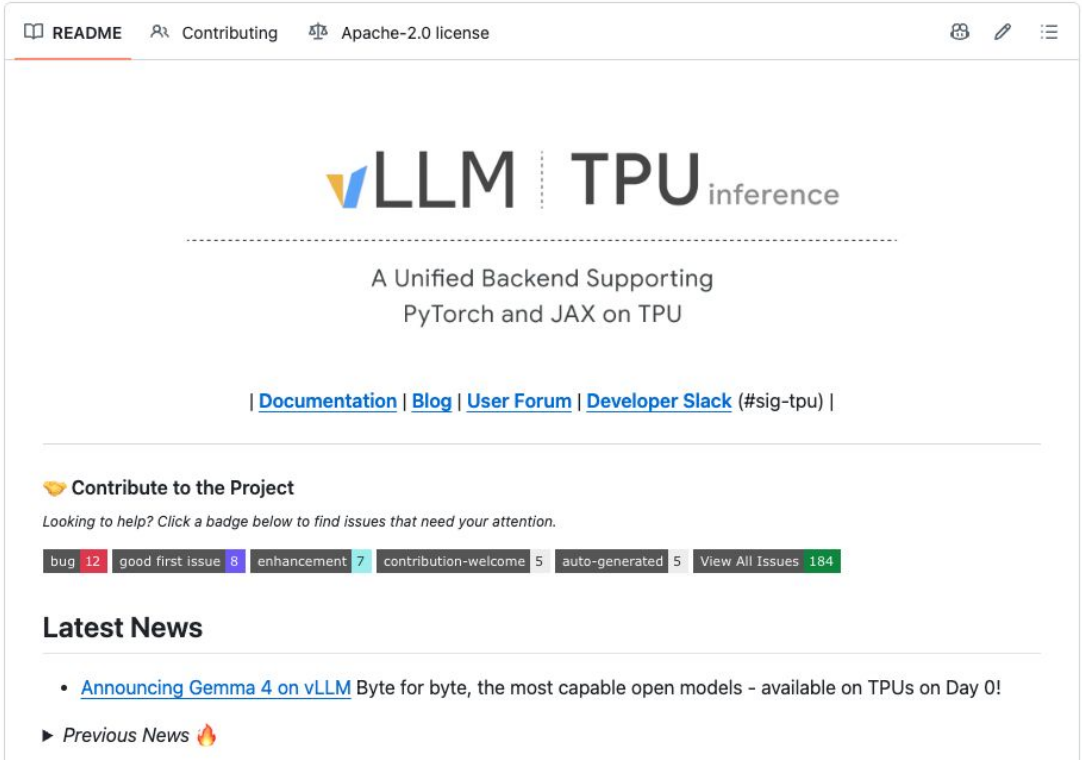
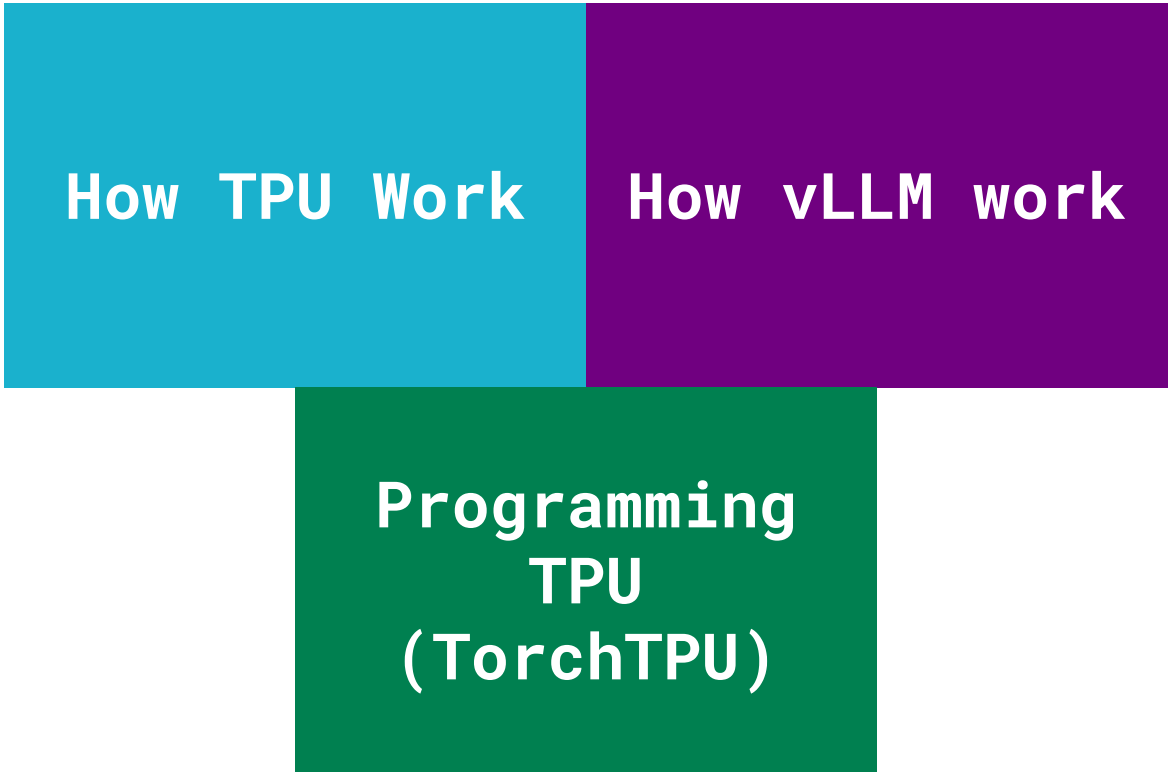
Recap

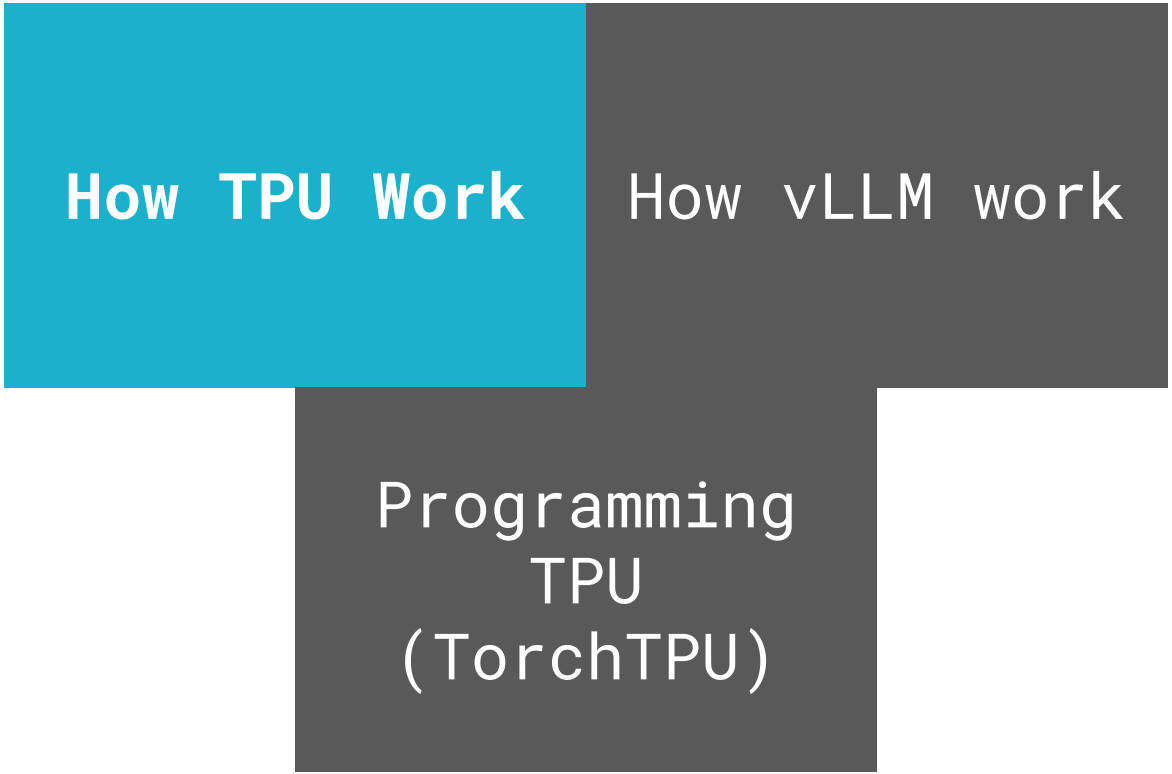
Accelerating Large Language Models: vLLM on TPUs

vllm-project/tpu-inference



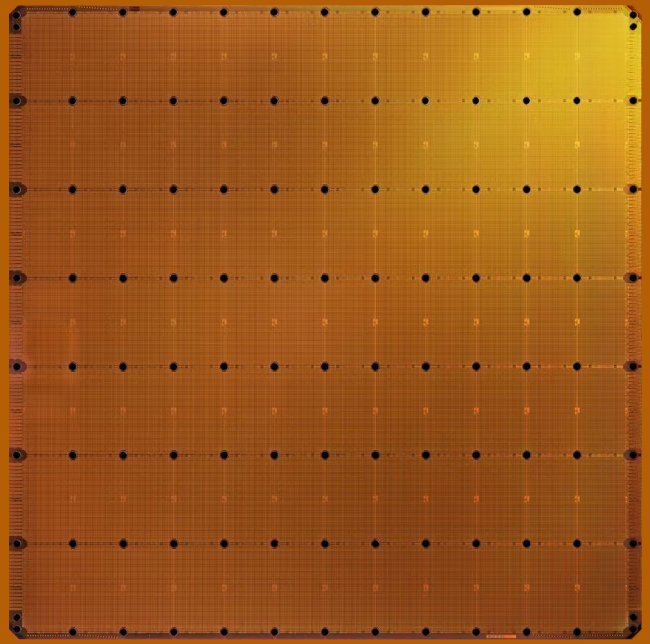
To get the most out of talk!




How TPU Work
How vLLM work
Programming
TPU
(TorchTPU)

How many AI accelerators (e.g. TPU, GPU) are there in the market?

Not TPU >>



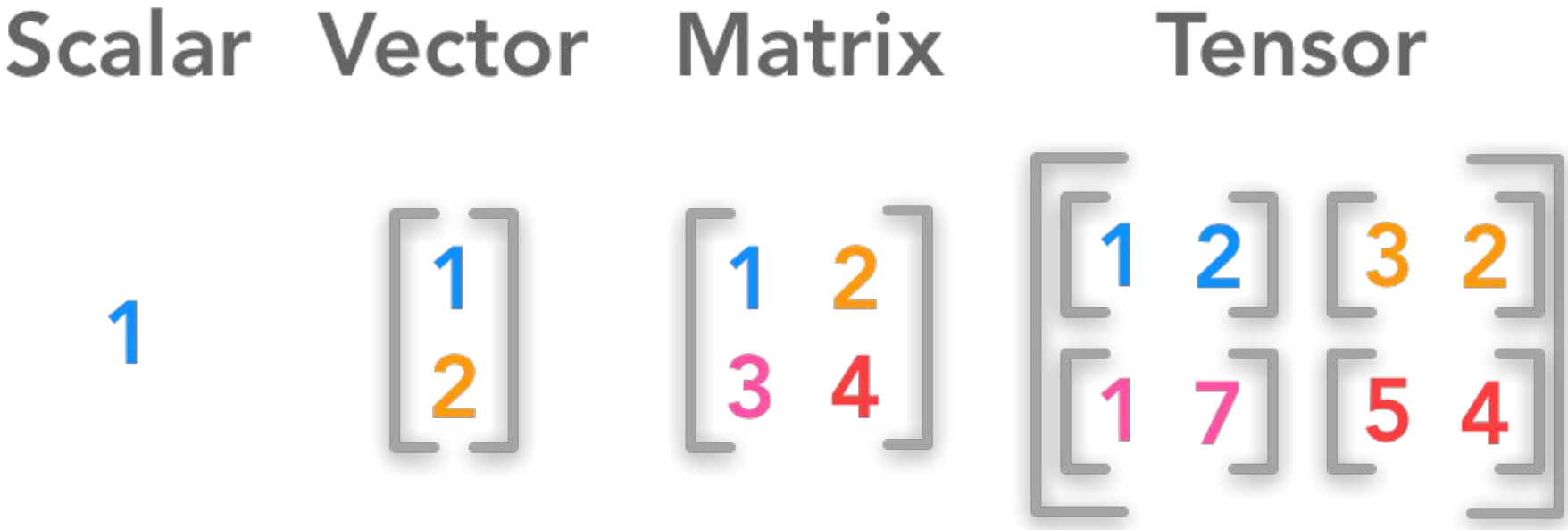
Cerebras WSE-3 Turbo
(WSE-3T)



Nvidia GPU
B300

Over 50+ startup are building AI Accelerator (i.e. hardware/chip)

New Market Pitch. (2026, October 10). *Top AI inference chip startups by valuation (2026)*. https://newmarketpitch.com/blogs/news/ai-inference-chips-top-startups-valuation



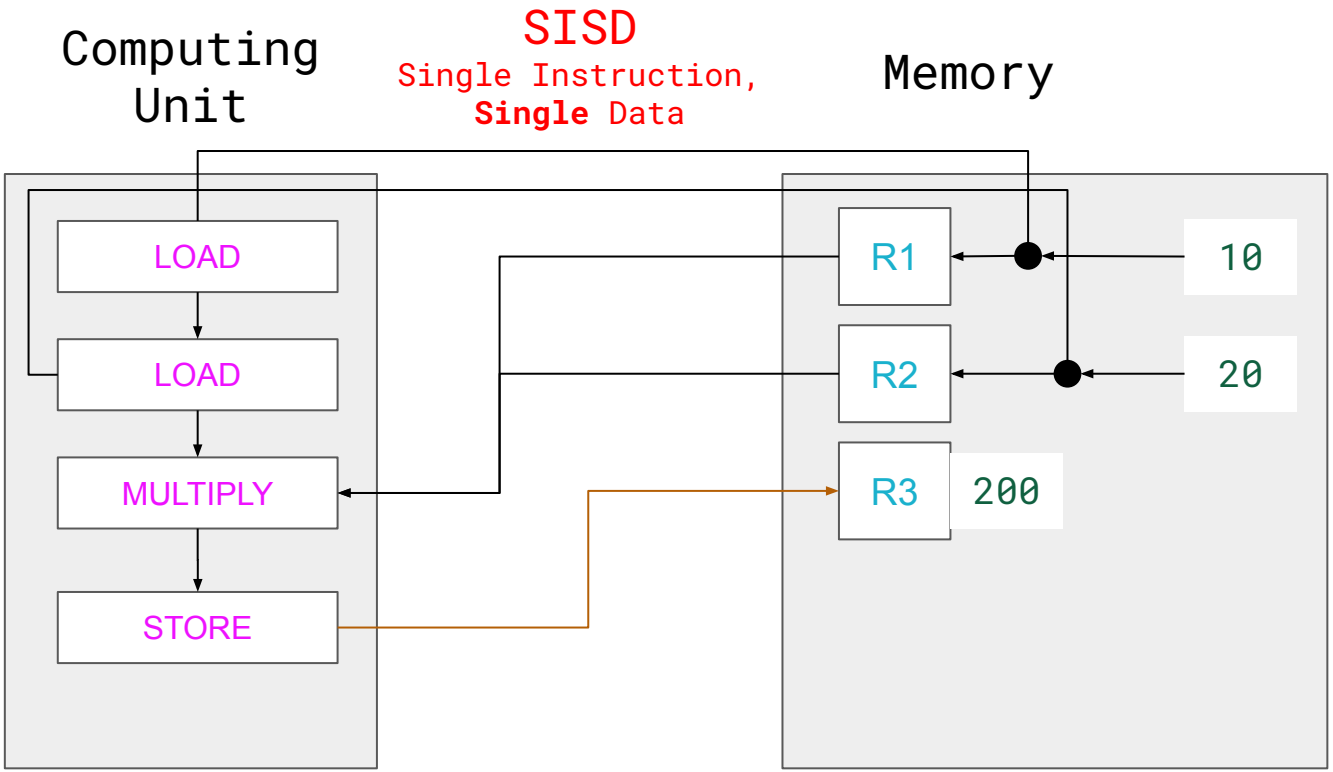
Jean, H. (2018, March 26). *Deep Learning Book Series · 2.1 Scalars, vectors, matrices and tensors*. Code · Data Science. https://hadrienj.github.io/posts/Deep-Learning-Book-Series-2.1-Scalars-Vectors-Matrices-and-Tensors/

```
int oprd_1 = 10;

int oprd_2 = 20;

int result = oprd_1 * oprd_2;
```

How CPU compute result (simplified)



R = Register

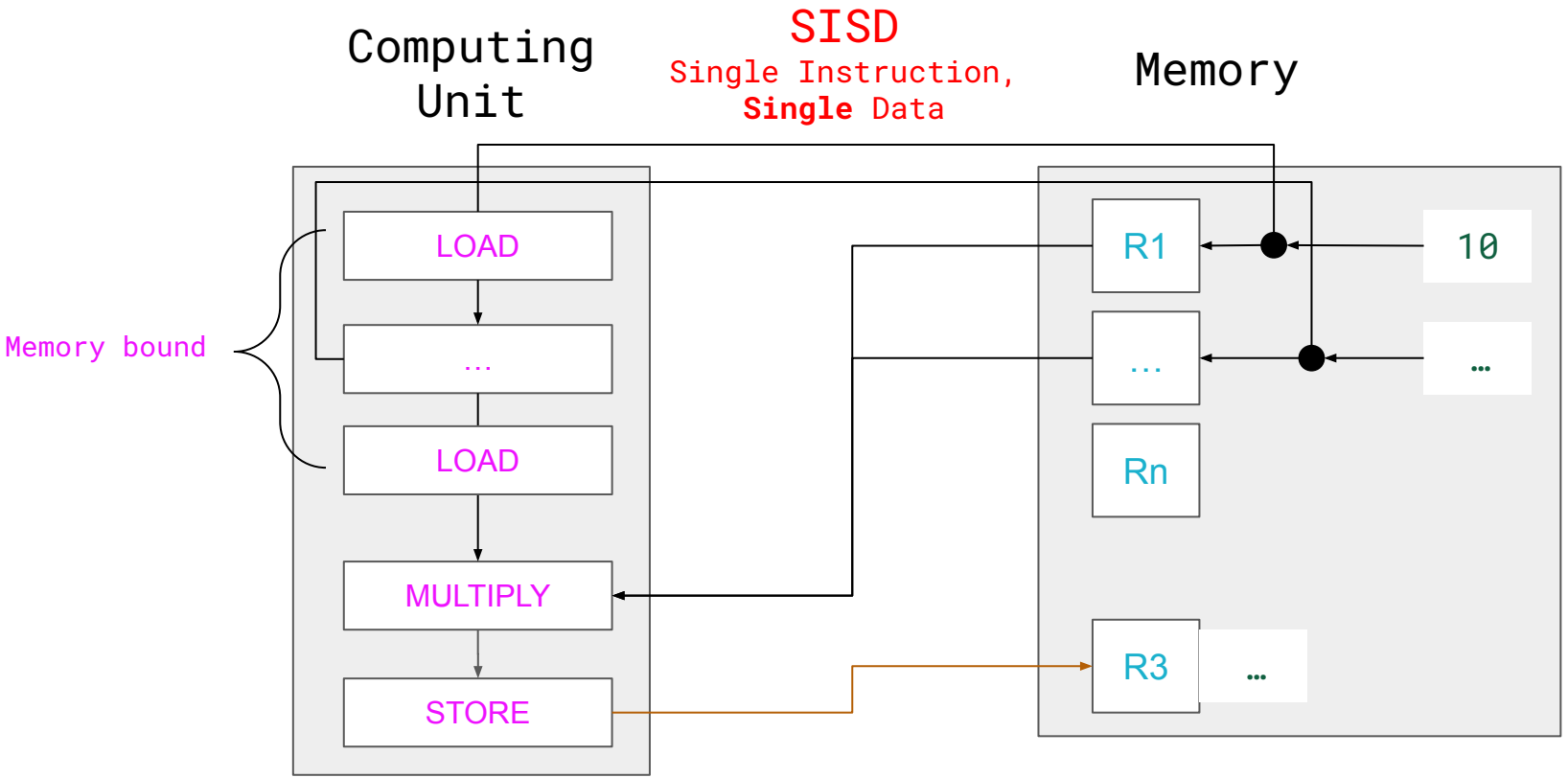
```
int oprd_1 = [10, 20, 30, 40, 50];

int oprd_2 = 10;

int result = vecMulByScr(oprd_1, oprd_2);
```

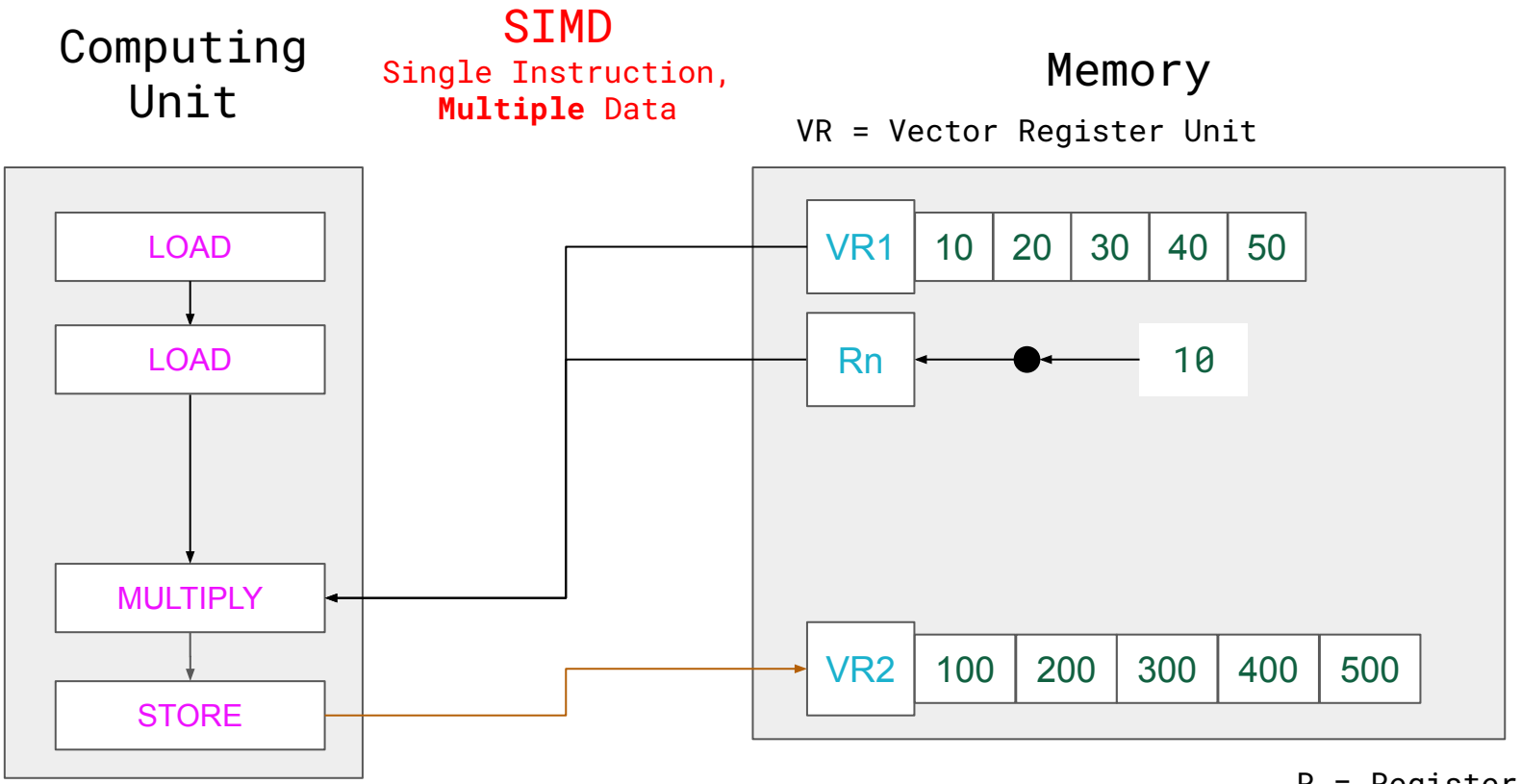
```
int vecMulByScr(int[]* vector, int scalar){
     for(int index = 0; index < vector.length; index++) {
          vector[index] = vector[index] * scalar;
     }
}
```

How CPU compute result (simplified)



R = Register

How CPU compute result (simplified)

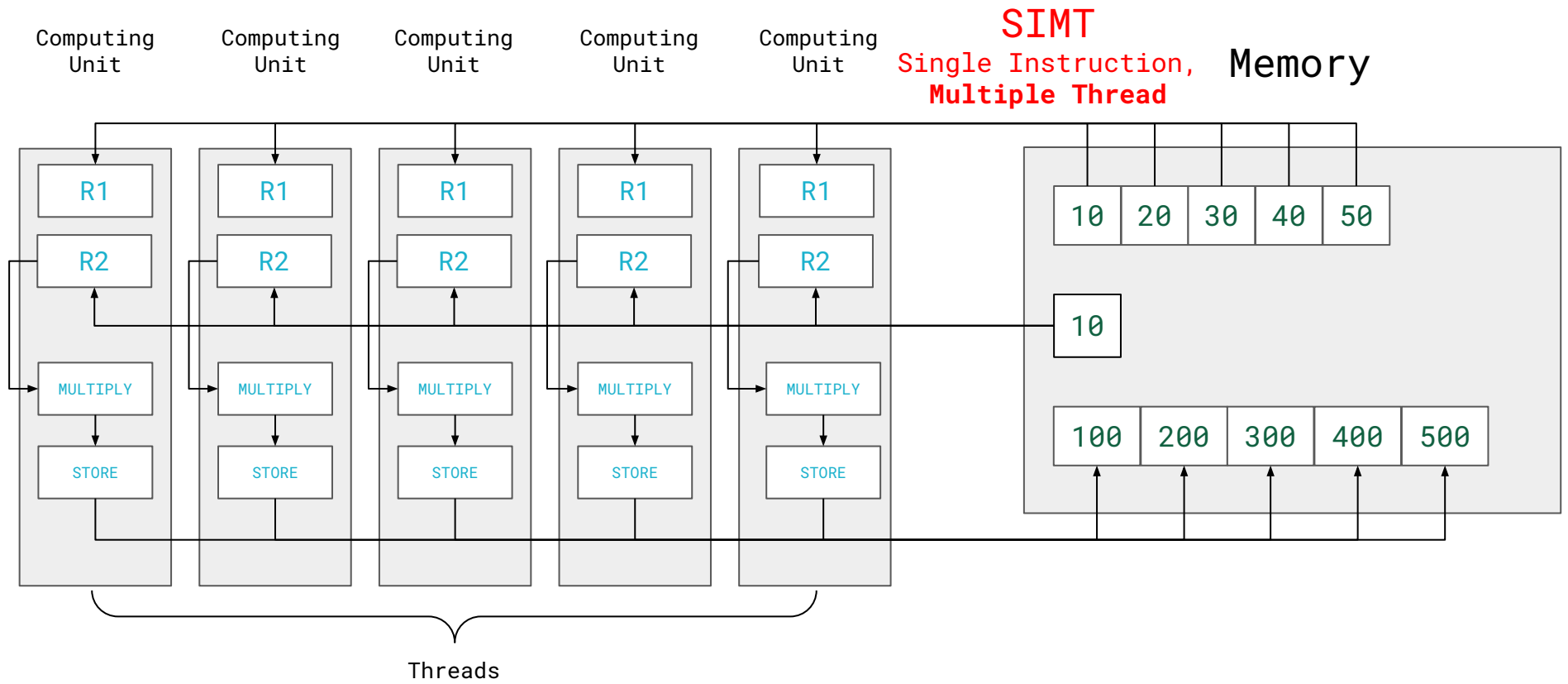


```
int oprd_1 = [10, 20, 30, 40, 50];

int oprd_2 = 10;

int result = vecMulByScr(oprd_1, oprd_2);
```

How GPU compute result (simplified)



```
int oprd_1 = [10, 20, 30, 40, 50];

int oprd_2 = 10;

int result = vecMulByScr(oprd_1, oprd_2);
```

```
int vecMulByScr(int[]* vector, int scalar){
     int xBlock = blockIdx.x * blockDim.x + threadIdx.x;
     if(xBlock < vector.length){
          vector[xBlock] = vector[xBlock] * scalar;
     }
}
```

Scalar

Vector, Matrix, Tensor

Single Instruction,
Single Data
SISD

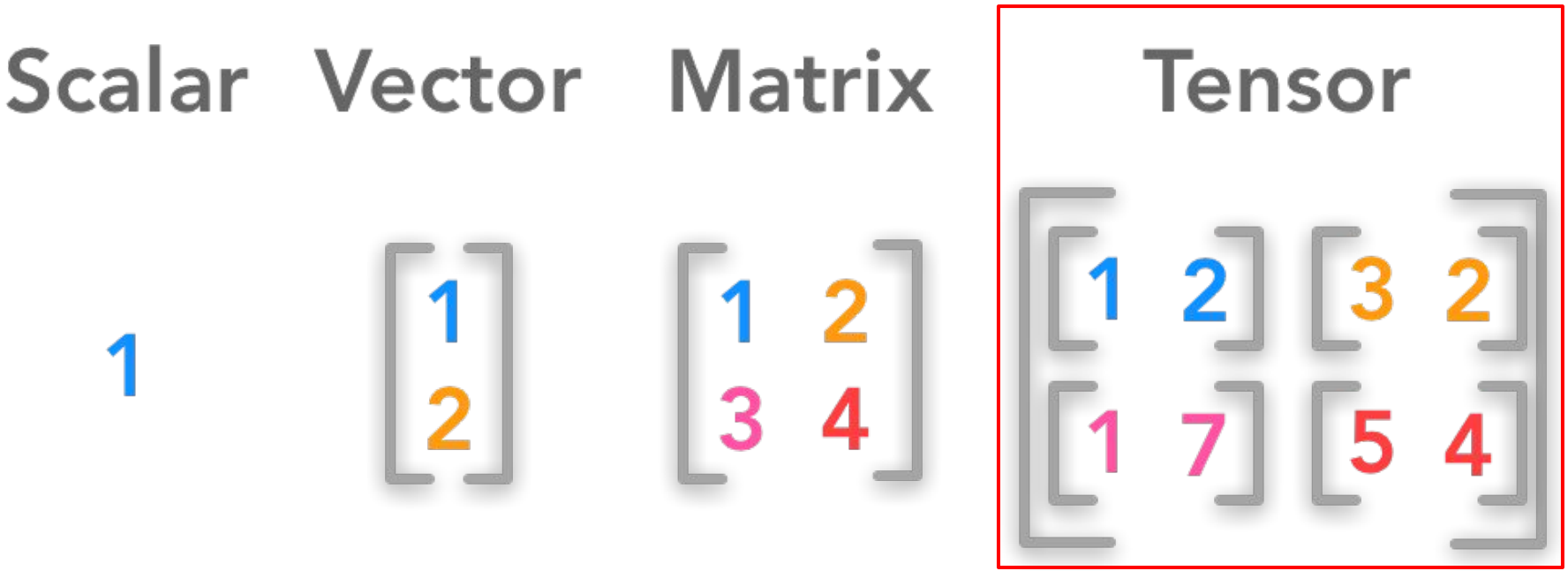
CPU

Single Instruction,
Multiple Data
SIMD

CPU
+ Vector Register

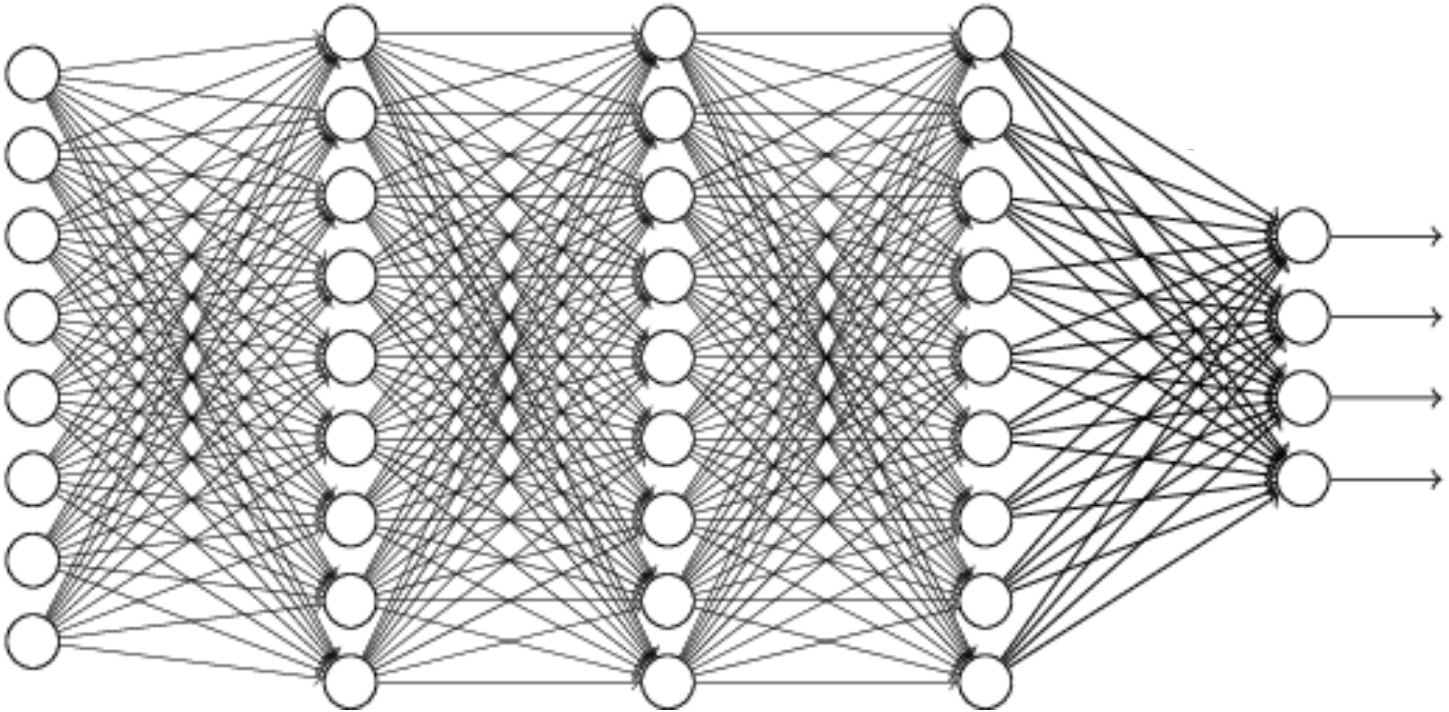
Single Instruction,
Multiple Threat
SIMT

GPU



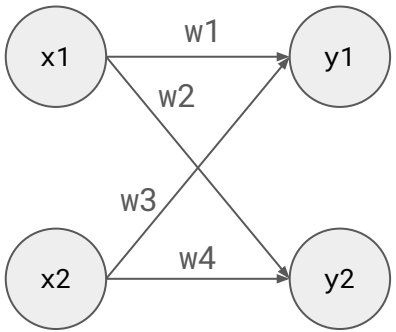
Jean, H. (2018, March 26). *Deep Learning Book Series · 2.1 Scalars, vectors, matrices and tensors*. Code · Data Science. https://hadrienj.github.io/posts/Deep-Learning-Book-Series-2.1-Scalars-Vectors-Matrices-and-Tensors/

Large Tensor



Nielsen, M. A. (2015). Why are deep neural networks hard to train? In Neural networks and deep learning (Chap. 5). Determination Press. http://neuralnetworksanddeeplearning.com/chap5.html

Neural Network (simplified)

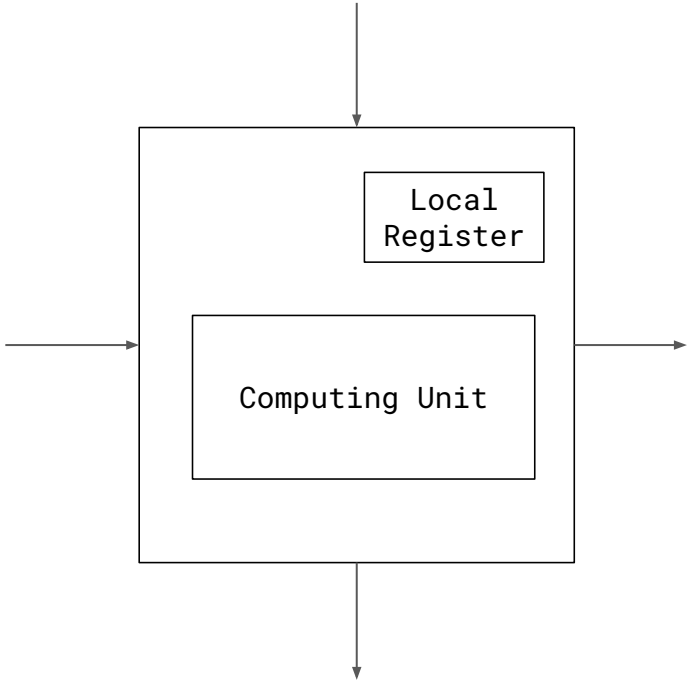


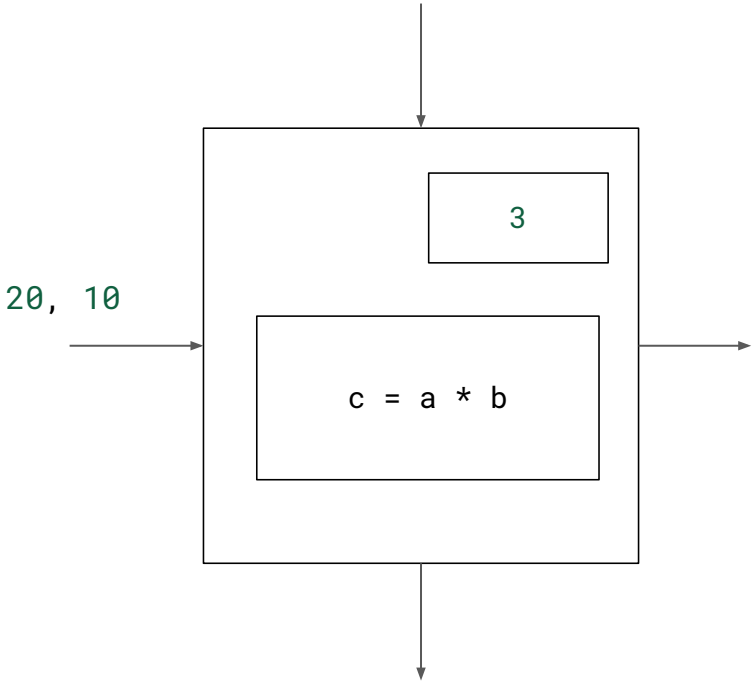

```
# Input
X = np.array(
      [
            [0.5, 1.2],
            [1.0, -0.7],
            [0.2, 0.4],
      ]
)

# Weight
W = np.array(
      [
            [0.1, -0.4, 0.5],
            [0.6, 0.2, -0.2],
      ]
)
```

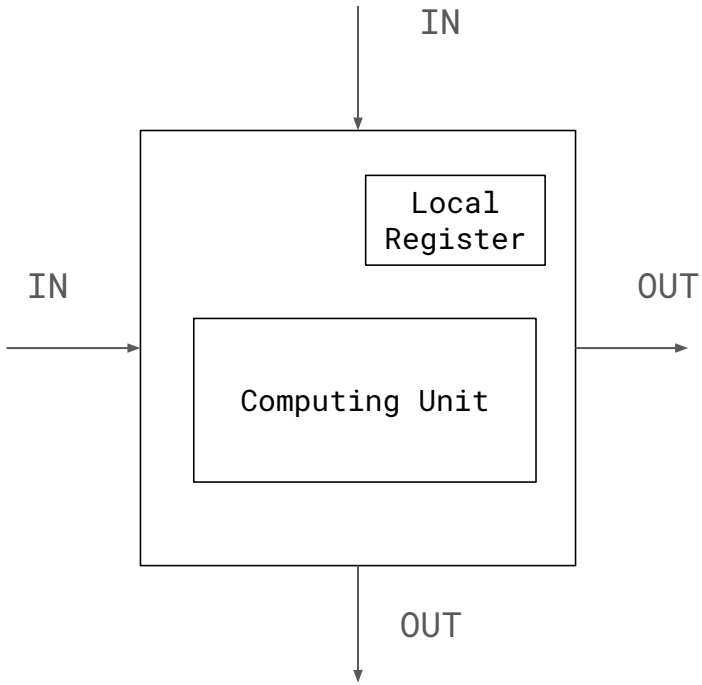
Systolic Array: Processing Element (PE)

```
a = [10, 20]
b = 3
```



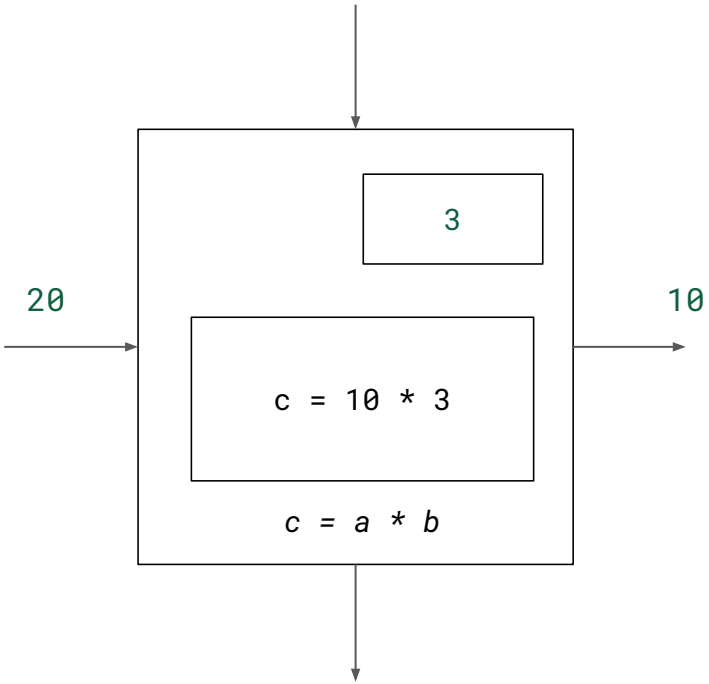


Systolic Array: Processing Element (PE)



Systolic Array: Processing Element (PE)

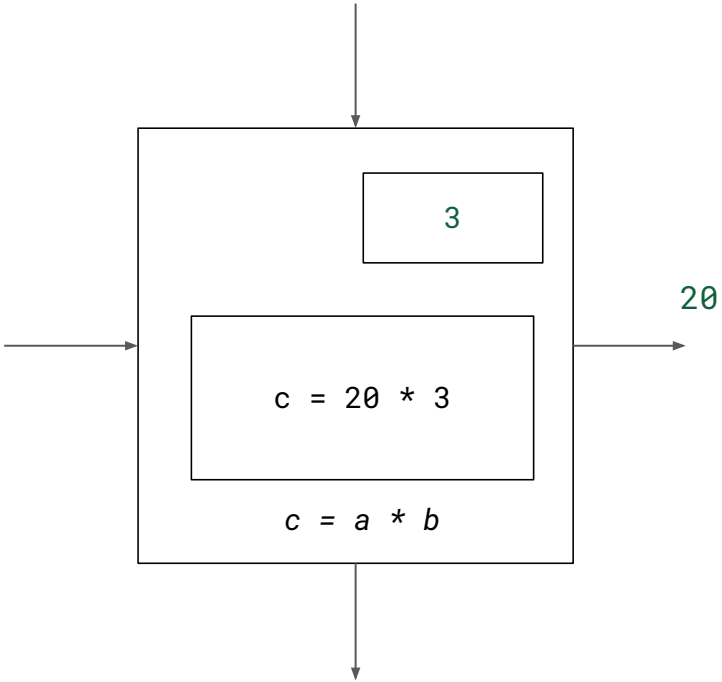
```
a = [10, 20]
b = 3
```




```
c = [30, ..]
```

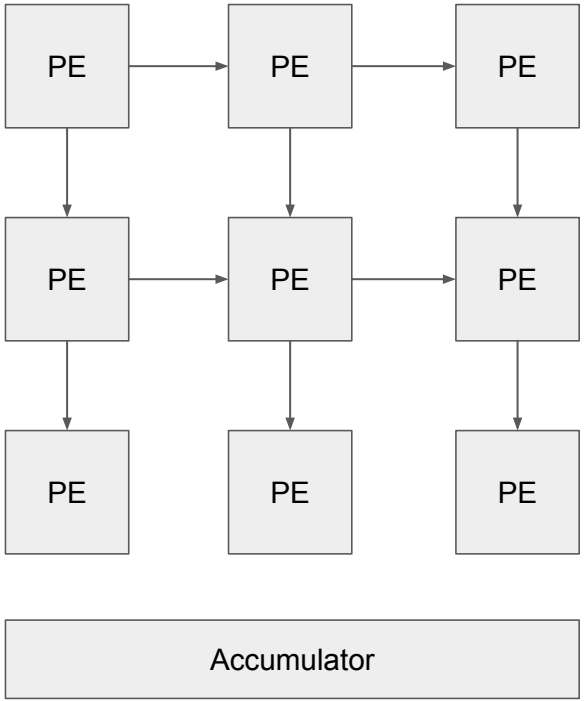
Systolic Array: Processing Element (PE)

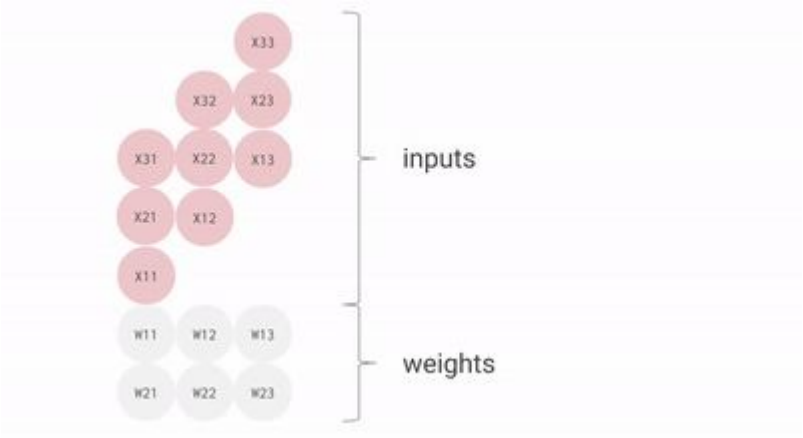
```
a = [10, 20]
b = 3
```



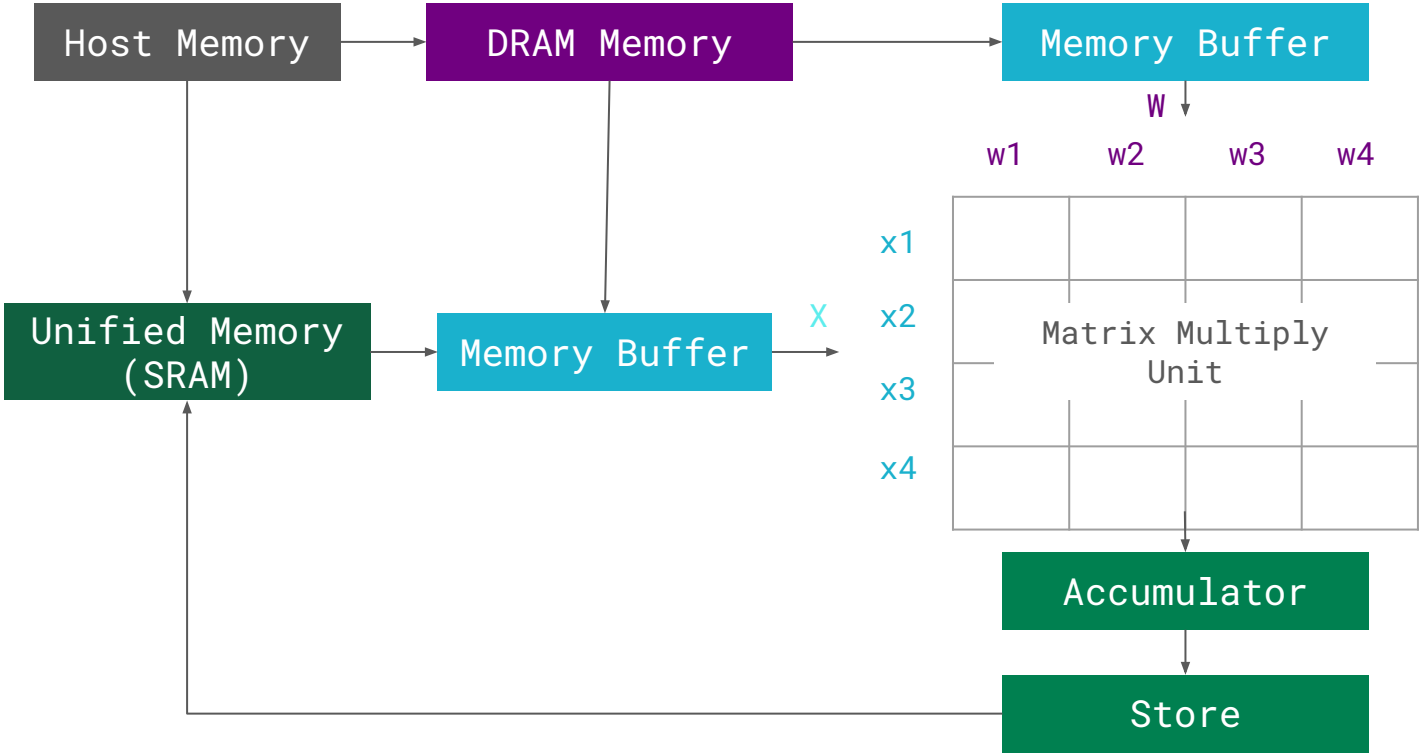

```
c = [30, 60]
```

Systolic Array: Matrix Multiply Unit (MXU)

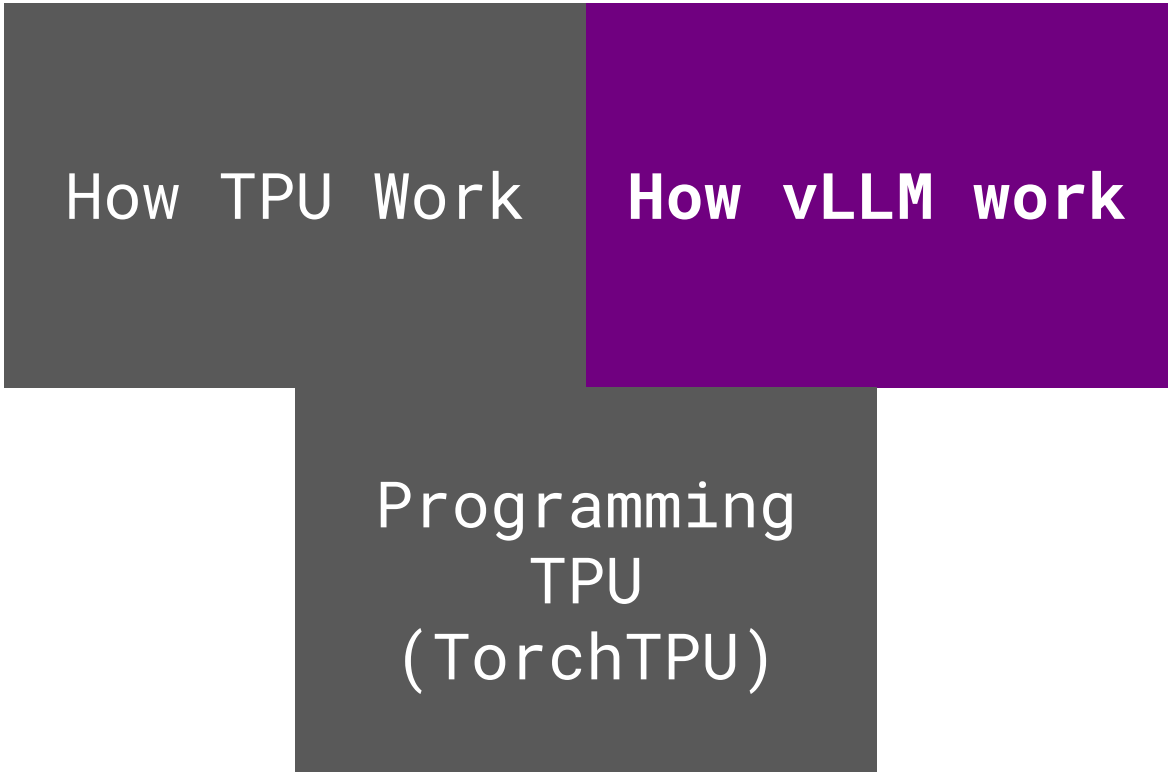




How TPU compute result (simplified)



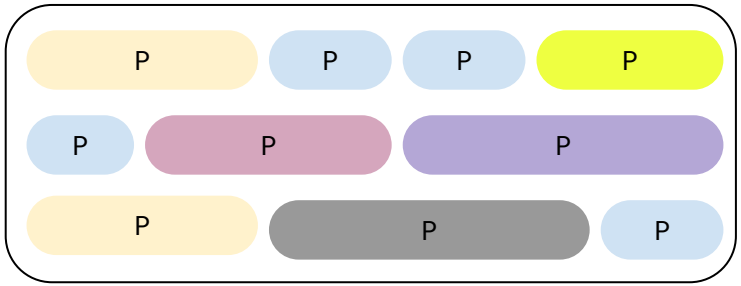
Sato, K., Young, C., & Patterson, D. (2017, May 12). An in-depth look at Google's first Tensor Processing Unit (TPU). *Google Cloud Blog*. https://cloud.google.com/blog/products/ai-machine-learning/an-in-depth-look-at-googles-first-tensor-processing-unit-tpu


How TPU Work
How vLLM work
Programming
TPU
(TorchTPU)

vLLM

utilize

~100% of GPU Computing



Continuous Batching

small proportion of

Memory Fragmentation

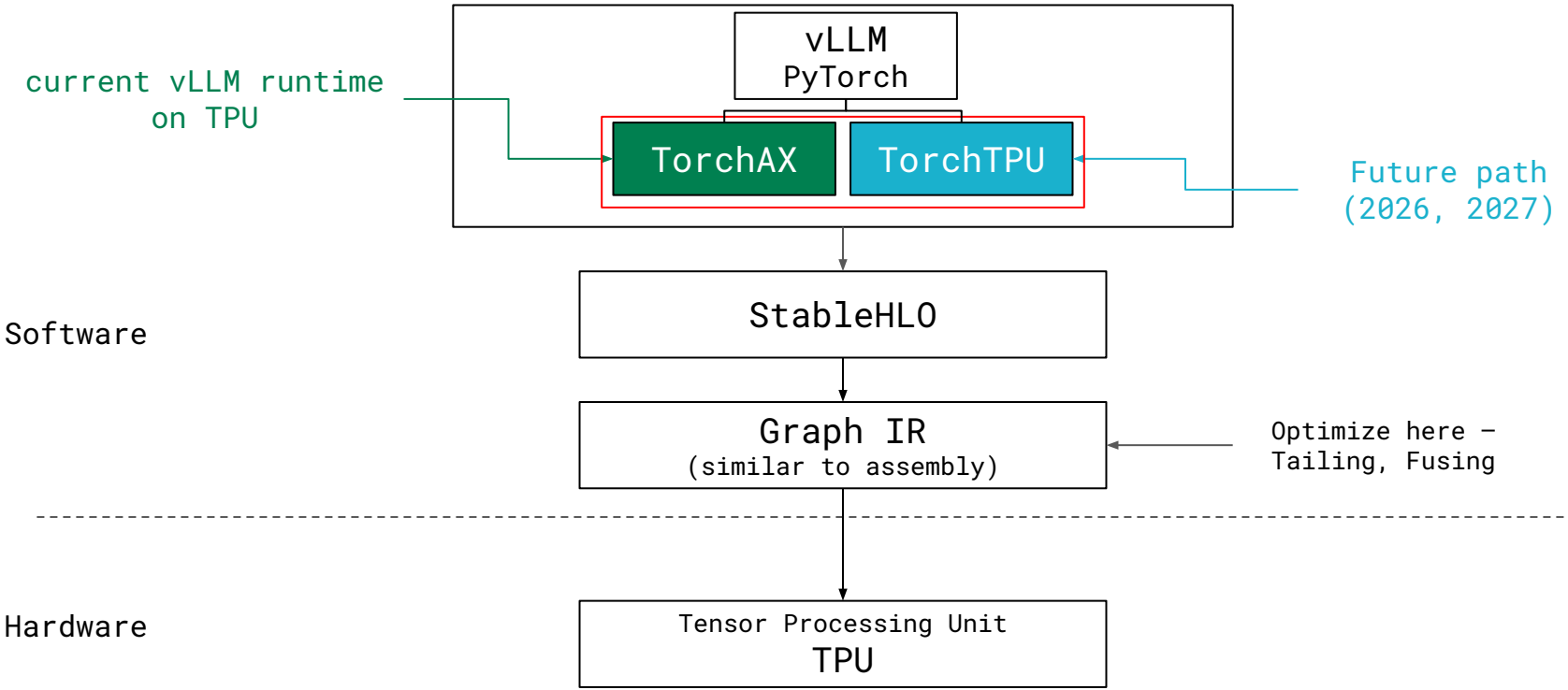


PagedAttention

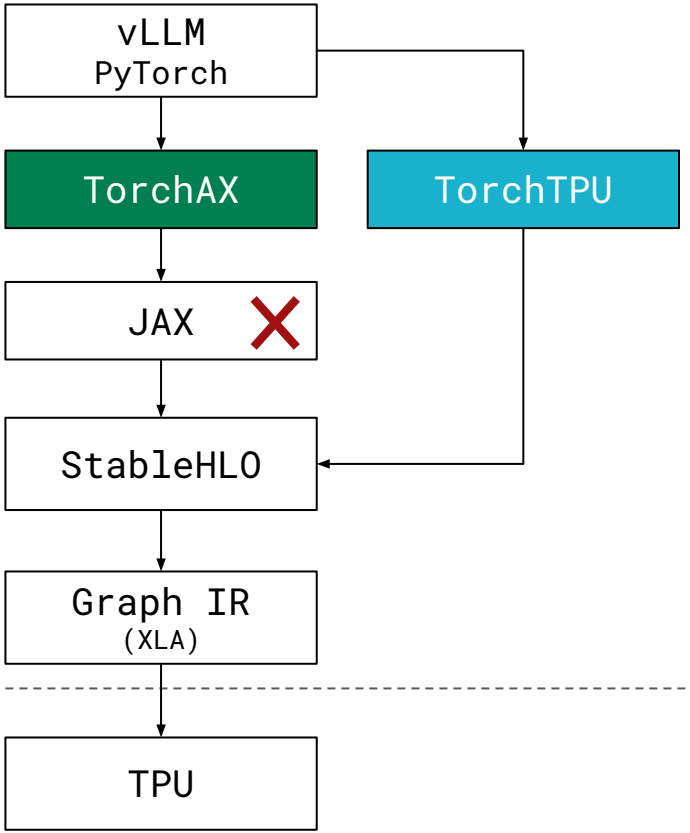
How TPU Work How vLLM work

Programming TPU (TorchTPU)

Programming TPU (Simplified)



Programming TPU (Simplified)




```python
a = torch.randn(
    (2, 2),
    device='tpu'
)
```

- Faster compilation
- PyTorch Native
- Speed enhancement (case by case)

Qi, H., Chan, C.-n., Cromar, W., Bai, M., & Gleason, K. (2025). *torchax: PyTorch on TPU and JAX interoperability* (Version 0.0.4) [Computer software]. Google. https://github.com/google/torchax

Key Takeaways

- vLLM can run on TPU (for very long time)
- No silver bullet for accelerators (GPU, TPU, CPU)
  - We are approaching heterogeneous
- TorchTPU* ~~is coming~~ has come for vLLM on TPU.

*official document was released in PyTorch conference, 21 Oct (3 days ago)

Quick Start: vLLM with TPU

