India’s Opportunity

Into the world of Exascale Computing & Beyond

EduHiPC 2022

Karthikeyan V
Principal Engineer, Intel

NOTICES & DISCLAIMERS

Performance varies by use, configuration and other factors. Learn more on the Performance Index site.

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. No product or component can be absolutely secure.

Your costs and results may vary.

Intel technologies may require enabled hardware, software or service activation.

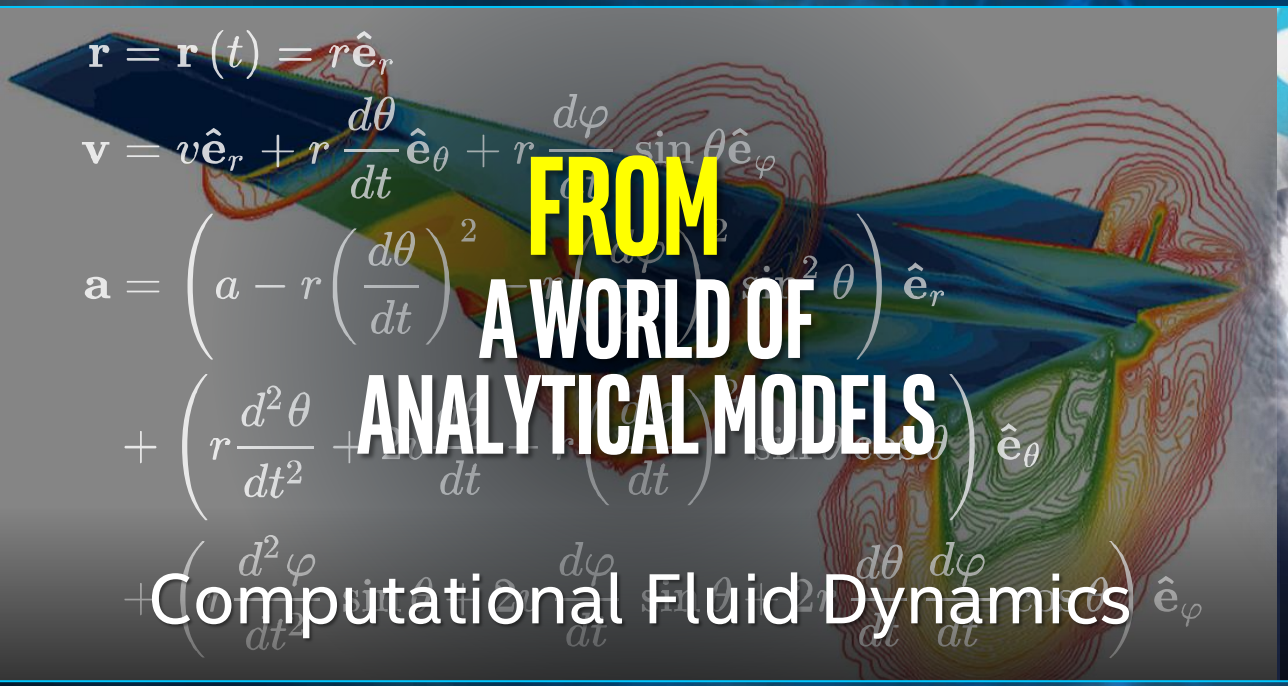
OUTLINE

1. We are at a cusp of unprecedented transformation
2. Compute appetite far exceeds pace of Moore's Law
3. Orders of magnitude opportunity in Algorithms & HW-SW co-design
4. How does India participate in this transformation
   - Curriculum, skillset, infrastructure, research, case studies
5. How do we make India a R&D Hub?

What will India miss?

THE NEW FRONTIER

Inside - Out ⇄ *Outside - In*



FROM A WORLD OF ANALYTICAL MODELS

Computational Fluid Dynamics

Start with Mathematical Model
Model → Simulate → Predict

TO A WORLD OF DATA DRIVEN MODELS

Event Detection from Social Media

Start with Data
Initial State → Increment → Steer

McKinsey Global Institute

The Bio Revolution

The scope & scale of the potential impact on economies & societies appears substantial

60%
Of world's physical inputs could be made using biological means

45%
Of world's disease burden could be addressed

30%
Of private-sector R&D spent in biology-related industries

Example of Applications

Agriculture | Health | Energy & Materials | Consumers e.g. Personalized Diet

Transformative new capabilities…

Bio based material production
Personalized & Precision products & services
Engineered Organisms
Higher Bio based R&D productivity
Bio-machine interfaces & computing

$2T- $4T annual direct economic potential in 2030-2040 …

downstream and secondary impact even more

https://www.mckinsey.com/industries/life-sciences/our-insights/the-bio-revolution-innovations-transforming-economies-societies-and-our-lives

Biology is Eating the World: A Manifesto*

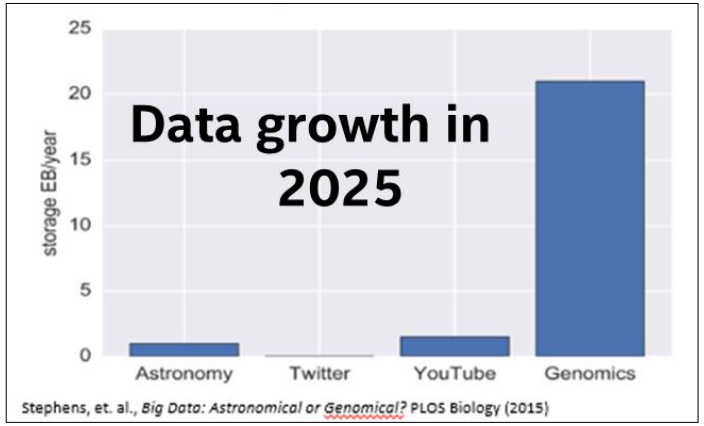
by Jorge Conde, Vijay Pande, and Julie Yoo

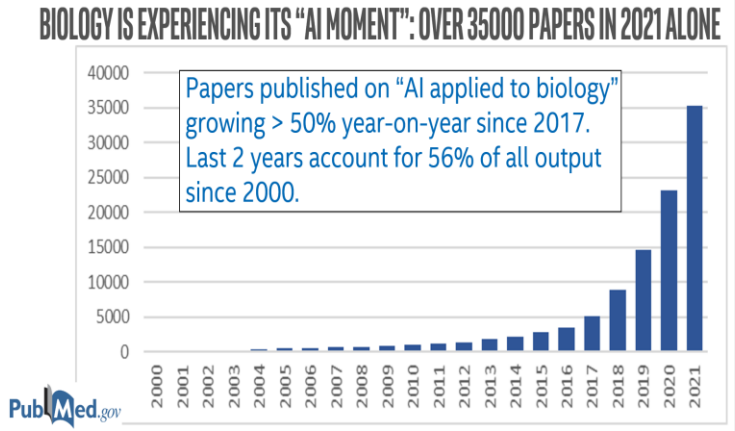
"The future is already here. It's just not evenly distributed yet"

Accumulated genetic variations as we age matter; Mayo finds benefits of genetic testing healthy people (Dec'20)

In 2018, 40% of FDA approved drugs are based on genomics

(Personalized Medicine at FDA)





*https://a16z.com/2019/10/28/biology-eating-world-a16z-manifesto/

intel

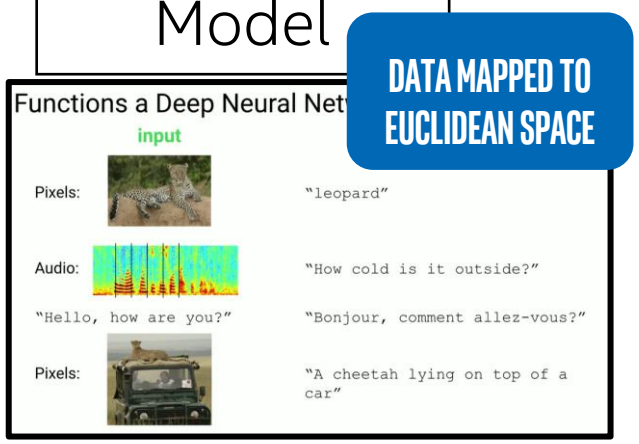
UPCOMING SHIFT IN AI ENABLES SIMULATION & MODELLING COMPLEX SYSTEMS

AI Capability

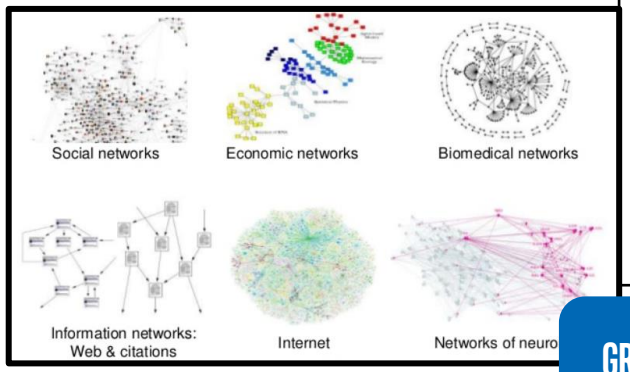
Pattern Detection

Trillion Parameter Model

DATA MAPPED TO EUCLIDEAN SPACE



> 100 Trillion Parameter & growing



Traditional Graph Analytics + AI

AI: Physics Inspired NNs, Meta Learning, Causality, Generative AI, Bayesian + DL, Neural ODE, PDE etc

Complement Traditional HPC

GRAPHS AS INPUT DATA

Today

Near Future

= Massive Appetite for Compute >> 1000x

1. A billion human genomes are expected to be sequenced over next decade
   - Today, 1 SPR process < 30 genomes/day
2. Learn Structure and Causality
   - Use of AI in Simulations
   - >>> Trillions of score computations
3. Search a very large hyper dimensional space >> internet O(10^9)
   - Molecule search > 10^60
   - Protein design > 10^130

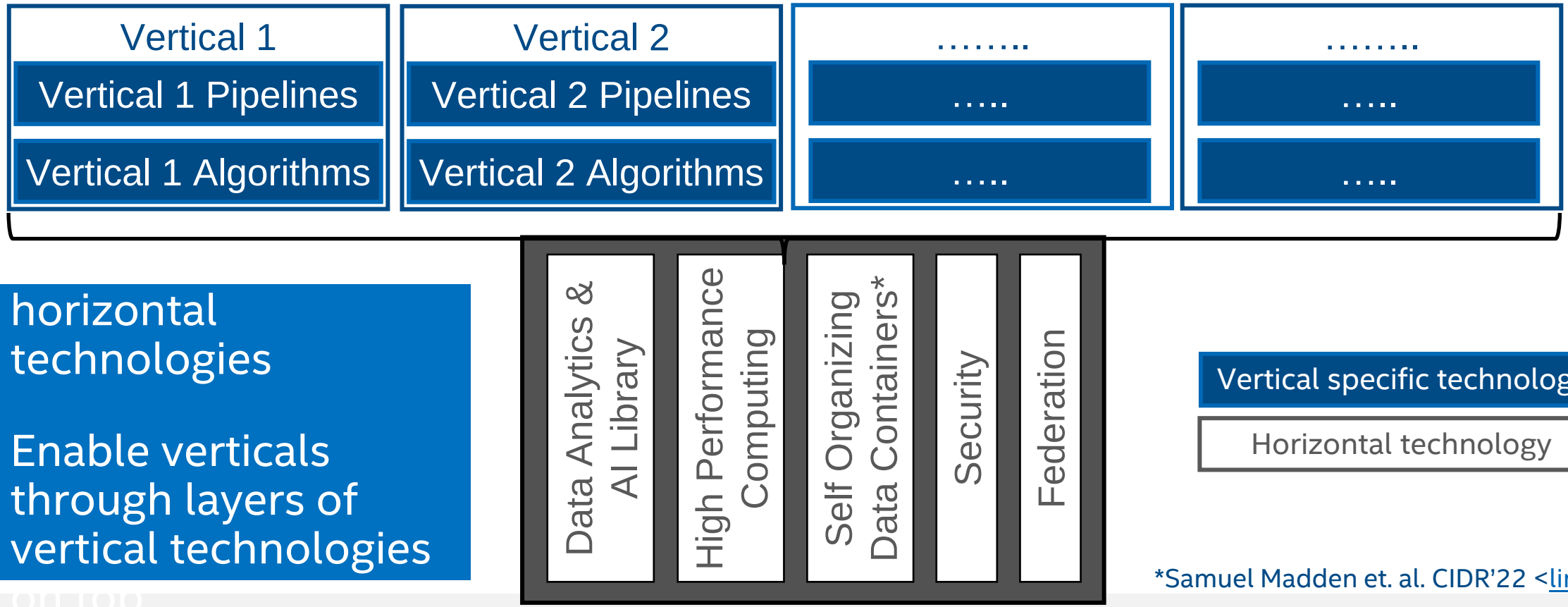
Innovation Needed at All Levels in the Stack

1. Novel methods & algorithms for AI, Biology and HPC
2. Data scientist friendly one-click SW stack
   - Data Management System, Biology, AI, HPC, Security
3. Domain specific architectural acceleration
   - Unique Compute (sparse, irregular)
   - Large cache (>> BW/Capacity)
   - Network Fabric (>>)

- World will go faster than Moore's law for compute performance – Algorithms, HW/SW co-design
- For intelligence @ scale world, Domain & Data scientists are the new developers
- Their productivity (data mountain to actionable intelligence) rests on a one-click SW stack

Our Vision & Opportunity

- Deliver AI pipelines from data ingestion to actionable recommendations that are:
  - one-click, containerized, customizable, adaptive, open-sourced, HW efficient, cloud ready



horizontal technologies

➢ Enable verticals through layers of vertical technologies on top

*Samuel Madden et. al. CIDR'22 <link>

SW opportunity is far bigger

The Next Frontier for AI

- We learn from all, and apply to serve one – all in real-time
  - For example: Learn from driving scenarios globally to navigate locally
- Performance
  - It will require computing and data management at unprecedented scale and speed.
- Productivity
  - Performance alone would not suffice if it compromised the productivity of domain experts and data scientists who are at the forefront of this transformation.

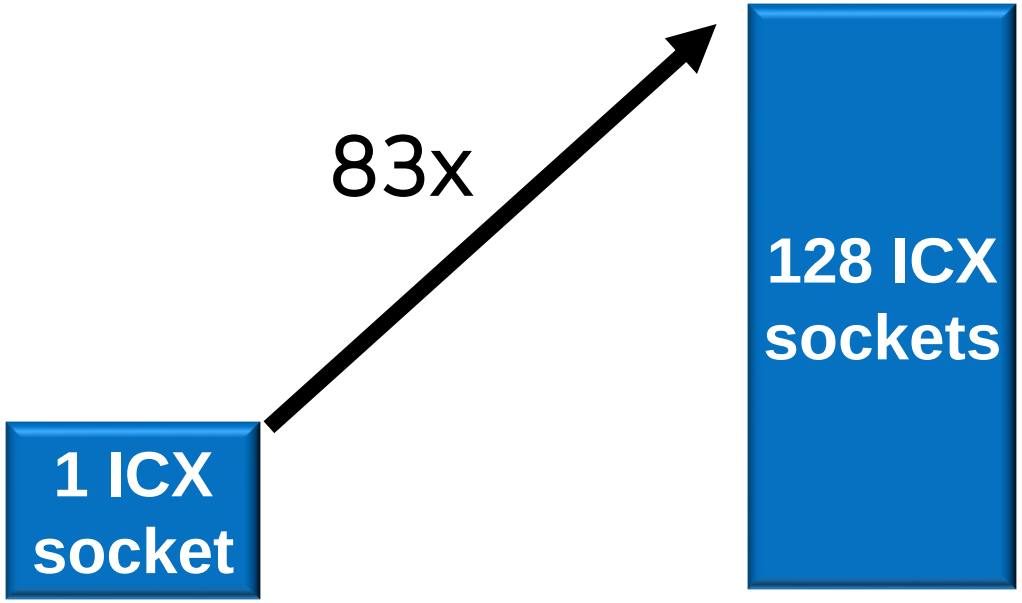
Table 1. Speedups from performance engineering a program that multiplies two 4096-by-4096 matrices. Each version represents a successive refinement of the original Python code. "Running time" is the ru… erations per second that the version executes. "Absolute speedup" is time rel… it of precision, is time relative to the preceding line. "Fraction of peak" is GFLO… ils.

60,000x against Python in 2020**

Version	Implementation	Running time (s)	GFLOPS	Absolute speedup	Relative speedup	Fraction of peak (%)
1	Python	25,552.48	0.005	1	–	0.00
2	Java	2,372.68	0.058	11	10.8	0.01
3	C	542.67	0.253	47	4.4	0.03
4	Parallel loops	69.80	1.969	366	7.8	0.24
5	Parallel divide and conquer	3.80	36.180	6,727	18.4	4.33
6	plus vectorization	1.10	124.914	23,224	3.5	14.96
7	plus AVX intrinsics	0.41	337.812	62,806	2.7	40.45

** Raja Koduri Keynote Hot Chips 2020

GRAPH NEURAL NETWORK



OGB-Papers dataset, 100M Graph Nodes

Full batch training

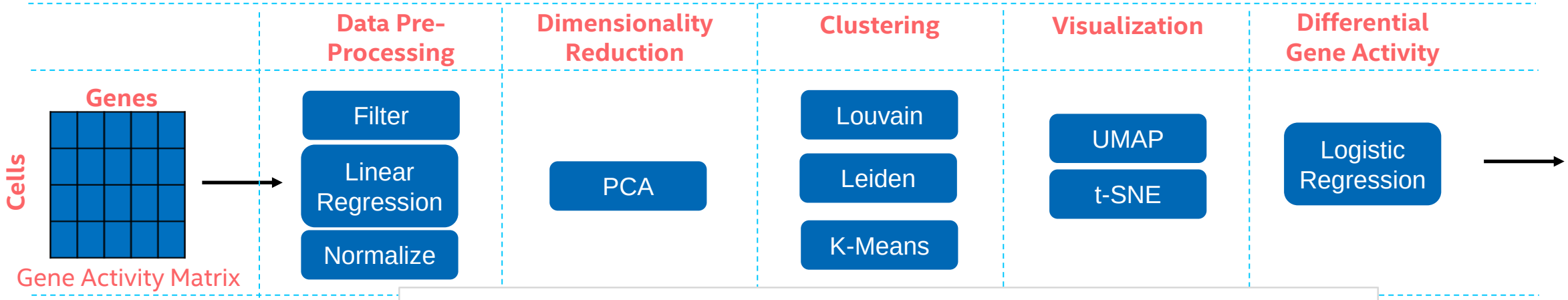
https://arxiv.org/abs/2104.06700v2

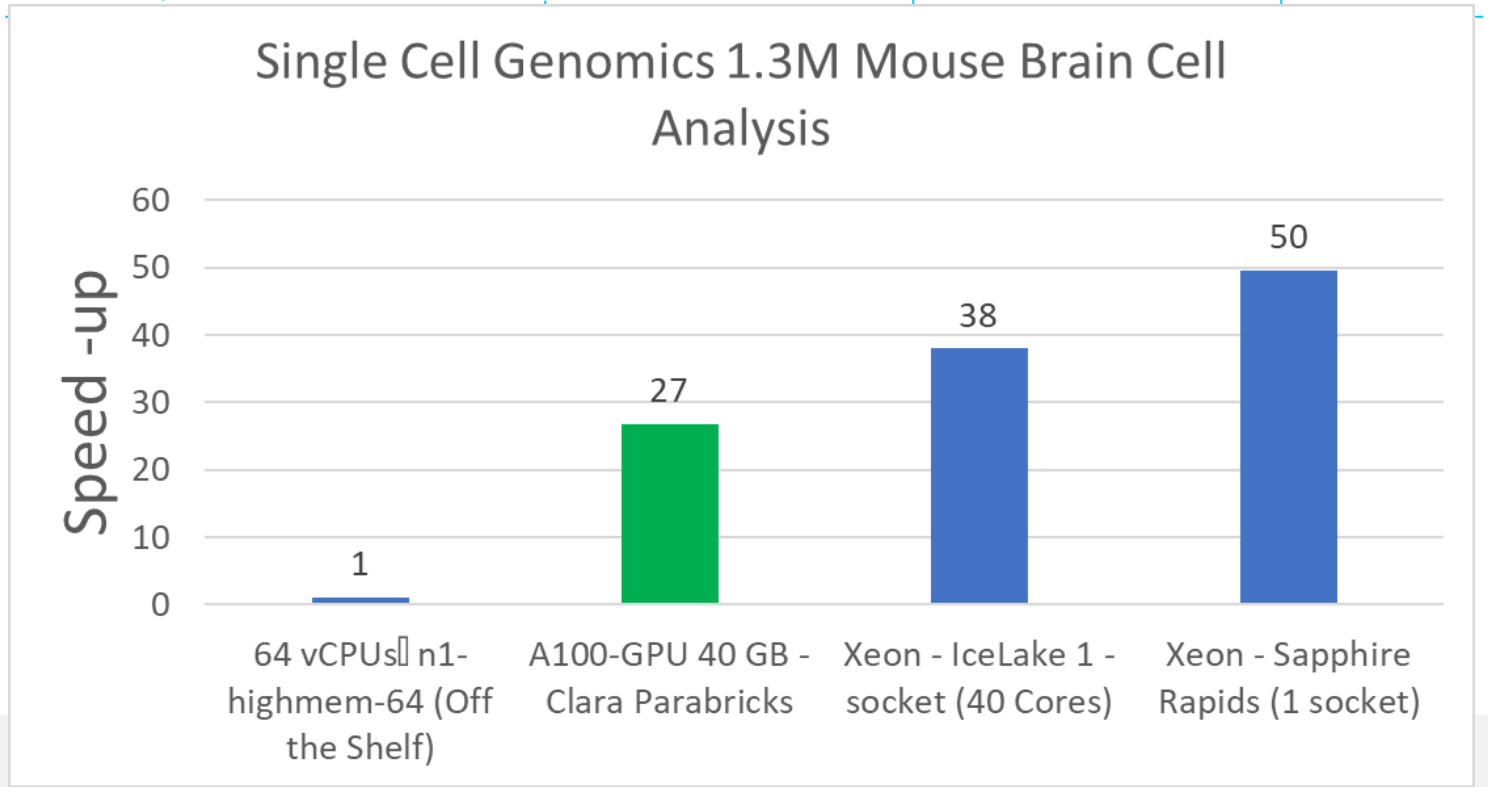
SUPERCOMPUTING'21 - DISTGNN: SCALABLE DISTRIBUTED TRAINING FOR LARGE-SCALE GRAPH NEURAL NETWORKS

Full batch Training ~2-3.7x faster on 1s-CLX (1s) for GraphSAGE on OGB-Products & Reddit ~83x for distributed training on 128 sockets on OGB-Papers

SCANPY: ANALYSIS OF SINGLE CELL RNA-SEQ DATA FOR 1.3 MILLION MOUSE BRAIN CELLS

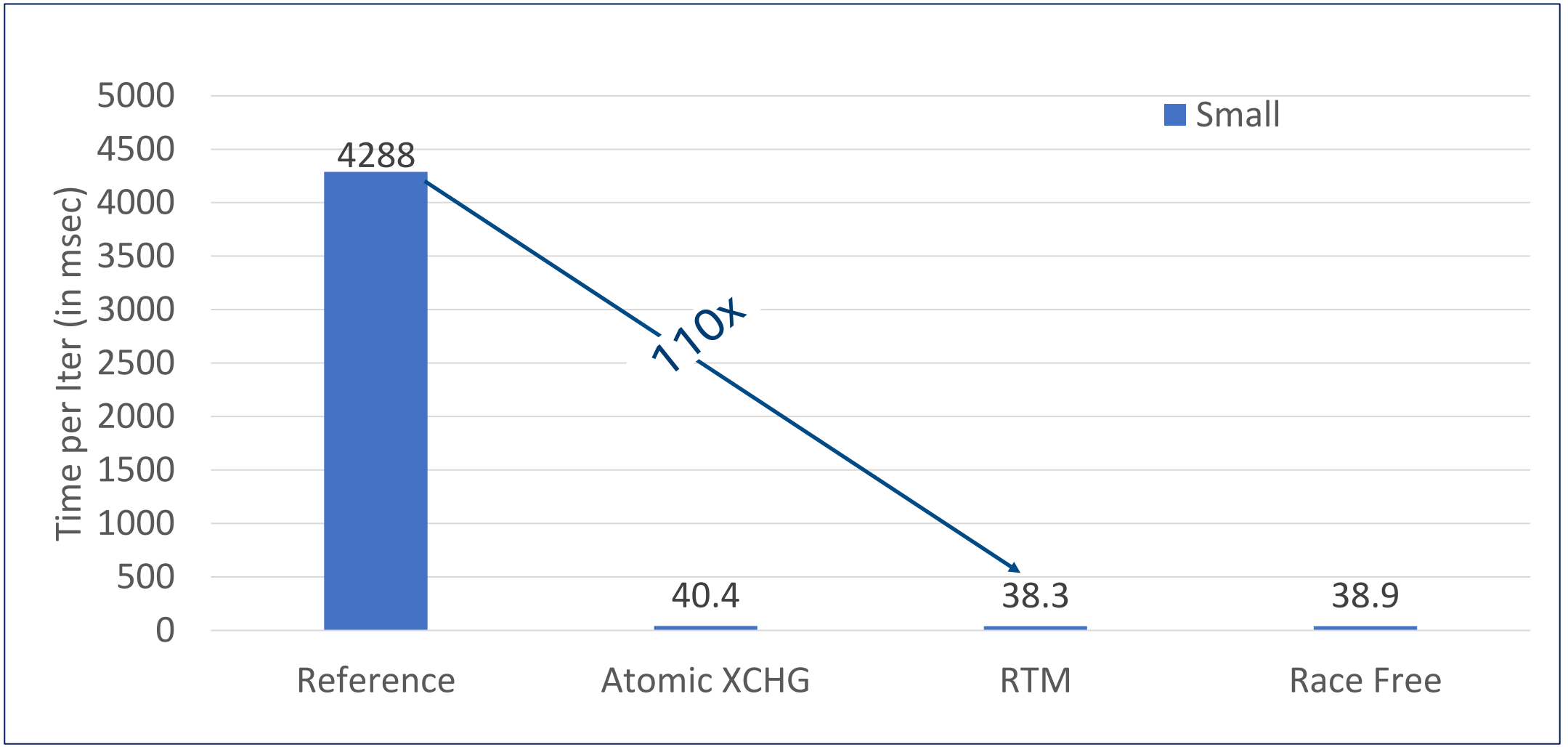
Widely used toolkit for analysis of single cell RNA-Seq data with 1300+ citations, ~600K downloads





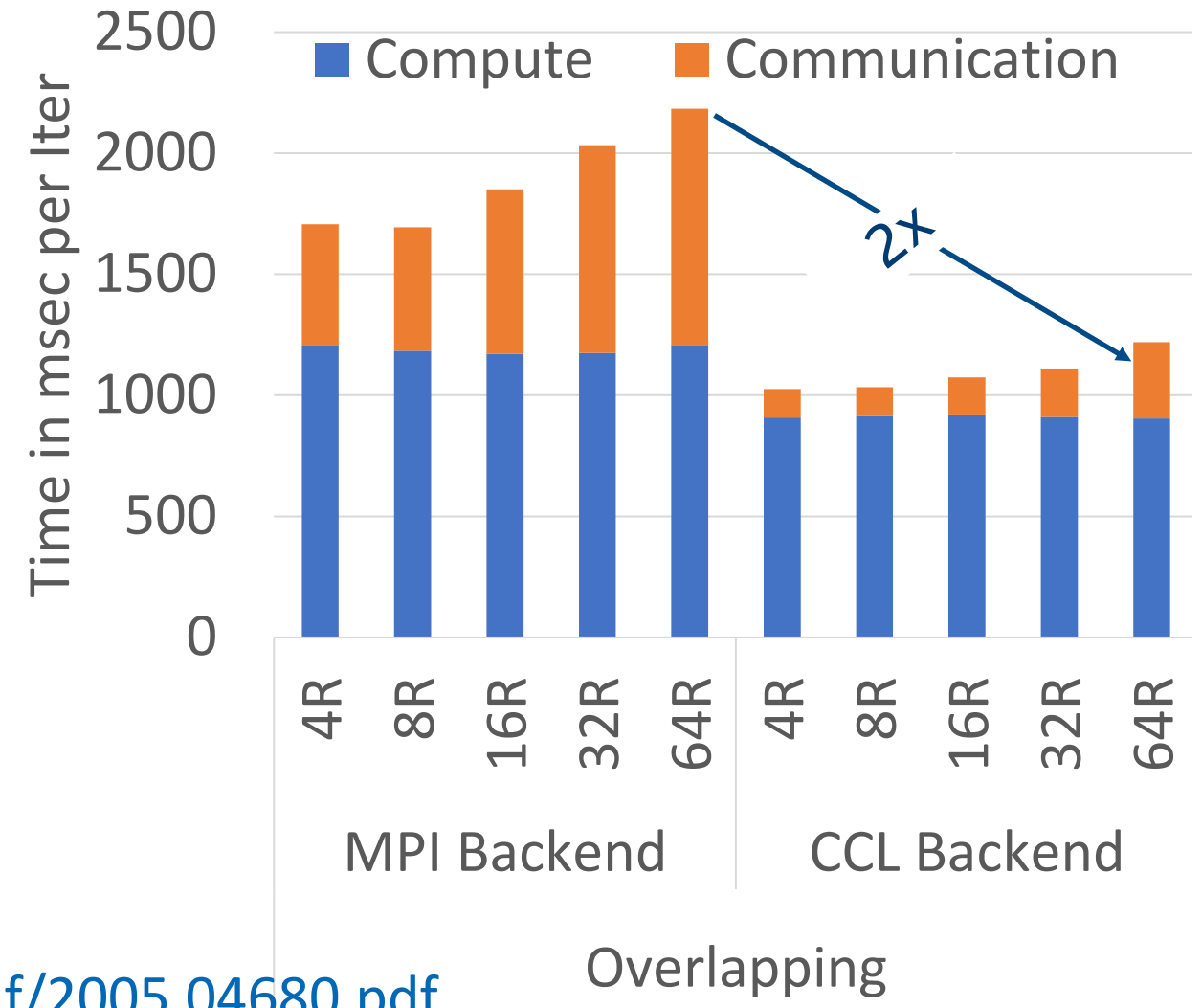
Source: Github repository as of Dec 16, 2020 - Example 2: Single-cell RNA-seq of 1.3 Million Mouse Brain Cells comparing CPU (n1-highmem-64 64 vCPUs) vs GPU (a2-highgpu-1g. https://github.com/clara-parabricks/rapids-single-cell-examples. Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.
1S Ice Lake: See Backup for workloads and configurations. Results may vary.

DEEP LEARNING RECOMMENDATION MODEL (DLRM) ON 1 SOCKET



https://arxiv.org/pdf/2005.04680.pdf

DEEP LEARNING RECOMMENDATION MODEL (DLRM) ON 64 SOCKETS



https://arxiv.org/pdf/2005.04680.pdf

So, what can India do to address this gap?

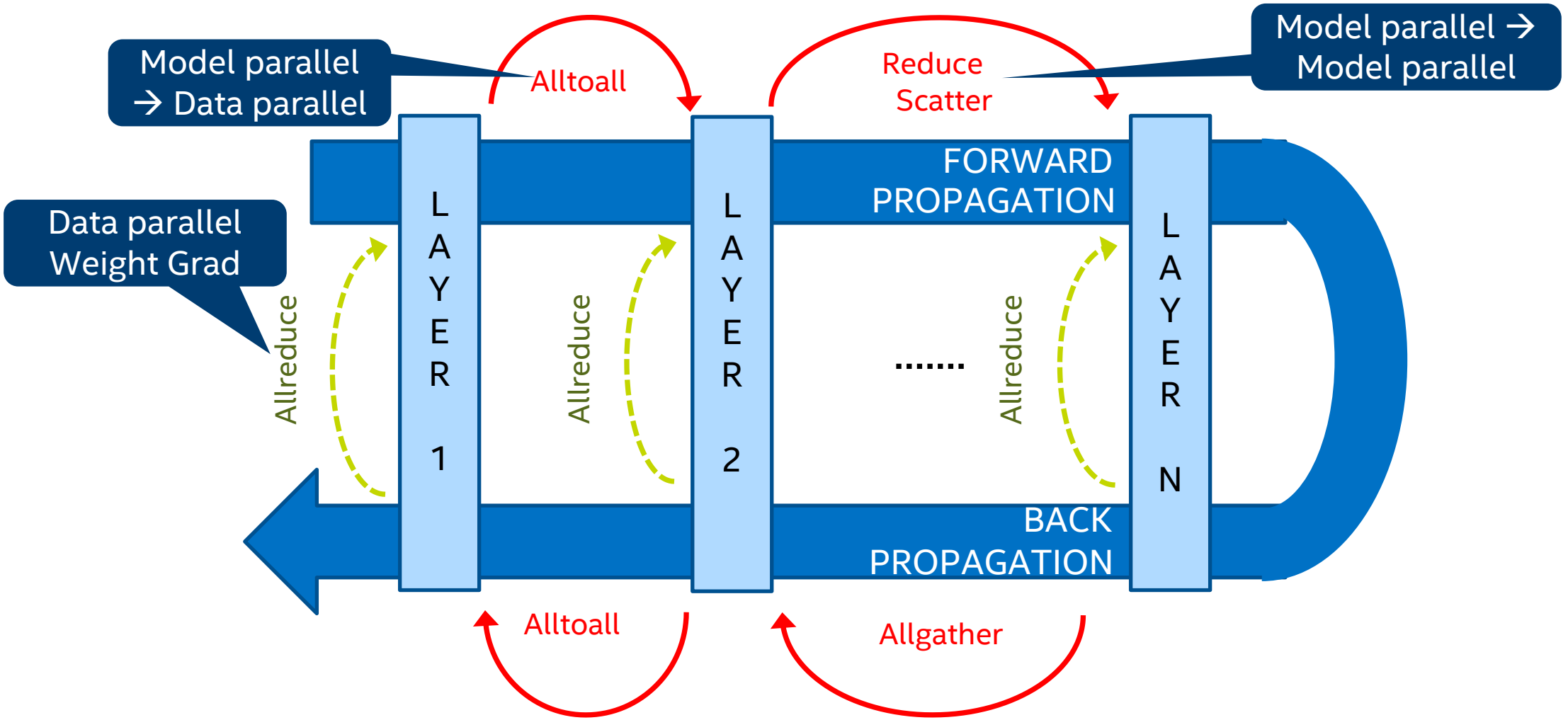
CURRICULUM

1. **Workload characterization**
   - compute-, memory-, network-, disk-bound
2. **Performance optimization recipe**
   - Parallelism, vectorization (SIMD), multithreading, multi-level cache blocking, SW pipelining, loop unrolling, iteration lookahead, Array of structures (AoS) or Structure of Arrays (SoA) or AoSoA data structures, …
3. **Case studies**
   - Matrix multiplication, convolutions, transpose, histogram, stencils, FFT

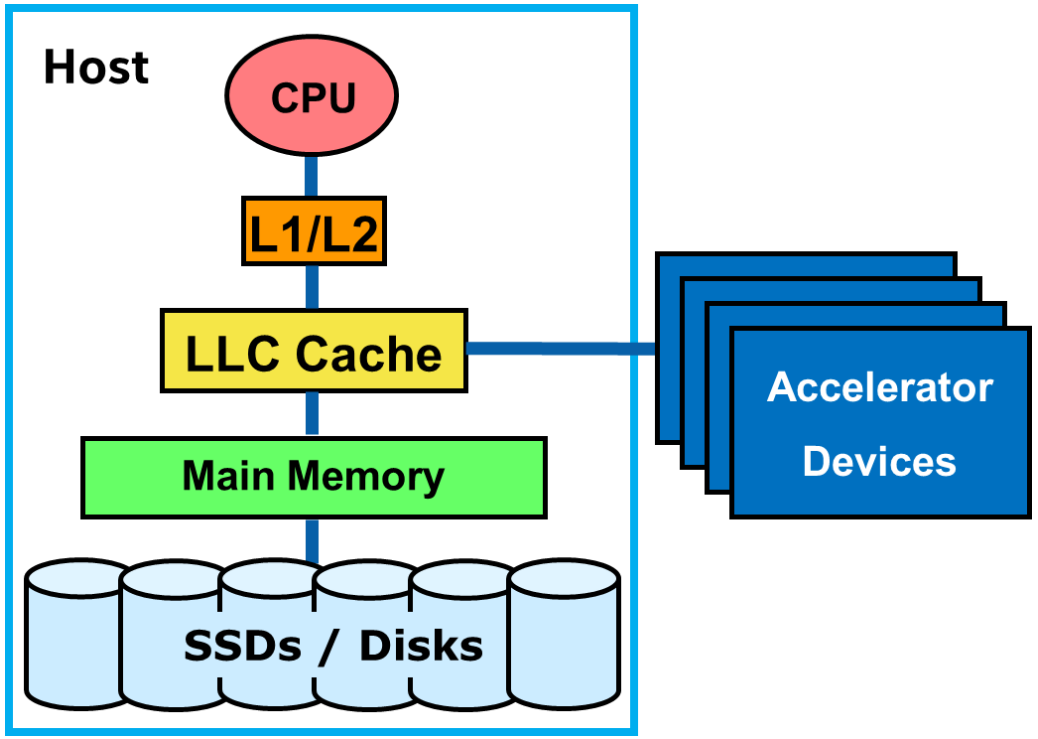
CURRICULUM

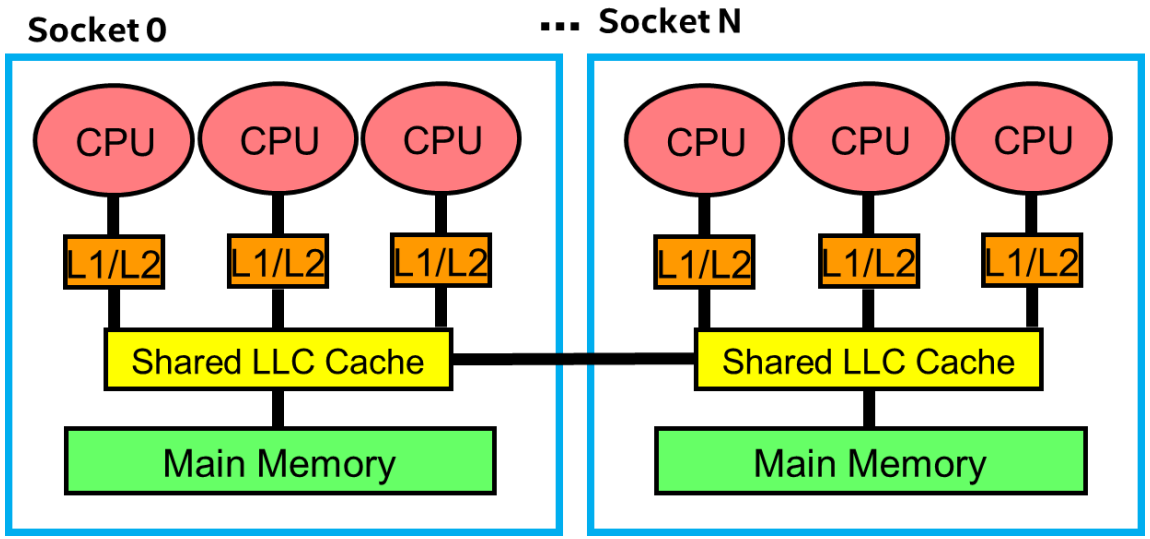
- **Compute bound applications:**
  - Parallelize computations to use more cores
  - Vectorize computations to use more functional units – cannot be ignored
  - Avoid stalls
- **Memory bandwidth bound applications:**
  - Increase data locality (e.g. by blocking)
  - Increase bandwidth efficiency (e.g. reduce randomness)
  - Trade computation for bandwidth
- **Memory latency bound applications:**
  - Latency hiding techniques
  - Avoid cache conflicts, TLB conflicts
  - Avoid synchronization
- **Network bound applications:**
  - Overlap communication with compute
  - Quantization, compression
  - Address load imbalance, congestion
  - Topology awareness, scheduling

CURRICULUM: MULTINODE DEEP LEARNING PARALLELISM

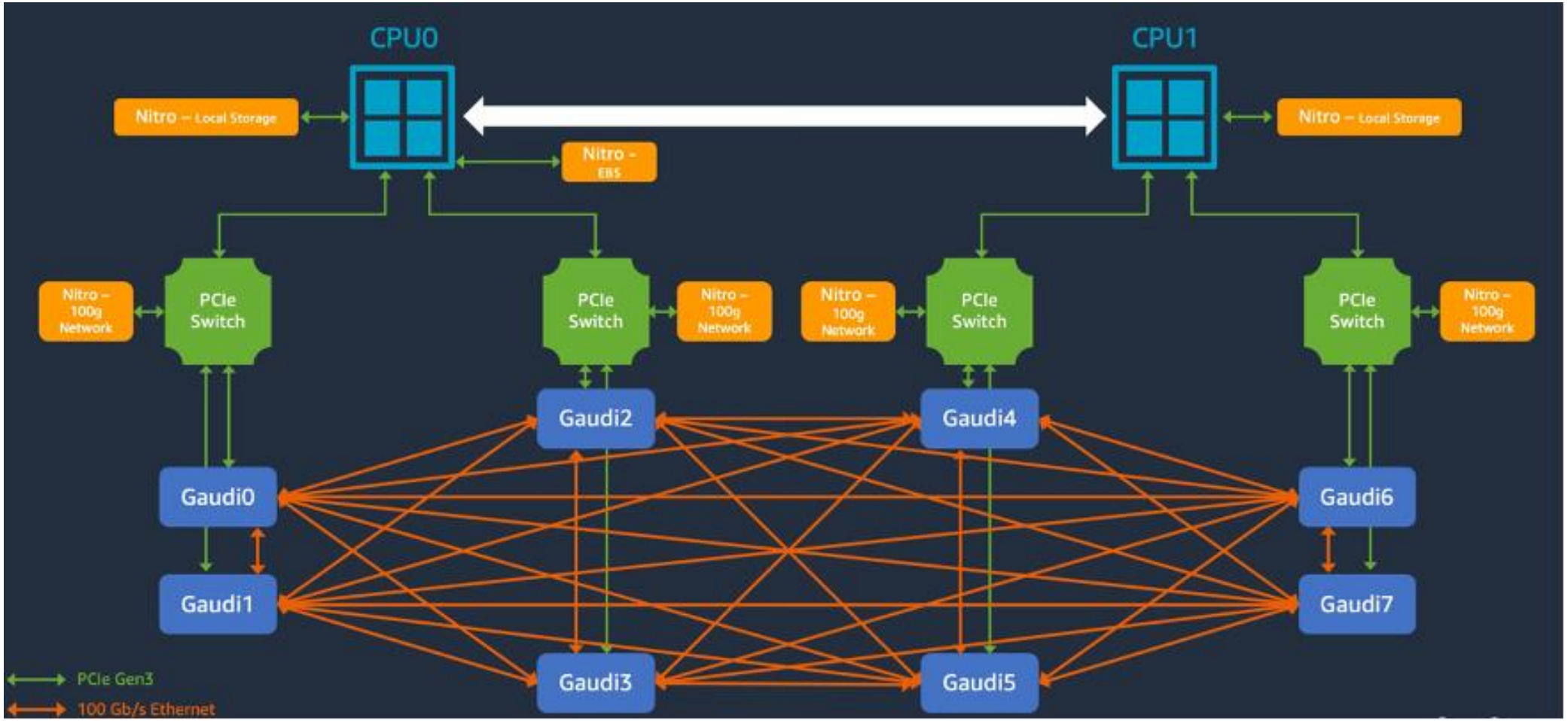


SKILLSET: UNDERSTANDING SYSTEMS





SKILLSET: UNDERSTANDING SYSTEMS



https://d2908q01vomqb2.cloudfront.net/1b6453892473a467d07372d45eb05abc2031647a/2022/03/23/Picture1-7.png

SKILLSET: UNDERSTANDING NETWORKING AND LARGE-SCALE TOPOLOGY

TACC
TEXAS ADVANCED COMPUTING CENTER

23.5 Petaflops
8K HDR InfiniBand Nodes
Fat-Tree Topology

KMA Korea Meteorological Administration

50 Petaflops
7.2K HDR InfiniBand Nodes
Dragonfly+ Topology

Australian National University

3K HDR InfiniBand Nodes
Dragonfly+ Topology

MISSISSIPPI STATE UNIVERSITY

3.1 Petaflops
1.8K HDR InfiniBand Nodes
Fat-Tree Topology

CSC
FINNISH METEOROLOGICAL INSTITUTE

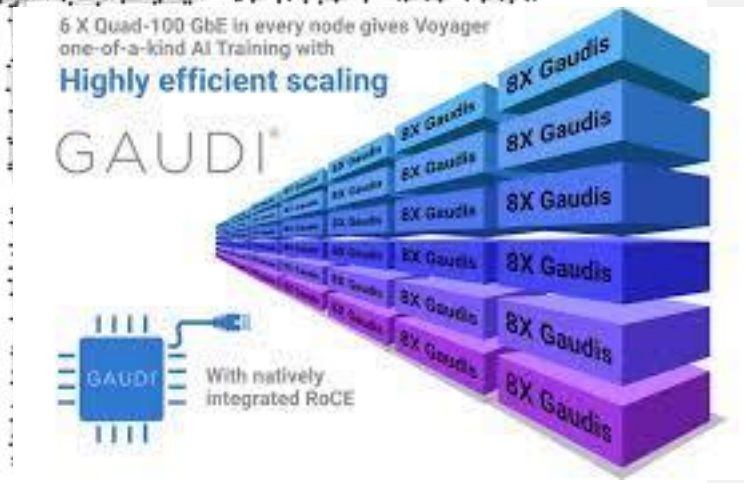
1.7 Petaflops
2K HDR InfiniBand Nodes
Dragonfly+ Topology

Microsoft Azure

Highest Performance Cloud

SDSC
SAN DIEGO SUPERCOMPUTER CENTER

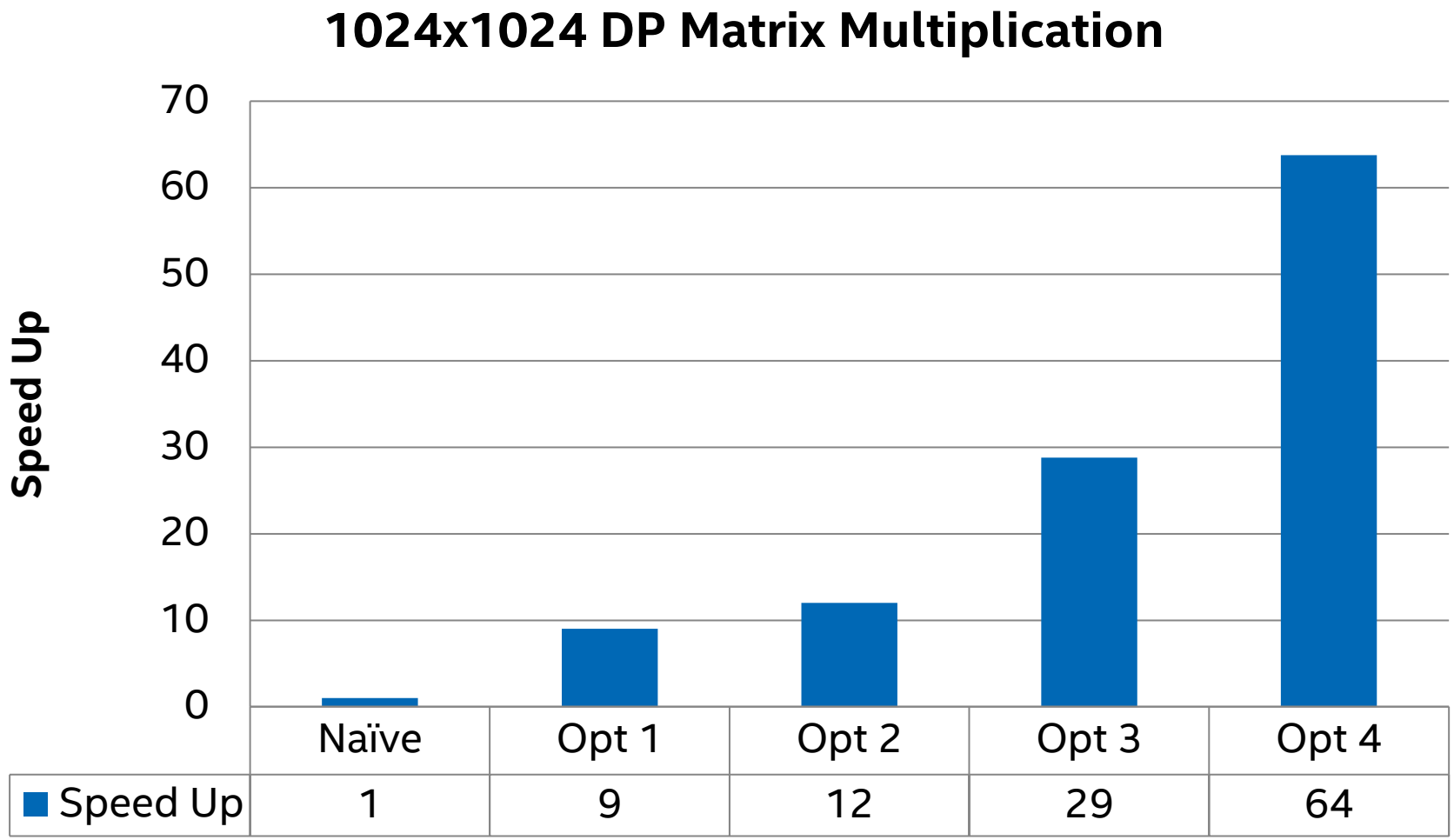
PITTSBURGH SUPERCOMPUTING CENTER



https://cse.iitk.ac.in/users/pmalakar/acmws/networks.pdf

https://www.sdsc.edu/News%20Items/PR20210409_Voyager.html

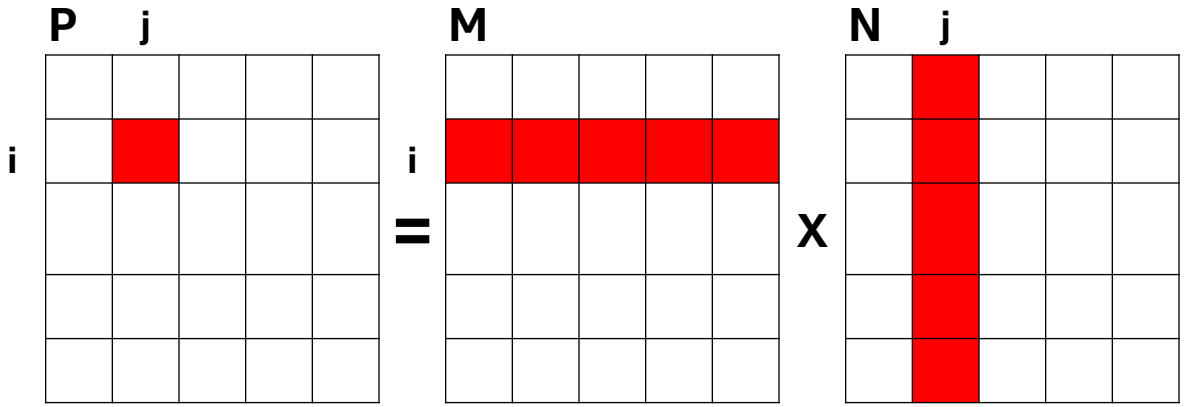
Case Study I – Matrix Multiplication



MATRIX MULTIPLICATION

```
// Core  of matrix multiplication

   for(int i = 0; i < Width; ++i) {
      for(int j = 0; j < Width; ++j) {
         for(int k = 0; k < Width; ++k) {
            P[i][j] += M[i][k] * N[k][j];
         }
      }
   }
```



PROBLEMS WITH NAÏVE IMPLEMENTATION

- Column-wise access
  - Cache miss on every access - high latency
  - Poor bandwidth utilization
    - Only 1/8 of cache line brought in is utilized
- Repeated loading of data into cache
- Uses only one core out of four
- Does not use full compute

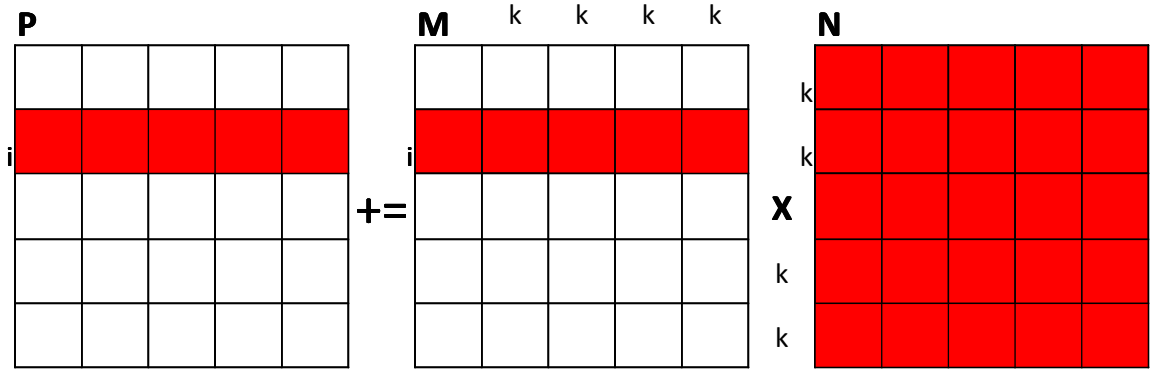
OPTIMIZATIONS

1. Loop Inversion
2. Tiling
3. Parallelization
4. Vectorization & Pipeline Specific Optimization

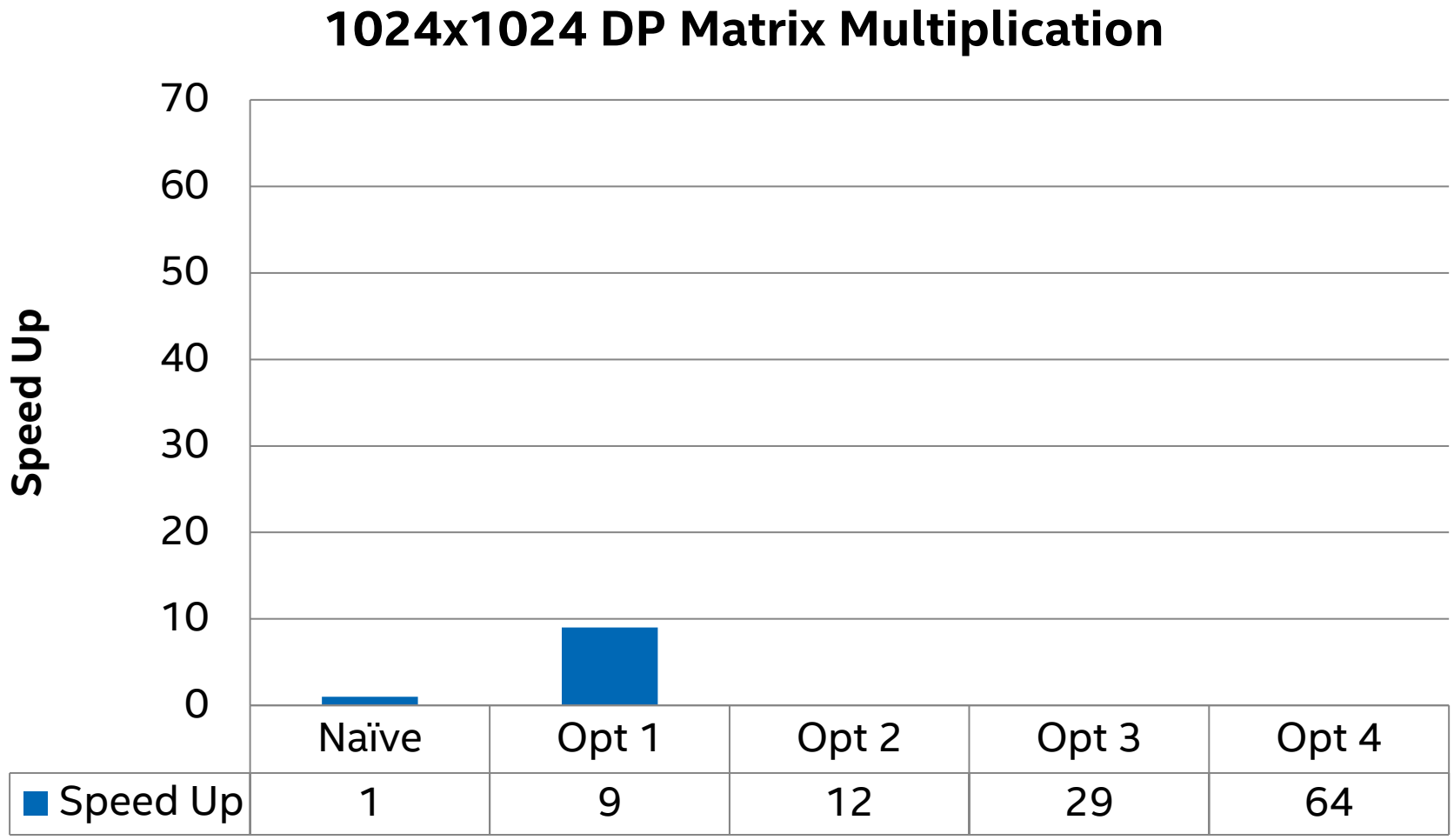
OPT 1: LOOP INVERSION

```
// Cache friendly matrix multiplication with loop order
changed

    for(int i = 0; i < Width; ++i) {
        for(int k = 0; k < Width; ++k) {
            for(int j = 0; j < Width; ++j) {
                P[i][j] += M[i][k] * N[k][j];
            }
        }
    }
```



MATRIX MULTIPLICATION – PERFORMANCE ON CPU*

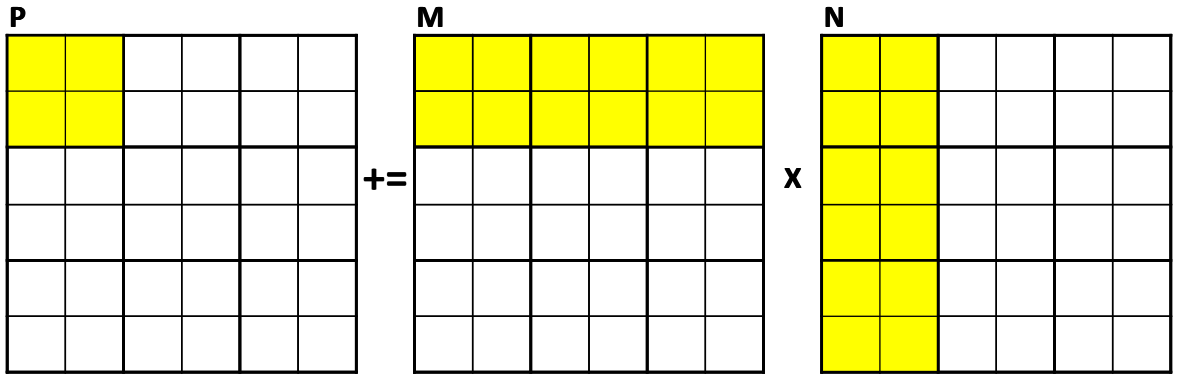


*** Intel Core i7 4C @ 3.3GHz, 6GB DDR2 DRAM**

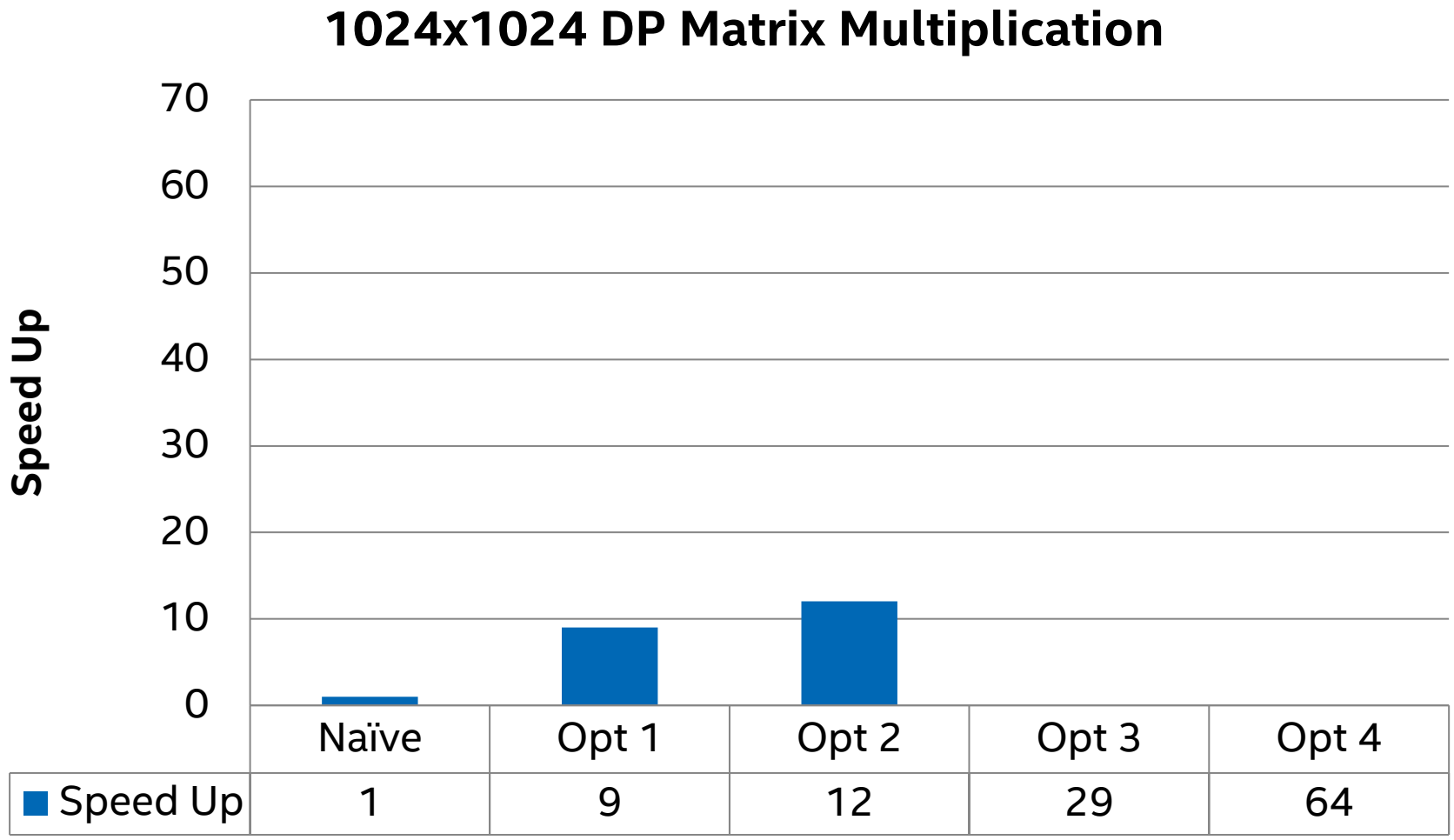
OPT 2: TILING

```
// Core of matrix multiplication


for(int a = 0; a < Width; a += TILE_SIZE) {
   for(int b = 0; b < Width; b += TILE_SIZE) {
      for(int c = 0; c < Width; c += TILE_SIZE) {
         for(int i = a; i < a + TILE_SIZE; ++i) {
            for(int k = c; k < c + TILE_SIZE; ++k) {
               for(int j = b; j < b + TILE_SIZE; ++j) {
                  P[i][j] += M[i][k] * N[k][j];
               }
            }
         }
      }
   }
}
```



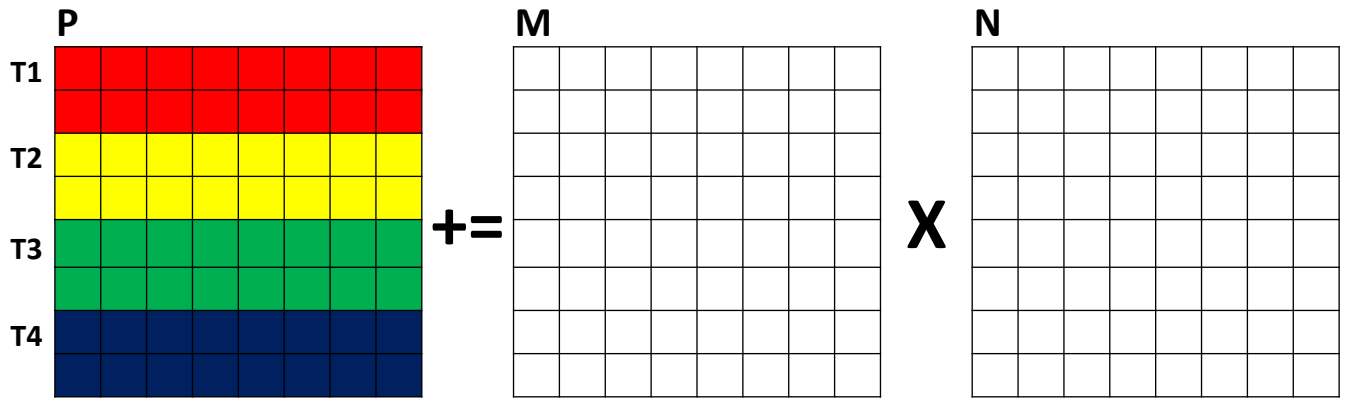
MATRIX MULTIPLICATION – PERFORMANCE ON CPU*



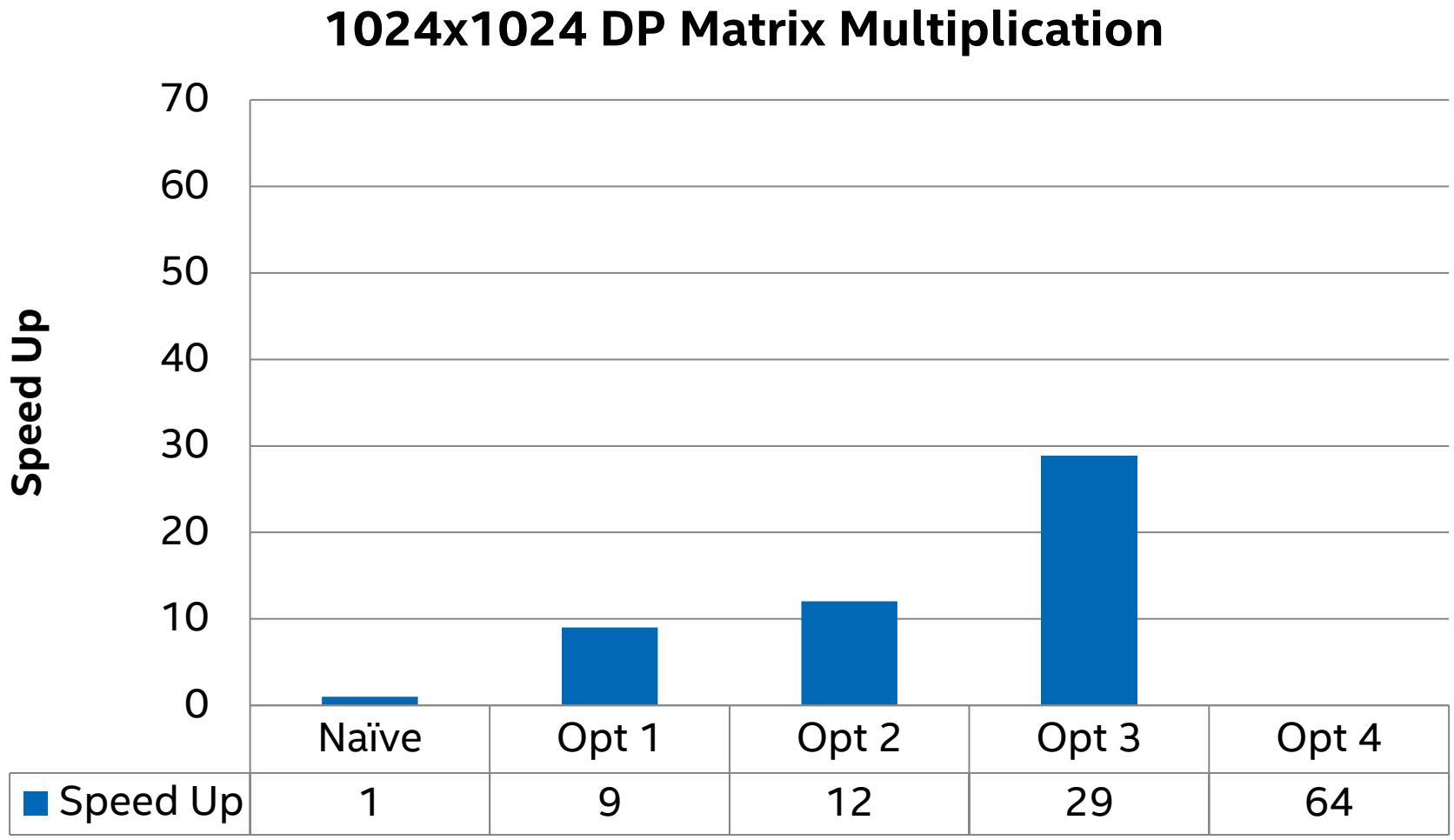
*** Intel Core i7 4C @ 3.3GHz, 6GB DDR2 DRAM**

OPT 3: PARALLELIZATION

```
// Core  of matrix multiplication
#pragma omp parallel for
for(int a = 0; a < Width; a += TILE_SIZE) {
   for(int b = 0; b < Width; b += TILE_SIZE) {
      for(int c = 0; c < Width; c += TILE_SIZE) {
         for(int i = a; i < a + TILE_SIZE; ++i) {
            for(int k = c; k < c + TILE_SIZE; ++k) {
               for(int j = b; j < b + TILE_SIZE; ++j) {
                  P[i][j] += M[i][k] * N[k][j];
               }
            }
         }
      }
   }
}
```



MATRIX MULTIPLICATION – PERFORMANCE ON CPU*



*** Intel Core i7 4C @ 3.3GHz, 6GB DDR2 DRAM**

OPT 4.1: SIMD

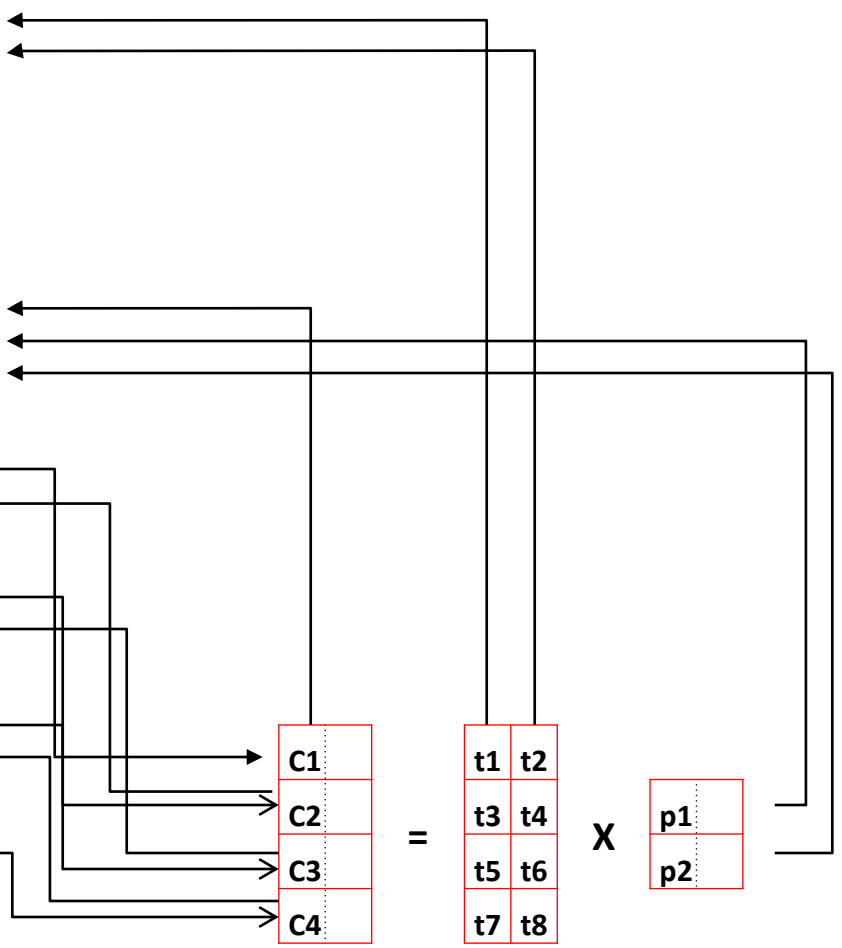
- Compiler auto-vectorizes code by default

```
..B1.45:
        movsd     m1(%r14,%rsi,8), %xmm0
        xorl      %edx, %edx
        unpcklpd  %xmm0, %xmm0
..B1.46:
        movaps    m2(%r12,%rdx,8), %xmm1
        movaps    16+m2(%r12,%rdx,8), %xmm2
        movaps    32+m2(%r12,%rdx,8), %xmm3
        movaps    48+m2(%r12,%rdx,8), %xmm4
        movaps    64+m2(%r12,%rdx,8), %xmm5
        movaps    80+m2(%r12,%rdx,8), %xmm6
        movaps    96+m2(%r12,%rdx,8), %xmm7
        movaps    112+m2(%r12,%rdx,8), %xmm8
        mulpd     %xmm0, %xmm1
        mulpd     %xmm0, %xmm2
        mulpd     %xmm0, %xmm3
        mulpd     %xmm0, %xmm4
        mulpd     %xmm0, %xmm5
        mulpd     %xmm0, %xmm6
        mulpd     %xmm0, %xmm7
        mulpd     %xmm0, %xmm8
        addpd     m3(%rax,%rdx,8), %xmm1
        addpd     16+m3(%rax,%rdx,8), %xmm2
        addpd     32+m3(%rax,%rdx,8), %xmm3
        addpd     48+m3(%rax,%rdx,8), %xmm4
        addpd     64+m3(%rax,%rdx,8), %xmm5
        addpd     80+m3(%rax,%rdx,8), %xmm6
        addpd     96+m3(%rax,%rdx,8), %xmm7
        addpd     112+m3(%rax,%rdx,8), %xmm8
        movaps    %xmm1, m3(%rax,%rdx,8)
        movaps    %xmm2, 16+m3(%rax,%rdx,8)
        movaps    %xmm3, 32+m3(%rax,%rdx,8)
        movaps    %xmm4, 48+m3(%rax,%rdx,8)
        movaps    %xmm5, 64+m3(%rax,%rdx,8)
        movaps    %xmm6, 80+m3(%rax,%rdx,8)
        movaps    %xmm7, 96+m3(%rax,%rdx,8)
        movaps    %xmm8, 112+m3(%rax,%rdx,8)
        addq      $16, %rdx
        cmpq      $64, %rdx
        jb        ..B1.46
```

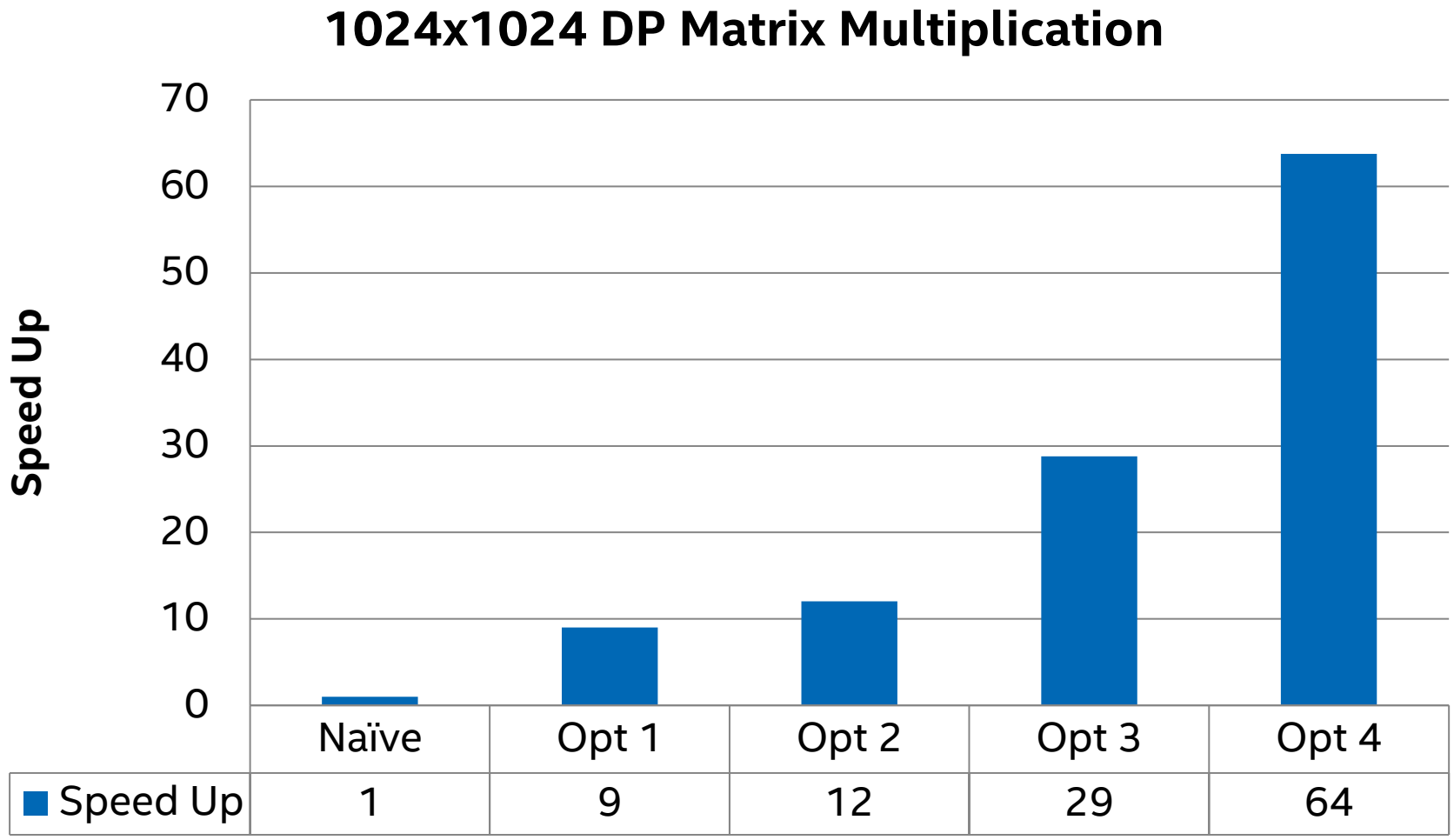
Can we do better?

OPT 4.2: SIMD + COMPUTEOPTIMIZATIONS

```
#pragma omp parallel for
for(a = 0; a < Width; a += TILE_SIZE) {
   for(b = 0; b < Width; b += TILE_SIZE) {
      for(c = 0; c < Width; c += TILE_SIZE) {
         for(i = a; i < a + TILE_SIZE; i+=4) {
            for(k = c; k < c + TILE_SIZE; k+=2) {
               __m128d t1 = _mm_loaddup_pd(&m1[i][k]);
               __m128d t2 = _mm_loaddup_pd(&m1[i][k+1]);
               __m128d t3 = _mm_loaddup_pd(&m1[i+1][k]);
               __m128d t4 = _mm_loaddup_pd(&m1[i+1][k+1]);
               __m128d t5 = _mm_loaddup_pd(&m1[i+2][k]);
               __m128d t6 = _mm_loaddup_pd(&m1[i+2][k+1]);
               __m128d t7 = _mm_loaddup_pd(&m1[i+3][k]);
               __m128d t8 = _mm_loaddup_pd(&m1[i+3][k+1]);
               for(j = b; j < b + TILE_SIZE; j+=2) {
                  __m128d C1 = _mm_load_pd(&m3[i][j]);
                  __m128d p1 = _mm_load_pd(&m2[k][j]);
                  __m128d p2 = _mm_load_pd(&m2[k+1][j]);
                  C1 = _mm_add_pd(C1, _mm_mul_pd(t1, p1));
                  C1 = _mm_add_pd(C1, _mm_mul_pd(t2, p2));
                  *((__m128d*)&m3[i][j]) = C1;
                  __m128d C2 = _mm_load_pd(&m3[i+1][j]);
                  C2 = _mm_add_pd(C2, _mm_mul_pd(t3, p1));
                  C2 = _mm_add_pd(C2, _mm_mul_pd(t4, p2));
                  *((__m128d*)&m3[i+1][j]) = C2;
                  __m128d C3 = _mm_load_pd(&m3[i+2][j]);
                  C3 = _mm_add_pd(C3, _mm_mul_pd(t5, p1));
                  C3 = _mm_add_pd(C3, _mm_mul_pd(t6, p2));
                  *((__m128d*)&m3[i+2][j]) = C3;
                  __m128d C4 = _mm_load_pd(&m3[i+3][j]);
                  C4 = _mm_add_pd(C4, _mm_mul_pd(t7, p1));
                  C4 = _mm_add_pd(C4, _mm_mul_pd(t8, p2));
                  *((__m128d*)&m3[i+3][j]) = C4;
               }
            }
         }
      }
   }
}
```



MATRIX MULTIPLICATION – PERFORMANCE ON CPU*



*** Intel Core i7 4C @ 3.3GHz, 6GB DDR2 DRAM**

MULTINODE OPTIMIZATIONS IN DEEP LEARNING (DL)

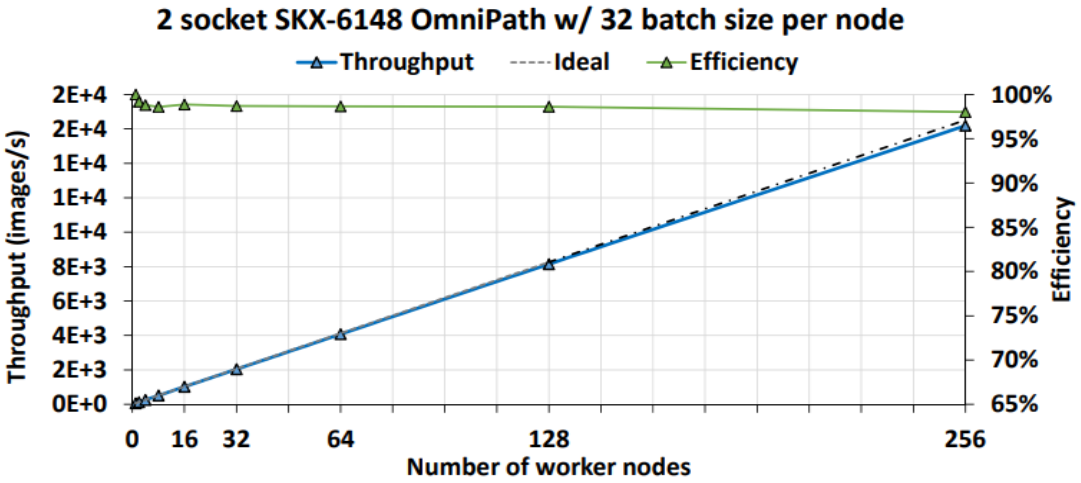
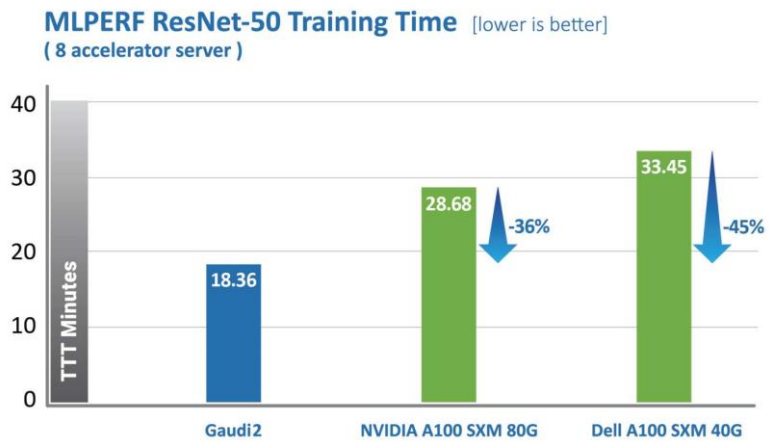


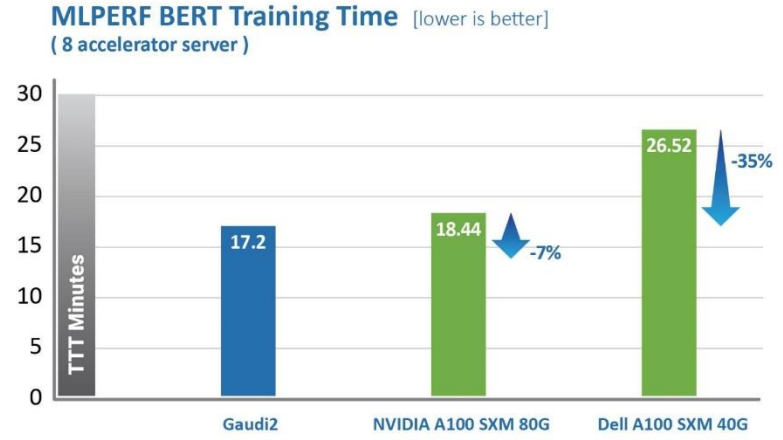
Figure 2: Resnet-50 scaling on Intel Xeon / Omnipath

https://arxiv.org/pdf/1801.08030.pdf

- Drives network efficiently with best bandwidth, Low latency
- Overlaps compute with comms with asynchronous progress
- Supports prioritized channels
- Data-types relevant for DL training
- Quantization/compression hooks
- Communication fusion
- Sparse data communication

Topology	# of Gaudi accelerators	V1.0 Submission Time-to-Train (min)	V1.1 Submission Time-to-Train (min)
ResNet50	8	62.55	56.13
	16		33.25
	32		17.38
	64		9.48
	128		5.41
	256		3.39
BERT-Large	8	164.37	80.47
	16		42.29
	32		21.28
	64		11.87





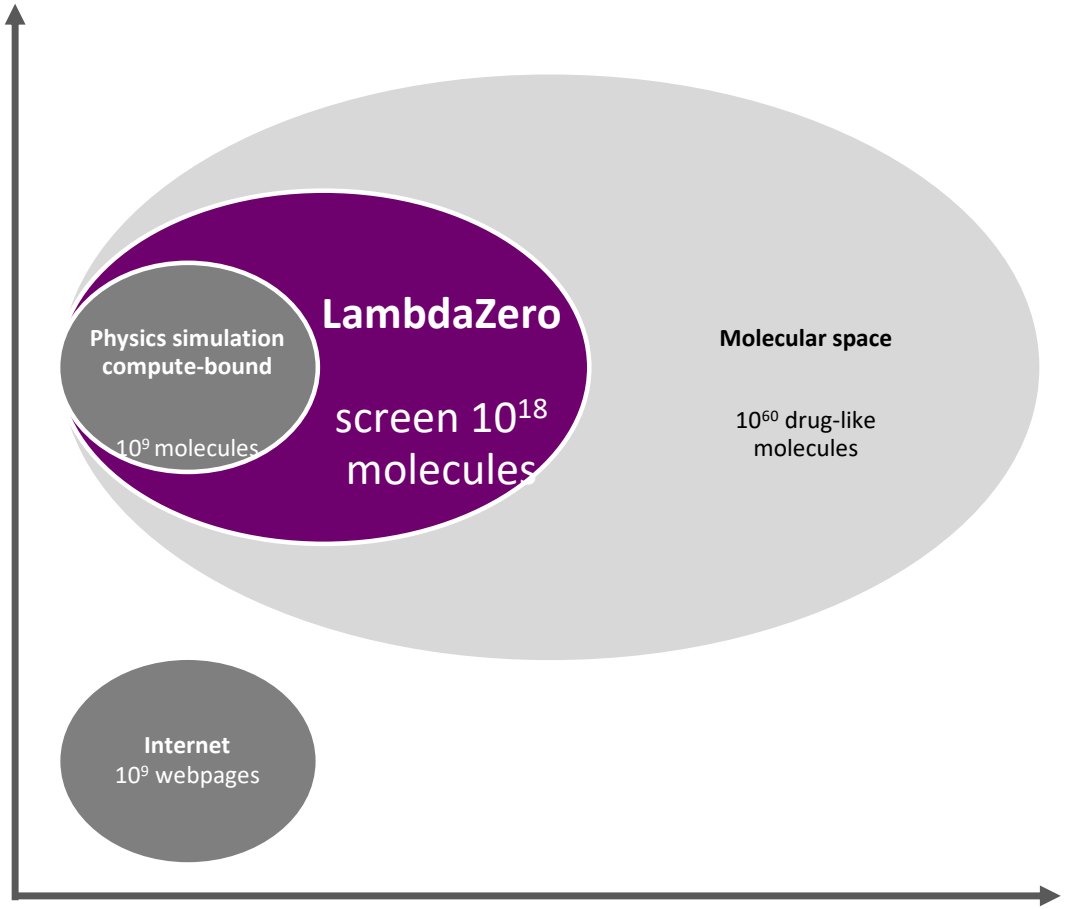
https://www.intel.com/content/www/us/en/newsroom/news/second-gen-habana-gaudi2-outperforms-nvidia-a100.html#gs.kjr1ky

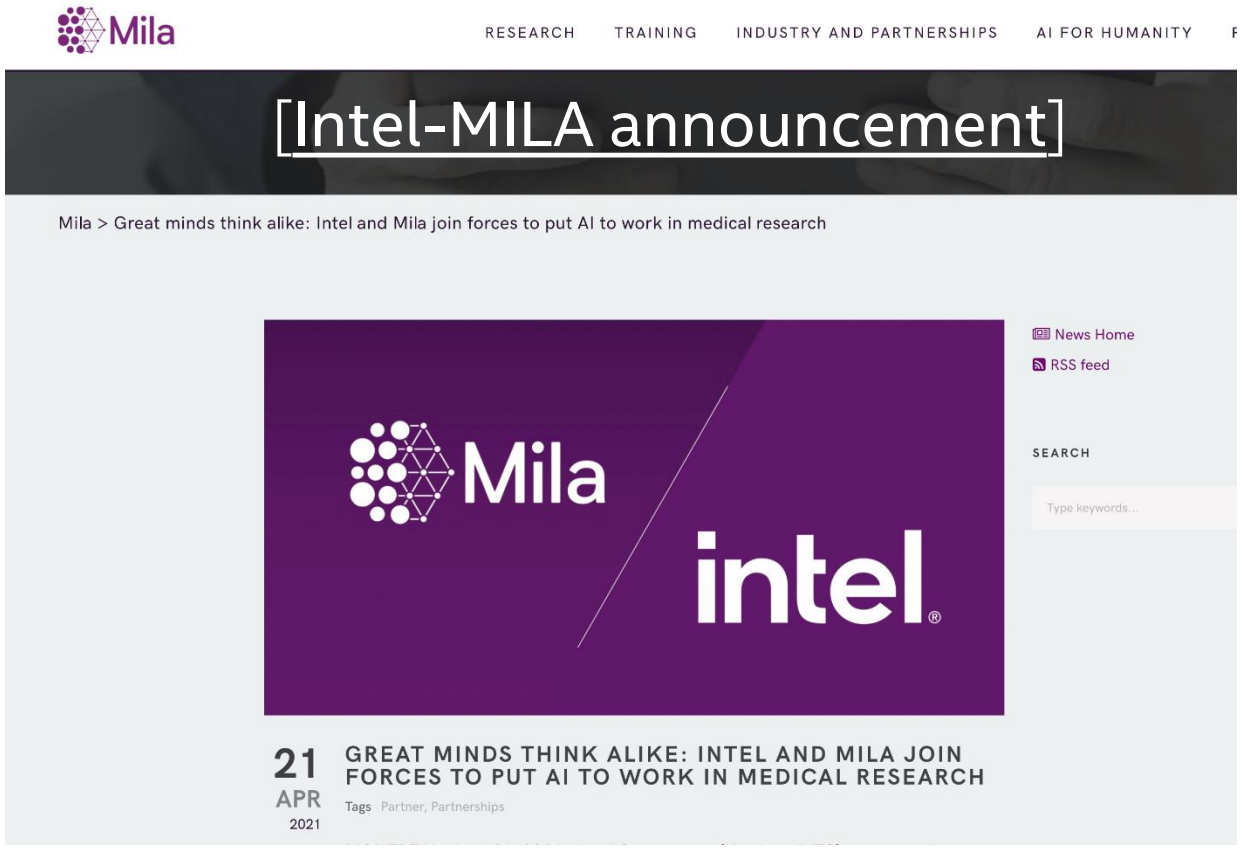
We have the opportunity to deliver several orders of magnitude speedup on current platforms

Necessary but not sufficient for leadership

How do we take next leap to Zettascale?

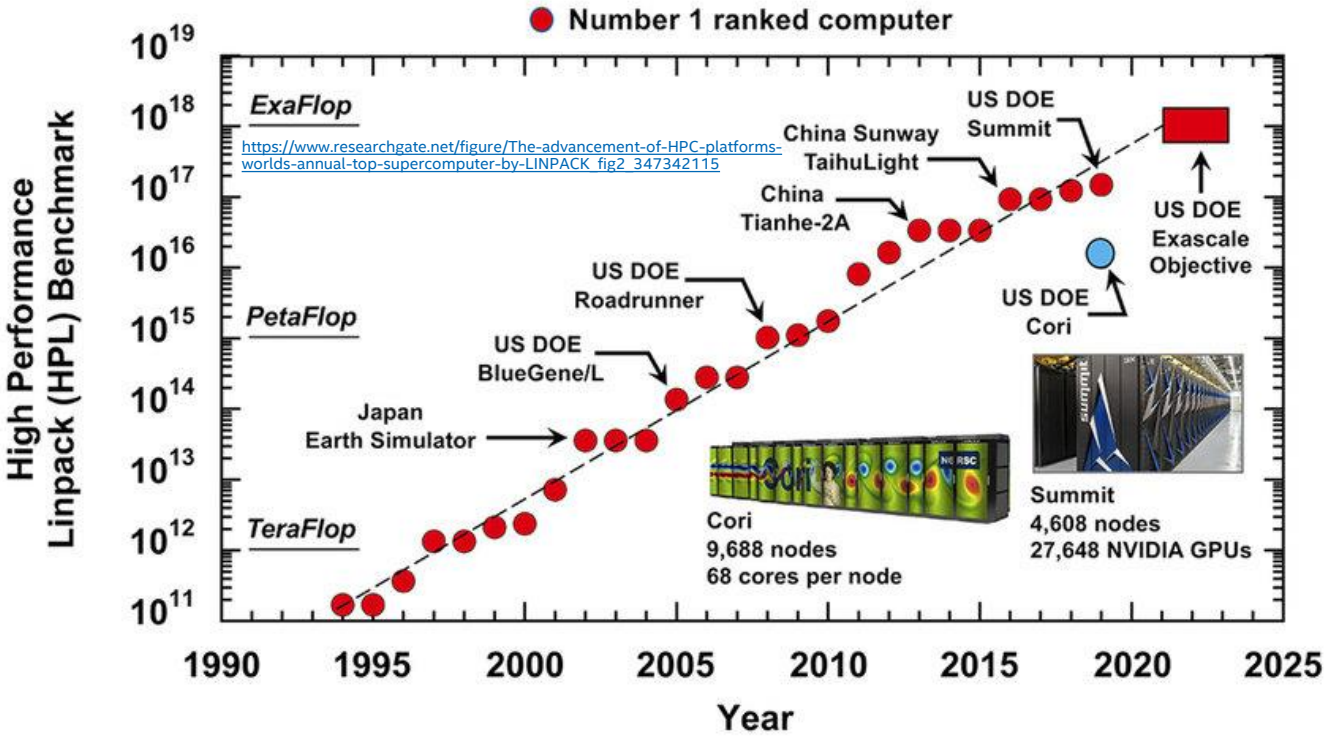
NOVEL AI METHODS FOR BREAKTHROUGH APPLICATIONS: LAMBDAZERO





Collaboration Objective: Make it computationally feasible to expand the search space of better molecules (with more negative binding energy) to 10^18 using machine learning and hardware-software codesign *

ACADEMIA CAN PLAY A SIGNIFICANT ROLE IN POLICY INFLUENCE



Highly cited scientific papers country rankings

1998-2000 (average) No. of papers		2008-2010 (average) No. of papers		2018-2020 (average) No. of papers	
1. U.S.	30,710	1. U.S.	36,910	1. China	46,352
2. Britain	6,071	2. China	9,011	2. U.S.	36,680
3. Germany	4,991	3. Britain	7,420	3. Britain	8,772
4. Japan	4,369	4. Germany	6,477	4. Germany	7,246
5. France	3,609	5. France	4,568	5. Italy	6,073
6. Canada	2,842	6. Japan	4,369	6. Australia	5,099
7. Italy	2,128	7. Canada	4,078	7. India	4,926
8. Netherlands	1,814	8. Italy	3,450	8. Canada	4,509
9. Australia	1,687	9. Australia	2,941	9. France	4,231
10. Spain	1,398	10. Spain	2,903	10. Spain	3,845
⋮				11. South Korea	3,798
13. China	1,217			12. Japan	3,780

Source: Education ministry
Note: Number of papers in top 10% of most-cited papers in each research field

At 0.7% of GDP, India's R&D expenditure in science is less than BRIC nations

https://www.businesstoday.in/latest/world/story/india-gdp-randd-expenditure-in-science-is-less-than-bric-nations-239968-2019-11-22

Table 2: Patent applications and grants in India, China and US

Year	China		United States of America		India	
	Filing	Grants	Filing	Grants	Filing	Grants
2016	13,38,503	4,04,208	6,05,571	3,03,049	45,444	9,847
2017	13,81,594	4,20,144	6,06,956	3,19,829	47,854	13,045
2018	15,42,002	4,32,147	5,97,141	3,07,759	50,659	15,283
2019	14,00,661	4,52,804	6,21,453	3,54,430	56,284	24,936
2020	14,97,159	5,30,127	5,97,172	3,51,993	56,771	26,361
2021	-	-	-	-	66,440	30,074

Source: World Intellectual Property organization (WIPO) and Office of the Controller General of Patents, Designs & Trade Marks (CGPDTM) for India

INVEST IN OUR FUTURE

1. Apply AI in the fields of Digital Biology, Chemical Sciences, Simulations & Population Scale AI Platforms

2. Create Global Research Hub to bring together research/academic institutes and industry + Inspire world class talent to stay in India to invent the future here
   - Science, Computer Science, SW Engineering, Systems
   - Collaboration with IISc and IIIT-Hyderabad at INAI center

3. Build curriculum, skillset and infrastructure in graduate programs for delivering end-to-end performance

4. Partner with industry on technology research (US, EU effectively do that)

We are open for partnership and collaborations

karthikeyan.vaidyanathan@intel.com
bharat.kaul@intel.com

Thank you