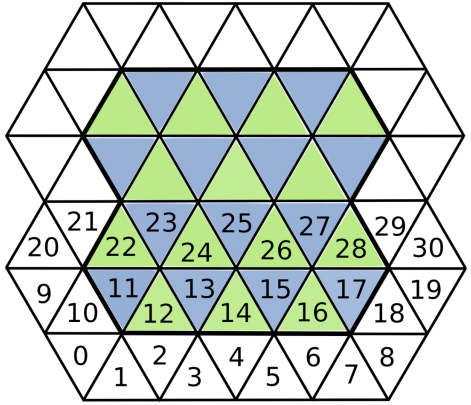
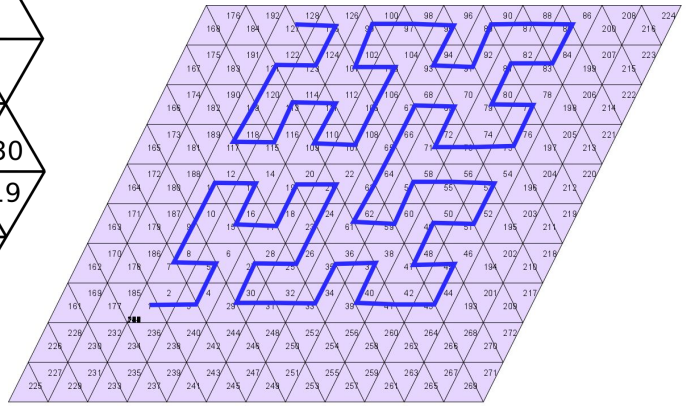
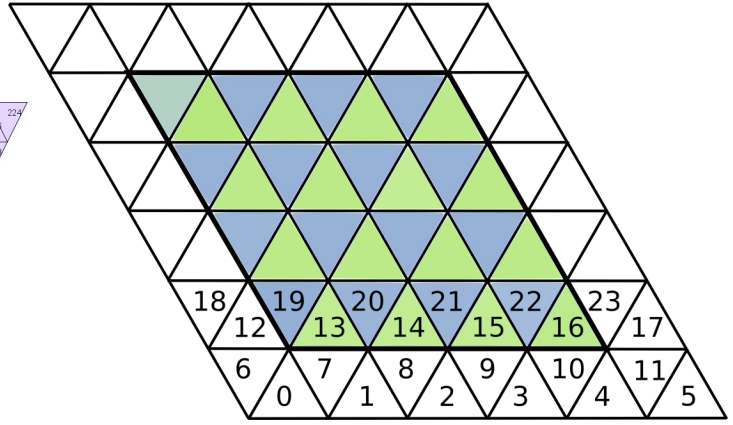
# Optimization: indexing patterns




Row Major



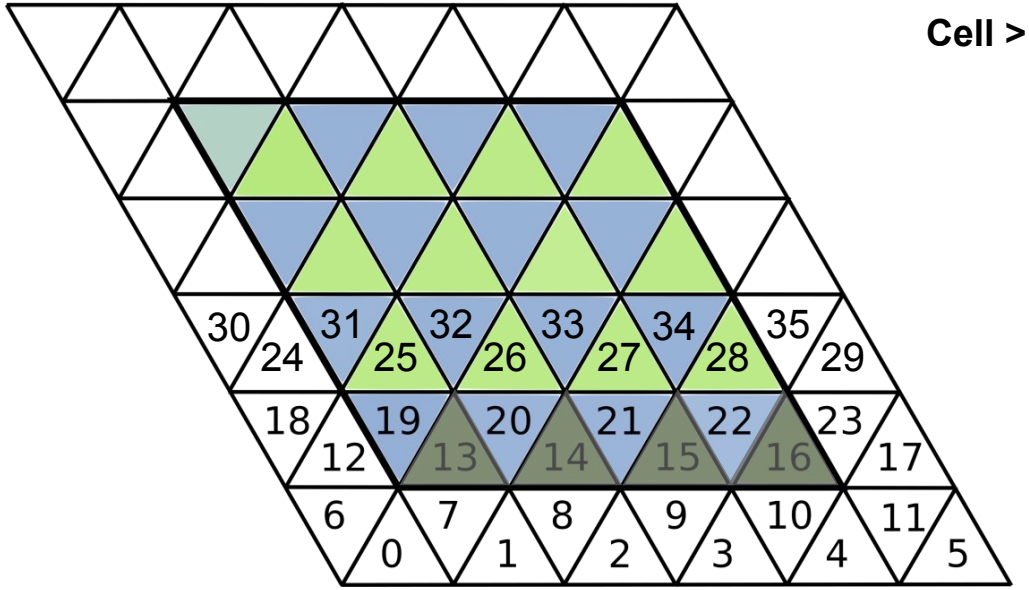
Space Filling Curve



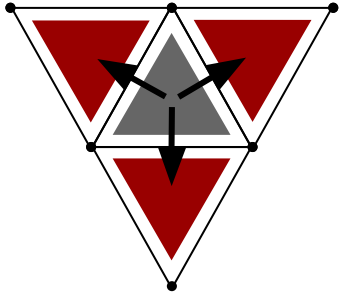

Structured Numbering

Order of elements of the dense dimension impacts performance when accessing neighbors. It has implications on locality (thus cache efficiency) and the overall number of memory transactions.

# Optimization: indexing patterns

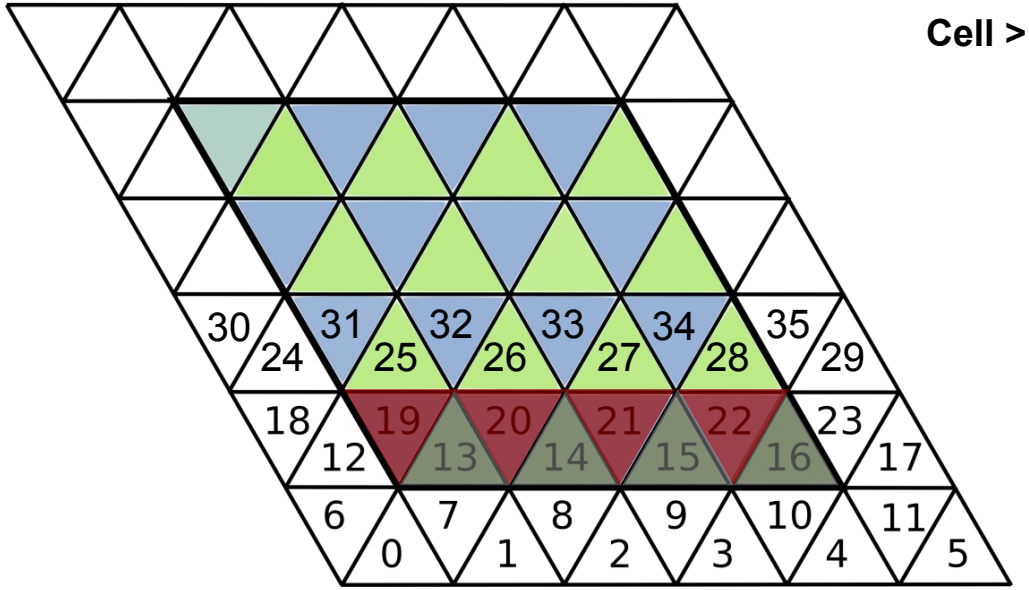



**Cell > Edge > Cell**

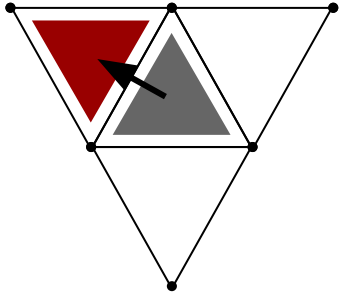


Structured numbering. Always coalesced accesses, worst locality.

# Optimization: indexing patterns

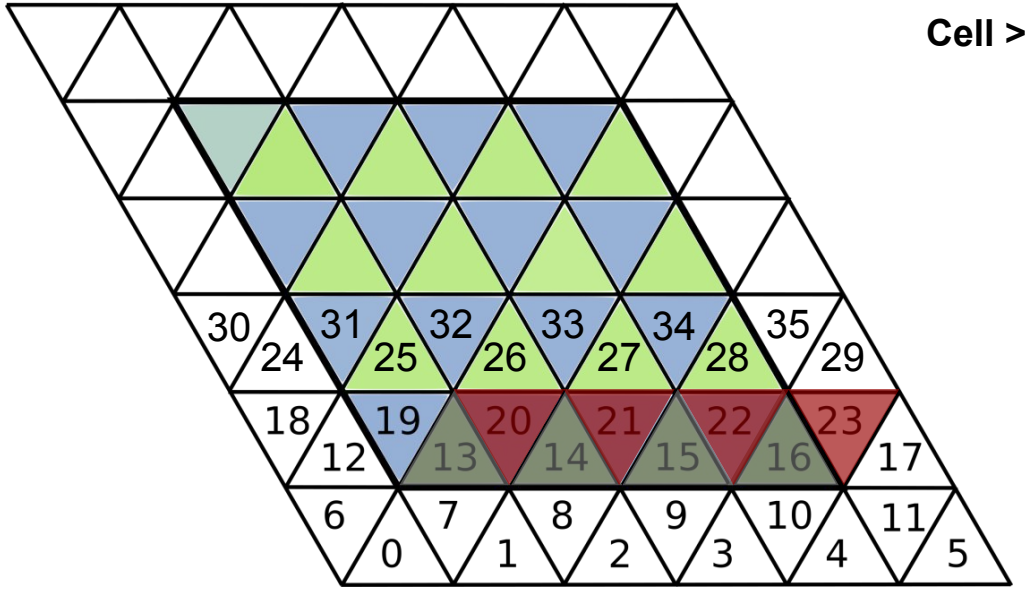



**Cell > Edge > Cell**

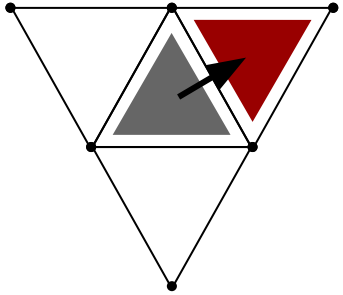


Structured numbering. Always coalesced accesses, worst locality.

# Optimization: indexing patterns

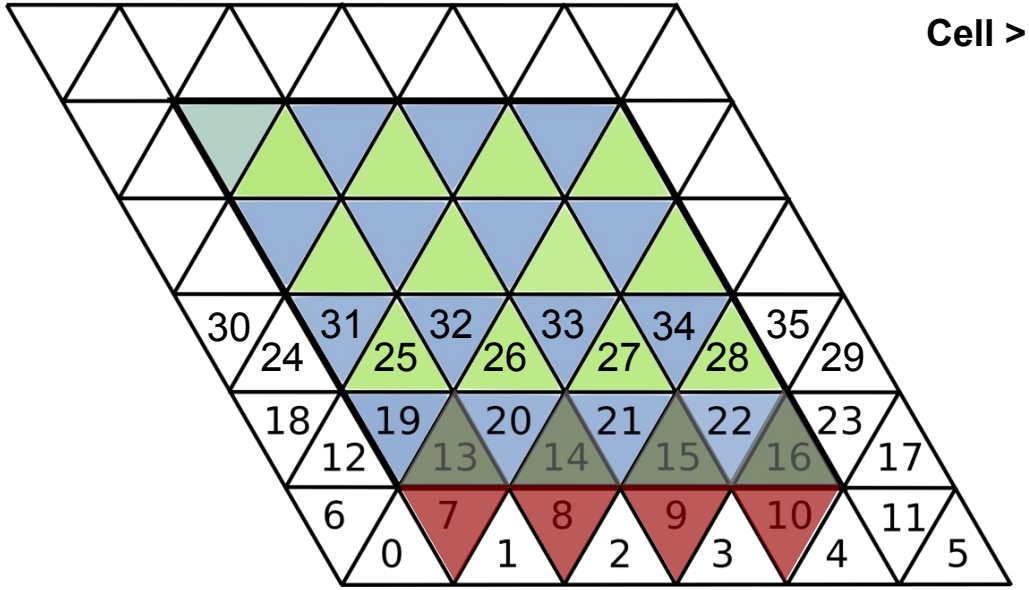



**Cell > Edge > Cell**

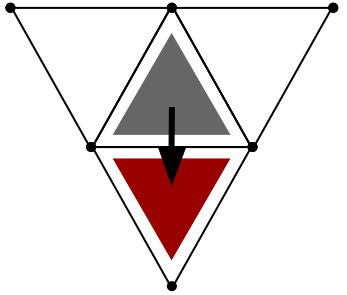


Structured numbering. Always coalesced accesses, worst locality.

# Optimization: indexing patterns

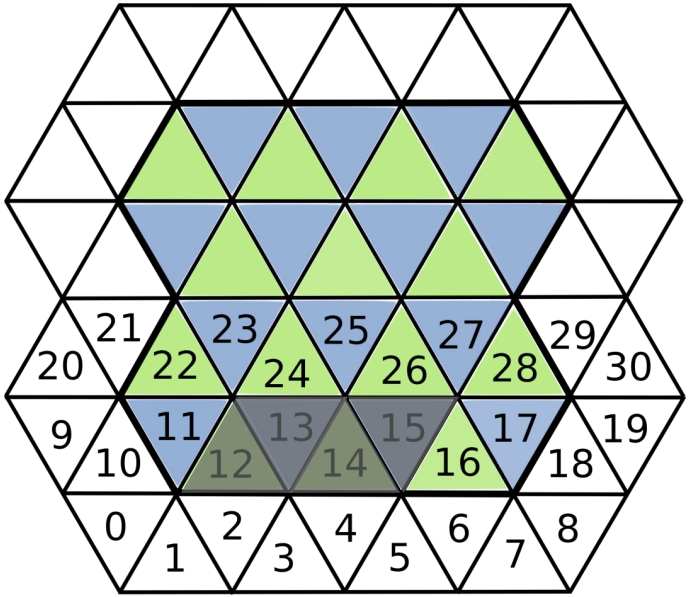



**Cell > Edge > Cell**

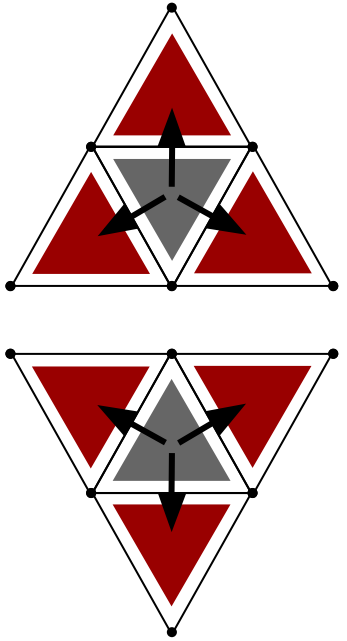


Structured numbering. Always coalesced accesses, worst locality.

# Optimization: indexing patterns

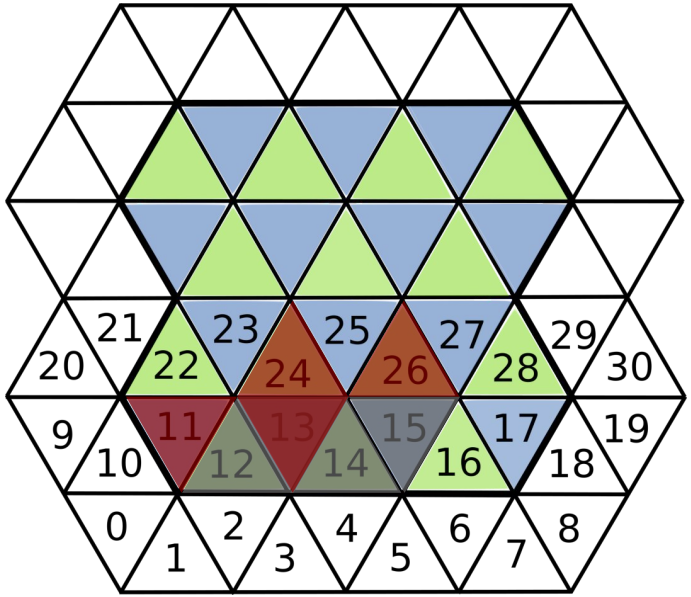



**Cell > Edge > Cell**

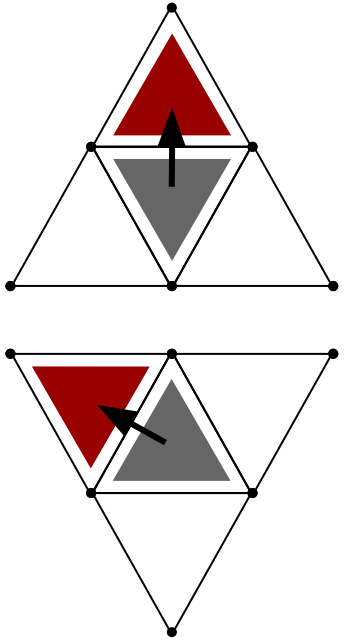


Row Major numbering. Compromise between access coalescing and locality.

# Optimization: indexing patterns

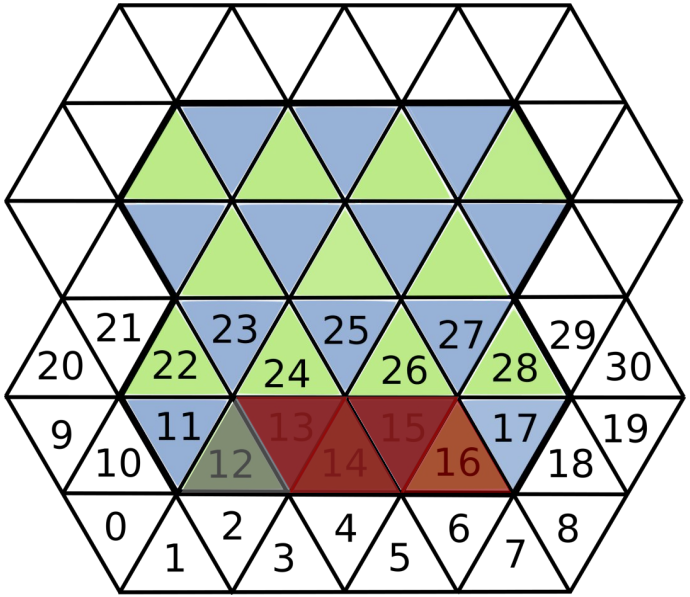



**Cell > Edge > Cell**

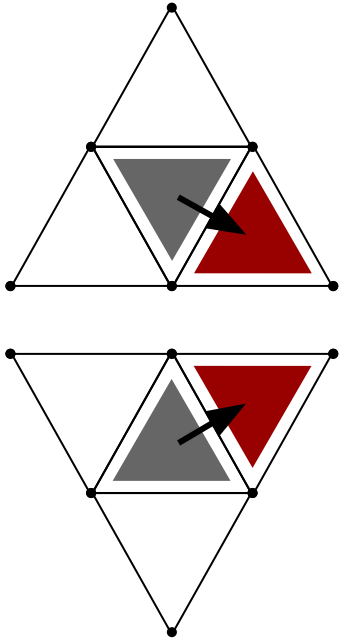


Row Major numbering. Compromise between access coalescing and locality.

# Optimization: indexing patterns

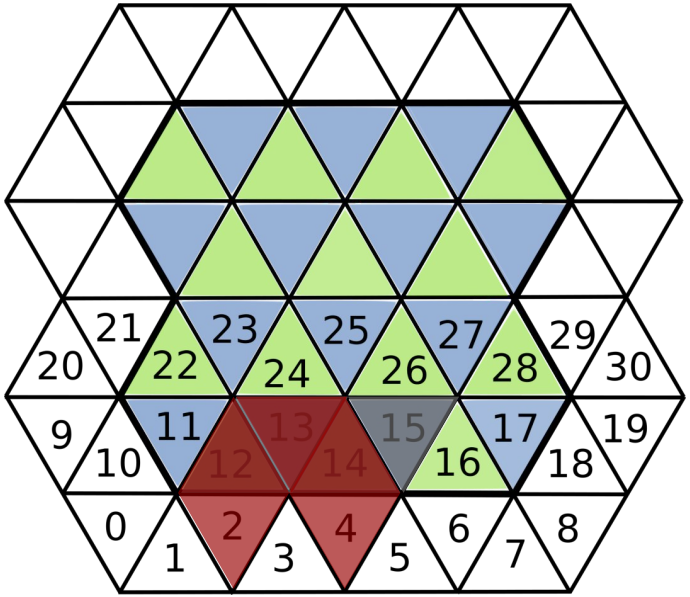



**Cell > Edge > Cell**

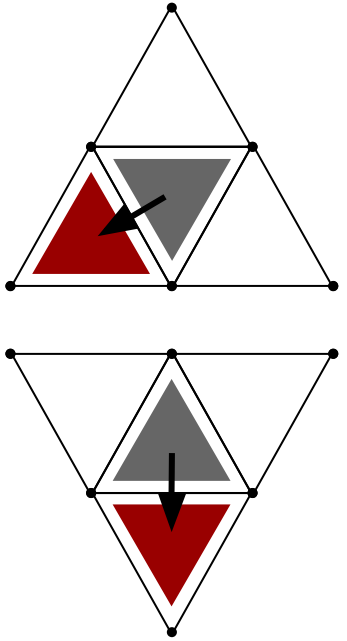


Row Major numbering. Compromise between access coalescing and locality.

# Optimization: indexing patterns

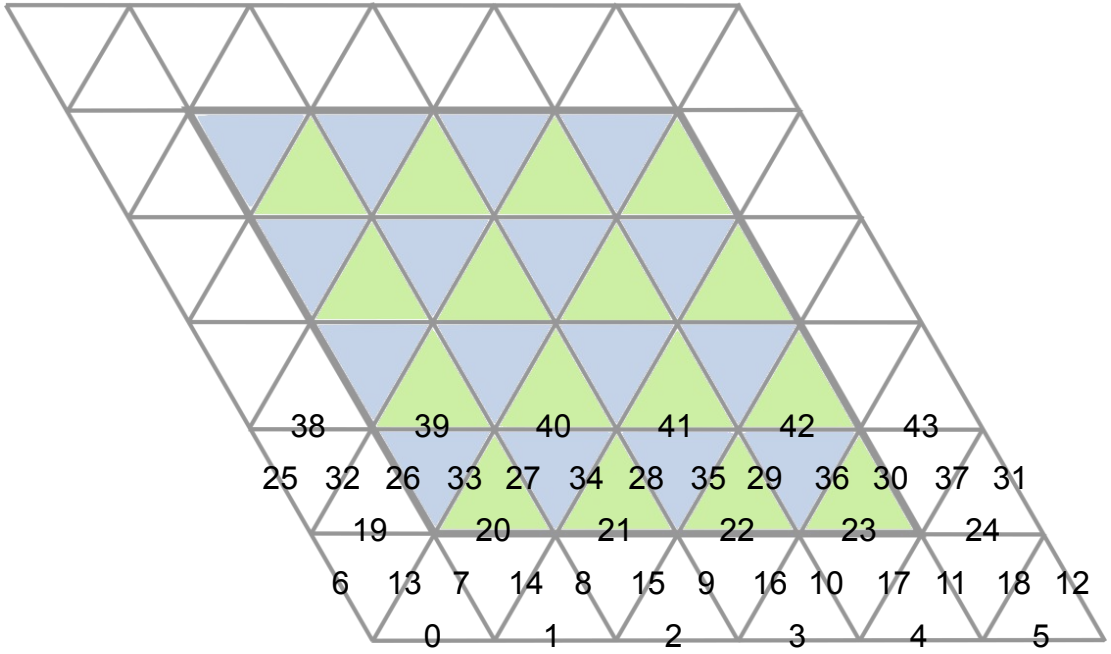



**Cell > Edge > Cell**



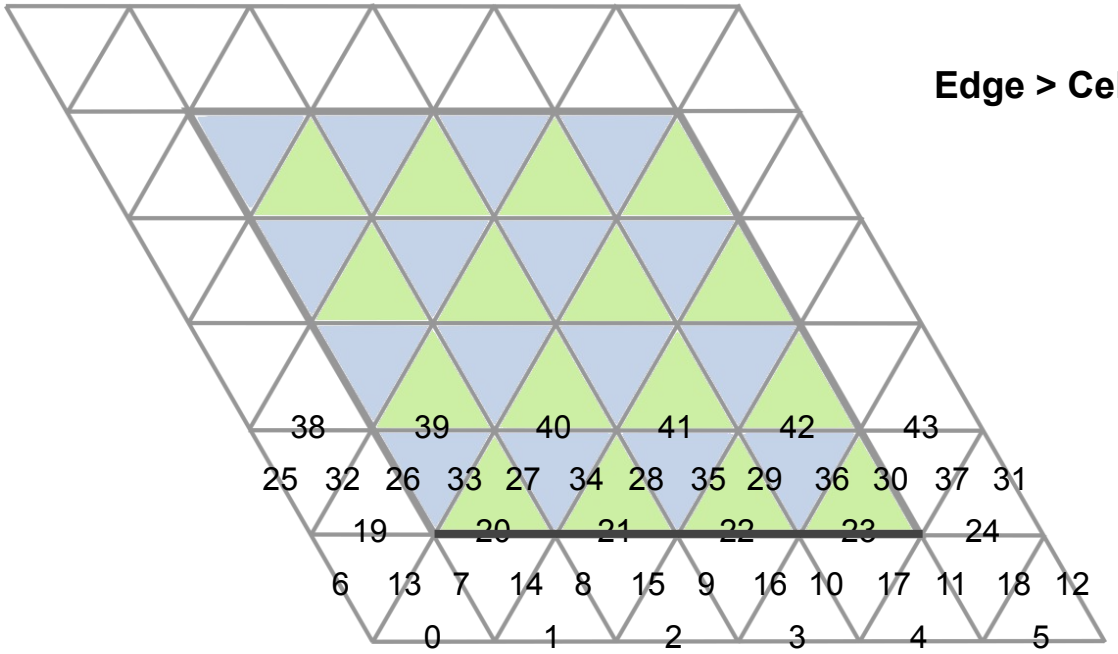
Row Major numbering. Compromise between access coalescing and locality.

# Optimization: indexing patterns

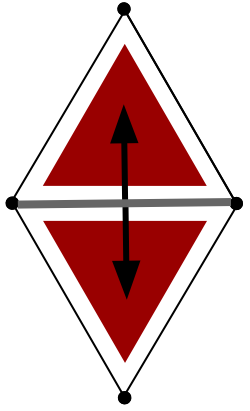



Structured numbering. Always coalesced accesses, worst locality.

# Optimization: indexing patterns

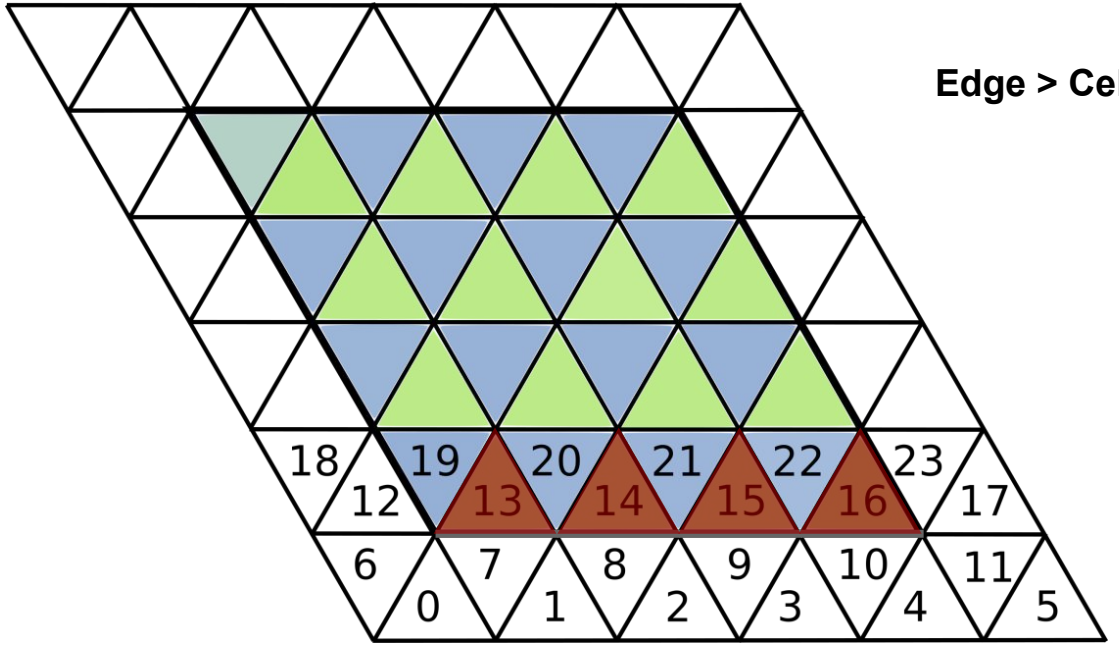



**Edge > Cell**

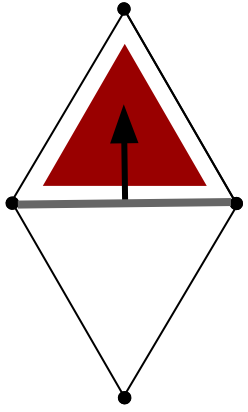


Structured numbering. Always coalesced accesses, worst locality.

# Optimization: indexing patterns

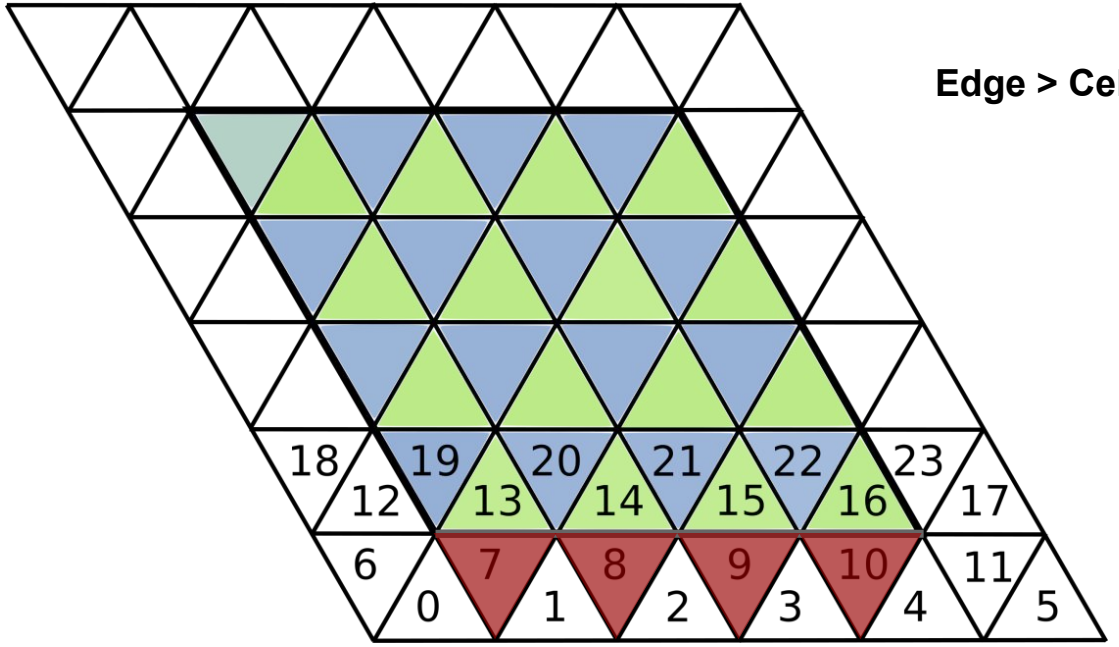



**Edge > Cell**

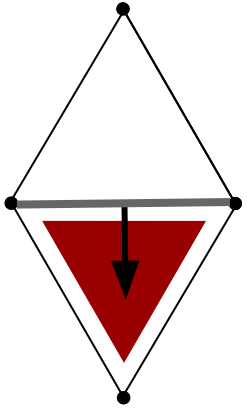


Structured numbering. Always coalesced accesses, worst locality.

# Optimization: indexing patterns

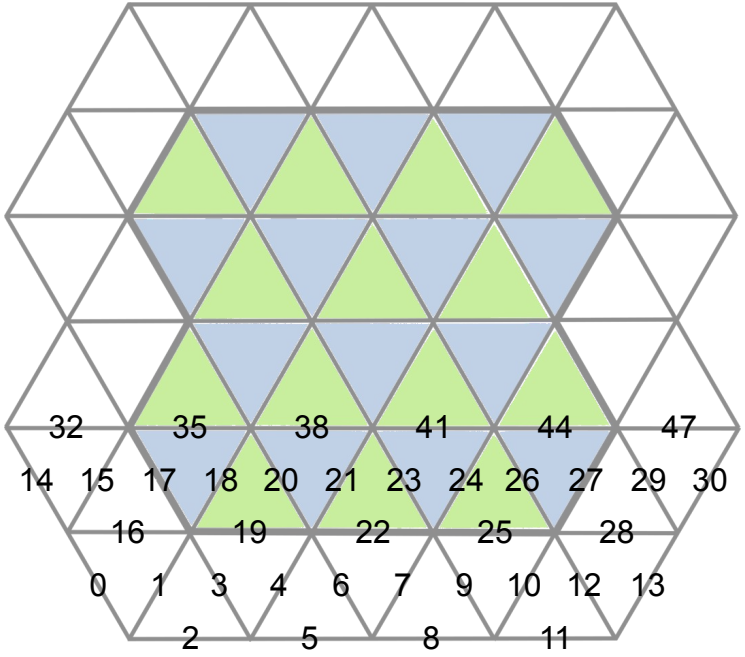



**Edge > Cell**



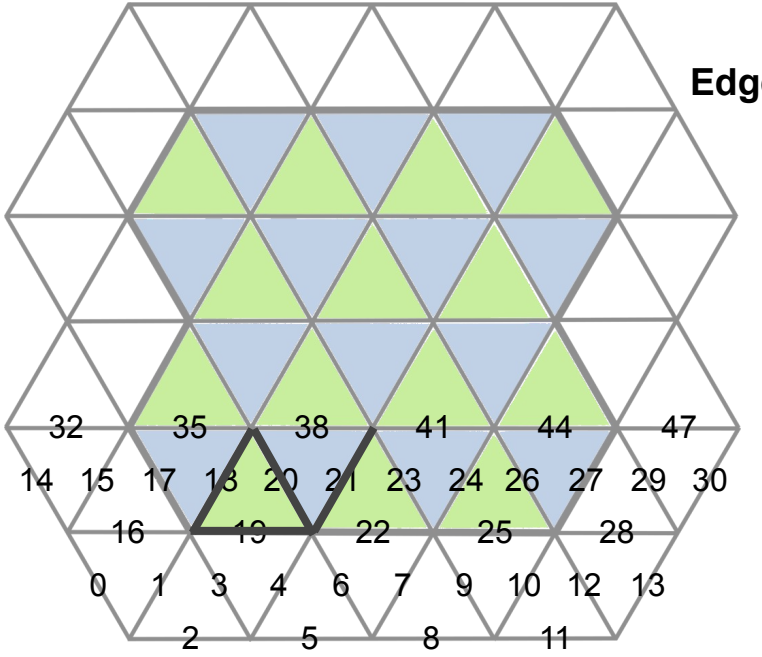
Structured numbering. Always coalesced accesses, worst locality.

# Optimization: indexing patterns

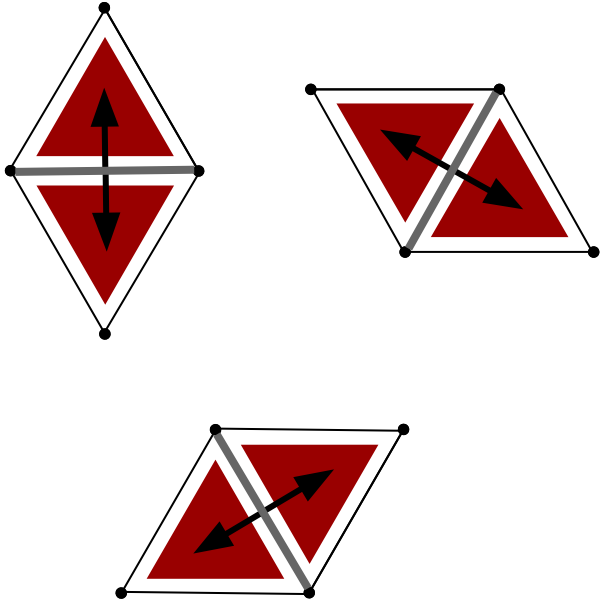



Row Major numbering. Compromise between access coalescing and locality.

# Optimization: indexing patterns

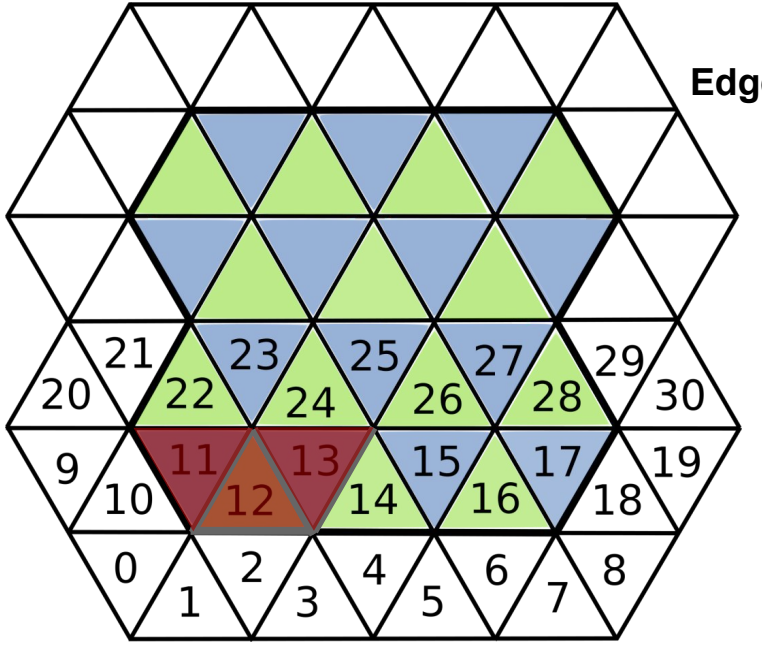


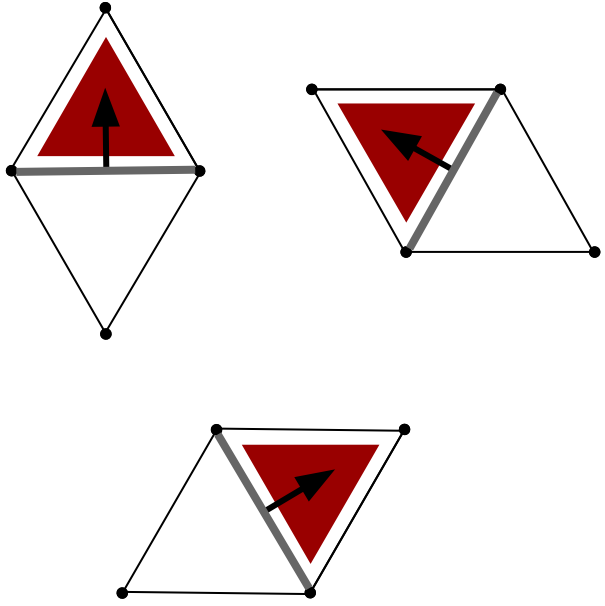

**Edge > Cell**



Row Major numbering. Compromise between access coalescing and locality.

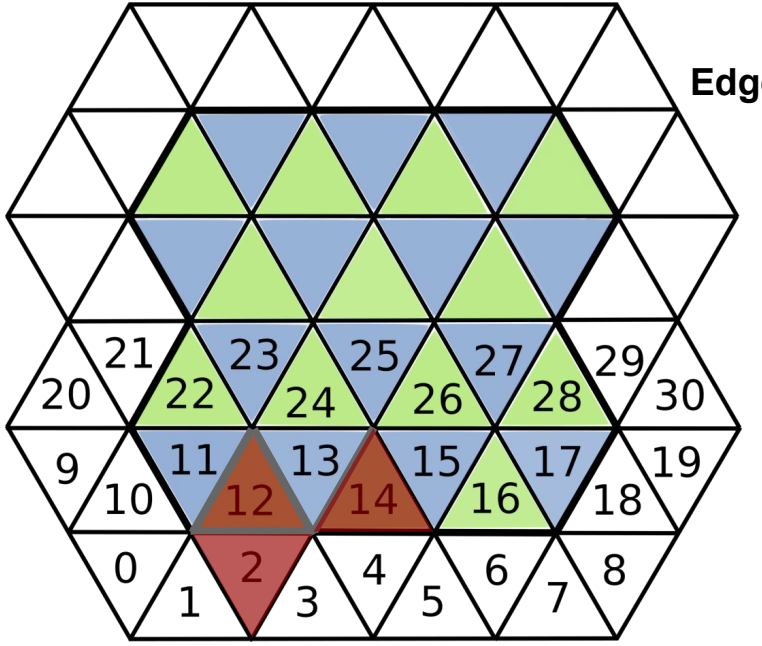
# Optimization: indexing patterns

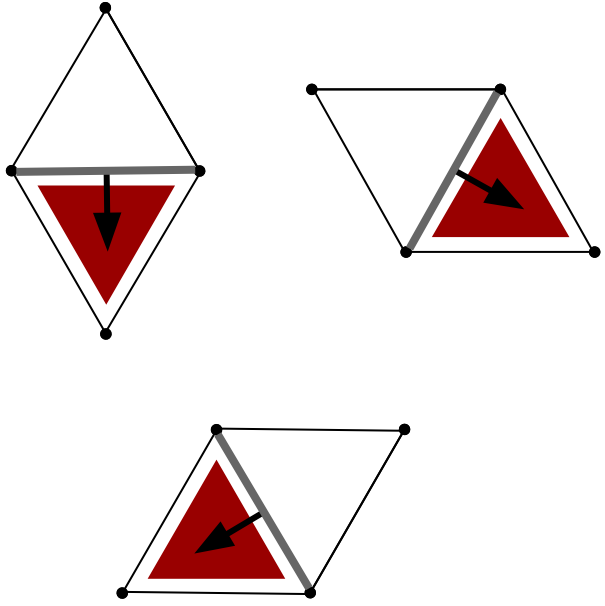


Row Major numbering. Compromise between access coalescing and locality.

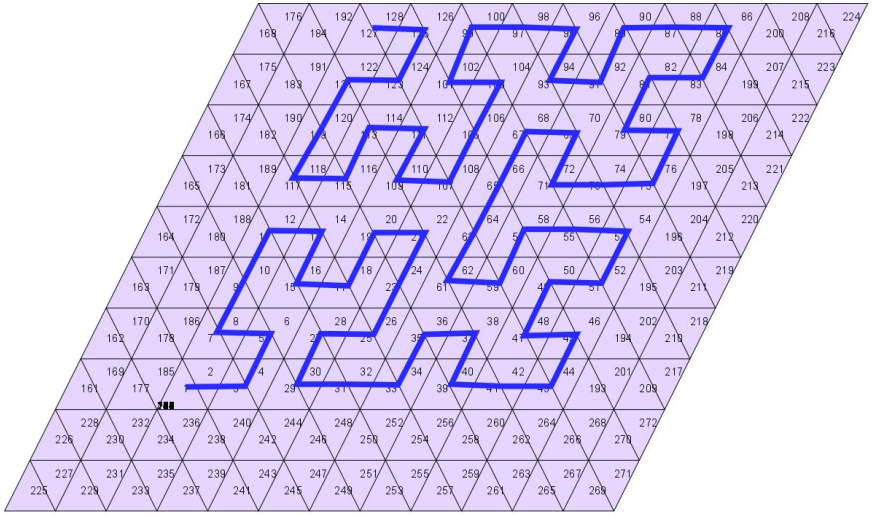
# Optimization: indexing patterns



Row Major numbering. Compromise between access coalescing and locality.

# Optimization: indexing patterns



A space filling curve provides the maximum data locality, to the benefit of cache efficiency. However access coalescing is almost absent.

# Case Study: ICON's "diamond" stencil

- Code extracted from ICON's dycore computing a Laplacian and a Smagorinsky coefficient.
- Only one local neighborhood is used: **Edge > Cell > Vertex** (graphically looks like a *diamond*).
- Taking timings with a 340x340x80 grid, i.e. ~174k edges and 80 k-levels
- 1 NVIDIA V100, compiling with CUDA Toolkit release 10.1
- Baseline of 13 CUDA kernels
- Manually applying optimizations one at a time
- Keep in mind that it's a single, limited example. Other stencils might not give the same results.

# Case Study: ICON's "diamond" stencil

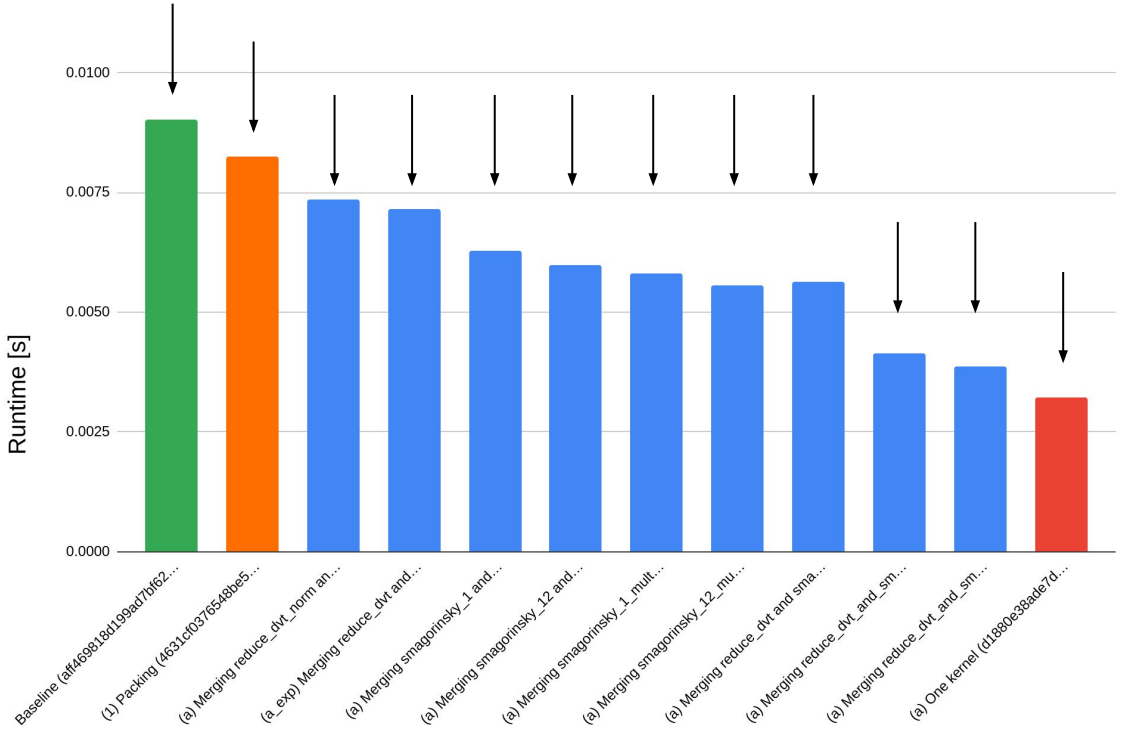
Baseline

Pack vectors

Fuse dense loops

Fuse reductions

Fuse reductions and sparse loop



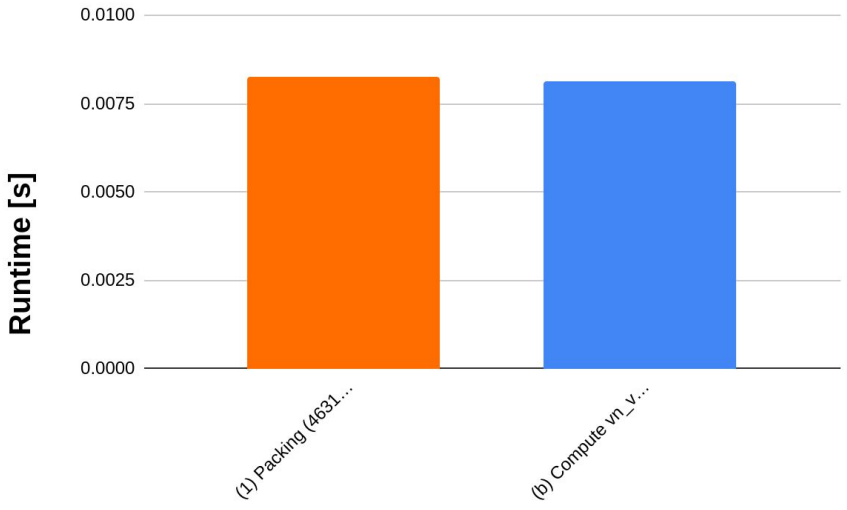

Std dev of measurements around $10^{-5}$

# Case Study: ICON's "diamond" stencil

*Recurrent stencil inlining* of sparse temporary field.

Computation is a dot product between 2 vectors and its result is required by 3 kernels.

Very little improvement, maybe an unlucky case.



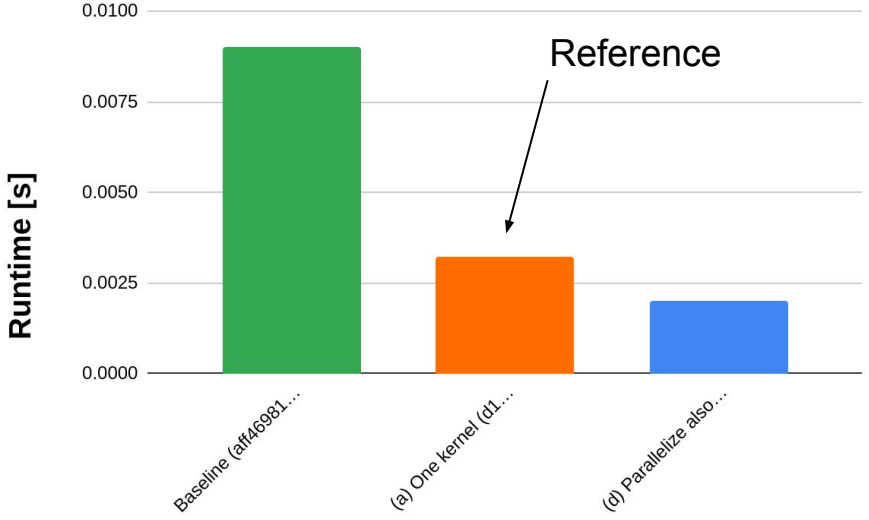
# Case Study: ICON's "diamond" stencil

Parallelizing also the k-loop.

Great improvement despite the fact that number of edges (and thus of threads before the opt.) is very big.

Improvement due to a better warp scheduling.

Avg inst/cycle almost doubled.

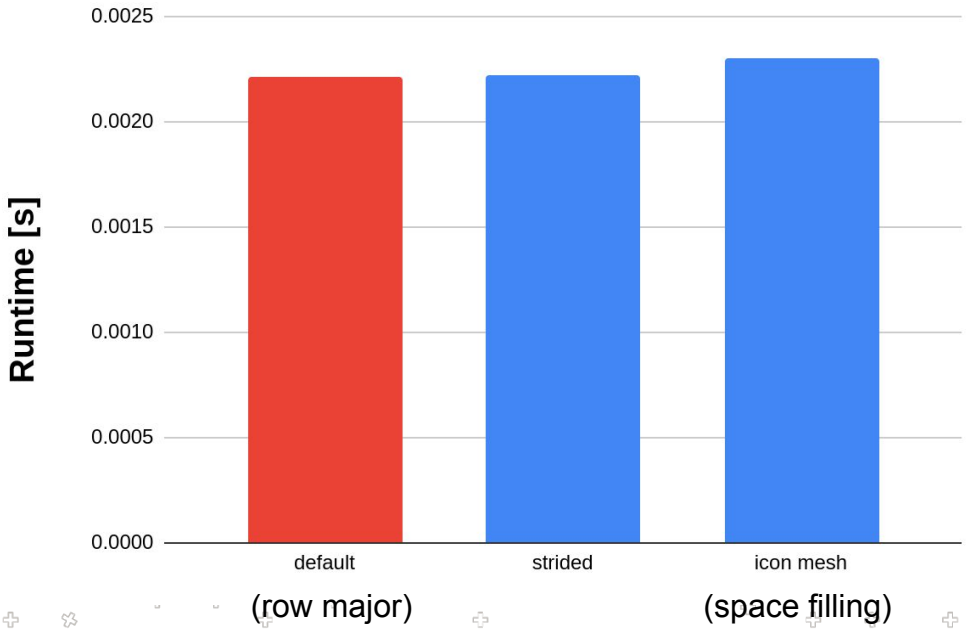


# Case Study: ICON's "diamond" stencil

Trying different indexing patterns.

Space filling curve pattern is ICON's one. Performing slightly worse than the others.

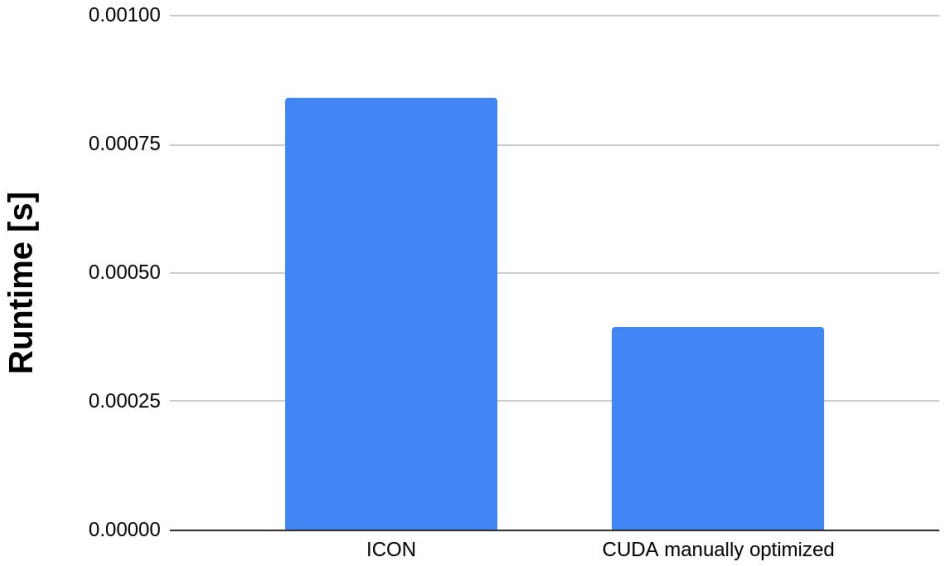
Overall, differences not very noticeable.



# Case Study: ICON's "diamond" stencil

P100, ~28k edges, 64 k-levels

All optimizations combined (packing + fusing + parallelize k-loop + row-major indexing) vs ICON OpenACC original stencil performance.



# State of Dawn's optimizer

Currently supports:

- Fusing dense "loops"
- Parallelizing k-loops

To be added:

- Fusing reductions and sparse loops
- One-time stencil inlining
- Recurrent stencil inlining
- Vector packing

Indexing patterns are implicit in the fields' storages, which are provided externally (transparent to Dawn).

# Outlook

- Results got so far are promising
- There's still a lot to experiment: trying other stencils, testing all the optimizations devised and coming up with others
- Dawn's optimizer is still work in progress, e.g. need appropriate data structures to represent fusion of reductions/sparse loops
- Still need to consider splitting the compute domain to run on several GPUs/nodes, halo exchanges and so on...

# Q&A

Questions?

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Swiss Confederation

Federal Department of Home Affairs FDHA
**Federal Office of Meteorology and Climatology MeteoSwiss**

**MeteoSwiss**
Operation Center 1
CH-8058 Zurich-Airport
T +41 58 460 91 11
www.meteoswiss.ch

**MeteoSvizzera**
Via ai Monti 146
CH-6605 Locarno-Monti
T +41 58 460 92 22
www.meteosvizzera.ch

**MétéoSuisse**
7bis, av. de la Paix
CH-1211 Genève 2
T +41 58 460 98 88
www.meteosuisse.ch

**MétéoSuisse**
Chemin de l'Aérologie
CH-1530 Payerne
T +41 58 460 94 44
www.meteosuisse.ch