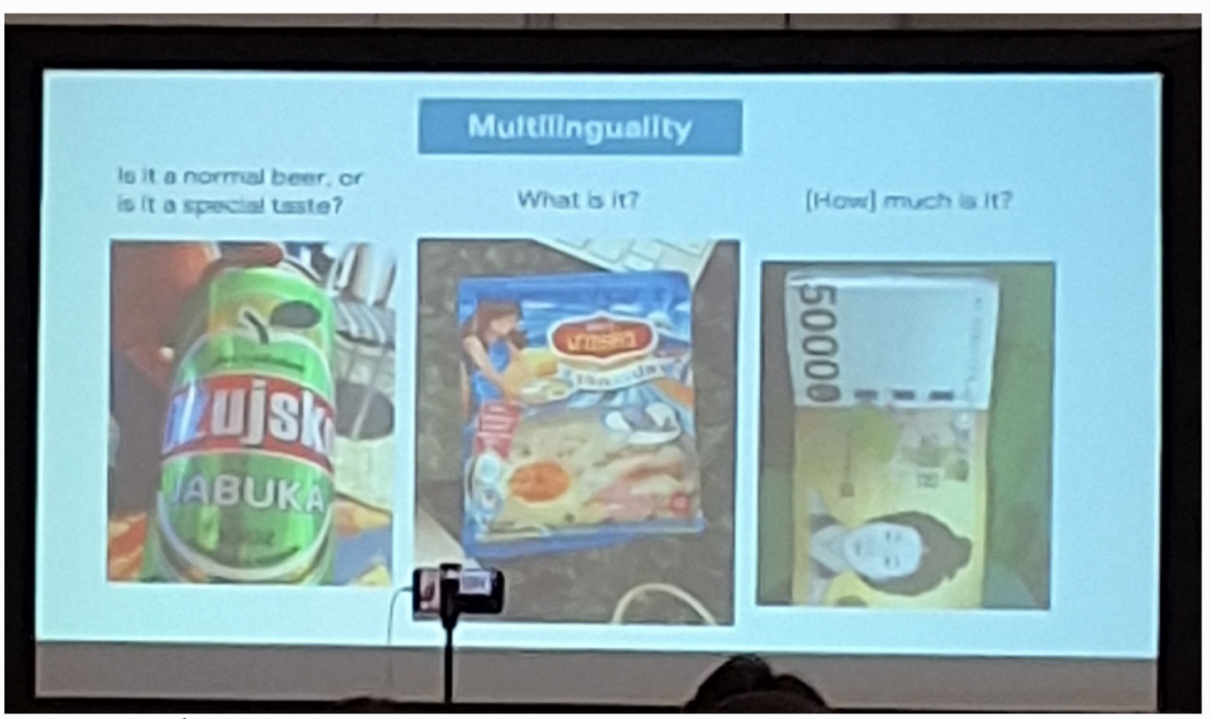
[Viz Wiz]

Speaker 1 - google speaker

- split OCR
  - PreSTU
  - PALI, PALI-X
  - VQ²A data set



↳ looks like Viz Wiz got rash ranting?

↳ Bunch of caption / VQA dataset

OpenAI Speaker

- see phone for slide pics

Main useful takeaways

↳ using a bunch of smaller models not as good as a big generalist

↳ when low vs high res, they default to high res so users no need to think about it
they do it by cropping input into different parts & using more tokens

↳ doesn't explicitly say is video or image based. they say both the same lol

Challenge Winner talk

- was told it will be recorded & uploaded to youtube

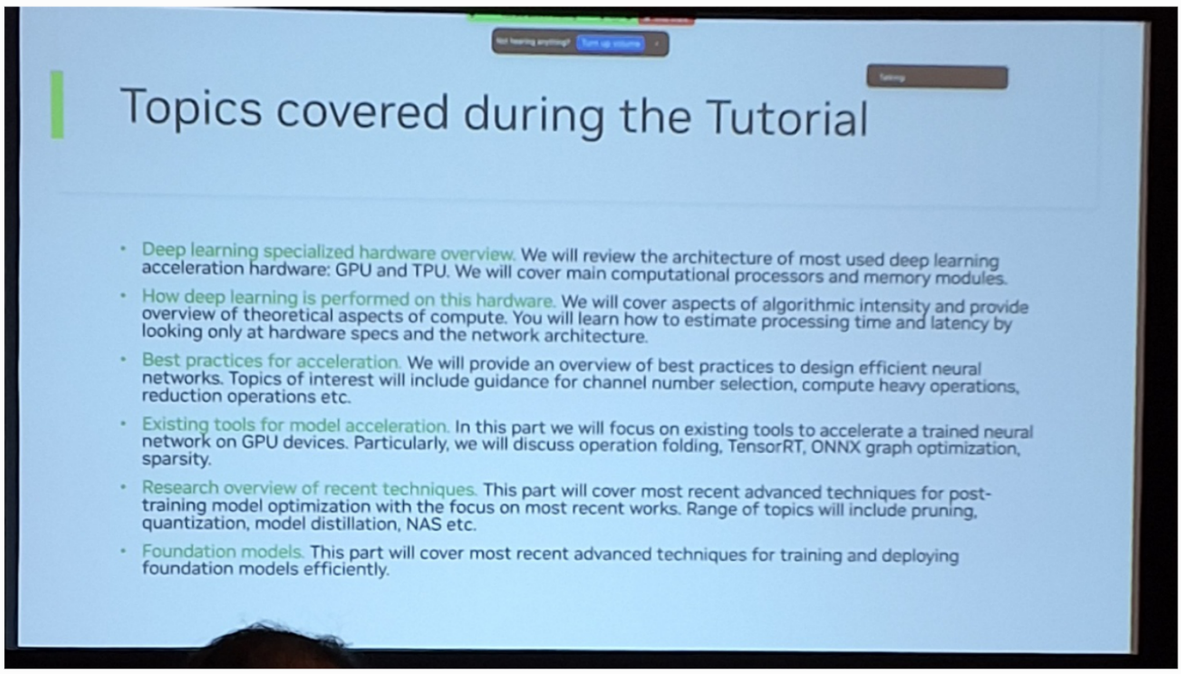
Posters

[illegible] GPU based acceleration [illegible]

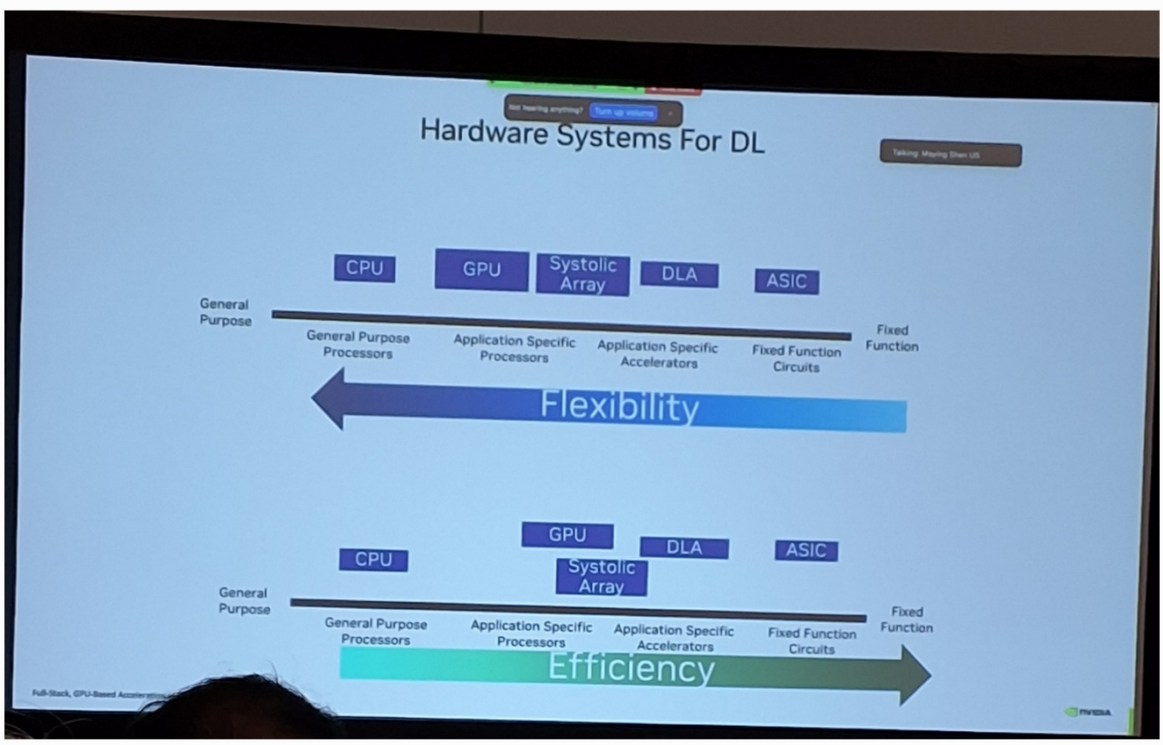
[Full stack, GPU based acceleration of DL]

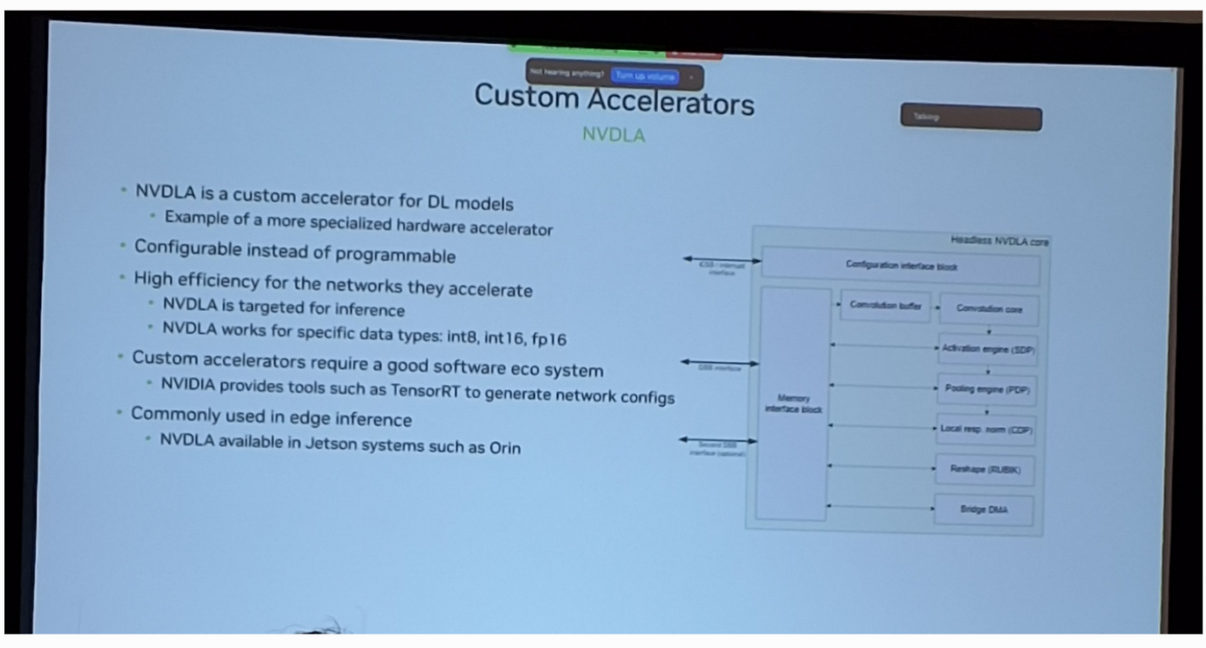
https://github.com/NVlabs/EfficientDL

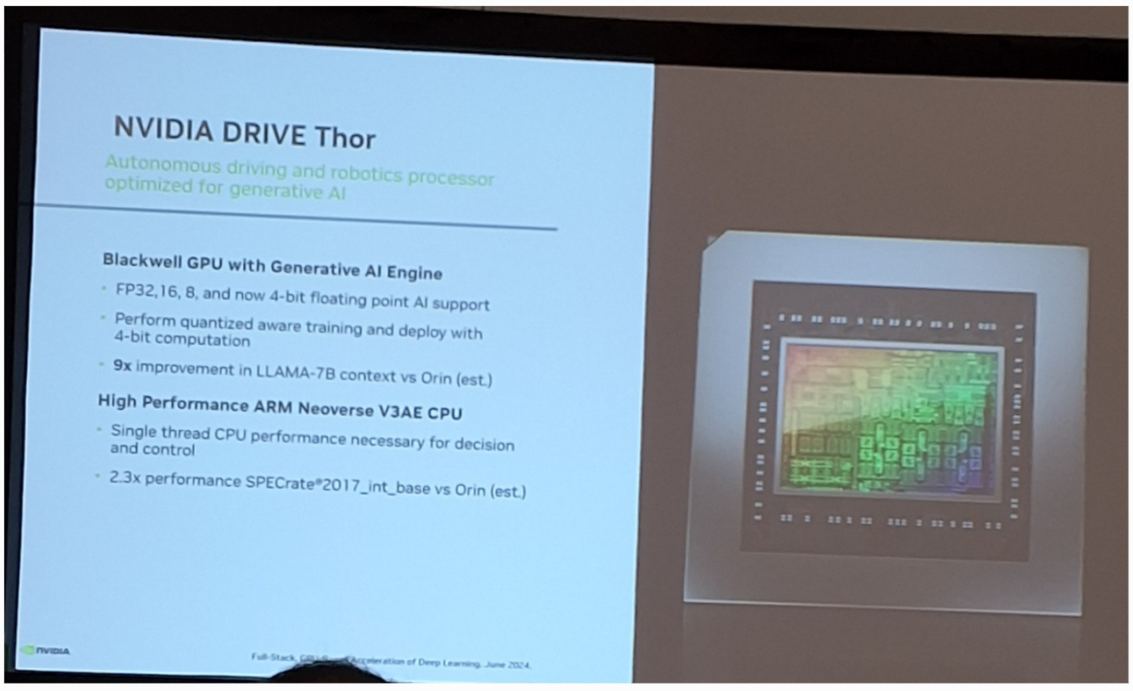
- slides
- videos recording



Foundations of DL hardware & how to apply

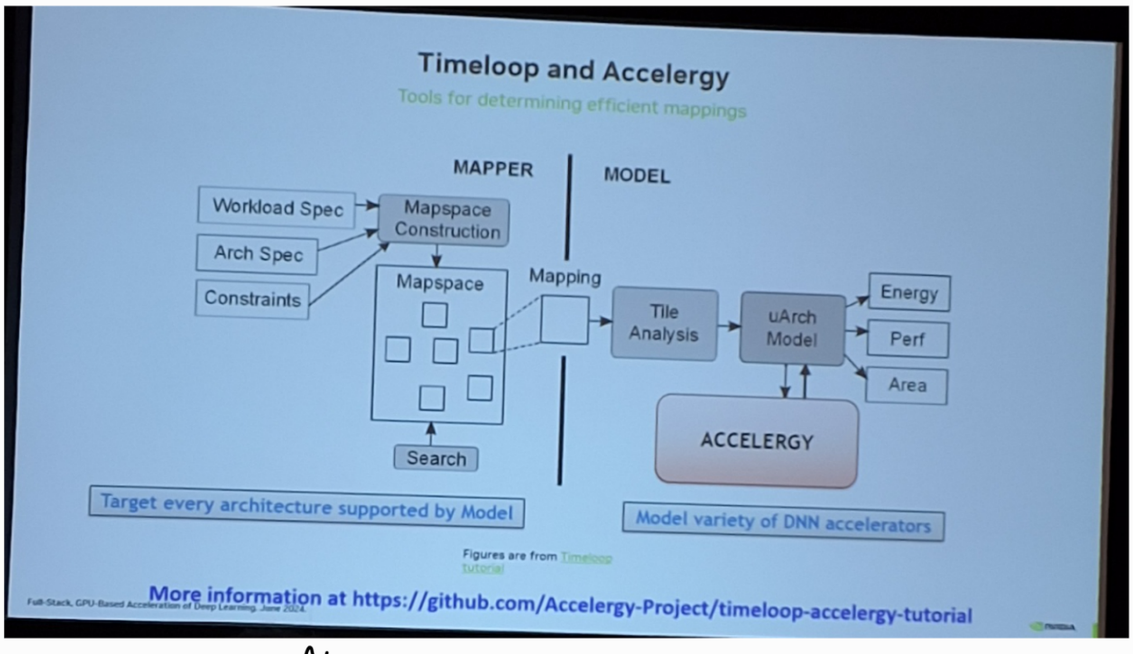






↳ Next version of orin

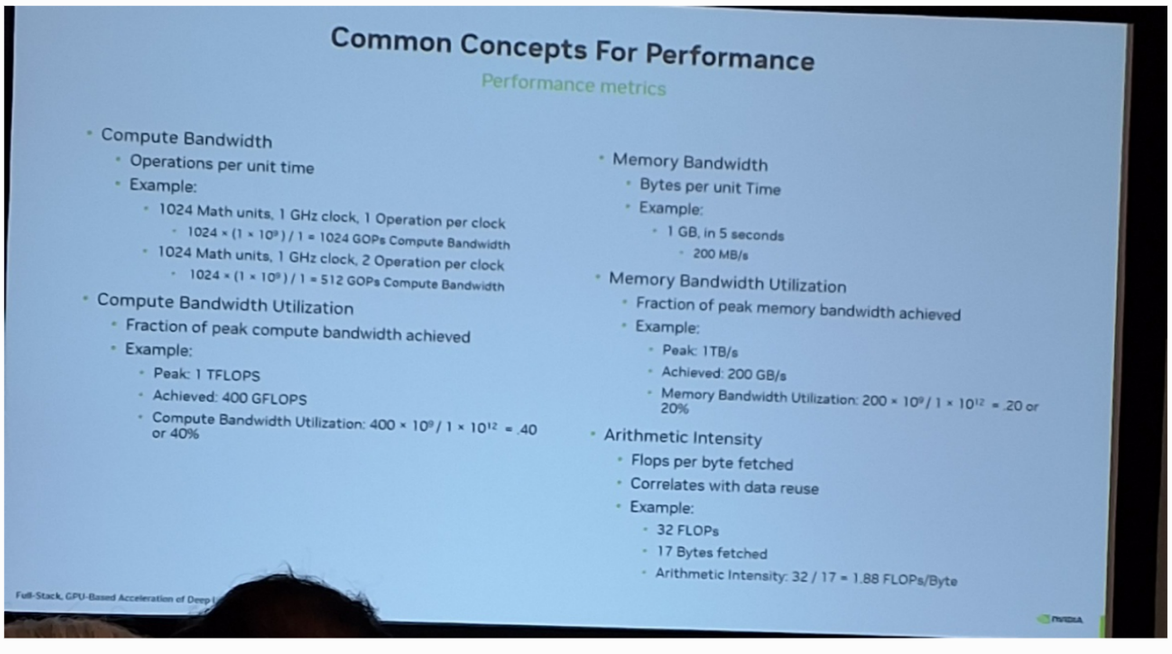
- Mapping



↳ Mapping software to hardware

↳ If want to really squeeze out every performance

↳ Quite low level



Estimating Performance

Bounding the performance

- Compute the FLOPs in the layers
  - 3×3 Conv2d with bias, Input is 1×3×416×480, Output is 1×16×208×240
    - FLOPs = ((3×3) × 3 +1) ×16 × 208 × 240 * 2= 22364160 * 2 = 44,728,320
    - FP32 Input tensor: 3×416×480*4 = 2,396,160
    - FP32 Output tensor: 16×208×240*4 = 3,194,880
    - FP32 Parameters: (3×3×3) × 16 + 16 = 448
  - Tools like torchinfo give can give you this
- Use the system peak compute and memory bandwidth to estimate performance
  - Assume peak compute bandwidth of 1 TOPS (1×10^{12} OPs)
    - Best case: 44,728,320/ 1×10^{12} = 0.00004472832 s = 44.8 ns compute
  - Assume peak memory bandwidth of 1000 GB/s
    - Best case: 559104 / 1000×10^{9} = 0.00000559104 s =5.6 ns memory transfer
- Realistically you assume some derating factor but now at least it is bounded

Note:
Some counters might not calculate ops not in their framework.

Understanding Performance

- Operations can be *memory-limited* and *math-limited*
  - Math limited – High utilization of the compute units while the memory bandwidth utilization is not high
    - Math limited if: (BWmath / PeakBWmath) ~> .7
  - Memory limited – High utilization of the memory bandwidth while the compute unit utilization is not high
    - Mem limited if: (BWmem / PeakBWmem) ~> .7
- Using Arithemtic intensity for estimating limitations
  - Math limited if: #ops / #bytes > PeakBWmath / PeakBWmem

Table 1. Examples of neural network operations with their arithmetic intensities. Limiters assume FP16 data and an NVIDIA V100 GPU.

Operation	Arithmetic Intensity	Usually limited by...
Linear layer (4096 outputs, 1024 inputs, batch size 512)	315 FLOPS/B	arithmetic
Linear layer (4096 outputs, 1024 inputs, batch size 1)	1 FLOPS/B	memory
Max pooling with 3x3 window and unit stride	2.25 FLOPS/B	memory
ReLU activation	0.25 FLOPS/B	memory
Layer normalization	< 10 FLOPS/B	memory

B – byte

b - bit

— How to check memory & GPU utilisation?

Recommendations

1. Operating In Math-Limited Regime Where Possible:

   If the speed of a routine is limited by calculation rate (**math-limited** or **math-bound**), performance can be improved by enabling Tensor Cores.

2. Using Tensor Cores Efficiently With Alignment:

   Use parameter shapes such that compute is aligned with Tensor Cores characteristics to minimize wave tails.

3. Choosing Parameters To Maximize Execution Efficiency:

   GPUs perform operations efficiently by dividing the work between many parallel processes

*Slide from Pavlo Mo

use the correct image shape

- use the correct numbers (×8)
- Mix precision
- check the slide talks about checking tensor cores utilisation
- check the hardware!

Neural Net Acceleration