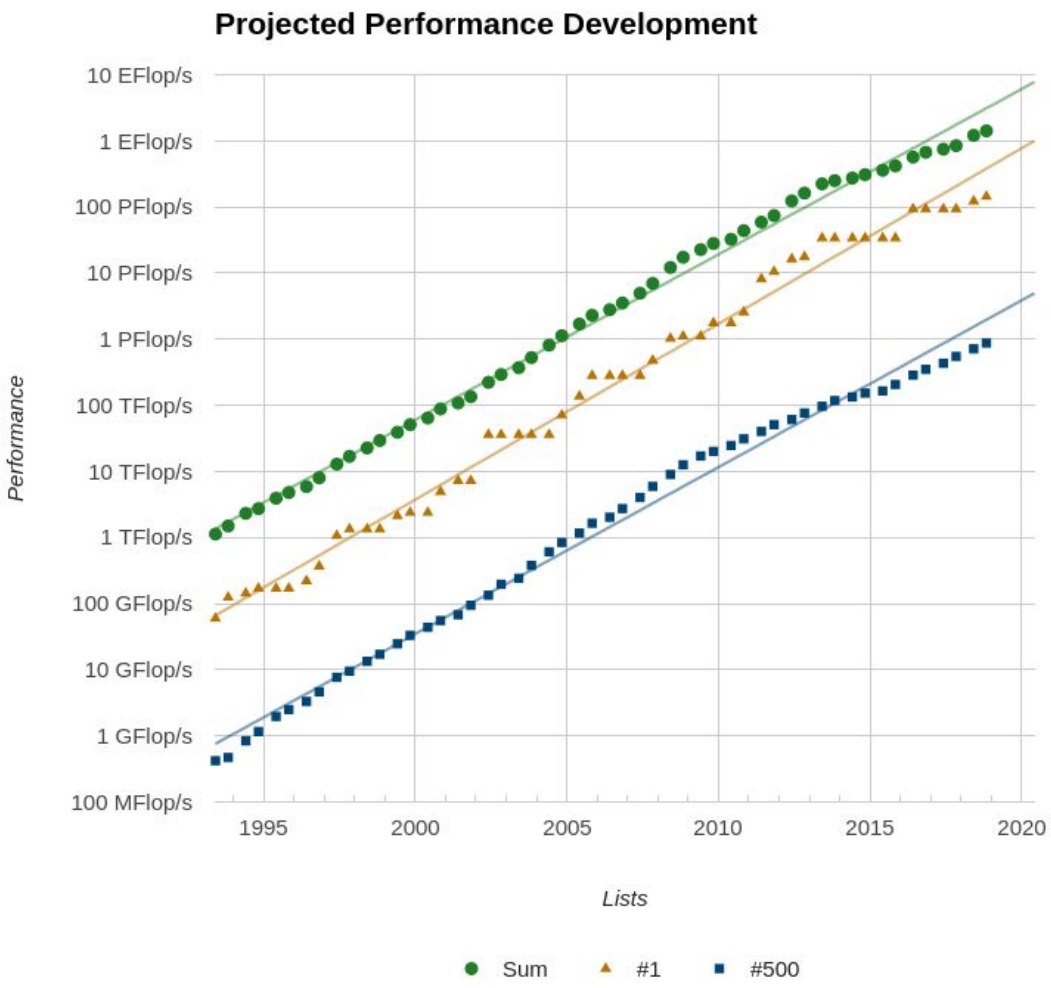
Introduction to Modern HPC Architectures

Alessandro Fanfarillo
elfanfa@ucar.edu

ISS (SEA) 2019
April 8th 2019

Performance Trend



Nov 2018: Summit most powerful supercomputer
(Oak Ridge): 200.8 PFLOPS

First exascale system hopefully by 2020.

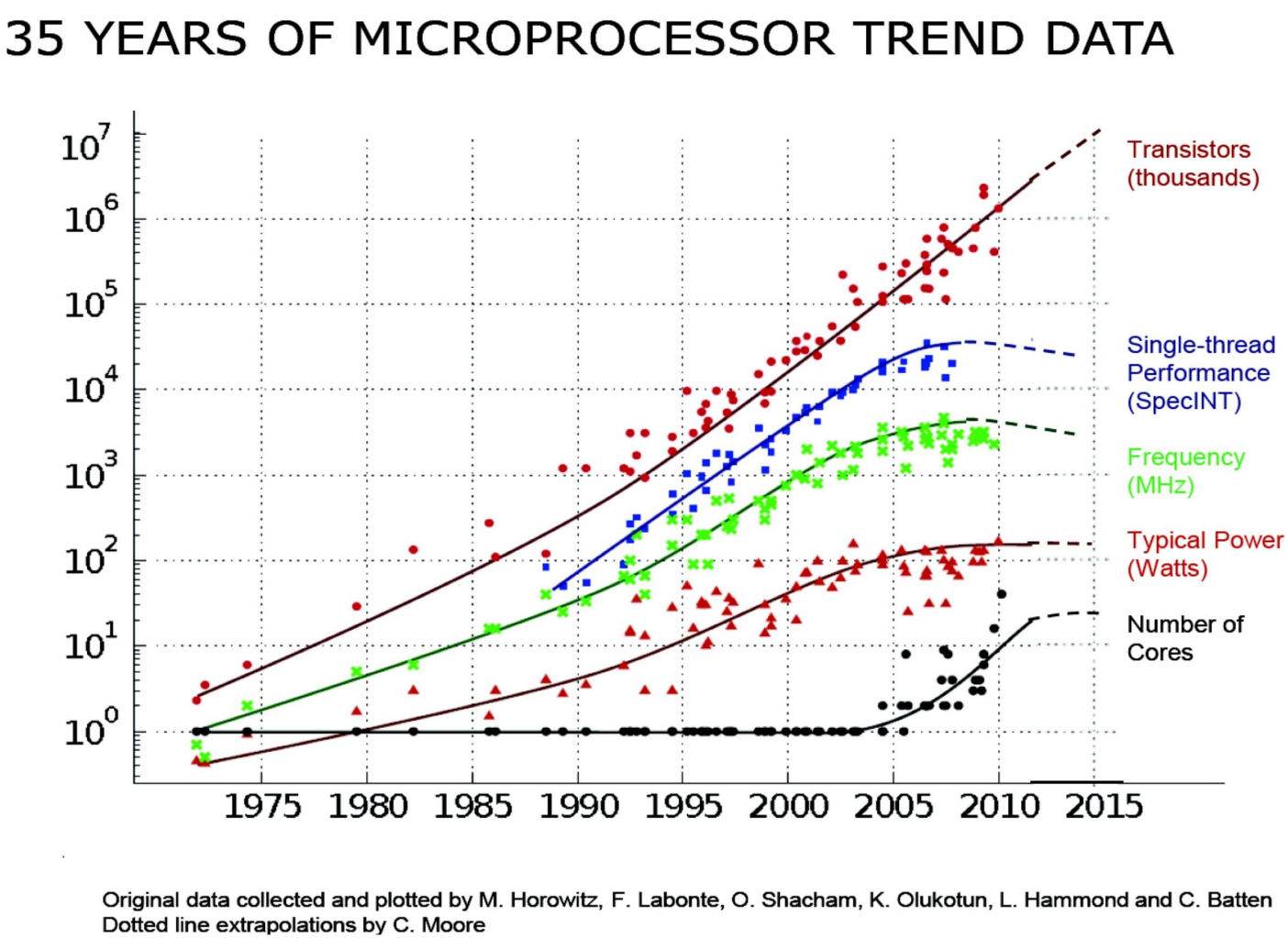
Architectural changes:

1. Low clock rate
2. More cores per node
3. More nodes per cluster
4. Longer vector instructions
5. Heterogeneous hardware (e.g. GPUs)
6. Domain-specific architectures

Consequences:

1. Less memory per core
2. More hw failures
3. More parallelism
4. New programming paradigms

Microprocessor Trend



Moore's law, Performance and Dennard's Scaling

Moore's law: the number of transistors of a typical processor chip doubles every 18-24 months.

Before 2005, we had faster (higher clock rates) processors every 18-24 months.

Size and speed are related: the smaller something is, the quicker it can be changed.

Smaller transistors can work at higher clock rates (higher performance).

END OF DENNARD'S SCALING

Power = alpha * C * F * V

- alpha: percent time switched
- C: Capacitance
- F: Frequency
- V: Voltage

Capacitance is related to the area

Smaller transistors, working at smaller voltage, could operate at higher frequencies at the same power.

Leakage current and threshold voltage was **NOT** consider by Dennard (Off is not totally off).

As transistors get smaller, power density increases because these don't scale with size.

Two laws and three walls

1. **Moore's law**: the number of transistors of a typical processor chip doubles every 18-24 months
2. **Dennard's law**: as transistors get smaller, their power density stays constant, so the power use stays in proportion with the area (**failed around 2005**)

Performance enhancement limited by:

1. **Power wall**: higher clock frequency means higher heat generation.
2. **Instruction-level parallelism (ILP) wall**: exploiting parallelism using longer instruction pipelines and/or longer vector instructions has limits.
3. **Memory wall**: the performance of memories is lower than the performance of microprocessors.

Parallelism is also limited (Amdahl's law)

Dark Silicon Apocalypse

- Dennard's law failed in 2005 (power use doesn't scale with transistor's size).
- Moore's law keeps holding (2 times the number of transistors after two years) although is slowing down and close to an end.

Result: prossors packed with transistors that cannot be powered on at the same time.

Possible solutions (4 Horsemen):

1. Shrinking chips (not smart)
2. Dim silicon (underclocked)
3. **Circuit specialization - domain-specific architectures**
4. Deus ex machina (better transistors?)

Exascale Compute Node

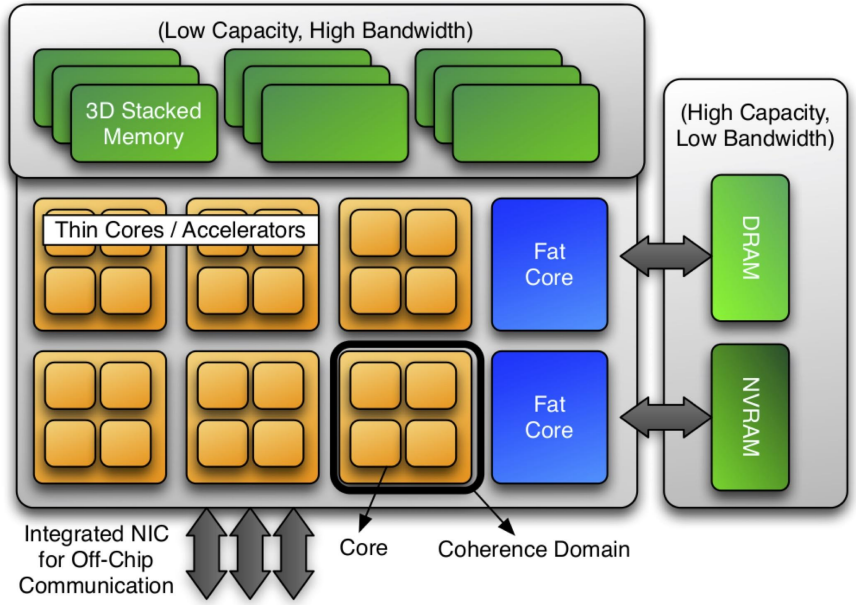


Image taken from J.A. ANd et al.
Abstract Machine Models and Proxy Architectures for Exascale Computing

- Fat and thin cores
- Several types of memory
- Small Coherence Domain (or manually programmable caching)
- Integrated NIC (low power, high message throughput)

By 2018, a floating point operation will consume ~ 40 pJ/bit.
Reading data from regular DDR3 DRAM will cost ~70 pJ/bit (Shalf et al. 2013)

Domain-specific architectures are currently the future of computing.

AI Specialized Hardware

- **NVIDIA Tensor Cores** (Volta Architecture) - Training
- **Google Tensor Processing Units** (TPUs) - Inference
- Microsoft Brainwave/Catapult (FPGAs) - Inference
- Intel Neural Network Processor (NPP) Nervana - Training, Inference
- **Intel Movidius Myriad 2 Vision Processing Unit** - Inference
- IBM TrueNorth (Neuromorphic) - Inference
- **Intel Cascade Lake** (x86 with AVX-512 VNNI) - Inference

From FP32 to INT8

FP32 can represent a large variety of numbers (up to ~2^128)

INT8 can represent 256 values…

Using INT8 for inference has several benefits:

- More power efficient operation due to smaller multiplies
- Lower pressure on cache and memory
- Precision and dynamic range sufficient for most models

It's usually possible to represent the values of a tensors with 8 bits and one common multiplier.

FP32 can be "squeezed" into INT8 with a "quantization" process.

Tensorflow and other frameworks already support INT8.

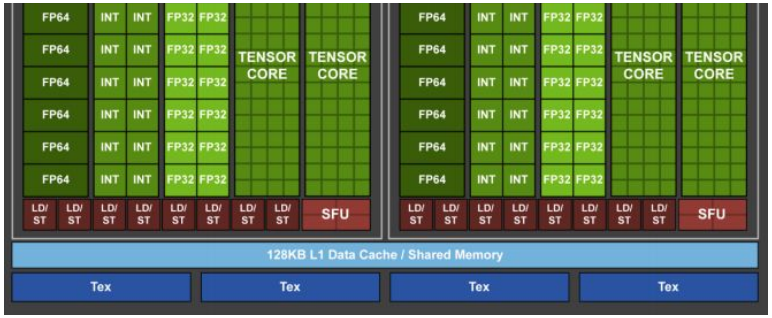
NVIDIA Tensor Cores - Volta

$$D = \begin{pmatrix} A_{0,0} & A_{0,1} & A_{0,2} & A_{0,3} \\ A_{1,0} & A_{1,1} & A_{1,2} & A_{1,3} \\ A_{2,0} & A_{2,1} & A_{2,2} & A_{2,3} \\ A_{3,0} & A_{3,1} & A_{3,2} & A_{3,3} \end{pmatrix} \begin{pmatrix} B_{0,0} & B_{0,1} & B_{0,2} & B_{0,3} \\ B_{1,0} & B_{1,1} & B_{1,2} & B_{1,3} \\ B_{2,0} & B_{2,1} & B_{2,2} & B_{2,3} \\ B_{3,0} & B_{3,1} & B_{3,2} & B_{3,3} \end{pmatrix} + \begin{pmatrix} C_{0,0} & C_{0,1} & C_{0,2} & C_{0,3} \\ C_{1,0} & C_{1,1} & C_{1,2} & C_{1,3} \\ C_{2,0} & C_{2,1} & C_{2,2} & C_{2,3} \\ C_{3,0} & C_{3,1} & C_{3,2} & C_{3,3} \end{pmatrix}$$

FP16 or FP32 FP16 FP16 FP16 or FP32

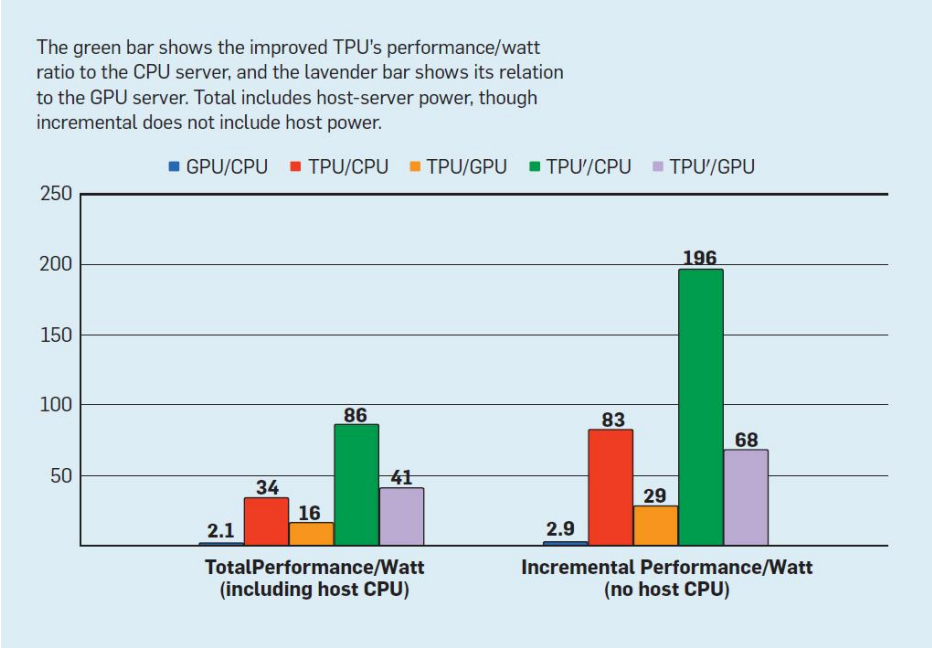
One matrix-multiply-and-accumulate on 4x4 matrices per clock cycle.

Input half precision, matmul half precision, accumulate single precision.



Google Tensor Processing Unit (TPU)

- One processor: **single-thread** makes it easier for the system to stay within a fixed latency limit.
- Large 2D multiply unit: matrix multiplies benefit from two-dimensional hardware.
- INT8: INT8 are used instead of FP32 to improve efficiency.
- Dropped features: TPUs don't have features that CPUs and GPUs have. The silicon gets used for domain-specific on-chip memory.

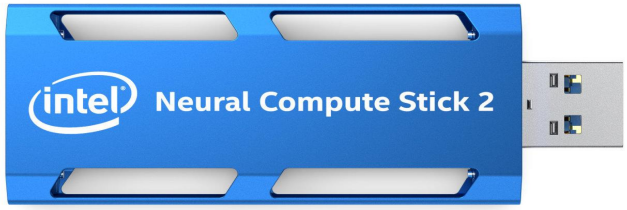


A Domain-Specific Architecture for Deep Neural Networks

by Norman P. Jouppi, Cliff Young, Nishant Patil, and David Patterson

Published in "Communications of the ACM", 09/2018, Vol. 61 NO. 09.

Real-time object detection - Intel Movidius

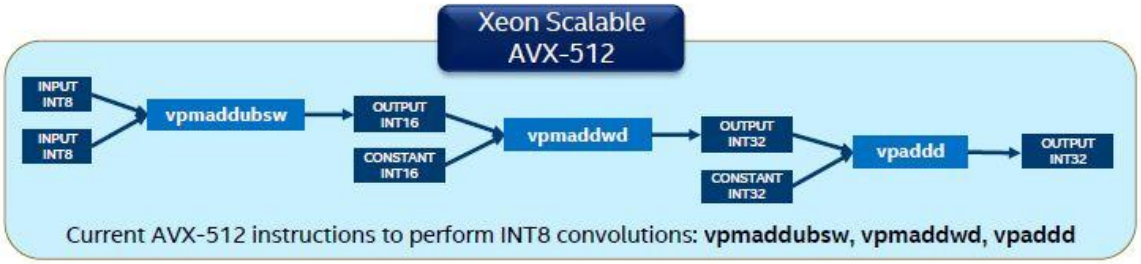


Highly specialized processor (VPU) for computer vision

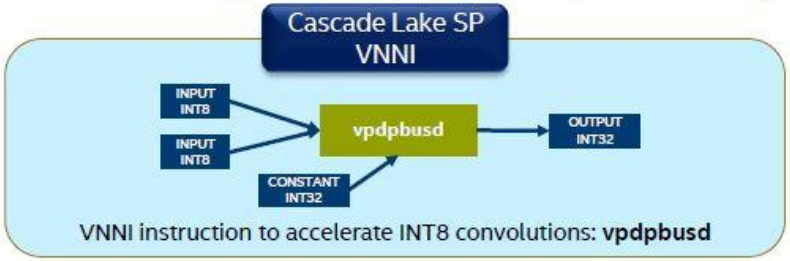
x86 - Intel Cascade Lake

- New design compared to Skylake
- 56 cores, higher memory bandwidth, support for Optane DC persistent memory
- New instructions: Vector Neural Network Instruction (VNNI), used for inference on INT8

AI/DL Inference Enhancements on INT8 with VNNI



New instructions for accelerating AI on Intel® Xeon® Scalable processors using int8 data



64 multiply-add in 3 instructions.
1.33x speedup compare to FP32

64 multiply-add in one instruction.
4x speedup compare to FP32

Alternatives to x86 - ARM

Although x86 is still dominant in HPC, there are several alternatives:

- Power
- SPARC
- **ARM** (new Scalable Vector Extension)

ARM's focus on low power solutions with high energy efficiency can be a good fit for HPC.

ARM sells Intellectual Property, not CPUs!

SVE supports vector length agnosticism: properly written code will run on a large range of vector lengths without any modification (from 128 to 2048 bits in increments of 128 bits).

SVE supports Predictation (if statements in loops), Gather and Scatter (as opposed to the old NEON SIMD ISA).

Conclusions and Discussion

- The ending of Moore's law and Dennard's scaling leaves domain-specific architectures as the future of computing.
- Some domain-specific architectures (e.g. TPUs), don't make use of explicit parallelism.
- Memories are changing a lot as well (Intel Optane DC Persistent Memory).
- Inference performance is everyday more important for AI applications.
- Quantum computing?

Thanks!

Please, ask your questions!