the tiny corp

We will commoditize the petaflop.

George Hotz (tinygrad.org)

Green team vs Red team

- A100
  - 312 BF16 TFLOPS
  - 40 GB
  - 2,039 GB/s
  - 400W
  - $10,450 (at minimum)
- 10x 7900XTX
  - 1230 BF16 TFLOPS
  - 240 GB
  - 9,600 GB/s
  - 3550W
  - $9,990 (on Amazon)

7900XTX is a 3x+ better buy

(even when accounting for machine cost and TCO over 3 years)

Problem: FLOPS aren't a commodity

- red team FLOPS =/= green team FLOPS
- PyTorch is difficult to add accelerators to. PyTorch 2.0 bragged about moving from 2000+ operators to a minimum set of 250. 250 is still so many, and you still need more to go fast.
- By being the dominant player, green team gets a whole ecosystem of developers to improve the developer experience for them.
- Crypto miners can be specified with two numbers, hash/W and hash/$. They are a commodity. We aren't there (yet) with machine learning.

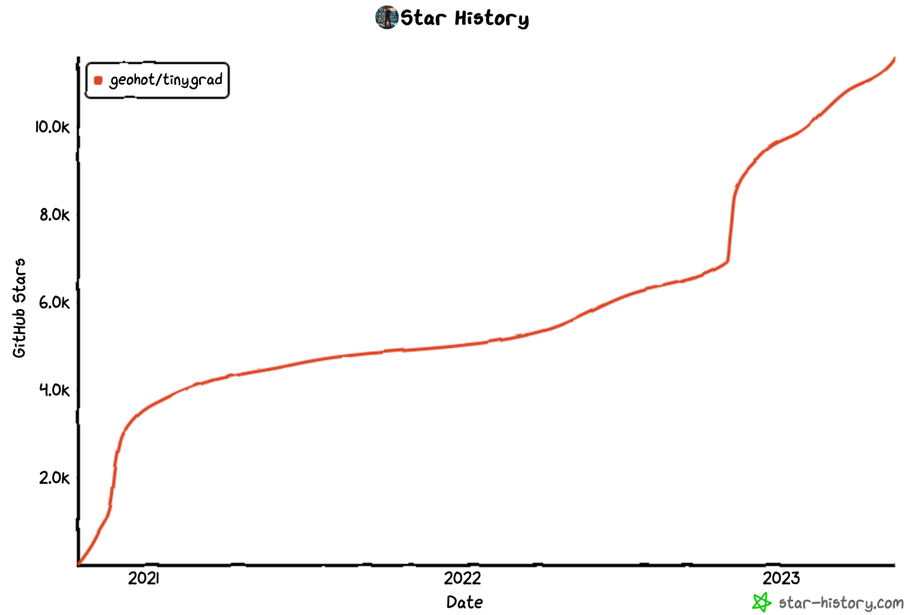
Fake Solution: Startups making chips

- If you cannot write a compelling software stack for x86/NVIDIA/AMD/M1, there’s no way you can write it for your chips.
- Your chips are worse. The first generation of your chips are definitely worse. Why did you tape out a chip before you had software for it?
- The only remotely successful AI training chip other than NVIDIA is the Google TPU. They wrote software (TensorFlow) before they built chips.

"Sequoia Capital to slash the value of its stake in semiconductor company Graphcore to zero”

Real Solution: A simple framework (tinygrad)

- tinygrad has effectively one kernel to implement with 12 ops. Adding new accelerators is easy.
- On the frontend, both Stable Diffusion and LLaMA are implemented in tinygrad with less lines than PyTorch versions.
- On the Snapdragon 845 GPU, it is already 2x faster than Qualcomm's SNPE, and is used to run the driving model in comma.ai's openpilot.
- It is conceptually better than others, but it still takes a while to close the performance gap for say, PyTorch on NVIDIA. (due to hugely disproportionate development time)



"bro, you just got those GitHub stars cause your (sic) famous" — some HN hater

```
class UnaryOps(Enum): NOOP = auto(); EXP = auto(); LOG = auto(); CAST = auto() # noqa: E702
class BinaryOps(Enum): ADD = auto(); SUB = auto(); MUL = auto(); DIV = auto(); POW = auto(); CMPEQ = auto(); MAX = auto() # noqa: E702
class ReduceOps(Enum): SUM = auto(); MAX = auto() # noqa: E702
```

The 12 ops. I didn't count NOOP.

“matmul” is not a Simple Op

```
1  from tinygrad.tensor import Tensor
2  N = 2048
3  a, b = Tensor.randn(N, N), Tensor.randn(N, N)
4  c = (a.reshape(N, 1, N) * b.permute(1,0).reshape(1, N, N)).sum(axis=2)
```

This is a fast matmul in tinygrad.

Laziness is the right compromise

```
class MovementOps(Enum): RESHAPE = auto(); PERMUTE = auto(); EXPAND = auto(); PAD = auto(); SHRINK = auto(); STRIDE = auto()
```

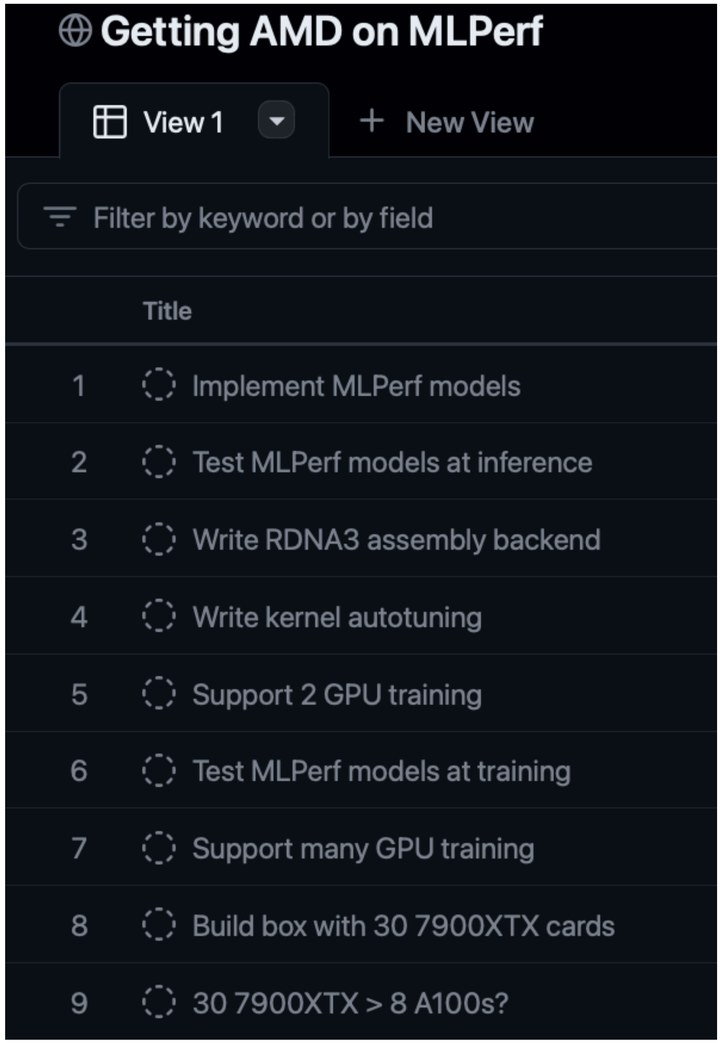
reshape in PyTorch makes copies sometimes….my poor RAM bandwidth

Business Model

- **We will sell (AI training) boxes for more than they cost us to make**
- These boxes will have the highest FLOPS/$ (TCO with power included)
- We will (slowly and appropriately) climb the stack, from sticking consumer GPUs in a prefab case, to making metal boxes and PCI-E switch boards, to making motherboards and custom cards, to making chips, to making fabs.
- As the primary developers of tinygrad, if people are using tinygrad, we have a moat forever. All else equal, people will always pay a premium to buy the most tested hardware that the software was developed on.
- Willing to manage clouds if customers request and sell “cloud mining”, but only if we also sell the box. Developer adoption is top priority.

“It’s an old business model, but it checks out”

Short Term Goal (6-18 months)



MLPerf is a great ad.

We do it with one of our custom boxes. 30x 7900XTX, 10U, one computer, priced appropriately under “market price” for training FLOPS.

Box hits the market when our MLPerf results do, with FLOPS/$ breakdown.

Long Term Goal

- I am an accelerationist. Cheaper compute means more compute, and more compute means the future comes faster.
- We will raise FLOPS/$ and FLOPS/W as fast as possible.
- By commoditizing the petaflop, we will harness the power of market dynamics to drive those key metrics.
- And we will take our 5-50% depending on market conditions 😊

“You should like, commoditize your complement, bro” — some dude on Twitter