Inference Speed Is the Key To Unleashing AI's Potential

Engagement has always been critical to exponential growth of technology. When tech is slow, we'll use it when we need it but we won't engage with it. Humans are quite impatient; research shows that when websites pages are delayed by 300 - 500 milliseconds (ms), engagement drops by around 20%[1]. Conversely, when tech is fast - wow fast - we'll use it over and over again. Speed drives engagement, and engagement drives productivity, collaboration, creativity, and innovation.

The same applies to artificial intelligence (AI) and large language models (LLMs). LLMs have opened up a world of possibilities of how AI can help people across a wide range of fields up their game. Pick any role in any industry, and you'll find humans in the value loop applying their smarts, experience, training, and judgment to get things done. These "humans-in-the-loop" already use computers as tools in their work, but now LLMs present an opportunity to boost performance by orders of magnitude. Regardless of our field, we can all have access to superhuman intellect and insight at human speed. Software development, content creation, customer service, analytics, fraud and crime prevention, health care—**AI can 10X human impact across them all.**

But not if it is slow. You can have the best LLM system in the world, and if it's too slow people won't use it. This is why speed is the top priority for most AI applications.

The problem is, when AI developers finish training their LLMs and turn to deploying and scaling them (aka inference), the only hardware option they have is the good old graphics processor unit (GPU). GPUs are cool for training models, but for inference, they're slowpokes, leading directly to the great-model-that-no-one-uses problem.

Speed is affected by other factors, such as quality and scale. **When determining an inference strategy for a given application, business and technology leaders need to ensure it can achieve the necessary quality and scale while still maintaining a fast enough pace.** In this paper, we look deeper into each of these factors and provide a clear set of questions leaders can pose to their teams and partners to guide them to the best strategy.

LLMs and other generative AI applications have the potential to transform markets and solve big challenges, but only if they are fast enough, which depends on getting inference right. This paper helps you do that.

1. For example, a 2017 study by Google showed that delays of 300-500ms reduce user engagement by 20%, while a 2020 study by the University of Michigan found the reduction to be 22%.

The Need for Speed

When measuring the speed of an AI workload, there are two metrics to consider:

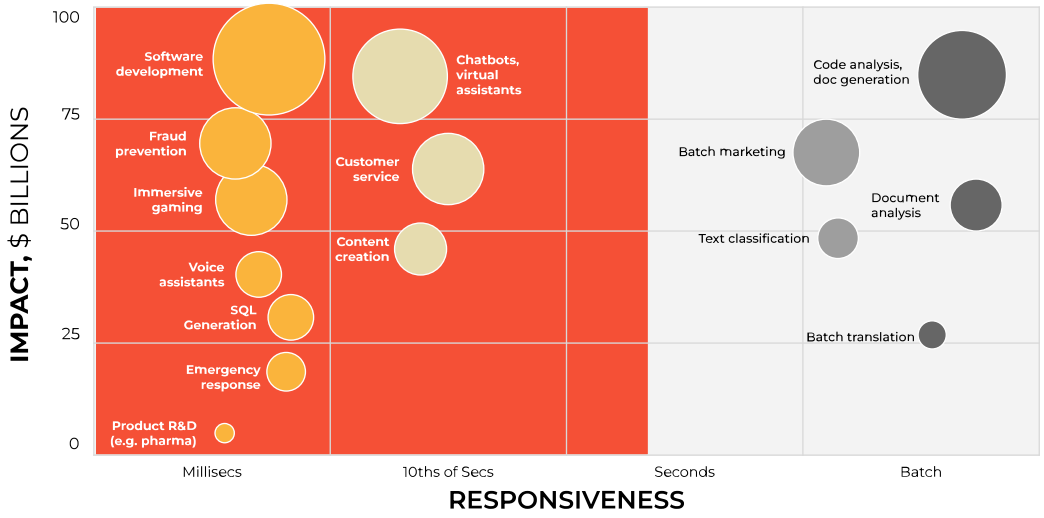
1. **The app's responsiveness is how quickly the app processes the input query and generates its first token. The metric is time to first token (TTFT).**

2. **The app's latency is its overall speed in delivering its complete answer. The metric is tokens per second (tokens / second).**

There is ample research, dating back to the advent of digital computing, that responsiveness needs to be under about 200ms to support humans in solving complex problems. This level of performance creates a natural back and forth that integrates seamlessly into our creative and problem solving processes. When computer response creeps above 200ms, different parts of the brain are activated that help us adapt to the slower pace. The brain is literally disrupted from its flow. This finding is consistent with most people's experiences in using low latency services such as Google. We expect a response within 200ms and our thought process is disrupted when we don't get it.

Both TTFT and tokens / second should be measured on a per user basis. Saying, for example, that an inference engine supports speeds of up to 12,000 tokens / second may sound awesome, but not when that metric is for a batch of 1,024 users. On a per user basis, that equates to about 12 tokens / second–not nearly as speedy.

There is a set of AI applications that don't require this level of performance. Non real-time solutions, such as offline document or data analysis, don't require a fluid conversation. Read all this data and text and find hidden patterns, tomorrow morning by 9am is fine.



Refer to: "The Economic potential of Generative AI: The next productivity frontier." McKinsey, 2023

These applications can add plenty of value, but they barely scratch the surface of generative AI's potential. Meanwhile, there is a much larger set of AI applications that do require this level of performance. For example, a customer service representative helping a customer with a problem or a product recommendation could be much more effective when paired with an AI app, but also doesn't have the luxury of waiting several seconds for their AI bot to respond. That will only annoy the customer.

Similarly, software development, real-time analytics, voice assistants, and gaming are all areas where the human in the value loop can use AI to up their game, but only if the compute performance is there. Give people in these fields the power of AI at their fingertips, and the impact will be enormous (at Groq, we call this concept **HumanPlus**.)

2. Delays in Human-Computer Interaction and Their Effects on Brain Activity. Kohrs C, Angenstein N, Brechmann A (2016) Delays in Human-Computer Interaction and Their Effects on Brain Activity. PLOS ONE 11(1): e0146250. https://doi.org/10.1371/journal.pone.0146250

The Need for Speed & Quality

While an LLM's speed is critical, so is its quality. There are many ways to measure quality. You can see how the model performs in standardized tests (e.g. the MCAT or state bar test)[1], or deploy the model and have humans evaluate its answers.

Regardless of how quality is measured in inference, the two biggest factors contributing to a model's quality[2] are model size (number of parameters) and sequence length (maximum size of the input query). Model size can be thought of as a search space; the bigger the space, the better the results. So a 70B parameter model will usually generate better answers, for example, than a 7B parameter model. Sequence length is akin to context. A bigger sequence length means more information - more context - can be fed into the model, leading to a more relevant and pertinent response. Conversely, a shorter sequence length may cause the model to be like a person who forgets what they were talking about. It loses the train of thought.

These parameters also affect speed. Bigger models and sequence lengths take more compute power and can bog things down. To evaluate an inference engine's speed, start with the model and sequence length size that the application will require to achieve its quality objectives.

Quality can also be boosted by various algorithmic techniques. Say someone asks you a question and you blurt out the first thing that comes to mind. Good for certain kinds of psychology tests, not so good for helping tackle complex issues. Unfortunately, that's what LLM bots do: they blurt out the first thing that comes to mind. Fortunately, there are a variety of techniques developers can use to improve upon this stream of consciousness.

Beam search, for example, is a kind of "search ahead" technique that generates better results by looking ahead at a possible set of outcomes, then selecting the best one based on a scoring function. Think of a chess-playing bot who, rather than just predicting its next move, gives itself ten different next-move options and plays out the match for each of those ten. It may discover that what it thought was its best next move was in fact inferior to another option. If it simply relies on its stream of consciousness, its quality won't be as good as it could be.

Self-reflexion is another way to boost quality, especially as training improves and search-ahead techniques like beam search become less effective. It refers to a model's ability to reflect and improve on its own output, sort of like how a writer might produce a first draft, then edit and edit some more to improve it. Self-reflexion allows models to improve quality without the need for manual intervention.

> **The two biggest factors contributing to a model's quality are model size (number of parameters) and sequence length (maximum size of the input query).**

These algorithms boost quality by improving on the application's initial stream of consciousness response, but they also require additional compute resources, so the inference solution will include enough compute performance "budget" to allow for them. **AI business leaders need to consider all of these quality factors–model size, sequence length, quality-boosting algorithm–when deciding upon their inference engine.**

3. arXiv:2303.11366 [cs.AI]

The Need for Speed & Scale

Having a racecar that goes 200 mph is fun, but what happens when you hitch it to a trailer crammed with a hundred or thousand people? It's not going 200 mph anymore. This can't happen when your racecar is an AI solution: it has to perform as fast when it is fully ramped up as when it has just a few users.

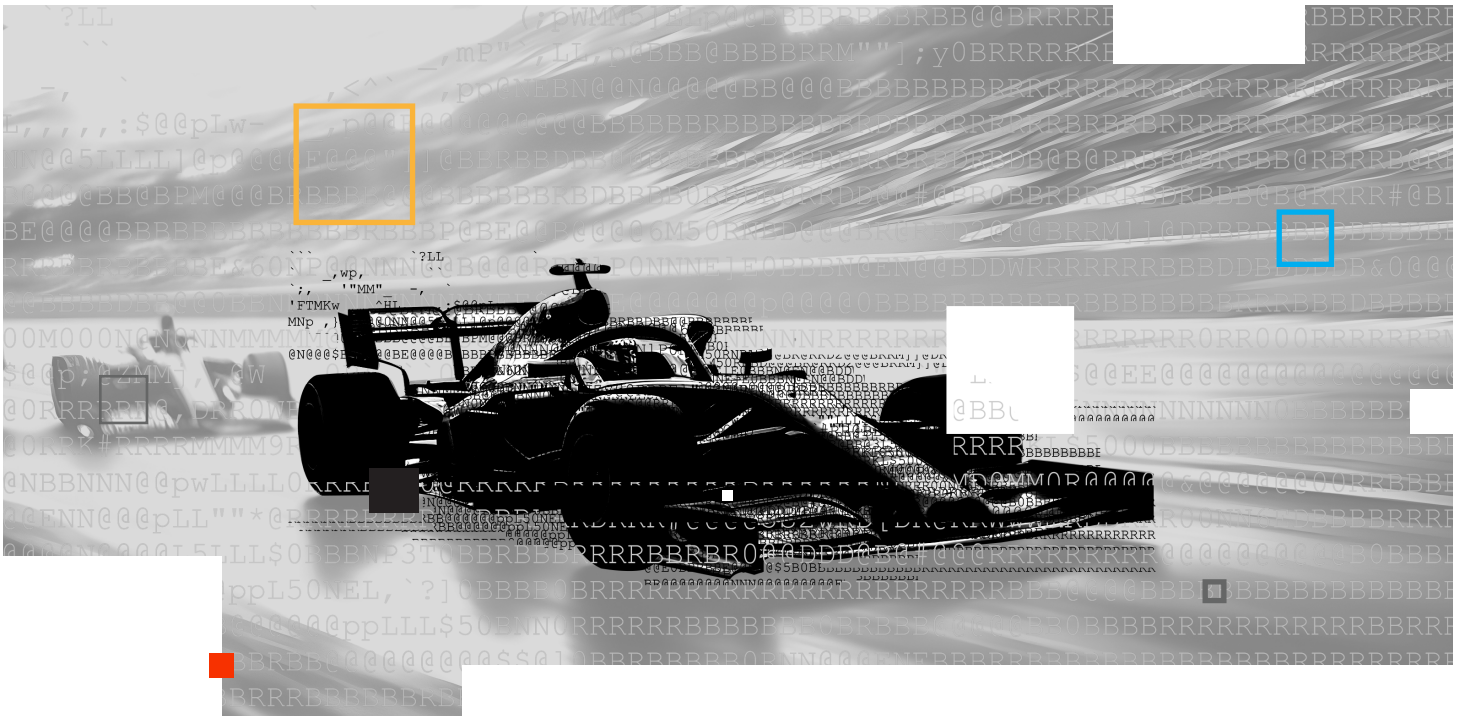
A popular way to measure scale for AI apps is concurrent users: how many people can be using the system at the same time. At first glance this metric makes sense, but dig a bit deeper and you'll see that it grossly oversimplifies things. When people are "using" an AI bot, they are engaging its inference compute resources only a small fraction of the time and their rate of compute resource usage varies greatly depending on the application. As a result, the concurrent users metric is pretty meaningless.

How many queries are hitting the application per minute system wide is a better, more nuanced metric. It gets much closer to gauging the actual load being placed on the compute resources. When determining inference strategy, make sure the engine has the performance capabilities to exceed speed requirements at the anticipated queries / minute.

Cost (Almost) Doesn't Matter

The one thing we haven't discussed so far is cost. Cost ($ / token) is important, and for a limited set of non-real time AI solutions it can be the primary decision factor. But focusing solely on cost is a recipe for failure, since it precludes the organization from putting AI to work in truly transformative solutions. To unlock the vast potential of AI, cost (almost) doesn't matter. It's all about speed.

> **An AI solution has to perform as fast when it is fully ramped up as when it has just a few users.**



What to Ask Your Team & Partners

When developing an AI inference strategy, here's questions business and technology leaders can ask of their teams

What are the transformative AI opportunities for my business or organization?

- Where can I employ AI in my organization to supercharge our people and make them much more productive and creative, to maximize HumanPlus? How can we apply today's AI LLMs to transform these components of our business?
- How could prompt engineering enhance or transform our products, services, or operations? Which roles could benefit from real-time conversations with a supercomputer? What new things could they do?

What are my requirements for an AI inference platform?

- For these solutions to be successful, what is our required speed as measured by TTFT per user and token / second per user? How fast does it need to be to keep users in the flow?
- For these solutions to be successful, what size model and sequence length is required? What are my quality parameters?
- Can it maintain this speed at the required scale (system-wide queries / minute)?

How do the available inference solutions stack up against my requirements?

- What infrastructure platforms can support my required latency and speed at my required model size and sequence length?
- Which of those platforms can support additional quality enhancing algorithms, such as beam search and self-reflection, while still achieving minimum latency and speed?
- What is the compute cost ($ / token) for the qualifying platforms?

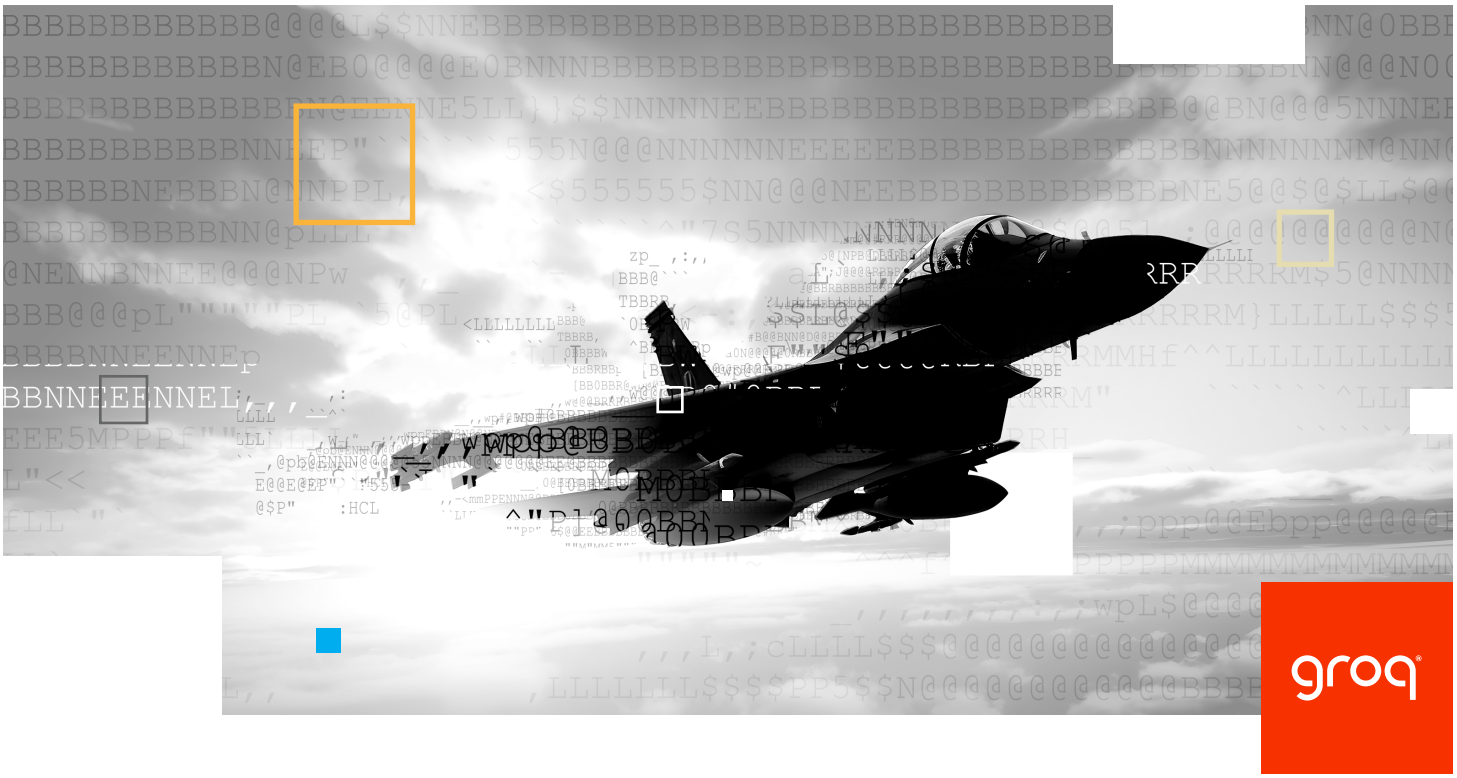
The Promise of Prompt Engineering

Think back to the early days of the internet (or–gasp–AOL), when we connected via dial-up (for younger readers, look it up on Wikipedia and prepare to be shocked). Trying to use the internet for anything useful, like finding a good restaurant, was practically impossible. You would get one answer, then give up and grab the newspaper instead.

Fast forward a few years to the advent of broadband and everything changes. Speed didn't change the game, it created an entirely new game. If you wanted an answer, you asked Google, not just once but dozens of times until you got what you needed.

Lightning fast LLM performance and high quality output will likewise change the game for people working across virtually every field. Fast, smart LLMs will create a brand new human skill–prompt engineering–that will quickly become critical to many roles and fields. Prompt engineering is about optimally designing and implementing prompts that elicit desired responses from AI systems and get the most out of LLM bots. This socratic, real-time, back and forth method between human and bot to refine and improve the bot's (and ultimately the human's) output is evolving into one of the most critical roles in the AI space. In virtually every field, a person skilled in prompt engineering will work with a bot to produce a far better result than either the bot or person would on their own.

This is what speed enables. It delivers in-the-flow collaboration between human and bot that will up the game for everyone. Maverick and Goose weren't the only ones: we feel the need, the need for speed.



Don’t miss the latest news on our inference performance results!

Groq LPU™ Inference Engine Crushes First Public LLM Benchmark

Groq Delivers up to 18x Faster LLM Inference Performance on Anyscale’s LLMPerf Leaderboard Compared to Top Cloud-based Providers

Hey Groq Prompters! We’re thrilled to announce that Groq is now on the LLMPerf Leaderboard by Anyscale, a developer innovator and friendly competitor in the LLM inference benchmark space.This benchmark includes a selection of Large Language Model (LLM) inference providers and the analysis focuses on evaluating for performance, reliability, and efficiency measured by:

- **Output Tokens Throughput (tokens/s):** The average number of output tokens returned per second. This metric is important for applications that require high throughput, such as summarization and translation, and easy to compare across different models and providers.
- **Time to first token (TTFT):** The duration of time that LLM returns the first token. TTFT is especially important for streaming applications that require low latency such as chatbots.

Not only is this our first public benchmark - it was a huge success. Meta AI’s Llama 2 70B running on the Groq LPU™ Inference Engine outperformed all other participants at 3-18x faster for output tokens throughput than other cloud-based inference providers.

Let’s walk through the Anyscale methodology in a bit more detail. This benchmark leverages:

- A 550 input token count and a 150 output token count
- The first metric, Output Tokens Throughput (aka the output speed) is determined by dividing the count of output tokens by the overall end-to-end time, which includes input tokens processing time and overall network latency.

On our end, we’d like to note:

- All Llama 2 calculations on the LPU are done in FP16, but we store some of the weights in FP8.
- We have no sparsity (i.e. we’re doing ALL of the Llama 2 matrix calculations and thus processing the entire model as provided by Meta AI).
- This is noteworthy in general as FP16 should provide a higher quality of results for inference.

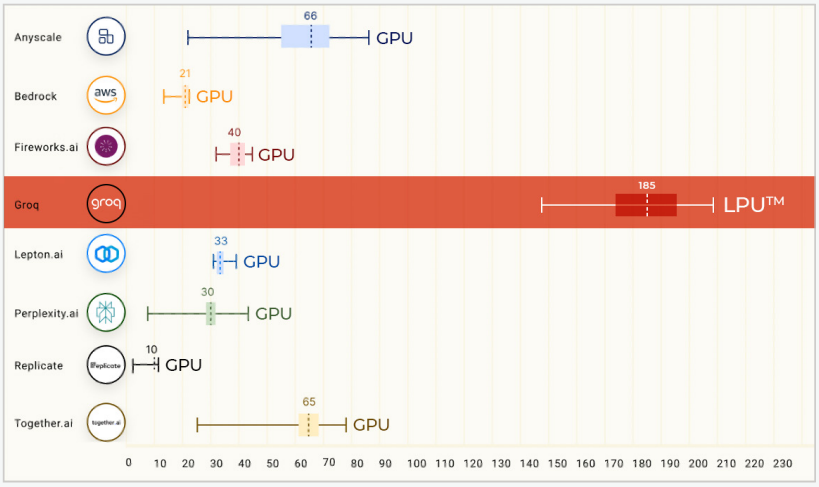
Now let's look a bit more closely at the results for each metric.

For **Output Tokens Throughput**, Groq achieved an average of 185 tokens/s, a result that ranges 3-18x faster than any other inference provider contributing to the leaderboard.

Output Tokens Throughput (tokens/s)

The output tokens throughput is measured as the average number of output tokens returned per second. We collect results by sending 150 requests to each LLM inference provider and calculate the mean output tokens throughput based on 150 requests to each LLM inference provider and calculate the mean output tokens throughput based on 150 requests. A higher output tokens throughout indicates a higher throughput of the LLM inference provider.

70B Models

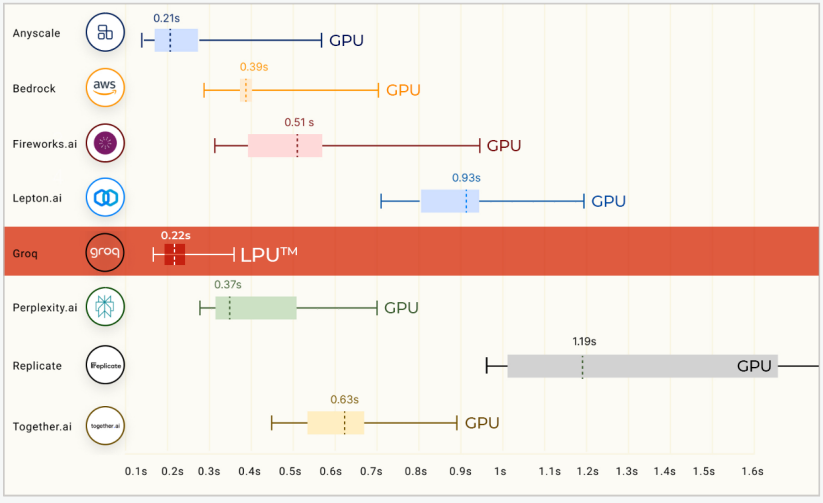


For **Time to First Token**, we hit 0.22s. Because of the deterministic design of the LPU, response times are consistent resulting in our API providing the smallest range of variability. This means more repeatability and less effort designing around potential latency issues or slow responses.

Time to First Tokens (seconds)

For streaming applications, the TTFT is how long before the LLM returns the first token.

70B Models



We're proud and excited to be leading this leaderboard in the initial phase of our ongoing roadmap for performance enhancements.

Now, we already know what you're thinking - "Groq has been saying they're getting 270+ tokens per second per user for Llama-2 70B. What's up with the difference?"

As mentioned, this benchmark leverages a 150 output token count and includes input processing time as part of the calculation, rather than just solely the output tokens throughput. For example, if you were to test with 1000 output tokens, the result would be closer to the 270+ tokens/s per user you see on chat.groq.com.

All in all, we couldn't be more excited to participate in our first public benchmark results with the world thanks to the help of the great team at Anyscale. We look forward to providing benchmarking for Llama 2 7B, and who knows, we just might mix things up, with a variety of experts, beyond that. (Much) more to come.