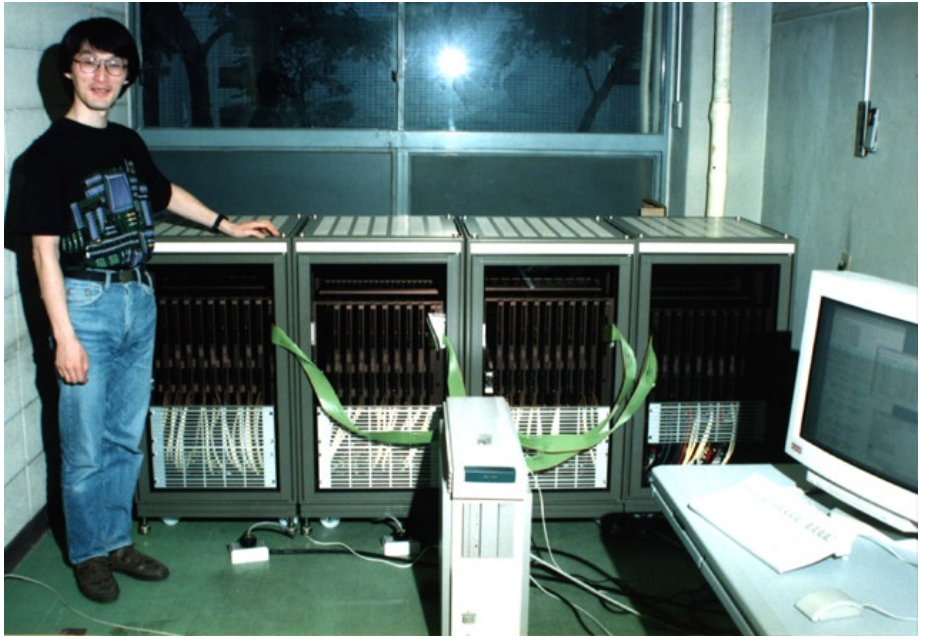
Accelerator for Deep learning and HPC

Jun Makino
Kobe University/Preferred Networks

SimonFest May 21 2025

Simon with GRAPE-6 (2002?)



J.M. with GRAPE-4 (1995)

Talk Structure

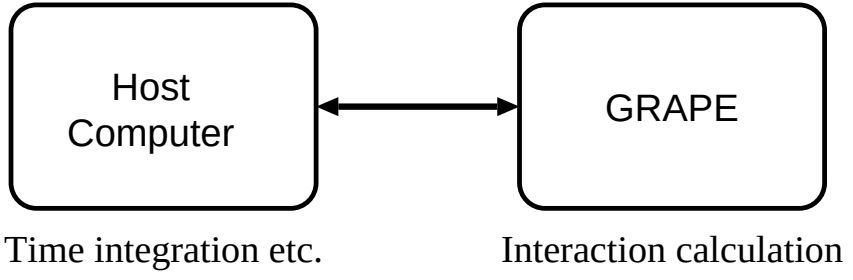
- GRAPE
- The life after GRAPE
- MN-Core
- What will come after GPUs?
- How we do simulations on post-GPU processors?
- Summary

Brief history of GRAPE(-DR)

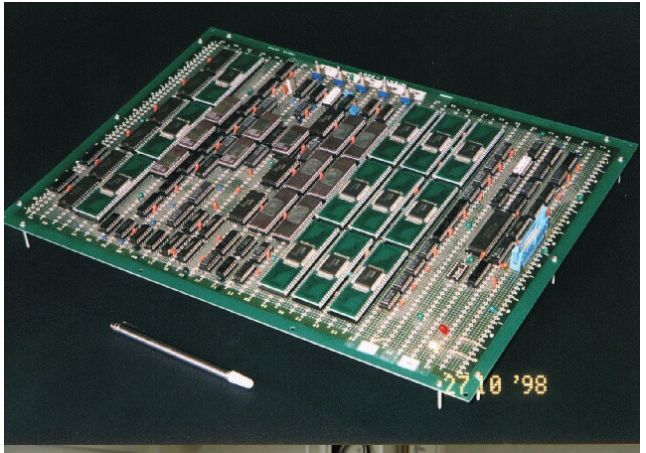
- Basic concept
- GRAPE-1 through 6
- GRAPE-DR

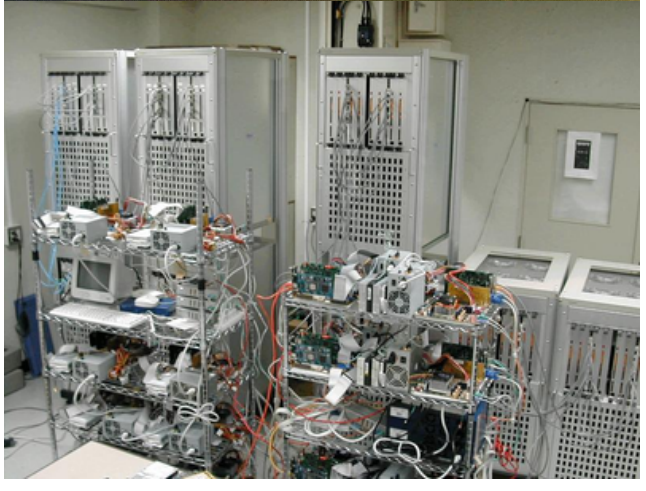
Basic concept (As of 1988)

- **With astrophysical N-body simulation, almost all calculation goes to the calculation of particle-particle interaction.**
- **This is true even for fast $O(N \log N)$ or $O(N)$ schemes**
- **A pipelined hardware which calculates the particle-particle interaction can accelerate overall calculation.**
- **Original Idea: Chikada (1988)**



GRAPE-1 to GRAPE-6



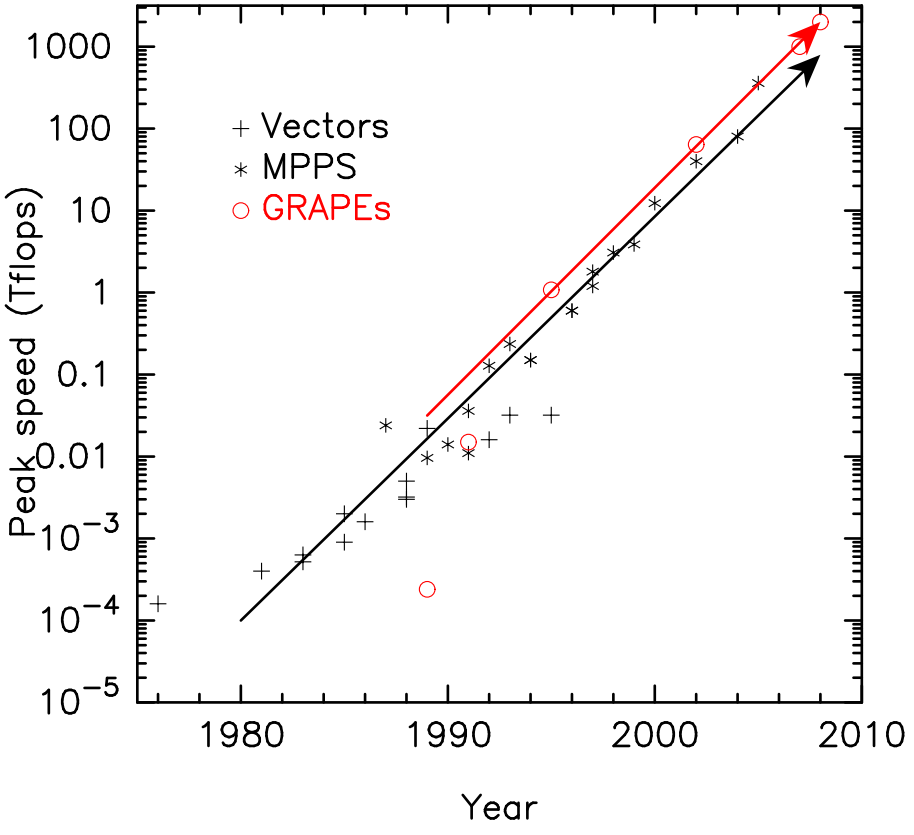


GRAPE-1: 1989, 308Mflops

GRAPE-4: 1995, 1.08Tflops

GRAPE-6: 2002, 64Tflops

Performance history (as of 200x)



Since 1995 (GRAPE-4), GRAPE has been faster than general-purpose computers.

Development cost was around 1/100. Performance per Watt was around 100.

Why no GRAPE-8?

Simon’s way to refer to GRAPE-8 (Great Ape)

GRAPE-DR: Why and what?

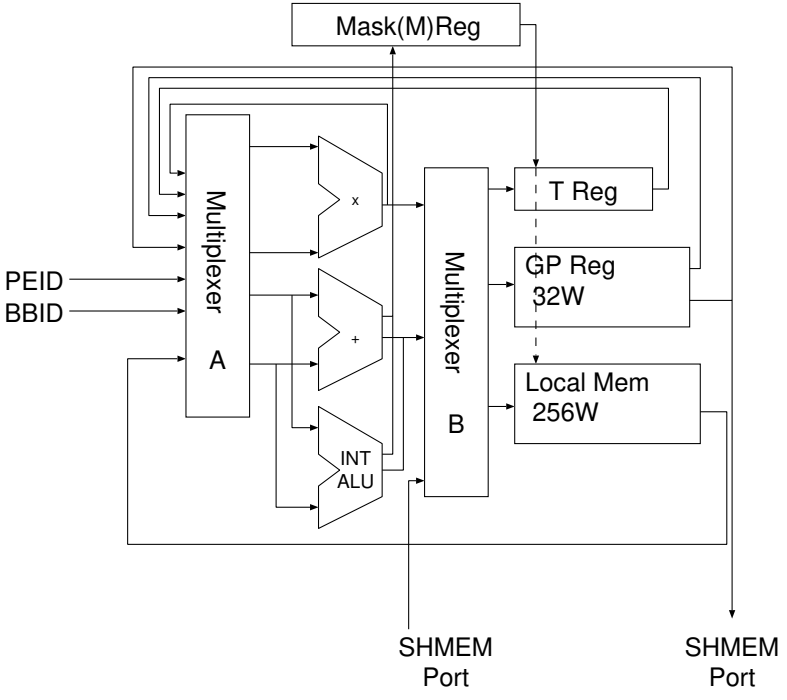
- Chip development cost becomes too high.

Year	Machine	Chip initial cost	process
1992	GRAPE-4	200K$	1μm
1997	GRAPE-6	1M$	250nm
2004	GRAPE-DR	4M$	90nm
2018	MN-Core	> 10 M$	N12
2022	MN-Core 2	~ 20 M$	N7
2024	MN-Core 3	~ 40 M?$	2nm

How we can continue?

Widen application area/user base

Processor architecture



- **Float Mult**
- **Float add/sub**
- **Integer ALU**
- **32-word registers**
- **256-word memory**
- **communication port**
- **512 of them in one chip**

(Rather recently I realized this architecture is very similar to that of FPS AP-120B, which Steve has used for his hybrid N-body+Fokker-Planck code)

Limitation of GRAPE-DR

Not fast enough compared to commercially available chips

Year	Our system	speed	What you could buy	speed
1995	GRAPE-4	640MF	DEC Alpha	150MF
2001	GRAPE-6	30GF	Intel P4	1.4GF
2008	GRAPE-DR	250GF	NVIDIA C2050	515GF

System cost per chip was very low for GRAPE-4 and 6, but not so low for GRAPE-DR.
For the same process technology, GRAPE-DR would have been better than GPU, but this was not quite enough.

In the meantime...

BRIDGE (Fujii+2006): Let two systems interact with fixed time intervals.

P^3T scheme (Oshino+2011): Parcile-Particle, Particle-Tree
— We finally combined individual timestep, Barnes-Hut tree, and large scale parallization (*cf.* McMillan and Aarseth 1993)

FDPS: Framework for Developping Particle Simulators (Iwasawa+2016)
https://github.com/FDPS/FDPS
— Make it possible to write paralell treecode for any particle-particle interaction

Resulted in: GPLUM (Ishigaki+2020) for planet formation, PeTar (Wang+2020) for star clusters, ASURA/FDPS (202X...) for galaxies.

AI-oriented processors

- One layer of NN is matrix-vector product (with O(n) other operations)
- In the case of CNN, matrix-matrix product.
- Very low precision numbers are used. FP16, FP8 and in some cases FP4.
- With Transformers (the core of GPT-x), matrix-matrix multiplication (but matrix width limited by the batch size) is the main operation.
- Special-purpose architectures (with matrix-vector calculation pipeline) might be possible.

AI-oriented processors

- **HotChips 24: 14 out of 24 oral presentations are on AI processors.**
- **Tenstorrent Blackhole, SK Hynix PiM, Blackwell, Sambanova SN40L, Intel Gaudi 3, AMD MI300X, FuriosaAI RNGD, AMD(Xilinx) Versal, Onyx (Stanford), Meta MTIA, Tesla Dojo, Cerebras CS-2, MS MAIA, PFN MN-Core 2**
- **Blackhole, SN40L, Onyx, MTIA, Dojo, Cerebras, MAIA: 2D on-chip network + MIMD cores with matrix multiplication units.**
- **Gaudi, RNGD: proprietary arch with very large matrix multiplication units.**
- **SK Hynix : custom GDDRx memory with FPUs.**
- **Versal : FPGA, remaining: Nvidia, AMD, PFN**

Characteristics of architectures

- **No hierarchical cache (except for GPUs from NVIDIA and AMD)**
- **Almost all AI-oriented processors have largely similar architectures. MIMD, 2D-mesh network with HBMs at the edge of the network.**
- **Writing application programs for these processors is ... not easy (Note that we only need matmul for AI).**
- **No FP64 support. Most processors lack FP32. Even with GPUs from NVIDIA and AMD, relative FP64 performance started to decrease.**

MN-Core

- AI-oriented processor developped by Preferred Networks (PFN) and JM.
- (JM moved to the cross-appointment position between Kobe U and PFN as of Nov 2023)
- At the time of completion, achieved highest FP16 performance per board and highest performance per watt number.
- Development started in 2016. First gen completed in 2020.
- FP16 Peak 524TF
- Power consumption less than 500W, 1.2TF/W
- 2.5x higher performance per watt compared to NVIDIA V100

(Still not quite good enough... I want to have 10x or more)

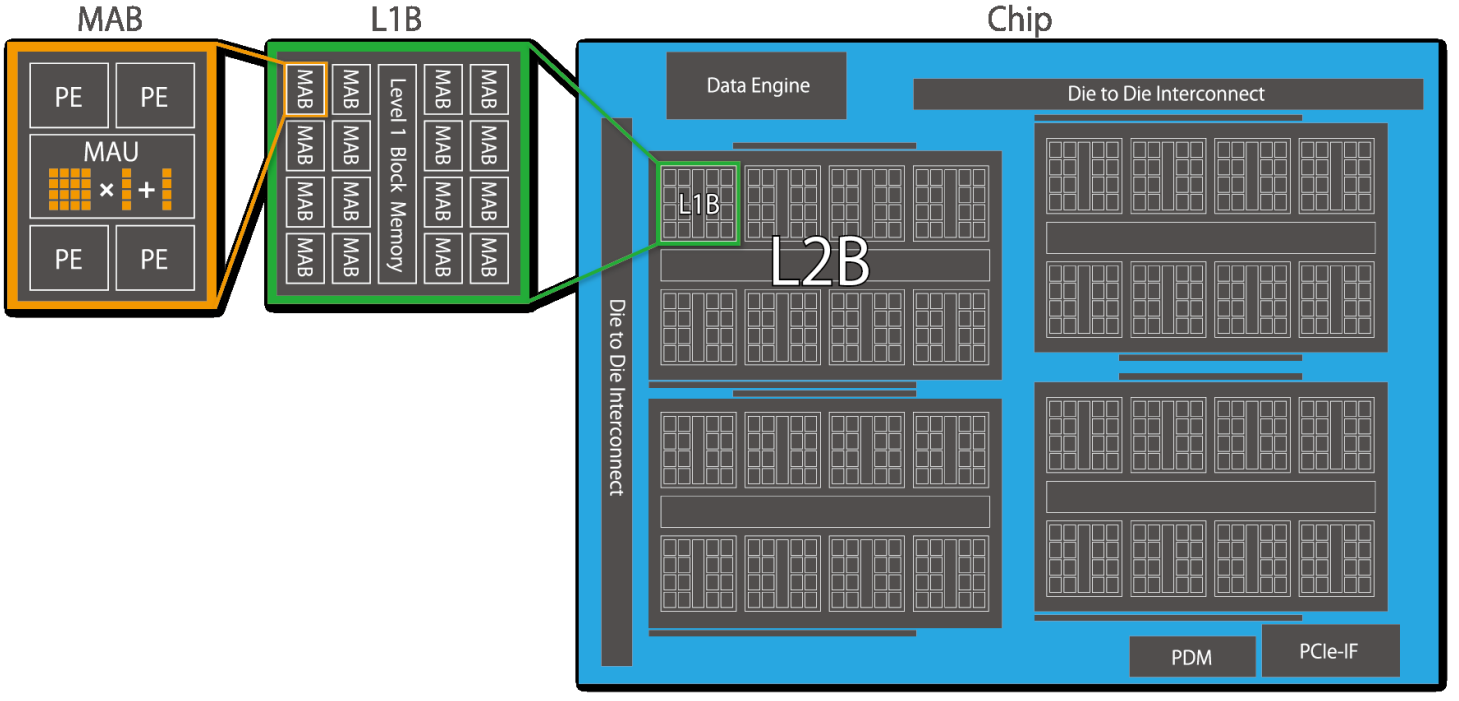
Why first-gen MN-Core was not good enough

- At the time of the design, it would be more than 10x better than NVIDIA P100 or its successor with a similar architecture.
- NVIDIA V100 adopted matrix-matrix multiplication unit (= NVIDIA also adopted a specialized architecture)
- MN-Core advantage was reduced by a factor of four...

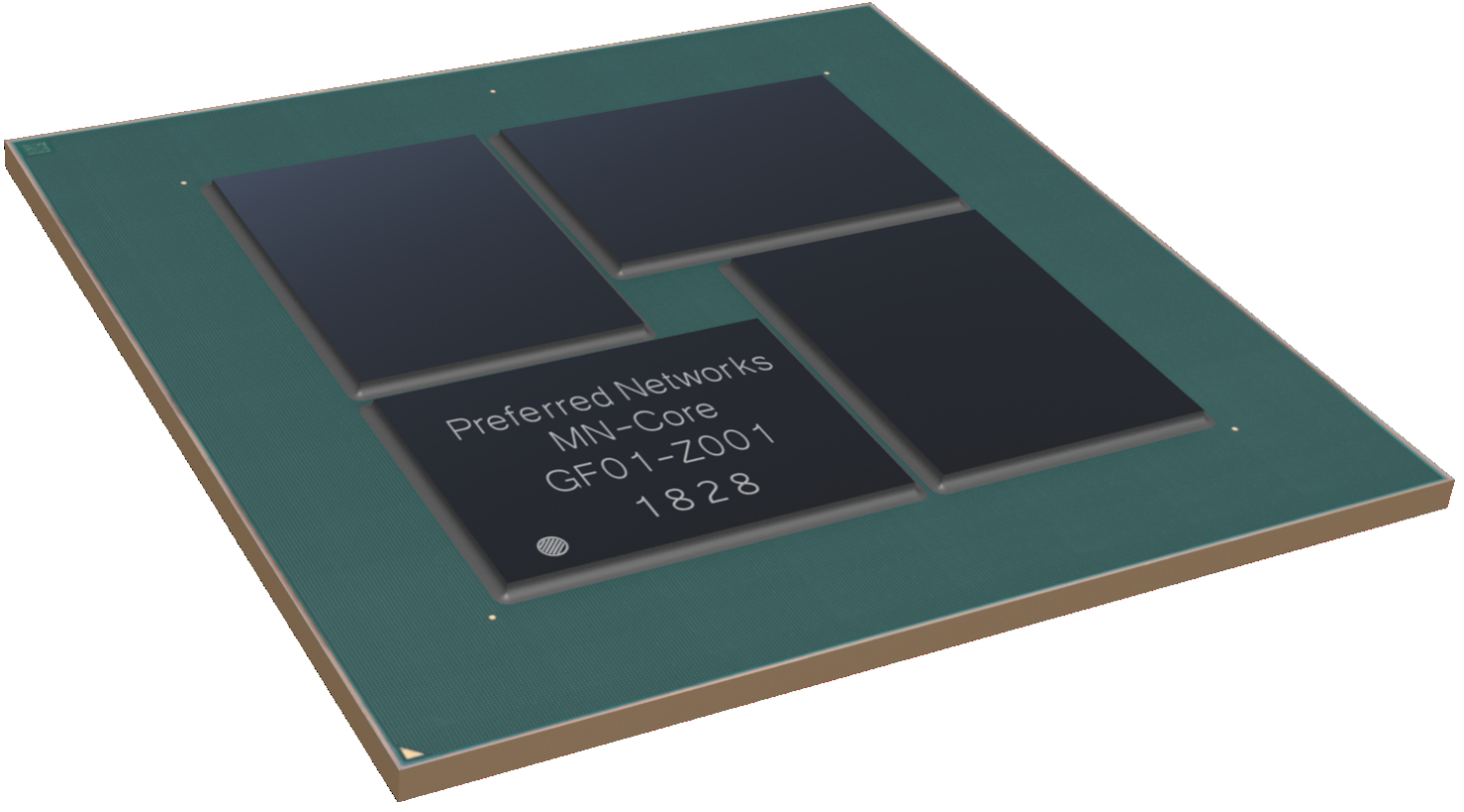
MN-Core overview

- 1 card : 1 module
- 1module : 4-die MCM
- 1 die: PCIe (gen4, x16), LPDDR4 memory, 4 “Level-2 broadcast blocks” (L2Bs)
- 1 L2B: 8 L1Bs
- 1 L1B: 16 MABs (Matrix Arithmetic Blocks)
- 1 MAB: 4 Processor Elements and one Matrix arithmetic unit
- FP64:FP32:FP16 performance ratio is 1:4:16
- Entire module operates as one huge SIMD processor with single instruction stream.

MN-Core Structure



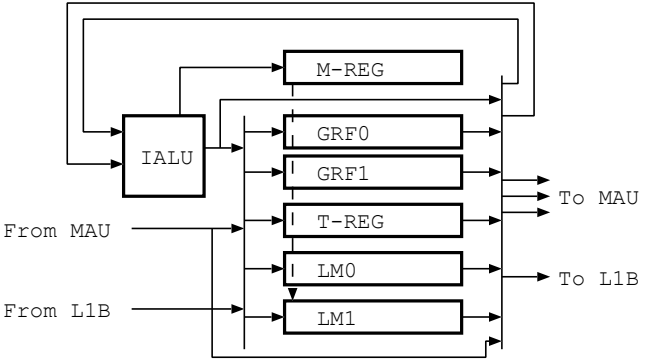
MN-Core



Details

- **PE(Processing element)**
- **L1B(Level 1 broadcast block)**
- **L2B(Level 2 broadcast block)**

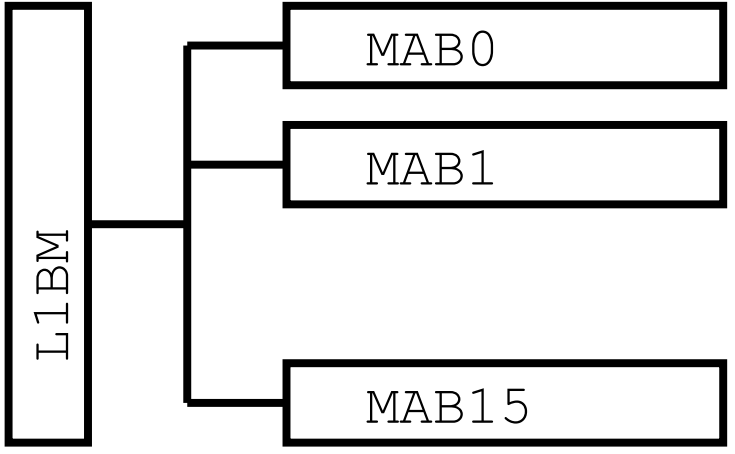
PE(Processing element)



- IALU and MAU as arithmetic units
- MAU performs FP64, FP32 and FP16 matrix-vector product

- GRF are 1R1W 2-port memories 。LMx are single-port
- T-reg: additional register, 1R1W 4 words (vector length)
- LMx (local memory): 2048 64bit-words x 2, GRF: 512 words
- all instructions are length 4 vector instructions.

L1B(Level 1 broadcast block)



- **16 MABs are connected to one L1BM(level 1 broadcast memory)**
- **Data read from L1BM are broad-casted to all PE (or MAB).**

- **Data read parallel from all PEs/MABs can be summed up and stored to L1BM with full speed.**
- **No direct connection between PEs.**

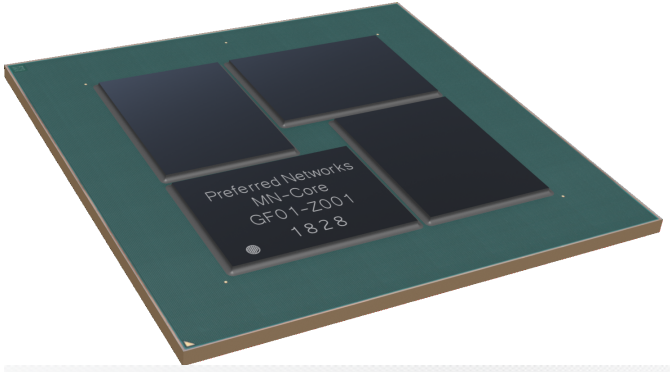
L1B characteristics

- **Using explicit broadcast/reduction, fine-grained parallel operations can be performed with very low overhead.**
- **In particular, reduction operations over multiple PEs. which are very slow on GPUs, can be done very quickly.**
- **As a result, large number of PEs can be used to parallelize relatively small matrix product. This feature is actually very important for inference performance of both CNN and LLM.**
- **L2B and top-level structures are largely the same as that of L1B.**

Difference from usual architecture

- **With typical present CPUs, from the machine code one need to restore the parallelism in the innermost loop by means of register renaming and OoO execution. This is because the number of architecture registers is too small to fill the pipeline.**
- **With MN-Core architecture, very large number of registers are all visible and there is no need for register renaming.**
- **By using fixed-length vector instruction, we removed the need of out-of-order execution as well. There is no need of real-time instruction scheduling since the result of one instruction is available to the next instruction.**
- **Large number of visible registers need long instruction words, which is okay because of chip-wide SIMD architecture.**

MN-Core/MN-3 system



PFNのスパコン「MN-3」が世界1位に、消費電力性能ランキングのGreen500で

岡林 凛太郎　日経クロステック／日経コンピュータ

2020.06.23

PR

Preferred Networks（PFN）のスーパーコンピューター「MN-3」が2020年6月22日（欧州時間）、スーパーコンピューターの消費電力性能ランキング「Green500」で世界1位を獲得した。HPC（ハイ・パフォーマンス・コンピューティング）に関する国際会議「ISC 2020 Digital」が同日ランキングを発表した。

PFNのスーパーコンピューター「MN-3」
（出所：PFN）
[画像のクリックで拡大表示]

MN-3はPFNが独自開発した深層学習用プロセッサー「MN-Core」を使ったスーパーコンピューターだ。PFNのスーパーコンピューター「MN-2」の後継機で、2020年5月に運用を始めた。160個のMN-Coreを搭載し、1ワット当たり21.11ギ

MN-Core 2 and next generations

- **MN-Core2 was completed in 2023. Now commercially available.**
- **Performance comparable to MN-Core with 1/5 of die area.**
- **Development of next generations already started.**
  - **Samsung 2nm, should achieve highest performance for training.**
  - **Also started the development of new processor for LLM inference.**

Software for MN-Core 2

- **MNSDK: AI-oriented**
  - **PyTorch — ONNX — actual machine code.**
  - **Existing PyTorch code (should) work with small changes.**
- **HPCSDK: For General-purpose HPC**
  - **Dialects of OpneCL and OpenACC**
  - **OpenACC direct resembles HPF.**

Application performance of MN-Core 2

	MN-Core 2	A100
GCN(PFN internal use)	5.41TF(FP32)	3.17TF
ResNet50 training	77TF(FP16)	33.2TF(BF16)
ResNet50 Inference	154TF(FP16)	33.7TF(BF16)
HIMENO benchmark	9.03TF(FP32)	0.634TF
OpenFDTD	0.655TF(FP32)	0.488TF

- Performance 1.5-5 times higher than that of A100
- Very high performance for finite-difference applications. (OpenFDTD implementation used the temporal blocking and HIMENO benchmark fits to the on-chip SRAM).

Looking back the evolution of computer architecture

- **Until 1976: Scalar computers. The last one: CDC 7600**
- **1976 to 1992: Shared memory parallel vector processors. Cray-1 to C-90.**
- **1993 to 2008: Distributed-memory parallel microprocessors. Cray T3D to Cray XT4.**
- **Since 2008: CPU + GPU (or some other accelerator). IBM Roadrunner**

Roughly in every 15 years big change architecture occurred. The successor of GPU has not appeared yet.

Why the change was necessary?

Basic reason: Existing architecture became unable to make use of the advance in the semiconductor technology

- **advance in the semiconductor technology = increase in the number of transistors**
- **The architecture itself limits the scalability**

Scalar to vector

- **Scalar computer: use the increased number of transistors to make a faster arithmetic unit.**
- **Magnetic core memory: much slower than transistors**
- **CDC 7600 reached the limit: fully pipelined arithmetic unit**

Scalar machines could not make use of

- **Gate count much larger than that for fully pipelined arithmetic unit**
- **Very fast SRAM main memory**

Vector machines could use fast SRAM(and later DRAM) memory and a small number of pipelines

Vector to MPP

- **Advance of vector processors: increase in pipeline per processor and number of processors which share the physical memory.**
- **The number of wires and switches increase faster than the number of pipelines. 64 pipelines seem to be the practical limit.**
- **Need to move to a system made of large number of simple processors, each with small memory, connected with relatively thin network.**
- **Early examples: Caltech Hypercube, Cray T3D etc**

MPP to GPU

- It becomes possible to fit a large number of processors (pipelines) in one chip.
- The same situation as that of vector-parallel processors.
- Many-core processors have hierarchical cache with coherency.
- hardware and power consumption to maintain coherency becomes dominant.
- GPU relaxes/removes coherency and thus push up the limit a bit.

GPU to ???

- **Even without coherency, the data movement between off-chip DRAM and multiple levels of cache memory becomes the bottleneck.**
- **The “obvious” solution is to give up the cache hierarchy completely and make the main memory physically close to processors.**
- **In other words, we need to move to an on-chip distributed-memory processor.**

Some exercise

- **A wire of length 10cm ($10^5 \mu\mathrm{m}$) has 20pF capacitance. For 1V swing, it consumes 10pJ/bit.**
- **Actual power consumption of a DDR5 is around 20pJ/bit. Voltage is around 1.3V.**
- **LPDDR5 and GDDR6x: $\sim$ 10 pJ/bit. Wire length is shorter than that of DDR5 modules. and swing voltage is smaller.**
- **HBMx: 3-4 pJ/bit. Wire length is around 25mm.**

Modern GPUs spend $>$ 50% of total power to move data from HBM to L2D$.

NVIDIA’s roadmap

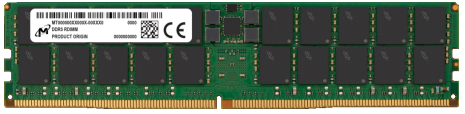
Year	Model	Memory Bandwidth (TB/s)	Power Consumption(W)	Efficiency (pJ/bit)
2020	A100	1.5	400W	33
2022	H100	3	700	29
2025	B300	8	1400?	22
2026	Rubin	13	1600?	15.4
2027	Rubin Ultra	32	3600?	14.1

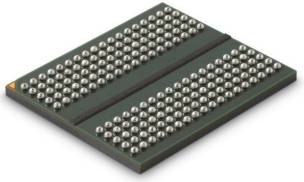
- 20x memory bandwidth in 5 years
- 10x power consumption...
- Improvement of energy efficiency of memory access is rather small.
- Essentially the limit of HBM memory

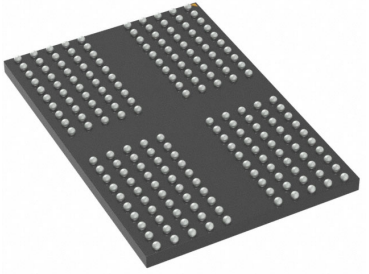
What we need

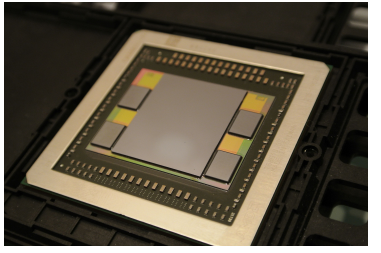
- **We need a processor architecture (or memory structure) which is less expensive and more power-efficient than that of HBMx.**
- **Here we discuss memory architecture**

Memory architecture





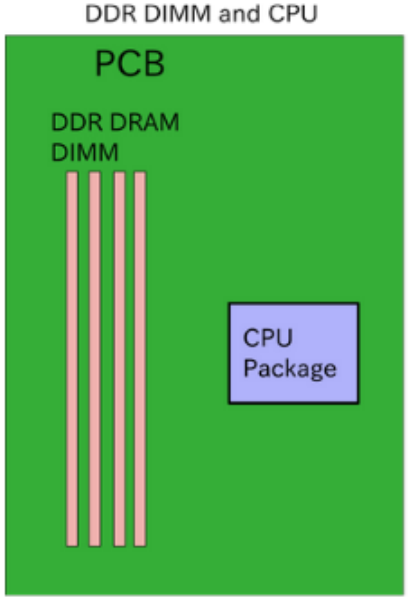


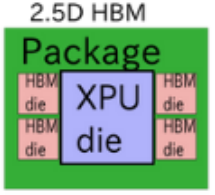


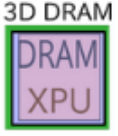
DDR(2000～) GDDR(2003～) LPDDR(2008～) HBM(2015～)

- DRAM cell structure is essentially the same
- difference: Core/Interface voltage, physical layout
- What consumes power is not the DRAM memory cell, but drivers and wires (PCB/on chip patterns).

What we can do and how will it look like.

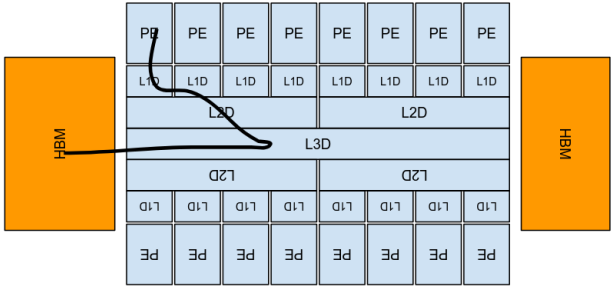


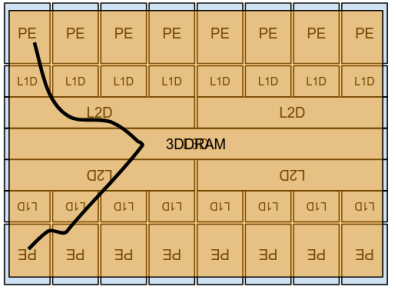


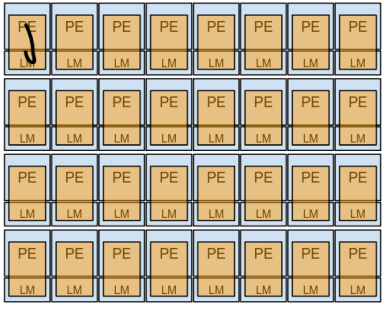


We need to realize “3D memory”, in other words, to put DRAM on top (or bottom) of processor die.
We also need to change the memory hierarchy.

Stacked DRAM with shared and distributed arch



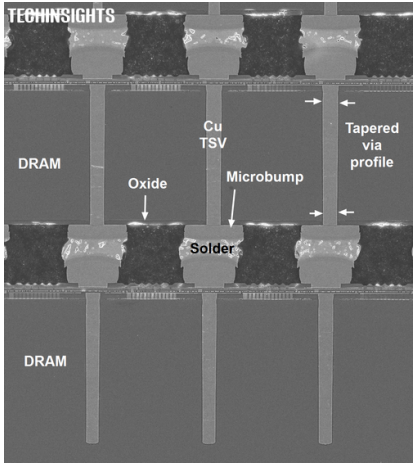




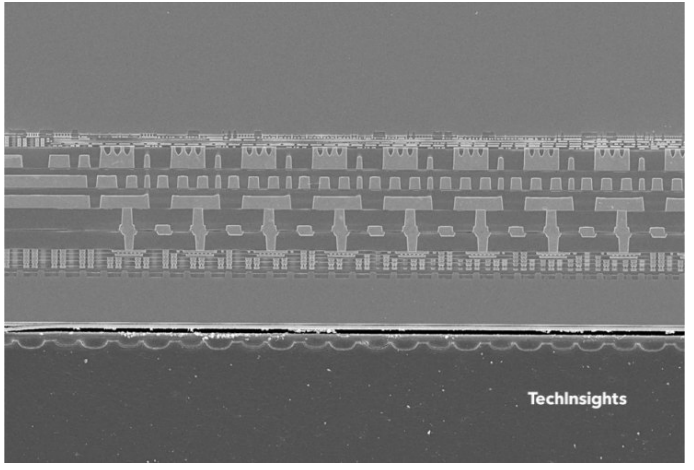
Shared memory: Data move distance not much different from that of HBM

Distributed memory: Data move minimized

Microbumps and Hybrid Bonding



Microbumps



Hybrid Bonding

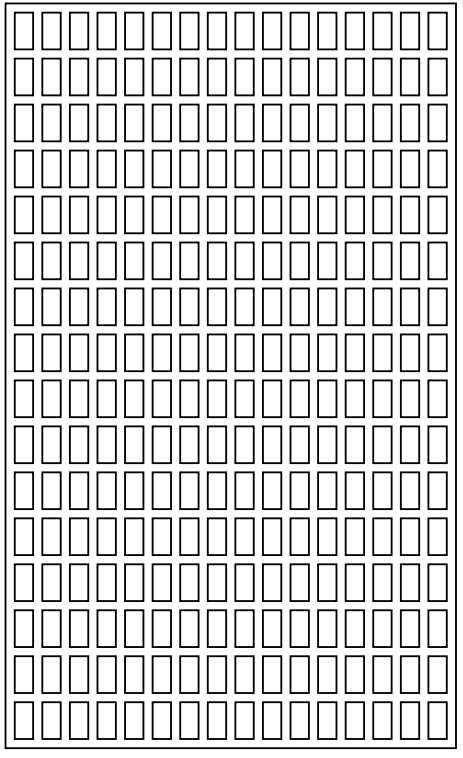
Microbump

- **Extension of solder balls and “C4” bumps.**
- **$\sim 40\mu\mathrm{m}$ pitch is available. In Intel MAX GPU $37\mu\mathrm{m}$ pitch has been used.**
- **Will go below $10\mu\mathrm{m}$ in future.**
- **Used in all HBM memories shipped so far up to HBM3e.**
- **High yield is possible since we stack dies after they are cut out from wafer.**

Hybrid bonding

- Cu pad on dies are bonded through thermal process, after dies are bonded (no adhesive or whatever is used. SiO2-SiO2 “direct bonding”)
- First used in Sony’s CMOS image sensors.
- $\sim 5\mu\mathrm{m}$ pitch is available now.
- $< 1\mu\mathrm{m}$ will be available.
- Roughly 100x more pads can be used compared to microbumps.
- Heat resistance is quite low. “DRAM on top” structure is possible.
- Bonding process is “Wafer-on-Wafer”. So the cost should be low. However, there is no way to remove defective dies before bonding. So some new design method which achieve near 100% yield is essential.

DRAM design image



Very large number of “small” DRAM blocks.

16x16, 32x32 etc.

Each block has its control input, address input and data I/O pads.

Example: $100\mathrm{Mbit}/\mathrm{mm}^2$ **density DRAM,** $800\mathrm{mm}^2$ **die = 80Gbit (effective 72Gbit)**

2048 36mbit blocks (with ECC). 144bit I/O. Total pads/die = 300K. With 4 dies 1.2M pads.

500MHz data rate gives 80TB/s.

So extremely high memory bandwidth is mechanically possible.

Question: power consumption.

Power consumption and capacity

Current goal with 3D DRAM: 0.5pJ/bit (around 1/15 of the actual power consumption of HBM)
This means 80TB/s = 640 Tb/s = 320W. NVIDIA Rubin Ultra: 32TB/s, 3-4kW.
10x better than Rubin Ultra (...)
Practical problem: 4 DRAM dies of $800\mathrm{mm}^2$ size gives only 36GB. We need much more.

- **Use more dies per package (possible)**
- **Use DRAMs with more advanced process technology (maybe in future)**

How far can we go?

- **With hybrid bonding very large number of pads is possible. This means that the DRAM design can be greatly simplified.**
- **For example, all of the digital logic circuits on current synchronous DRAM design could be moved to the logic die side.**
- **0.1-0.2 pJ/bit is within reach (1/20 — 1/40 of HBMx)**

MN-Core and 3D DRAM

- **With MN-Core architecture, it is natural to add DRAM units to each PE.**
- **Very similar to large-scale SIMD machines like TMC CM-2 and MasPar MP-2. So programming model will be similar. Data parallel language like HPF (OpenACC) can be used. Cuda-like one is also possible.**
- **We are currently developing MN-Core L1000, the first generation LLM processor with 3D stacked DRAM and on-chip distributed memory architecture.**

3D DRAM and HPC

- **HPC applications: regular grid, irregular grid, particles, dense matrices.**
- **irregular data structure (graphs) may be another class.**
- **Regular and irregular grid: fast DRAM would be of great help.**
- **Particles: fast DRAM would improve the performance of short-range interaction calculation (and treecode) greatly.**

  (PeTar is built on top of parallel treecode (FDPS))
- **matrix calculations: "Efficient" algorithms generally reduces the calculation cost by increasing the memory access cost. So fast DRAM would help.**

Programming MN-Core L1000

(This is my personal view and not the official view of Preferred Networks)

- **We should/can develop a programming environment rather similar to that of CM-2 or Maspar MP-2.**
- **I hope that we will be able to make de facto standard.**

Summary

- GRAPE was good because of highly specialized architecture.
- Our first try to make "multi-purpose" processor, GRAPE-DR, was not a great success, because it was not much faster than GPUs.
- With MN-Core, we could have achieved 10x better performance, but NVIDIA V100 adopted a similar specialized architecture.
- With MN-Core L1000, we will introduce on-chip distributed memory architecture, which I hope will achieve > 10x memory bandwidth.
- (With L2000, I hope to add full support for FP64.)


申