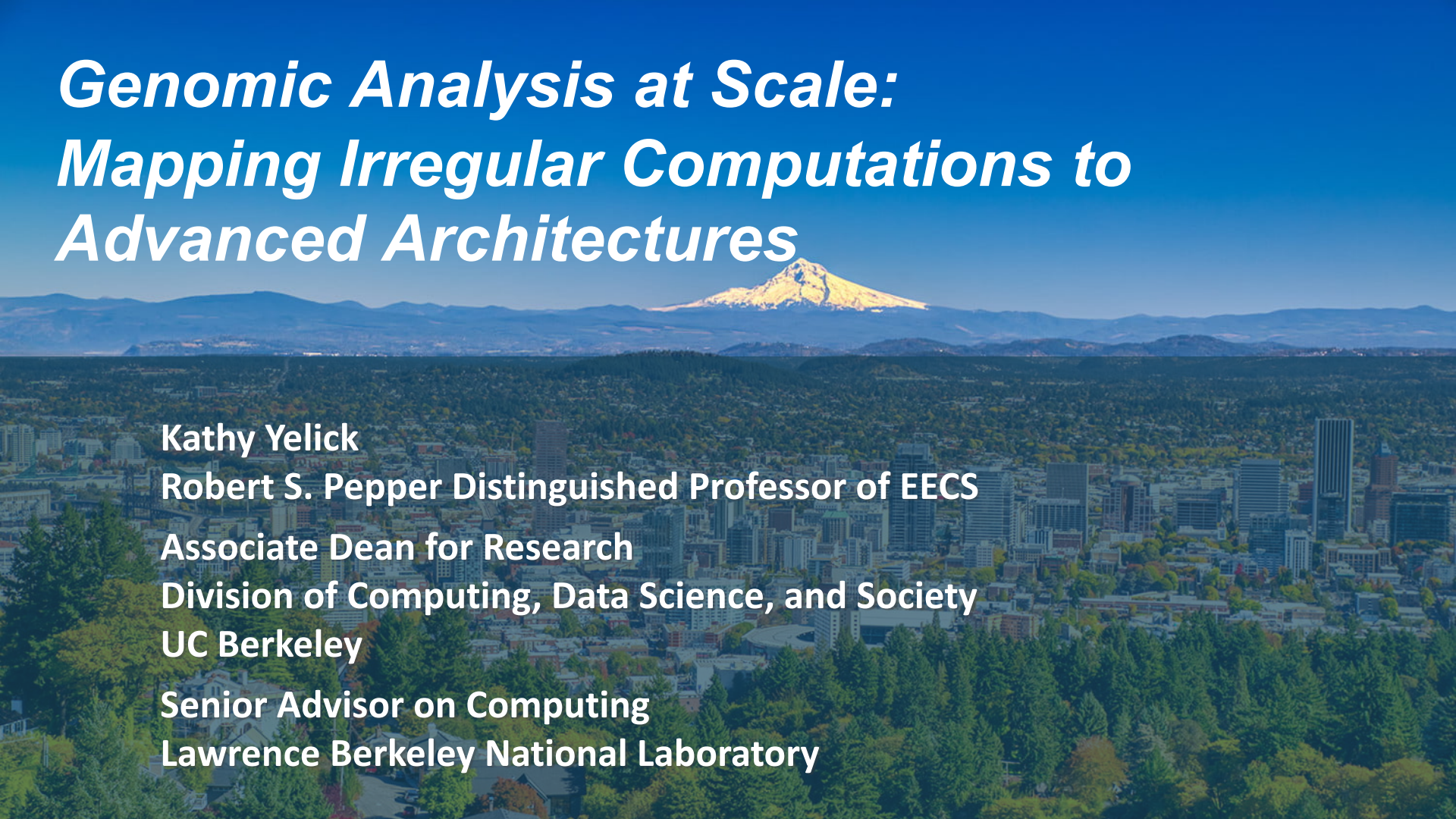
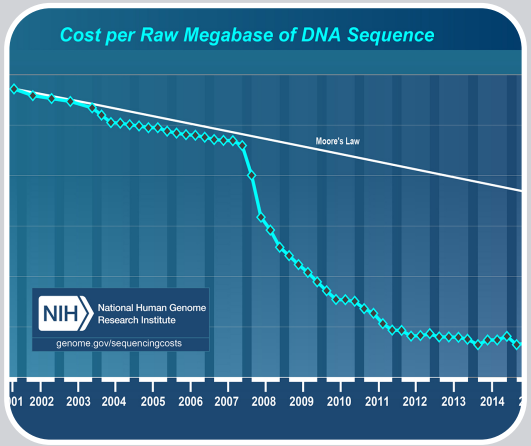
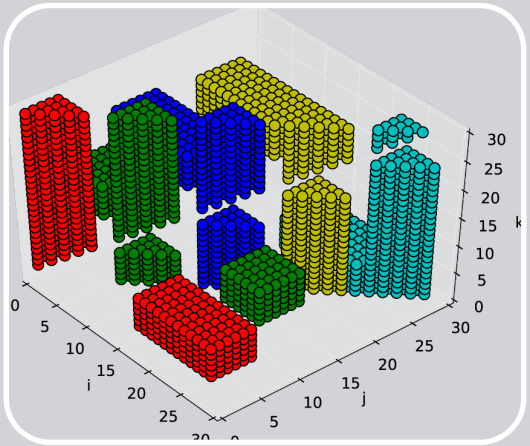
Genomic Analysis at Scale: Mapping Irregular Computations to Advanced Architectures



Kathy Yelick
Robert S. Pepper Distinguished Professor of EECS
Associate Dean for Research
Division of Computing, Data Science, and Society
UC Berkeley
Senior Advisor on Computing
Lawrence Berkeley National Laboratory

Genomic Analysis at Scale

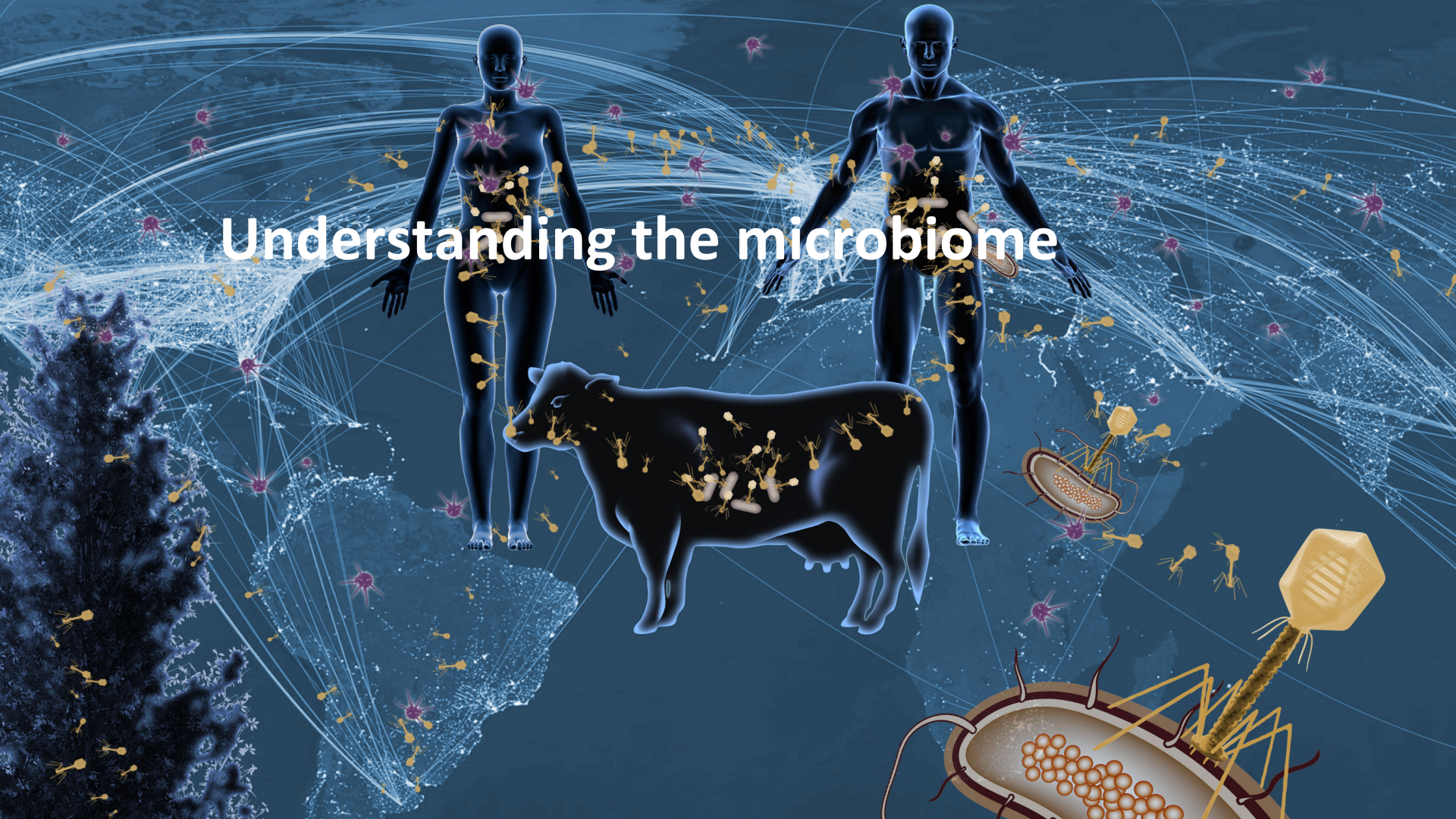


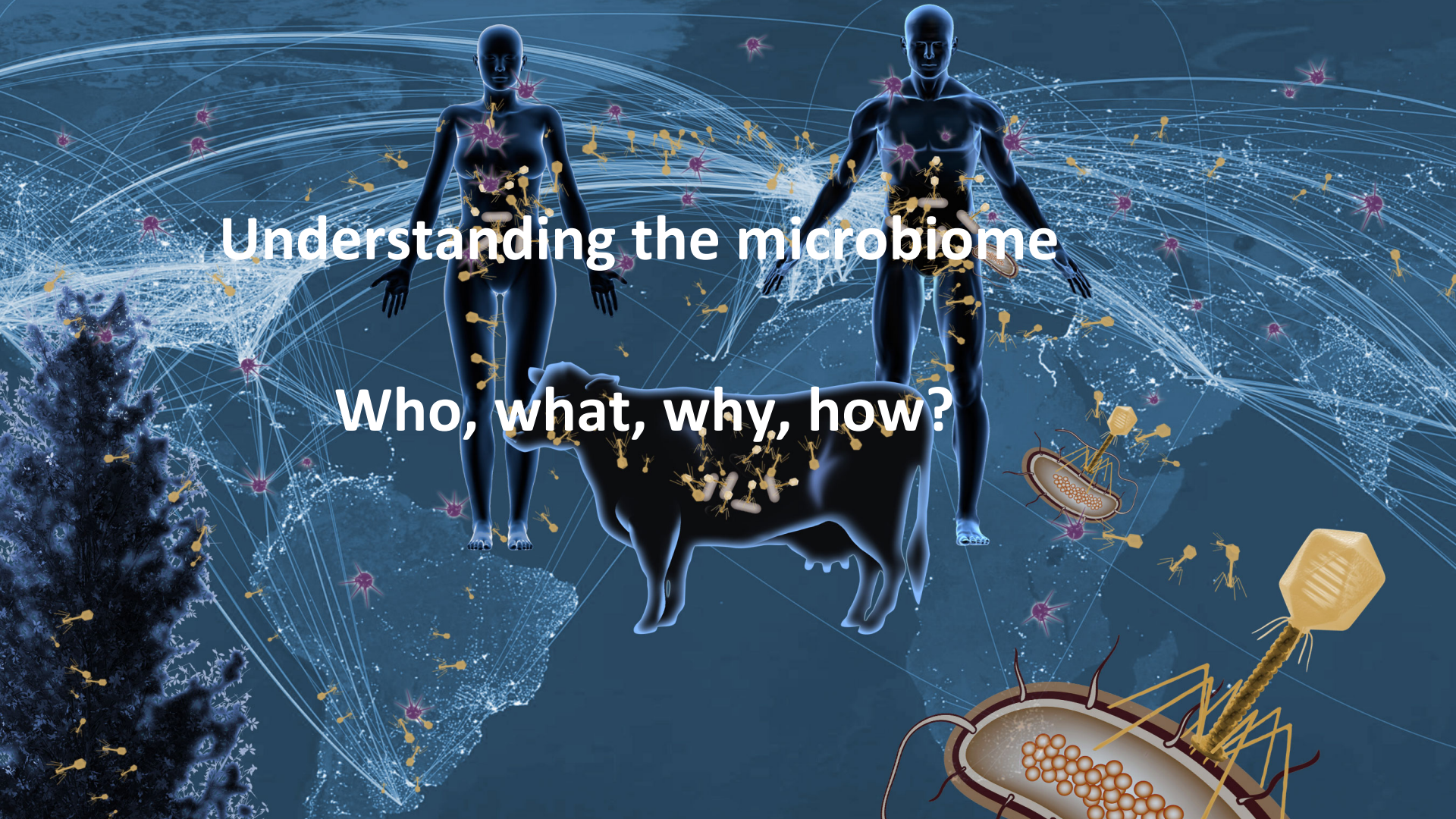


Big Data

Big Machines

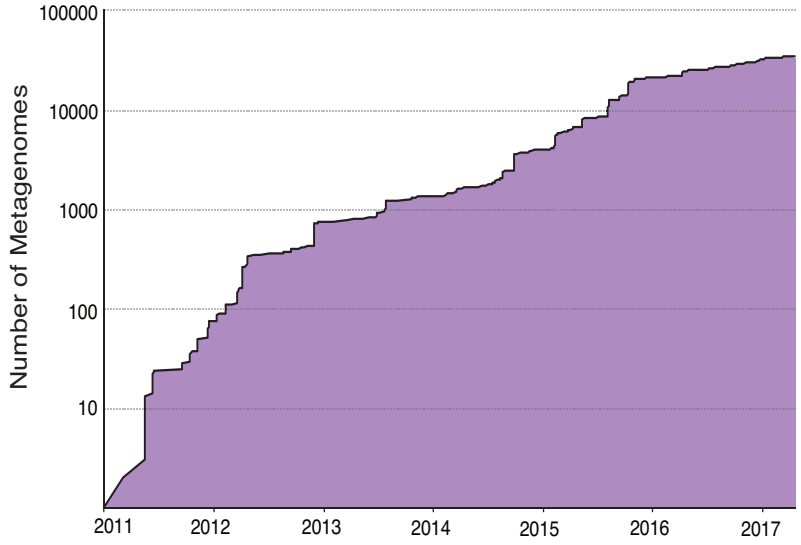
Scalable Algorithms

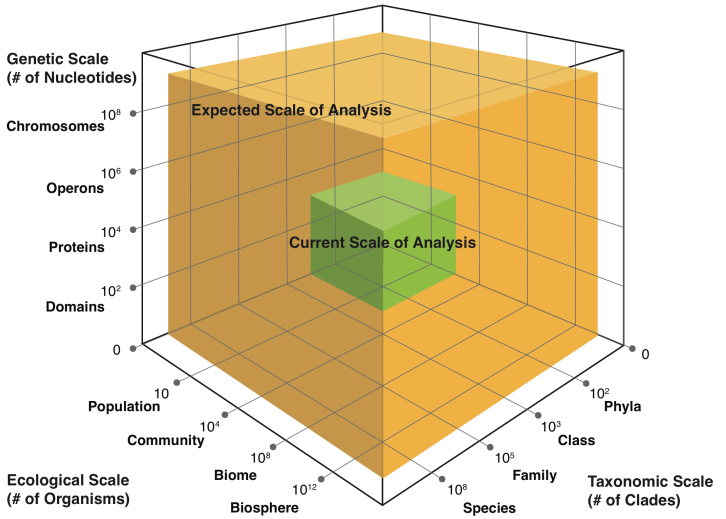

Understanding the microbiome


Understanding the microbiome
Who, what, why, how?

New Era of Microbiome Data Science

nmdc
National Microbiome Data Collaborative



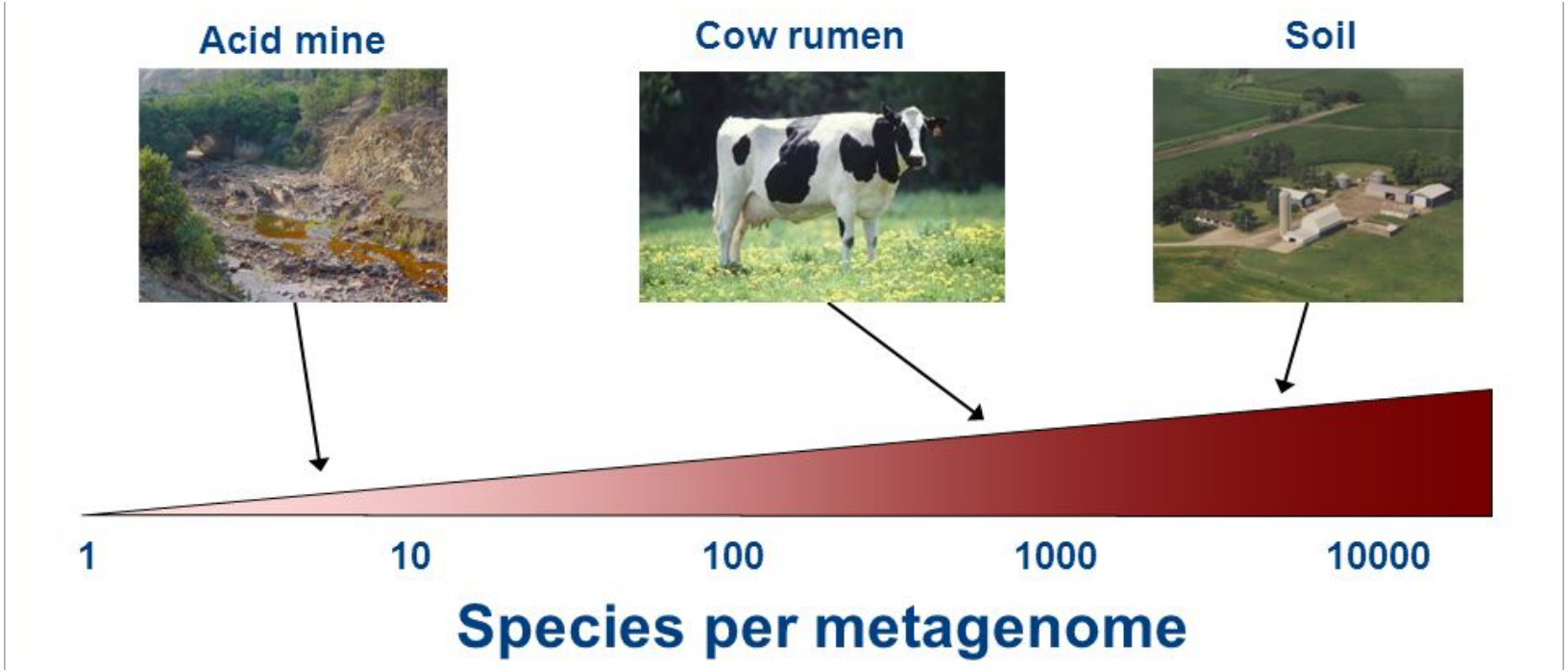


NMDC *launched in 2019 by DOE to collect, analyze, and serve microbial data.*

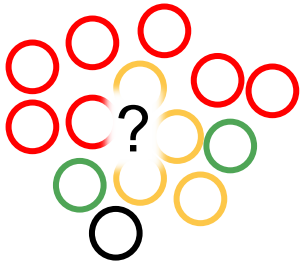
Courtesy of Emiley A. Eloe-Fadrosh, JGI, LBNL

BERKELEY LAB

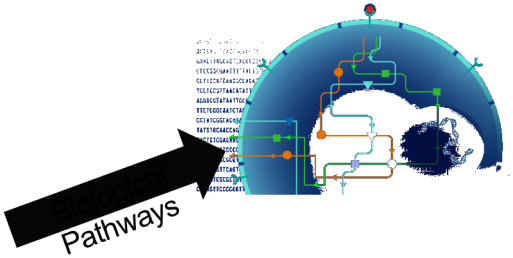
Metagenome Complexity

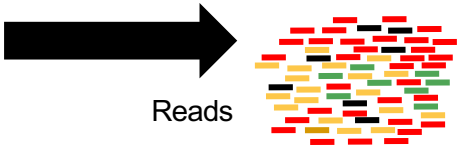


Microbiome analysis: metagenome

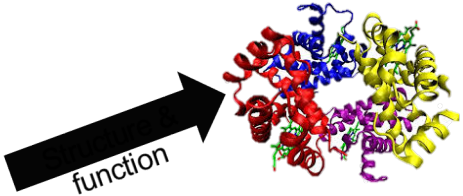


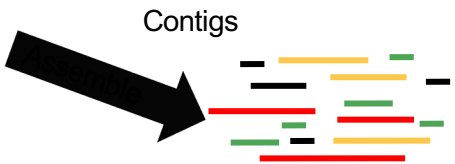
Microbial community





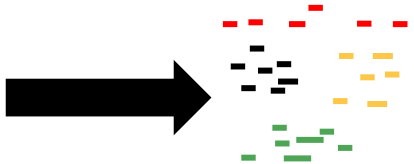
Reads





Contigs

Species Bins

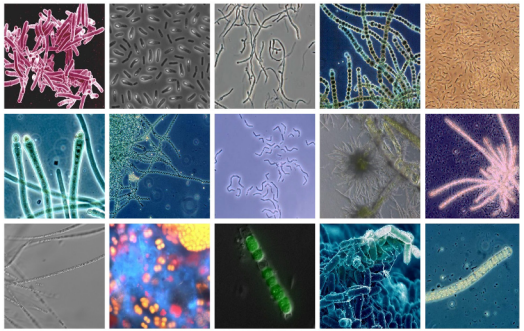


Proteins

Big Science Questions

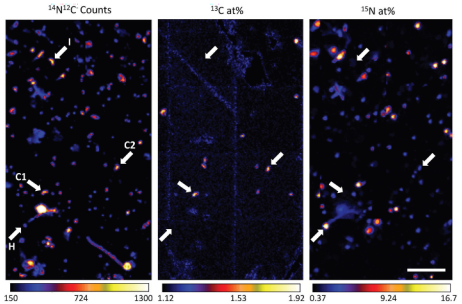
What happens to microbes after a wildfire? (1.5TB)

What at the seasonal fluctuations in a wetland mangrove? (1.6 TB)



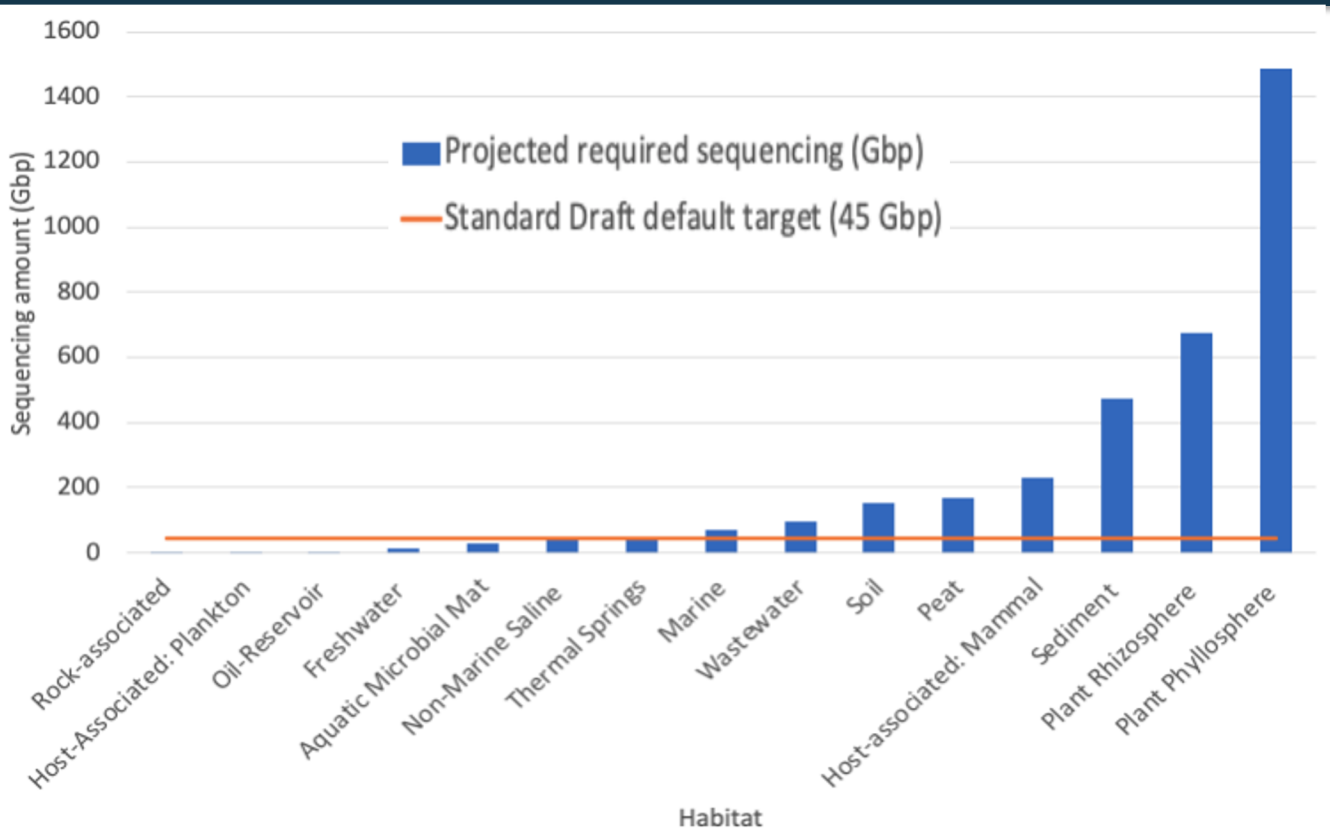
What are the microbial dynamics of soil carbon cycling? (3.3 TB)

How do microbes affect disease and growth of switchgrass for biofuels (4TB)



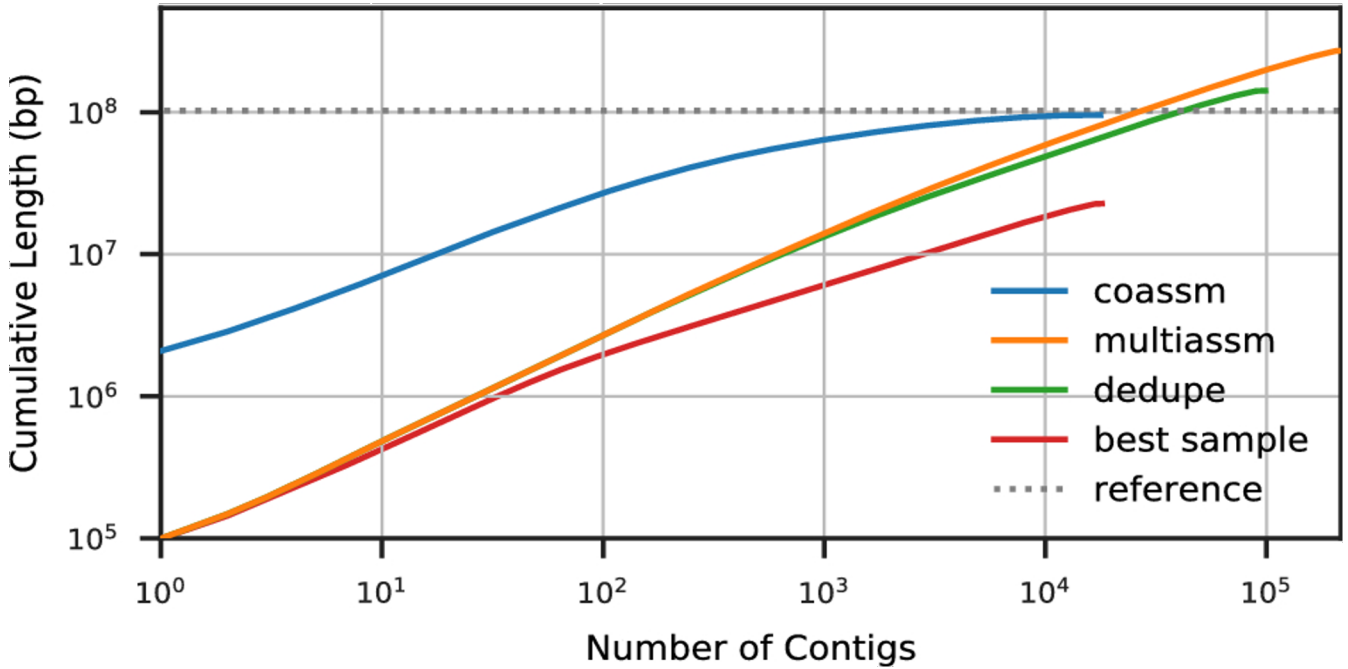
Combine genomics with isotope tracing methods for improved functional understanding (8TB)

Complex metagenomes require terascale data



Coverage analysis done by Kelly Cobaugh using Nonpareil 3 (Rodriguez-R, et al. mSystems 2018)

Big Data, Big Iron → Better Science



Multiassembly: assembling many samples separately

Coassembly: assembling many samples together

S. Hofmeyr, R. Egan, E. Georganas, A. Copeland, R. Riley, A. Clum, E. Eloe-Fadrosh, S. Roux, E. Goltsman, A. Buluç, D. Rokhsar, L. Oliker, K. Yelick, 2020

Clustering Huge Protein Datasets

- **Protein families via clustering**
- **Functional diversity**
  - Oceans vs human microbiome
- **New genes and proteins, e.g.**
  - Novel CRISPR/Cas genes
  - Gene clusters encoding antibiotics

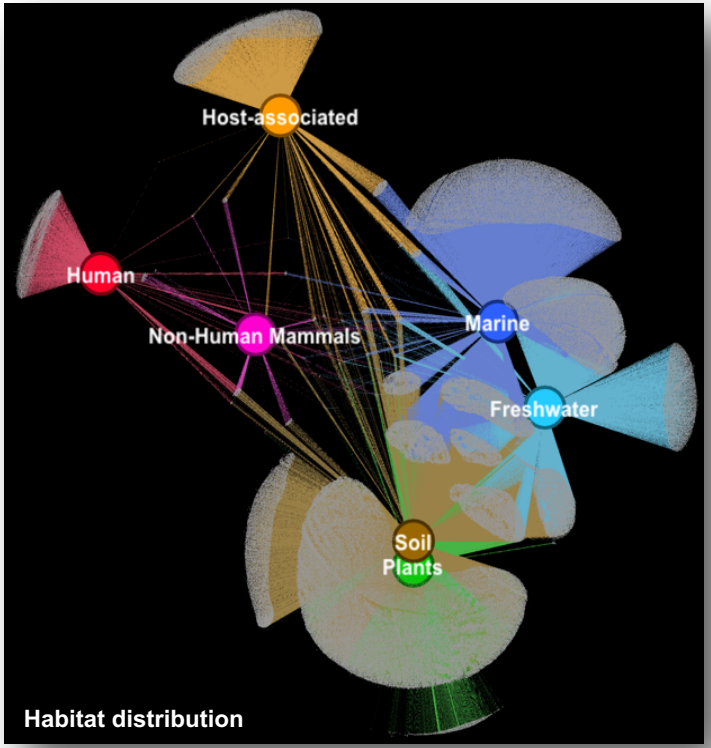


Image: G. Pavlopoulos and N. Kyrpides

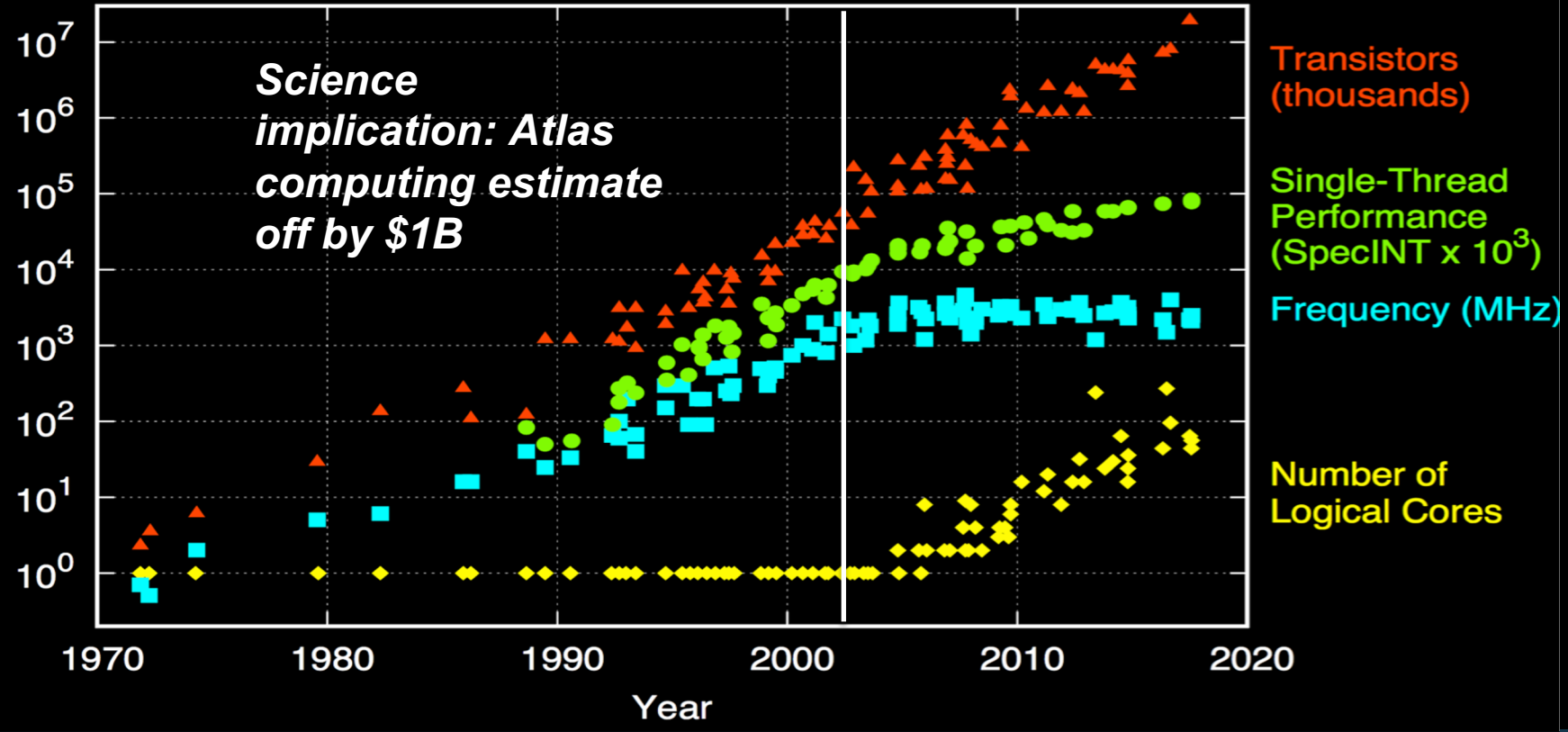
[1] Wang, Y., Shi, Q., Yang, P. *et al.* Fueling ab initio folding with marine metagenomics enables structure and function predictions of new protein families. *Genome Biol* **20**, 229 (2019).

Moore’s Law

It’s hard to think exponentially

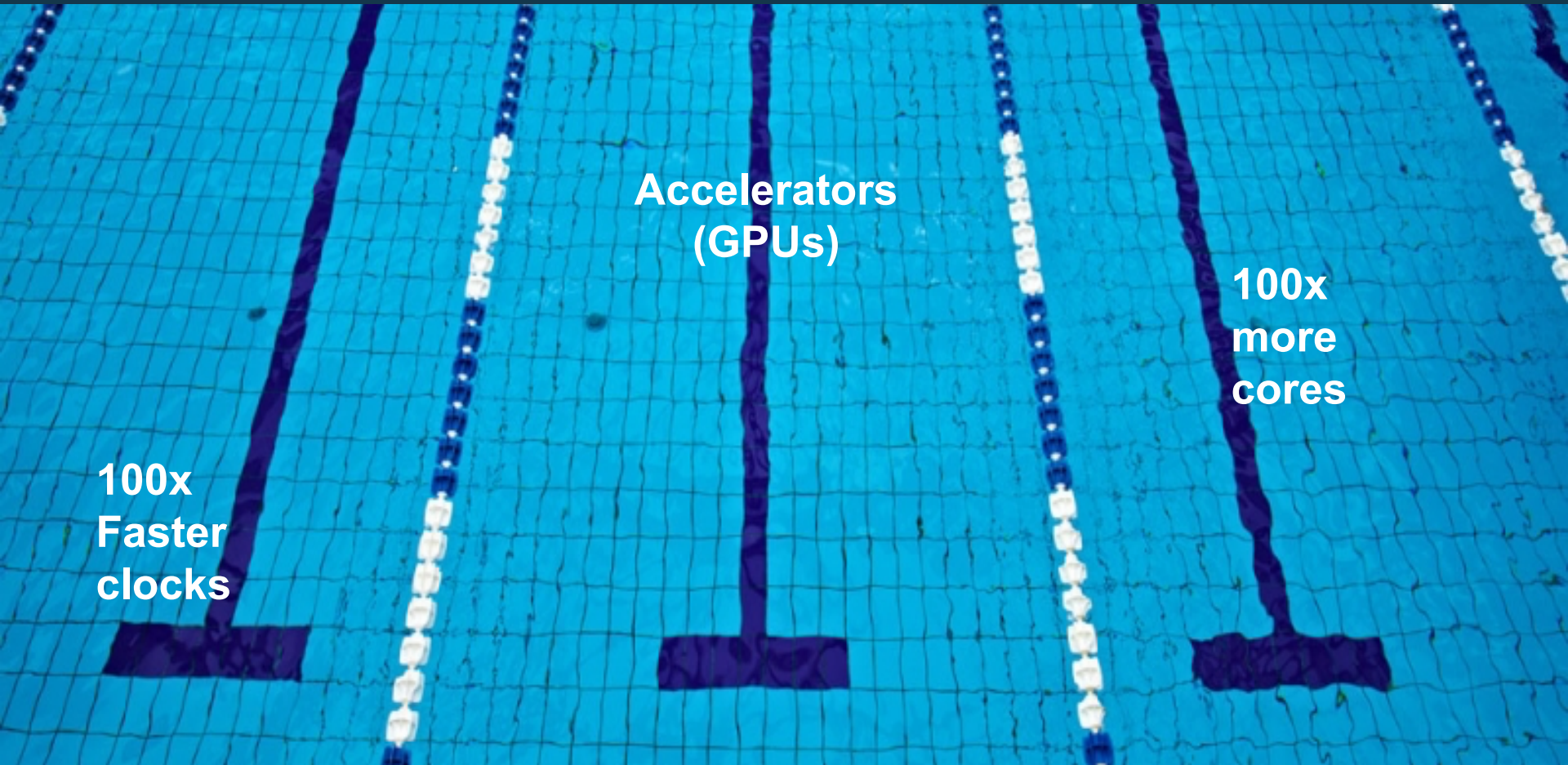
But it’s also hard to stop

Dennard Scaling is Dead; Moore's Law Will Follow



M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, C. Batten, and K. Rupp

Exascale Architecture Plans (2008)



Exascale Architecture Plans (2021)

Accelerators

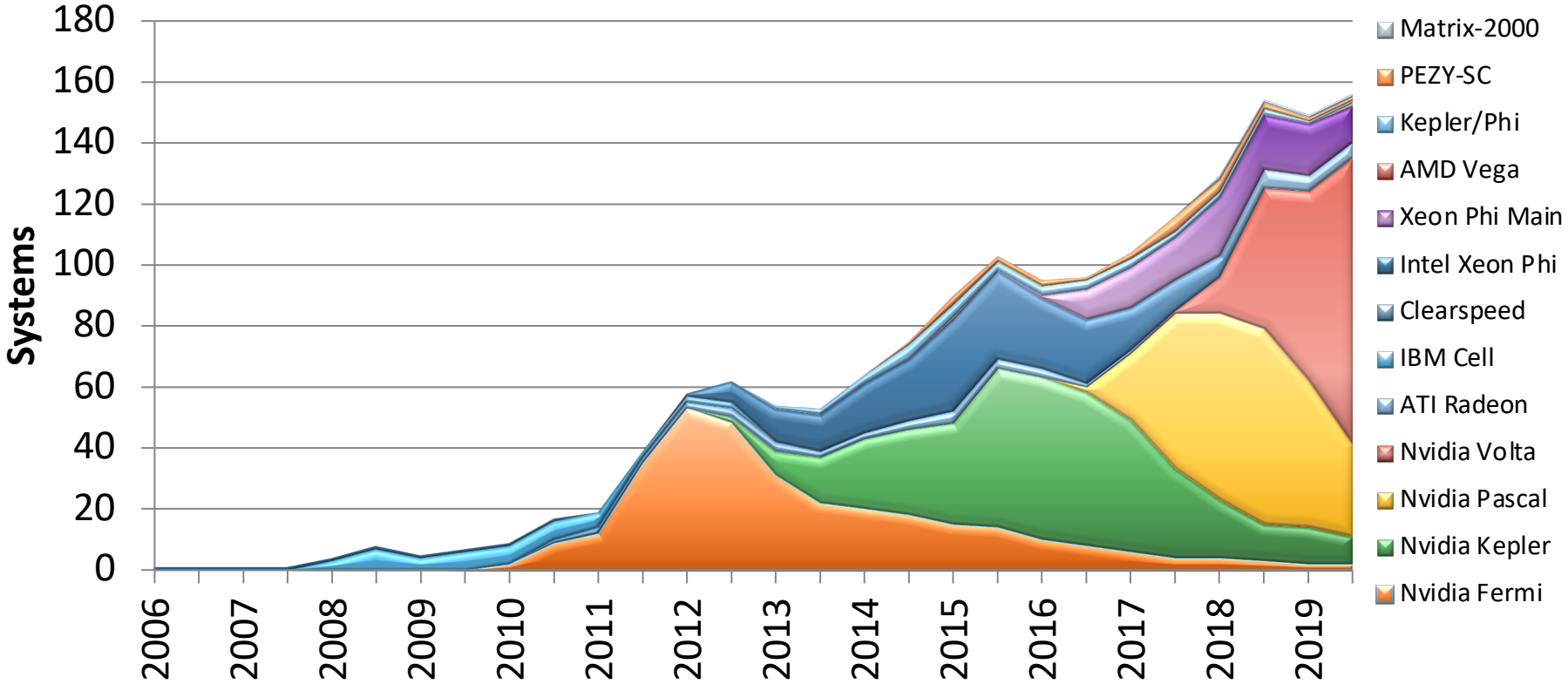
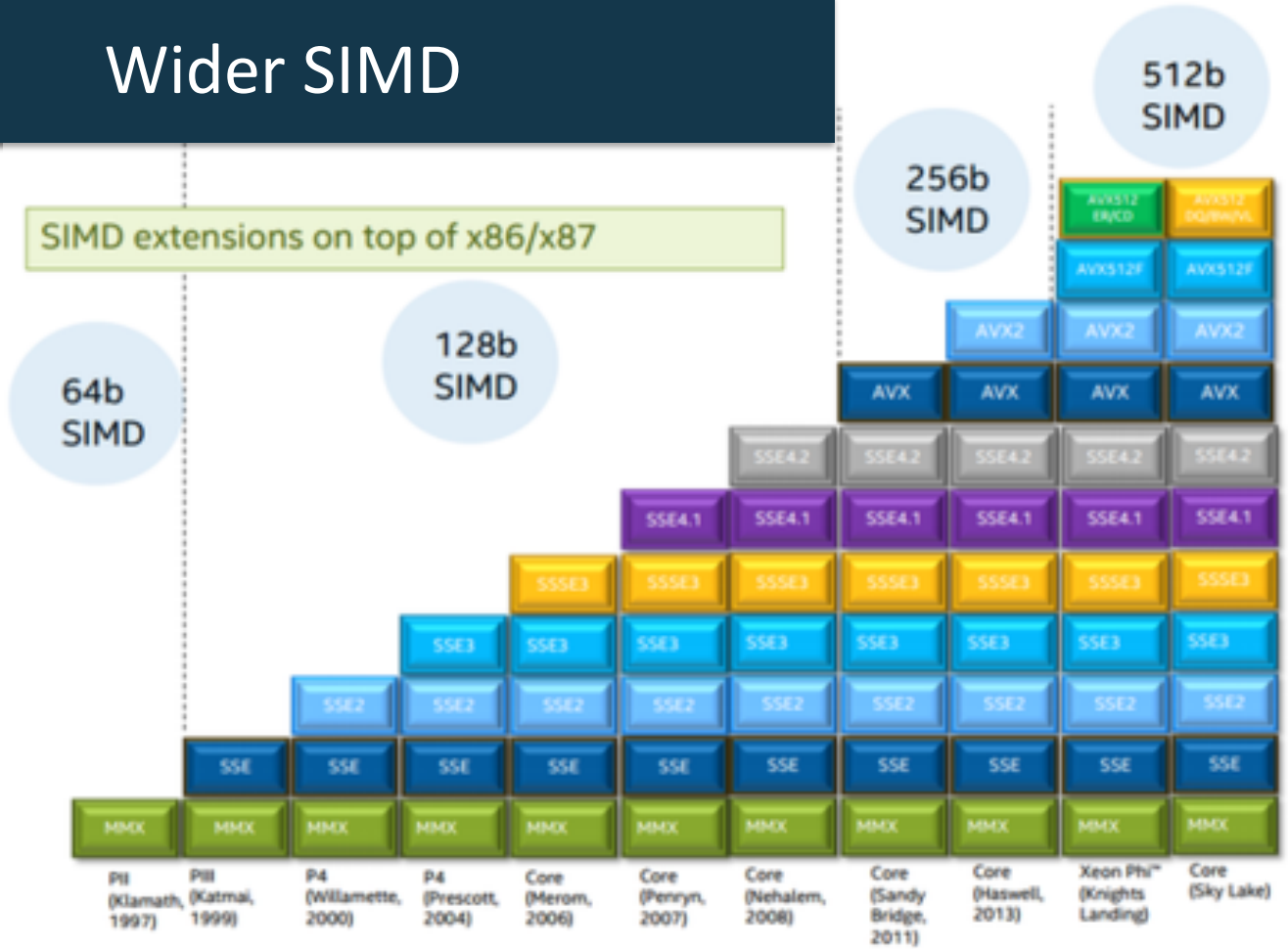


Image: http://slideplayer.com/slide/7541288/

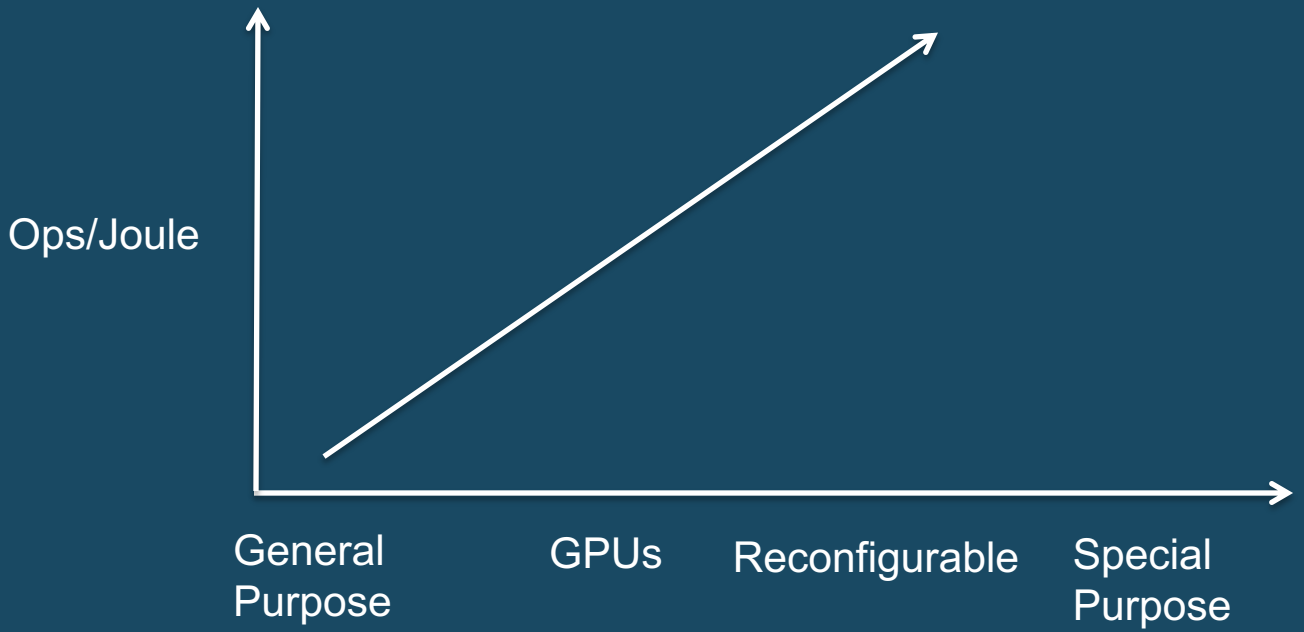
Wider SIMD



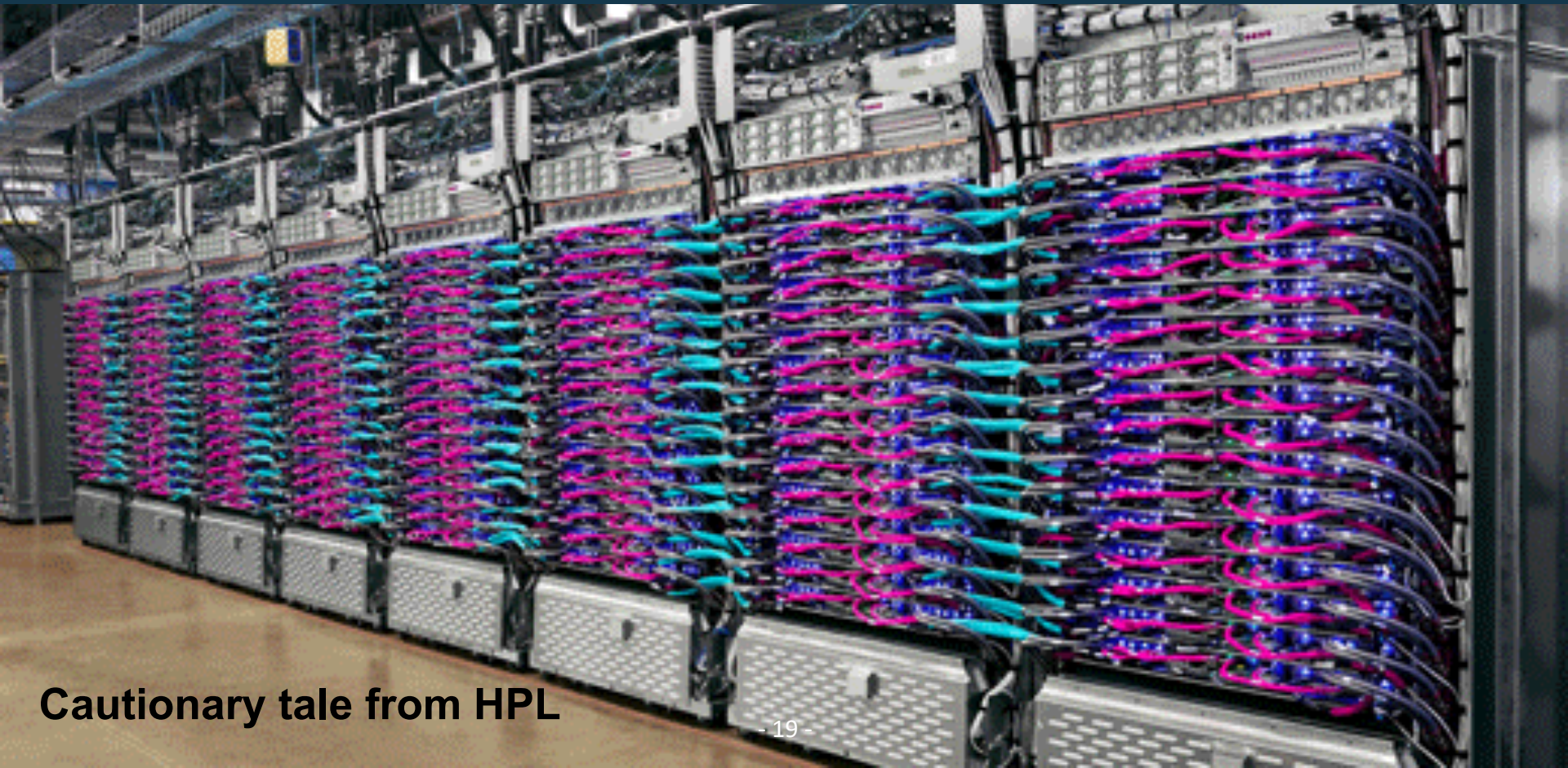
Top 500
#1 Fugaku
ARM w/ 512b

Image from Dezso Sima presentation

Specialization: End Game for Moore's Law

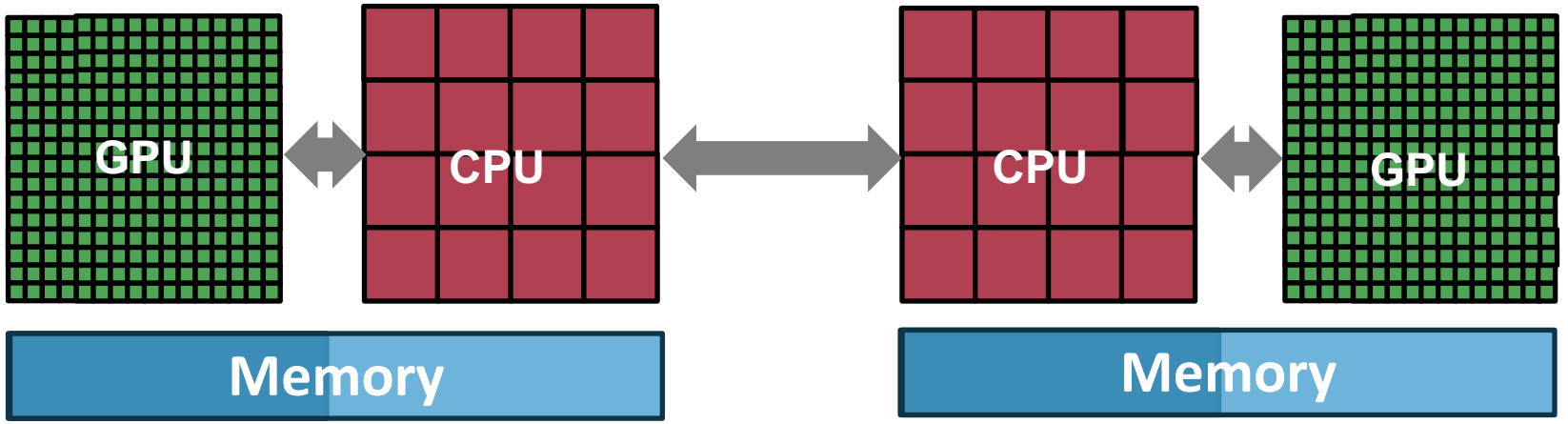


Is deep learning the only application?



Cautionary tale from HPL

Specialization, Yes Accelerators, No!



More cores	More data parallelism	Narrow data types	More memory spaces	CPUs in control	CPUs communicate

Data Movement is Expensive

Hierarchical energy costs.

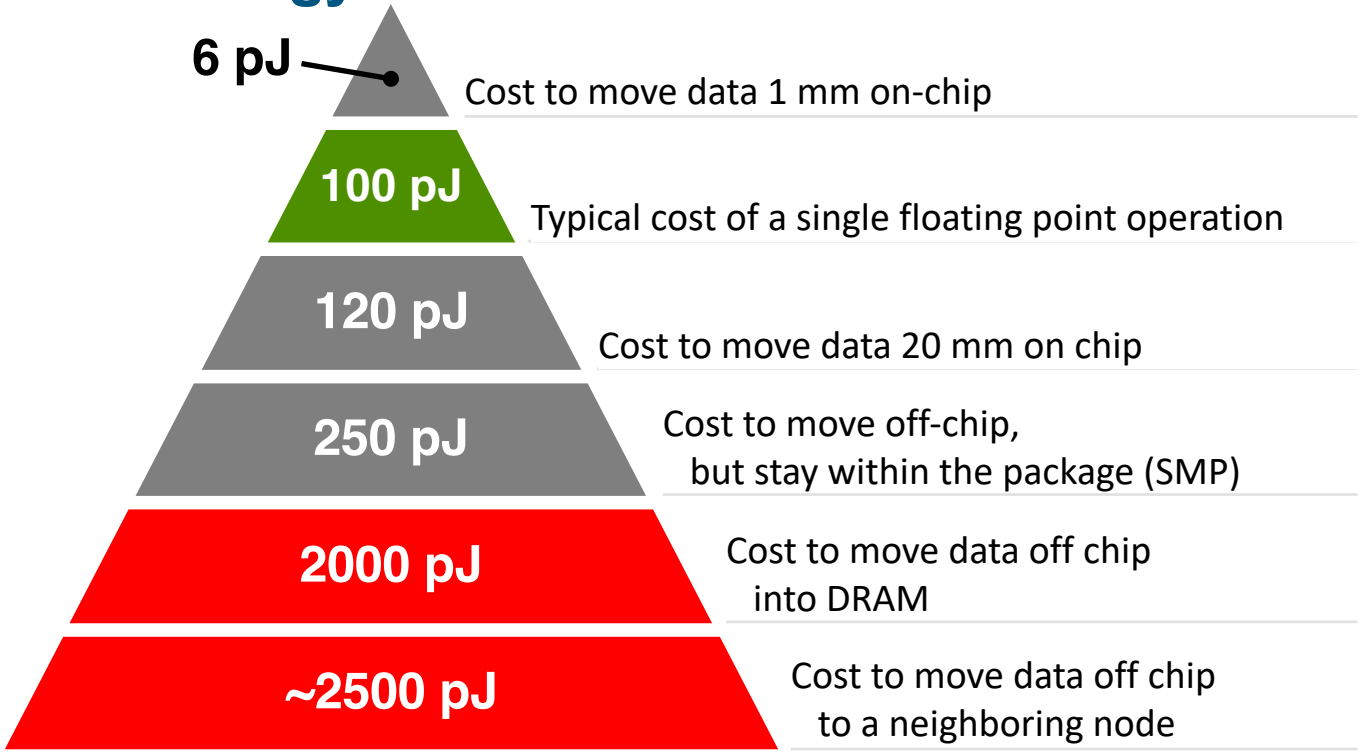
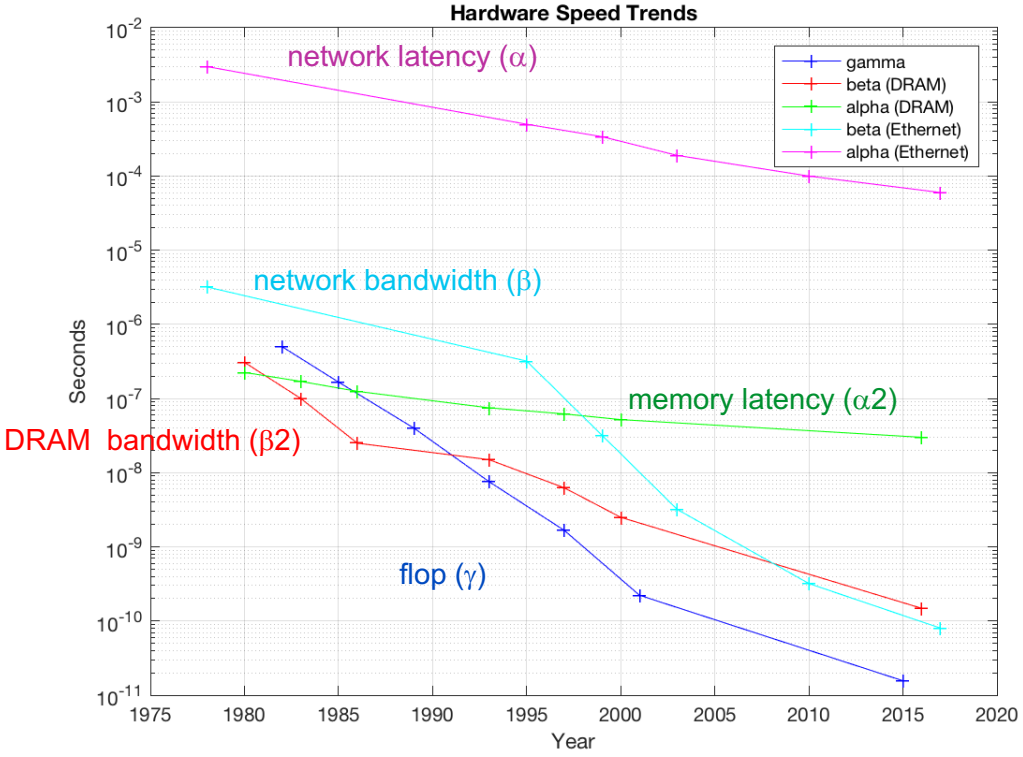


Image: http://slideplayer.com/slide/7541288/

Communication Dominates: Dennard was too good



Time =

flops * γ +

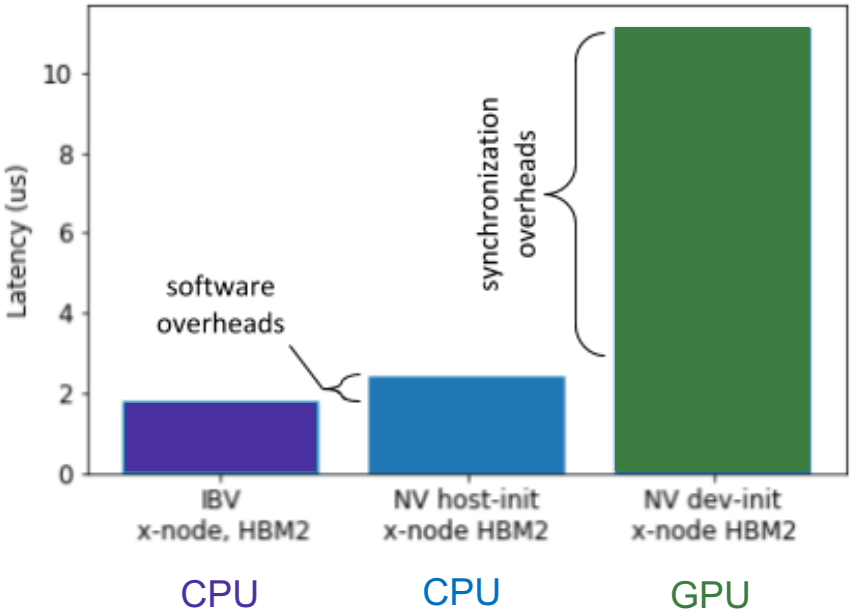
message * α +
bytes comm * β +

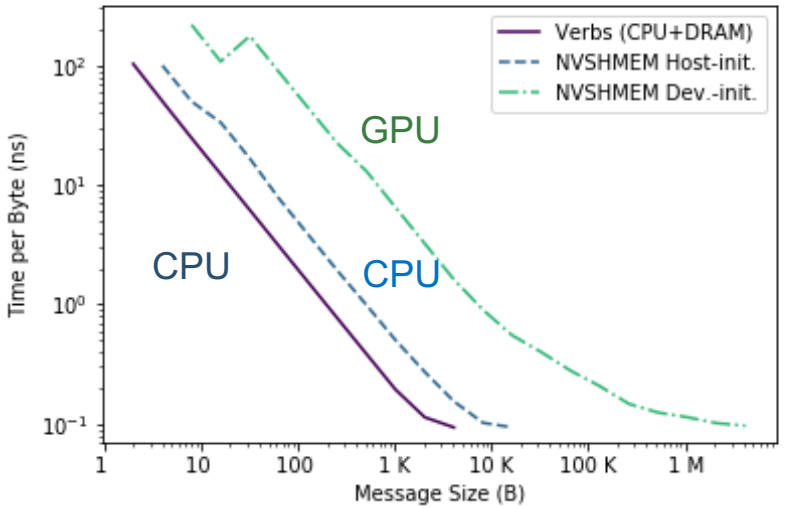
diff memory locs * α2 +
memory words * β2

Data from Hennessy / Patterson, Graph from Demmel

Put Accelerators in Charge of Communication

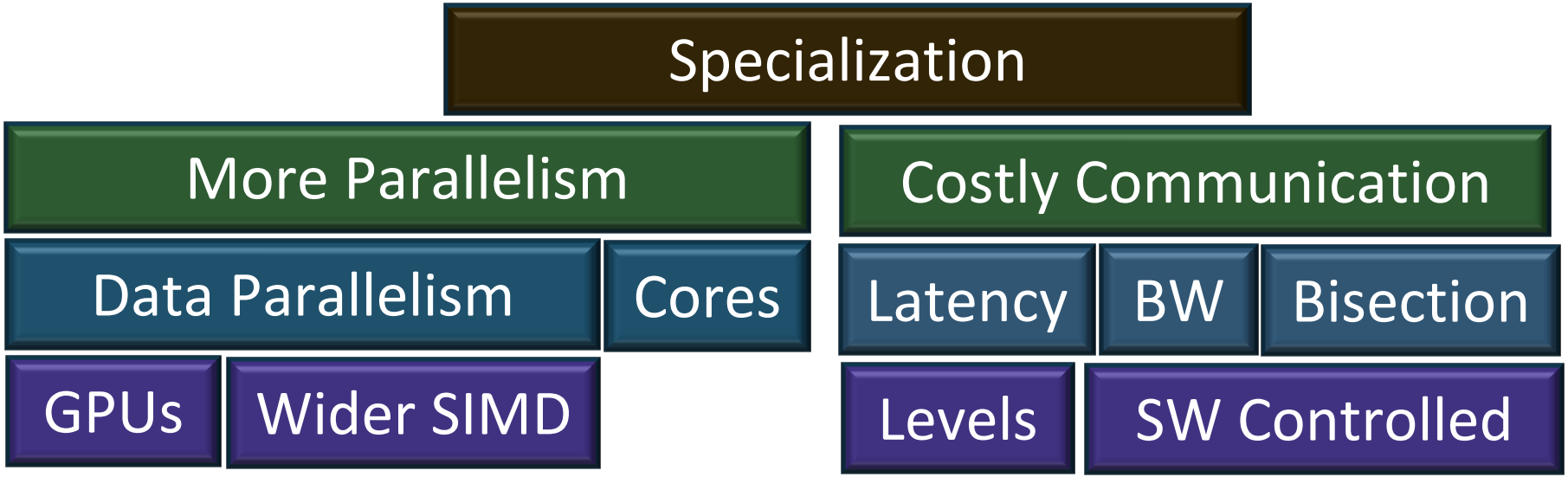
Architecture and software are not yet structured for accelerated-initiated communication (Summit with NVLink between Power9 CPUs and NVIDIA GPUs)





Taylor Groves et al

Hardware Trends

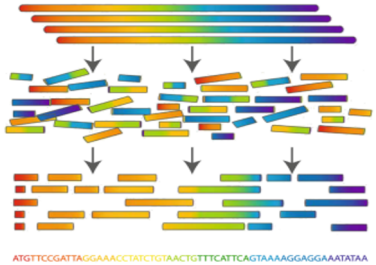


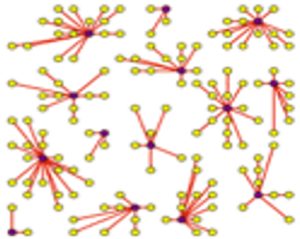
Tradeoffs in integration (faster communication) vs scale (amount of fast memory) and flexibility

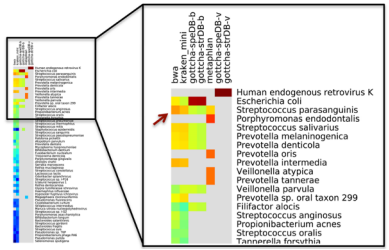
Algorithms and Software

ExaBiome project overview

Exascale algorithms & systems for previously intractable problems







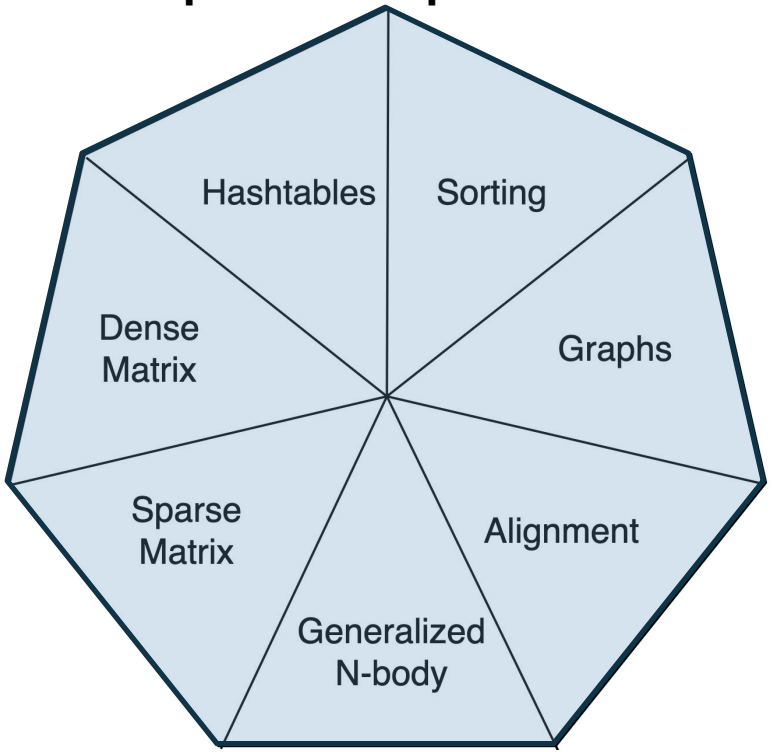
	Metagenome Assembly	Protein Clustering and Annotation	Comparative Metagenome Analysis
Science Need	Find species, genes and relative abundance in microbial communities	Improve understanding of tree of life for microbes; aid in identifying gene function	Track microbiome over time or space, changes in environment, climate, etc.
Computing Technique	hash tables, alignment, k-mer counts, graph walks	direct tables, alignment, k-mer counts, sparse matrices, ML (clustering, GNNs)	hash tables, alignment, k-mer counts, ML (dimensionality reduction)

BERKELEY LAB · Los Alamos NATIONAL LABORATORY EST.1943 · JGI · INDIANA UNIVERSITY · UNIVERSITY OF CALIFORNIA

http://exabiome.org

Motifs of Genomic Data Analysis

These computational patterns dominate ExaBiome Project experience



Application problems

- Assemble genomes
- Compute distances
- Cluster (contigs, proteins,…)
- Annotate

Distributed memory platforms open up new approaches and science questions

Yelick, et al. “The Parallelism Motifs of Genomic Data Analysis”, Philosophical Transactions A, 2020

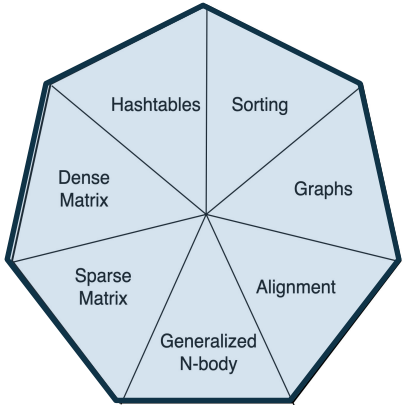
Analytics vs. Simulation Kernels:

7 Giants of Data	7 Dwarfs of Simulation
Basic statistics	Monte Carlo methods
Generalized N-Body	Particle methods
Graph-theory	Unstructured meshes
Linear algebra	Dense Linear Algebra
Hashing	Sparse Linear Algebra
Sorting	Spectral methods
Alignment	Structured Meshes

NRC Report + our paper

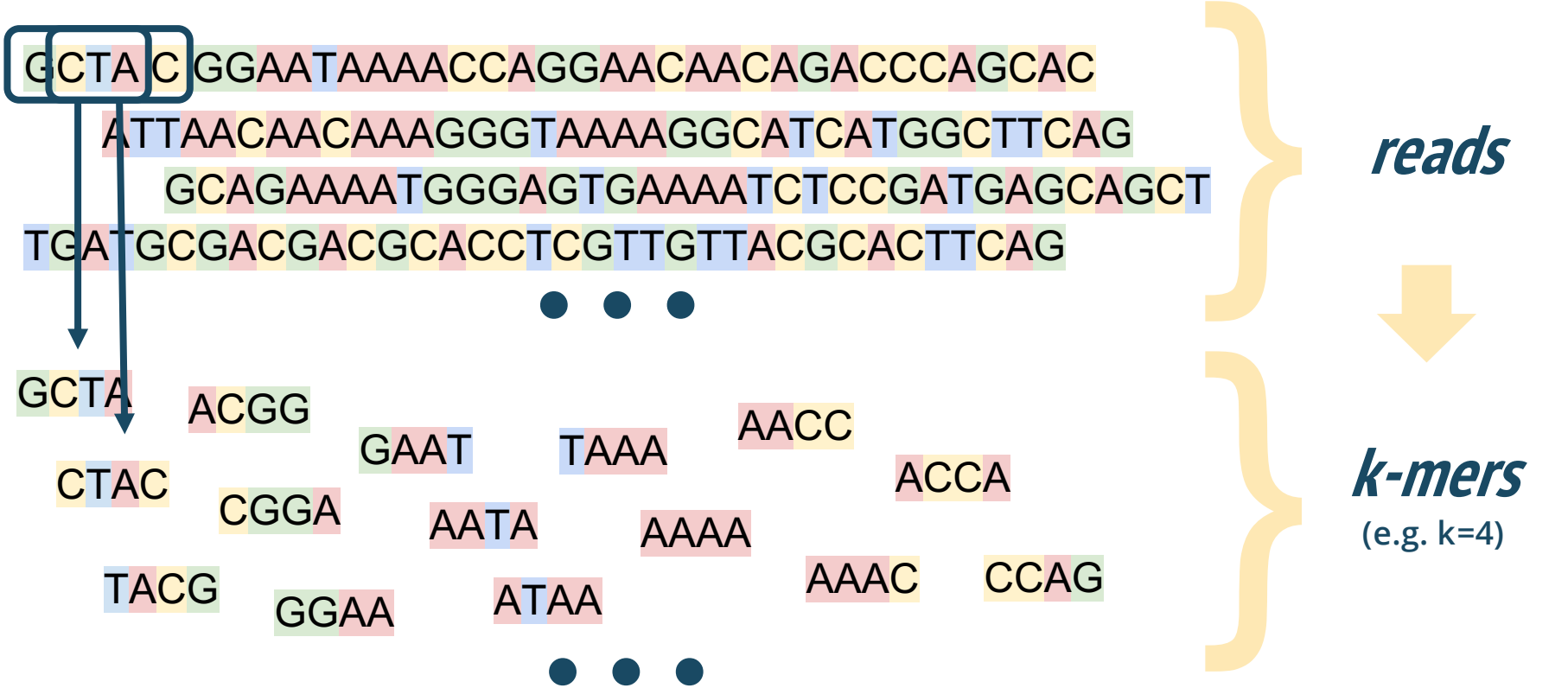
Phil Colella

Yelick, et al. “The Parallelism Motifs of Genomic Data Analysis”, Philosophical Transactions A, 2020


Hashtables
Sorting
Dense Matrix
Graphs
Sparse Matrix
Alignment
Generalized N-body

Hashing

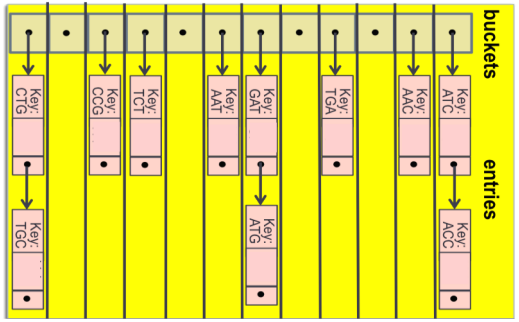
Common Technique: Analyze K-mers



Distributed Hash Tables of K-Mers

Make hash table of k-mers

AAC TGA CCG
ACC GAT CGT
CCT ATT GTC



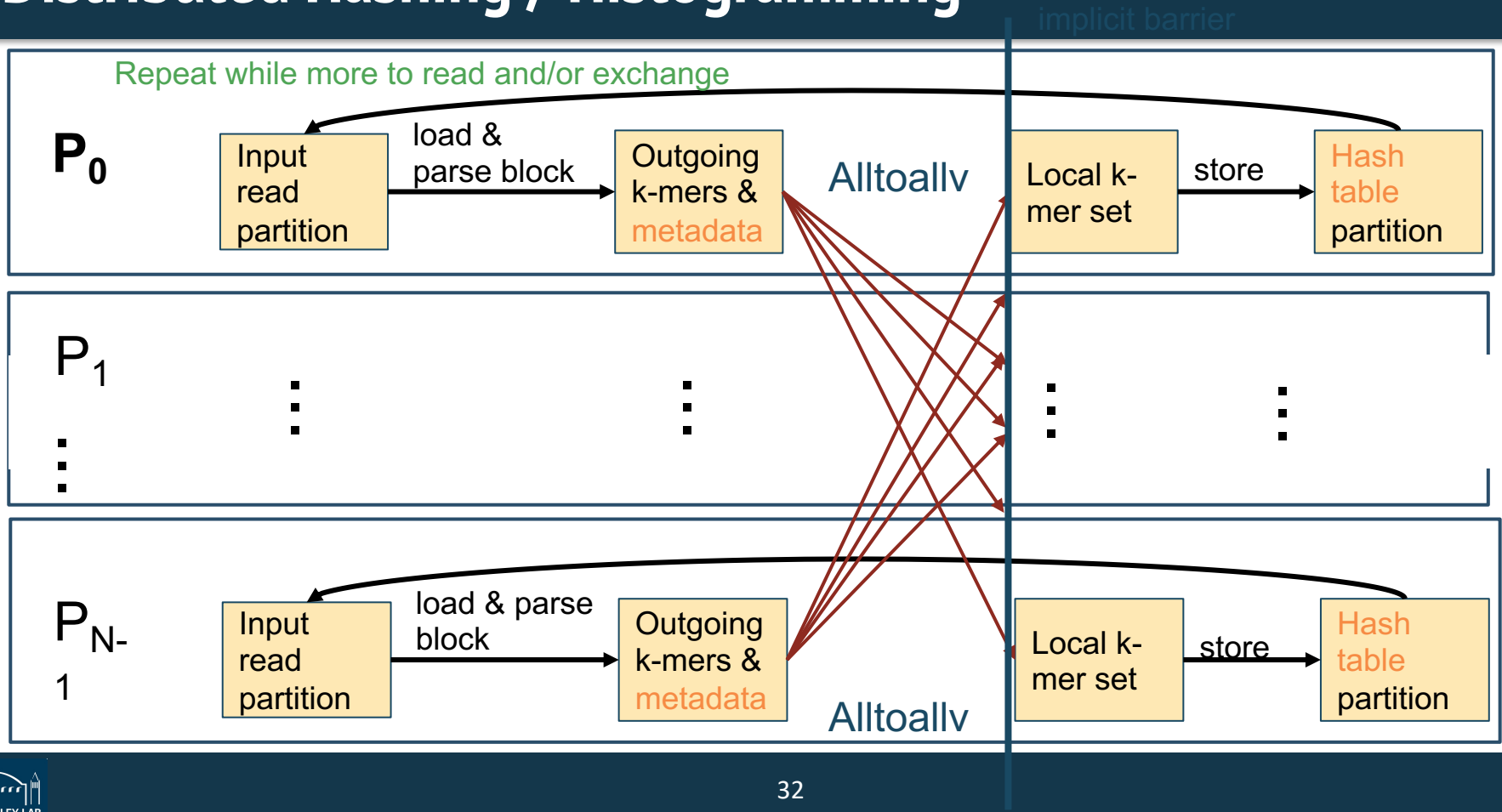
1-sided comm or irregular all-to-all + memory

Keys are fixed-length strings:

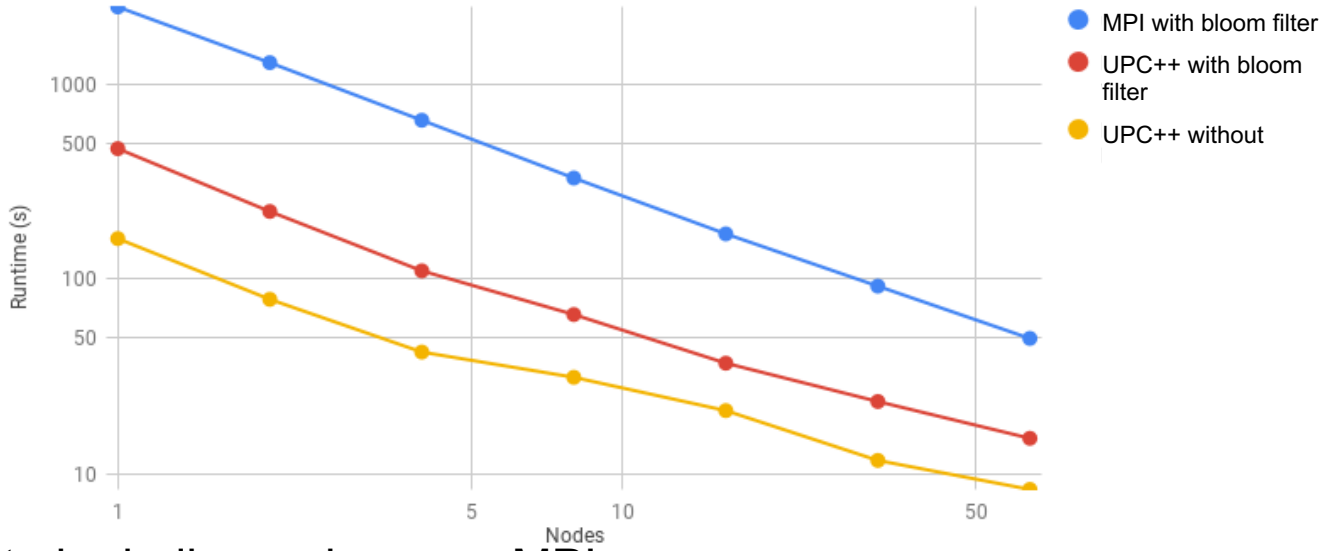
Values depend on application:

- Count and remove errors (singletons)
- Find strings with matching k-mer
- Connected components
- Use histogram as approximation

Distributed Hashing / Histogramming



K-mer counting: All the Wires All the Time



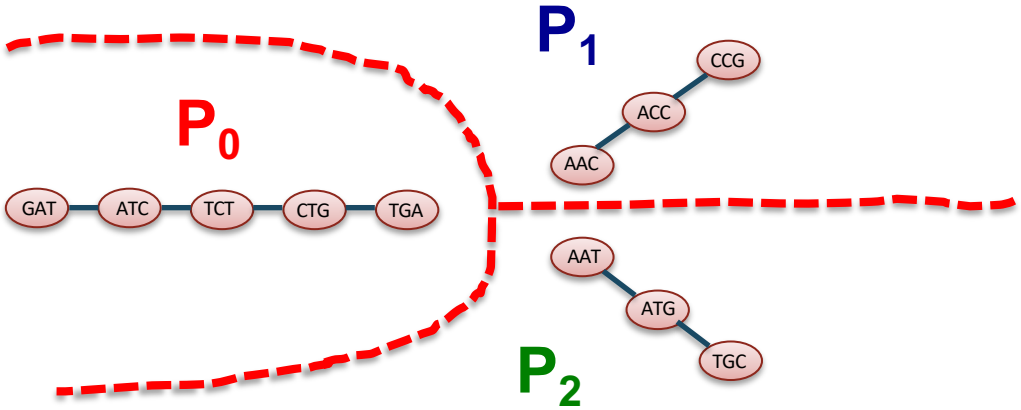
- Used to be bulk-synchronous MPI
- UPC++ communication is asynchronous and 1-sided
- UPC++ version is faster, avoids barriers, saves memory (one runtime)
- And it's simpler!

Steve Hofmeyr, Rob Egan, Evangelos Gerganas, leads on MetaHipMer software

Communication-Avoiding hash table lookup

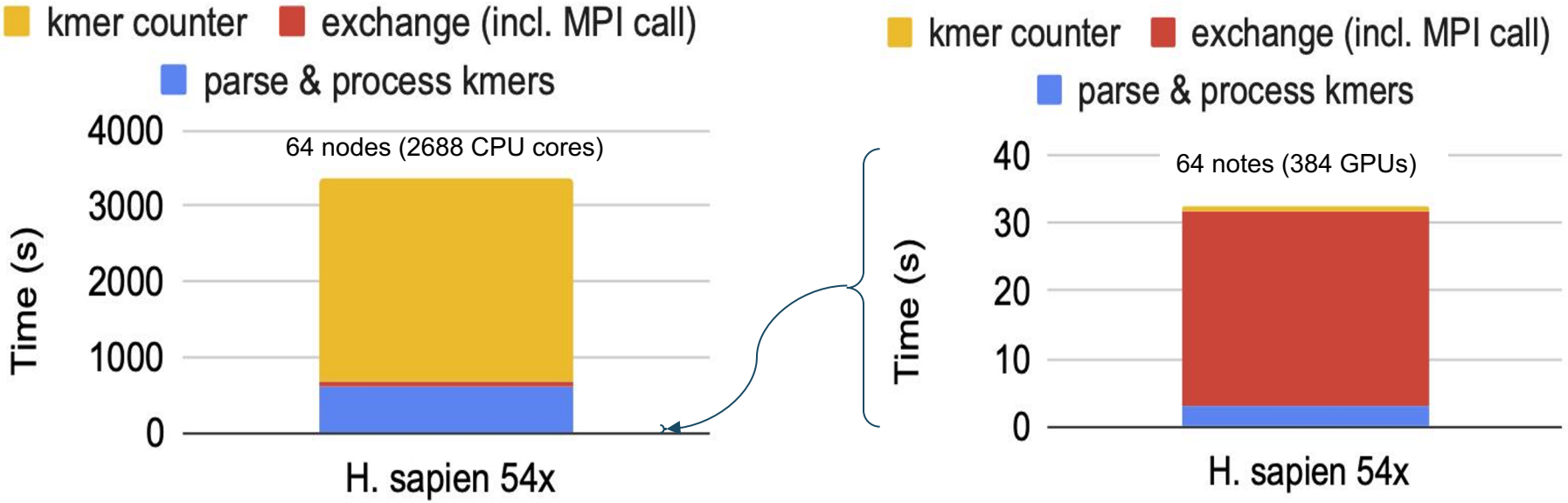
Caching for temporal locality (reuse): if few large items, so lookups will repeat

Layout for spatial locality: if we have an “oracle” that approximate final genome



Traversal is up to 2.8x faster!
Up to 76% reduction of off-node communication !

K-mer Counting: Finding Data Parallelism

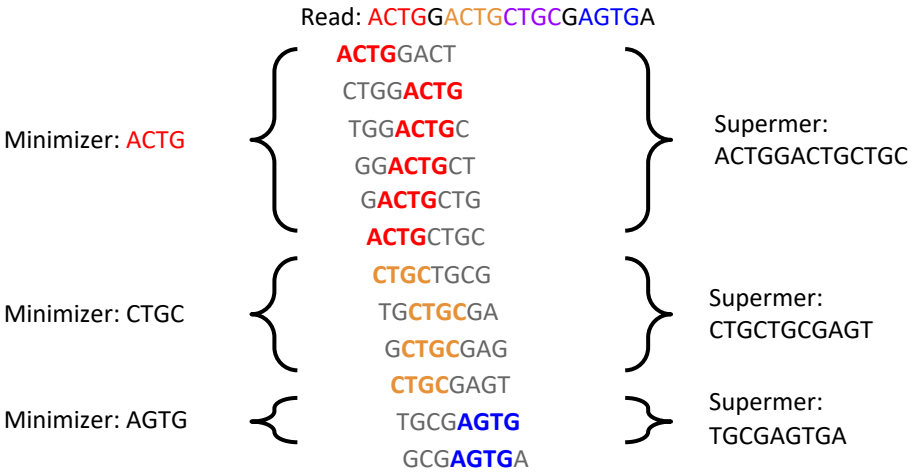


- K-mer counter on Summit. (Note scales -- red k-mer exchange time is roughly equal.)
- Reduce CPU/GPU communication by parsing as well as processing on GPU

Over 100x speedup!!

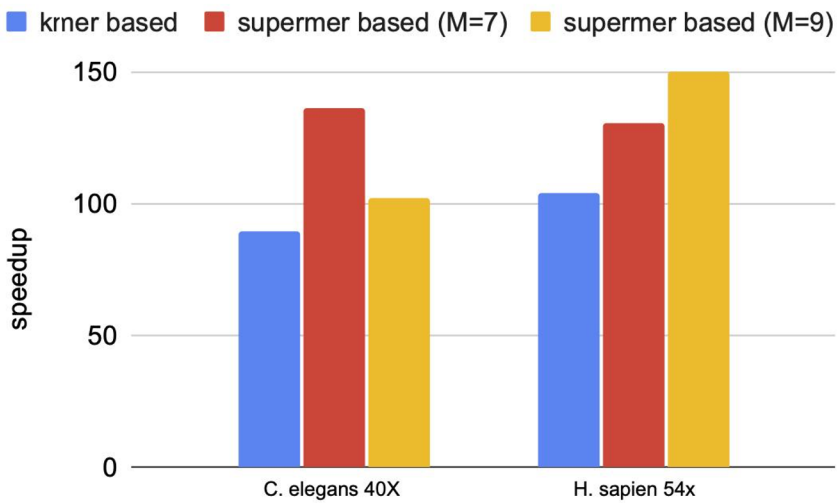
Israt Nisa, P. Pandey, M. Ellis, L. Oliker, A. Buluç, K. Yelick. Distributed-Memory k-mer Counting on GPUs. *IPDPS '21 (to appear)*

K-mer Counting: Reducing Communication



Reduce communication with "Supermers"

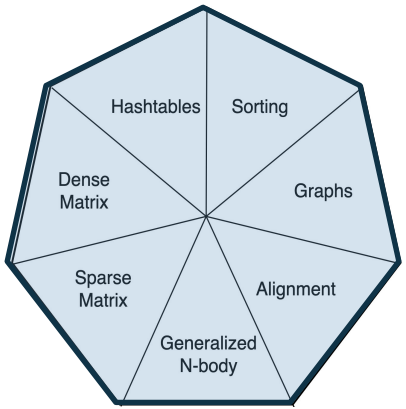
- Multiple contiguous k-mer
- map to the same process ID with minimizer-based hashing
- Saves volume (bandwidth) and number of messages (latency)



Speedup on 64 Summit nodes

- 6 GPUs / node
- baseline: 42 cores / node

Israt Nisa, P. Pandey, M. Ellis, L. Oliker, A. Buluç, K. Yelick. Distributed-Memory k-mer Counting on GPUs. *IPDPS '21 (to appear)*


Hashtables
Sorting
Dense Matrix
Graphs
Sparse Matrix
Alignment
Generalized N-body

Alignment

Smith-Waterman: Dynamic Programming

	_	G	A	T	C	A	G	C	T
_	0	0	0	0	0	0	0	0	0
G	0	1	0	0	0	0	0	0	0
A	0	0	2	0	0	1	0	0	0
T	0	0	0	3	1	0	0	0	1
A	0	0	1	1	2	2	0	0	0
G	0	0	0	0	2	1	3	1	0
C	0	0	0	0	1	1	2	4	2
C	0	0	0	0	1	0	2	3	3

GATCACCT
GAT_ACCC

Scoring
insert/delete = -2
match = 1
mismatch = -1.

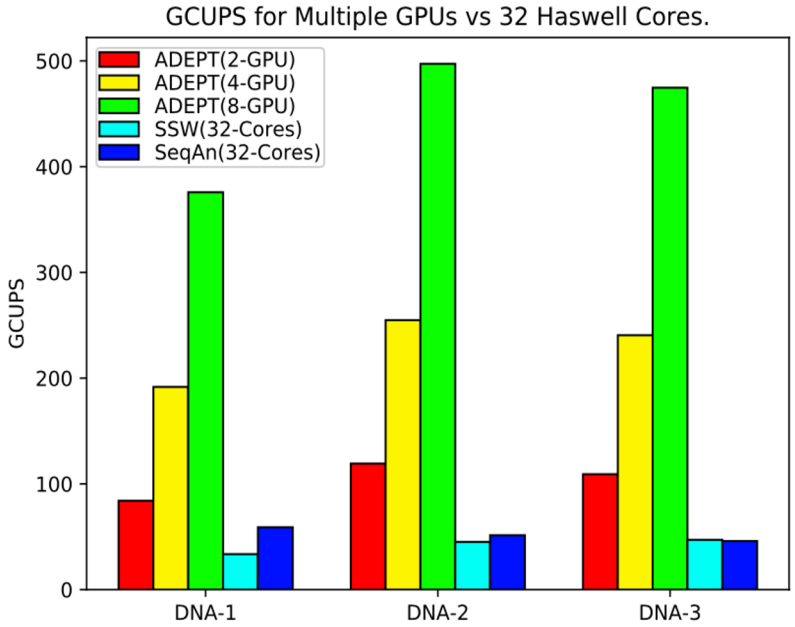
Options to search matrix

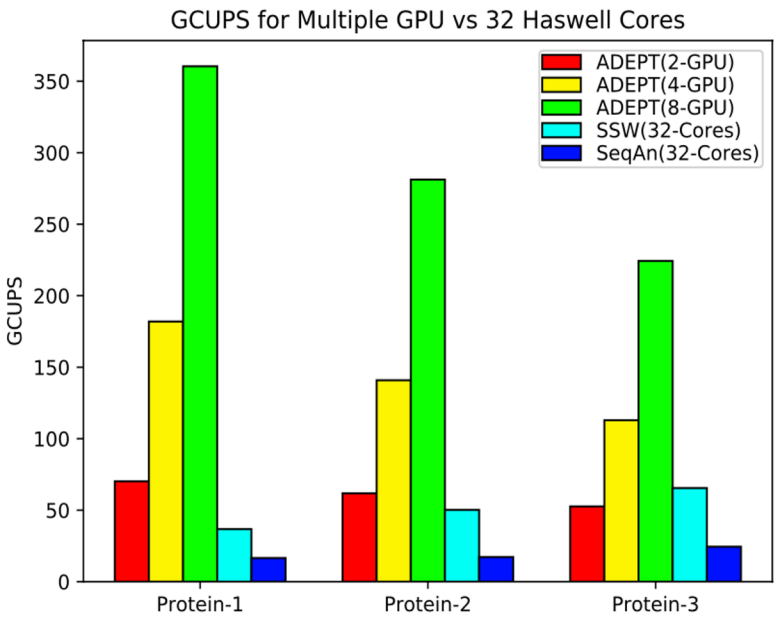
- Full search (Smith-Waterman)
- Banded (only search near diagonal)
- X-Drop stop search when the score drops by more than X

Variations for local / global alignment, per character penalties, seeding, etc.

ADEPT: Batch Alignment on GPUs

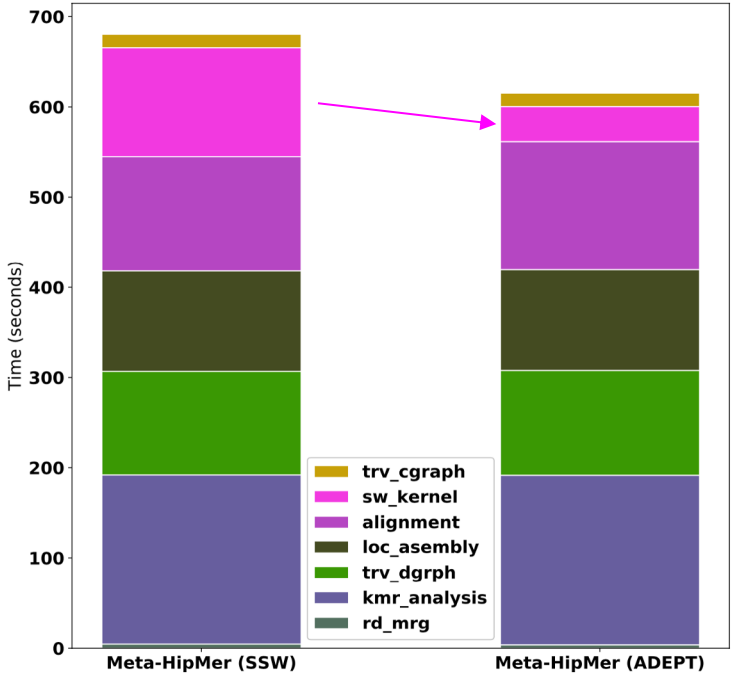
GCUPS = Giga-Cell Updates per second

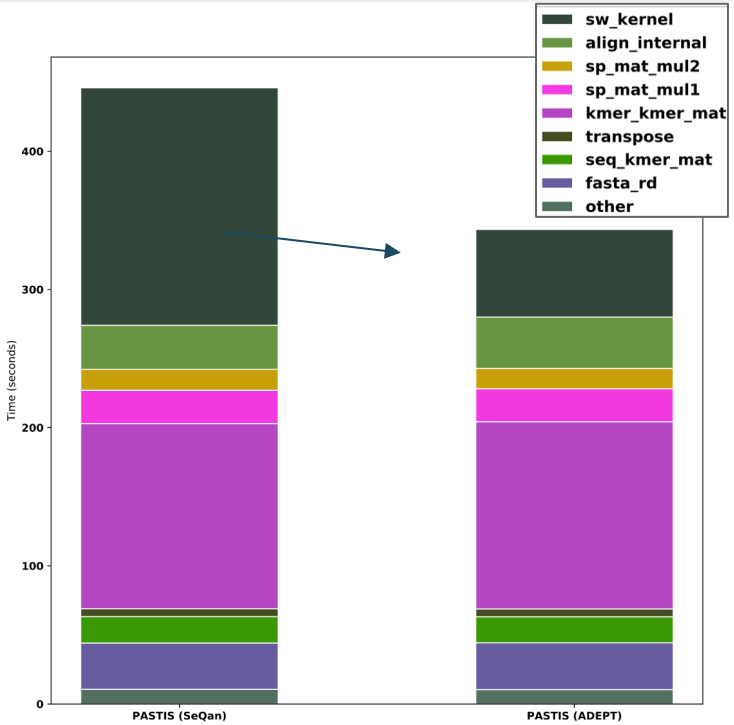




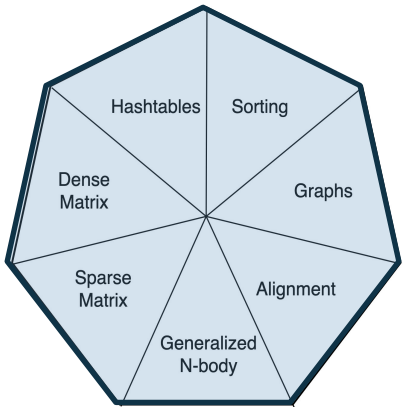
Adept is designed for relatively short, low-error sequences, both DNA (left) and proteins (right)
SSW and SeqAn are vectorized implementations of Smith-Waterman Algorithm on CPU.

ADEPT: Impact on ExaBiome Applications





Soil assembly SW time: 2.8 node hours on Cori, 0.1 node hours on Summit (hidden behind CPU work)



Generalized N-Body

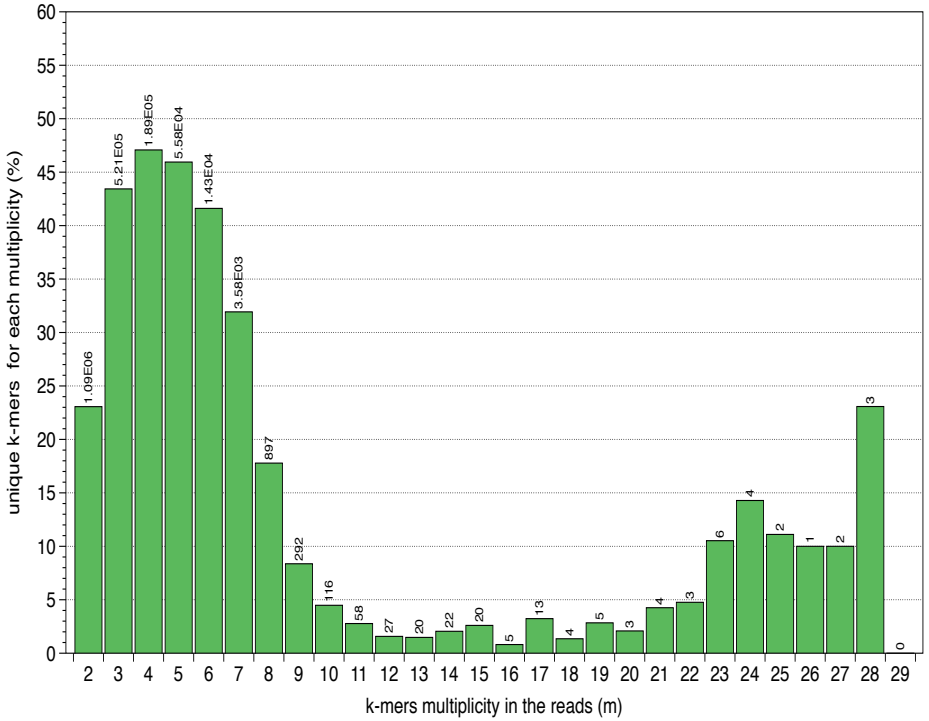
diBELLA: Towards a Long Read Assembler

Long reads (PacBio, etc.)

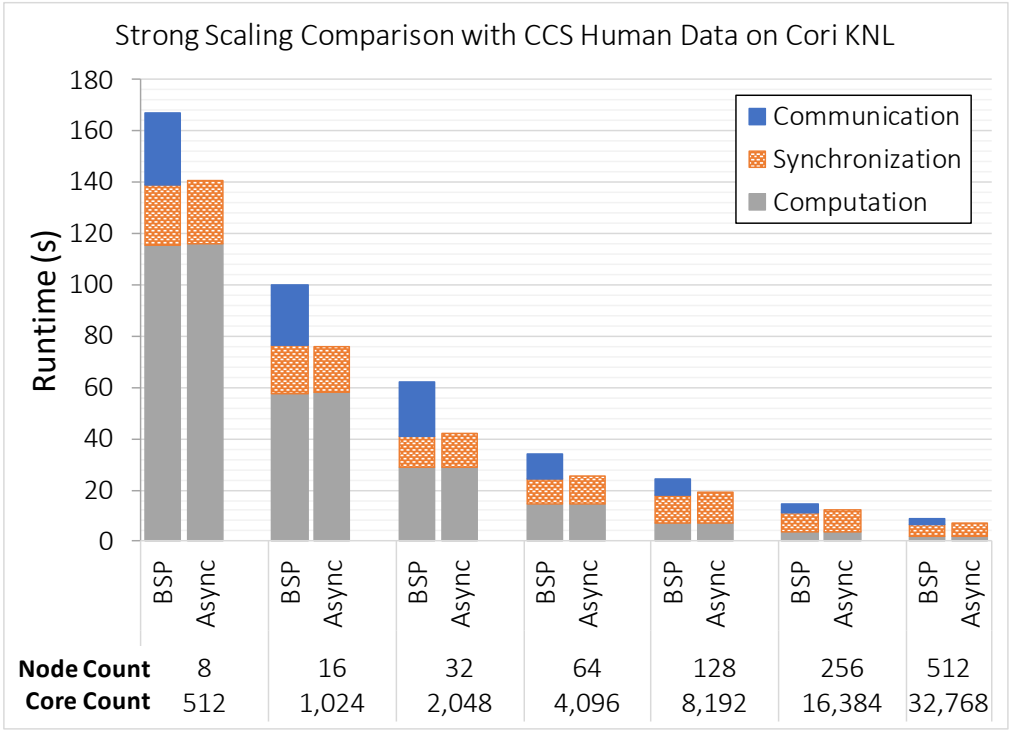
- **Longer alignments**
- **More compute-intensive**
- **More GPU friendly**

No need for De Bruijn graph

- **Pairwise alignments**
- **Filtered k-mers**



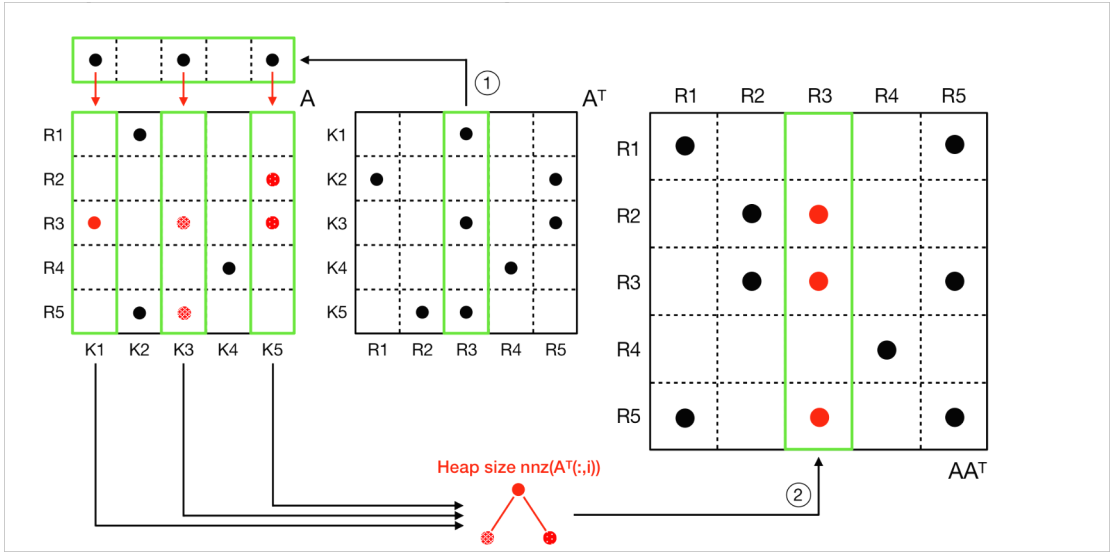
Bulk-Synchronous vs 1-sided Asynchronous



Asynchronous communication hides latency and uses less memory in general

Set Alignment is a Sparse All-to-All

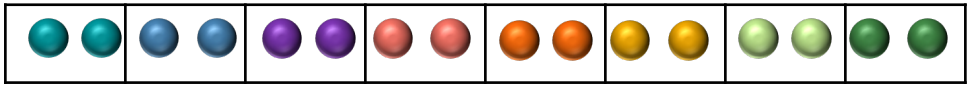
Run expensive alignment on all pairs with a common k-mer



Avoid Communication, Maximize Parallelism

Compute on all pairs of particles or strings, or…

Obvious solution

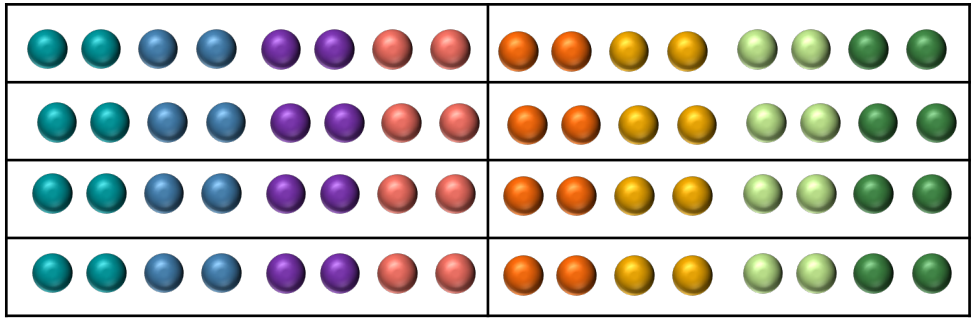


16 particles on 8 processors
Pass all particles around (p steps)

Decreases

- #messages by factor **c^2**
- #volume sent by factor **c**

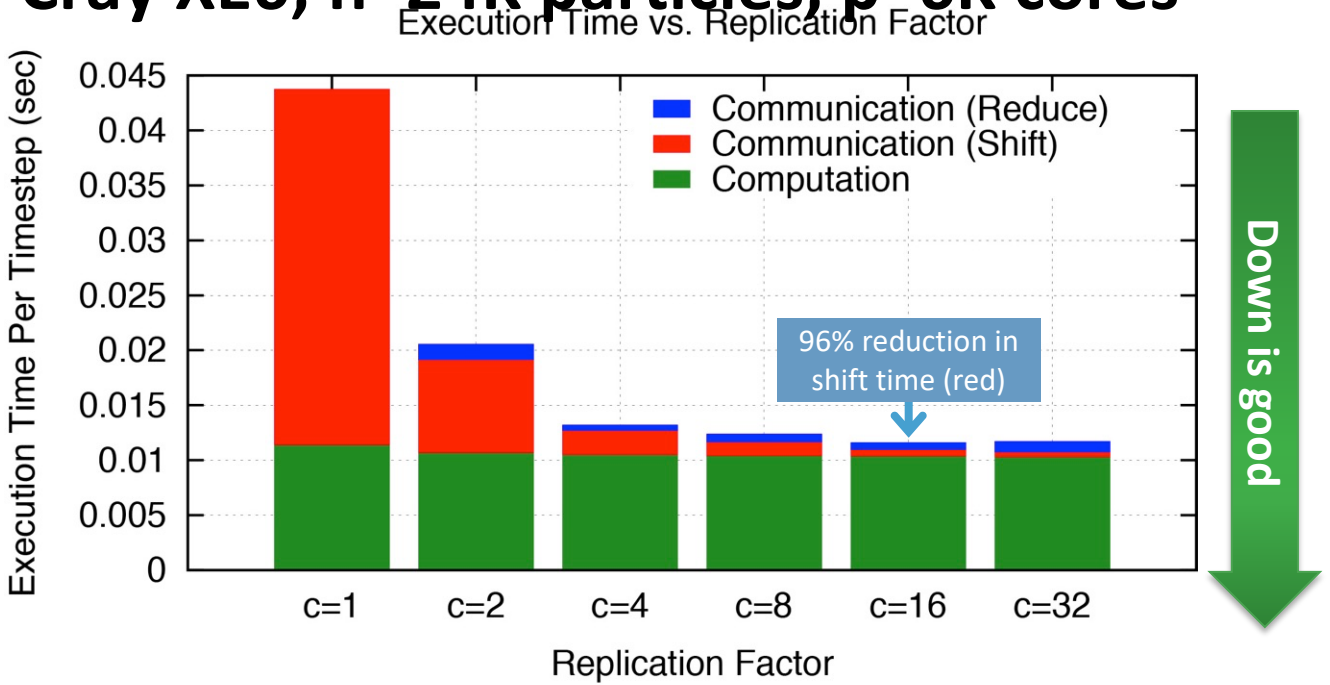
Better solution



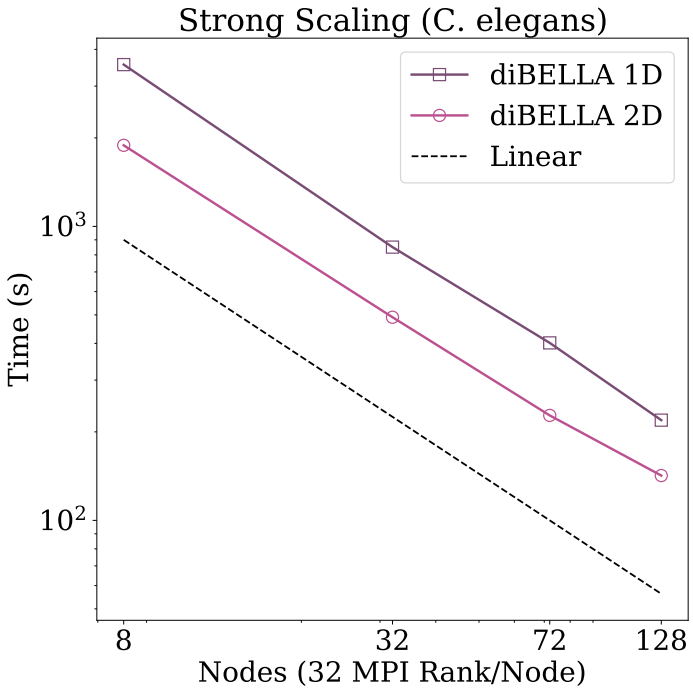
c = 4 copies of particles
8 particles on each

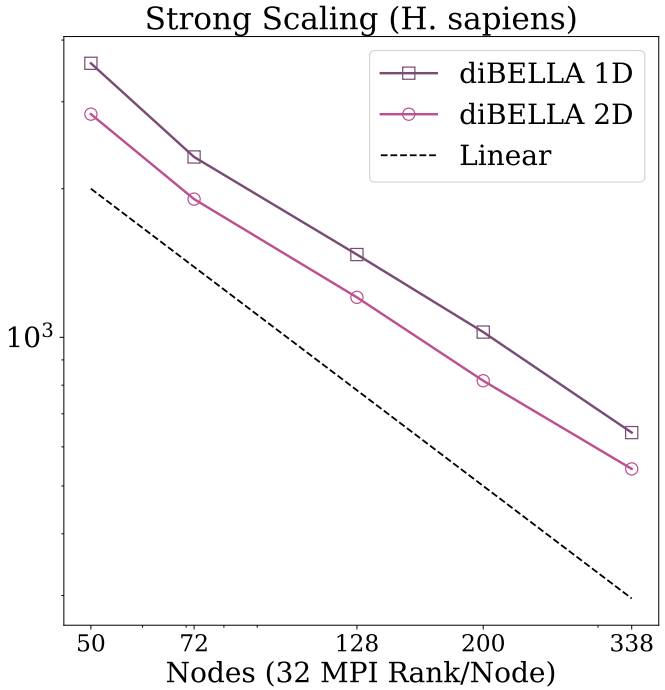
Less Communication..

Cray XE6; n=24K particles, p=6K cores



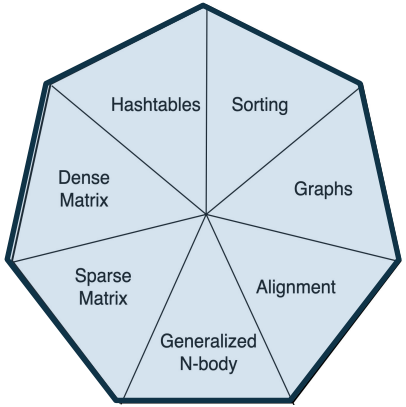
1D vs 2D Algorithm on DNA “overlap”





Lower is Better

G. Guidi, O. Selvitopi†, M. Ellis, L. Oliker, Y, A. Buluc (IPDP ‘21 to appear)

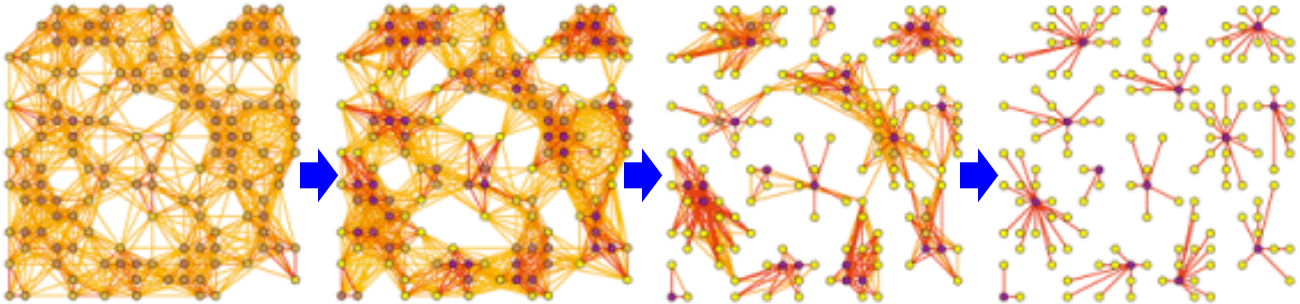


Graphs and Sparse Matrices (unsupervised learning)

Protein Clustering with Sparse Matrices

Input: Adjacency matrix A (sparse)

Image source: http://micans.org/mcl/

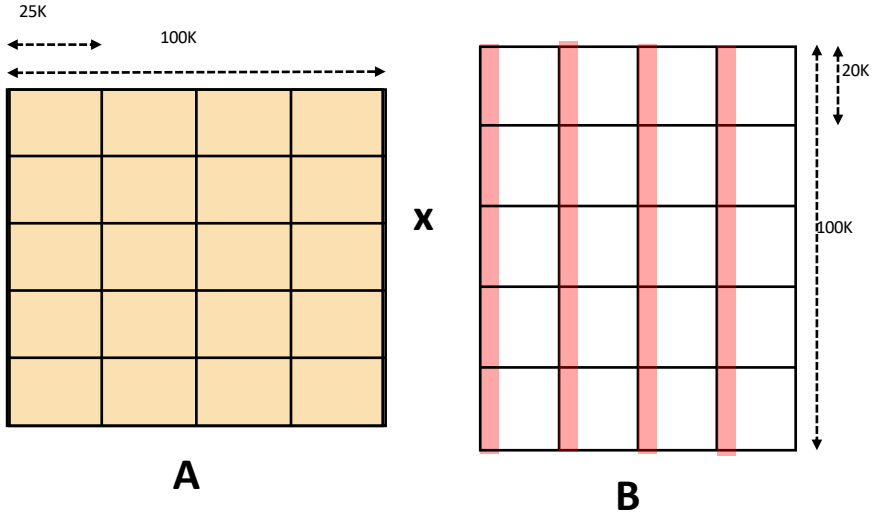


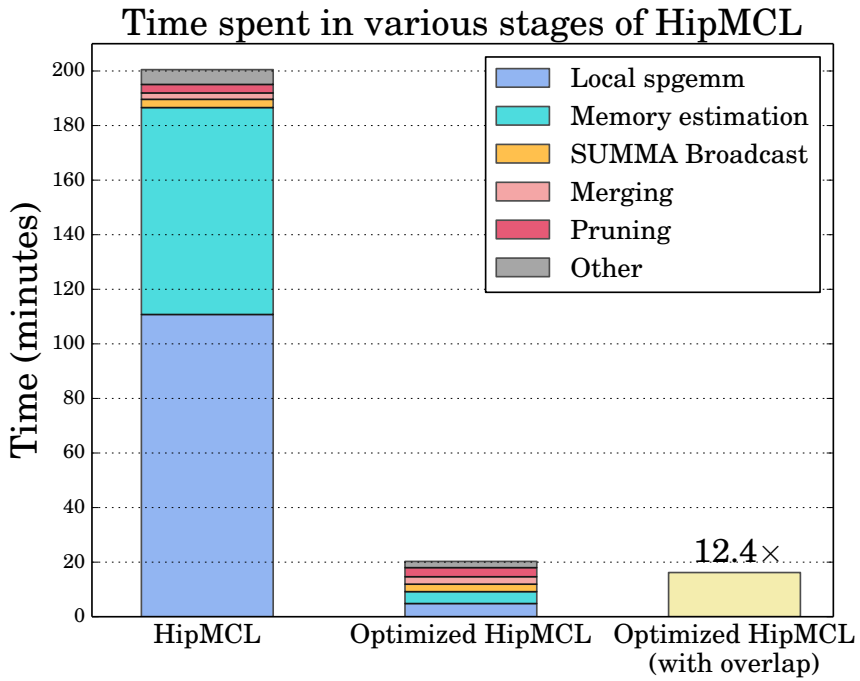
- **Similarity Matrix**: “Many-to-many” protein alignment
- **Expansion**: Square matrix, pruning small entries, dense columns
- **Inflation**: element-wise powers

PASTIS + HipMCL

Oguz Selvitopi; Md Taufique Hussain; Ariful Azad; Aydın Buluç

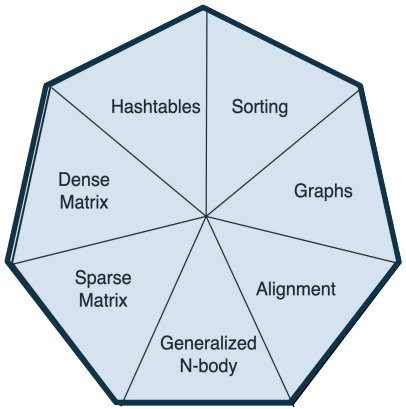
Sparse Matrix Algorithms





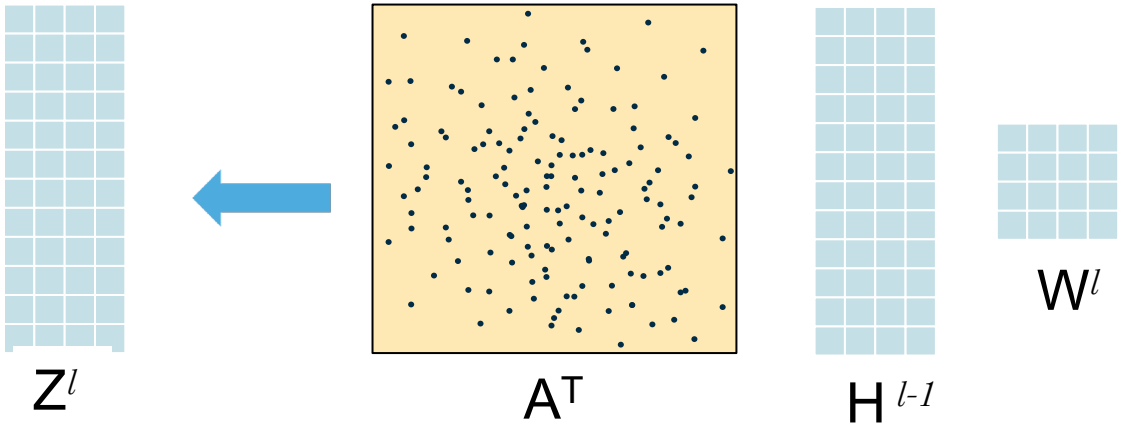
Distributed memory enabled new science

12.4× faster with GPUs!



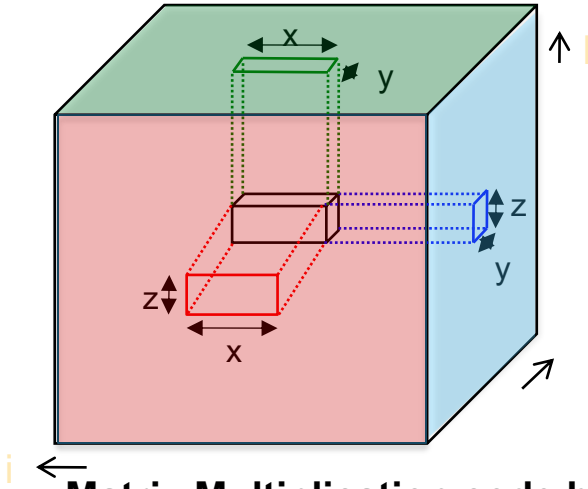
Graphs and Sparse and Dense Matrices (supervised learning)

Bottleneck in GNN Training



- $A^T H^{l-1}$ sparse-dense matmul (SpMM)
- $(A^T H^{l-1}) W^l$ dense-dense matmul (DGEMM)
- **SpMM is the bottleneck, not DGEMM!**

Communication-Avoiding Matrix Multiply

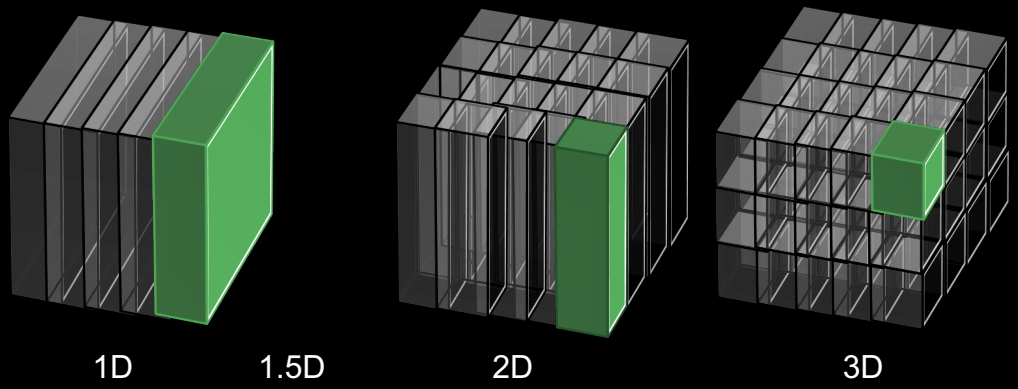


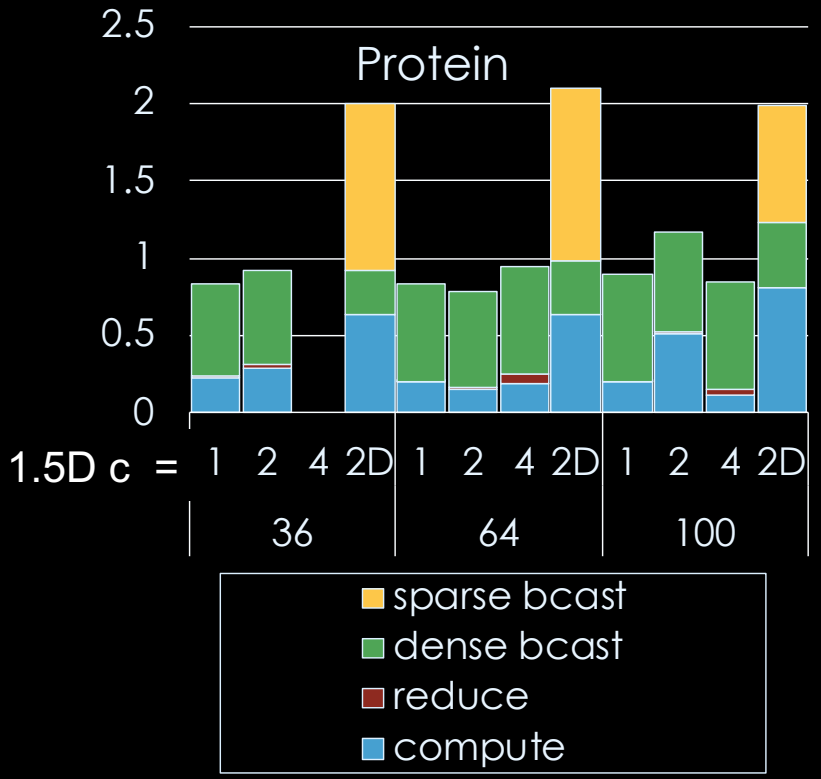
- 2D algorithm: never chop k dim
- 3D: Assume + is associative; chop k, which is → replication of C matrix

Matrix Multiplication code has a 3D iteration space
Each point in the space is a constant computation (*/+)

```
for i
  for j
    for k
          C[i,j] …  A[i,k] …    B[k,j]  …
```

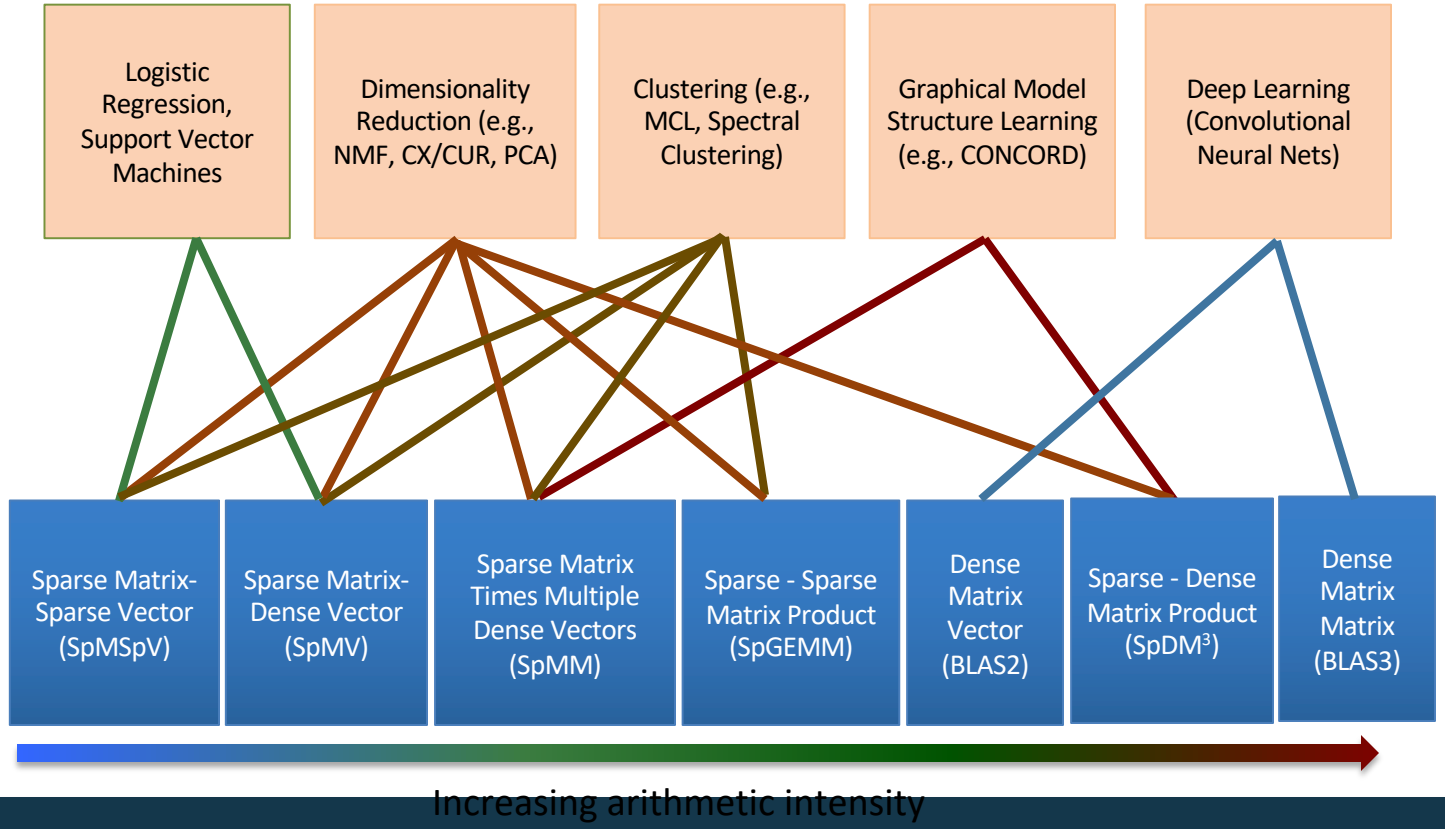
Avoiding Communication in GNNs





Tripathy, Yelick, Buluc, Reducing Communication in Graph Neural Network Training, SC'20

Machine Learning Mapping to Linear Algebra



Take-Aways

- **Applications**
  - More data, more computing can reveal new insights
  - Genomics problems dominated by ~7 motifs
- **Architectures**
  - Specialization and data parallelism will be increasingly important
  - Communication will (still) dominate
  - Need better integration, lower overheads mechanism
- **Algorithms**
  - Irregular, fine-grained problems
  - Can map to distributed memory and data parallelism
  - Avoid communication:
    - Hide latency or aggregating messages (can trade off)
    - Reduce bandwidth (volume), and
    - Use all the wires all the time