HPC and AI Convergence in a Homogeneous Exascale Cluster

Rob Reiner

Director of Product Marketing

Tachyum

HPC vs. AI – Processor and Memory

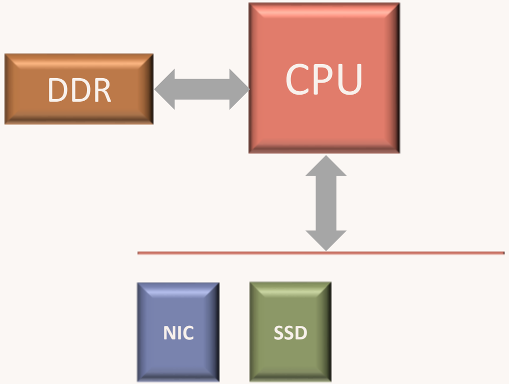
	HPC	AI
High Performance Parallel Processing	Very Important	
FP Precision	High Precision	Low Precision
Vector Processing vs. Matrix Processing	HPC typically uses vectors	Deep learning typically uses matrices
Memory Bandwidth	Very Important	
Memory Latency	Important to the extent it affects effective bandwidth	
Scalable Processor and Memory	Very Important	
Cost and Power Efficient	Very Important	

HPC vs. AI – Storage and Networking

	HPC	AI
Storage Type	Distributed clusters	
Storage Characteristics	Parallel file system	
	Object, Block, File	Object – Scalability for data and metadata. Optimal for small files
	Mostly large files	Small files with a lot of metadata
	Write intensive	Read intensive
	Mostly sequential access	Mixed sequential and random
Networking Characteristics	Fast efficient access to storage and between compute nodes	
	Infiniband or Ethernet with RoCE for hardware interconnect	
	MPI for software interconnect	

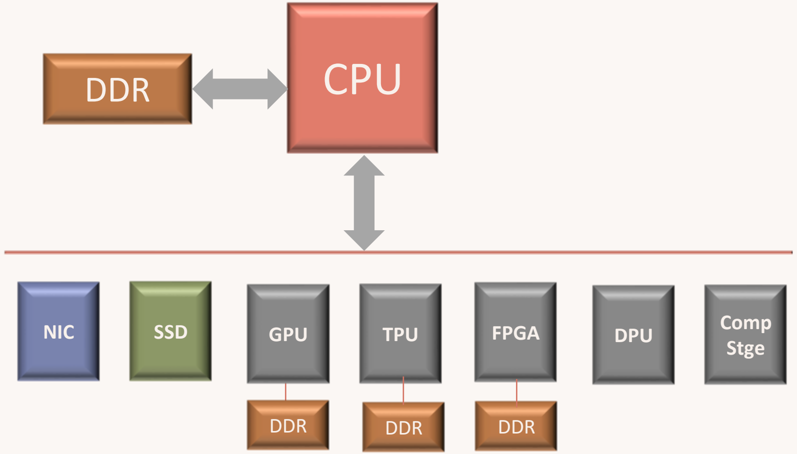
Traditional Homogeneous vs. Heterogeneous Architectures

Homogeneous



Pros	Cons
• General purpose, flexible • Easy deployment/ maintenance	• Not designed for HPC or AI • Low parallel performance for modern workloads

Heterogeneous

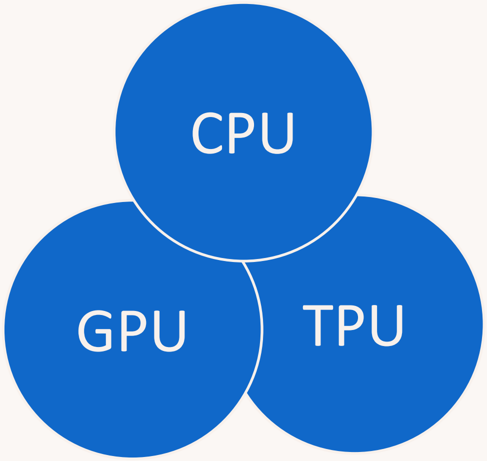


Pros	Cons
• Accelerates specific workloads, including HPC and AI • Scalable	• Needs special programming • Expensive, power-hungry • Under-utilized – contrary to software-defined data center

Tachyum Prodigy – World's First Universal Processor

- Prodigy incorporates the functionality of CPU, GPU, and TPU into a single device
- Prodigy is faster than x86/GPU/TPU
  - Faster, 10x less power, 1/3 cost of x86
  - Faster than highest performing GPUs in HPC and AI
- 128 64-bit cores in a Single Device
  - High-performance across a range of workloads in a homogeneous compute environment
- Humanity: 1st human brain sized AI
  - Not only Focus on Deep Learning AI
  - Also Explainable, Bio, Spiking and General AI

Prodigy Integrates the Best of

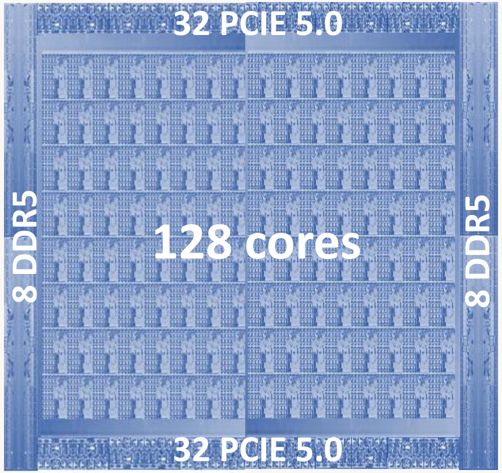


Long Vectors

Matrix Operations

Tachyum Prodigy - Advantages of Homogeneous and Heterogeneous Architectures without the Disadvantages

- High Performance Processor Subsystem
  - Up to 128 general purpose 64-bit cores
- High Floating-Point Performance for Parallel Workloads
  - Range of precision from FP64 to AI data types
  - Performance greater than 2 x 512-bit vector units
- Matrix Operations Accelerate Deep Learning
- Scalable
  - Family of 32 – 128 core devices with support for 2P and 4P Platforms
- High Memory and PCIe Bandwidth
  - 16 DDR5 controllers provide leading-edge bandwidth
  - 64 x PCIe 5.0
- Runs binaries for x86, Arm, and RISC-V
- 5nm Process Technology
- Common Software – Easy Deployment and Maintenance
- No need for costly and power-hungry accelerators



Samples Q3 2022

Prodigy Software Ecosystem

Application Software

- The below applications have been recompiled to run natively on Prodigy.

AI Software, AI Models, Scientific Libraries

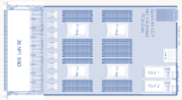
- In addition to TensorFlow, TensorFlowLite, and Pytorch, the following AI models also run natively on Prodigy: Resnet, MobileNet, ShuffleNet, YOLO, SSD, MaskRCNN, BERT, Neural ODE, Graph Neural ODE, and Neural PDE. Scientific libraries supporting Prodigy include Eigen, FFT, BLAS, ODE/PDE, and LAPACK.

OS, Compilers, Debuggers

- Linux has been ported to Prodigy along with the GNU libraries, and Prodigy is compiled using GCC.

Device Emulation

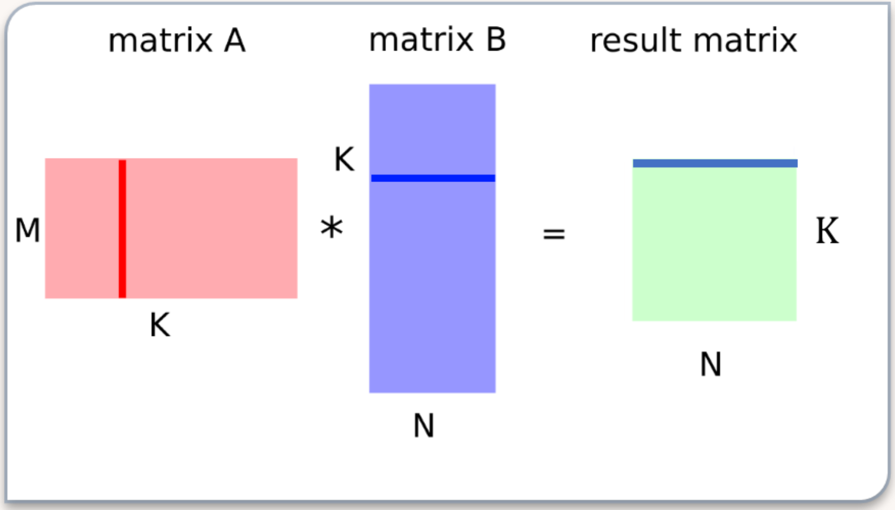
- Chip emulation is supported by software emulators QEMU and gdbsim, as well as Tachyum's Prodigy FPGA Emulator



SW Ecosystem Roadmap

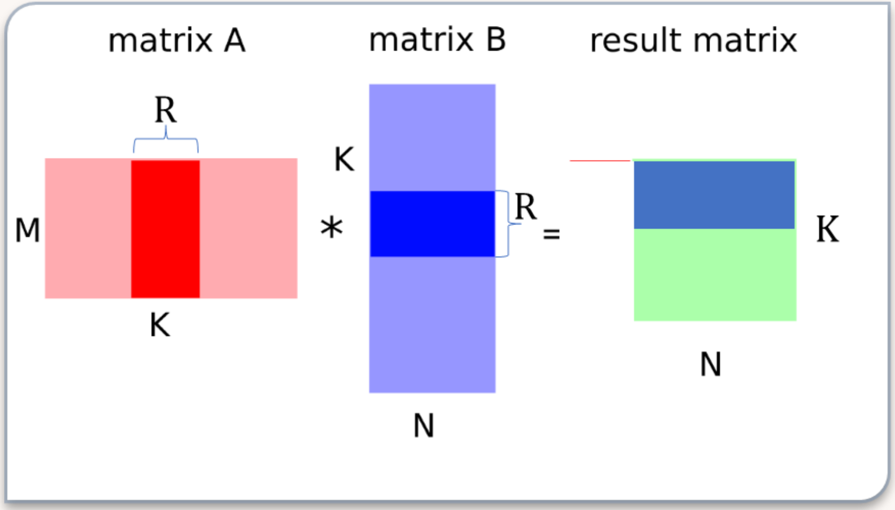
Matrix Multiplication – Matrix vs. Vector Instructions
Functional Comparison

Vector Instructions



In one step N MACs operations are performed

Matrix Instructions



In one step $N \ x \ R$ MACs operations are performed

Matrix Multiplication – Matrix vs. Vector Instructions

Instruction Comparison

Vector Instructions

```
for (int k = 0; k < Kc; k++)
{
    for (int j = 0; j < Mr; j++)                    //rows in matrix A
        for (int i = 0; i < Nr; i+= Nk)             //cols in Matrix B
            for (int ii = 0; ii < Nk; ii++)         //cols in Matrix B
                y[j*Mr + i + ii]+= a[j]*b[i + ii];

    a+= Mr; //next row
    b+= Nr; //next row
}
```

Total $K \times M$ vector instructions calls

- Note, loop for ii is performed in vector instruction

Matrix Instructions

```
//call float MMADD
__asm__ volatile (
"vld w16, p0, [%0]\n\t"
"vld w8, p0, [%1]\n\t"
"vld w0, p0, [%2]\n\t"
"mmaddq f0, p0, f8, f16, f0\n\t"

"vst [%2], p0, w0\n\t"
:
: "r" (b),
  "r" (a),
  "r" (y)
: "v0", "v8", "v16"
);
```

No loops required in code – just call MMADD

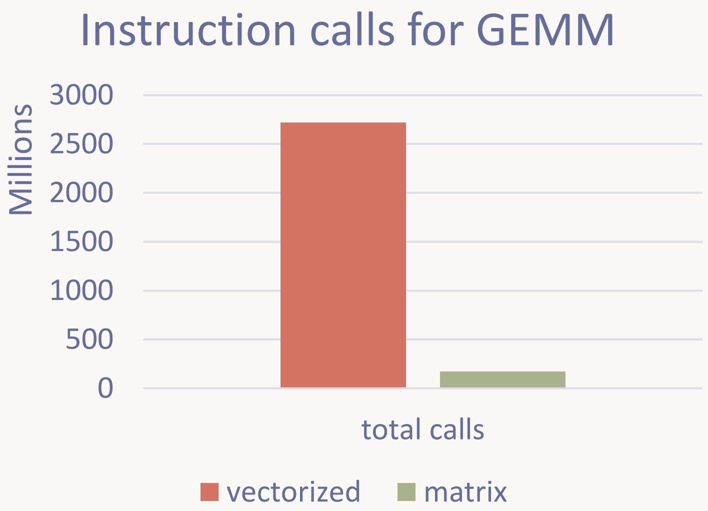
GEMM Computation Reduction using Matrix Instructions

MAC instructions calls for vectorized version:

- fadd4 1,439,257,600 calls
- fmul4 1,280,659,456 calls
- **Total 2,719,917,056 calls**

MMADD instructions calls for matrix version:

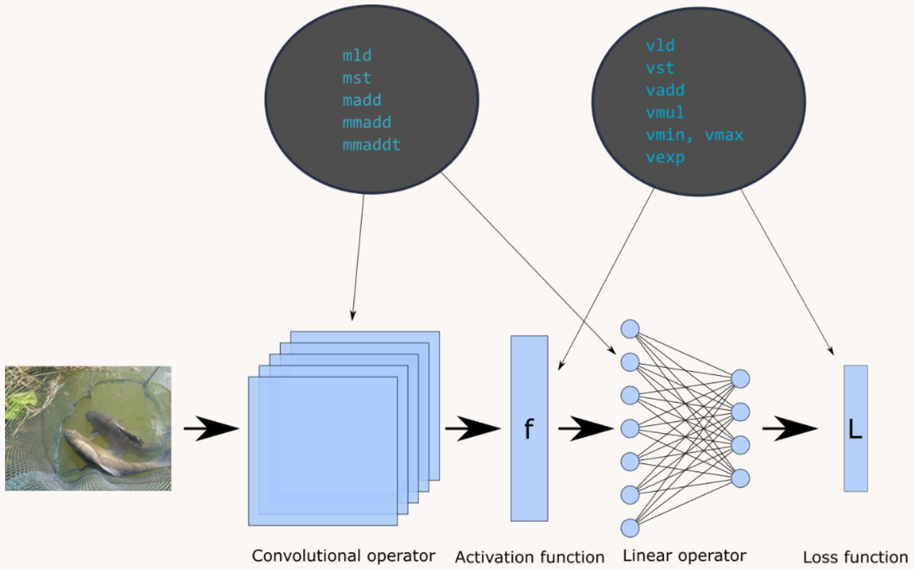
- fadd4 164,189,184 calls
- mmadd 9,961,472 calls
- **Total 174,150,656 calls**



15x Fewer Calls with Matrix Instructions

Prodigy Matrix and Vector Instructions for Deep Learning

- Prodigy Matrix Instructions Utilized for Convolutional and Linear Operators
- Prodigy Vector Instructions Utilized for Activation and Loss functions



Sparsity and Quantization

- Sparsity
  - Sparse matrices include many zeros or values that will not significantly impact a calculation.
  - Prodigy supports sparse matrix operations optimized for compressed networks/ models, thus reducing memory and computation requirements
  - Prodigy incorporates specific instructions for efficient storing and loading sparse matrices and for sparse structured matrix multiplication
- Quantization
  - Quantization reduces the precision of the weights in the neural network while maintaining the required accuracy
  - Prodigy support for quantization includes:
    - Support for low precision data types
    - Support for quantization-aware training, mixed-precision training, post-training quantization inference

Prodigy Delivers Key Requirements for HPC and AI

	HPC	AI/ML
High Performance Parallel Processing	✓	✓
Range of Floating-Point Precision	✓	✓
High Performance Vector and Matrix Operations	✓	✓
Lower Precision and Sparse Data Types		✓
Hardware Acceleration for Sparse Operations		✓
Scalable, including large memory footprint	✓	✓
High Memory Bandwidth	✓	✓
High Performance I/O subsystem	✓	✓
Easy Deployment and Maintenance	✓	✓
Cost and Power Efficient	✓	✓
Simple Programming Model	✓	✓

Next Generation Exascale Supercomputer

SLOVAK ACADEMY OF SCIENCES

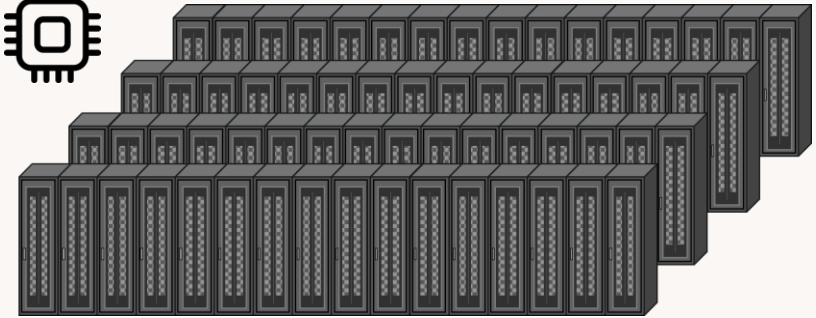
NSCC Slovakia Supercomputer

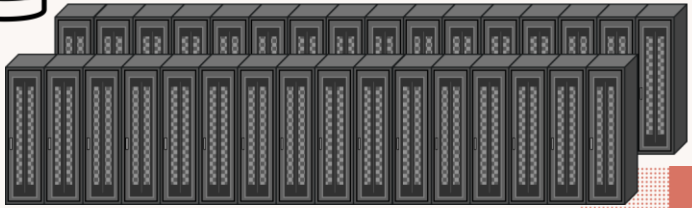
64 Compute Racks

64 AI ExaFLOPs
>500 DP PetaFLOPs

16 – 32 Storage Racks

100 – 200 PB



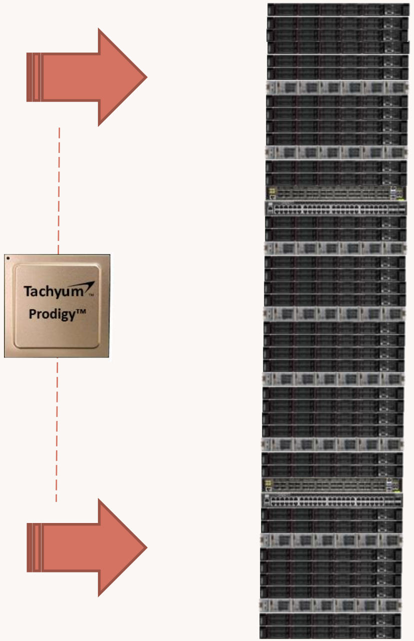


Installation by End of 2022

NSCC-SC Compute Rack

- High – Performance
  - 1 AI ExaFLOPs of Training and Inferencing per rack
- Prodigy T16128 Universal Processor
  - 128 64-bit cores
- Rack Configuration
  - 32 Prodigy 1U compute nodes
  - 8 sockets per compute node
  - 256 sockets per rack
- Power and Cooling
  - Busbar-based power distribution for servers
  - Liquid cooling for processors and DIMMs

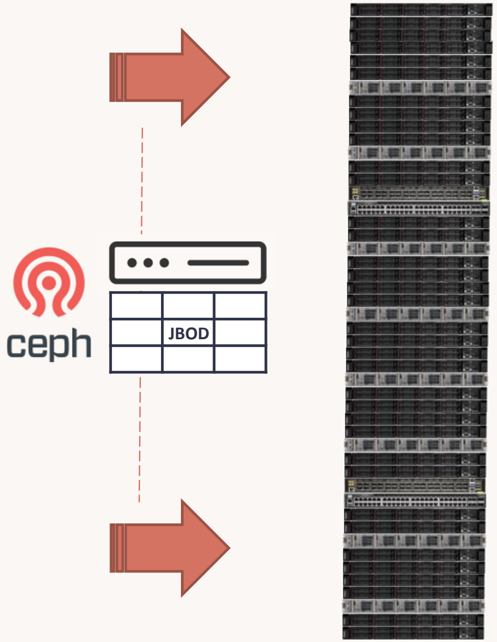
1 AI ExaFLOP Compute Rack



NSCC-SC Storage Rack

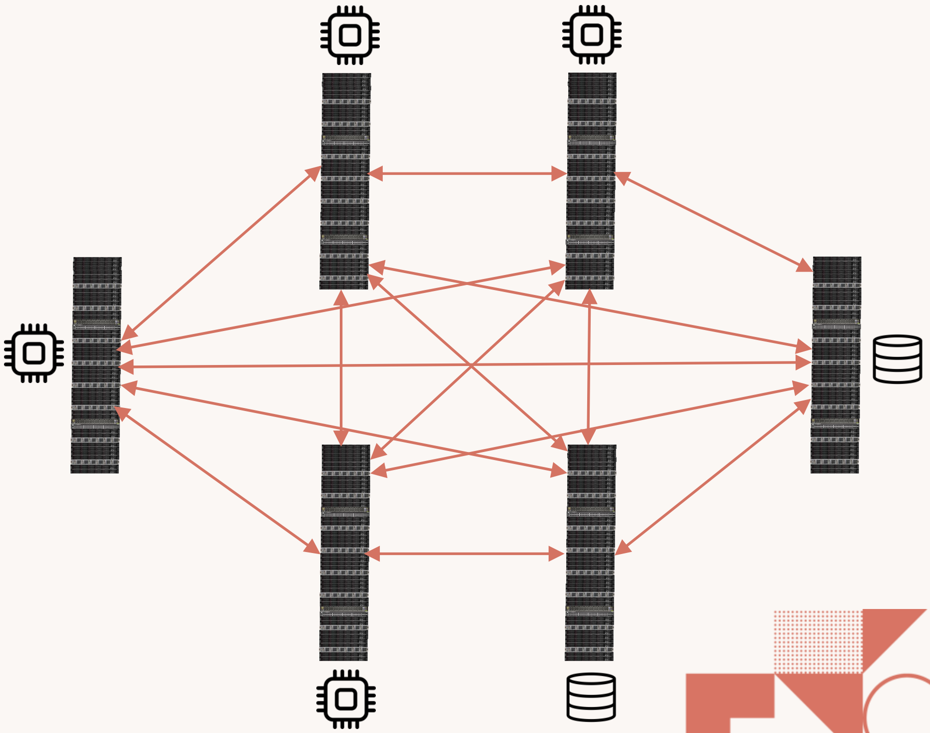
- Ceph-based storage rack
  - Support for object, block, and file storage
  - High reliability
- Six storage building blocks per rack, each with:
  - 1 server + 1 JBOD
  - 1.1 petabyte per block
  - 4 x 100 Gb/s Ethernet
- 6.6 petabyte per rack of usable storage

6.6 PB Storage Rack



NSCC-SC Networking Architecture

- Peer-to-Peer data network connecting all compute and storage racks
  - Minimizes latency and cost
  - Maximizes efficiency
- 400 Gb/s switches with 100 Gb/s breakout cables to NICs, compute, and storage racks
- Management network connects all compute, storage, power, and cooling nodes



NSCC-SC System Architecture Provides Infrastructure for HPC and AI/ML

	HPC	AI/ML
Fast Efficient Access to Storage and other Compute Nodes	✓	✓
Parallel File System for Distributed Clusters	✓	✓
Support for Object, Block, and File Storage	✓	✓
Scalability for Small Files and Large Amounts of Metadata		✓
High Performance for Write Intensive Workloads	✓	
High Performance for Read Intensive Workloads		✓

SC21

St. Louis, MO | science & beyond.

Thank You and Stay Tuned!

Please visit Tachyum at Booth #906

Tachyum™