

Avaliação de Desempenho na Inferência de LLMs: Um Estudo Comparativo entre TPU e GPU utilizando vLLM

Jose Augusto M. de Lima¹, Calebe P. Bianchini¹,

¹Programa de Pós-Graduação em Computação Aplicada – PPGCA/FCI
Universidade Presbiteriana Mackenzie – São Paulo, SP – Brasil

joseaugustomarcellino.lima@mackenzista.com.br

calebe.bianchini@mackenzie.br

Resumo. Este estudo apresenta uma análise de desempenho focada na inferência de Grandes Modelos de Linguagem (LLMs), comparando a arquitetura Tensor Processing Unit (TPU) com aceleradores GPU tradicionais. A avaliação utiliza o modelo Llama-3.1 e o framework de inferência vLLM. Os resultados demonstram que a TPU supera a GPU de referência em throughput total, além de apresentar tempos de resposta (TTFT e TPOT) aproximadamente 2,5 vezes mais rápidos.

Abstract. This study presents a performance analysis focused on Large Language Model (LLM) inference, comparing the Tensor Processing Unit (TPU) architecture with traditional GPU accelerators. The study utilizes the Llama-3.1 model and the vLLM inference framework. The results demonstrate that the TPU outperforms the reference GPU in total throughput, while also delivering response times (TTFT and TPOT) that are approximately 2.5 times faster.

1. Introdução

O avanço da Inteligência Artificial (IA) generativa, impulsionado pelos Grandes Modelos de Linguagem (LLMs), transformou radicalmente as demandas computacionais. Se nos últimos anos o grande desafio se concentrava no treinamento de redes neurais profundas, o paradigma atual deslocou-se para a fase de inferência. A implantação de modelos como o Llama 3 [Touvron et al. 2023] exige infraestruturas capazes de processar requisições concorrentes massivas com baixíssima latência e alto *throughput*, impondo desafios significativos aos aceleradores de *hardware* tradicionais.

Enquanto as Unidades de Processamento Gráfico (GPUs) consolidaram-se como o padrão de mercado devido à sua flexibilidade arquitetural e ecossistema maduro de software [Silva et al. 2022], as Unidades de Processamento de Tensor (TPUs) oferecem uma abordagem de Arquitetura de Domínio Específico (DSA). TPUs foram concebidas desde sua gênese para maximizar a eficiência em operações tensoriais, abrindo mão do paralelismo de propósito geral em favor de Unidades de Multiplicação de Matriz (MXUs) extremamente densas e eficientes [Jouppi et al. 2017].

Este artigo propõe uma avaliação comparativa de desempenho na inferência de LLMs, contrapondo a TPU v6e (Trillium), com a GPU NVIDIA A100, utilizando o *framework* de inferência vLLM [Kwon et al. 2023]. O objetivo é investigar como as estas arquiteturas se comportam diante de gargalos reais de inferência, especificamente analisando o *Memory Wall* e a fragmentação do KV Cache, avaliando métricas específicas, como o *Tempo até o Primeiro Token* (TTFT) e o *Tempo por Token de Saída* (TPOT).

2. Arquiteturas de *Hardware* para Inferência: GPU vs TPU

O processo de inferência dos LLMs modernos, baseado na arquitetura *Transformer*, é dividido em duas fases [Agrawal et al. 2023] com características computacionais opostas:

- Fase de *Prefill* (Processamento do *Prompt*): onde a requisição do usuário é processada para gerar o primeiro *token*, através de operações de multiplicação de matrizes densas.
- Fase de *Decode* (Geração de Texto): onde o modelo gera os *tokens* subsequentes, precisando acessar e atualizar continuamente um *KV Cache* (histórico de tensores “chave-valor” das atenções passadas).

Para compreender os resultados de desempenho na inferência de LLMs entre as arquiteturas de GPU e TPU, é necessário analisar as diferenças de design entre elas, especialmente no que se refere ao gerenciamento de memória e processamento matricial.

A força da **GPU** reside na sua flexibilidade para executar qualquer tipo de *kernel* customizado. No entanto, na fase de *Decode* de LLMs, o tráfego constante de leitura do *KV Cache* da HBM gera subutilização do poder matemático da GPU, tornando a largura de banda da memória o principal teto de desempenho [Silva et al. 2022].

Já as **TPUs** utilizam uma arquitetura de *Systolic Array* em suas Unidades de Multiplicação de Matrizes (MXU). Essa arquitetura foi desenhada para a matemática de tensores, permitindo que os pesos dos modelos permaneçam estáticos no *chip* enquanto os dados de ativação fluem através das células de processamento, reduzindo drasticamente os acessos à memória [Jouppi et al. 2017]. Na TPU, a alta densidade das MXUs combinada com uma memória HBM de grande vazão permite um ganho de velocidade na fase de *Prefill* e garante a rápida alimentação dos dados na fase de *Decode*.

As diferenças entre essas arquiteturas podem ser vistas na Figura 1, adaptada de [Nikolić et al. 2022].

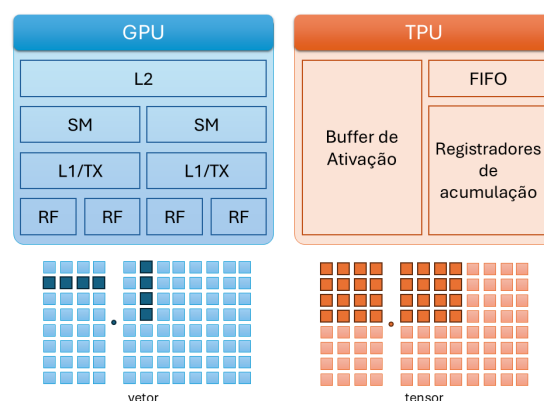


Figura 1. Hierarquia de memória e sistema de computação da GPU e da TPU.

3. Otimização de Inferência com vLLM e *PagedAttention*

Durante uma inferência, o tamanho do *KV Cache* de cada requisição é imprevisível, pois o número de *tokens* que o modelo gera é desconhecido de antemão. Consequentemente, sistemas não otimizados pré-alocam blocos massivos e contíguos de memória na GPU ou

TPU para acomodar o limite máximo de *tokens*. Essa abordagem desperdiça até 80% da memória devido à fragmentação interna e alocação superestimada, limitando severamente o número de usuários simultâneos [Kwon et al. 2023].

O vLLM resolve esse problema introduzindo o *PagedAttention*, algoritmo inspirado no gerenciamento de memória virtual de sistemas operacionais tradicionais. Em vez de exigir blocos contíguos de memória, o vLLM divide o *KV Cache* em páginas (blocos de *tokens*) de tamanho fixo. Desta forma, a memória física é alocada apenas quando novos *tokens* são efetivamente gerados, preenchendo as páginas dinamicamente, eliminando a necessidade de espaço contíguo e praticamente eliminando o desperdício de memória.

4. Experimentos e Resultados

Os experimentos foram executados em ambiente de nuvem do *Google Cloud Platform* utilizando o *framework* vLLM configurado para submeter cargas sintéticas de estresse à partir da adaptação de *benchmarks* públicos¹. O modelo escolhido para os testes foi o *Llama-3.1-8B-Instruct*, carregado com precisão *bfloat16*. O *benchmark* de estresse foi configurado com os seguintes parâmetros: (i) 3.000 prompts gerados aleatoriamente; (ii) 128 requisições simultâneas; (iii) tamanho fixo de entrada de 1.800 *tokens*, e tamanho fixo de saída de 128 *tokens* (total aproximado de 5,4 milhões de *tokens* de entrada processados e 384 mil *tokens* de saída gerados); e (iv) a reserva para o *KV Cache* e os pesos do modelo foi definida em 98% da capacidade de memória.

No ambiente TPU, utilizou-se uma instância com 1 *chip* TPU v6e-1 (32 GB HBM). Para a GPU, utilizou-se uma instância com 1 *chip* NVIDIA A100 (40 GB HBM). A escolha destas arquiteturas foi baseada na capacidade total de HBM e largura de banda teórica de 1,6 TB/s. A execução do *benchmark* revelou um distanciamento expressivo de desempenho entre as arquiteturas, consolidado na Tabela 1.

O TTFT reflete o quão rapidamente o acelerador consegue ingerir o contexto de entrada (1.800 *tokens*). É uma fase limitada primariamente por FLOPS. A TPU registrou um TTFT médio de 5,2 segundos contra 12,6 segundos da GPU (ganho de 2,4 vezes). Esse ganho mostra a eficácia superior dos *Systolic Arrays* ao processar multiplicações matriciais densas em relação à arquitetura generalista da GPU testada.

Simultaneamente, o TPOT, que avalia a latência de geração na fase de *Decode* foi de apenas 41,55 ms na TPU, contra 110,80 ms na GPU. Isso evidencia que a largura de banda de memória HBM integrada à TPU é capaz de fornecer os tensores de *KV Cache* de forma mais fluida para as MXUs, operando em harmonia com o sistema *PagedAttention* do vLLM e suas características.

Já o *Throughput* Total de *Tokens* (agregando entrada e saída) é a métrica utilizada para avaliar o custo-benefício e a escalabilidade de um servidor de IA. A TPU alcançou uma vazão de 23.195 *tokens/s*, superando os 9.156 *tokens/s* da GPU.

5. Considerações finais

Os resultados apresentados neste artigo reforçam que a especialização do *hardware* é vital tanto para a fase de treinamento de LLMs como especialmente importante para

¹Disponível em <https://github.com/AI-Hypercomputer/tpu-recipes/>

Tabela 1. Consolidação dos Resultados de Desempenho de Inferência (vLLM)

Métrica Analisada	TPU (v6e-1)	GPU (A100)
Duração do <i>Benchmark</i> (s)	249,2	631,67
<i>Throughput</i> de Requisições (req/s)	12,04	4,75
<i>Throughput</i> Total de <i>Tokens</i> (tok/s)	23.195,06	9.156,66
<i>Throughput</i> de <i>Tokens</i> de Saída (tok/s)	1.540,72	607,91
Tempo Médio até Primeiro <i>Token</i> - TTFT (ms)	5.217,95	12.670,35
Tempo Médio por <i>Token</i> de Saída - TPOT (ms)	41,55	110,80

a inferência. Mais do que uma simples vantagem baseada em FLOPS na fase de *Prefill*, o estudo evidenciou que a TPU consegue lidar com o *Memory Wall* durante a fase de *Decode*. A vantagem apresentada nos resultados pela TPU ressalta a eficiência dos Arranjos Sistólicos acoplados a memórias HBM de alta vazão e reforça seu uso não apenas como um acelerador alternativo, mas como uma arquitetura altamente competitiva e promissora em termos de desempenho e eficiência.

No entanto, cabe ressaltar que a atual avaliação se baseou em uma configuração específica de estresse. Como próximos passos, pretende-se explorar novos experimentos variando o tamanho dos *prompts*, o volume de *tokens* gerados e os níveis de concorrência. Adicionalmente, busca-se realizar uma análise do Custo Total de Propriedade (TCO) e da eficiência energética baseada na métrica de *tokens* gerados por Watt, parâmetros vitais para o planejamento de infraestruturas sustentáveis.

Agradecimentos

Os autores agradecem o apoio da MackCloud e do projeto SPRACE – Processo nº 2018/25225-9, Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP). Este trabalho foi financiado em parte pelo Fundo Mackenzie de Pesquisa e Inovação (MackPesquisa) – Projeto nº 251005.

Referências

- Agrawal, A. et al. (2023). Sarathi: Efficient llm inference by piggybacking decodes with chunked prefills. <https://arxiv.org/abs/2308.16369>.
- Jouppi, N. P. et al. (2017). In-datacenter performance analysis of a tensor processing unit. In *Proc. of 44th ISCA*, pages 1–12.
- Kwon, W. et al. (2023). Efficient memory management for large language model serving with pagedattention. In *Proc. of the 29th SOSP*, page 611–626.
- Nikolić, G. S. et al. (2022). A survey of three types of processing units: Cpu, gpu and tpu. In *Proc. of the 57th ICESS*, pages 1–6.
- Silva, G. P., Bianchini, C. P., and Costa, E. B. (2022). *Programação Paralela e Distribuída com MPI, OpenMP e OpenACC para computação de alto desempenho*. CasaDoCodigo.
- Touvron, H. et al. (2023). Llama: Open and efficient foundation language models. <https://arxiv.org/abs/2302.13971>.