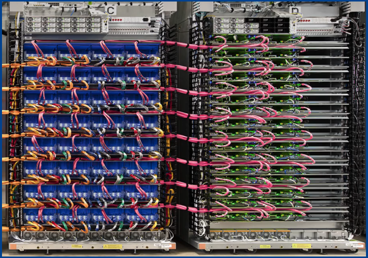
AARHUS UNIVERSITY

LESSON 10-2: Hardware (and Frameworks)

CARSTEN EIE FRIGAARD

AUTUMN 2024



"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience E." — Mitchell (1997).

L10-2: Hardware (and Frameworks)

Agenda

- AI definition revisited.
- Frameworks for Machine Learning.
- Hardware for Machine Learning,
  - CPUs,
  - GPUs,
  - TPUs (and DPUs),
  - Exotic Hardware,
  - (Coin-mining, low-hash-rate,)
  - ML-training costs.

OUTDATED: ChatGPT and the Press

https://openai.com/blog/chatgpt-plugins
https://www.zetland.dk/historie/s8D3lrNQ-mejv5vgV-9170d
https://techmonitor.ai/technology/ai-and-automation/chatgpt-ai-compute-power
https://www.version2.dk/artikel/elon-musk-steve-wozniak-og-tusind-andre-i-faelles-opraab-saet-de-gigantiske-ai
https://ing.dk/artikel/det-italienske-datatilsyn-forbyder-chatgpt
https://www.dr.dk/nyheder/seneste/australsk-borgmester-truer-selskabet-bag-chatgpt-med-injuriesag

AI Definition Revisited

EU AI Act

ING/VERSION2 Nyheder Blogs Debat Tip os IT job Event Mere Bliv abonnent

EU | Offentlig it | AI | Cybersikkerhed | Kunstig Intelligens | Digital Tech Summit | MitID | Kryptovaluta | Politik | AI Act

Endelig definition af kunstig intelligens klar til at blive godkendt i AI Act

AI Act | 13. november kl. 10:48 | 2

Illustration: Europa-Parlamentet.

OECD har netop vedtaget en ny definition af 'kunstig intelligens'. EU forventes at indarbejde den nye afgrænsning i AI Act.

OECD, The Organisation for Economic Co-operation and Development, har vedtaget sin nye definition af kunstig intelligens. Det skete onsdag i sidste uge, da OECDs udvalg for digital økonomisk politik og

[https://www.version2.dk/artikel/endelig-definition-af-kunstig-intelligens-klar-til-blive-godkendt-i-ai-act]

AI Definition Revisited

Der foreligger endnu ikke en dansk oversættelse af den ny definition, men den engelske version lyder:

> *An AI system is a machine-based system that, for explicit or implicit objectives, infers, from the input it receives, how to generate outputs such as predictions, content, recommendations, or decisions that can influence physical or virtual environments. Different AI systems vary in their levels of autonomy and adaptiveness after deployment.*
>
> —[oecd.ai/en/ai-principles]

[https://www.version2.dk/artikel/endelig-definition-af-kunstig-intelligens-klar-til-blive-godkendt-i-ai-act]

FRAMEWORKS FOR MACHINELEARNING

Frameworks for Machine Learning

name	note
Sci-Kit Learn	(INRIA)
Keras	interface
Tensorflow	Google
Pytorch	Facebook
CNTK	Microsoft
Apache MXNet	Amazon
Core ML	Apple
Caffe	(Berkeley)
H2O	??
Shogun	??

Frameworks for Machine Learning

Intel: `scikit-learn-intelex`

Intel® Extension for Scik ×

https://intel.github.io/scikit-learn-intelex/

Intel(R) Extension for Scikit-learn* 2021.5 documentation

Search the docs ...

ABOUT

- Acceleration
- Patching
- Medium Blogs
- System Requirements
- Memory Requirements

GET STARTED

- Installation

Intel® Extension for Scikit-learn*

With Intel® Extension for Scikit-learn* you can accelerate your Scikit-learn applications and still have full conformance with all Scikit-Learn APIs and algorithms. Intel® Extension for Scikit-learn* is a free software AI accelerator that brings over 10-100X acceleration across a variety of applications.

Intel® Extension for Scikit-learn* offers you a way to accelerate existing scikit-learn code. The acceleration is achieved through patching: replacing the stock scikit-learn algorithms with their optimized versions provided by the extension.

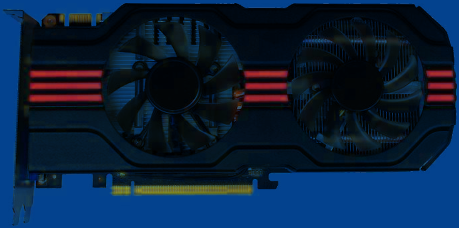
Designed for Data Scientists and Framework Designers

Intel® Extension for Scikit-learn* was created to provide data scientists with a way to get a better performance while using the familiar scikit-learn package and getting the same results.

Contents

- Usage
- Important Links

HARDWARE FOR MACHINELEARNING



Hardware for Machine Learning

Methods and Terminology (SKIP most except μ, η)

Objective:

Why optimize using 'application specific' hardware?

- Effectiveness:
  - cost of purchasing/operating systems,
    μ=FLOPS/$
    η=FLOPS/Watt
  - cut-down developer waiting time,
  - make modelling iterations fast (say minutes).
- Big-data:
  - enable training on x-large data.
- Real-time constraints:
  - inference (on visual data) in real-time,
  - low-power constraints.

Hardware for Machine Learning

Methods and Terminology

Heterogen computing: systems that use more than one kind of processor or cores, say CPU + GPU.

Cluster computing: (loosely or) tightly connected computers that work together; can be viewed as a single system.

HPC: High-performance computing; a 'super-computer' with high level of performance (vs. general-purpose computer).

Flynn's taxonomy:

SISD: single instruction, single data,

SIMD/SIMT: single instruction, multiple data/threads (data parallel); this one is our objective,

(MISD: multiple instructions, single data (fault tolerance)),

(MIMD: multiple instructions, multiple data (distributed)).

Methods and Terminology

Amdahl's law

What speedup can we expect when optimizing/ parallelizing a program?

$$S = \frac{1}{(1-p) + p/s}$$

where

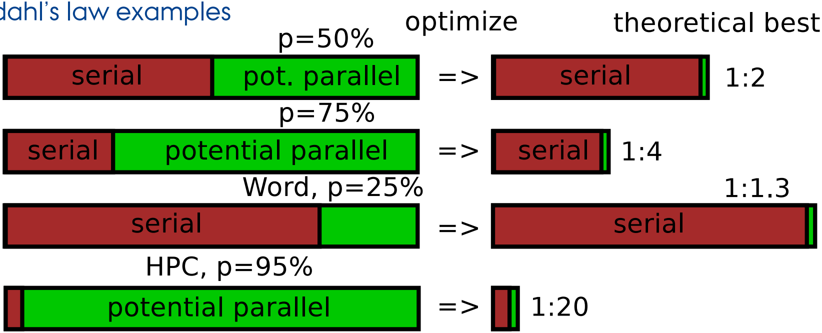
- S: total program speedup,
- p: fraction of the program that can be run in parallel,
- s: parallel fraction speedup factor.

If we take $s = \infty$ the theoretical max. total speedup will be

$$S_{\max} = \frac{1}{(1-p) + p/\infty} = \frac{1}{1-p}$$

Methods and Terminology

Amdahl's law examples

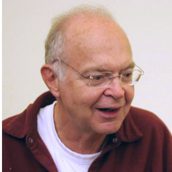


And there is seldom need to optimize unless $S_{max} \gg 2$.

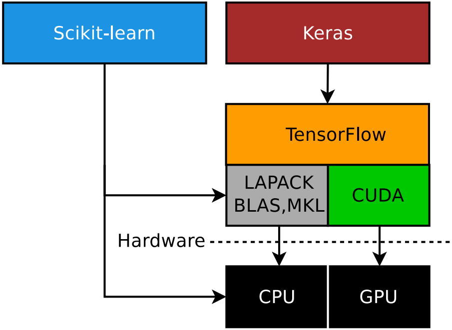
A 1:5 speedup is a good, ad-hoc compromise btw. a fast program and a costly optimization dev. phase.

What does Donald E. Knuth say about premature optimization?

"Root of all evil"!



RESUMÉ: Keras and Tensorflow



GP-GPU: General-Purpose Graphics Processing Unit...or just **GPU**.

CUDA: Compute Unified Device Architecture, API for SIMD/SIMT on GPU,

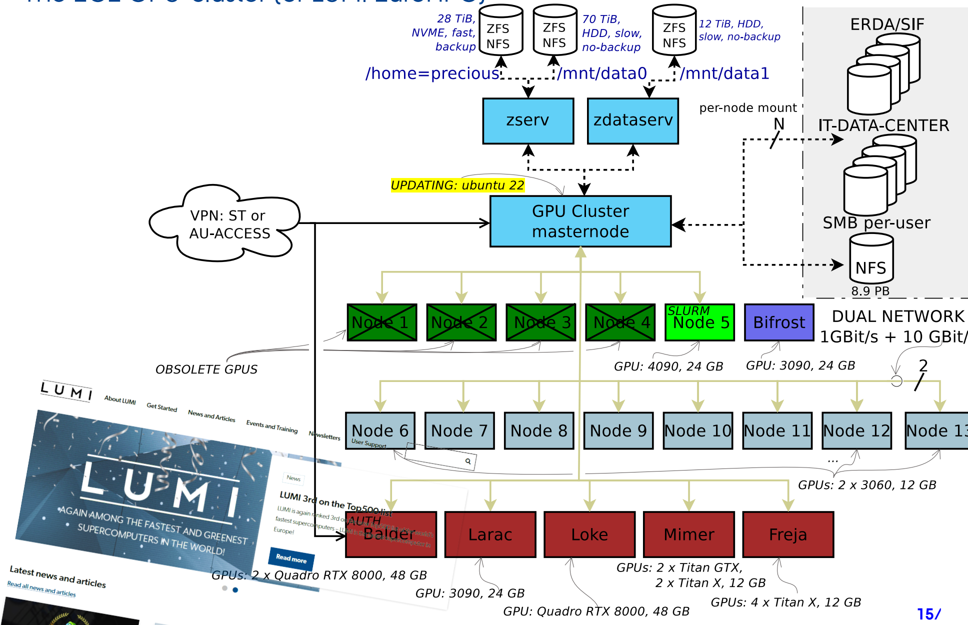
LAPACK: Linear Algebra Package, numerical linear algebra lib, with roots in FORTRAN 77,

BLAS: Basic Linear Algebra Subprograms; vector/matrix lib,

MKL: Math Kernel Library; fast Intel-arch optimized math lib.

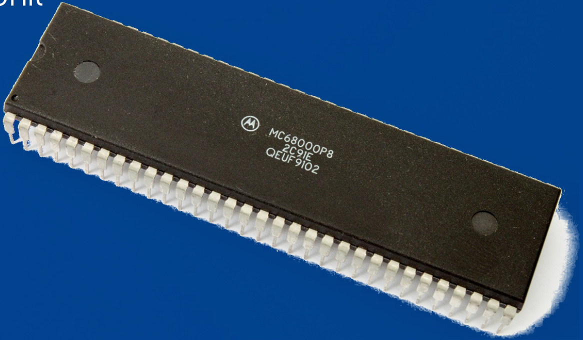
High-Performce-Computing (HPC)

The ECE GPU-cluster (or LUMI EuroHPC)



CPUS

Central Processing Unit



CPUs

Build Tensorflow from source (SKIP)

- for specific architecture, say ARM,
- or for HPC optimization for all CPU feature

> lscpu

```
Architecture:  x86_64
Model name:    Intel(R) Core(TM) i7-6600U CPU @ 2.60GHz
Flags:         fpu vme de pse tsc msr pae mce cx8 apic sep mtrr
               pge dts acpi mmx fxsr fma..sse sse2..sse4_1sse4_2..
```

Using Docker and pulling TF from GIT + a lot of scripting!

```
> git clone https://github.com/tensorflow/tensorflow
> git checkout 1.12
```

(lots of scripting and pain..)

```
> bazel build -copt=-mfma -copt=-msse4.2 tensorflow
```

Howtos:

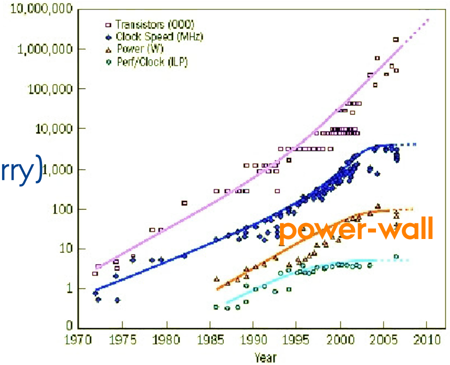
https://www.tensorflow.org/install/source
https://www.pugetsystems.com/labs/hpc/Build-TensorFlow-CPU-with-MKL-and-Anaconda-Python-3-6-using-a-Docker-Container-1133
https://www.pugetsystems.com/labs/hpc/Build-TensorFlow-GPU-with-CUDA-9-1-MKL-and-Anaconda-Python-3-6-using-a-Docker-Container-1134

CPUs

SIMD/Data parallel compute using CPU

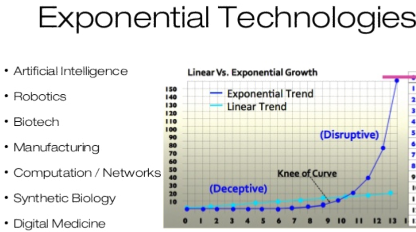
CPU architectures:

- i386/AMD64,ARM (Raspberry)
- all CPU types: lots of cores; end of Moore's Law, 'Singularity University', problem w. exp. growth?



Scikit-learn problem with SIMD:

- no ML-algo GPU hardware enabled,
- many ML-algo not even multicore aware!



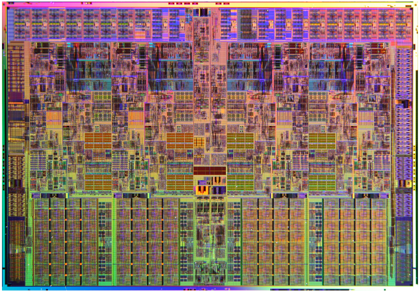
Keras/TensorFlow:

- CPU optimization of TF possible,
- (GPU enabled TF possible, using CUDA + cudNN).

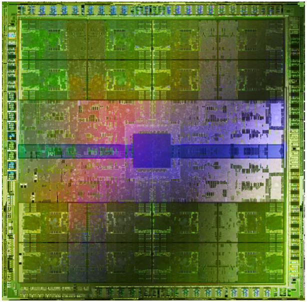
GP-GPUS

General-Purpose Graphical Processing Unit

CPUs vs GPUs



Nehalem CPU
die size: $\sim$700 mm^2
transistors: $\sim 2.3 \cdot 10^9$



Fermi GPU
die size: 520 mm^2
transistors: $\sim 3 \cdot 10^9$

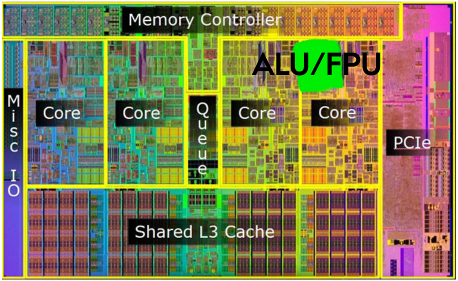
Nvidia arch. transistor counts	
Pascal	$\sim 15 \cdot 10^9$
Turing	$\sim 19 \cdot 10^9$
Volta	$\sim 21 \cdot 10^9$
Ampere	$\sim 28 \cdot 10^9$ (GPU 3090, 8nm)

CPUs vs GPUs

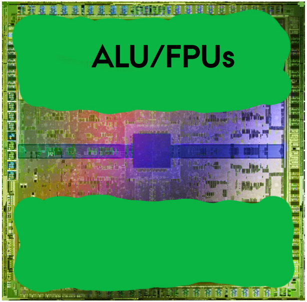
So many transistors, but how many for the ALUs/FPUs?

What makes GPUs so excellent HW for Machine Learning?

ALU: Arithmetic logic unit
FPU: Floating-point unit
Memory Controller: six controllers on GPU,
CPU: lots of speculative execution; waste of transistors.



CPU (type?)
with one ALU/FPU marked



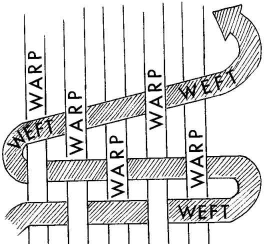
Fermi GPU
ALUs/FPUs all over

GPUs

Fundamental Problems with the GPUs Hardware (SKIP most, except WARP)

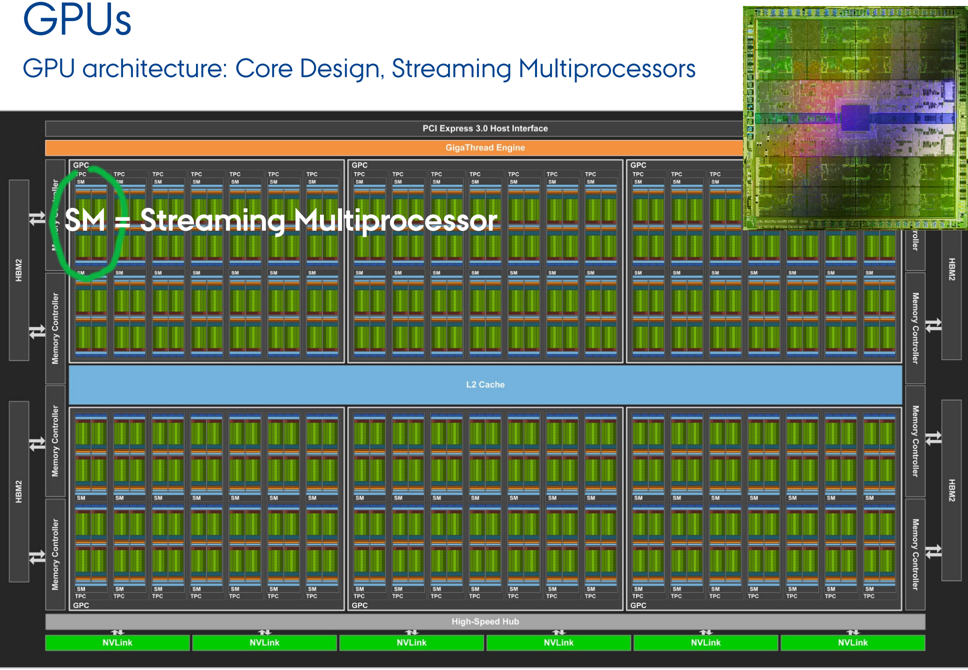
GPUs had several *Achilles heels* related to its hardware, many of them addressed in the latest Volta V100 architecture:

- coding problems graphically, now CUDA,
- no STACK, now added,
- no CACHE (or Texture only), now both L_1 and L_2 cache,
- distinct GPU memory, now UNIFIED memory,
- SIMT WARP-bunch of 32-threads, now true SIMD,



GPUs

GPU architecture: Core Design, Streaming Multiprocessors



GPUs

Core Design for a SM (Streaming Multiprocessor)

Volta SM design
(new gen. GPU):

FP64/32:FPUs
INT: ALUs
TENSOR CORES: ?

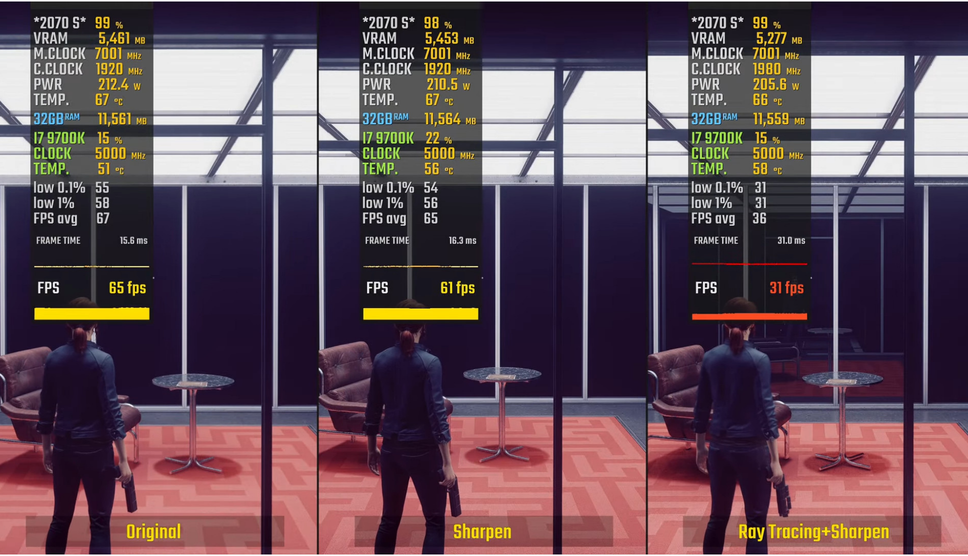
Ampere, RTX, 3090:
Raytracingkerner: ?

GEFORCE RTX 3090

NVIDIA CUDA® kerner	10496	Raytracingkerner	2. generation	Standard hukommelseskonfiguration	24 GB GDDR6X
Høj CPU-hastighed (GHz)	1.70	Tensor Cores	3. generation	Hukommelsesgrænsefladens bredde	384-bit
Normal CPU-hastighed (GHz)	1.40				

GPUs

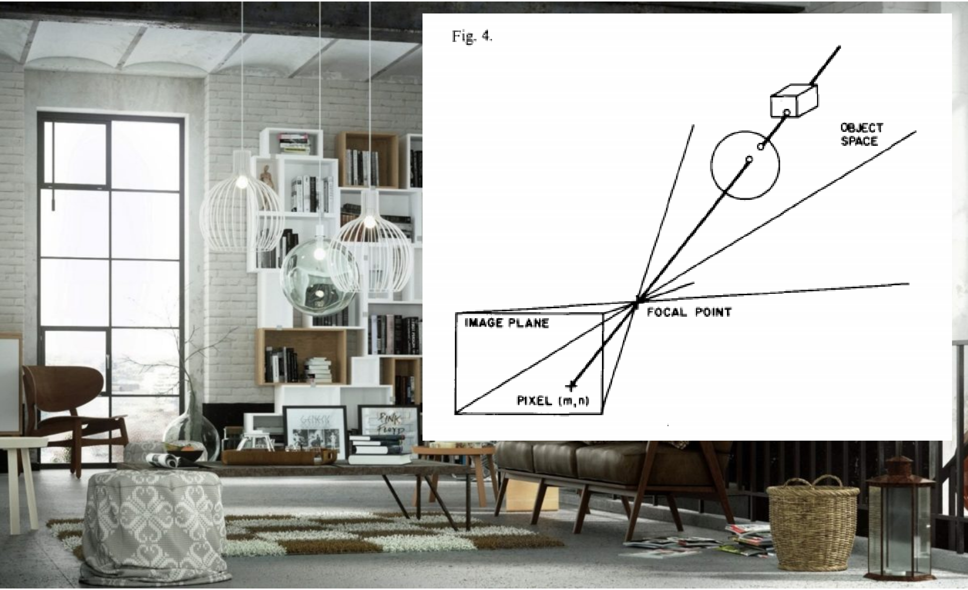
Demo: RTX and raytracing



[http s://www.gamingscan.com/nvidia-rtx-vs-gtx/]

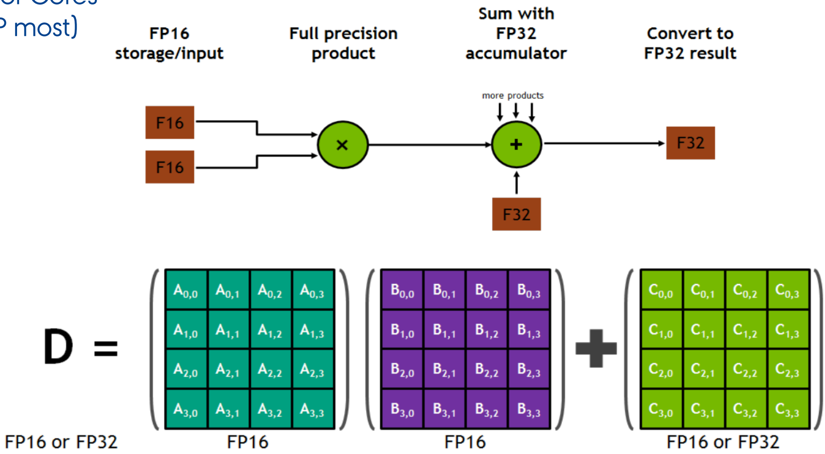
GPUs

Tensor cores: Raytracing vs Rasterization on GPUs



GPUs

Tensor Cores (SKIP most)



Tesla V100's Tensor Cores deliver up to 125 Tensor TFLOPS for training and inference applications.

[volta-architecture-whitepaper.pdf]

HPC Top500

Best High-Performance Computer

[https://www.top500.org/list/2007/06/]
[https://en.wikipedia.org/wiki/List_of_Nvidia_graphics_processing_units]

Model	Launch	Transistors (billion)	Die size (mm^2)	Processing power (TFLOPS)			
				Single precision	Double precision	Half precision	Tensor compute (FP16) (2:1 sparse)
GeForce RTX 3070[120]	October 29, 2020[121]	17.4	392.5	17.66 20.37	0.276 0.318	17.66 20.37	141.31 162.98
GeForce RTX 3080[123]	September 17, 2020	28.3	628.4	25.06 29.76	0.392 0.465	25.06 29.76	200.54 238.14
GeForce RTX 3090[124]	September 24, 2020	28.3		29.38 35.68	0.459 0.558	29.38 35.68	235.08 285.48

HOME LISTS STATISTICS RESOURC

Home »Lists »TOP500 »June 2007

JUNE 2007

Rank	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	**BlueGene/L** - eServer Blue Gene Solution, IBM DOE/NNSA/LLNL United States	131,072	280.6	367.0	1,433
2	**Jaguar** - Cray XT4/XT3, Cray/HPE DOE/SC/Oak Ridge National Laboratory United States	23,016	101.7	119.3	
3	**Red Storm** - Sandia/ Cray Red Storm, Opteron 2.4 GHz dual core, Cray/HPE NNSA/Sandia National Laboratories	26,544	101.4	127.4	

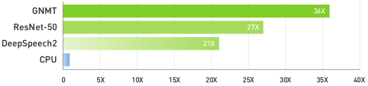
GPUs

Turing Tensor Core GPU (this is not a commercial!)

NVIDIA T4 enterprise GPUs supercharge the world's most trusted mainstream servers, bla. bla...

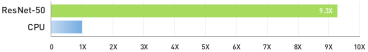
...s. Its low-profile, 70-watt (W) design

Inference Performance

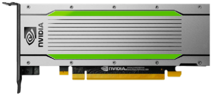


Comparisons made of one NVIDIA Tesla T4 GPU and servers with a dual-socket Xeon Gold 6140 CPU.

Training Performance



Comparison made between dual NVIDIA Tesla T4 GPUs and servers with a dual-socket Xeon Gold 6140 CPU.



SPECIFICATIONS

GPU Architecture	**NVIDIA Turing**
NVIDIA Turing Tensor Cores	**320**
NVIDIA CUDA® Cores	**2,560**
Single-Precision	**8.1 TFLOPS**
Mixed-Precision (FP16/FP32)	**65 TFLOPS**
INT8	**130 TOPS**
INT4	**260 TOPS**
GPU Memory	**16 GB GDDR6 300 GB/sec**
ECC	**Yes**
Interconnect Bandwidth	**32 GB/sec**
System Interface	**x16 PCIe Gen3**
Form Factor	**Low-Profile PCIe**
Thermal Solution	**Passive**
Compute APIs	**CUDA, NVIDIA TensorRT™, ONNX**

[t4-tensor-core-datasheet-951643.pdf]

GPUs

When is the GPU faster than the CPU for NN?

GPU slower for CPU for a three-layer NN + MNIST, why?

- GPU needs a reasonable amount of trainable paramters + data to beat the CPU!

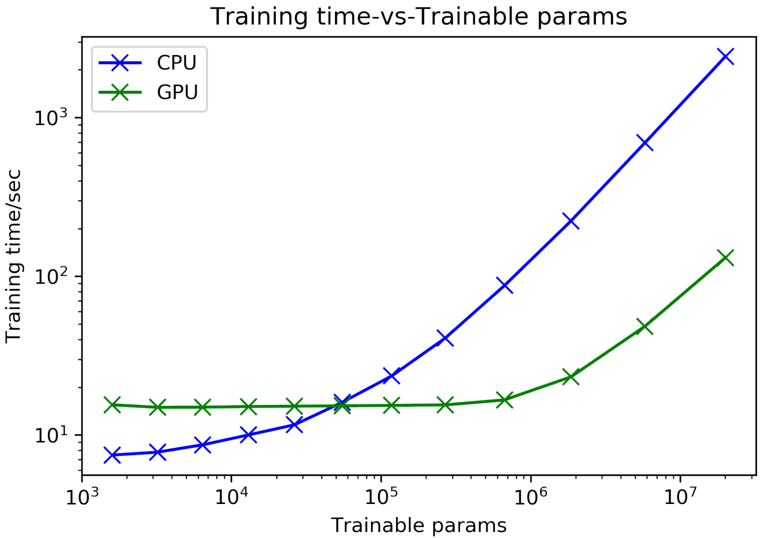
`model.summary()`:

```
#  i=12, n=4096
#  _________________________________________________________________
#  Layer (type)                 Output Shape              Param #
#  =================================================================
#  dense_46 (Dense)             (None, 4096)              3215360
#  _________________________________________________________________
#  dropout_31 (Dropout)         (None, 4096)              0
#  _________________________________________________________________
#  dense_47 (Dense)             (None, 4096)              16781312
#  _________________________________________________________________
#  dropout_32 (Dropout)         (None, 4096)              0
#  _________________________________________________________________
#  dense_48 (Dense)             (None, 10)                40970
#  =================================================================
#  Total params: 20,037,642
#  Trainable params: 20,037,642
#  Non-trainable params: 0
```

GPUs

Actual test on the GPU-server

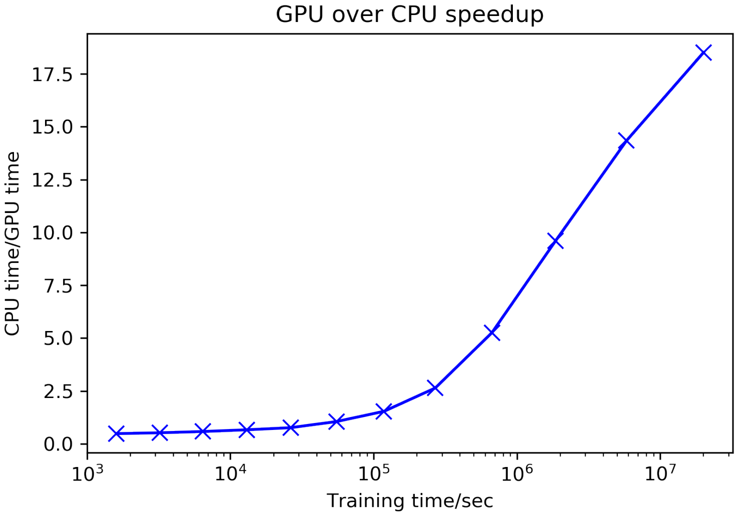
CPU vs GPU on MNIST for a three layer NN with dropout...



GPUs

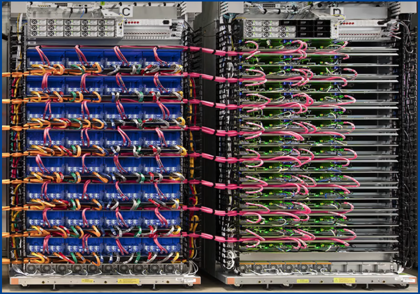
Actual test on the GPU-server

CPU vs GPU on MNIST for a three layer NN with dropout...



TPUS

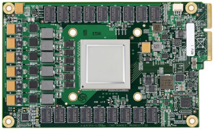
Tensor Processing Units



TPUs

Tensor Processing Units

Custom ASICs by Google

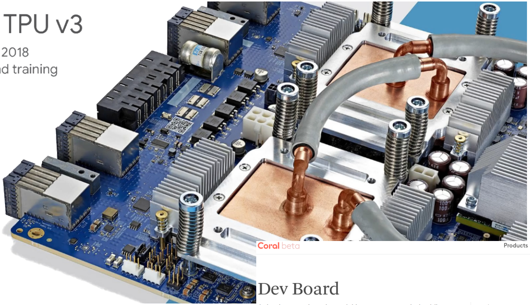


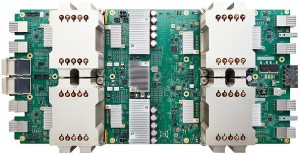
TPU v1

Launched in 2015
Inference only

Cloud TPU v3

Launched in 2018
Inference and training





TPU v2

Launched in 2017
Inference and training

Dev board with Google Edge TPU ML accelerator coprocessor

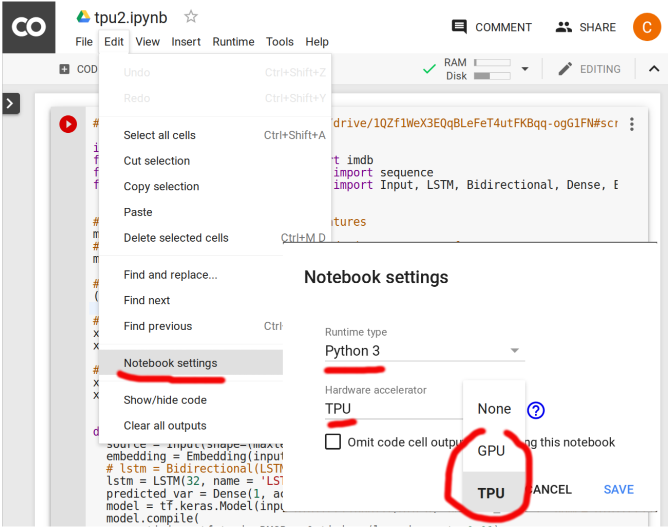
[https://coral.withgoogle.com/products/dev-board/]

TPUs

Access to TPUs (SKIP)

Free Jupyter Notebook environment with access to TPUs:

`https://colab.research.google.com`



TPUs

TPU Cloud

[https://storage.googleapis.com/nexttpu/index.html]

TPUs vs GPUs

Performance, TPUs vs GPUs, who wins? (SKIP most)

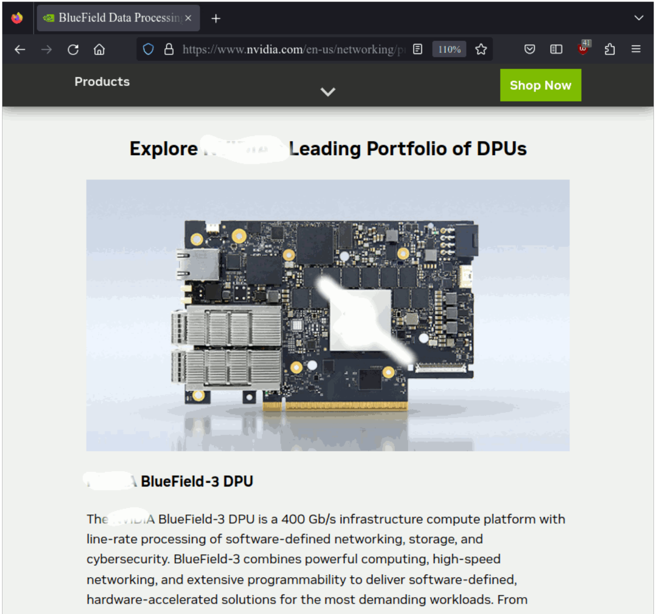
- Huge advantage for TPU performance-per-watt,
- Colab performance:
  inconclusive (TPU part does not work yet),
- TPU only for inference?

	K80 2012	TPU 2015	P40 2016
Inferences/Sec <10ms latency	1/13 TH	1X	2X
Training TOPS	6 FP32	NA	12 FP32
Inference TOPS	6 FP32	90 INT8	48 INT8
On-chip Memory	16MB	24 MB	11 MB
Power	300W	75W	250W
Bandwidth	320 GB/S	34 GB/S	350 GB/S

[https://www.extremetech.com/computing/247403-nvidia-claims-pascal-gpus-challenge-googles-tensorflow-tpu-updated-benchmarks]

DPUs

Data Processing Units

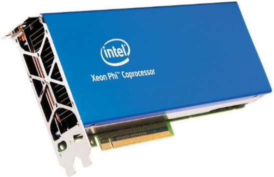


[https://www.nvidia.com/en-us/networking/products/data-processing-unit/]

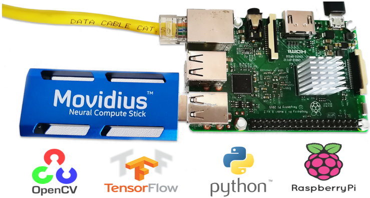
EXOTIC HARDWARE

Exotic Hardware

- Intel Phi multicore CPU, 64 i386 cores:



- Raspberry PIs + Intel Movideus stick



- FPGAs

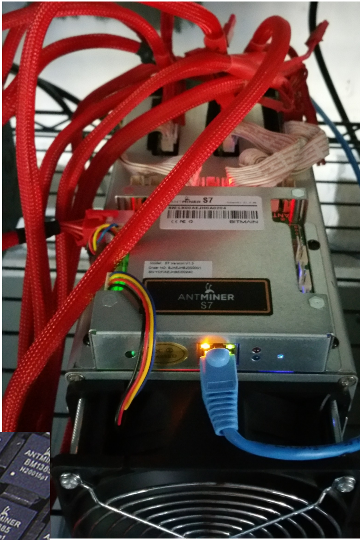
ASICs vs GPUs

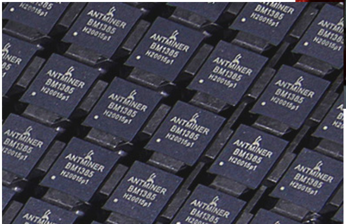
Bitcoin-mining and Low-hash-rate GPUs

AntMiner S7: based on ASICs.

Some specs:

- hash rate: 4.8 THash/s
- chips per unit: 162 x BM1385
- power consumption: **1210 W**
- power efficiency: 0.25 W/GHash
- price: \$479.95 $\sim$ 2880,- DKK
- production: **0.16 bitcoin/month**





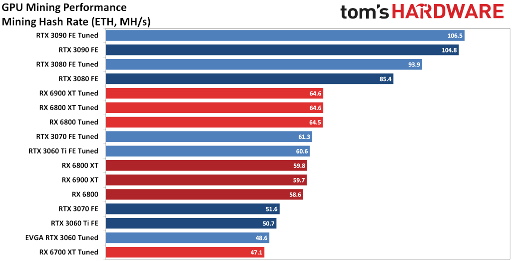
ASICs vs GPUs

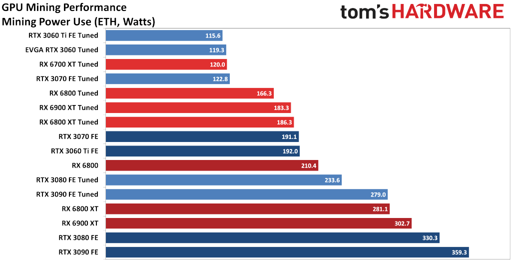
78 x GeForce RTX 3080 Mining Rig..

[https://www.techarp.com/computer/78-rtx-3080-mining-rig/]

ASICs vs GPUs

Coin-mining and Low-hash-rate GPUs





[https://www.tomshardware.com/how-to/optimize-your-gpu-for-ethereum-mining]

Training Cost and Hardware

The AI Brick Wall – A Pr

https://www.semianalysis.com/p/the-ai-brick-wall-a-prac

Subscribe

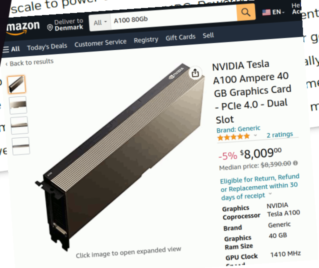
Sign in

Once you know the parameter count, token count, and model architecture, you can easily calculate the theoretical training costs for many popular models. In this example, we will use Nvidia A100s, using $1.5 per hour per GPU. Model/hardware "FLOPS utilization" will increase from 40% to 60% with larger model sizes, as we explained here, but generally, there isn't much room to go higher on large distributed systems.

State-Of-The-Art Training Costs

Optimal LLM Training Cost				
Model	**Size (# Parameters)**	**Tokens**	**GPU**	**Optimal Training Compute Cost**
MosaicML GPT-30B	30 Billion	610 Billion	A100	$ 325,855
Google LaMDA	137 Billion	168 Billion	A100	$ 368,846
Yandex YaLM	100 Billion	300 Billion	A100	$ 480,769
Tsinghua University Zhipu.AI GLM	130 Billion	400 Billion	A100	$ 833,333
Open AI GPT-3	175 Billion	300 Billion	A100	$ 841,346
AI21 Jurassic	178 Billion	300 Billion	A100	$ 855,769
Bloom	176 Billion	366 Billion	A100	$ 1,033,756
DeepMind Gopher	280 Billion	300 Billion	A100	$ 1,346,154
DeepMind Chinchilla	70 Billion	1,400 Billion	A100	$ 1,745,014
MosaicML GPT-70B	70 Billion	1,400 Billion	A100	$ 1,745,014
Nvidia Microsoft MT-NLG	530 Billion	270 Billion	A100	$ 2,293,269
Google PaLM	540 Billion	780 Billion	A100	$ 6,750,000

is table is a theoretical optimal cost to train the mod
ount for the people required



[https://www.semianalysis.com/p/the-ai-brick-wall-a-practical-limit]
[https://www.nvidia.com/en-us/data-center/a100/]

Training Cost and Hardware

The Company & its Products | Bloomberg Terminal Demo Request

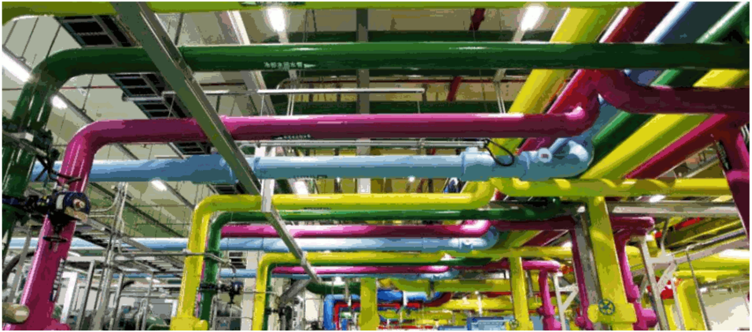
Bloomberg

Subscribe

Green | New Energy

Artificial Intelligence Is Booming—So Is Its Carbon Footprint

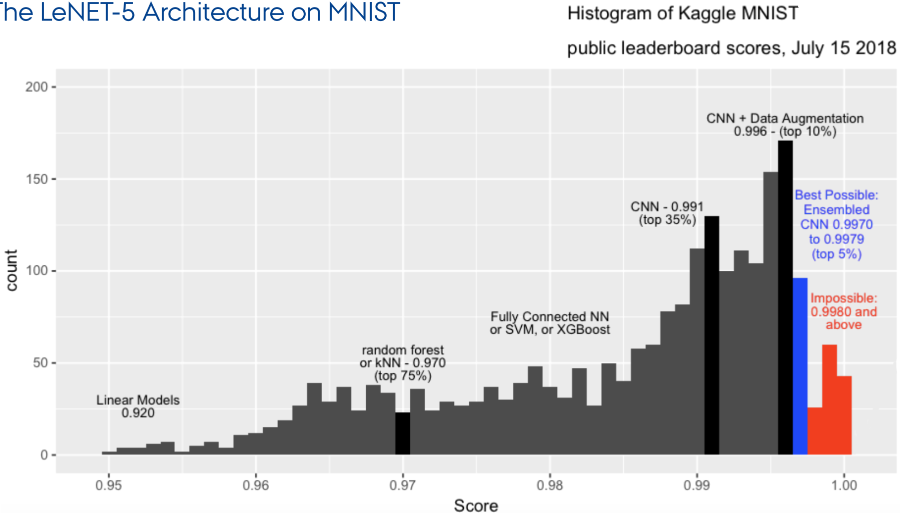
Greater transparency on emissions could also bring more scrutiny



https://www.bloomberg.com/news/articles/2023-03-09/
how-much-energy-do-ai-and-chatgpt-use-no-one-knows-for-sure?leadSource=uverify%20wall

Qd MNIST Search Quest II

The LeNET-5 Architecture on MNIST



Qd MNIST Search Quest II

Fra tidligere semester

```
F2024:  Grp25:  score=0.972, RandomForestClassifier
E2023:  Grp13:  score=0.984, SVC
F2023:  Grp01:  score=0.981, SVC
E2022:  Grp05:  score=0.983, SVC
F2022:  Grp05:  score=0.990, SGDClassifier (iris?)
E2021:  Grp28:  score=0.973, KNN
F2021:  Grp20:  score=0.979, SVC
```

Qd MNIST Search Quest II

... => L11, and cake!