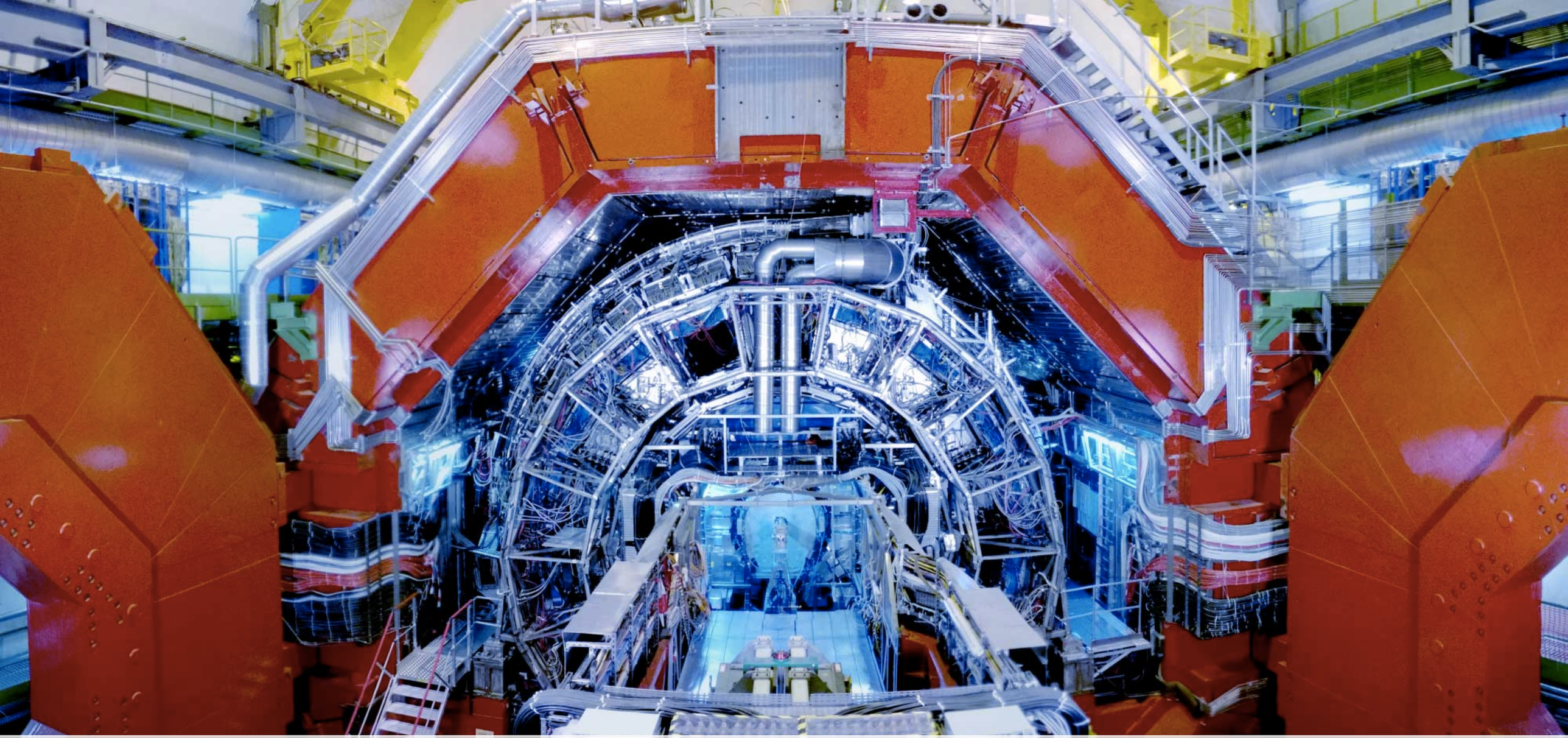


Track reconstruction in high multiplicity events

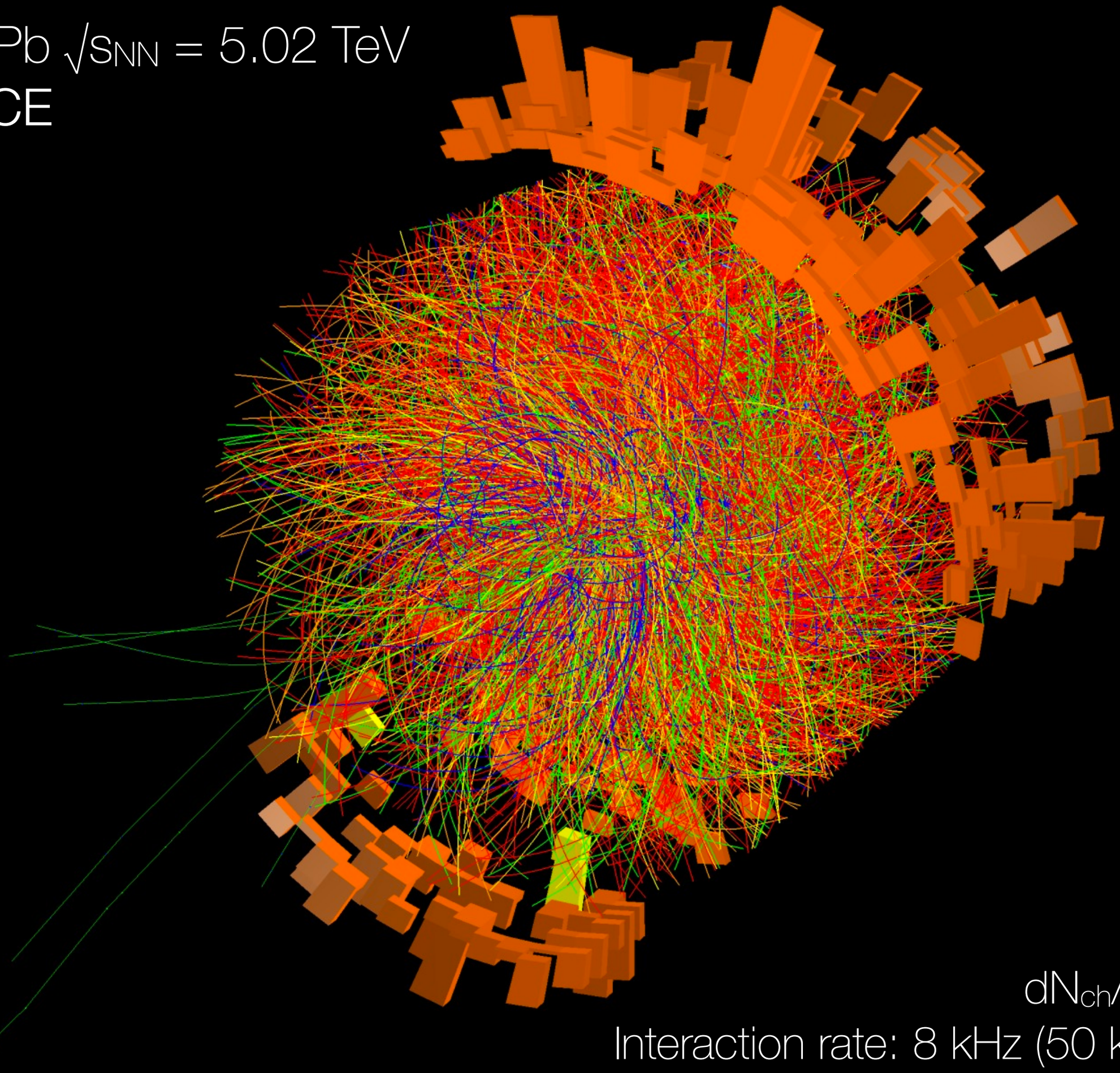
Maximiliano Puccio for the ALICE collaboration

Università & INFN Torino

VERTEX 2016 - La Biodola - September 30th 2016

Pb-Pb $\sqrt{s_{NN}}$ = 5.02 TeV

ALICE



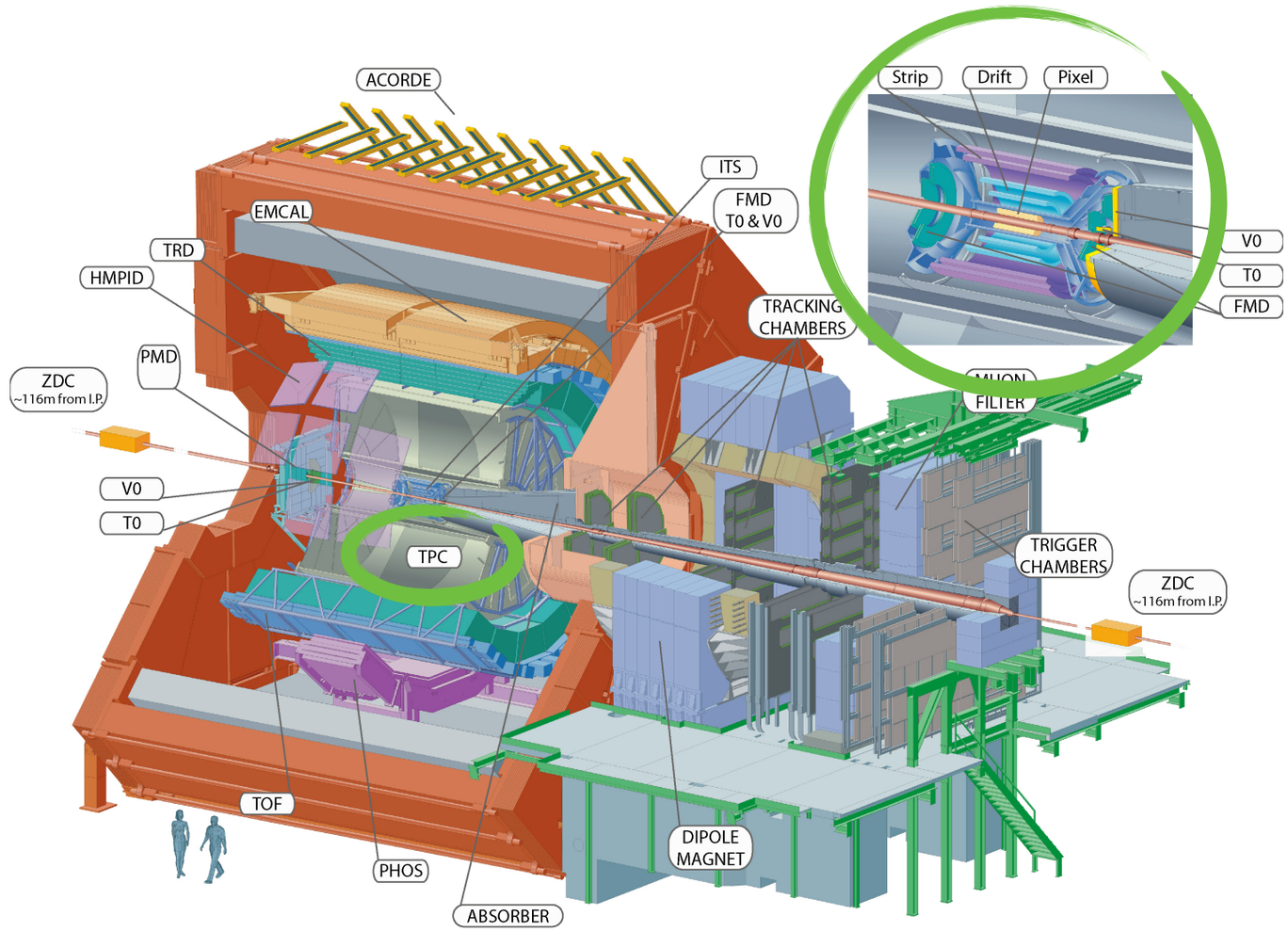
$dN_{ch}/d\eta \sim 2000$

Interaction rate: 8 kHz (50 kHz Run 3)

How to track every charged particle?

CHALLENGES:

- High track density
  - ✓ Highly granular detectors are necessary: up to 50 tracks/cm^2 in the innermost ITS layer
- High combinatorial complexity
  - ✓ Algorithms must reject combinatorics as early as possible



The silicon tracker (ITS) and the Time Projection Chamber provide the required granularity

Challenge for Run3:

- Online data processing (thus *tracking* and *vertexing*) for both ITS and TPC

Current strategy in ALICE

OFFLINE

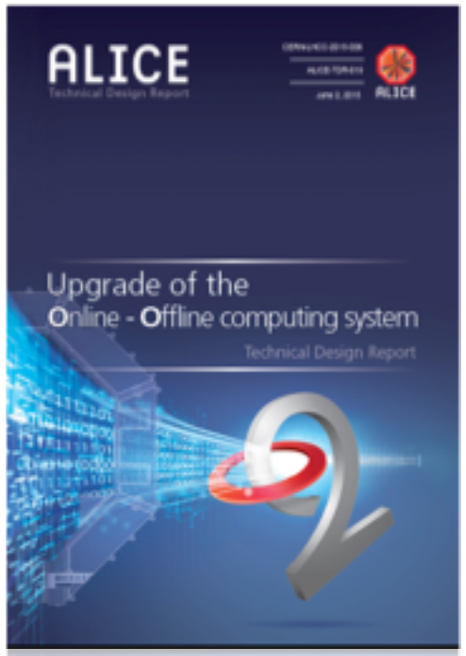
- Tracks are reconstructed in the TPC
- Reconstructed tracks are then propagated to the ITS
  - ITS clusters are picked and the track parameters updated
- A ITS standalone algorithm runs on the clusters not associated to tracks in the previous iteration
- The ITS standalone algorithm is also used in low multiplicity events and on a small sample of Pb-Pb events for dedicated analyses

ONLINE

- A faster version of the TPC tracking based on Cellular Automata is run in the High Level Trigger (HLT)
- Tracks found in the HLT can be used as seed in the offline reconstruction and for online calibration of the drift velocity

Future strategy: O^2 project in ALICE

- Starting from Run3 we expect Pb-Pb collisions at peak rate of 50kHz
  - 1.1 TB/s of data will be collected
- Data volume must be reduced before writing to tape
  - Online processing is the only option



O^2 TDR

O^2 project integrates in a single infrastructure DAQ, HLT and Offline reconstruction systems to provide online processing of the collected data

Infrastructure specs:

- ~250 **F**irst **L**evel **P**rocessors worker nodes equipped with FPGA
- ~1500 **E**vent **P**rocessing **N**odes worker nodes equipped with GPU
- Data collected (2021, 2022): 54 PB/yr

Detector	< Event size >	Pb-Pb@50kHz
TPC	20.7 MB	1012 GB/s
ITS	0.8 MB	40 GB/s
TRD	0.5 MB	20 GB/s
MFT	0.2 MB	10 GB/s
Others	0.3 MB	12.2 GB/s

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 particle tracks
  - data format optimisation
- Largely based on the present HLT results

Online calibration and tracking are mandatory

Still uncertainties for the ITS Upgrade:

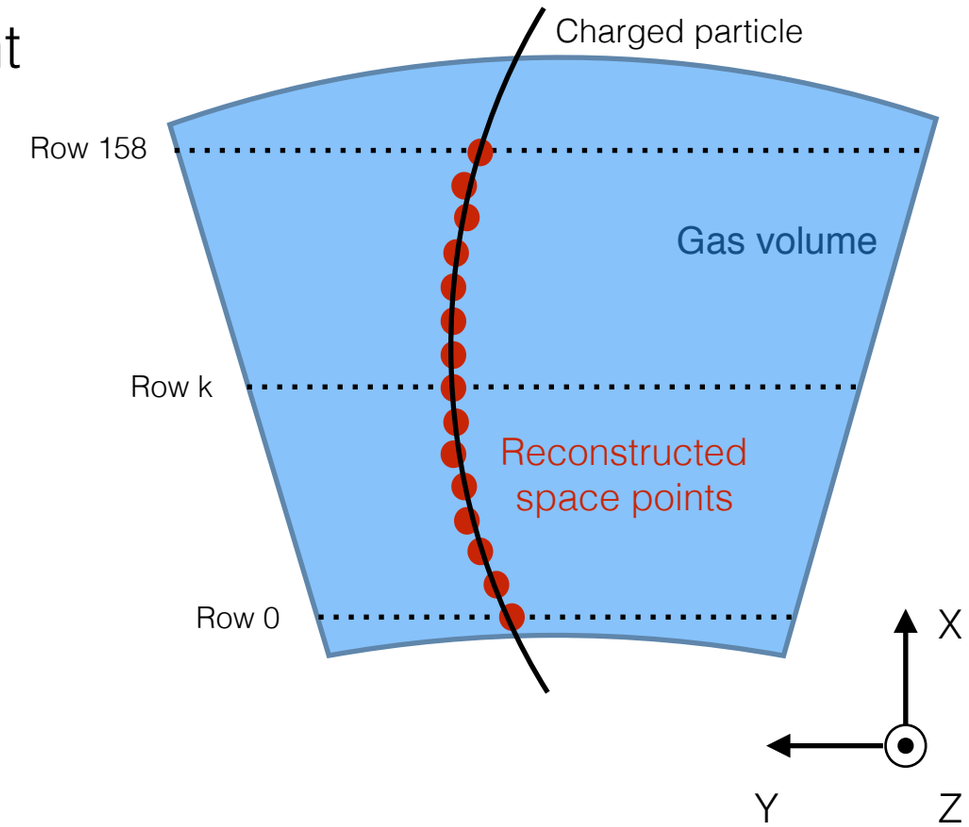
- The contribution from noisy clusters is unknown: here a very pessimistic estimate of a probability of 10^{-5} per pixel has been made (as was stated in the ITS Upgrade TDR)
- If full synchronous reconstruction will be feasible a higher reduction factor will be achieved (noise removal)

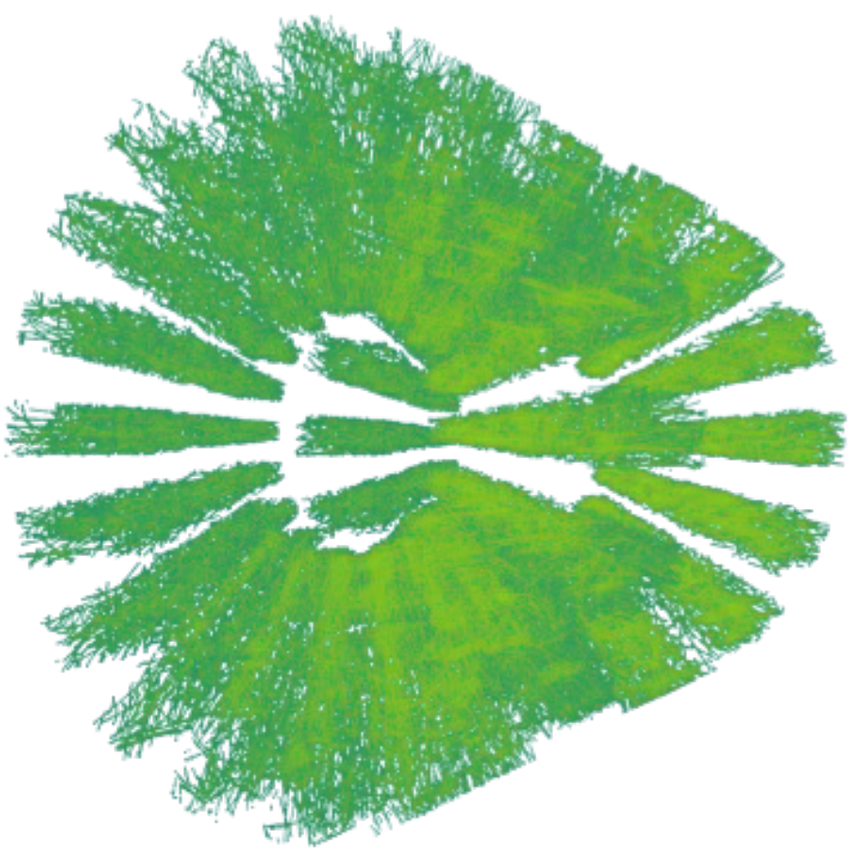
Detector	Data rate for Pb-Pb @ 50 kHz	Compressed data rate	Data reduction factor
TPC	1012 (GB/s)	50 (GB/s)	20.2
ITS	40 (GB/s)	26 (8) (GB/s)	1.5 (5)
TRD	20 (GB/s)	3 (GB/s)	6.7
MFT	10 (GB/s)	5 (GB/s)	2
Total	**1082 (GB/s)**	**84 (66) (GB/s)**	**12.9 (16.4)**

Online tracking in the TPC

TPC tracking with the HLT is currently the most relevant example of High Performance Computing 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.
- Track are then merged and refitted using Kalman Filter





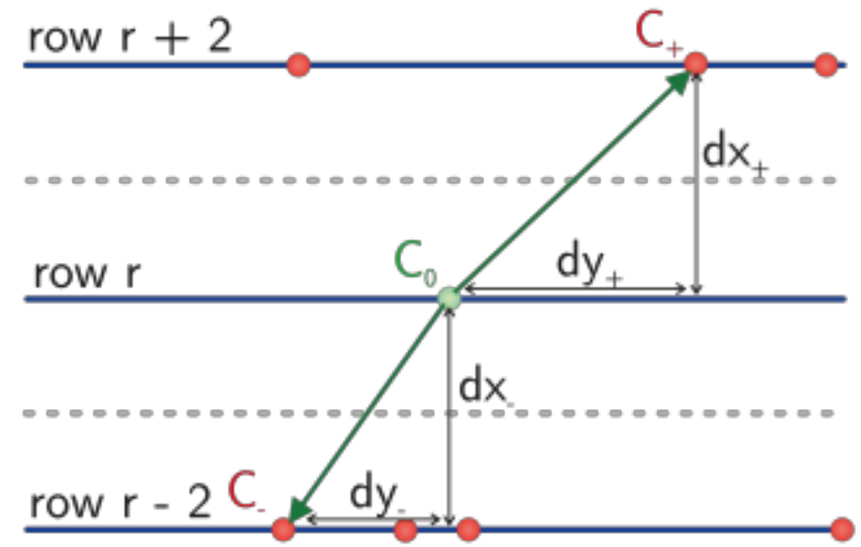
The radial and azimuthal coordinates of the clusters are measured by charge collection in 159 rows.

- Inner radius: 85 cm
- Outer radius: 250 cm
- The coordinate along the beam axis is measured via the drift time

Cellular Automaton in the HLT

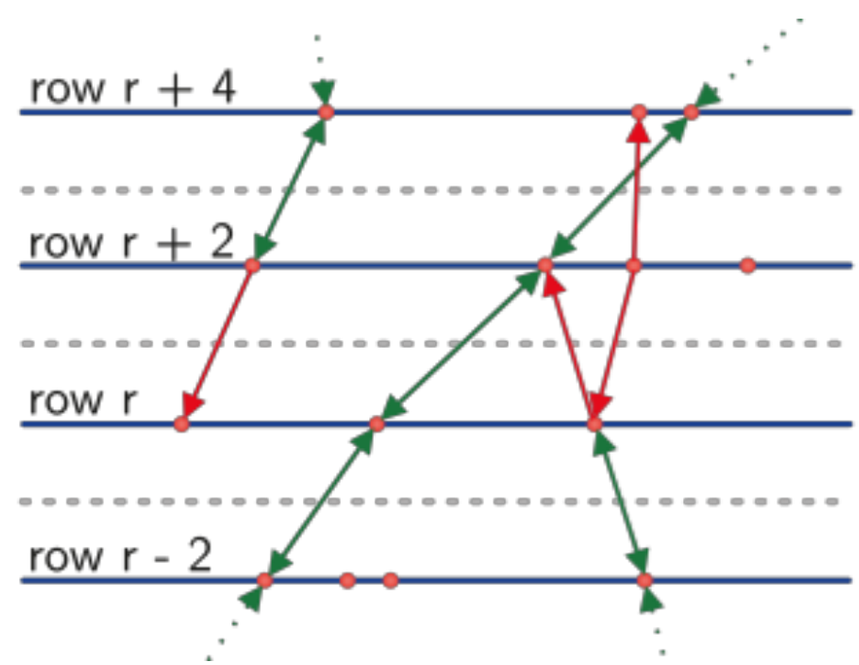
Neighbours finder

- For each hit at row k, the best pair of neighbouring hits from row k+2 and k-2 is found (best=straight line)
- The hit-to-hit links (3-points straight lines) are determined and saved



Tracklet reconstruction

- Track segments (tracklets) are created following hit-to-hit links
- A simplified Kalman filter is used to fit geometrical trajectories



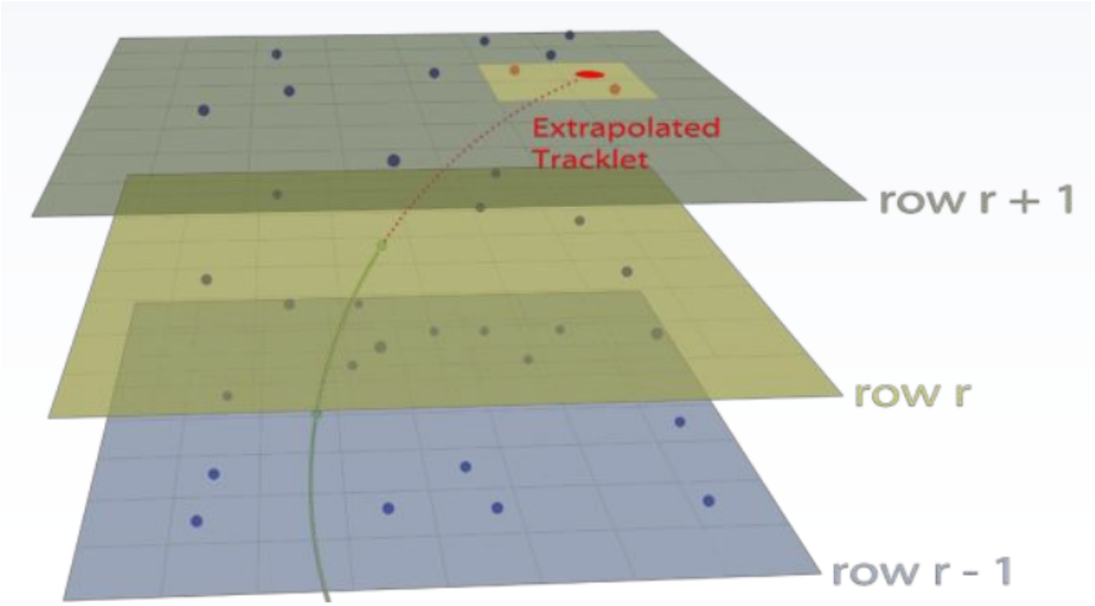
Tracklet selection

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

Track merging and fitting

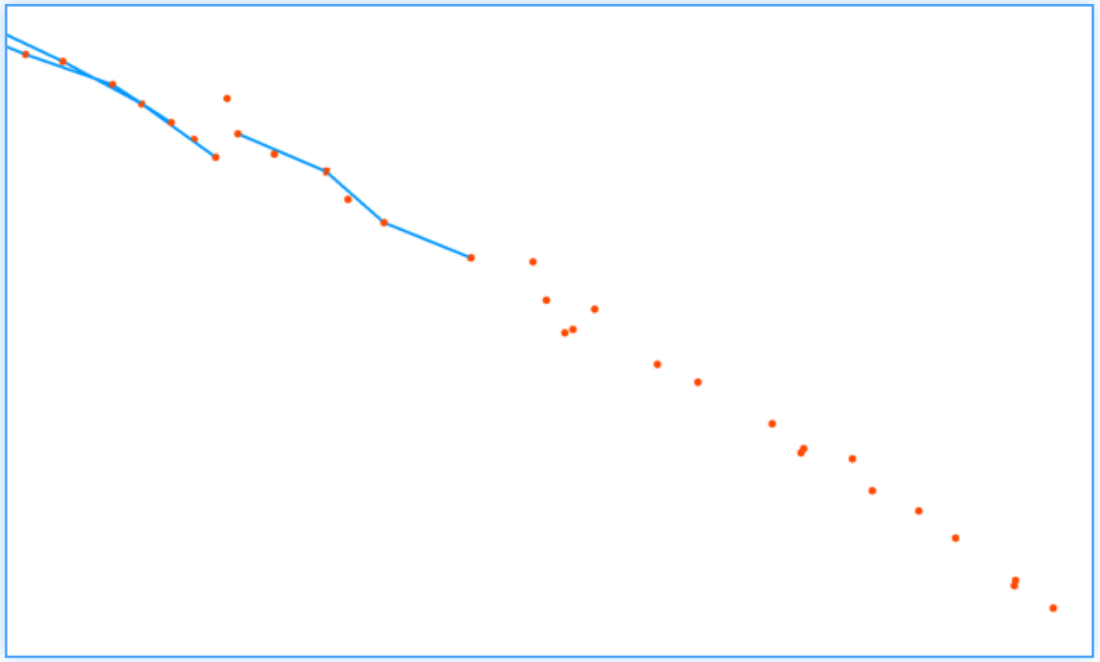
Track following

- Track parameters are fit to the seed
- The trajectory is extrapolated to adjacent TPC row
- Cluster closest to extrapolated position is found and the fit is improved with the new cluster.



Track merging

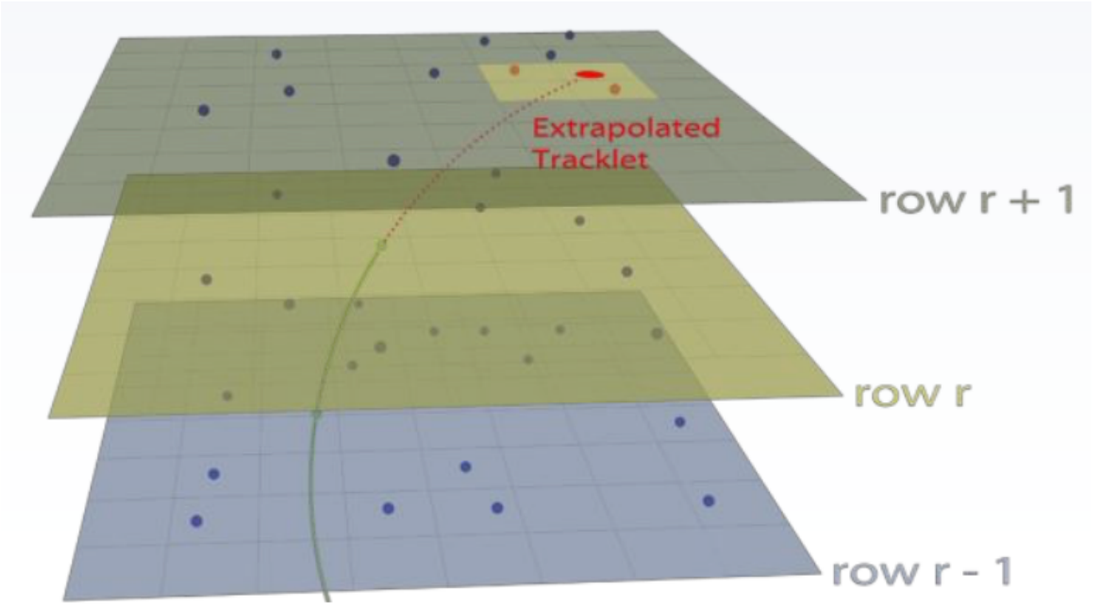
- All the track segments found in the different sectors are then combined and merged if their parameters are compatible.



Track merging and fitting

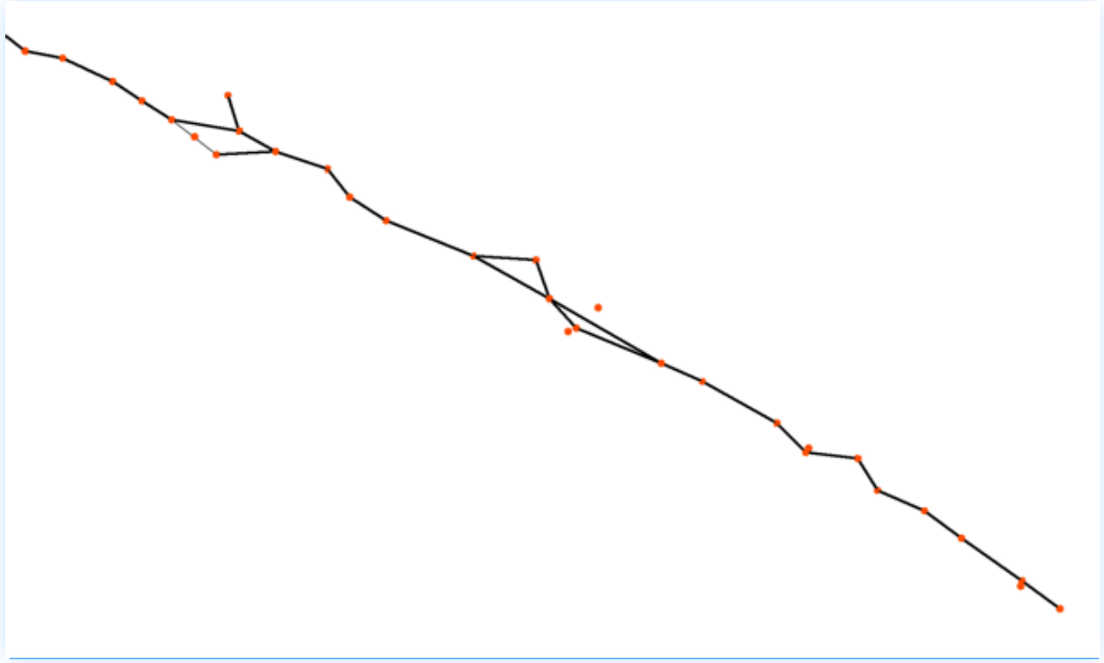
Track following

- Track parameters are fit to the seed
- The trajectory is extrapolated to adjacent TPC row
- Cluster closest to extrapolated position is found and the fit is improved with the new cluster.



Track merging

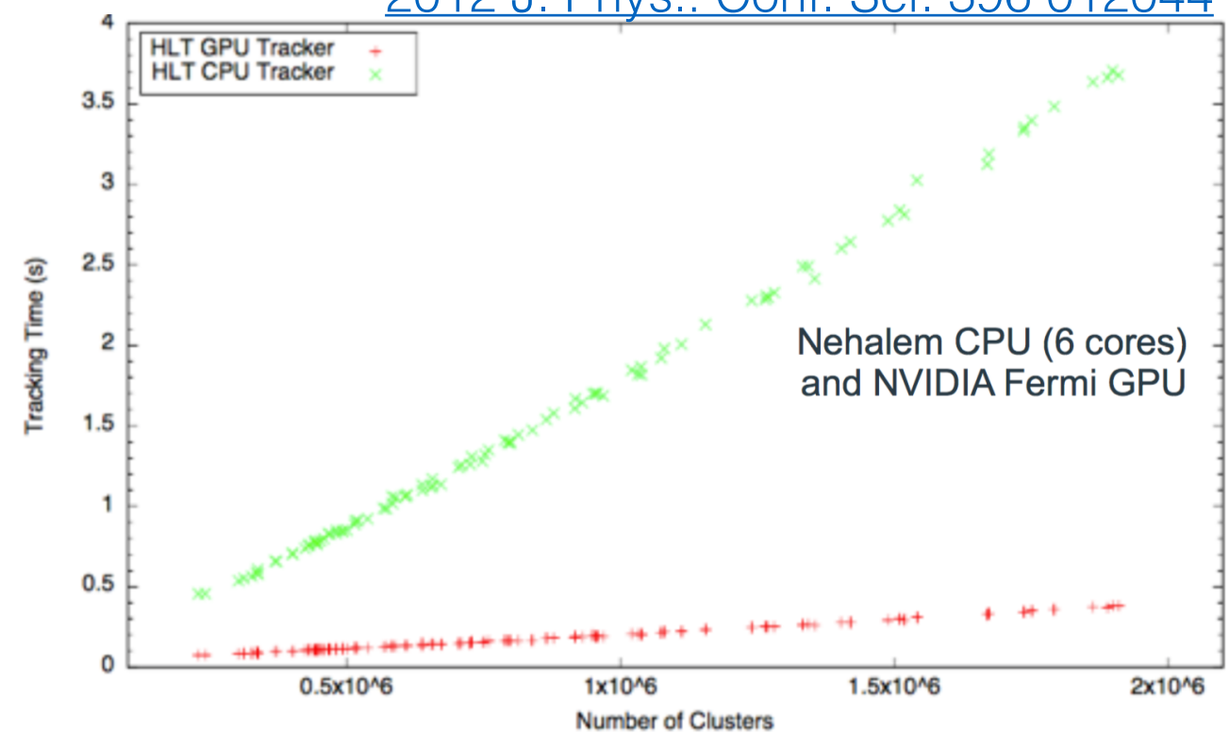
- All the track segments found in the different sectors are then combined and merged if their parameters are compatible.



TPC tracking performance

- GPU tracker is about 10 times faster w.r.t. the CPU version
- Track merging and fitting is done on CPU (no gain on GPU)
- The whole tracking is done with 1 GPU +3 CPU cores
- Performance of 1GPU+3 CPU cores ~ 27 CPU cores

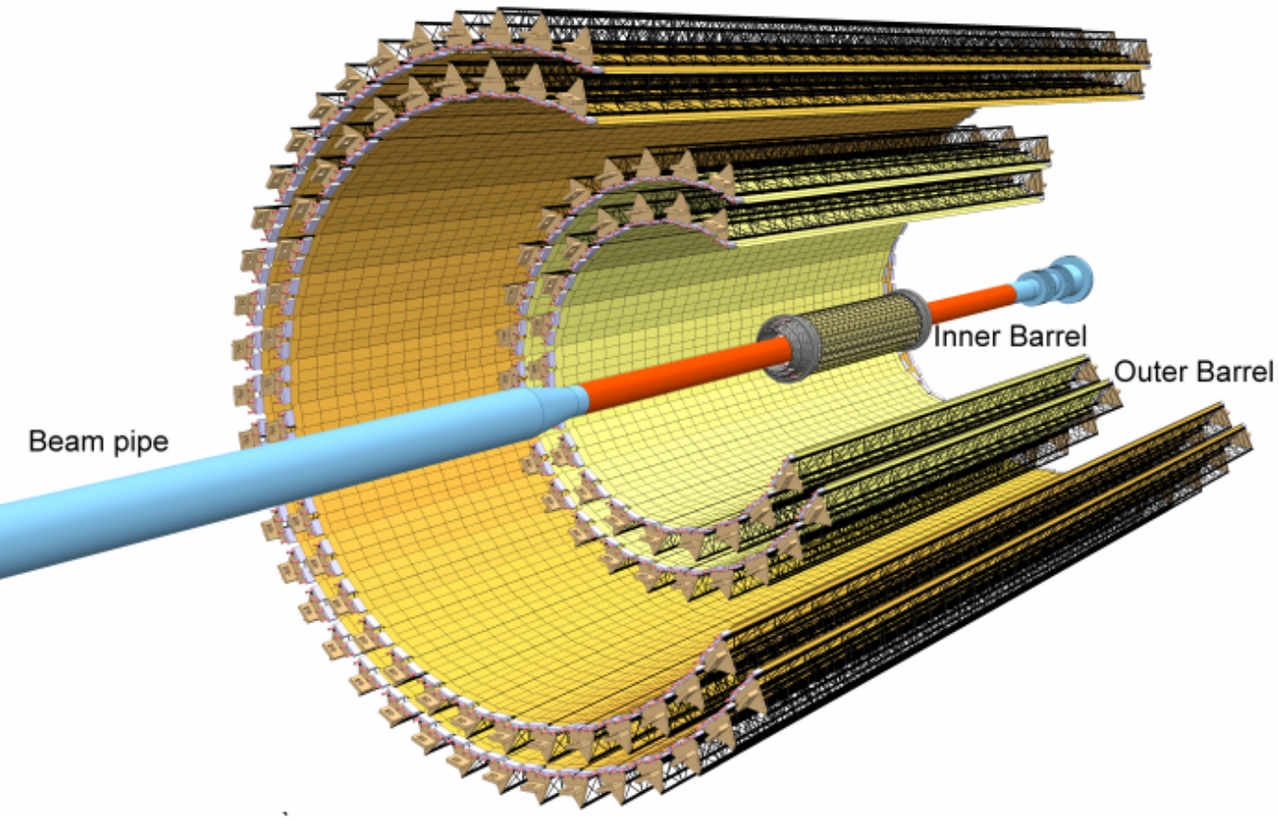
2012 J. Phys.: Conf. Ser. 396 012044



Steps	Method	Time Fraction	Where
Seeding	Cellular Automaton	~30%	CPU or GPU
Track following	Kalman Filter	~60%	
Track merging	Combinatorics	~2%	CPU
Track fit	Kalman Filter	~8%	

ITS Upgrade

Talk by S. Beolè on Tuesday



Feature	ITS	ITS Upgrade
# Layers	6	7
Acceptance	$\|\eta\|<0.9$	$\|\eta\|<1.5$
Material budget/layer	1.1% X_0	0.3-1% X_0
Spatial resolution	12 x 100 μm² 35 x 20 μm² 20 x 830 μm²	5 x 5 μm²
Max Pb-Pb readout rate	1 kHz	100 kHz
Technology	Hybrid Silicon Pixels, Silicon drift detectors, Silicon strip	Monolithic Active Pixel Sensors (MAPS)

UPGRADE

- ✓ Better pointing resolution by a factor of 3(5) in rϕ (z) at p_T =500 MeV/c
  - innermost layer is closer to the IP: 39 mm → 22 mm
  - less material budget: ~0.3% X_0 for the 3 inner barrel, 1% X_0 for the outer.
  - reduced pixel size: from 50×425 μm² → ~(30×30 μm²)
- ✓ Better tracking efficiency and p_T resolution at low p_T thanks to the higher resolution and to the additional layer
- ✓ Faster readout
- ✓ Accessible for maintenance during winter shutdowns

Tracking in the ITS Upgrade

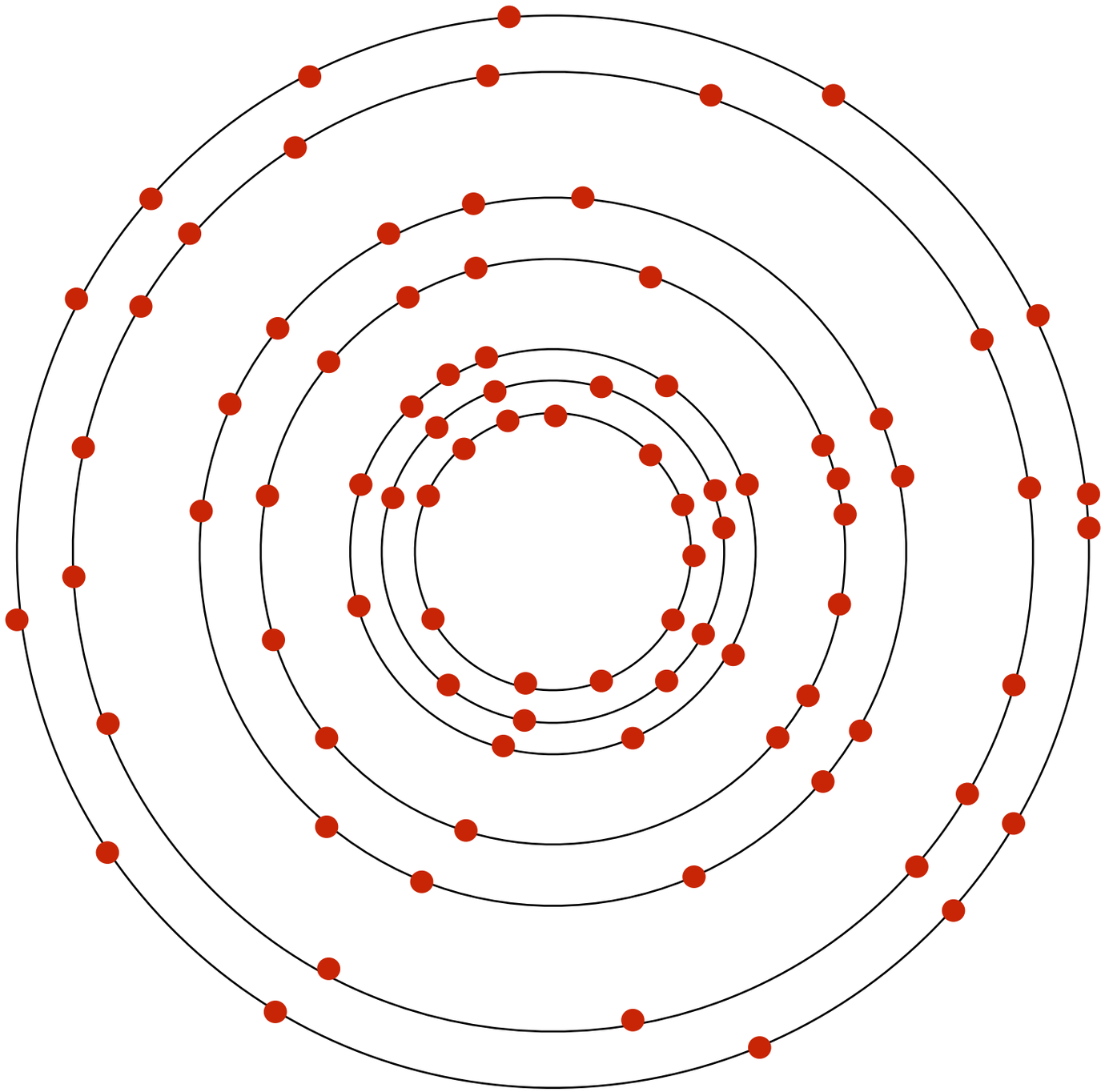
REQUIREMENTS

- TPC tracks can be prolonged inwards to the ITS
  - ➡ However distortions due to space charge will affect the standalone tracking in the TPC
- The ITS can be used as a standalone detector and tracks found in the ITS can be prolonged to the TPC.
- The ITS tracks will be used for the TPC calibration to correct for the distortions
  - ✓ Method already in use offline for the Run2 and working!
  - ✓ In Run3 the same method can be used to calibrate online the TPC
- A ITS standalone tracker based on a Cellular Automaton has been coded and tested on CPU, within the present ALICE offline framework.

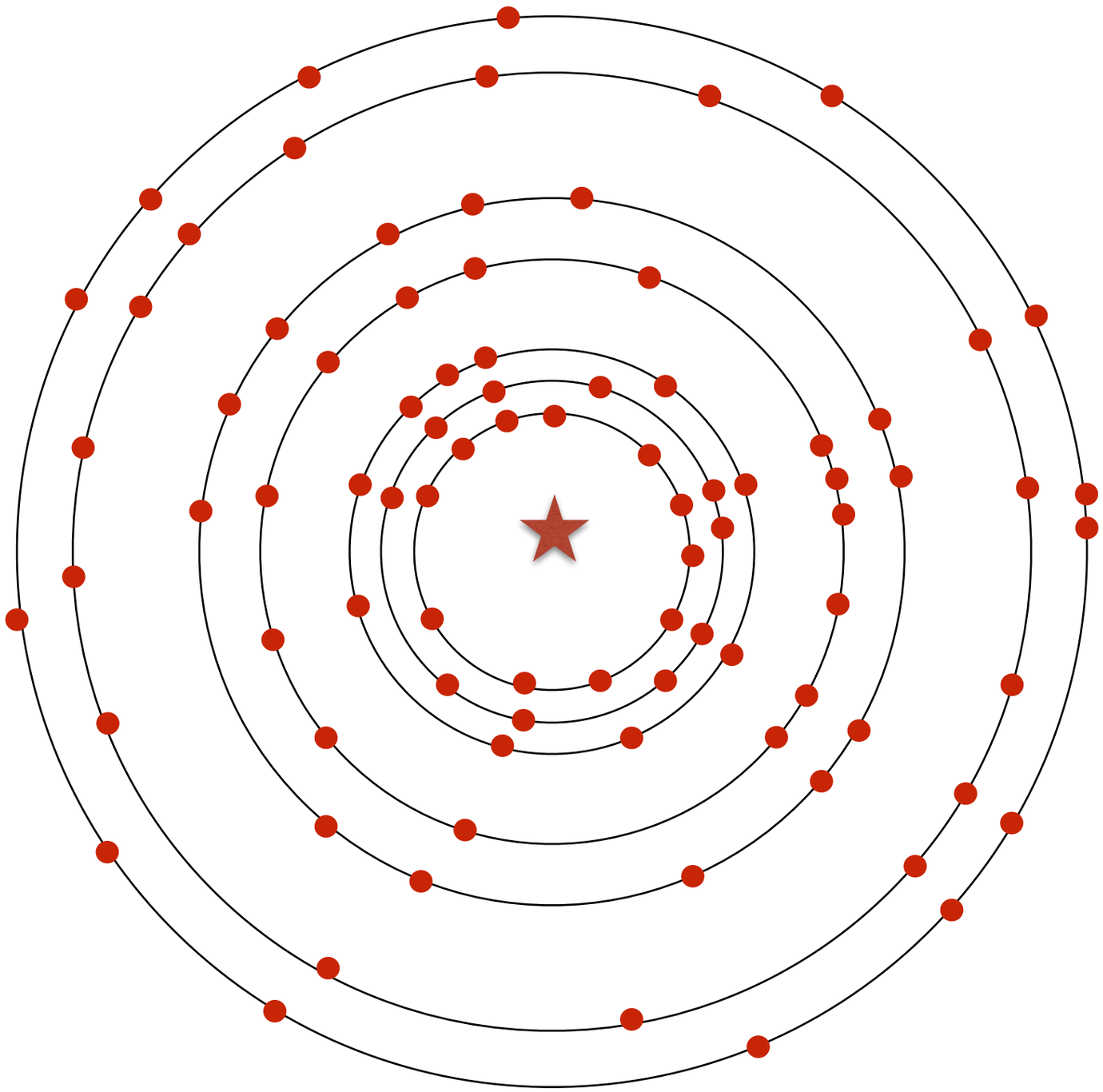
Tracking in ITS Upgrade

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

Tracking in ITS Upgrade

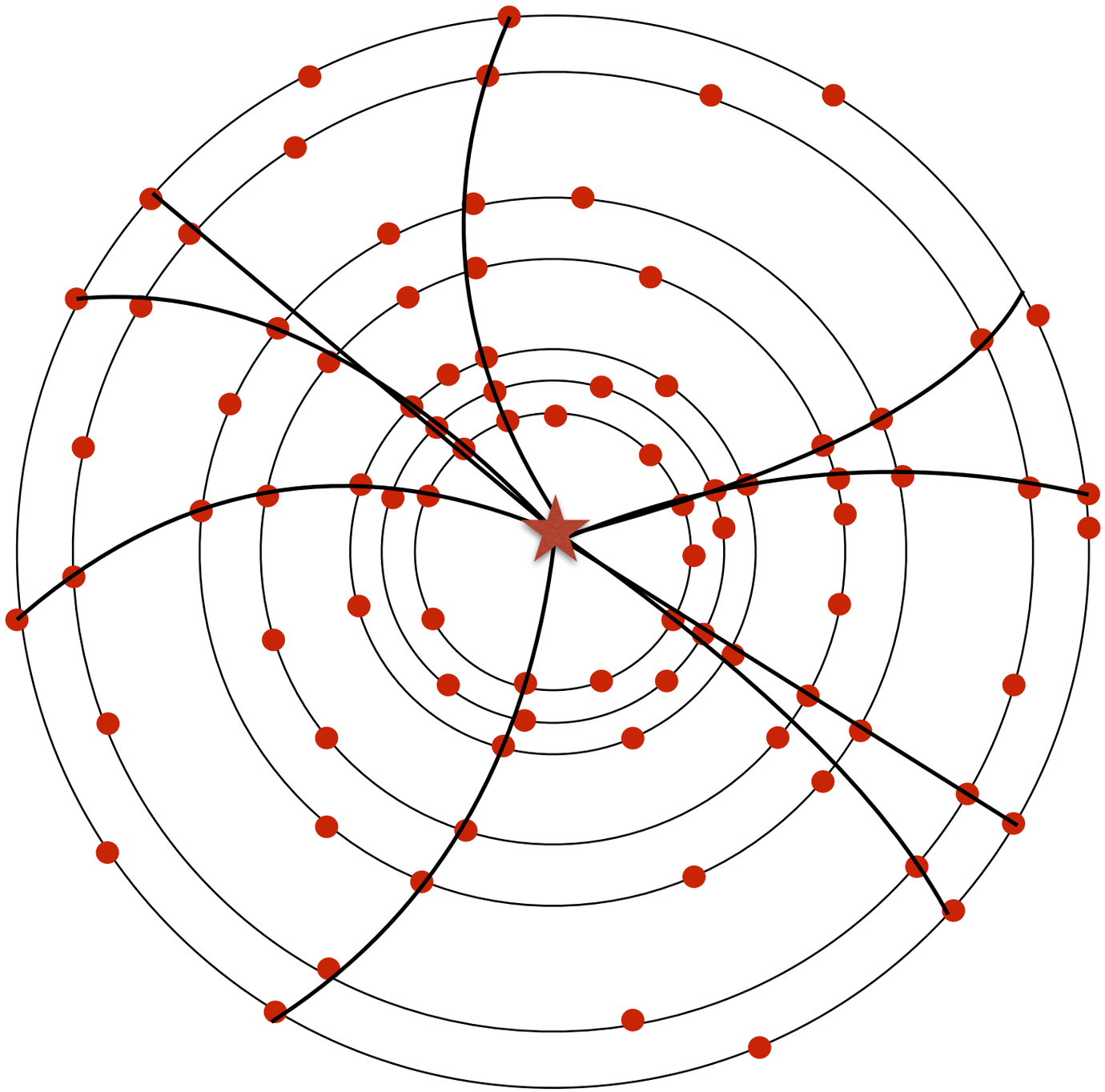


Tracking in ITS Upgrade



Determination of the primary vertex with fast algorithms

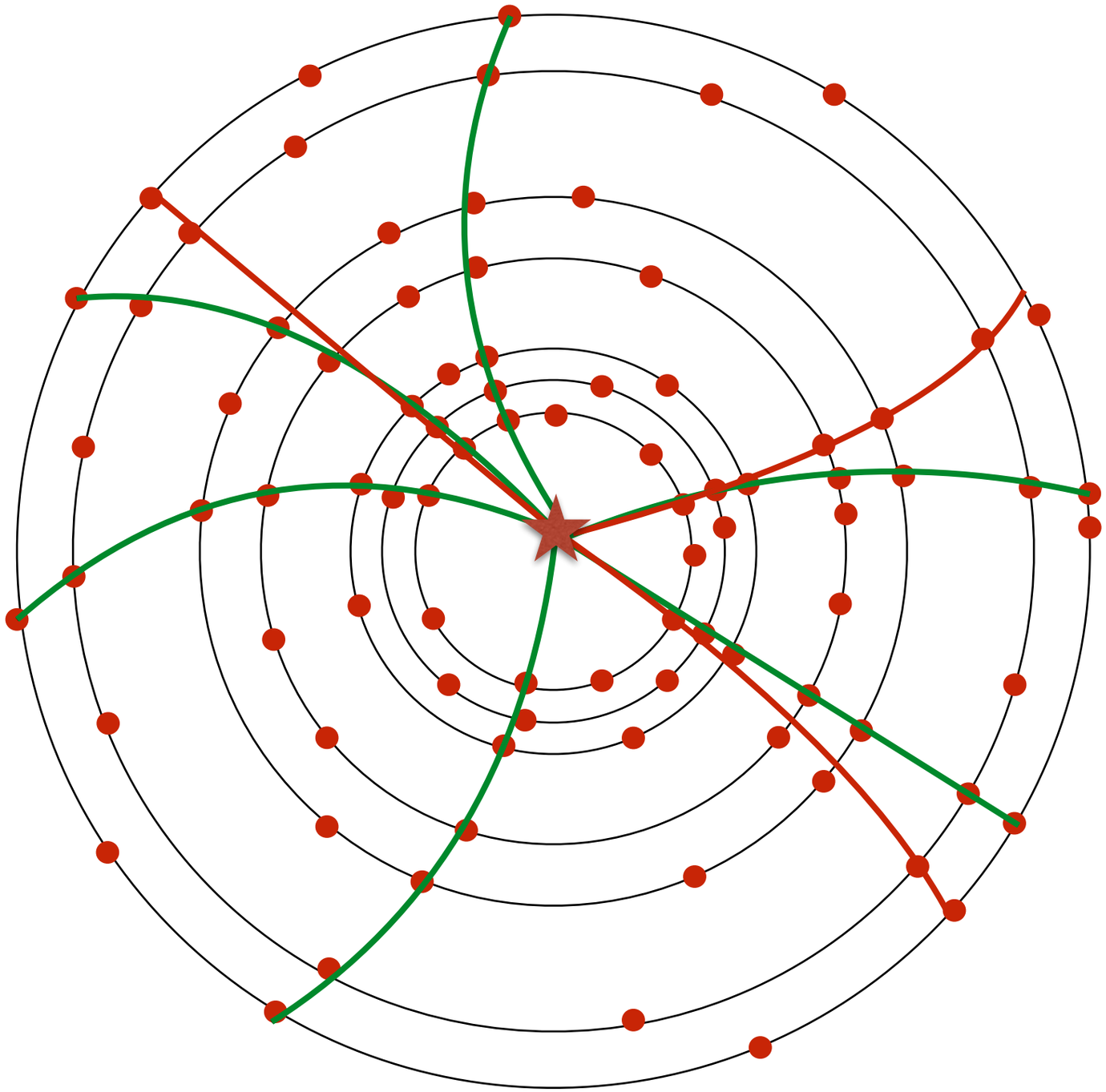
Tracking in ITS Upgrade



Determination of the primary vertex with fast algorithms

Using a pattern recognition method, find track candidates

Tracking in ITS Upgrade

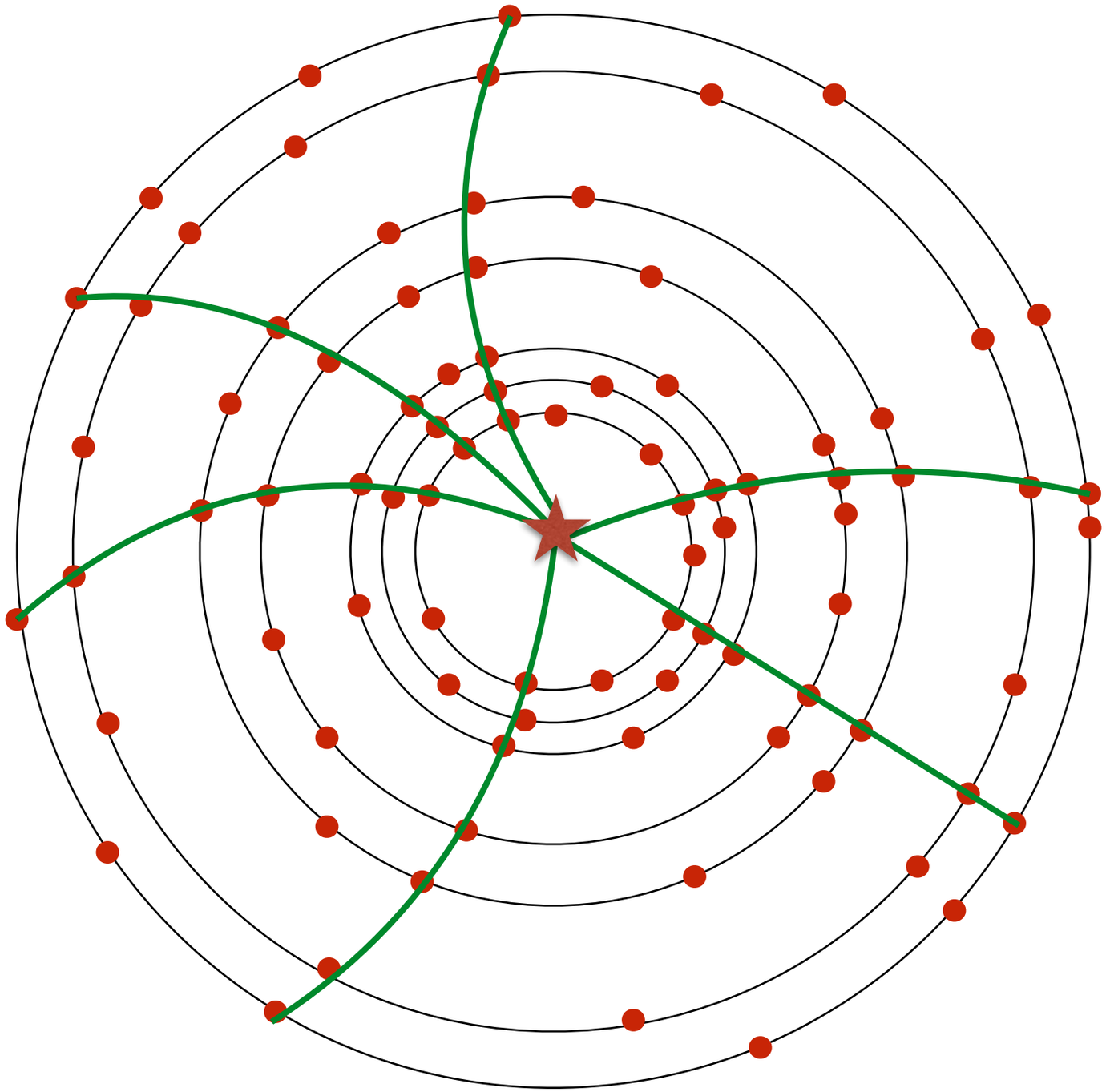


Determination of the primary vertex with fast algorithms

Using a pattern recognition method, find track candidates

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

Tracking in ITS Upgrade



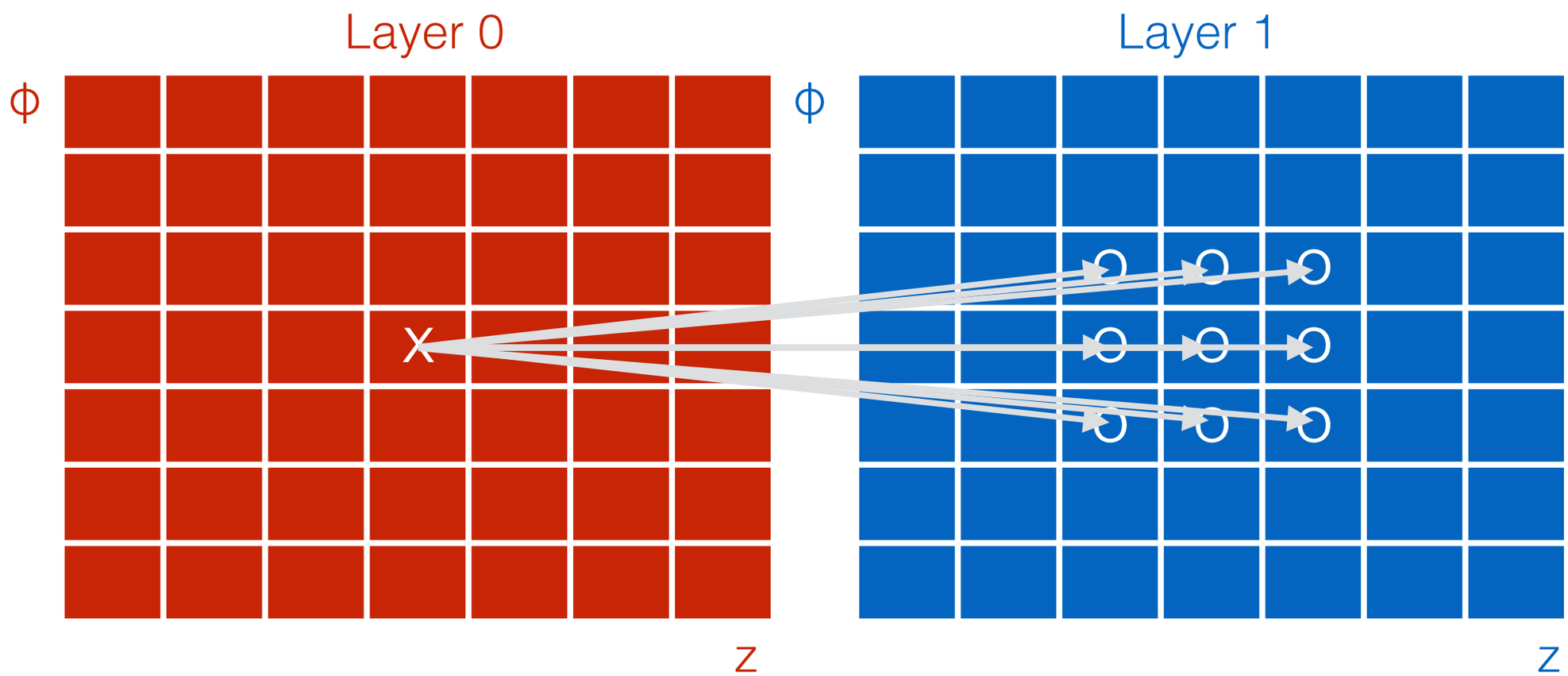
Determination of the primary vertex with fast algorithms

Using a pattern recognition method, find track candidates

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

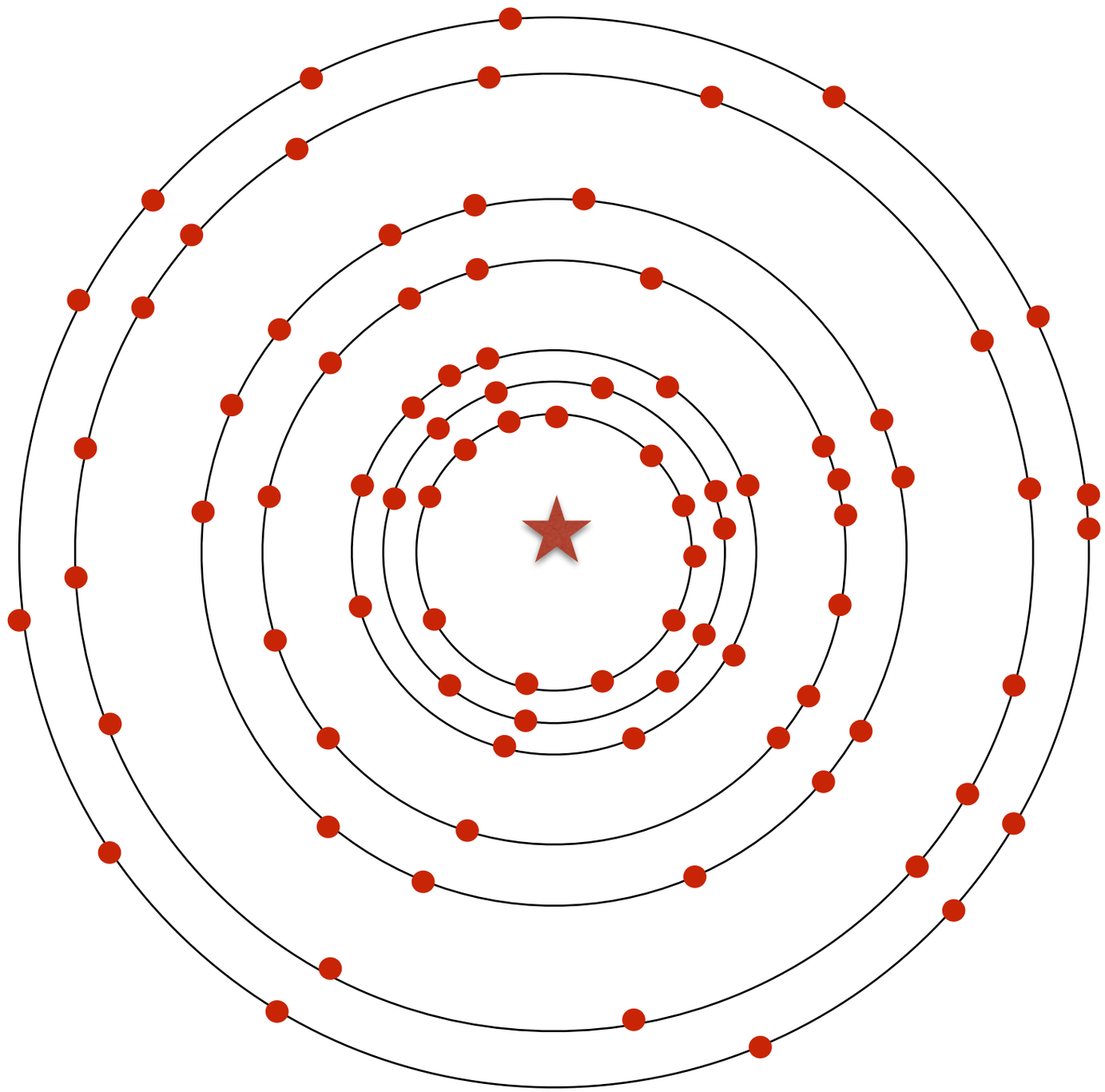
Candidates with the best χ^2 are saved as tracks

Data layout

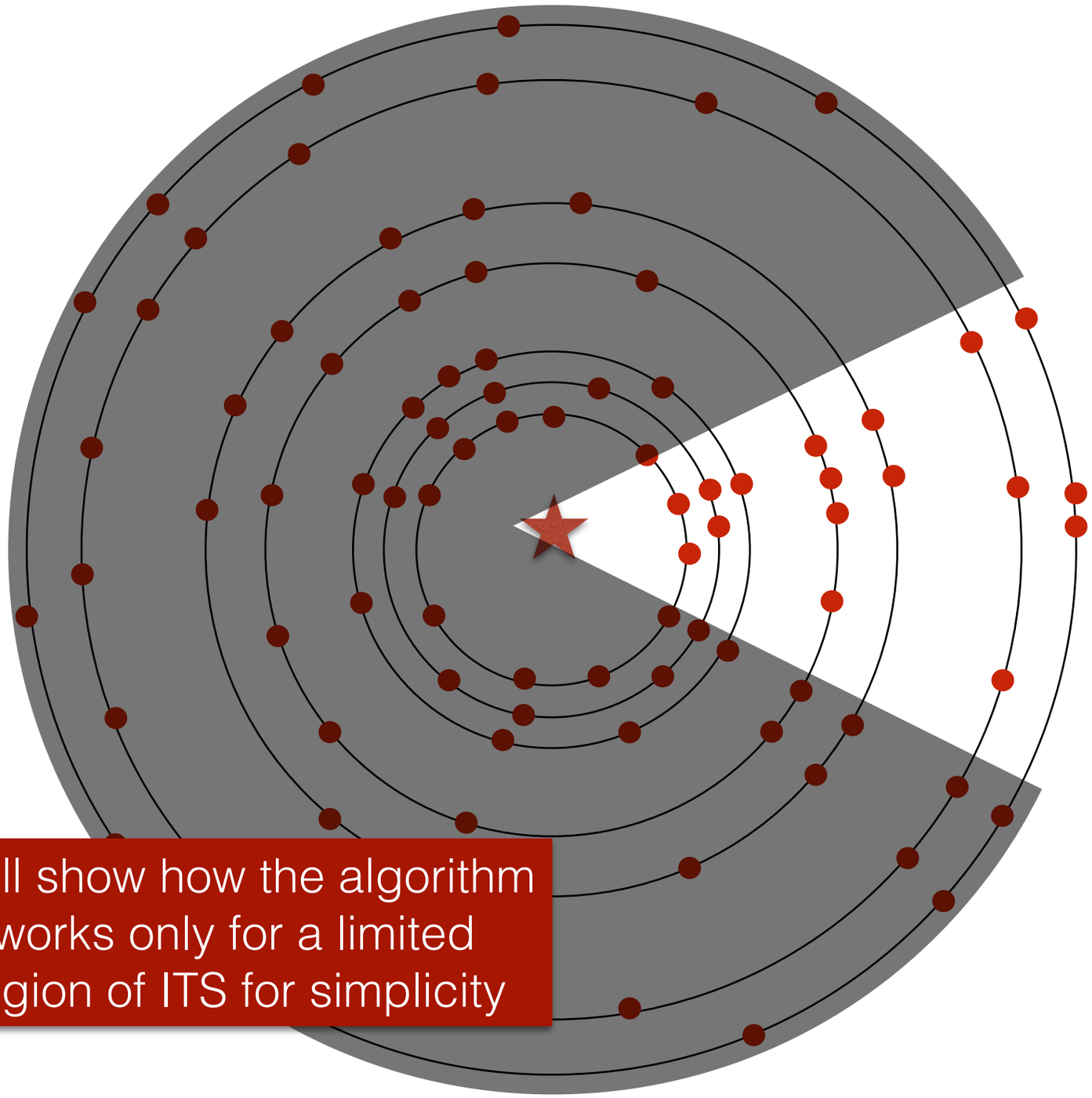


- Hits are sorted and stored according their azimuthal angle and their z coordinate
- An index table is filled to quickly fetch the hits in the region of interest of the detector
  - ✓ Increase data locality
  - ✓ Possible parallelisation

Cellular Automata in the ITS Upgrade

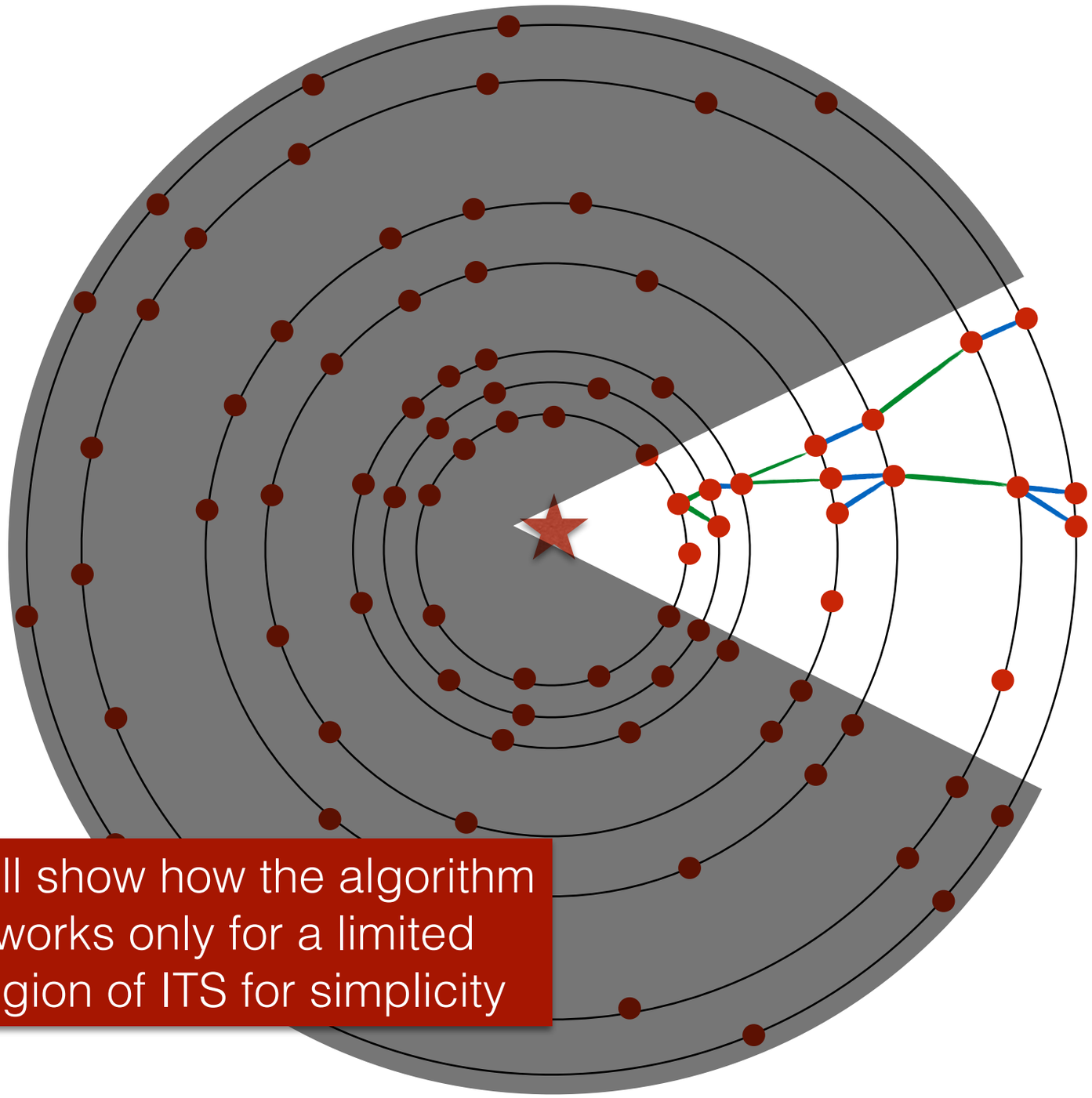


Cellular Automata in the ITS Upgrade



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

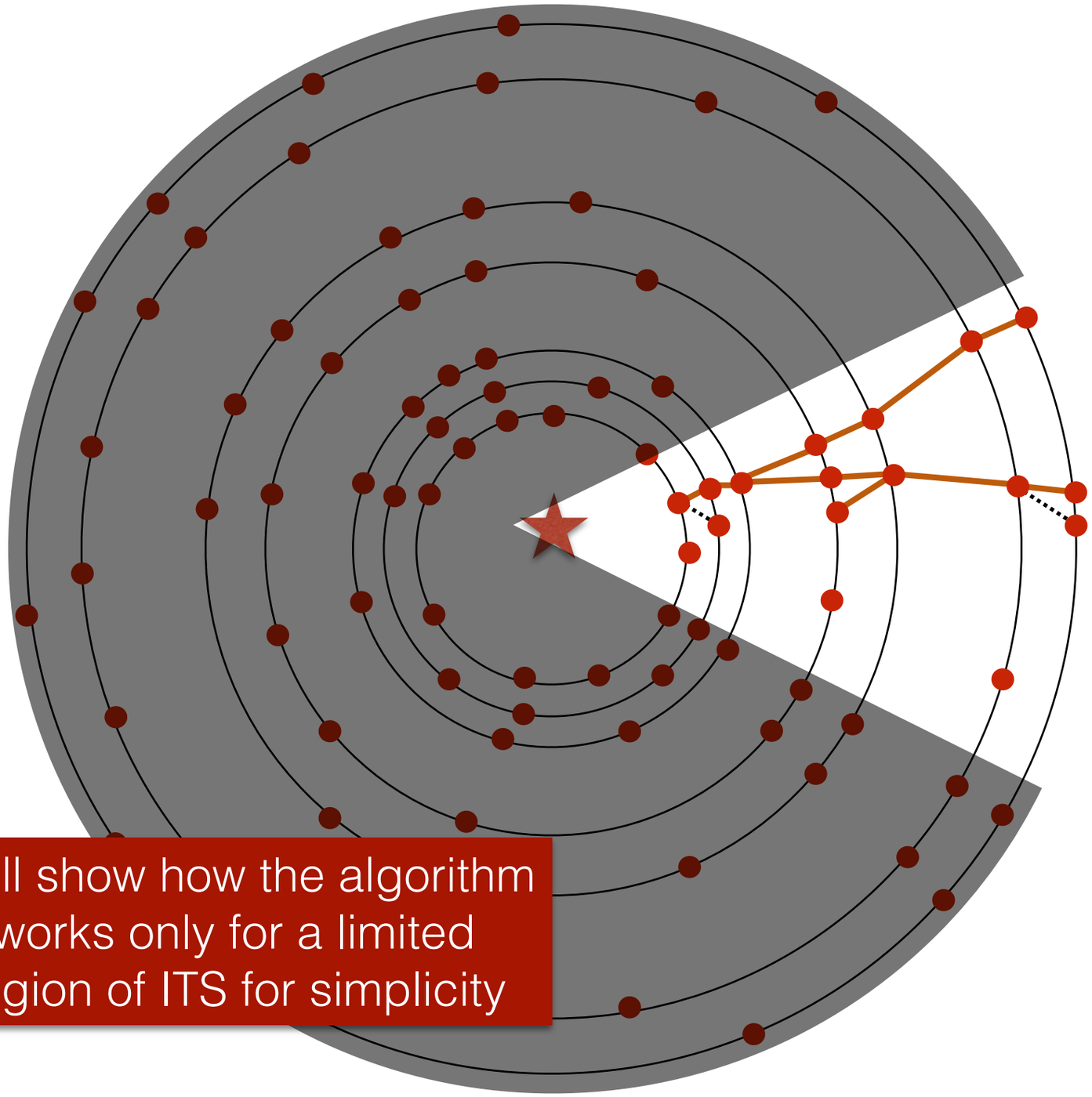
Cellular Automata in the ITS Upgrade



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

Cellular Automata in the ITS Upgrade

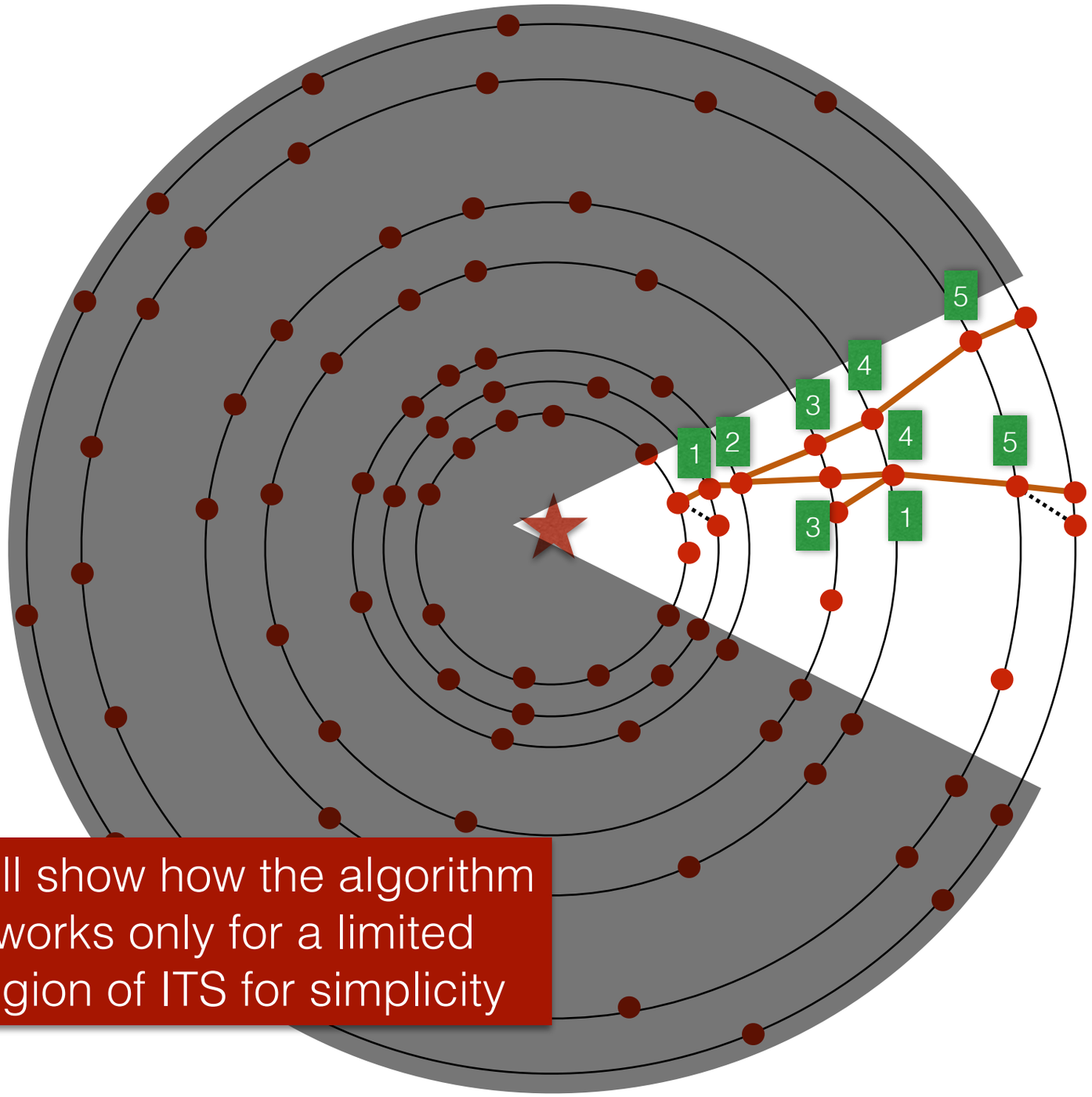


For each cluster on each layer a 2D window is opened. Then the clusters are joined with those on the next layer within the window

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

Cellular Automata in the ITS Upgrade



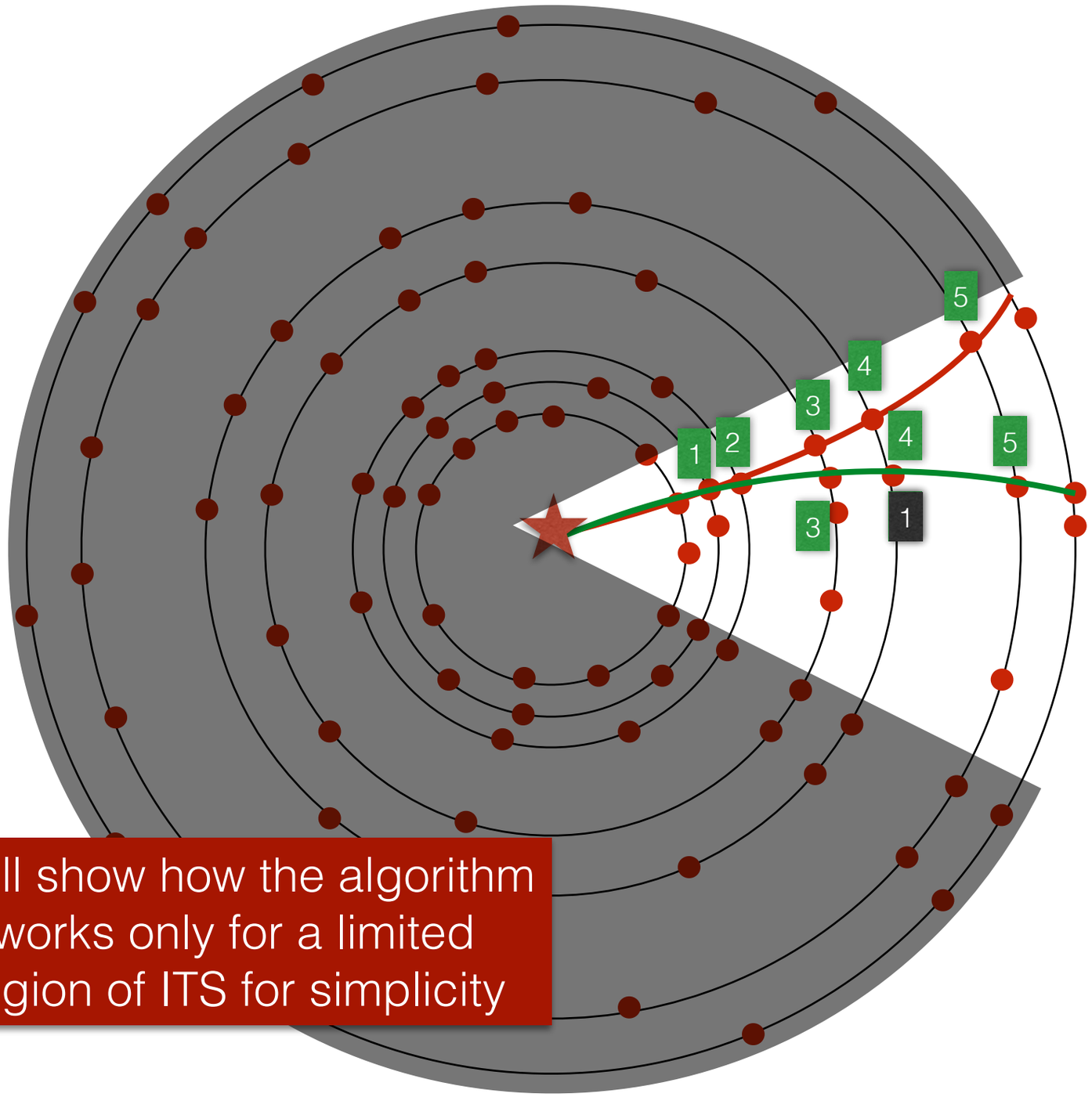
For each cluster on each layer a 2D window is opened. Then the clusters are joined with those on the next layer within the window

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

Cellular Automata in the ITS Upgrade



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

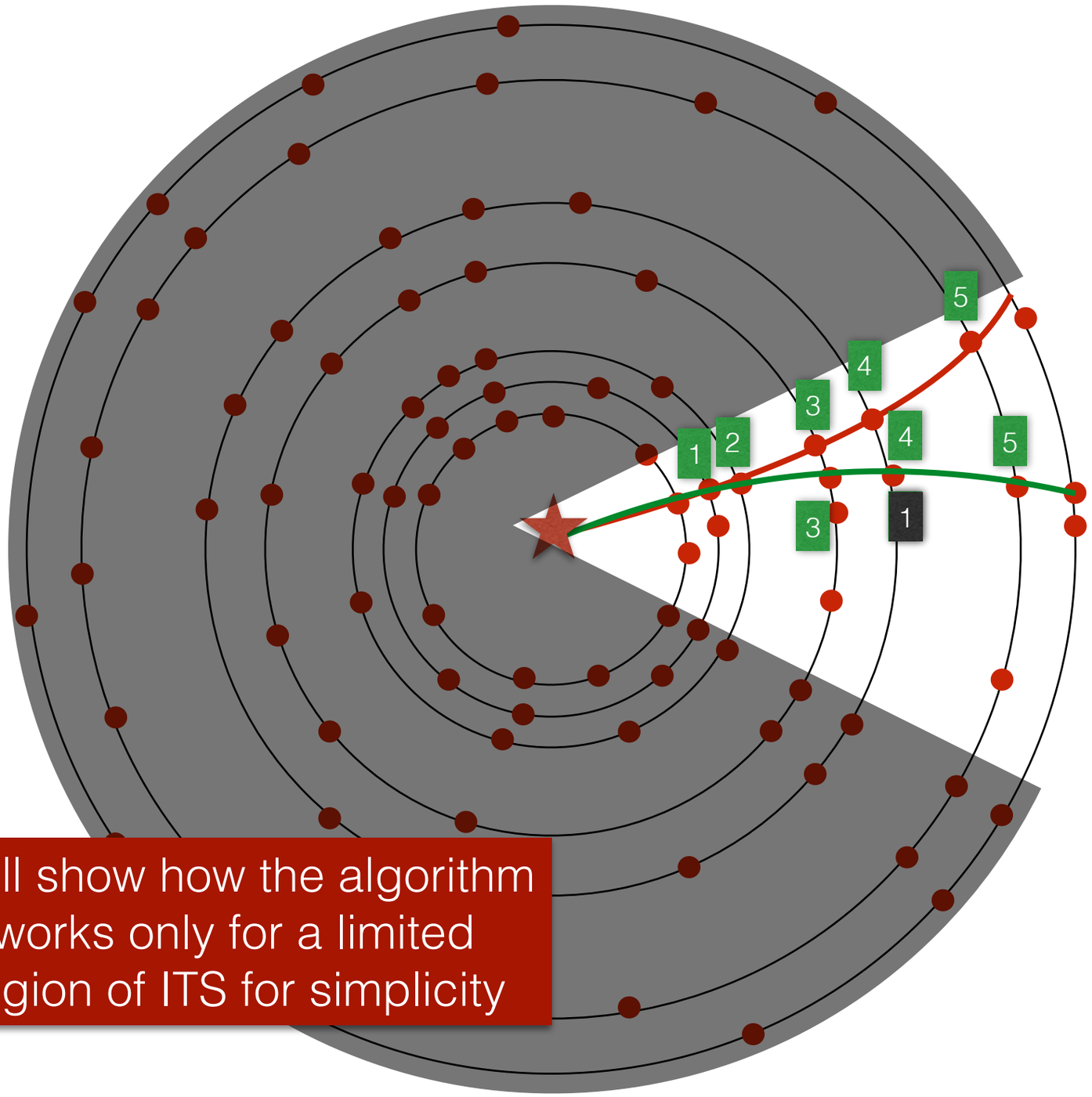
For each cluster on each layer a 2D window is opened. Then the clusters are joined with those on the next layer within the window

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

Cellular Automata in the ITS Upgrade



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

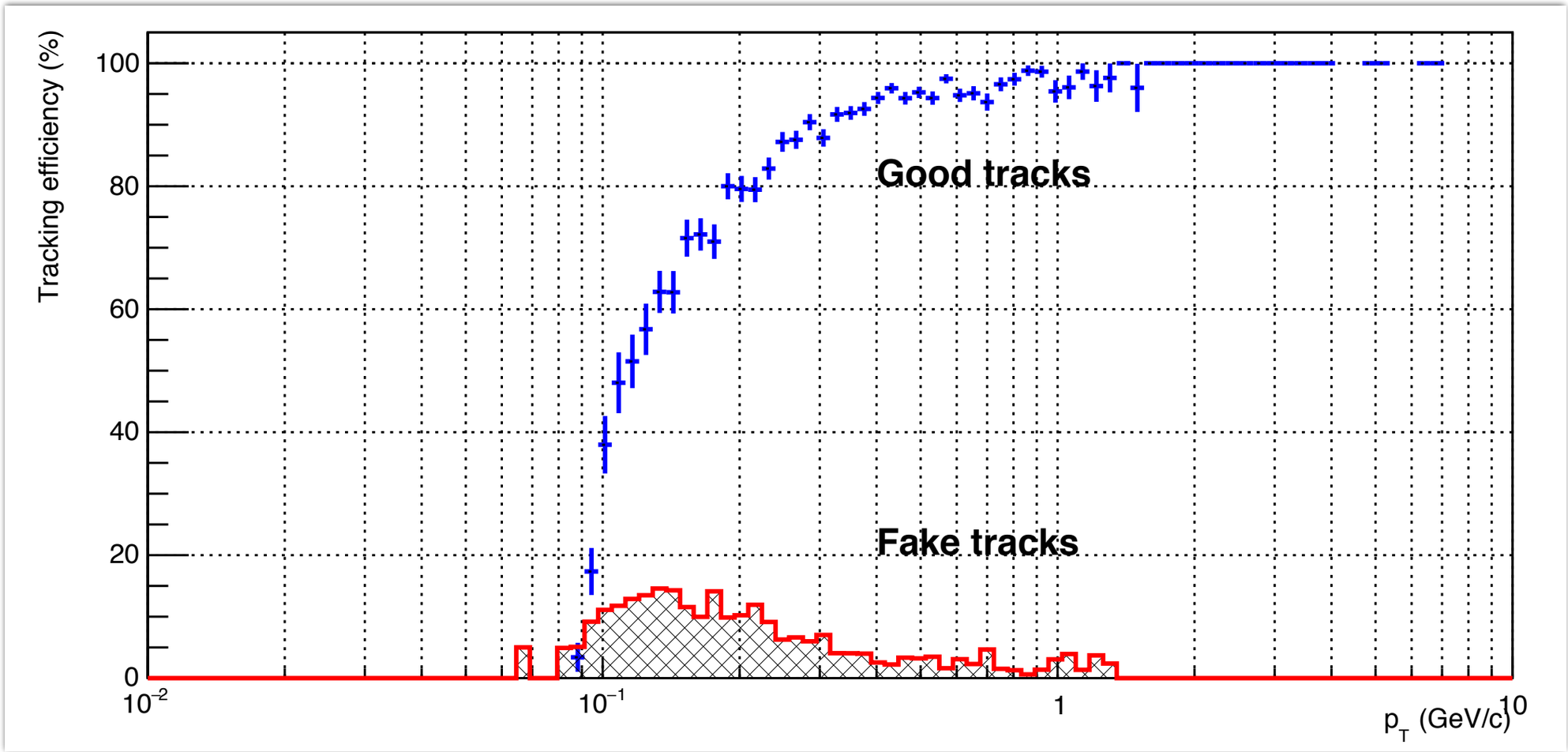
Each cell has an index representing the number of connected inner cells + 1

Longest, continuous sequences of indices represent candidates

KF Fitting

Performance of CA tracker for ITSU

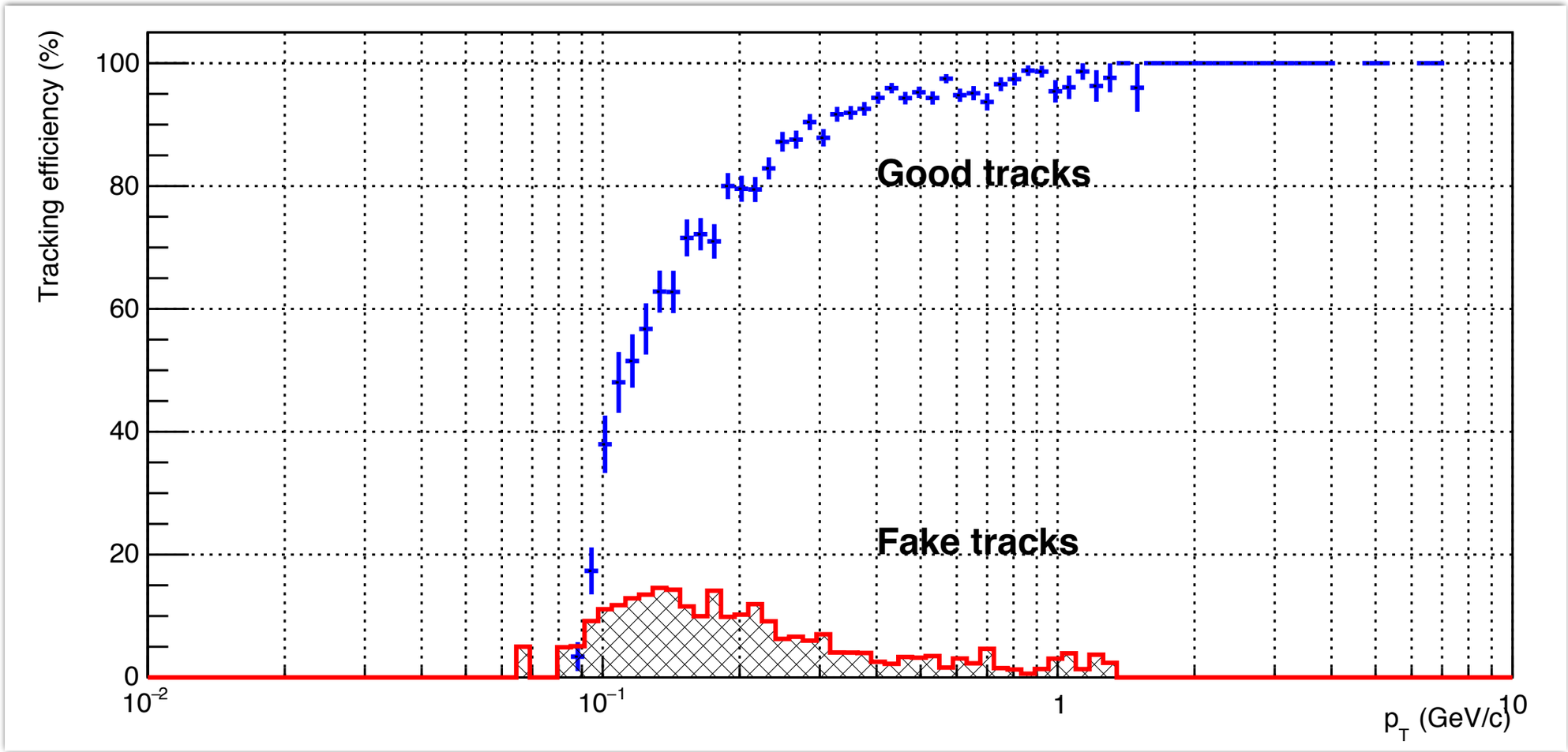
Efficiency for long tracks in central Pb-Pb (0-5%) w/o noise



Event type	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 10^{-5} noise	~1.3 s	~1.0 s	~5 s
pp with noise	~0.6 s	~0.7 s	~2.3 s

Performance of CA tracker for ITSU

Efficiency for long tracks in central Pb-Pb (0-5%) w/o noise

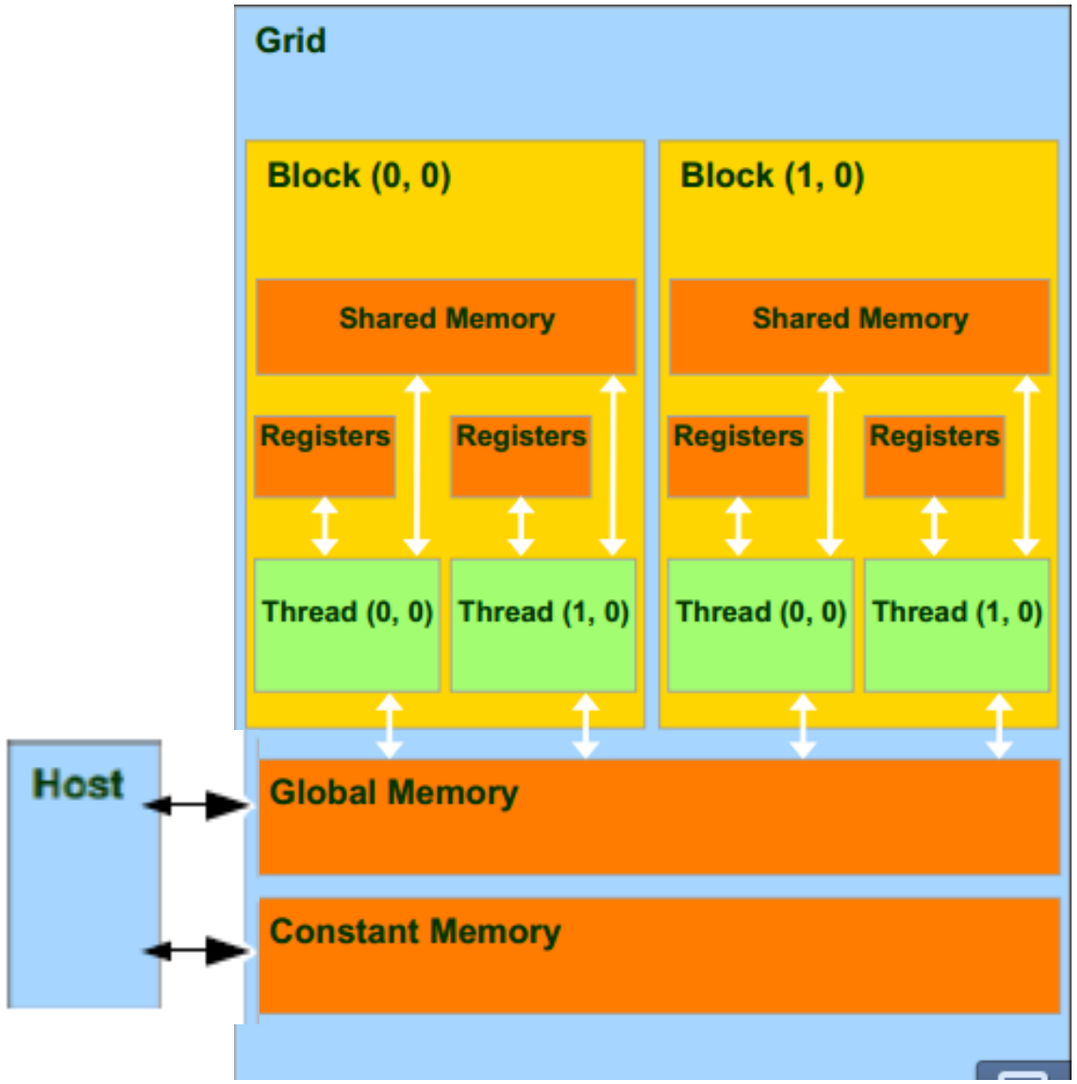


Event type	p_T > 0.6 GeV/c	p_T > 2 GeV/c	Full
Central Pb-Pb	~0.3 s	~0.2 s	~0.7 s
Central Pb-Pb with 10^{-5} noise	~1.3 s	~1.0 s	~5 s
pp with noise	~0.6 s	~0.7 s	~2.3 s

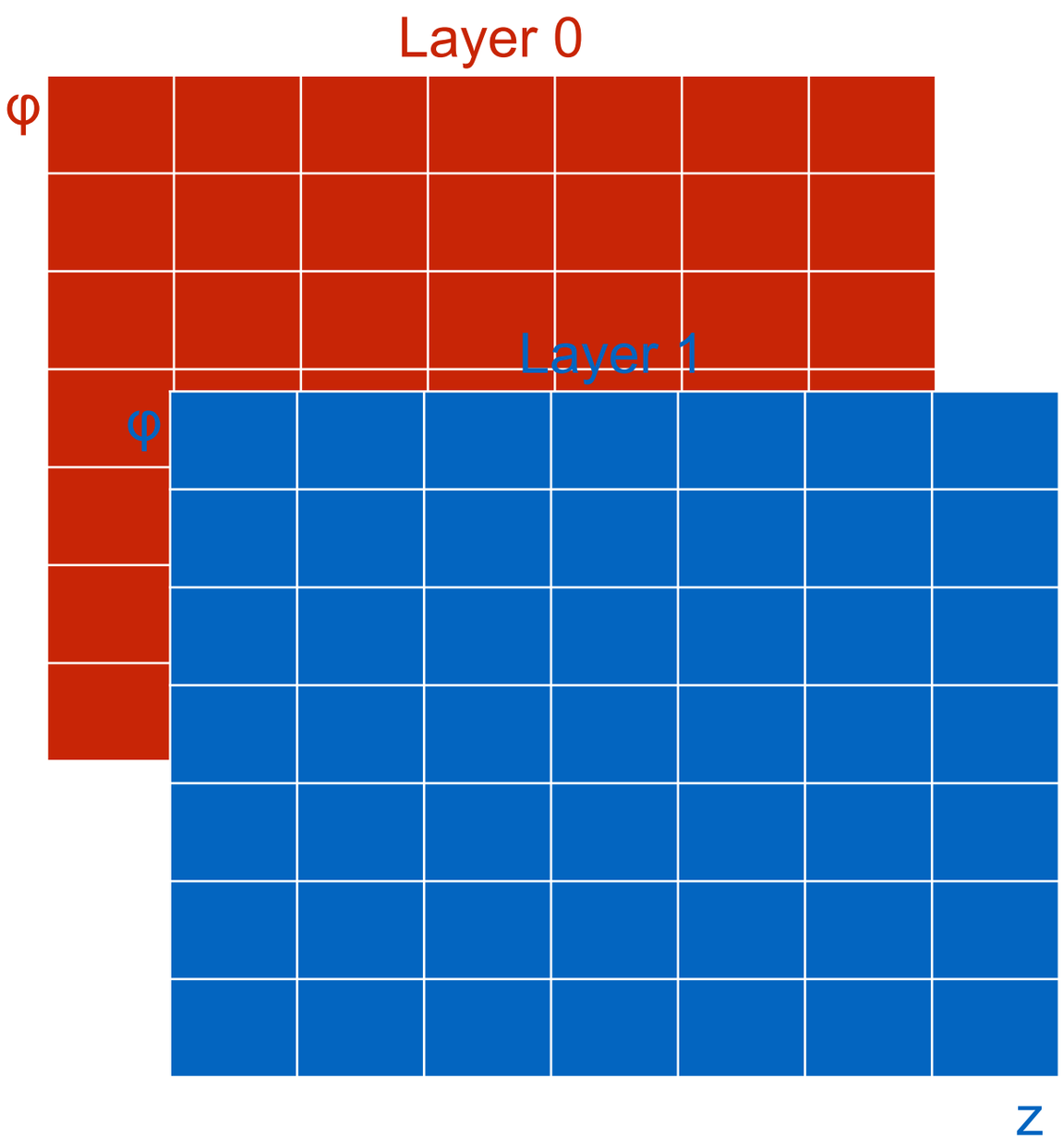
New baseline for the chip noise is much lower!

First porting of the doublet finder on OpenCL

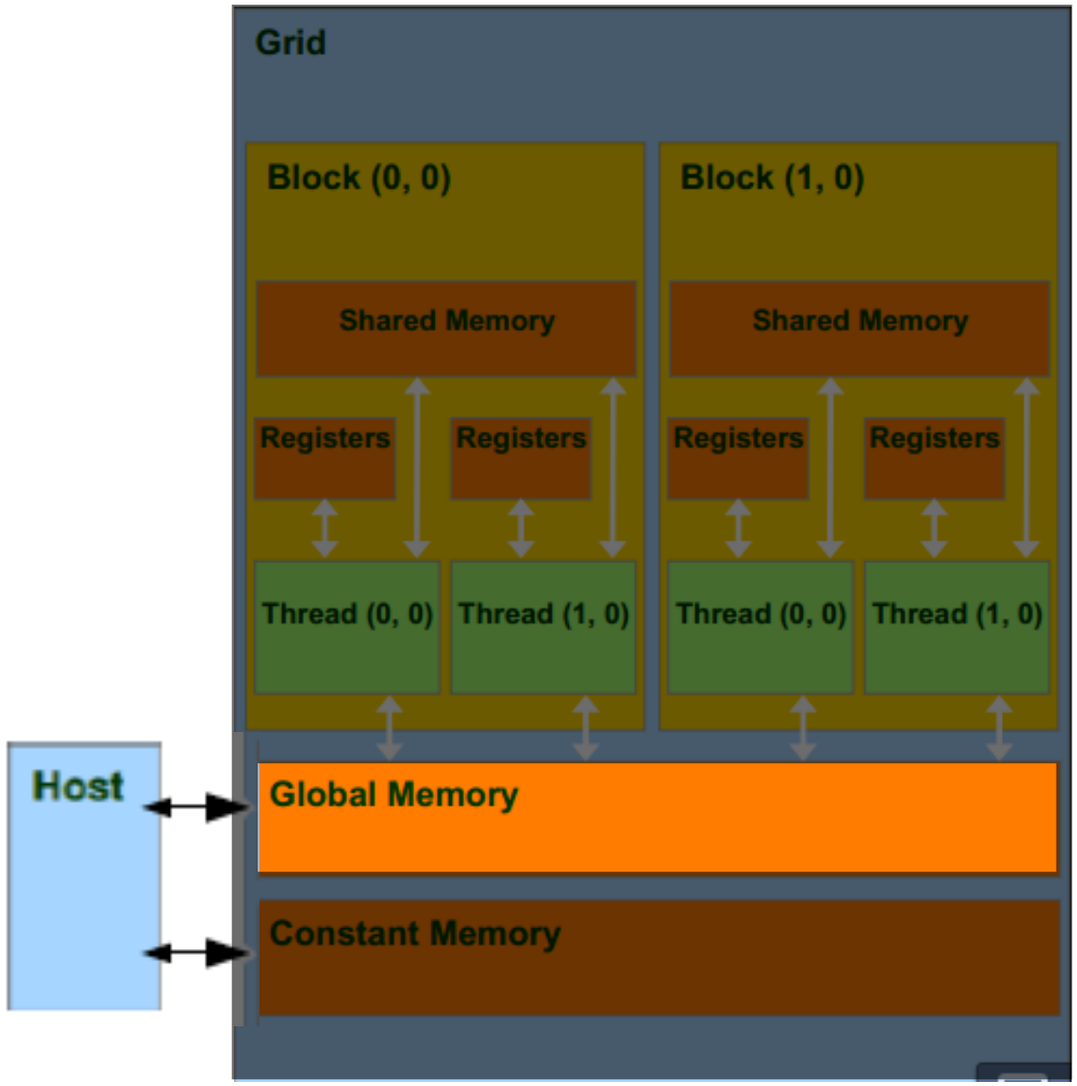
OpenCL reference



First porting of the doublet finder on OpenCL

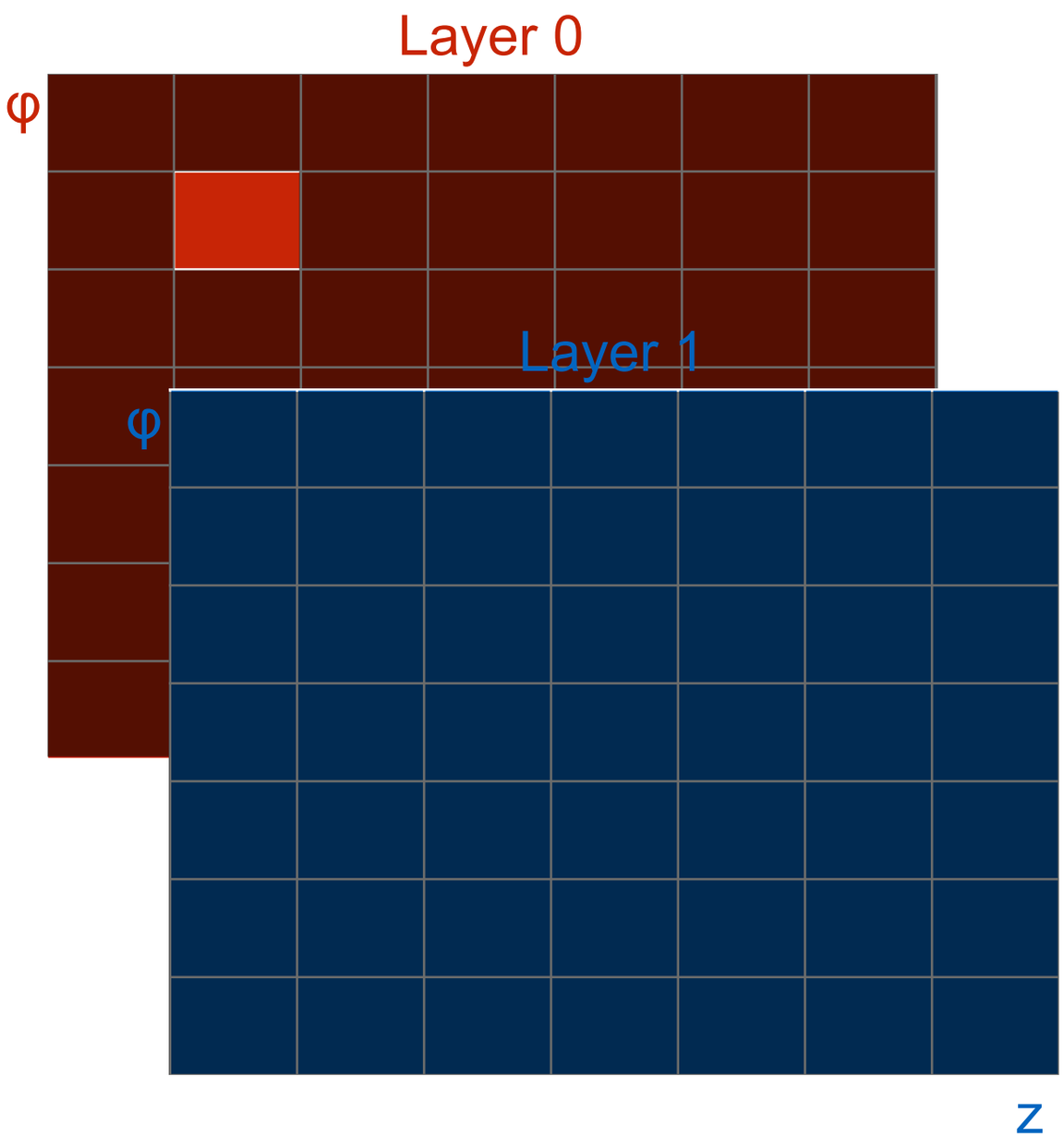


OpenCL reference

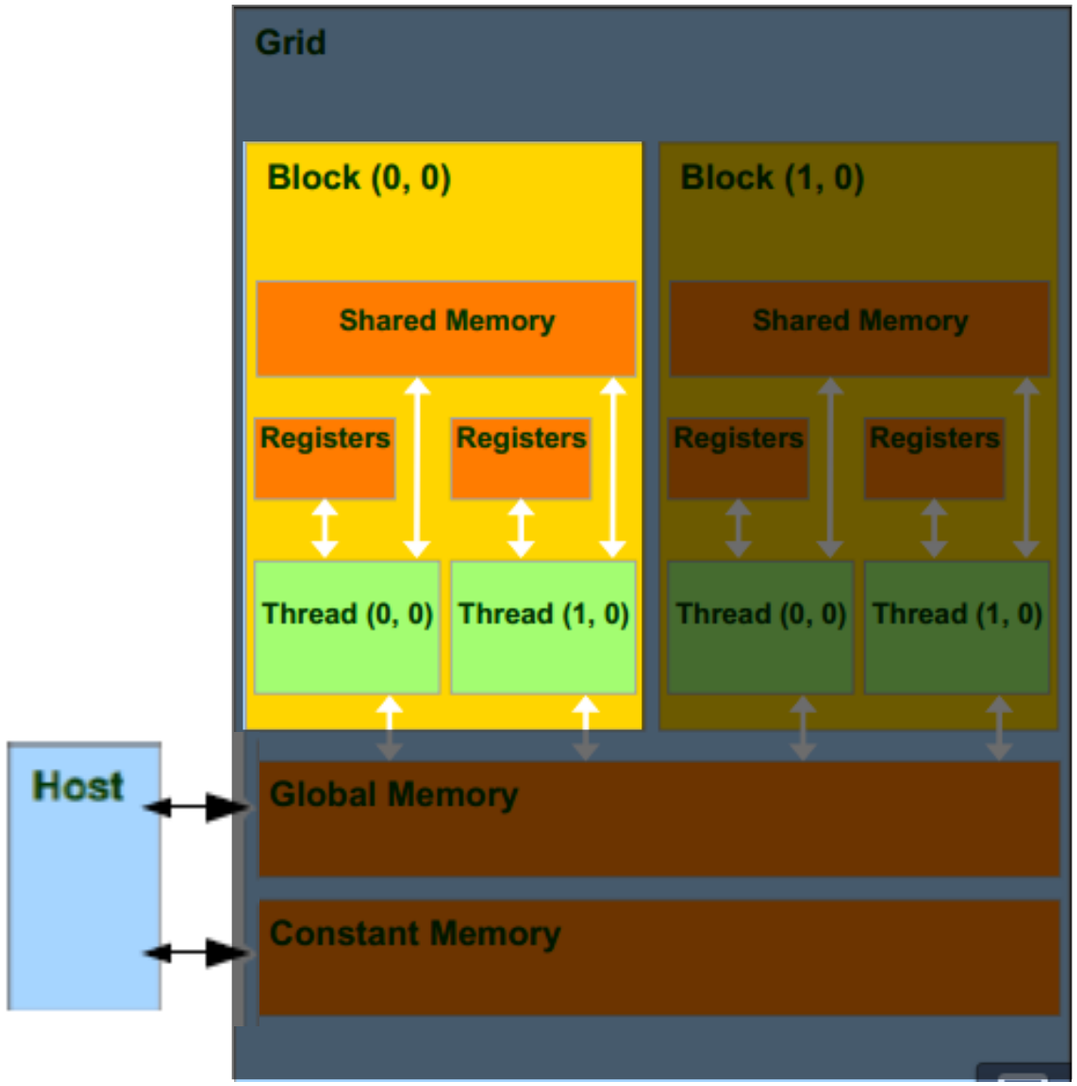


Transfer of the clusters and LUT on the GPU global (constant) memory

First porting of the doublet finder on OpenCL

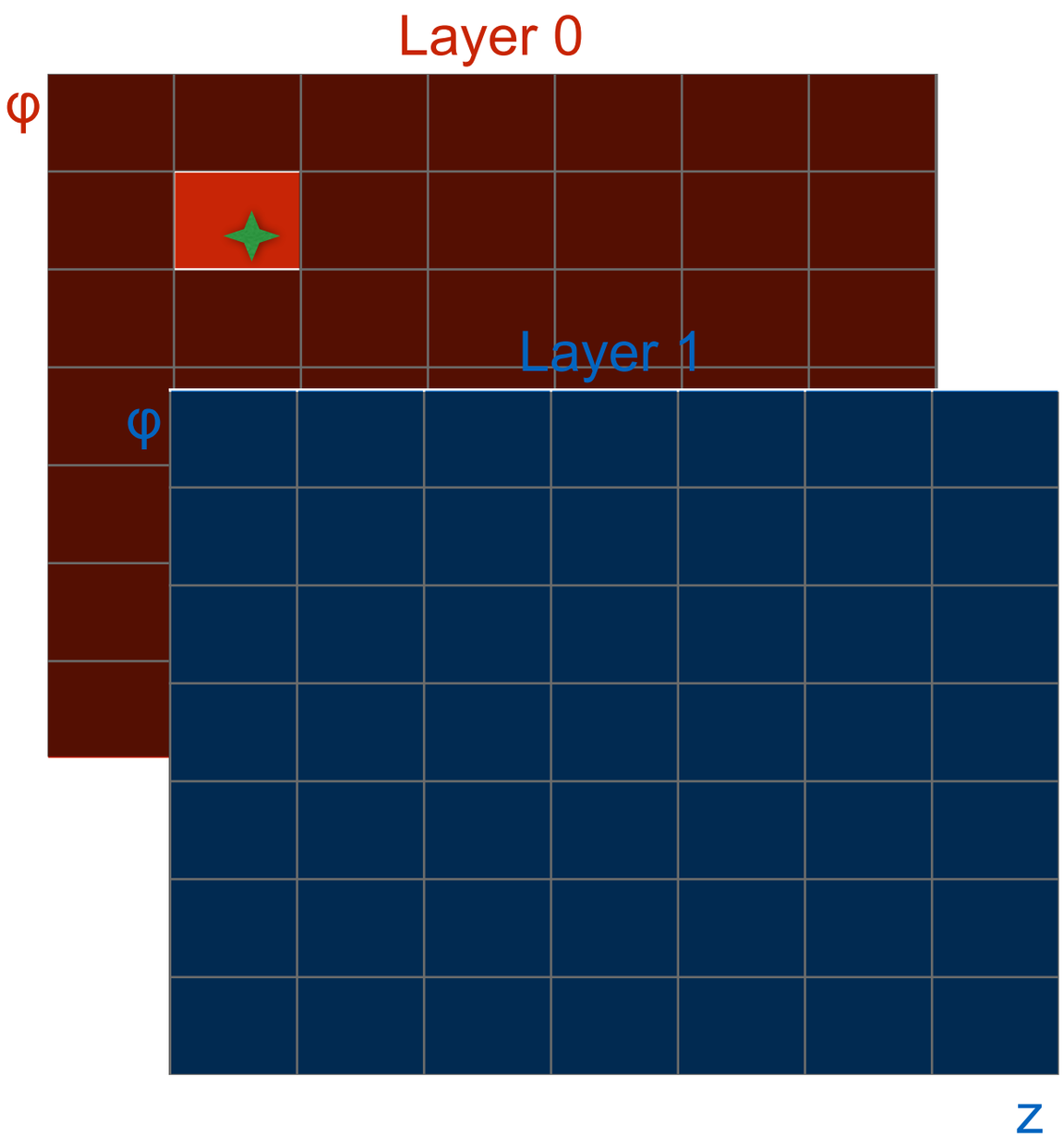


OpenCL reference

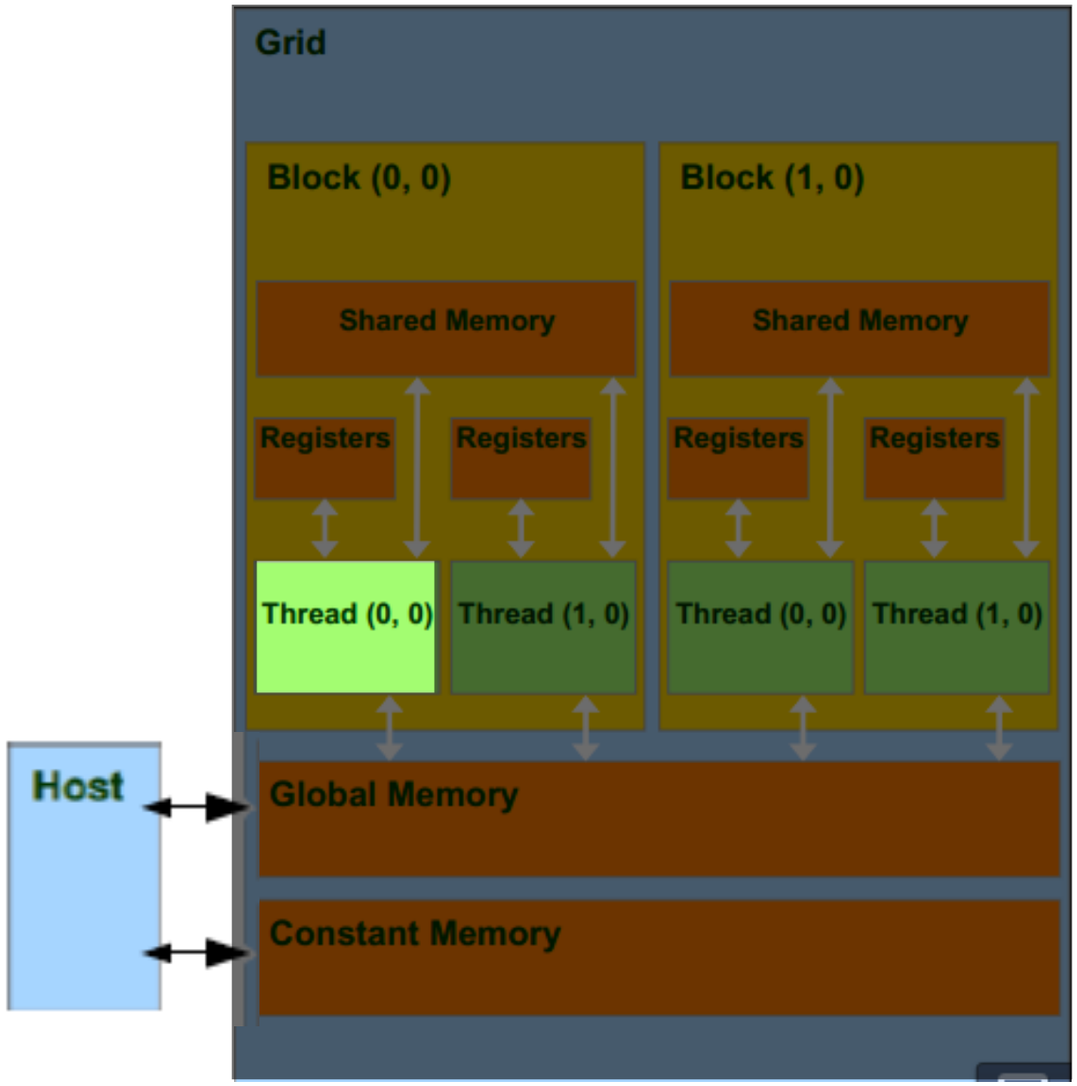


Each block of thread will process one bin on layer 0

First porting of the doublet finder on OpenCL

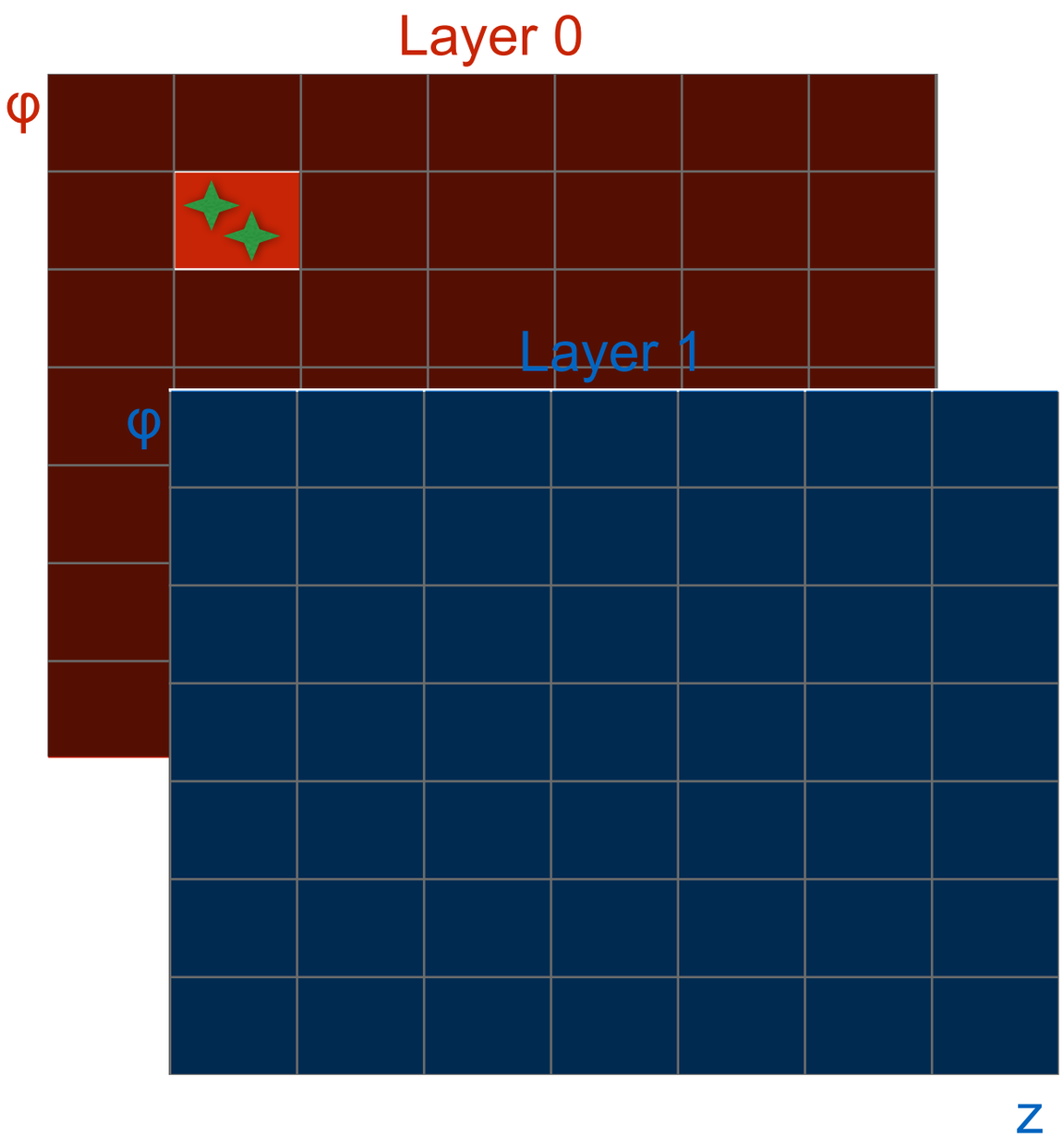


OpenCL reference

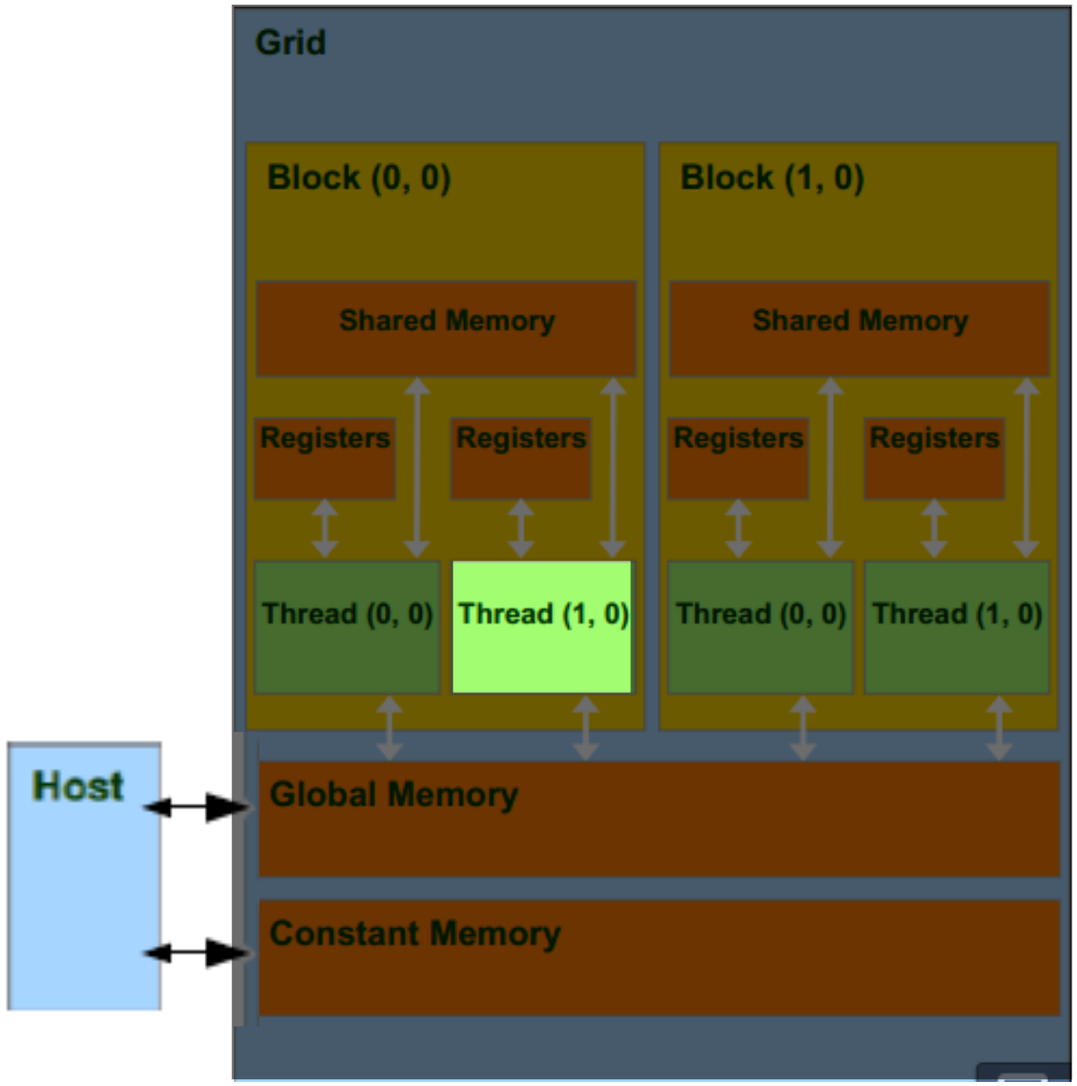


Each thread will take care of one cluster layer 0

First porting of the doublet finder on OpenCL

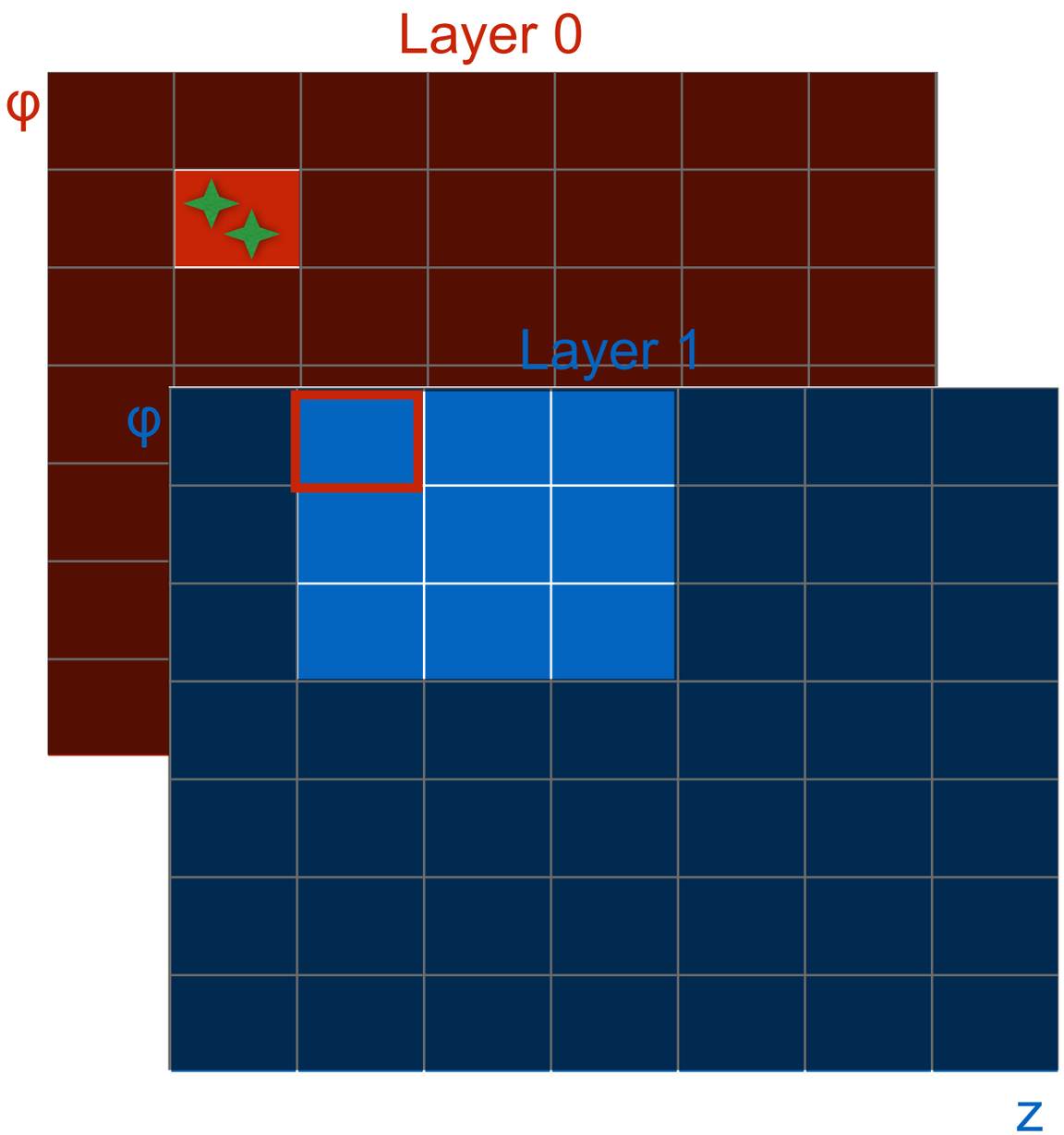


OpenCL reference

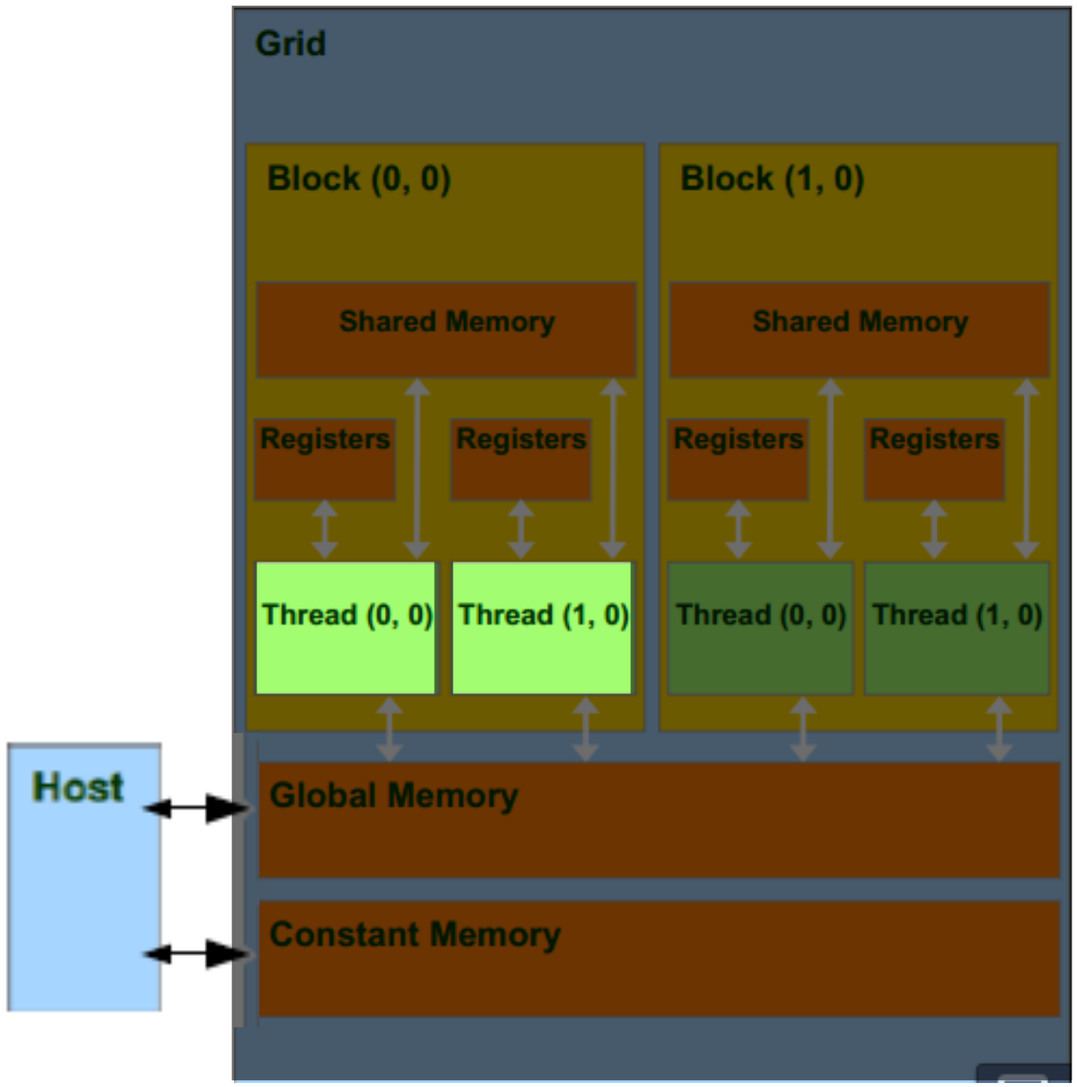


Each thread will take care of one cluster layer 0

First porting of the doublet finder on OpenCL

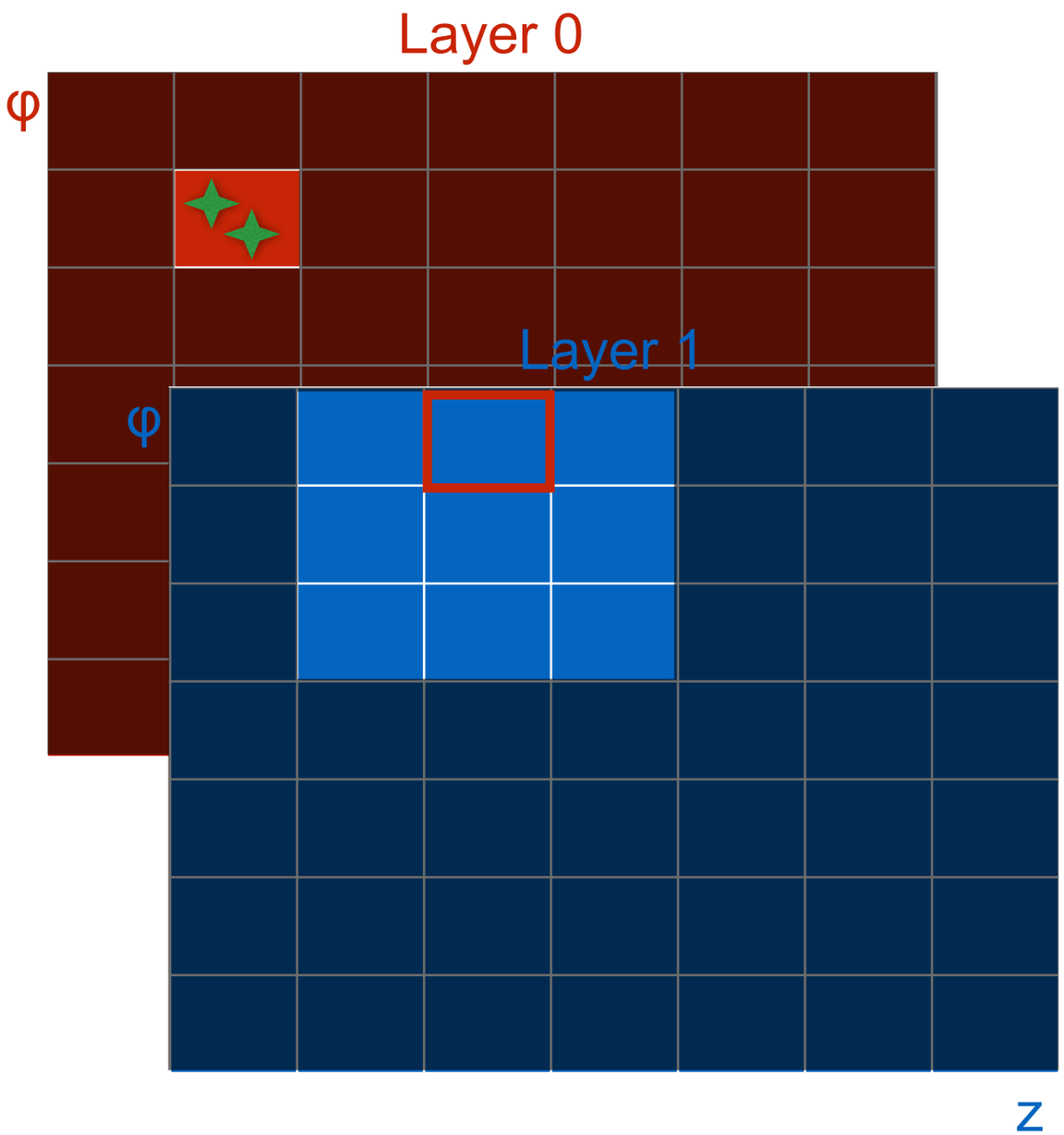


OpenCL reference

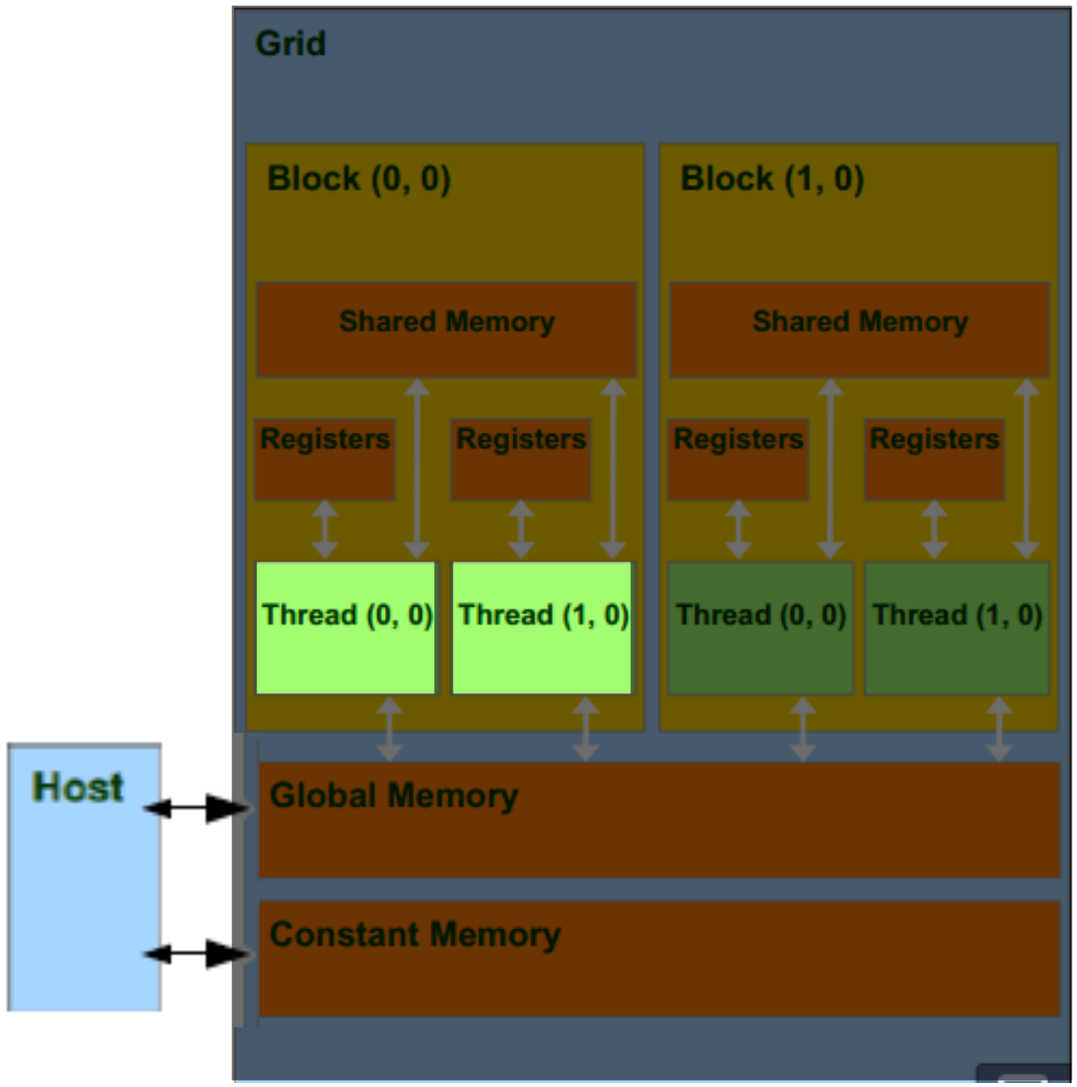


All the clusters of one bin on layer 1 are copied in the shared memory and a for loop starts over these clusters looking for good doublets

First porting of the doublet finder on OpenCL

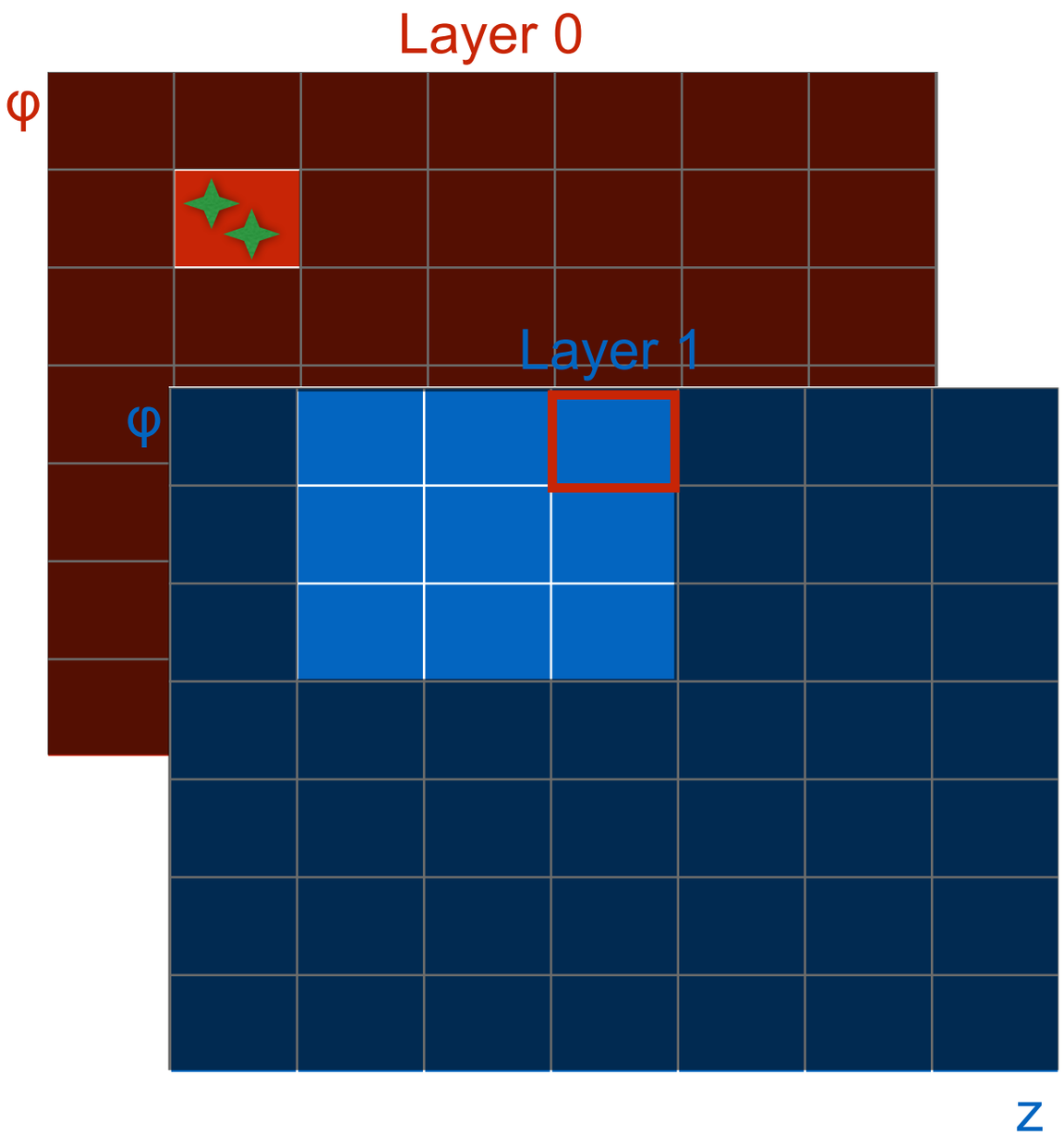


OpenCL reference

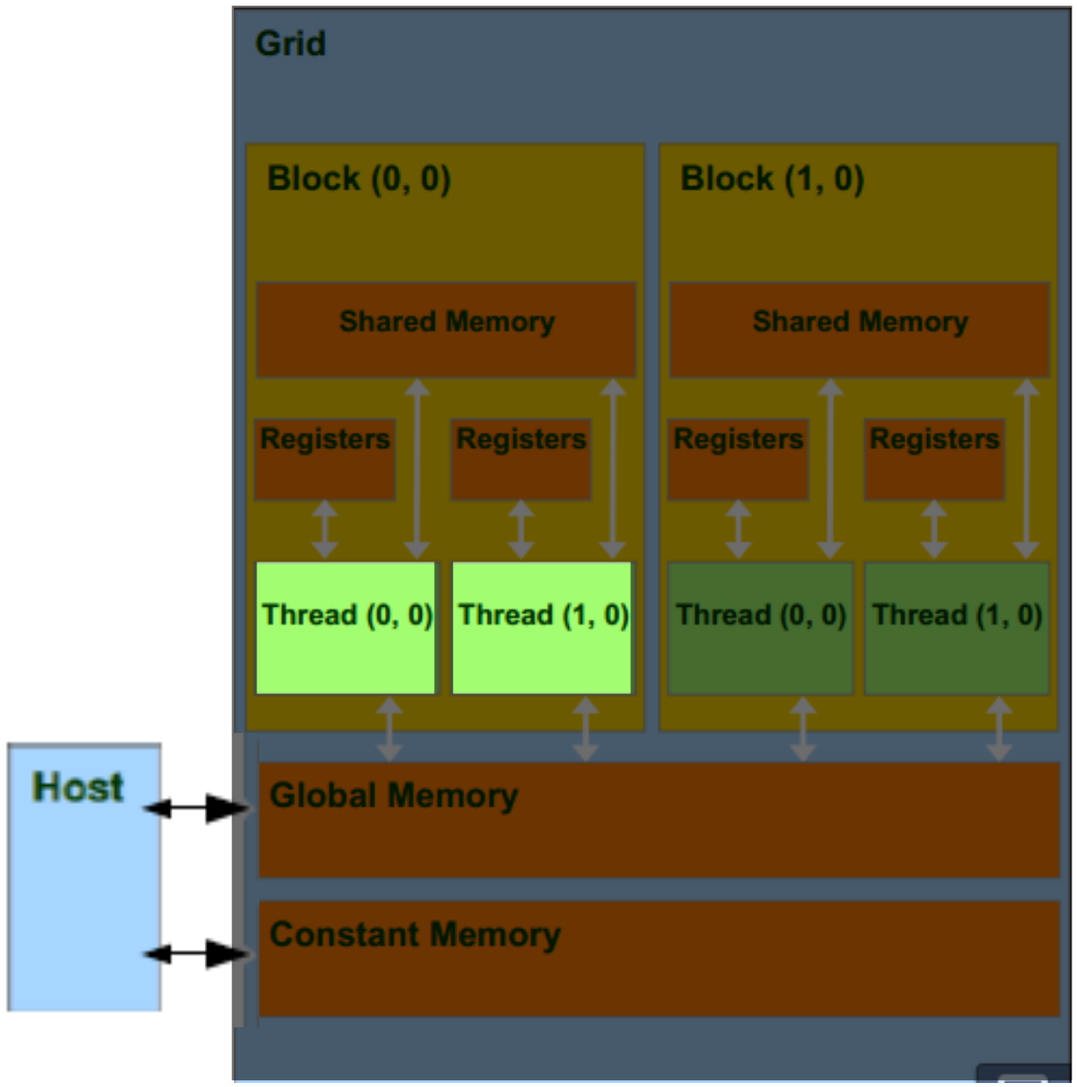


All the clusters of one bin on layer 1 are copied in the shared memory and a for loop starts over these clusters looking for good doublets

First porting of the doublet finder on OpenCL

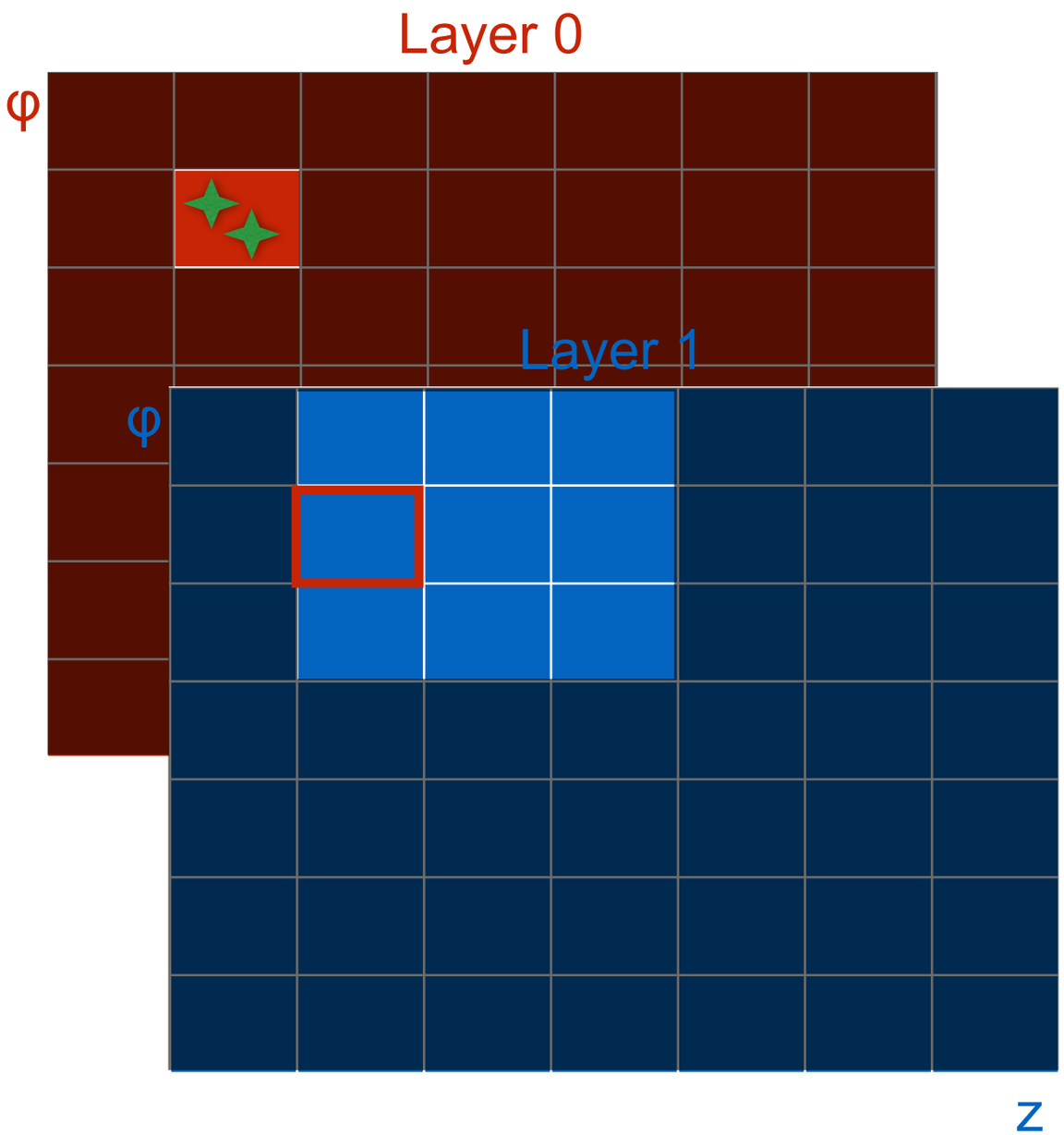


OpenCL reference

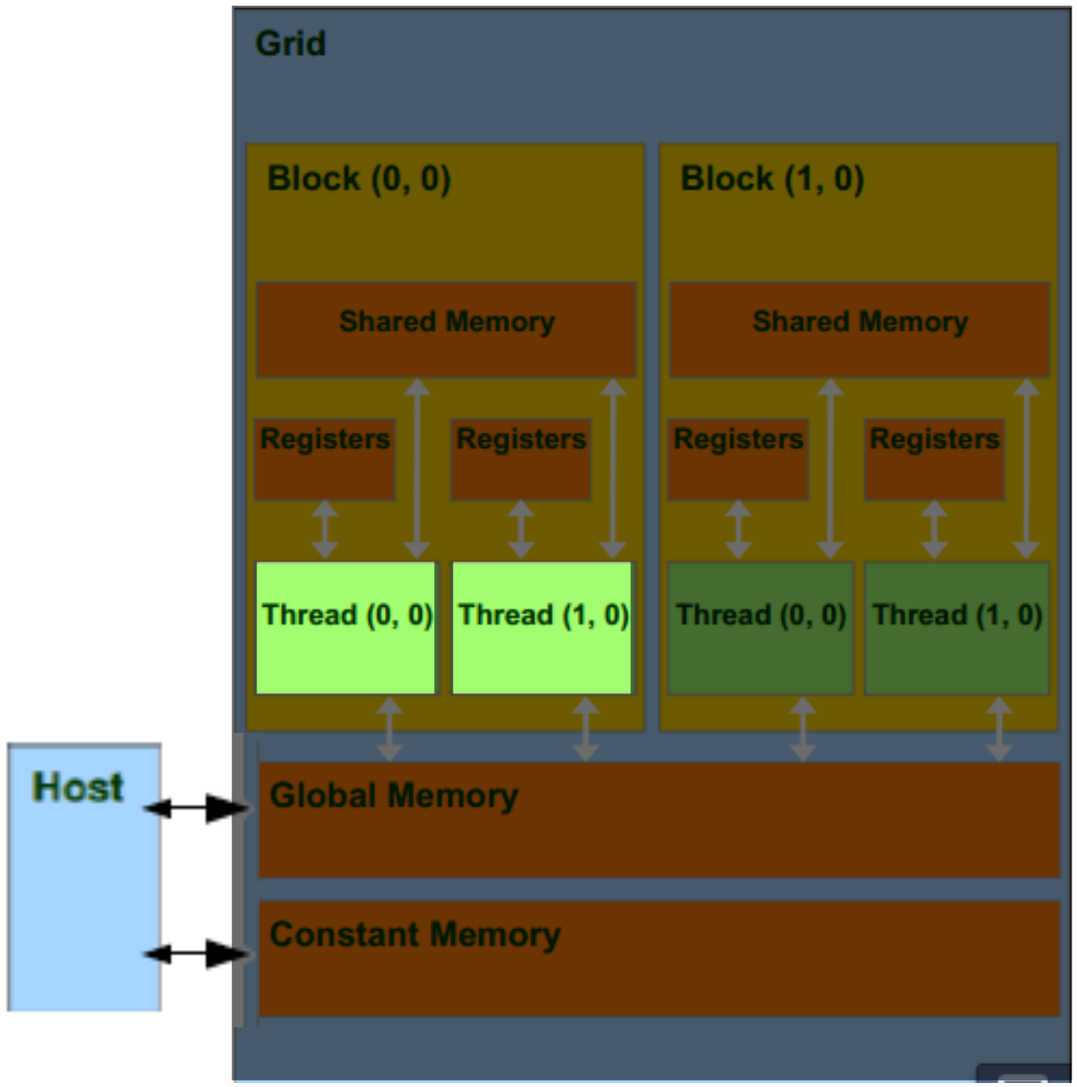


All the clusters of one bin on layer 1 are copied in the shared memory and a for loop starts over these clusters looking for good doublets

First porting of the doublet finder on OpenCL

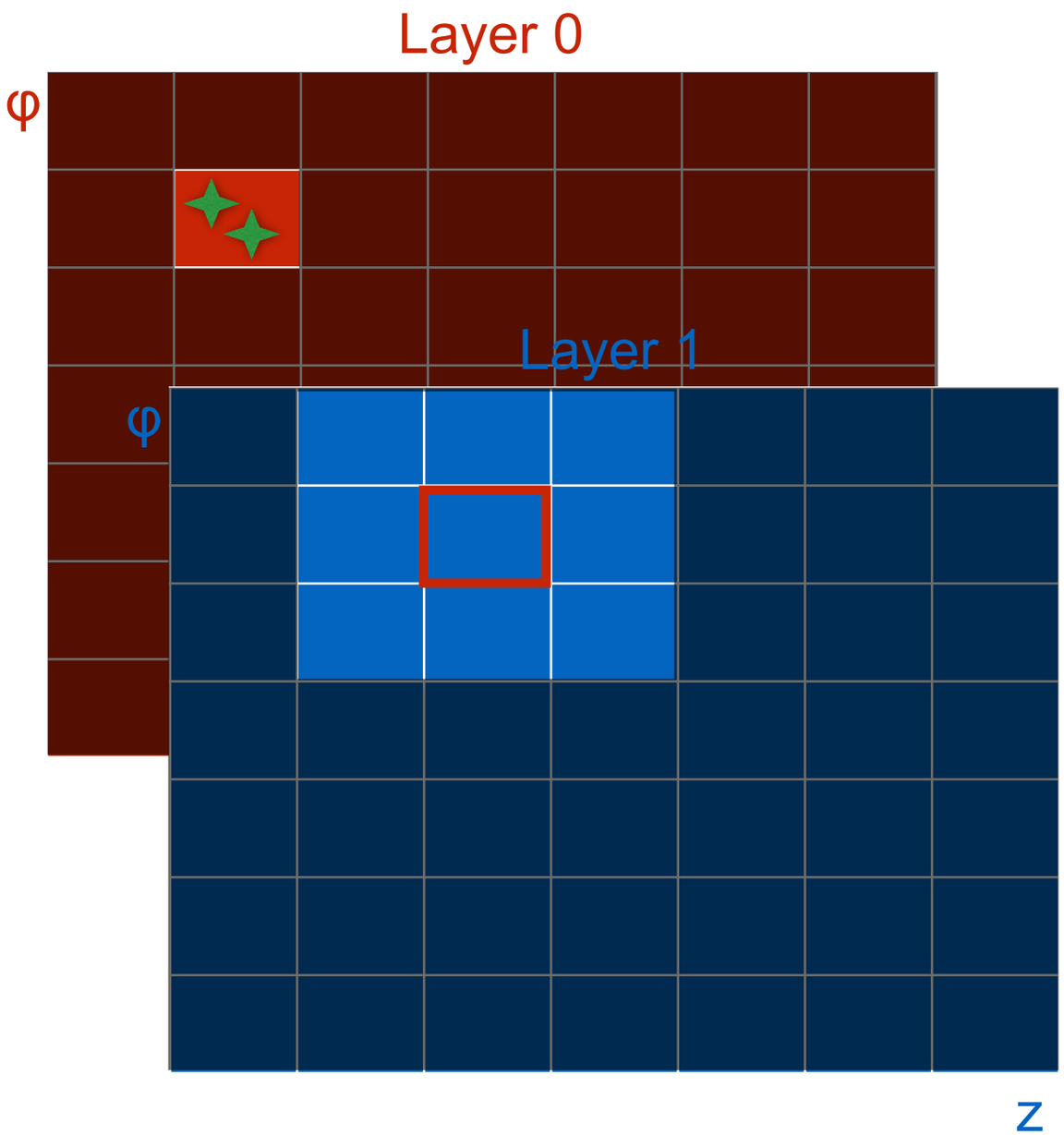


OpenCL reference

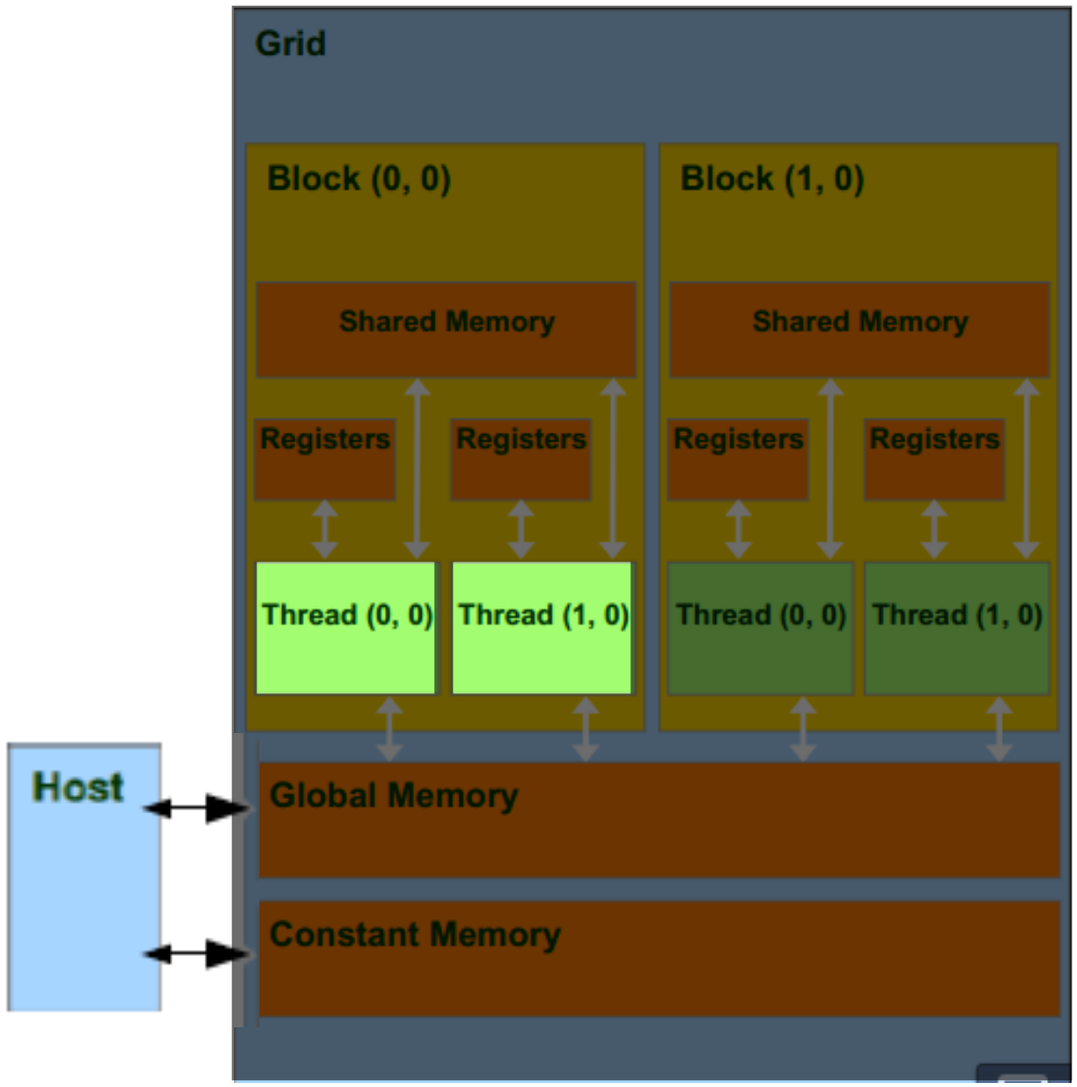


All the clusters of one bin on layer 1 are copied in the shared memory and a for loop starts over these clusters looking for good doublets

First porting of the doublet finder on OpenCL

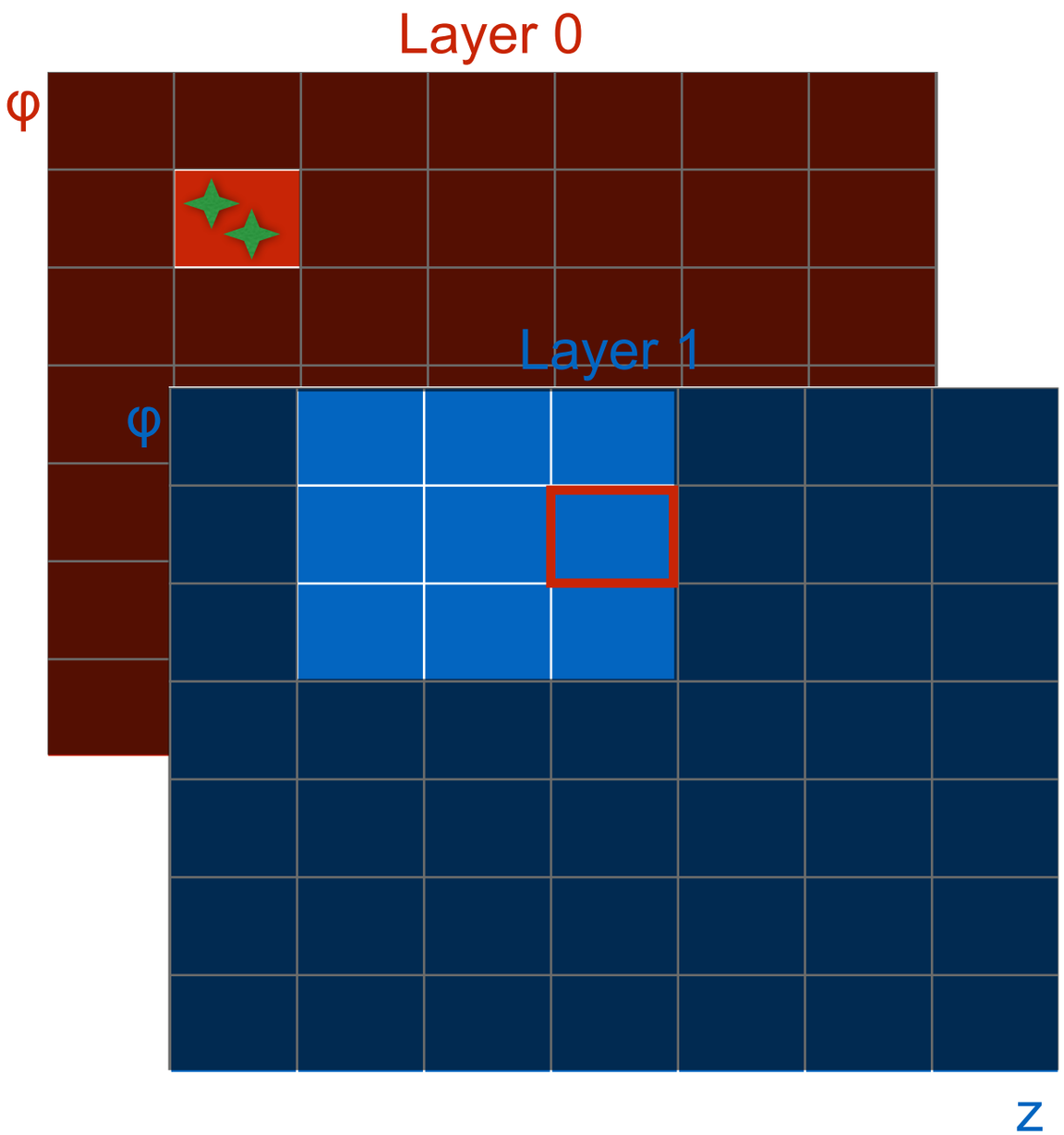


OpenCL reference

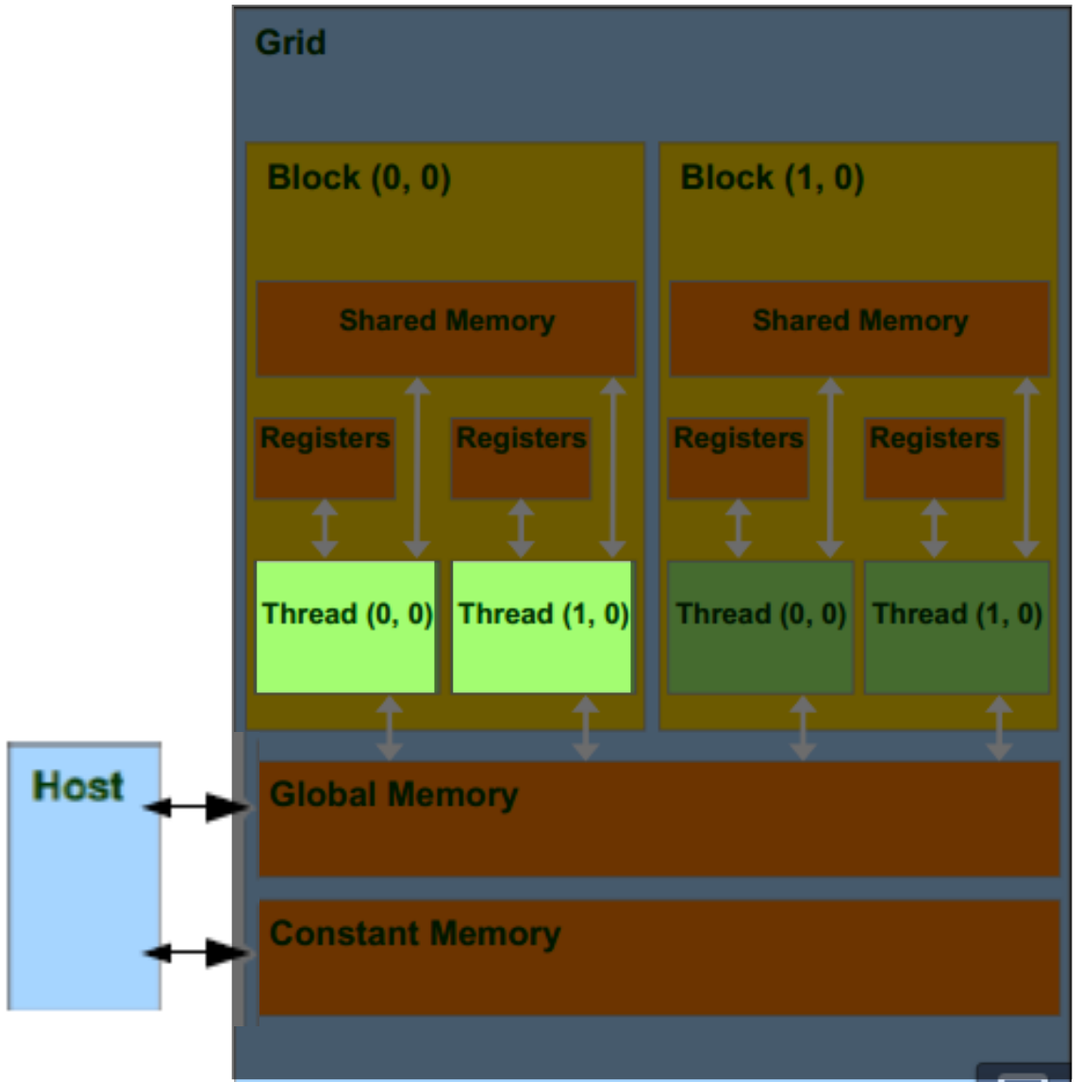


All the clusters of one bin on layer 1 are copied in the shared memory and a for loop starts over these clusters looking for good doublets

First porting of the doublet finder on OpenCL

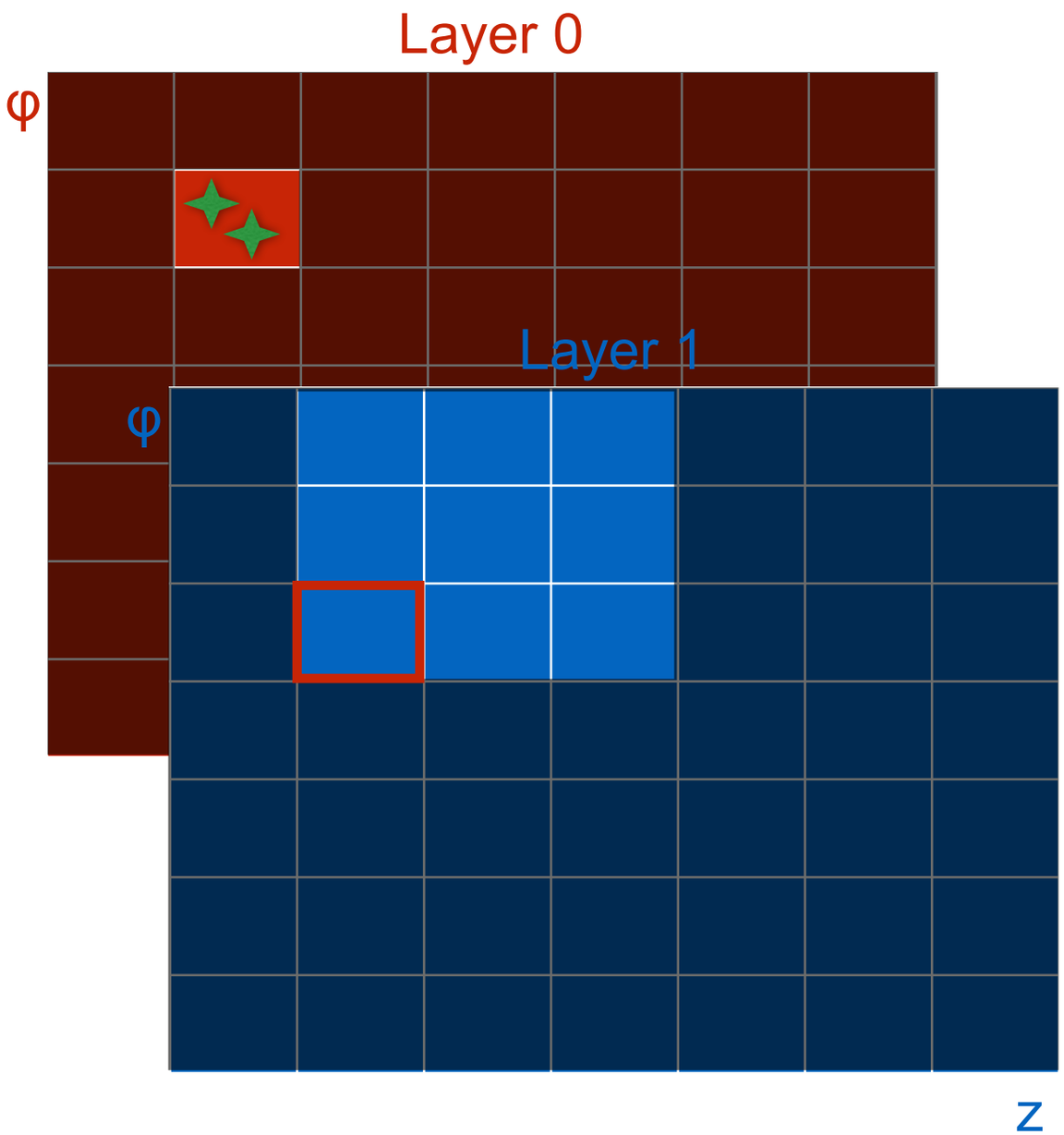


OpenCL reference

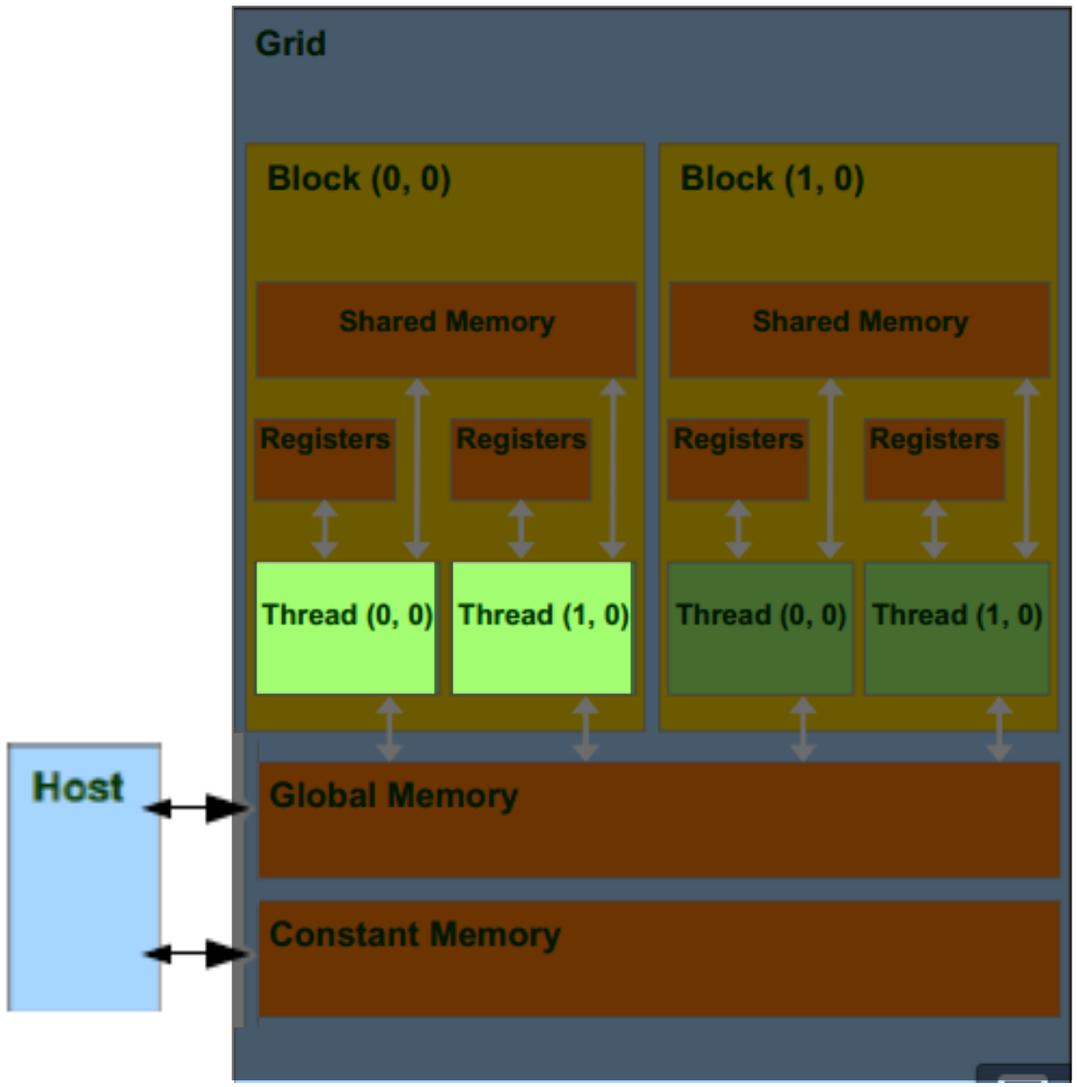


All the clusters of one bin on layer 1 are copied in the shared memory and a for loop starts over these clusters looking for good doublets

First porting of the doublet finder on OpenCL

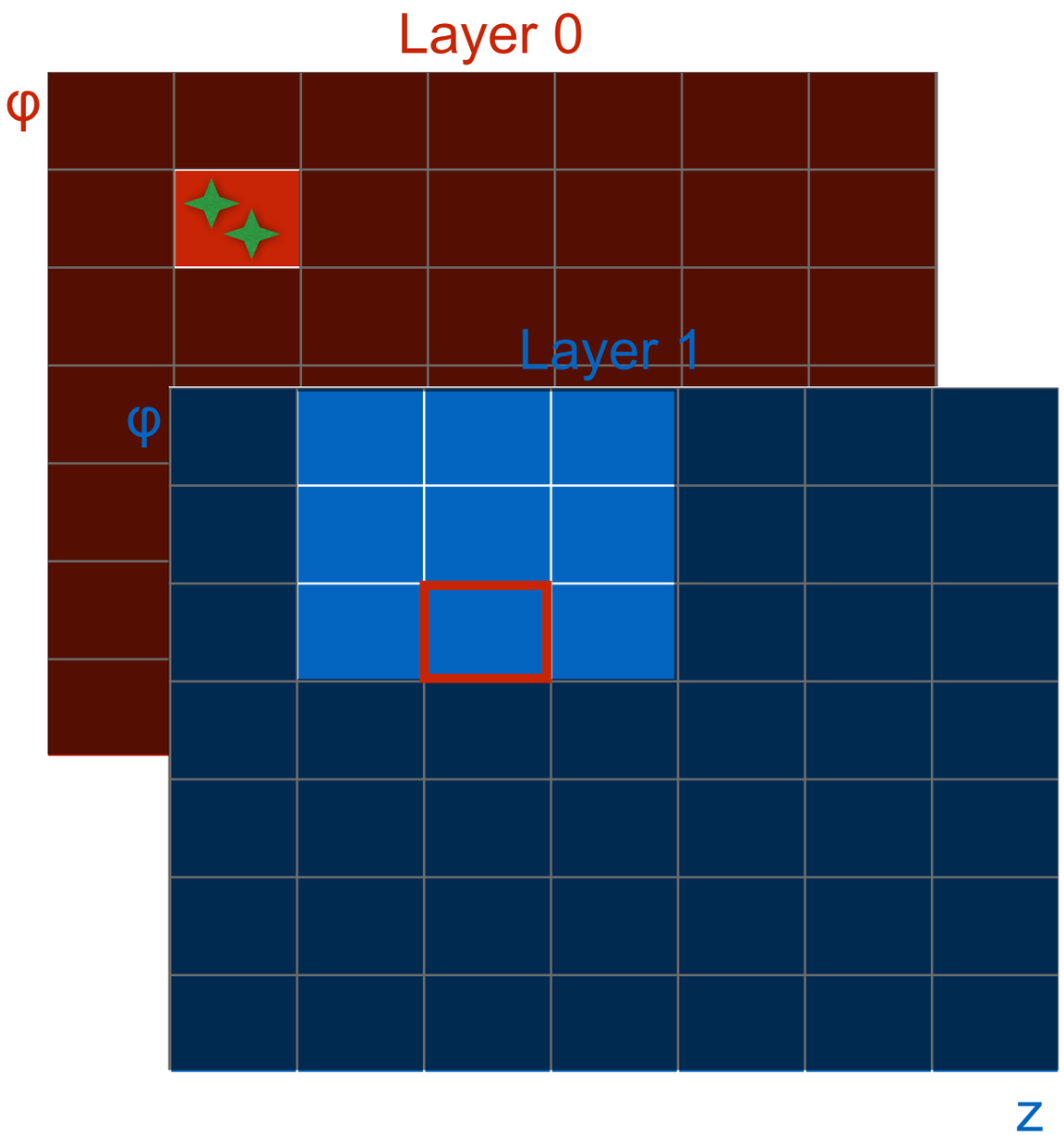


OpenCL reference

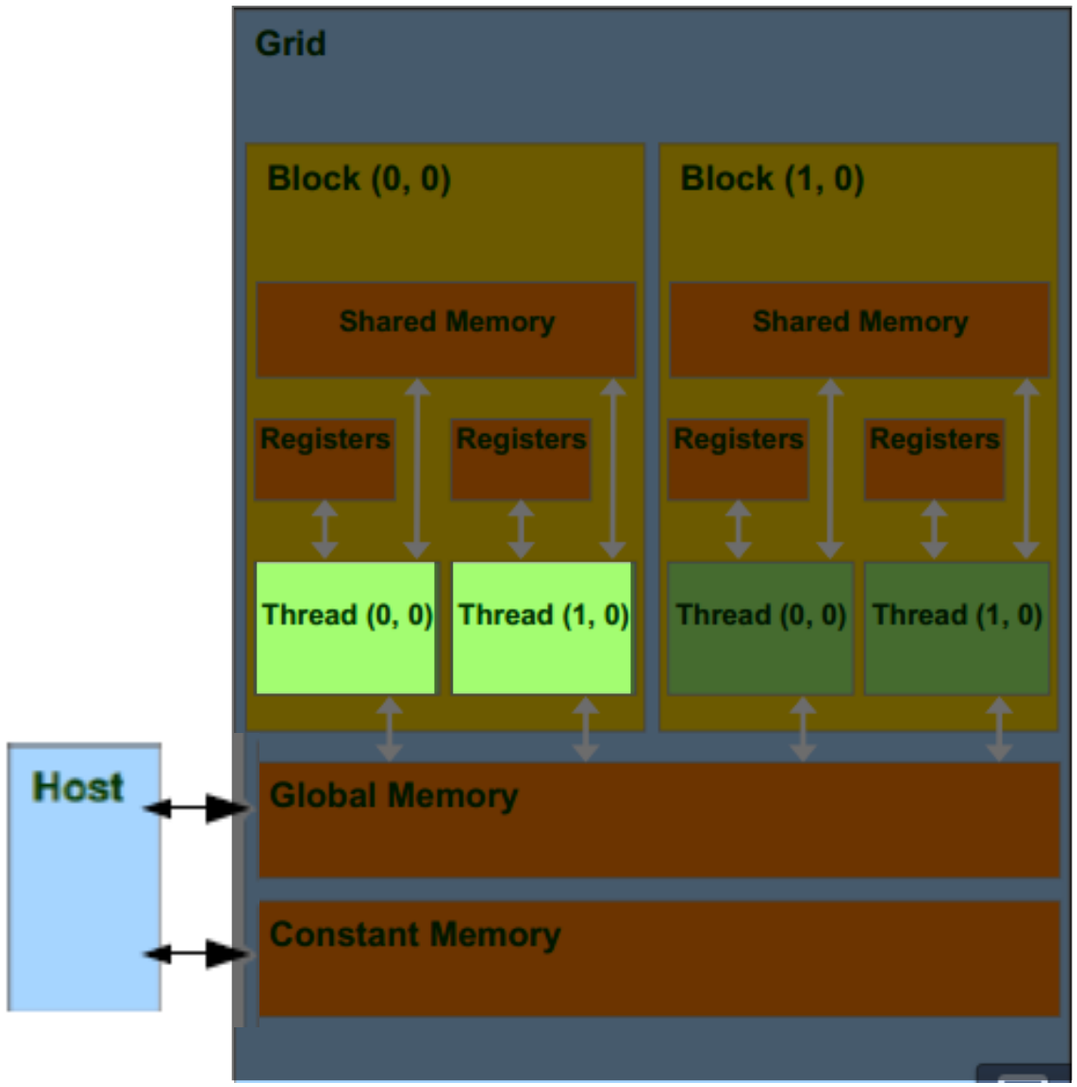


All the clusters of one bin on layer 1 are copied in the shared memory and a for loop starts over these clusters looking for good doublets

First porting of the doublet finder on OpenCL

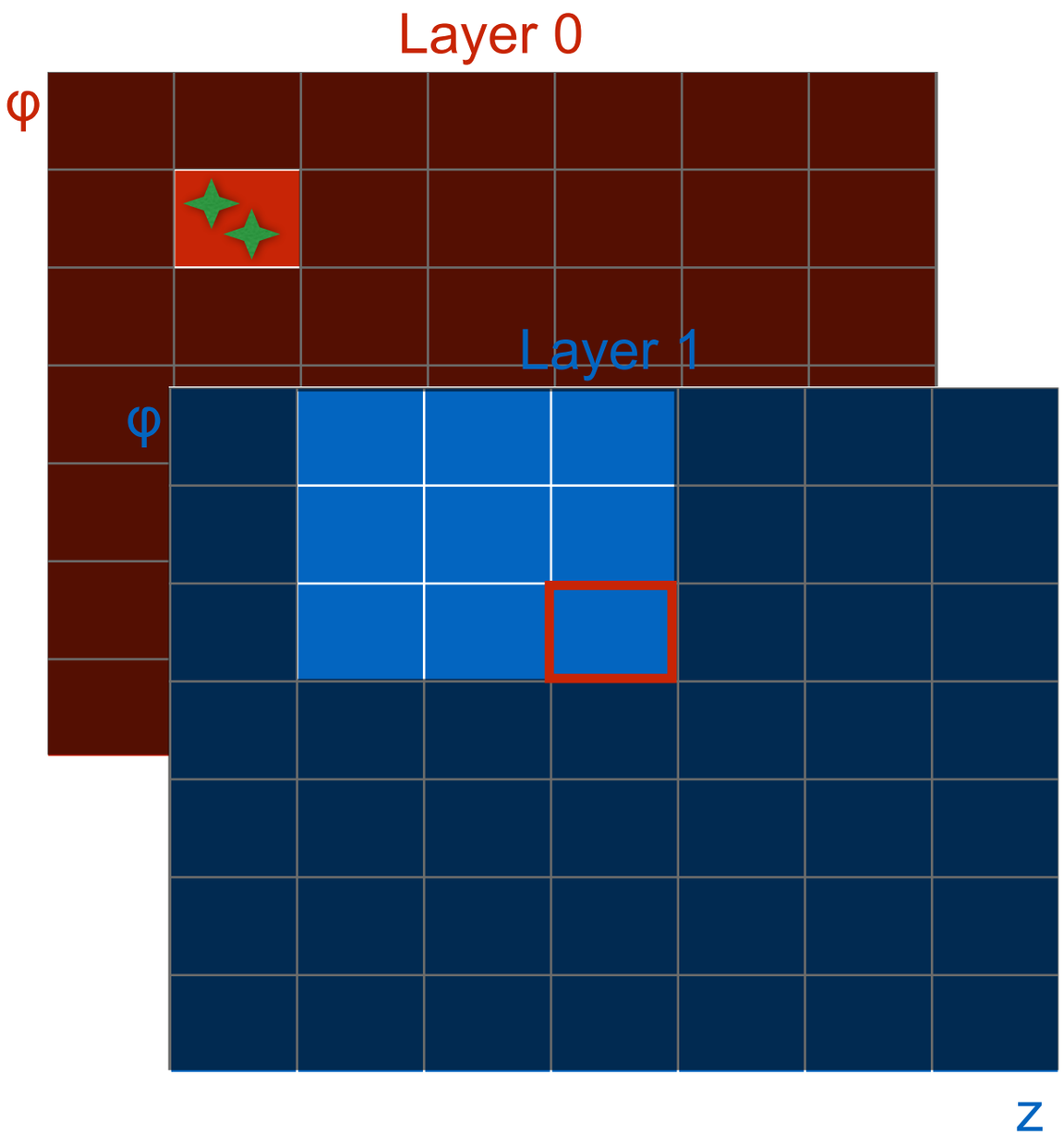


OpenCL reference

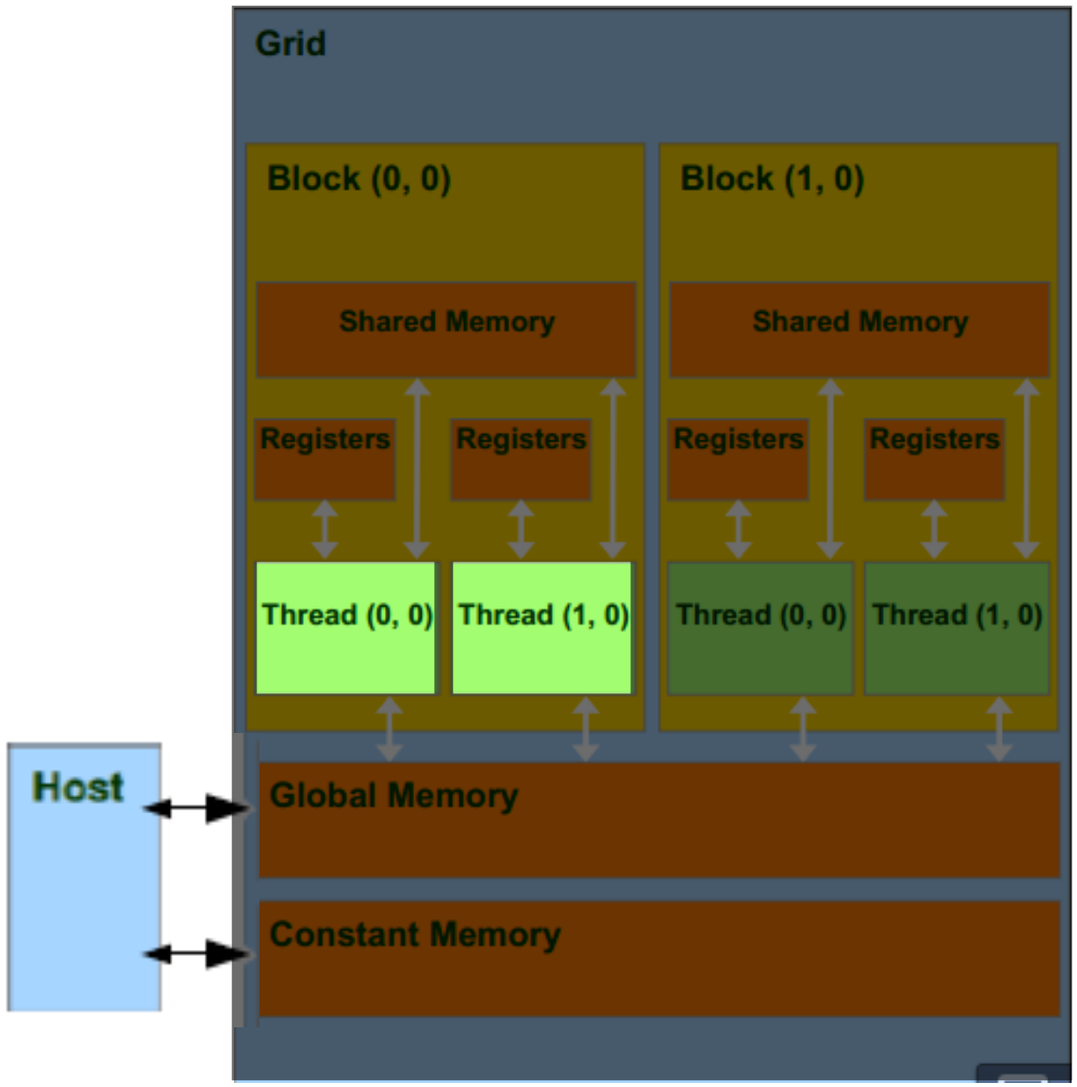


All the clusters of one bin on layer 1 are copied in the shared memory and a for loop starts over these clusters looking for good doublets

First porting of the doublet finder on OpenCL

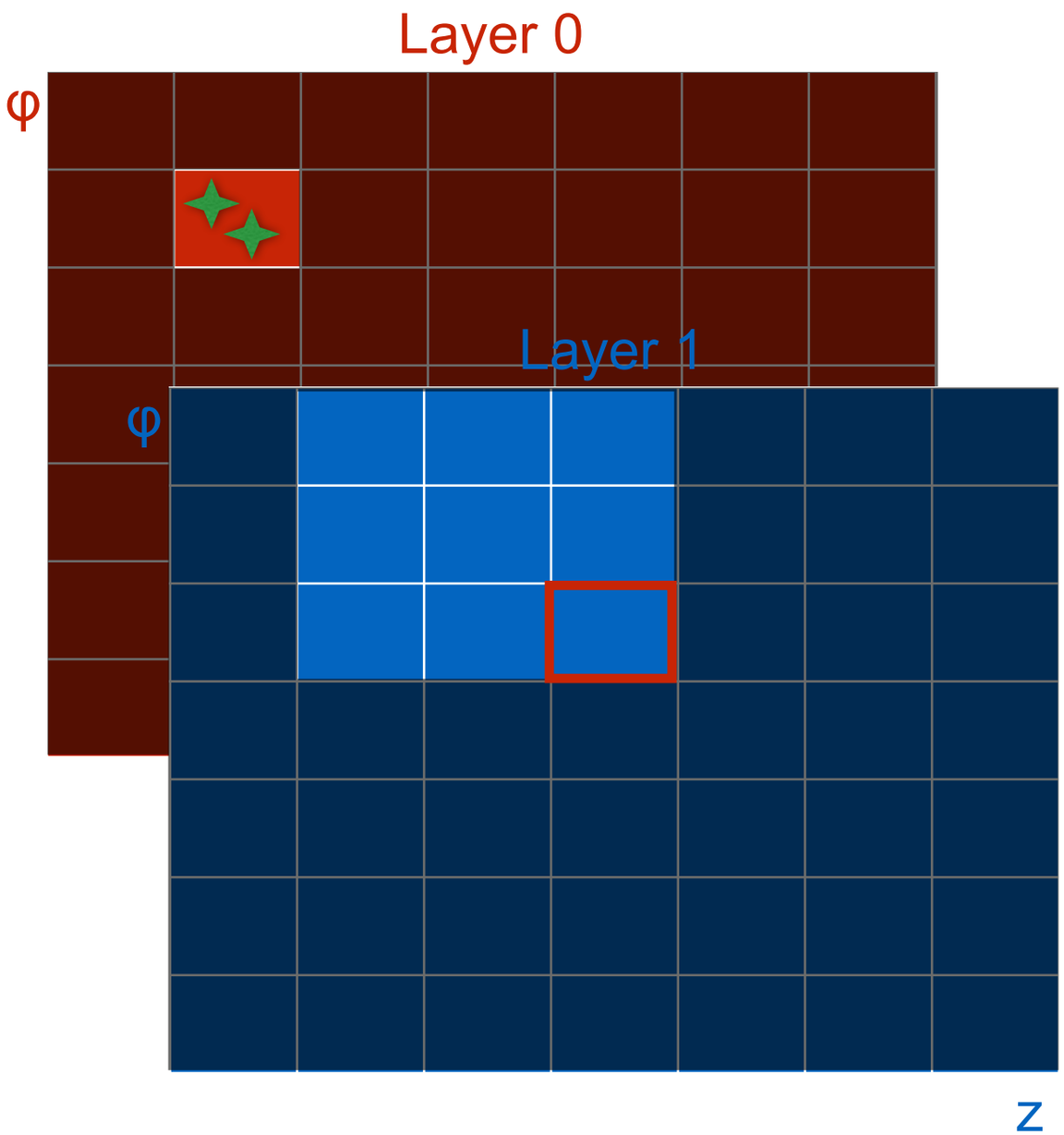


OpenCL reference

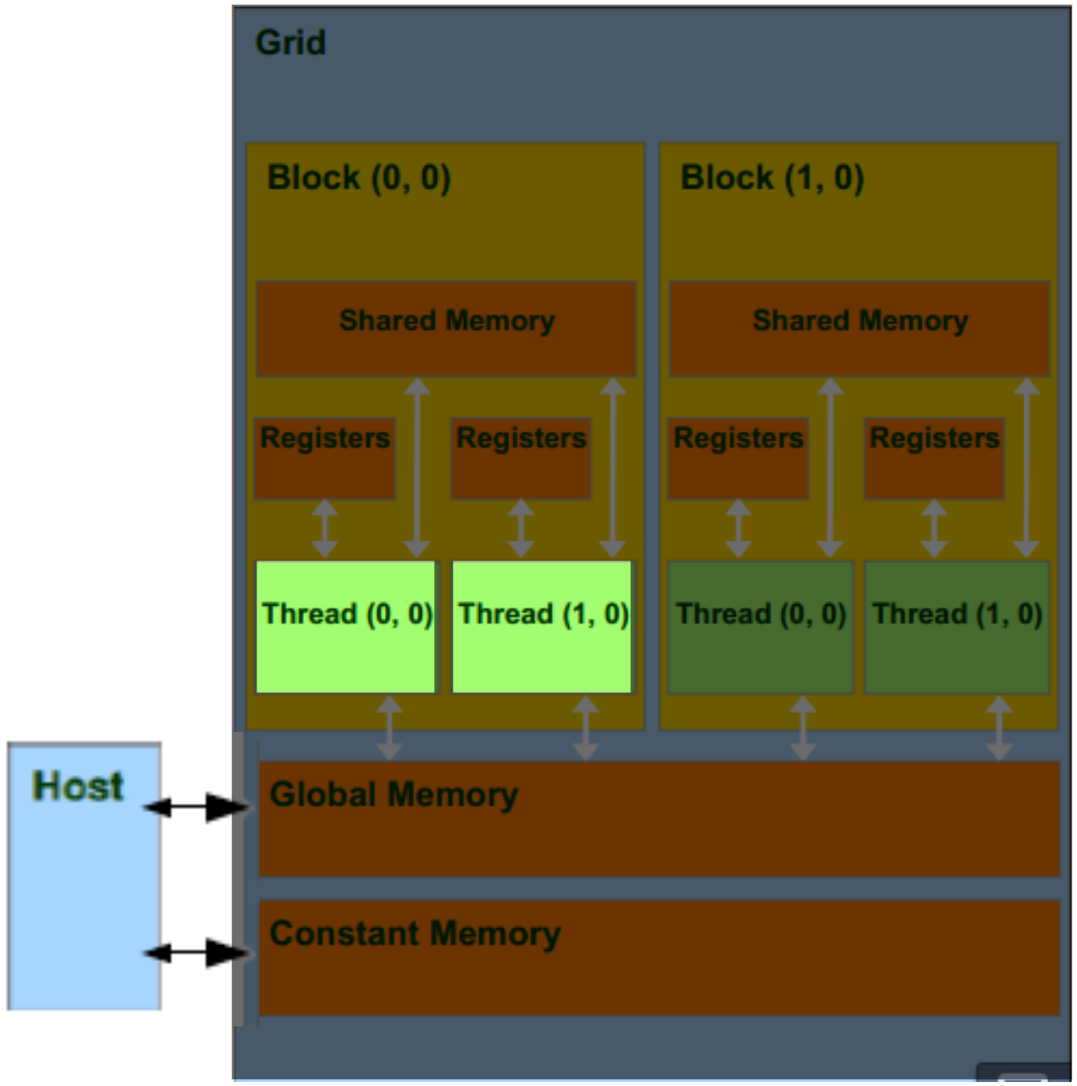


All the clusters of one bin on layer 1 are copied in the shared memory and a for loop starts over these clusters looking for good doublets

First porting of the doublet finder on OpenCL



OpenCL reference



Factor 4 speed-up on 8 threads CPU, still no acceleration on GPU due to the memory transfer overhead.

Conclusions

- ALICE is running GPU based online track reconstruction in the TPC since 2010
  - ✓ Overall good performance and no major issue in the operation!
- From Run3 on we will run also the reconstruction of tracks and vertices in the Inner Tracking System
  - ✓ Feasibility proven by first results
  - ✓ Ongoing effort for the optimisation and porting on heterogeneous architectures
- Key ingredient for the used algorithms:
  - ✓ Combinatorics rejection based on local information