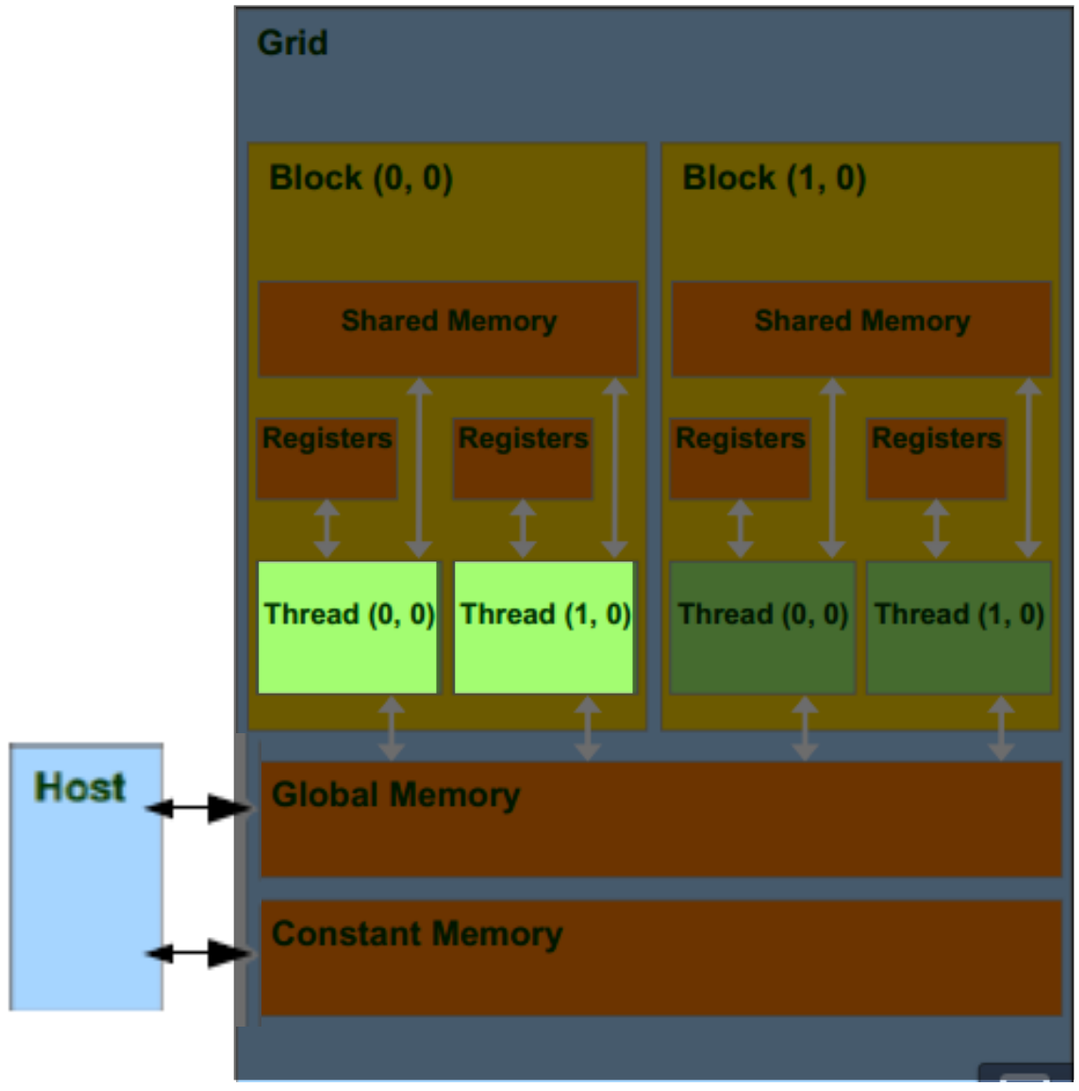




All the clusters of one bin on layer 1 are copied in the shared memory and a for loop starts over these clusters looking for good doublets

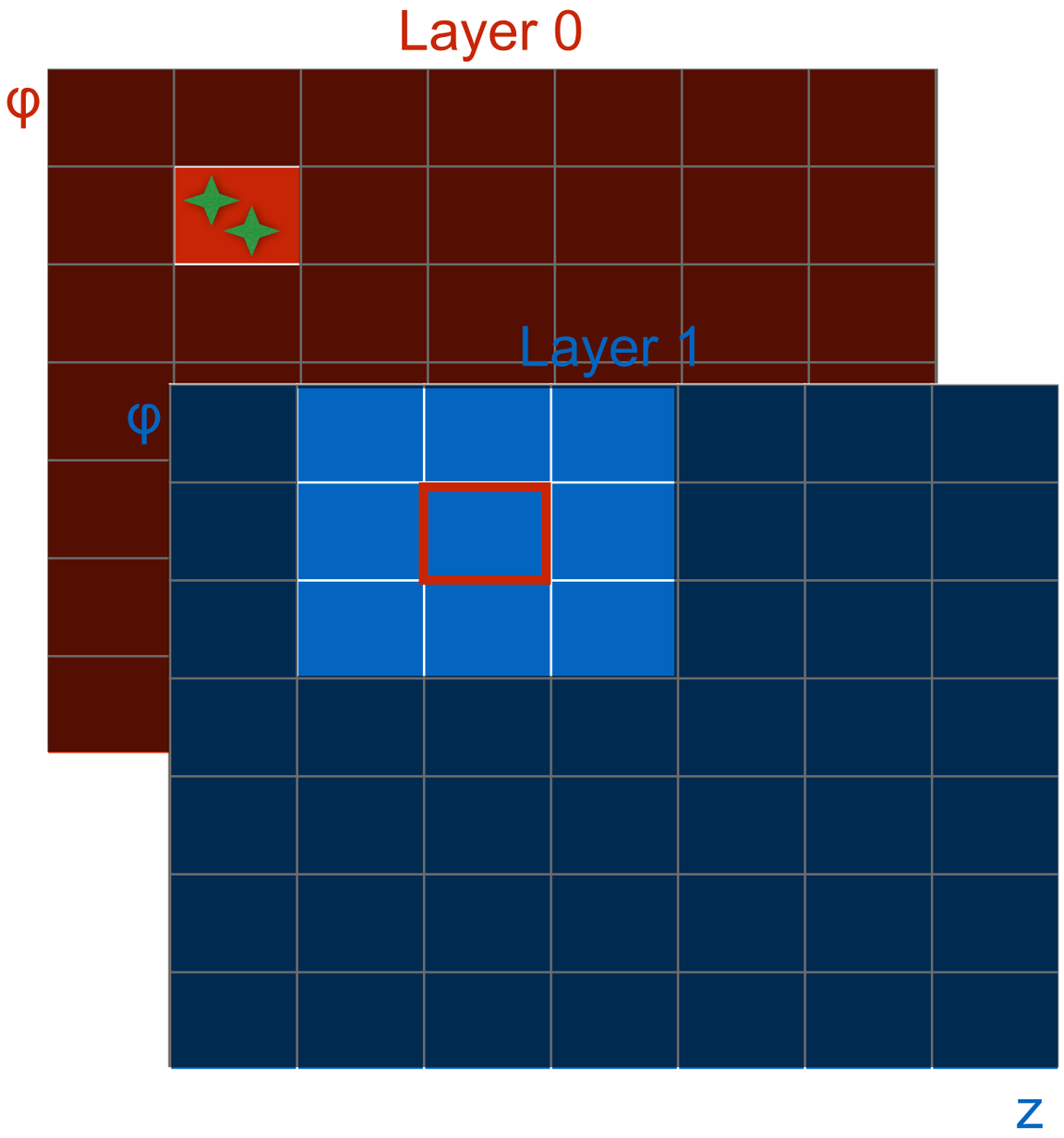
Porting on GPU: how to make doublets

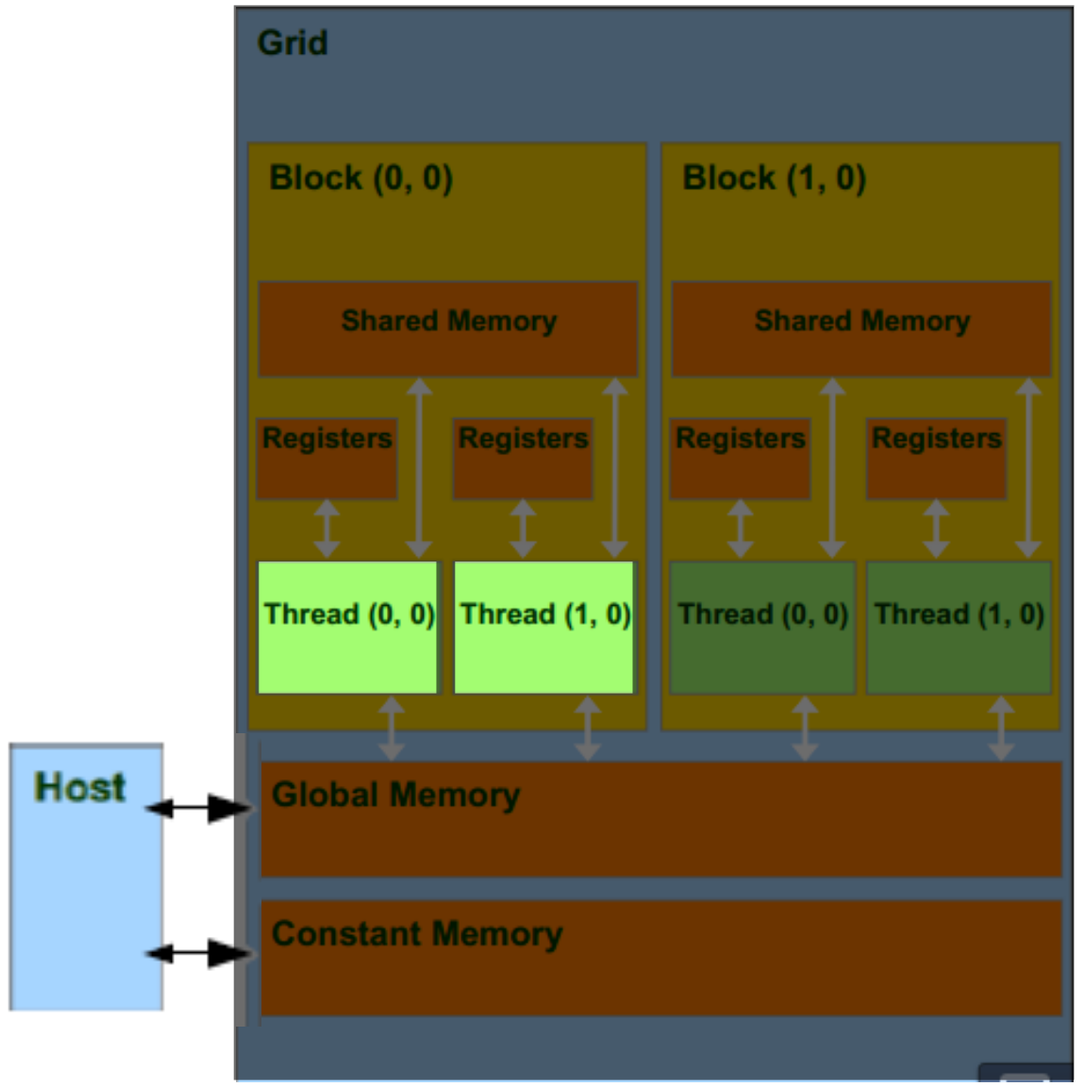




All the clusters of one bin on layer 1 are copied in the shared memory and a for loop starts over these clusters looking for good doublets

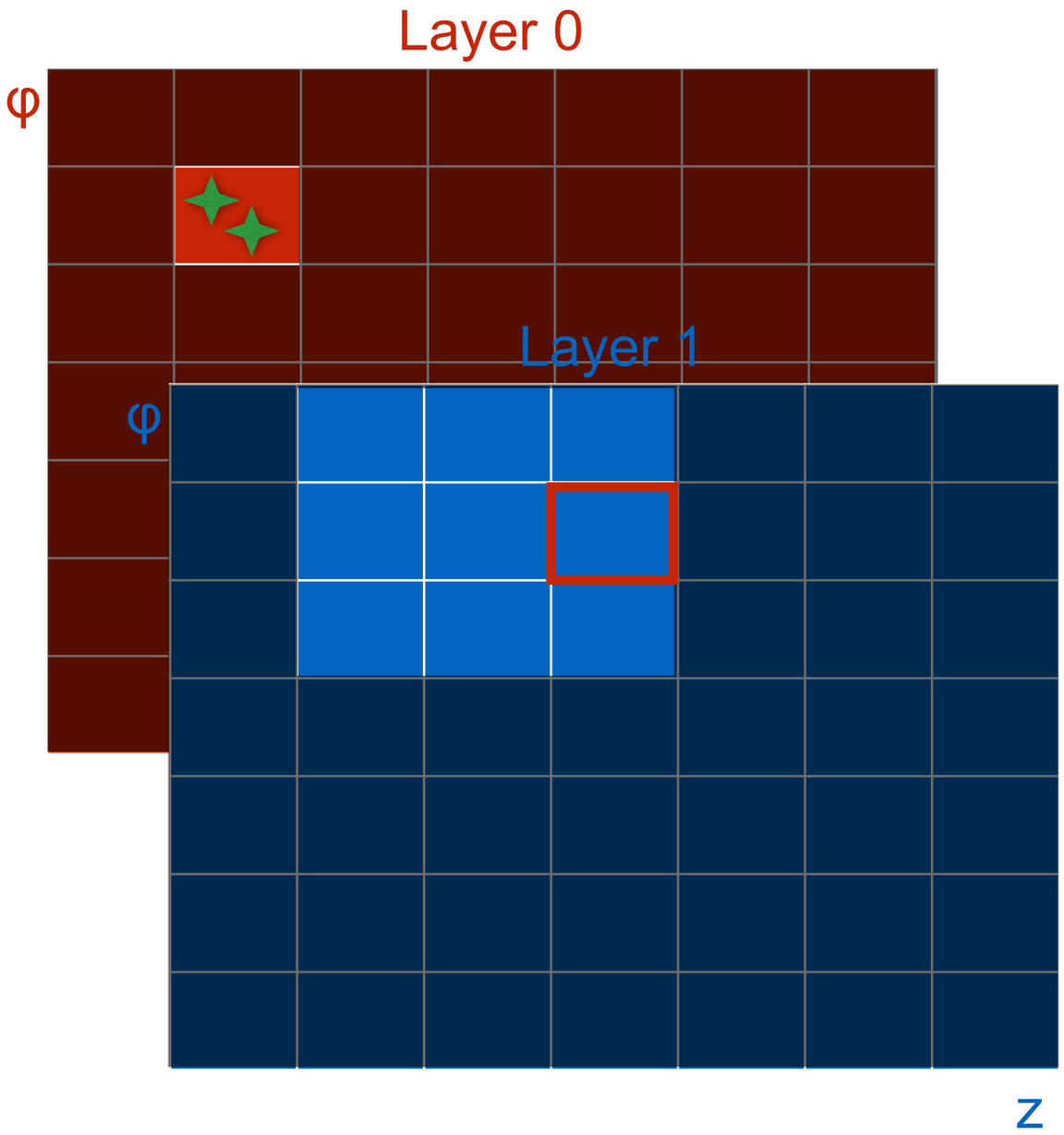
Porting on GPU: how to make doublets

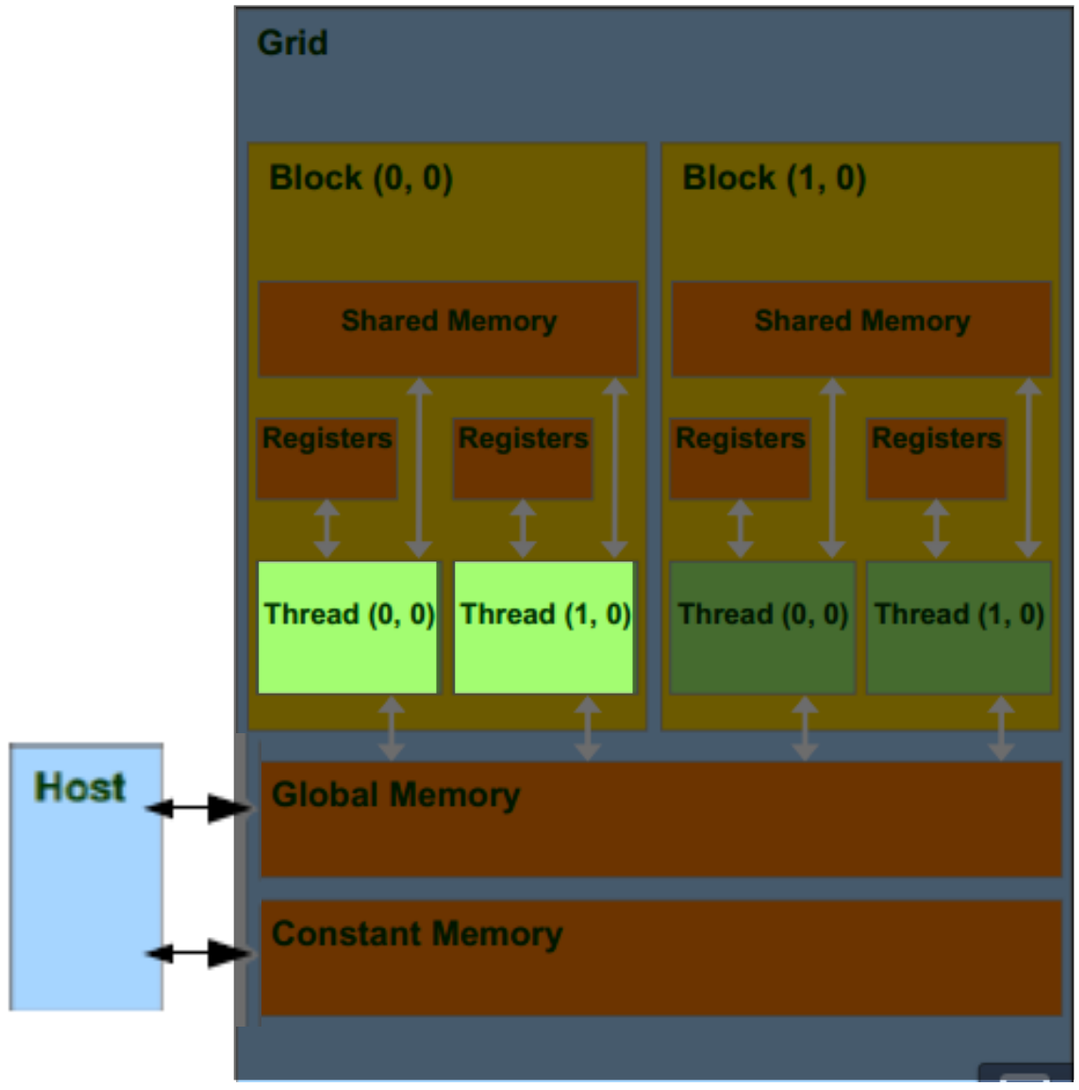




All the clusters of one bin on layer 1 are copied in the shared memory and a for loop starts over these clusters looking for good doublets

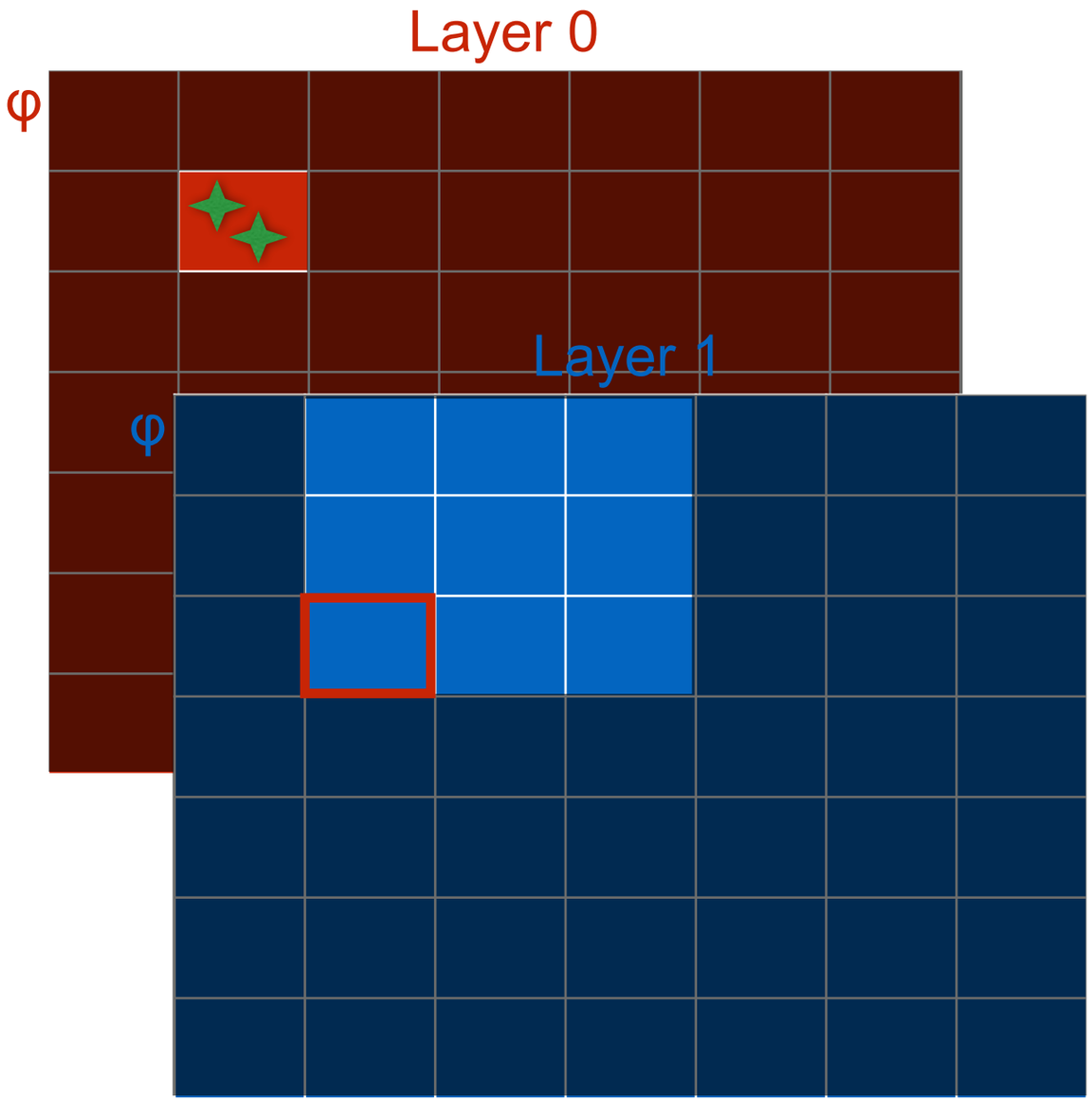
Porting on GPU: how to make doublets

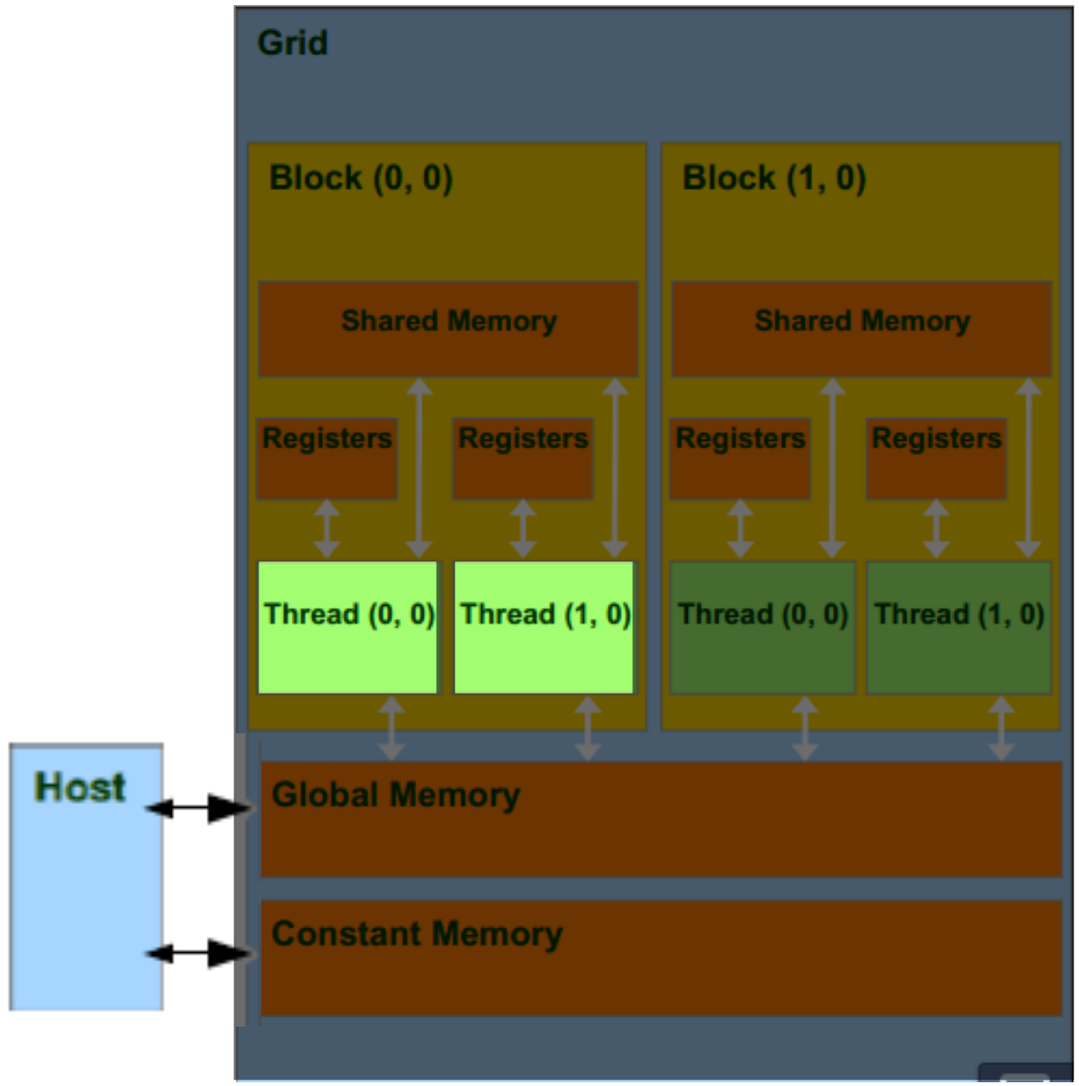




All the clusters of one bin on layer 1 are copied in the shared memory and a for loop starts over these clusters looking for good doublets

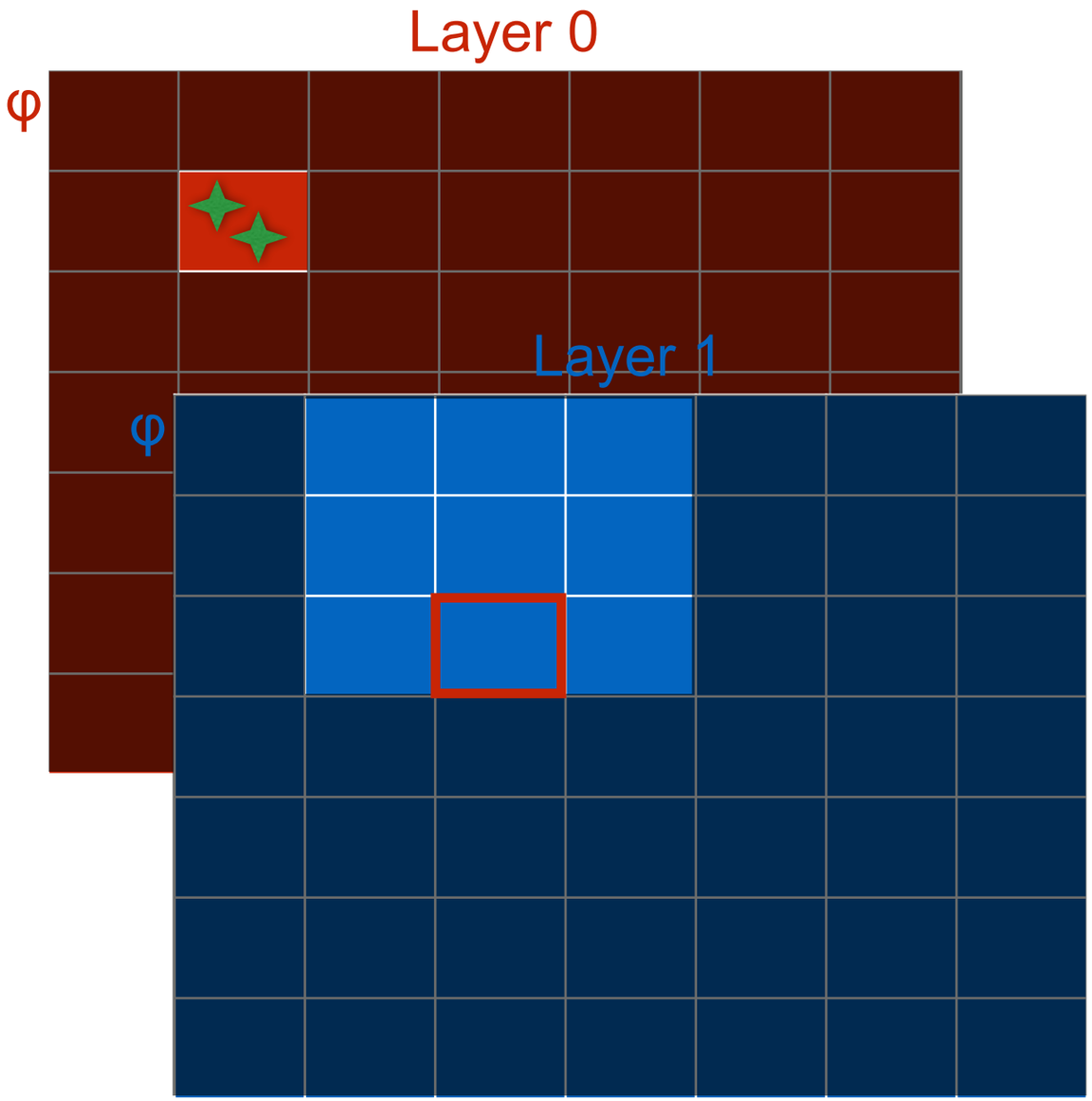
Porting on GPU: how to make doublets

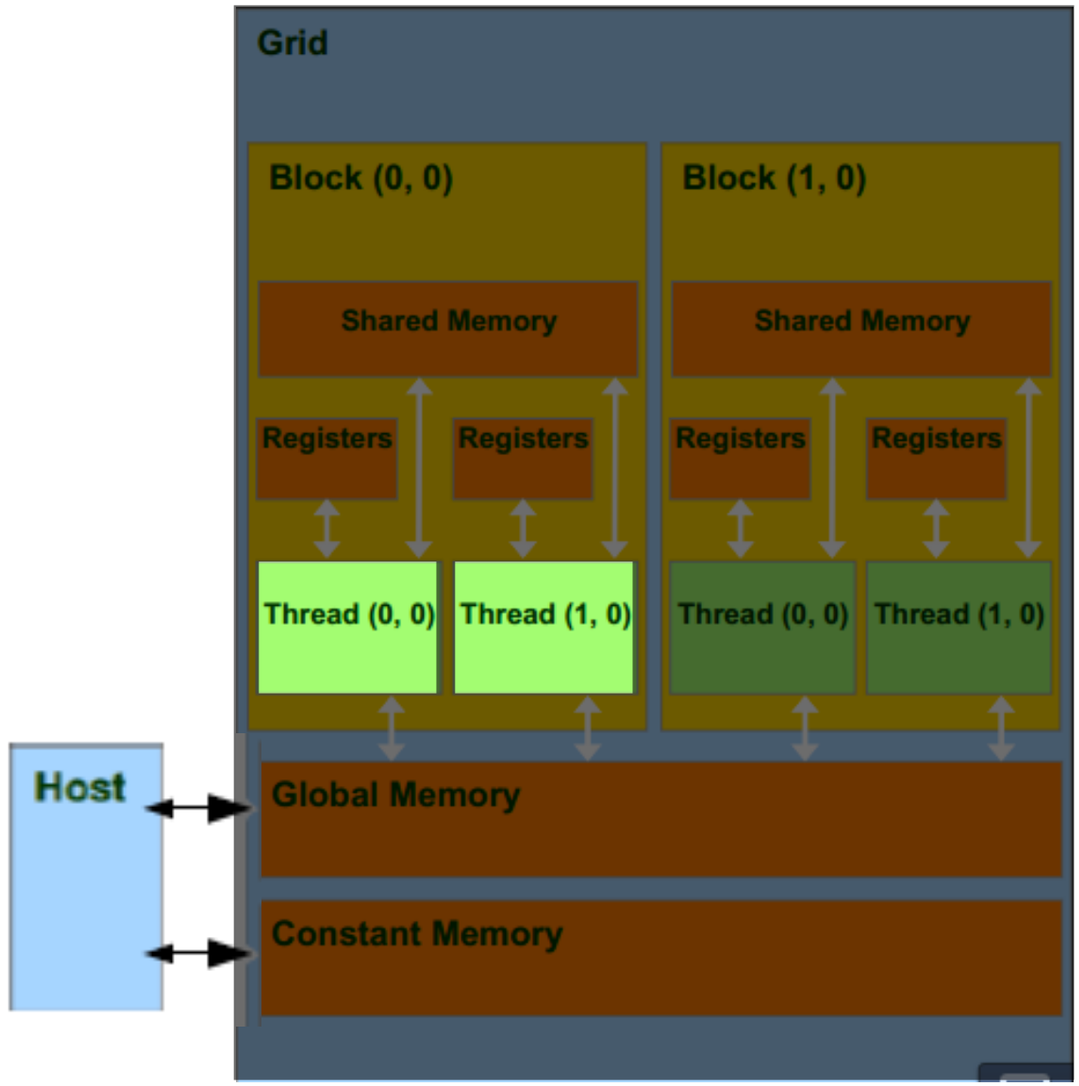




All the clusters of one bin on layer 1 are copied in the shared memory and a for loop starts over these clusters looking for good doublets

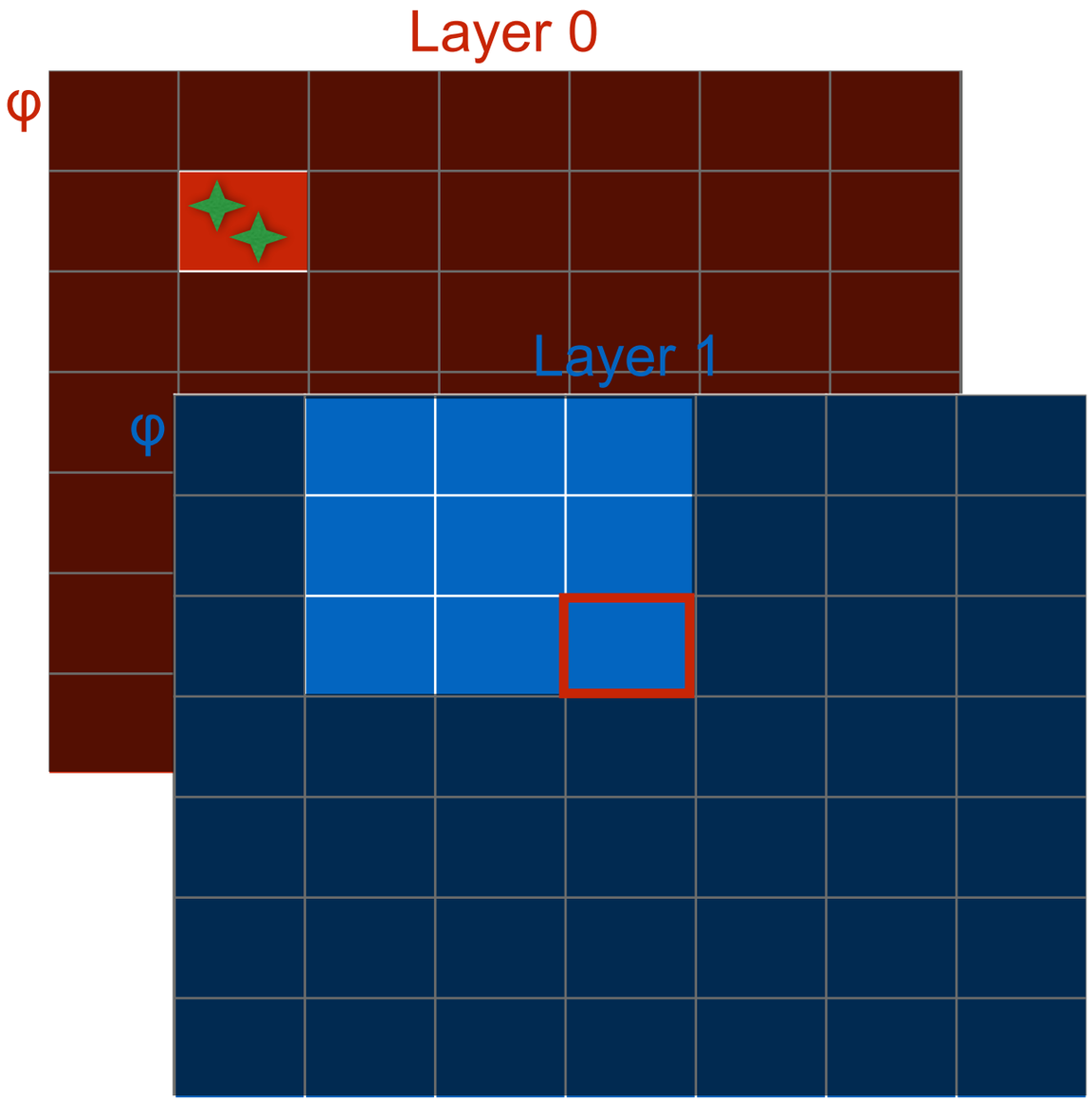
Porting on GPU: how to make doublets

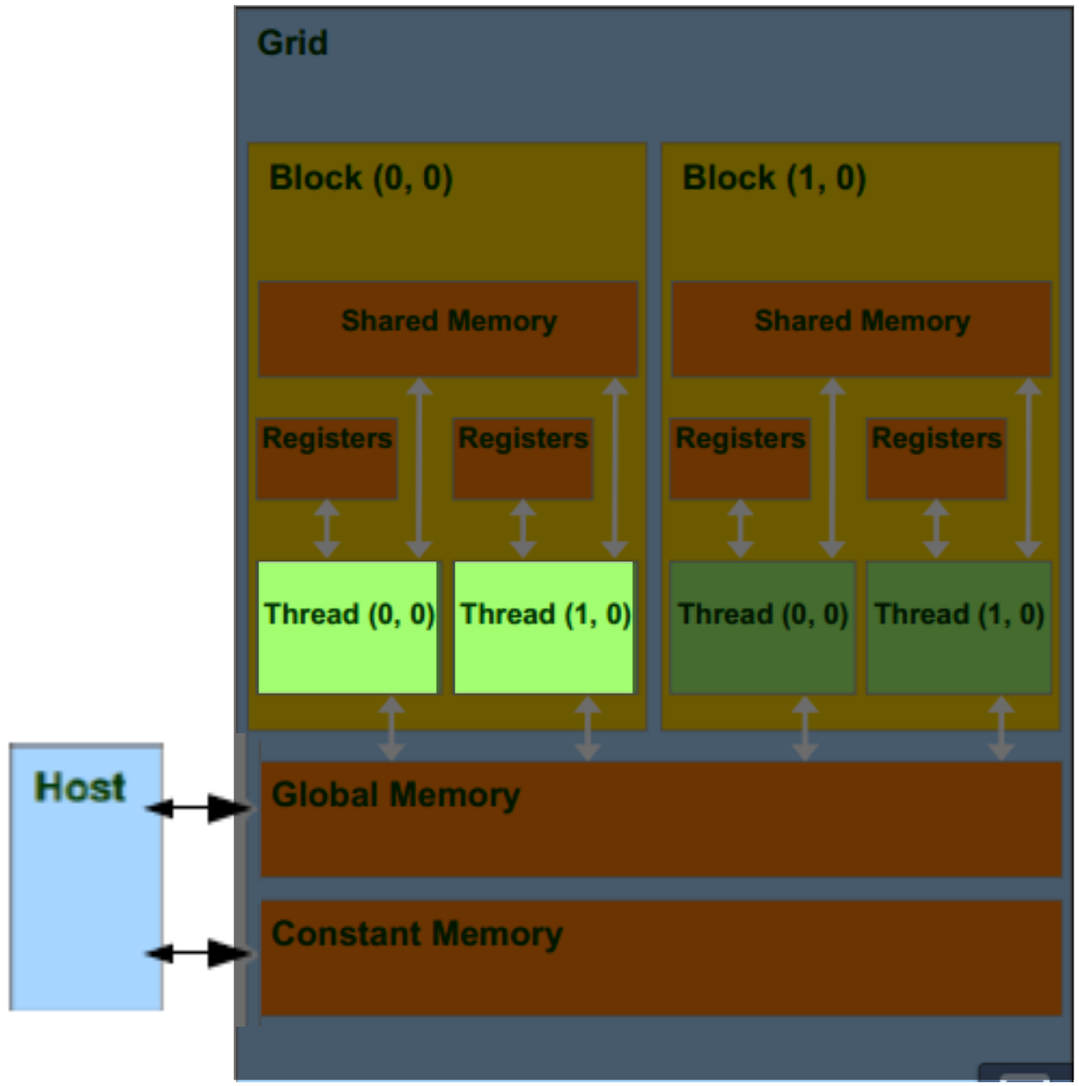




All the clusters of one bin on layer 1 are copied in the shared memory and a for loop starts over these clusters looking for good doublets

Porting on GPU: how to make doublets





All the clusters of one bin on layer 1 are copied in the shared memory and a for loop starts over these clusters looking for good doublets

Conclusions

- Computing in ALICE is still largely based on the intrinsic parallelism offered by the independence of different events.
- The necessity of performing a synchronous data reconstruction online, during data taking triggered a computing approach that exploits the potentials of a heterogeneous architecture.
- This solution is very successful in the present High Level Trigger system.
- In the next LHC Run 3 this approach will be extended to the whole ALICE data reconstruction.
- The availability of CPU+GPU systems also on facilities used for MonteCarlo reconstruction will allow us to use the same computing solution also for offline data processing.