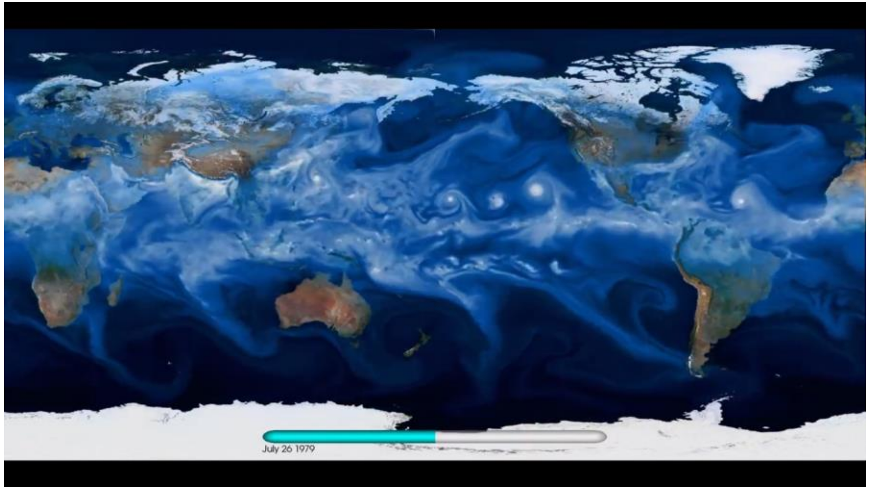
PROJECT DISCO: PHYSICS-BASED DISCOVERY OF COHERENT STRUCTURES IN SPATIOTEMPORAL SYSTEMS

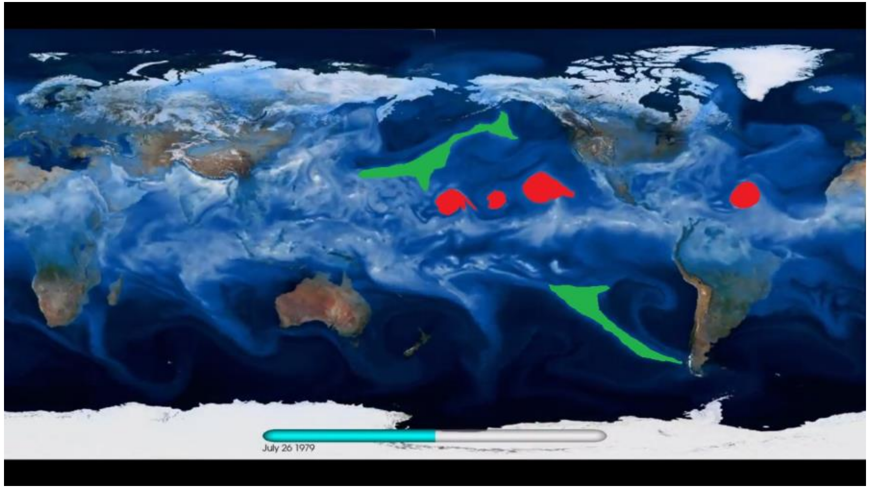
Adam Rupe[*], Karthik Kashinath[†], **Nalini Kumar**[§], James P. Crutchfield[*], Ryan G. James[*], Mr Prabhat[†], Victor Lee[§]

[*] University of California Davis
[†] Lawrence Berkeley National Laboratory
[§] Intel Corporation

Overview of DisCo

Goal: Unsupervised detection of spatiotemporal structures in climate data





Overview of DisCo

Deep learning is quickly becoming a standard approach for such analyses

- But, climate data is highly complex and unlabeled
- Interpretability and detection of new mechanisms are key to scientific discovery

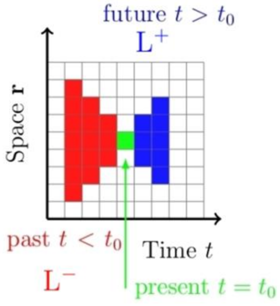
Our solution: Coherent Discovery (DisCo) – Physics based machine learning

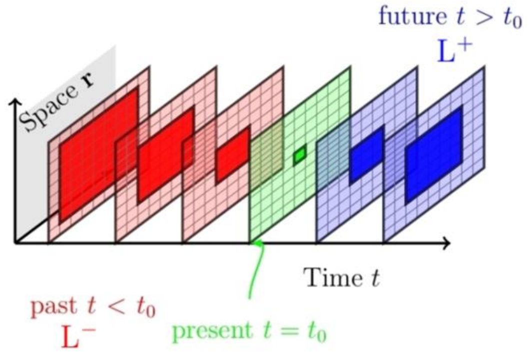
- Unsupervised approach that exploits the causal nature of spatiotemporal data sets generated by local dynamics (e.g. hydrodynamic flows).
- Can be used to discover novel patterns and coherent structures in data

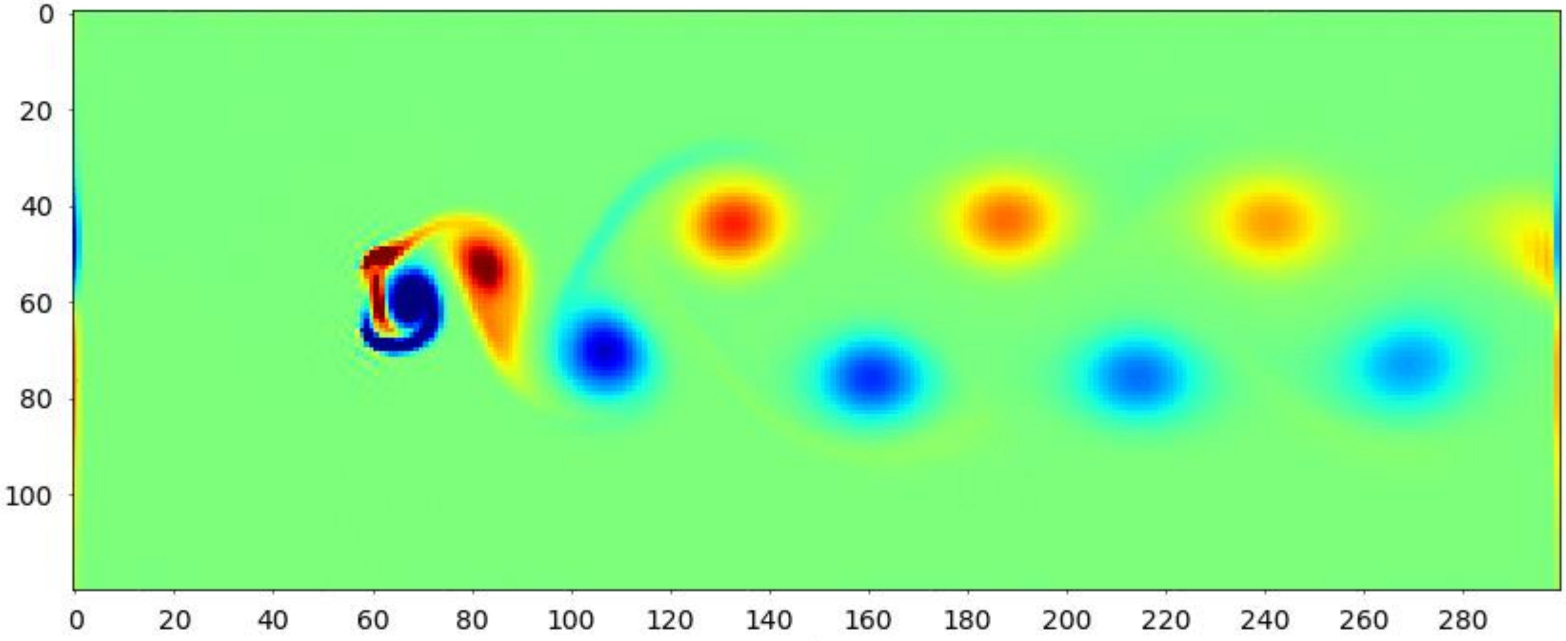
DisCo Algorithm

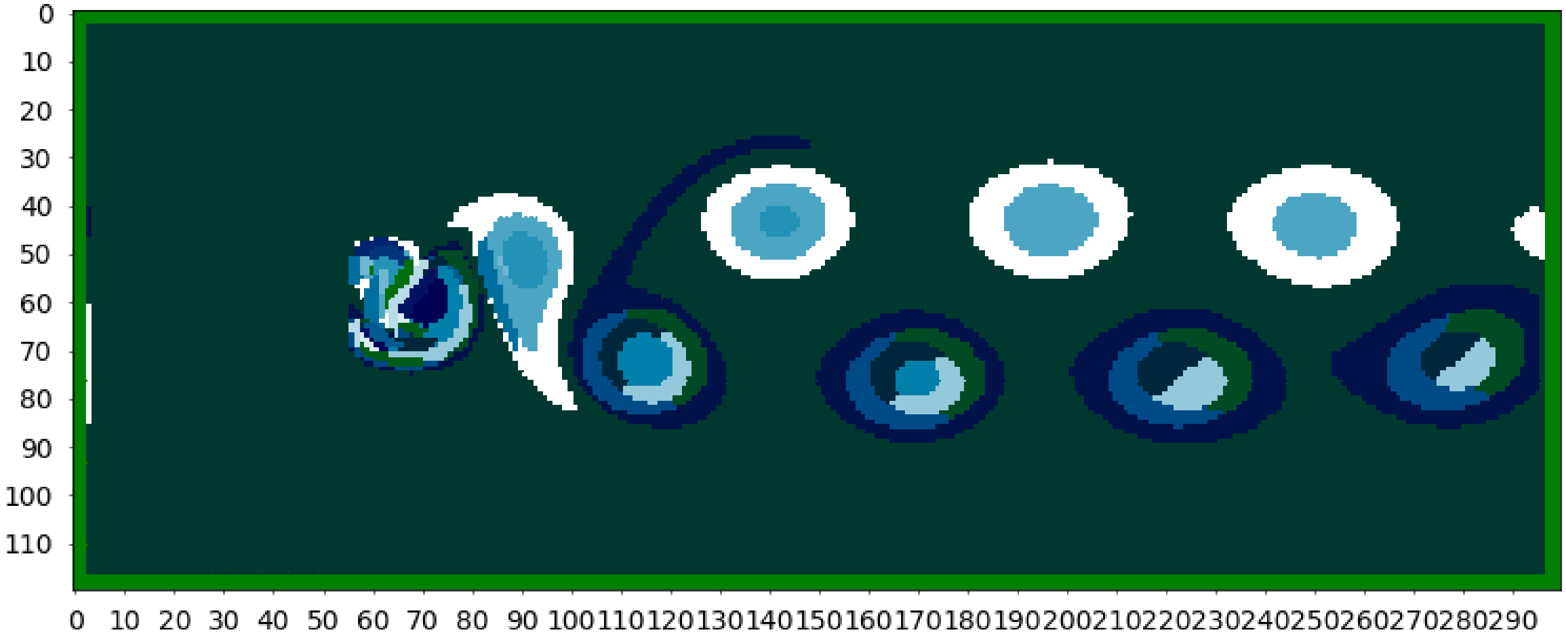
0. Extracting light cones from data
1. Clustering stage 1
2. Building conditional distributions
3. Clustering stage 2
    1. Clustering over distributions (labels)
    2. 1000-1M labels
4. Causal filtering using epsilon-machines: embarrassingly parallel
5. Last step: Community detection on a weighted graph of O(M) nodes

Lightcones are used as notions of past and future






0
20
40
60
80
100
0
20
40
60
80
100
120
140
160
180
200
220
240
260
280



0
10
20
30
40
50
60
70
80
90
100
110
0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180 190 200 210 220 230 240 250 260 270 280 290

Challenges

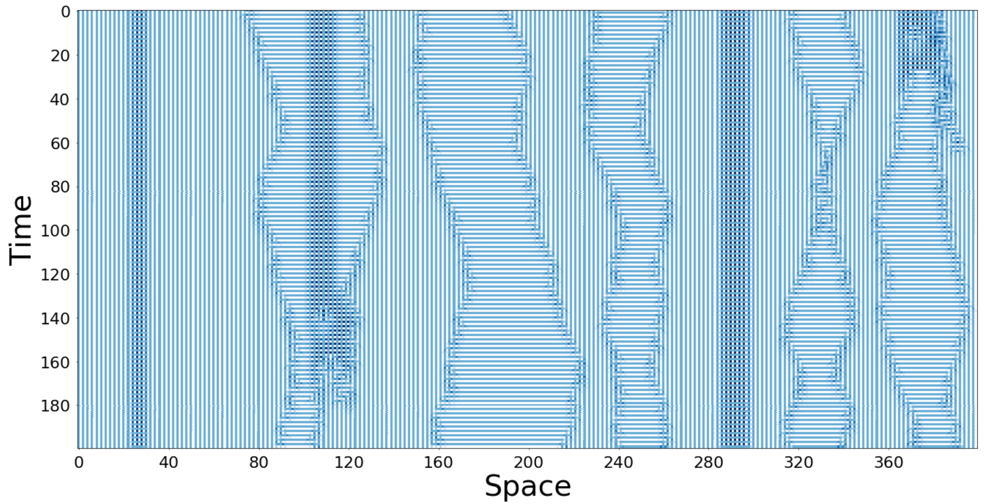
Need an unsupervised clustering method for lightcone clustering that

- Works well for O(100)-O(10,000) dimension data
- Works well for ~ 100 TB simulation data resulting in O(1M) – O(100M) data
- Can be efficiently parallelized to O(1,000) – O(10,000) nodes
- Create optimized implementation of clustering algorithm - Python is the primary language for development

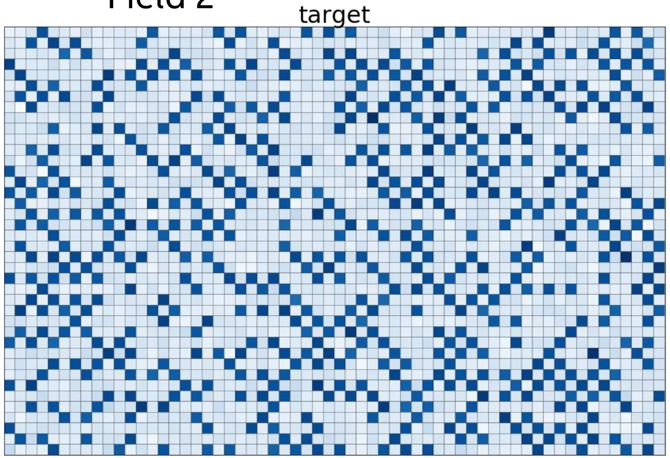
- E.g. Coupled Map Lattice field
  - 200 1D spatial field, 400 time; 80,000 – 4,000,000 data points (625kB – 30GB)
  - Lightcone depth of 3 implies each lightcone vector has dimensionality 9

Clustering Method Evaluation

Field 1



Field 2

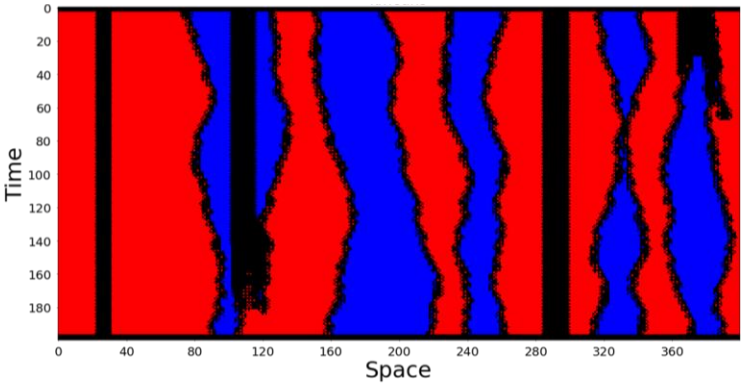


- circle map lattice with r = 1.0 and c = 1.0
- random initial conditions
- two background regions; horizontal and vertical stripes
- coherent structures: overlap and interfaces of backgrounds

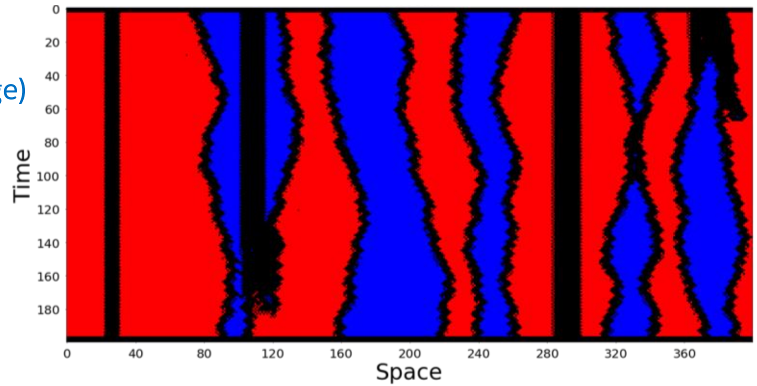
- stochastic checkerboard with Gaussian noise
- first clustering step should account for noise
- second clustering step should account for stochasticity and return a normal checkerboard

Clustering Results – Field 1

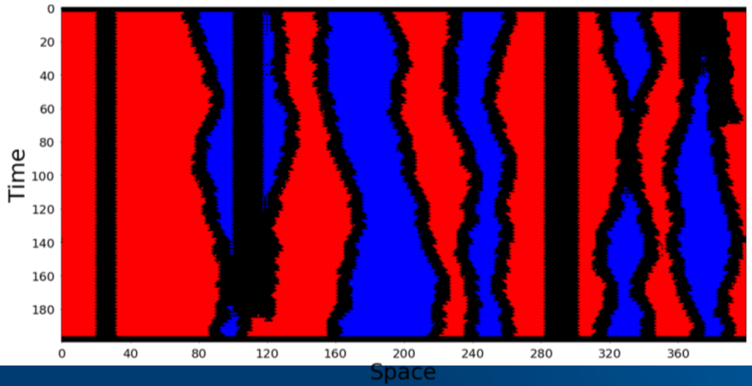
K-means
(scikit-learn,
Intel Python)



HDBSCAN
(conda-forge)

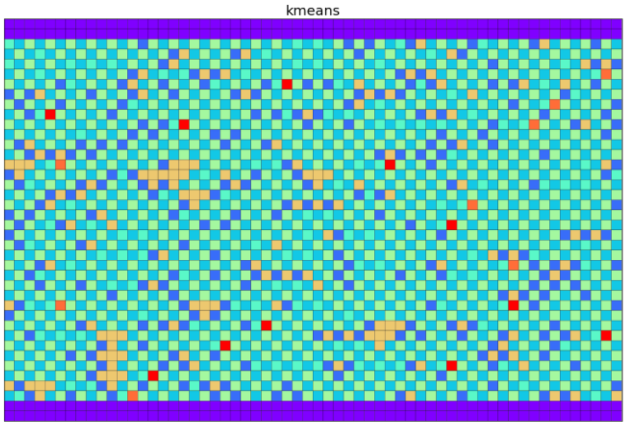


DBSCAN
(conda-forge)

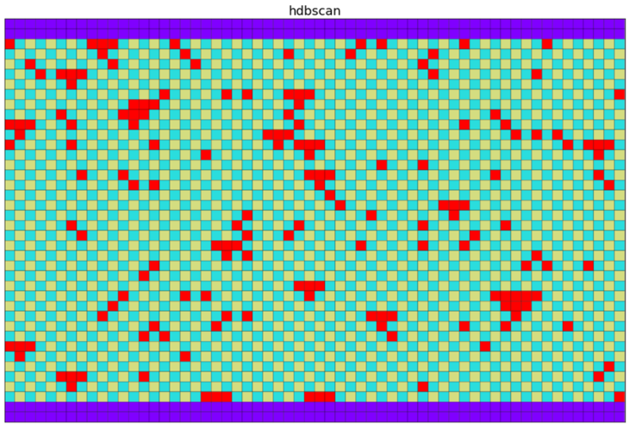


Clustering Results – Field 2

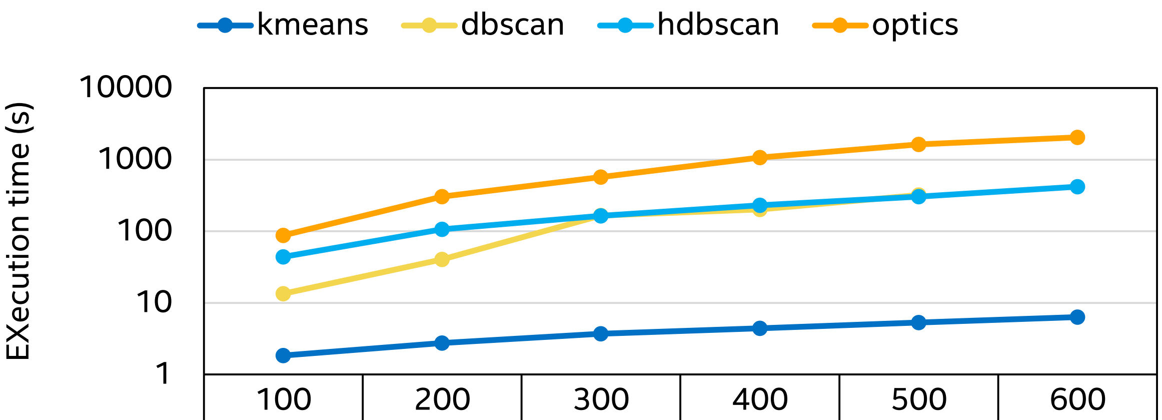
K-means (scikit-learn, Intel Python)



HDBSCAN (conda-forge)



Scaling w/ no. of Lightcones

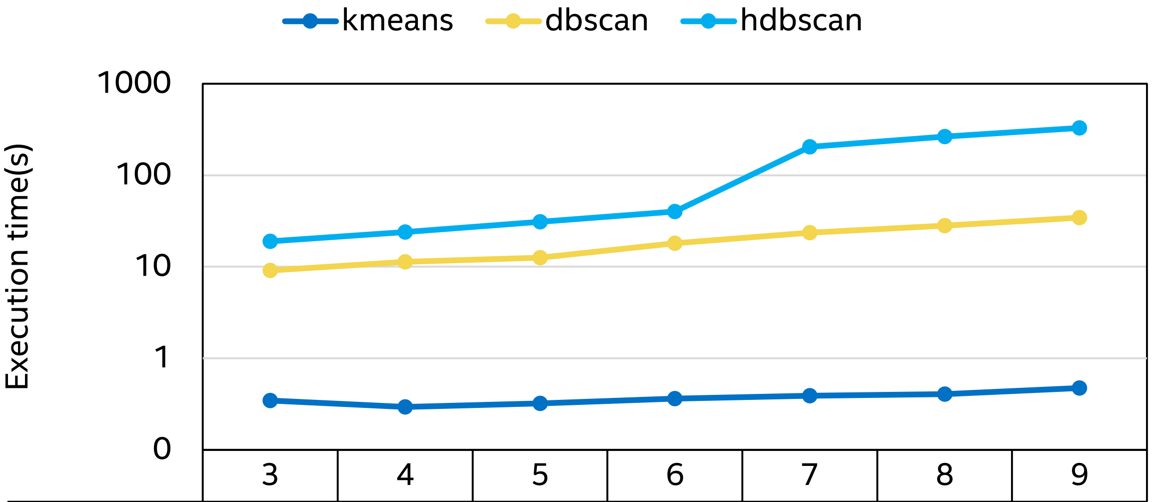


	100	200	300	400	500	600
kmeans	1.844	2.761	3.713	4.414	5.295	6.354
dbscan	13.456	40.649	166.366	202.254	319.809	0.000
hdbscan	43.779	106.108	164.566	229.534	305.531	418.350
optics	87.636	305.750	573.513	1068.640	1626.016	2043.050

Target field (200 x S)

Configuration:
- Intel® Xeon Phi™ 7250 processor ("Knights Landing")
- Intel® Distribution of Python* (3.6)
- Intel® DAAL 2018.0.2.20180124
- Numba 0.36.2
- Scikit-learn 0.19.1
- Hdbscan 0.8.16
- OPTICS (scikit-learn v0.21.dev0)

Scaling w/ Lightcone Depth



	3	4	5	6	7	8	9
kmeans	0.345	0.295	0.320	0.362	0.390	0.406	0.474
dbscan	9.075	11.345	12.601	17.986	23.450	28.060	34.275
hdbscan	18.959	23.958	31.017	40.130	204.200	264.593	327.066

lightcone depth

Configuration:
- Intel® Xeon™ Processor E5-2698 v3 ("Haswell")
- Intel® Distribution of Python* (3.6)
- Intel® DAAL 2018.0.2.20180124
- Numba 0.36.2
- Scikit-learn 0.19.1
- Hdbscan 0.8.16
- OPTICS (scikit-learn v0.21.dev0)

References

1. Rupe, Adam, and James P. Crutchfield. "Local causal states and discrete coherent structures.“ Chaos: An Interdisciplinary Journal of Nonlinear Science 2018 28:7

2. Rupe, Adam, James P. Crutchfield, and Karthik Kashinath. "A Physics-Based Approach to Unsupervised Discovery of Coherent Structures in Spatiotemporal Systems." arXiv preprint arXiv:1709.03184 (2017).

3. Rupe, Adam, and James P. Crutchfield. "Computational Mechanics of Coherent Structures in Spatiotemporal Systems." Bulletin of the American Physical Society 61 (2016).

Notices and Disclaimers

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer.

NOTICES AND DISCLAIMERS

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration.

No computer system can be absolutely secure.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. For more complete information about performance and benchmark results, visit http://www.intel.com/benchmarks .

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit http://www.intel.com/benchmarks .

Intel® Advanced Vector Extensions (Intel® AVX)* provides higher throughput to certain processor operations. Due to varying processor power characteristics, utilizing AVX instructions may cause a) some parts to operate at less than the rated frequency and b) some parts with Intel® Turbo Boost Technology 2.0 to not achieve any or maximum turbo frequencies. Performance varies depending on hardware, software, and system configuration and you can learn more at http://www.intel.com/go/turbo.

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Performance results are based on testing as of September 21, 2018 and may not reflect all publicly available security updates. See configuration disclosure for details.

Configurations: Testing on Cori at NERSC was performed with spectre_v1 and meltdown patches.

Cost reduction scenarios described are intended as examples of how a given Intel-based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

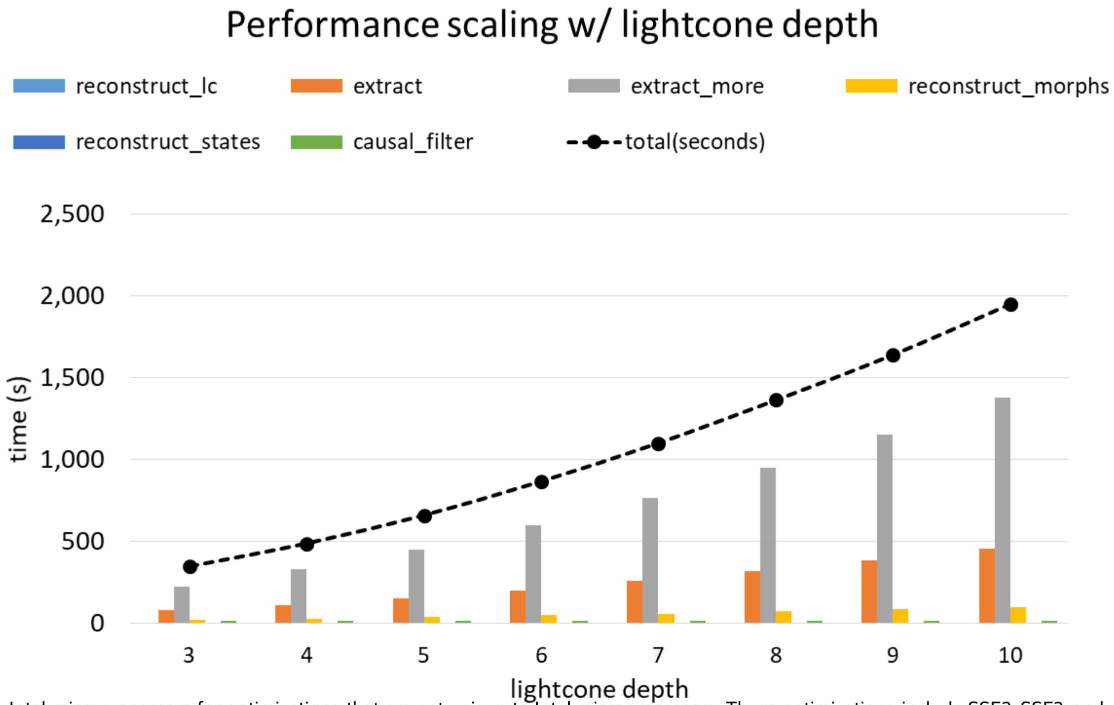
Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.
Intel, the Intel logo, and Intel Xeon are trademarks of Intel Corporation in the U.S. and/or other countries.
*Other names and brands may be claimed as property of others.

intel®

Preparing for Advanced Science Datasets

- Lattice Boltzmann simulations might need deeper lightcones
- DBSCAN/HDBSCAN is prohibitively slow on a single-node – used k-means
  - NERSC Cori KNL nodes ran out of memory after (300x300)

Good news – lightcone extraction is embarrassingly parallel



Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.
Performance results are based on testing as of September 21, 2018 and may not reflect all publicly available security updates. See configuration disclosure for details.
Configurations: Testing on Cori at NERSC was performed with spectre_v1 and meltdown patches.

Work in Progress: Optimizing Lightcone Extraction

Currently, each past lc and future lc is stored as a separate 1d array

- For lightcone depth l, only 2l new reads per new point

Reshape kernel to reuse data

- Reduce memory reads and writes for creating lightcone arrays
- Better memory capacity and bandwidth utilization