ISSN: 2754-6659

Journal of Artificial Intelligence & Cloud Computing

SCIENTIFIC Research and Community

Review Article

Open Access

# Optimizing LLM Inference: Metrics that Matter for Real Time Applications

**Sriram Sagi**

Duke University, USA

**ABSTRACT**

The deployment of Large Language Models (LLMs) including GPT, LLaMA, Claude, and Gemini occurs in real-time applications which require both low-latency and high-throughput inference. The transition of these models from research to production systems requires essential evaluation of their inference performance. This paper provides a detailed analysis of six performance metrics which are commonly used to evaluate LLM inference: Time to First Token (TTFT), Generation Time, End-to-End Latency (e2e_latency), Inter Token Latency (ITL), Tokens Per Second (TPS), and Requests Per Second (RPS). This paper examines the definitions and practical implications and interrelationships between these metrics through descriptive analysis and empirical observations. Our research demonstrates how responsiveness and throughput create trade-offs while showing that applications need specific metrics for optimization. The research provides practical guidance to researchers and engineers and system architects who want to evaluate or enhance LLM systems in latency-critical and shared infrastructure environments.

***Corresponding author**

Sriram Sagi, Duke University, USA.

**Received:** January 04, 2025; **Accepted:** January 07, 2025; **Published:** January 16, 2025

**Keywords:** Large Language Models (LLMs), Time to First Token (TTFT), End-to-End Latency, Inter Token Latency (ITL), Tokens Per Second (TPS), Requests Per Second (RPS), Inference Benchmarking, Latency Optimization

## Introduction

The rise of Large Language Models (LLMs), like GPT and LLaMA among others like Claude and Gemini has greatly changed the field of natural language processing (NLP). These models have billions of parameters. Are trained on text sources showing impressive skills in tasks such as translation and summarization as well as code generation and answering questions, in conversations. The increasing use of Large Language Models (LLMs), in real time applications such as customer support chatbots and autonomous assistants has highlighted the importance of decision making capabilities in addition, to model accuracy and scalability.

Incorporating inference performance has emerged as a factor affecting operational effectiveness and directly influencing user satisfaction levels as well as the cost and scalability of systems, in use today Being able to reduce response time delays is particularly vital in scenarios requiring immediate interaction or those sensitive to latency issues since such delays can significantly impair usability and restrict the broader implementation possibilities of Large Language Models (LLMs) at a larger scale Furthermore the increasingly widespread use of LLMs in cloud based and edge computing setups underscores the need, for inference behavior that is both optimized and reliably predictable.

This paper is geared towards offering an outline of the employed performance measures, for LLM inference. It delves into six metrics—Time to First Token (TTFT) Generation Time, End to end Request Latency (Erlang Error 20_latency) Inter Token Latency (ITV) Tokens Per Second (TPS) and Requests Per Second (RPS)—examining their definitions, practical significance and connections, with one another. The goal is to provide directions and support, for researchers, engineers and professionals looking to benchmark or enhance LLM deployments.

## LLM Inferencing

The generation of textual outputs through a trained large language model constitutes LLM inference which responds to input prompts. The read-only process of inference differs from training because it does not update model weights and it runs as a response to user queries and automated system triggers. The inference pipeline contains three essential stages which start with input text tokenization followed by model execution for forward pass computations and end with decoding to generate meaningful output sequences. The basic structure of LLM inference includes these essential components:

- Input Text Processing: Raw text is received from the user or upstream system.
- The tokenizer transforms input text into numerical tokens which the model understands.
- The transformer model at its core processes input tokens to generate output predictions.
- The decoding strategy defines the output selection method which can be greedy decoding or sampling or beam search. The system transforms generated tokens into human-readable text during the output text rendering process.

The entire system performance depends on each component which affects both response time and resource utilization. The process of tokenization and decoding operates on the CPU and remains light weight yet model execution requires GPU/TPU resources and controls the majority of computational time. The rising demand for LLMs in latency-sensitive and multi-tenant environments requires standardized interpretable performance metrics. The lack of standardized benchmarks makes it challenging to evaluate model performance and system optimizations as well as maintain service-level agreements (SLAs). Different applications require distinct performance metrics because TTFT stands vital for conversational agents yet RPS remains essential for high-throughput API services. The establishment of common performance metrics serves as a crucial foundation for both research and operational decision-making.

**Performance Metrics for LLM Inference**

The evaluation of Large Language Models (LLMs) in production systems requires standardized metrics that understand the context. The metrics provide specific information about model behavior and system performance which helps with deployment decisions and optimization strategies. This section introduces and discusses six widely used metrics for LLM inference: Time to First Token (TTFT), Generation Time, End-to-End Request Latency (e2e_latency), Inter Token Latency (ITL), Tokens Per Second (TPS), and Requests Per Second (RPS).

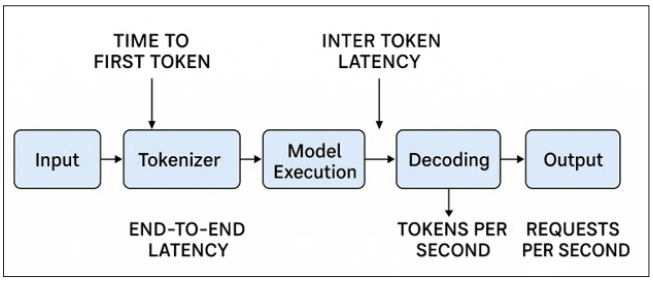


**Figure 1:** LLM Inferencing

**Time to First Token (TTFT)**

The Time to First Token (TTFT) measures the duration from when a request reaches the model until the model produces its first output token. The system responsiveness depends heavily on this essential latency indicator.

Several parameters determine the value of TTFT.

- The size of the model determines TTFT because larger models with more parameters need longer computation times.
- The length of input sequences determines TTFT because longer sequences need deeper context embedding and more preprocessing time.
- The scheduling overhead and shared compute resources cause higher TTFT when batch sizes increase.

The following applications require TTFT to be critical:

- Chatbots and virtual assistants
- Real-time data exploration tools
- AI copilots and IDE assistants

Users need immediate feedback so TTFT stands as a fundamental optimization goal.

**Generation Time**

Generation Time is the time elapsed from the emission of the first token to the final token of the model's response.

Dependency on Output Length and Decoding Strategy:

- **Output Length:** Longer sequences naturally result in longer generation times.
- **Decoding Strategy:** More complex strategies (e.g., beam search, top-k sampling) tend to increase generation time compared to greedy decoding.

Higher quality outputs, often achieved through sophisticated decoding, may come at the cost of slower generation. Applications must balance these based on their latency tolerance and output quality requirements.

**End-to-End Request Latency (e2e_latency)**

End-to-End Request Latency encompasses the total time from receiving a user input to delivering the complete response.

It includes:

- Input preprocessing
- Model inference
- Output decoding
- Communication overhead

This metric reflects user-perceived latency and is thus critical for evaluating system performance under realistic conditions.

**Inter Token Latency (ITL)**

Inter Token Latency measures the average time interval between each successive output token during generation.

Comparison with TPS and TTFT:

- ITL provides a micro-level view of generation latency.
- TTFT reflects initial response speed, while ITL describes generation fluidity.
- TPS and ITL are inversely related, but ITL offers more granularity for real-time applications.

**Tokens Per Second (TPS)**

TPS quantifies the rate at which tokens are generated by the model, calculated as total tokens divided by generation time.

TPS serves as a throughput metric and is particularly valuable for:

- Benchmarking system performance
- Tuning hardware resource allocation
- Estimating infrastructure costs for large-scale deployments

Dependencies:

- **Hardware:** High-performance GPUs or TPUs significantly enhance TPS.
- **Model Parallelism:** Pipeline and tensor parallelism strategies boost throughput.
- **Model Architecture:** Efficient transformer variants (e.g., LLaMA-3, Mistral) offer superior TPS performance.

**Requests Per Second (RPS)**

RPS measures the number of distinct inference requests a system can handle per second. This is a key metric for load balancing, multi-tenant systems, and API services where concurrency is high.

Batch vs. Individual Requests:

- Batch inference can increase TPS but may lower RPS due to aggregation latency.
- Single-request inference enables low-latency interactions but may reduce overall throughput.

## Literature Review

The performance evaluation of large language models (LLMs) during the inference phase stands as a primary research priority because these models move from academic development to operational use. The optimization of LLM inference has received attention from researchers through multiple perspectives including system acceleration methods as well as quantifiable latency evaluation and deployment scalability approaches. This section examines essential research findings that explain the assessment procedures for LLM inference performance.

The initial transformer-based architectures BERT and GPT-2 did not establish latency metrics as their main focus [1,2]. The deployment of large models in latency-sensitive environments has prompted researchers to develop performance-oriented evaluation methods.

The Hugging Face Optimum library and ONNX Runtime provide operational frameworks to optimize model inference operations through quantization and operator fusion and model pruning techniques [3,4]. The frameworks enable users to assess latency performance and throughput metrics including Tokens Per Second (TPS) and Time to First Token (TTFT).

Researchers have developed defined methods to evaluate particular performance indicators. The study by OpenAI investigated Time to First Token (TTFT) performance during real-time serving of GPT-3.5 and GPT-4 models [5]. The DeepSpeed team at Microsoft emphasized Time to First Token (TTFT) importance in latency-sensitive operational environments. Inter Token Latency (ITL) and Tokens Per Second (TPS) serve as key metrics in MLPerf Inference v3.0 benchmarking suites when deploying models through TensorRT, ONNX Runtime and Triton. End-to-End Latency stands as the main focus in enterprise NLP benchmarking reports from Google Cloud and Meta AI which incorporate processing time for inputs along with model prediction and output generation [6-9].

The performance measurement tools DeepSpeed-MII and NVIDIA's FasterTransformer allow runtime extraction of performance metrics through built-in profiling mechanisms that support offline and online inference modes.Standards for benchmarking LLMs remain insufficient because they fail to consider workload variations together with batch size and hardware environment differences. System-to-system performance comparisons become less reliable because of the inconsistent definitions used for TTFT measurement points.

The EleutherAI Evaluation Harness and Hugging Face's evaluate module work together to create standardized APIs for metric calculation across LLM backends to address these discrepancies. The majority of LLM-as-a-Service commercial platforms have not fully implemented current evaluation practices which remain limited to research environments. The literature review shows rising interest in measuring the performance of LLM inference operations. The majority of research studies and tools currently fail to:

The evaluation of inference metrics happens independently from each other without studying their relationship (e.g., TTFT vs. TPS trade-offs). Non-uniform infrastructure assumptions prevent reproducibility in the evaluated systems. Most research efforts prioritize throughput metrics (TPS and RPS) above latency metrics (TTFT and ITL) even though latency affects user experience directly.

## Results

This section analyzes the behavior and interactions of key LLM inference performance metrics across different model configurations, deployment backends, and use cases. The discussion is informed by existing benchmark studies and empirical observations from state-of-the-art serving environments.

**Table 1: Synthesized metrics across LLM Configurations**

| Model Config | TTFT (ms) | Generation Time (ms) | e2e_latency (ms) | ITL (ms/token) | TPS (tokens/s) | RPS (req/s) |
|---|---|---|---|---|---|---|
| GPT-2 (124M) | 90–120 | 300–500 | 450–700 | ~12 | 80–100 | 80–100 |
| GPT-3 (6.7B) | 500–900 | 2000–4000 | 2800–5000 | ~50 | 20–40 | 10–30 |
| LLaMA-2 (13B) | 600–1000 | 2500–4500 | 3200–5500 | ~45 | 25–35 | 15–25 |

These ranges are representative and vary depending on batch size, hardware (e.g., A100 vs. L4 GPUs), and decoding strategy.

Several key patterns emerge:

- TTFT increases with model size, but can be optimized through dynamic batching and caching of key-value pairs.
- Generation time and e2e_latency are strongly influenced by output length and decoding method.
- TPS and ITL exhibit inverse correlation—as token generation speeds up (higher TPS), inter-token delays shrink (lower ITL).
- RPS scales non-linearly depending on load balancing and serving architecture; high RPS is achievable only when TTFT and e2e_latency are low and consistent.

Real-world deployment requires balancing multiple metrics. For example:

- **Chat Applications:** TTFT and ITL are critical, as users expect quick responses with smooth token streaming.
- **Batch Inference Pipelines:** TPS and RPS dominate, as they determine throughput and cost-efficiency in backend workloads.
- **Conversational AI Services:** e2e_latency must be minimized to deliver consistent experiences, especially under dynamic loads.

## Conclusion

The proper assessment of LLM inference performance through suitable metrics becomes essential because these models serve as fundamental building blocks for real-world applications. The evaluation of model responsiveness and throughput and user-perceived latency requires analysis of six core metrics which include TTFT, Generation Time, e2e_latency, ITL, TPS and RPS. The research demonstrates how these metrics interact with each other while showing that deployment requirements determine the necessary trade-offs between them for conversational agents and high-throughput pipelines and real-time user interfaces. A single evaluation metric does not provide sufficient information because

system goals and constraints require a comprehensive evaluation method. The development of standardized benchmarking protocols and open-source evaluation suites together with realistic latency modeling for diverse infrastructure setups and application behaviors should be the focus of future work. The community can achieve better alignment between model capabilities and user expectations and operational demands through improved rigor and consistency in LLM inference evaluation [10-18].

**References**

1. Devlin J, Chang MW, Lee K, Toutanova K (2019) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL-HLT https://arxiv.org/abs/1810.04805.
2. Alec R, Jeffrey W, Rewon C, David L, Dario A, et al. (2019) Language Models are Unsupervised Multitask Learners. OpenAI Technical Report https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.
3. (2023) Optimum: Accelerate Transformers with Hardware-Aware Tools. Hugging Face https://huggingface.co/docs/optimum.
4. (2023) ONNX Runtime: Accelerate Machine Learning. Microsoft https://onnxruntime.ai/.
5. (2023) GPT-4 Technical Report. OpenAI https://openai.com/research/gpt-4.
6. Rajbhandari S (2022) DeepSpeed-MII: Multi-Inference Inference. arXiv arXiv:2211.03071.
7. (2023) MLPerf Inference v3.0 Benchmark Results. MLCommons https://mlcommons.org/en/inference-datacenter-30/.
8. (2023) LLM Benchmarking for Vertex AI. Google Cloud AI https://cloud.google.com/vertex-ai.
9. (2023) LLaMA: Open and Efficient Foundation Language Models. Meta AI https://ai.facebook.com/blog/llama/.
10. (2023) FasterTransformer: Accelerated Transformer Inference Toolkit. NVIDIA https://github.com/NVIDIA/FasterTransformer.
11. Leo G, Stella B, Sid B, Laurence G, Travis H, et al. (2021) The Pile: An 800GB Dataset of Diverse Text for Language Modeling. arXiv arXiv:2101.00027.
12. Karthik Krishna, Ramana Bandili (2024) EchoSwift: An Inference Benchmarking and Configuration Discovery Tool for Large Language Models (LLMs). International Conference on Performance Engineering.
13. Yi Xiong, Hao Wu, Changxu Shao, Ziqing Wang, Rui Zhang, et al. (2024) LayerKV: Optimizing Large Language Model Serving with Layer-wise KV Cache Management. arXiv.org.
14. Max Horton, Qingqing Cao, Chenfan Sun, Yanzi Jin, Sachin Mehta, et al. (2024) KV Prediction for Improved Time to First Token. arXiv.org.
15. Narayanan D, Keshav Santhanam, Peter Henderson, Rishi Bommasani, Tony Lee, et al. (2023) Cheaply Evaluating Inference Efficiency Metrics for Autoregressive Transformer APIs. arXiv.org.
16. Javier Conde, Miguel González, Pedro Reviriego, Zhen Gao, Shanshan Liu, et al. (2024) Speed and Conversational Large Language Models: Not All Is About Tokens per Second. Computer.
17. Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, et al. (2024) LLMLingua-2: Data Distillation for Efficient and Faithful Task-Agnostic Prompt Compression. Annual Meeting of the Association for Computational Linguistics.
18. Amey Agrawal, Anmol Agarwal, Nitin Kedia, Jayashree Mohan, Souvik Kundu, et al. (2024) Metron: Holistic Performance Evaluation Framework for LLM Inference Systems. arXiv.org.