Benchmarking FLOPs utilization on TPU v4

Efficiency is a critical metric for large scale ML training platforms

While MLPerf™ remains the gold standard for comparing training performance between platforms (and Google submitted systems based on TPU v4 to MLPerf Training v0.7, v1.0, and v1.1), the MLPerf benchmark suite is not yet representative of large-scale models like GPT-3 and PaLM that are increasingly central to modern machine learning. Training these models in a reasonable amount of time requires highly scalable systems, and **the cost-effectiveness of these systems is directly related to their end-to-end efficiency: how much of their peak computational performance can be harnessed for a large-scale training workload**.

In our recent PaLM paper, we introduced an efficiency metric called Model FLOPs Utilization (MFU). This is measured as the ratio of the observed throughput (in, for example, tokens per second for a language model) to the theoretical maximum throughput of a system harnessing 100% of peak FLOPs. It differs from other ways of measuring compute utilization because it doesn't include FLOPs spent on activation rematerialization during the backward pass, meaning that efficiency as measured by MFU translates directly into end-to-end training speed.

TPU v4 Pods demonstrate exceptional training efficiency

To evaluate the MFU of a key class of workloads on TPU v4 Pods, we carried out an in-depth benchmark campaign on a series of decoder-only Transformer language model (GPT) configurations that range in size from billions to trillions of parameters. In particular, we evaluated training efficiency with two scaling patterns that are relevant to both Google's internal users and Google Cloud customers: "weak scaling," where we grew the model size in proportion to the number of chips used, and "compute-optimal scaling," where we grew the model size in proportion to the square root of the number of chips used (as recommended by a recent result in language model scaling research).

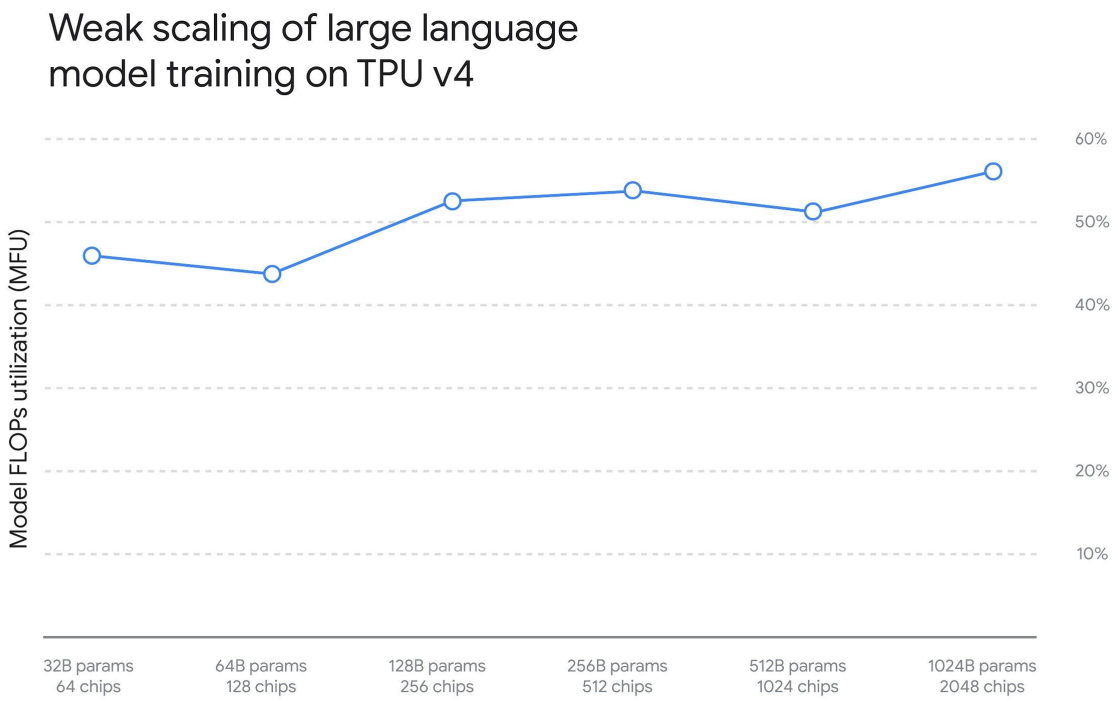

Figure 1: *Model FLOPs Utilization (MFU) for a series of GPT language models trained on TPU v4 Pod slices with different numbers of chips. The size of the models was scaled linearly with the size of the slices, making this a demonstration of "weak scaling." MFU is described in more detail in the PaLM paper.*

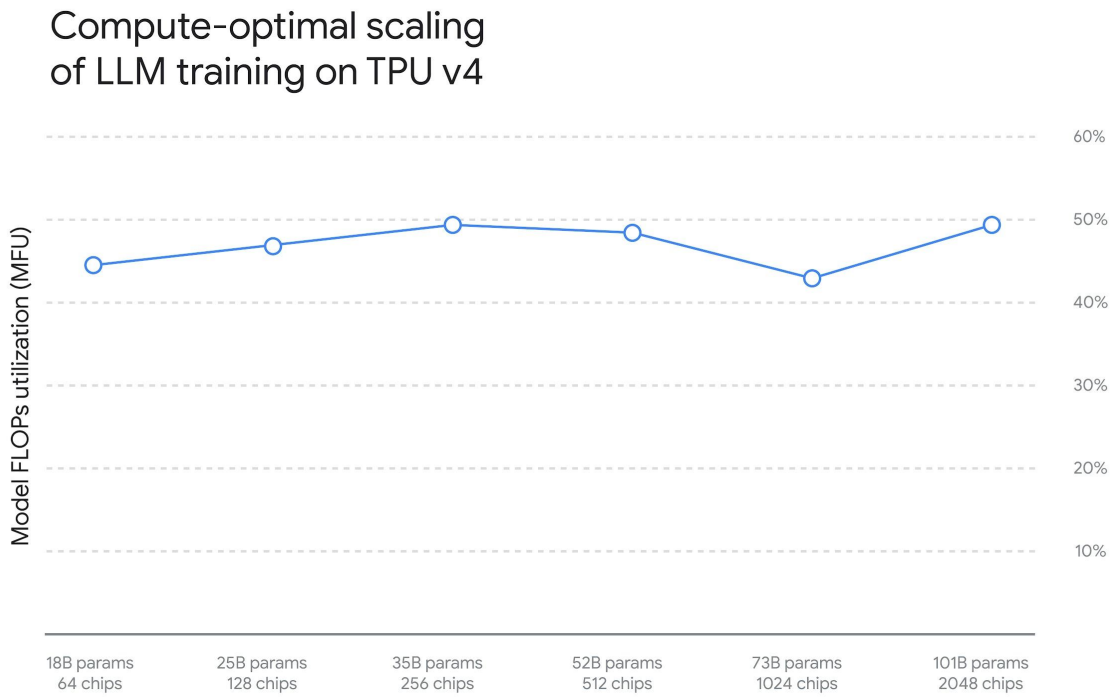

Figure 2: *Model FLOPs Utilization (MFU) for a series of GPT language models trained on TPU v4 Pod slices. The size of the models was scaled as the square root of the number of chips, such that each can train for the number of tokens recommended in "Training Compute-Optimal Large Language Models" in about 50 days.*

With both scaling approaches, TPU v4 demonstrated exceptional efficiency at scale. In particular, TPU v4 achieved 44-56% MFU in our benchmark campaign, and PaLM itself, training with a more efficiency-friendly architecture but at an unprecedented scale of 6144 TPU v4 chips, achieved 46%. In comparison, similar models trained on other systems have achieved between 30% and 40% MFU:

Model/Benchmark	Parameters	Chips	MFU
GPT-3 *Brown et al.*	175B	V100	21%
Gopher *Rae et al.*	280B	4096 TPU v3	32%
Megatron–Turing NLG *Smith et al.*	535B	2240 A100	30%
Megatron benchmarks *Narayanan et al.*	1.7B–1T	32–3072 A100	32–39%
PaLM *Chowdhery et al.*	540B	6144 TPU v4	46%*
TPU v4 weak scaling	32B–1T	64–2048 TPU v4	45–56%
TPU v4 optimal scaling	18B–101B	64–2048 TPU v4	44–50%

Table 1: *Model FLOPs utilization for several recent Transformer models and benchmarks. MFU numbers reported here for GPT-3, Gopher, Megatron–Turing NLG, and PaLM are derived in the PaLM paper, while numbers for Megatron benchmarks (originally including activation rematerialization) are converted to MFU by multiplying by 0.75. *PaLM used a model architecture variant that runs attention and feedforward layers in parallel, so its MFU is not directly comparable to the others, all of which used standard GPT architectures.*

Together with PaLM, the weak scaling results demonstrate exceptional compute utilization for the largest-scale models, while TPU v4 maintains or increases its efficiency as the model and system get larger. The compute-optimal scaling results demonstrate that this also holds true when the model size increases more slowly than the system size, a more challenging scaling configuration that emphasizes network performance.

Note that the weak scaling and optimal scaling benchmark results used synthetic language datasets, and the models were run on Google-internal TPU v4 infrastructure and were not trained to convergence.

Authors: James Bradbury and Qiao Zhang, Software Engineers, Google Research, Aarush Selvan, Product Manager, Google