Supermicro SuperServer SYS-222C-TN with 2x NVIDIA RTX PRO 6000 Blackwell Series GPUs

SUT ID: SMCI260303

STAC-AI™ LANG6 (Inference-Only) Benchmarks

Test date: 26 March, 2026

Release: v1, 24 April, 2026

STAC REPORT

These tests followed STAC benchmark specifications proposed or approved by the STAC Benchmark Council (see www.stacresearch.com). Be sure to check the version of any specification used in a report. Different versions may not yield results that can be compared to one another.

References

[1] https://www.stacresearch.com/SMCI260303

Introduction

STAC recently performed a STAC-AI™ LANG6 (Inference-Only) audit of *the STAC-AI Pack for TensorRT-LLM (Rev D), running on a Supermicro SYS-222C-TN server hosting NVIDIA RTX PRO 6000 GPUs managed by Red Hat OpenShift 4.20*. These tests were conducted at the request of Supermicro, NVIDIA and Red Hat. This report provides test results related to the performance, efficiency, and quality of the Stack Under Test (SUT), and other salient aspects of the test project. An overview of the benchmark specification is included at the end of this document.

Qualified STAC subscribers also have access to further detailed performance, quality and efficiency reports for the SUT, as well as other materials related to this test [1]. Contact STAC for details on subscription programs, and how to access these materials.

Summary of Key Results

In all, the STAC-AI™ LANG6 (Inference-Only) specifications deliver scores of test results, which are detailed in this report and accompanying subscriber-only materials. Supermicro, NVIDIA and Red Hat would like to point out the following:

The NVIDIA RTX PRO 6000 Blackwell Server Edition, tested within the Supermicro SuperServer SYS-222C-TN system, and running TRT-LLM in RedHat Openshift, delivered the following STAC-AI LANG6 batch and interactive results across the reported workloads.

EDGAR4a Batch mode

- *The system achieved 32.9 inferences/s and 5,549 words/s on Llama-3.1-8B EDGAR4a*

EDGAR4a Interactive mode

- *The system achieved a 4.00x increase in arrival rate, from 7.50 to 30.0 inferences/s with:*
  - *increased 95p reaction time by 2.44x, from 0.131 s to 0.320 s,*
  - *increased 95p response time by 4.93x, from 2.96 s to 14.6 s.*
- *At 30.0 inferences/s, the system still operated at about 91% of the 32.9 inferences/s batch-mode rate*

EDGAR5a Batch mode

- *The system achieved 0.345 inferences/s and 139 words/s on Llama-3.1-8B EDGAR5a*

EDGAR5a Interactive mode

- *The system achieved a 4.00x increase in arrival rate, from 0.0800 to 0.320 inferences/s with:*
  - *increased 95p reaction time by 2.96x, from 9.82 s to 29.1 s*
  - *increased 95p response time by 4.58x, from 27.5 s to 126 s*
- *At 0.320 inferences/s, the system still operated at about 93% of the 0.345 inferences/s batch-mode rate*

EDGAR4b: Batch mode

- *The system achieved 5.28 inferences/s and 834 words/s on Llama-3.1-70B EDGAR4b*

EDGAR4b: Interactive mode

- *The system achieved a 4.00x increase in arrival rate, from 1.25 to 5.00 inferences/s with:*
  - *increased 95p reaction time by 2.47x, from 0.916 s to 2.26 s*
  - *increased 95p response time by 2.80x, from 16.0 s to 44.8 s*
- *At 5.00 inferences/s, the system still operated at about 95% of the 5.28 inferences/s batch-mode rate*

Precision of quantized models are described in further detail in configuration disclosure.

Stack Under Test (SUT) Summary

The implementation relied on the following key components for this project:

- STAC-AI™ LANG6 (Inference-Only) Pack for NVIDIA TensorRT-LLM (Rev D)
- NVIDIA TensorRT-LLM 1.2.0rc2 with PyTorch backend
- NVIDIA TensorRT 10.13.3.9
- NVIDIA Model Optimizer (nvidia-modelopt) 0.37.0 for NVFP4 quantization
- PyTorch 2.9.0a0 (NVIDIA PyTorch container 25.10)

- Red Hat Enterprise Linux CoreOS 9.6
- Red Hat OpenShift Container Platform 4.20

- Supermicro Super Server SYS-222C-TN (2U CloudDC with DC-MHS)
  - 32 x 64GiB DDR5 DIMMs @ 5200MTs (2TiB total)
  - 2 x Intel® Xeon® 6730P CPUs
- 2x NVIDIA RTX PRO 6000 Blackwell Series GPUs, each with 96GiB of memory

The STAC Configuration Disclosure for the SUT in this report is also available [1] to qualified STAC subscribers, providing the exact product version numbers, detailed tuning and configuration options, and other important information.

Report Card(s)

The key results are summarized in the Report Card table(s) below. Batch-mode and Interactive-mode test cases (if any) are reported separately.

STAC-AI™ LANG6 (Inference-Only)

Supermicro SuperServer SYS-222C-TN with 2 x NVIDIA RTX PRO 6000 Blackwell Series GPUs

SUT ID: SMCI260303

Batch-Mode Report Card

* = STAC-AI.LANG6.[Model].[Data Set]

Model	Llama-3.1-8B (NVFP4)		Llama-3.1-70B (NVFP4)	
Data Set	EDGAR4a	EDGAR5a	EDGAR4b	EDGAR5b
SUT Variant	TLLM	TLLM	TLLM	TLLM
***.BATCH.INF_RATE.v1** Inference Rate Inferences / sec	32.9	0.345	5.28	0.0411
***.BATCH.TPUT.v1** Throughput Words / sec	5,549	139	834	13.2
***.BATCH.LOAD.v1** Load Time seconds	137	33.4	46.4	48.2
***.BATCH.FIDELITY.v1** Fidelity, %	89.55%	89.48%	91.29%	72.22%
***.BATCH.WCR.v1** Total Word Count Ratio, %	103.1%	105.6%	98.0%	87.1%
***.BATCH.ENERG_EFF.v1** Energy Efficiency Words / kWh	9.320M	234.6K	1.358M	22.34K
***.BATCH.SPACE_EFF.v1** Space Efficiency Words / (ft³·hour)	9.501M	237.4K	1.428M	22.61K

Source: STAC
www.STACresearch.com

Batch-Mode Report Card

STAC-AI™ LANG6 (Inference-Only)

Supermicro SuperServer SYS-222C-TN with 2 x NVIDIA RTX PRO 6000 Blackwell Series GPUs

SUT ID: SMCI260303

Interactive-Mode Report Card

* = STAC-AI.LANG6.[Model].[Data Set]

Model	Llama-3.1-8B (NVFP4)												Llama-3.1-70B (NVFP4)					
Data Set	EDGAR4a						EDGAR5a						EDGAR4b					
SUT Variant	E4aI						E5aI						E4bI					
Lambda	7.50	15.0	22.5	28.0	29.0	30.0	0.0800	0.160	0.240	0.280	0.300	0.320	1.25	2.50	3.75	4.60	4.80	5.00
*.INTERACTIVE.TPUT.v1 Throughput Words / sec	1,263	2,522	3,775	4,686	4,851	5,013	31.9	64.2	96.1	113	121	128	196	391	578	696	717	743
*.INTERACTIVE.REACT.v1 Median Reaction Time seconds	0.0555	0.0623	0.0812	0.118	0.132	0.153	3.65	4.34	5.53	6.76	7.91	9.66	0.313	0.349	0.480	0.628	0.685	0.796
*.INTERACTIVE.RESP.v1 Median Response Time seconds	2.08	2.63	3.96	6.62	7.64	9.16	12.7	22.6	38.1	52.8	65.3	82.3	11.7	12.4	17.0	23.3	25.9	29.5
*.INTERACTIVE.OUT_RATE.v1 5p Output Rate Words / second	72.3	53.9	32.9	18.6	16.1	13.7	19.0	9.94	5.94	4.69	4.32	4.00	12.1	10.9	7.55	5.26	4.64	4.10
*.INTERACTIVE.OUT_PROF.v1 5p Output Profile Words / second	69.5	50.0	29.7	16.7	14.3	12.1	10.2	4.00	3.25	3.13	3.05	3.00	10.9	9.00	5.75	3.80	3.30	3.00
*.INTERACTIVE.LOAD.v1 Load Time seconds	33.2	33.2	33.2	33.2	33.2	33.2	39.9	39.9	39.9	39.9	39.9	39.9	46.0	46.0	46.0	46.0	46.0	46.0
*.INTERACTIVE.FIDELITY.v1 Fidelity, %	89.61%	89.48%	89.96%	89.65%	89.72%	89.56%	89.98%	88.10%	87.68%	87.73%	88.25%	89.50%	92.45%	91.60%	92.20%	91.44%	90.95%	92.15%
*.INTERACTIVE.WCR.v1 Total Word Count Ratio, %	103.1%	103.1%	103.1%	103.2%	103.2%	103.2%	104.8%	105.2%	104.8%	105.9%	106.2%	105.9%	97.9%	98.0%	98.0%	98.0%	98.0%	98.0%
*.INTERACTIVE.ENERG_EFF.v1 Energy Efficiency Words / kWh	2.478M	5.551M	7.392M	8.343M	8.465M	8.579M	90.83K	132.6K	174.5K	200.6K	214.3K	221.6K	319.4K	719.1K	987.1K	1.147M	1.173M	1.206M
*.INTERACTIVE.SPACE_EFF.v1 Space Efficiency Words / (ft³·hour)	2.162M	4.318M	6.464M	8.023M	8.306M	8.584M	54.60K	110.0K	164.5K	192.9K	207.3K	218.7K	336.1K	668.8K	989.4K	1.191M	1.228M	1.271M

Source: STAC
www.STACresearch.com

Interactive-Mode Report Card

Commentary on the Results

Supermicro, NVIDIA and Red Hat would also like to point out the following regarding the test results reported in this document:

> *These results show the NVIDIA RTX PRO 6000 Blackwell Server Edition delivering STAC-AI LANG6 inference in a compact 2U Supermicro SuperServer SYS-222C-TN configuration that completed the full reported workload set with just two GPUs.*
>
> *In this tested stack, NVIDIA TensorRT-LLM and NVIDIA Model Optimizer were used with NVFP4 quantization, illustrating how Blackwell-class inference capabilities can be deployed in a smaller, more flexible on-prem server form factor than is typical of larger accelerator platforms.*
>
> *The clearest benchmark advantages for this configuration appear in efficiency per unit of rack space: batch space efficiency reached 9.501M words/ft-hour on EDGAR4a, 1.428M words/ft-hour on EDGAR4b, and 237.4K words/ft-hour on EDGAR5a, exceeding the corresponding GH200 results of 6.148M, 799.5K, and 226.9K words/ft-hour on the same workloads.*
>
> *The system also maintained interactive arrival rates close to its batch-mode inference rates, reaching 30.0 versus 32.9 inferences/s on EDGAR4a, 0.320 versus 0.345 on EDGAR5a, and 5.00 versus 5.28 on EDGAR4b. These tests were run on Red Hat OpenShift Container Platform 4.20, with Red Hat's STAC-AI operator automating benchmark deployment and execution, so the reported results reflect the tested containerized software environment used for this submission.*

The STAC Pack

Apart from generally available proprietary or open-source software, all the programs, scripts, config files and documentation required for a benchmark solution are known as a *STAC Pack*. This project used the *STAC-AI™ Pack for TensorRT-LLM (Rev D)*.

The source code is available to qualified STAC subscribers. NVIDIA provided the following description:

> *The STAC-AI™ Pack for TensorRT-LLM is a comprehensive benchmarking suite developed by the Strategic Technology Analysis Center (STAC) to evaluate the performance, efficiency, and scalability of AI inference workloads—particularly large language models (LLMs)—on modern accelerator and server architectures. This benchmark suite leverages NVIDIA's TensorRT-LLM, a high-performance deep learning inference library optimized for transformer-based models, to measure real-world throughput, latency, and power efficiency across diverse model sizes and configurations.*
>
> *STAC-AI Packs are designed to provide vendor-neutral, industry-validated performance results, enabling financial institutions, research centers, and enterprise AI users to objectively compare AI infrastructure platforms. The TensorRT-LLM Pack specifically assesses decoder-only and encoder-decoder transformer workloads, focusing on low-latency inference, batch scalability, and system-level efficiency.*
>
> *By standardizing test methodologies across different hardware and software stacks, the STAC-AI Pack for TensorRT-LLM delivers a trusted framework for evaluating GPU and CPU configurations, memory architectures, and containerized AI deployment environments. It helps organizations identify optimal system designs for production-grade AI inference, ensuring consistent performance metrics across vendors and architectures such as NVIDIA RTX6000 BSE.*

Supermicro provided the following commentary about the products used in this SUT:

The STAC-AI™ benchmark conducted on the Supermicro SuperServer SYS-222C-TN with the STAC-AI Pack for TensorRT-LLM (Rev D) underscores Supermicro's engineering expertise in delivering enterprise-grade AI inference performance.

The STAC-AI™ benchmark was conducted using the STAC-AI Pack for TensorRT-LLM (Rev D) on a Supermicro SuperServer SYS-222C-TN 2U server, configured to deliver impressive AI inference performance. The system under test featured TensorRT-LLM version 1.2.0rc2, running in a Red Hat OpenShift Container Platform 4.20 environment on Red Hat Enterprise Linux CoreOS 9.6. It was powered by dual Intel® Xeon® 6700/6500 series CPUs and equipped with two NVIDIA RTX 6000 BSE GPUs, each with 96 GB of GDDR7 memory, enabling high-throughput, low-latency large language model inference. The server was populated with 32 DIMM slots of ECC DDR5 memory running at 5200 MT/s, providing substantial system memory capacity. This configuration showcases the SYS-222C-TN's capability to support memory and compute-intensive AI workloads, particularly in real-time inference scenarios.

Commentary:
This benchmark configuration represents the cutting edge of AI inferencing infrastructure, validating Supermicro's SuperServer SYS-222C-TN as a balanced and versatile platform that bridges data center scalability with AI performance efficiency in a compact 2U form factor. The combination of two NVIDIA RTX 6000 GPUs and high-capacity DDR5 memory positions the system to handle the increasing computational and memory demands of LLM inference, especially for enterprise-grade generative AI deployments. Furthermore, the dual Intel Xeon 6700/6500 series CPUs ensure optimized data orchestration and throughput between compute and accelerator resources. The use of TensorRT-LLM 1.2.0rc2 further reinforces performance tuning for low-latency, high-throughput inferencing across transformer-based models. Overall, this setup highlights Supermicro's engineering focus on enabling high-performance, energy-efficient AI systems designed to accelerate inference workloads across industries—from finance and healthcare to manufacturing and research.

Red Hat provided the following commentary about the platform used in this SUT:

The SUT runs on Red Hat OpenShift Container Platform 4.20, deployed as a Single Node OpenShift (SNO) configuration on Red Hat Enterprise Linux CoreOS 9.6. OpenShift provides the container orchestration and lifecycle management layer. In this audited STAC-AI result, the platform delivered performance consistent with Red Hat's expectation — informed by prior audited STAC benchmark submissions (STAC-N1 and STAC-A2) — of bare-metal-like performance on OpenShift, while enabling containerized Kubernetes operations. As in those prior submissions, the container and orchestration layers did not appear to introduce material performance limitations in practice.

This architecture allows STAC-AI to achieve the full performance potential of the underlying hardware with negligible orchestration overhead. These results reaffirm the bare-metal-equivalent efficiency Red Hat has consistently demonstrated across prior STAC benchmarks (STAC-N1, STAC-A2, and now STAC-AI), proving that the containerization layer does not impede raw compute throughput.

The STAC-AI operator, developed by Red Hat, automates the full benchmark lifecycle on OpenShift — from workspace provisioning and model quantization through benchmark execution and results collection. The same OpenShift deployment model applies equally to workstation-class GPUs at the edge and datacenter-class GPU clusters, giving organizations a unified platform from edge to datacenter.

SUT Details

Supermicro provided the following commentary about the products used in this SUT:

The Supermicro CloudDC SuperServer SYS-222C-TN delivers high-performance computing, empowering organizations to validate AI inference and analytics workloads with speed and efficiency. Built on a robust 2U rackmount platform with DC-MHS compliance, it features dual Intel® Xeon® 6700/6500 series processors—offering up to 86 cores/172 threads per CPU with P-cores or 144 cores/144 threads with E-cores—combined with massive memory scalability reaching 4TB of DDR5-6400 across 32 DIMM slots for seamless handling of complex time-series data and model processing. Its expansive storage architecture includes 24 hot-swap 2.5" NVMe/SATA/SAS drive bays plus dual M.2 PCIe 5.0 slots, providing the high-bandwidth, low-latency I/O suitable for data-intensive analytics, benchmarking and other performance-sensitive workloads evaluated under the STAC-AI test framework. With flexible PCIe 5.0 expansion—up to six x16 FHFL slots and two AIOM slots—this system supports up to four single-width or two double-width GPUs, including, NVIDIA H200 NVL, L40S, and Blackwell Server Editions, to accelerate AI model inference while maintaining optimized power efficiency through redundant 2000W Titanium-level PSUs. Engineered for cloud, HPC, and CSP environments with OpenBMC-based management and full server automation tools, the SYS-222C-TN ensures effortless deployment, serviceability, and scalability in production data centers. The SUT for STAC-AI testing sets the benchmark for low-latency, high-throughput AI performance, helping financial institutions and enterprises achieve inference accuracy, energy efficiency, and overall system responsiveness.

Contributors and Roles

- Supermicro
- NVIDIA
- Red Hat
- STAC

The Project Participants had the following responsibilities:

- Supermicro provided the SYS-222C-TN system, provided lab space and setup, and sponsored this report.
- NVIDIA wrote the STAC Pack in accordance with the benchmark specifications.
- Red Hat provided, installed, and configured Red Hat OpenShift Container Platform; wrote the STAC-AI Kubernetes operator that automates the full benchmark lifecycle on OpenShift; and integrated the Yokogawa WT1804R power monitoring and DS18B20 temperature monitoring solutions.
- STAC inspected and tested the STAC Pack for conformance to specifications; inspected the test system; executed the tests; and prepared the STAC Report and STAC Configuration Disclosure.

Contacts

- Supermicro: chloelee@supermicro.com, shahzadas@supermicro.com
- NVIDIA: Martin Marciniszyn Mehringer (martinma@nvidia.com), Dan Blanaru (dblanaru@nvidia.com)
- Red Hat: Douglas Shakshober (dshaks@redhat.com), Sebastian Jug (sejug@redhat.com
- STAC: info@STACresearch.com

Detailed Results

The remainder of this report summarizes the performance and efficiency results of the SUT. Qualified STAC subscribers have access to more detailed analysis of these test results, including more details of the quality tests. Please contact info@STACresearch.com for information on how to obtain access to these results.

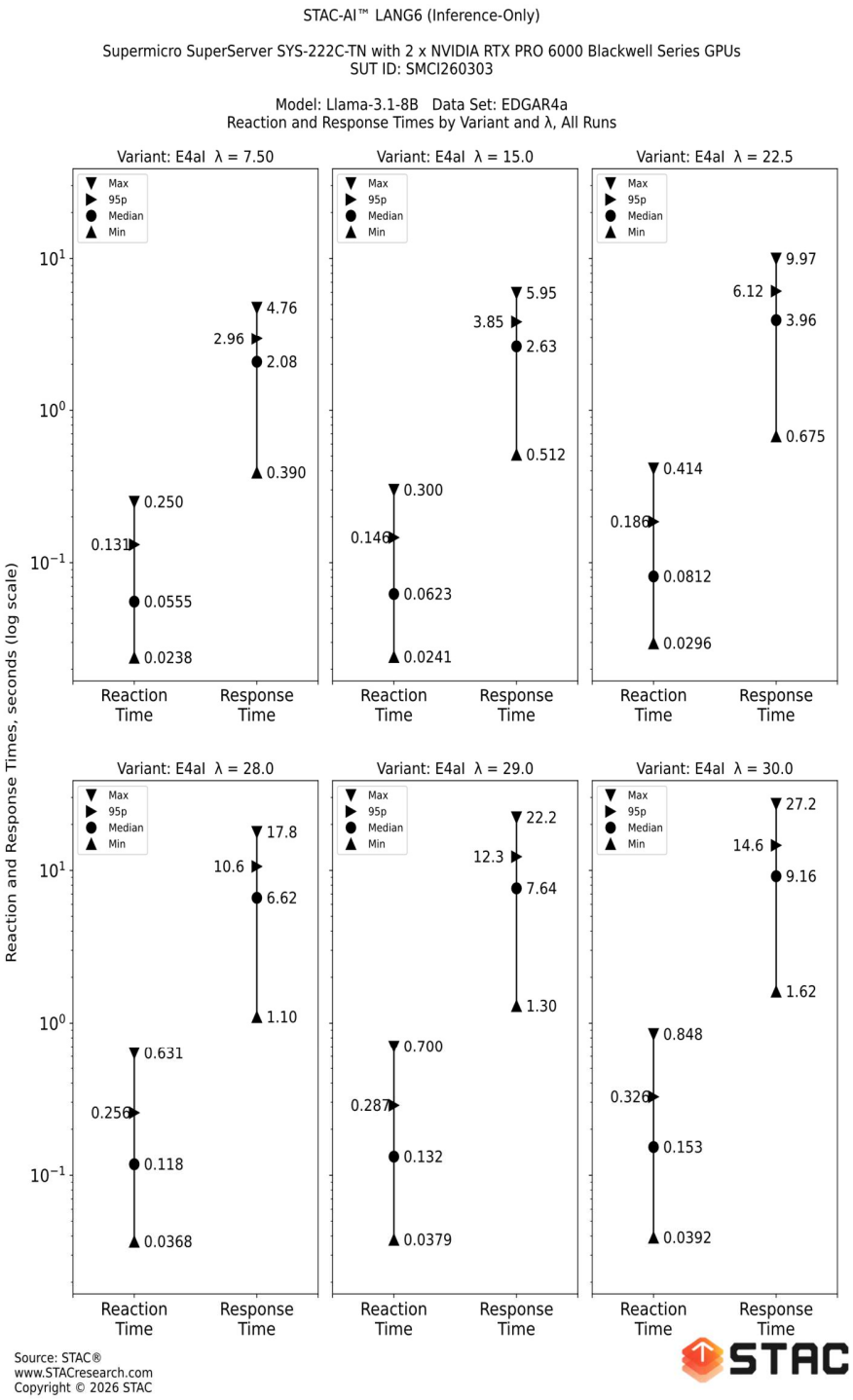
Performance Results

Key performance metrics for the SUT were previously presented in the Report Card table(s) above. No other performance details are available for Batch-mode workloads (if any).

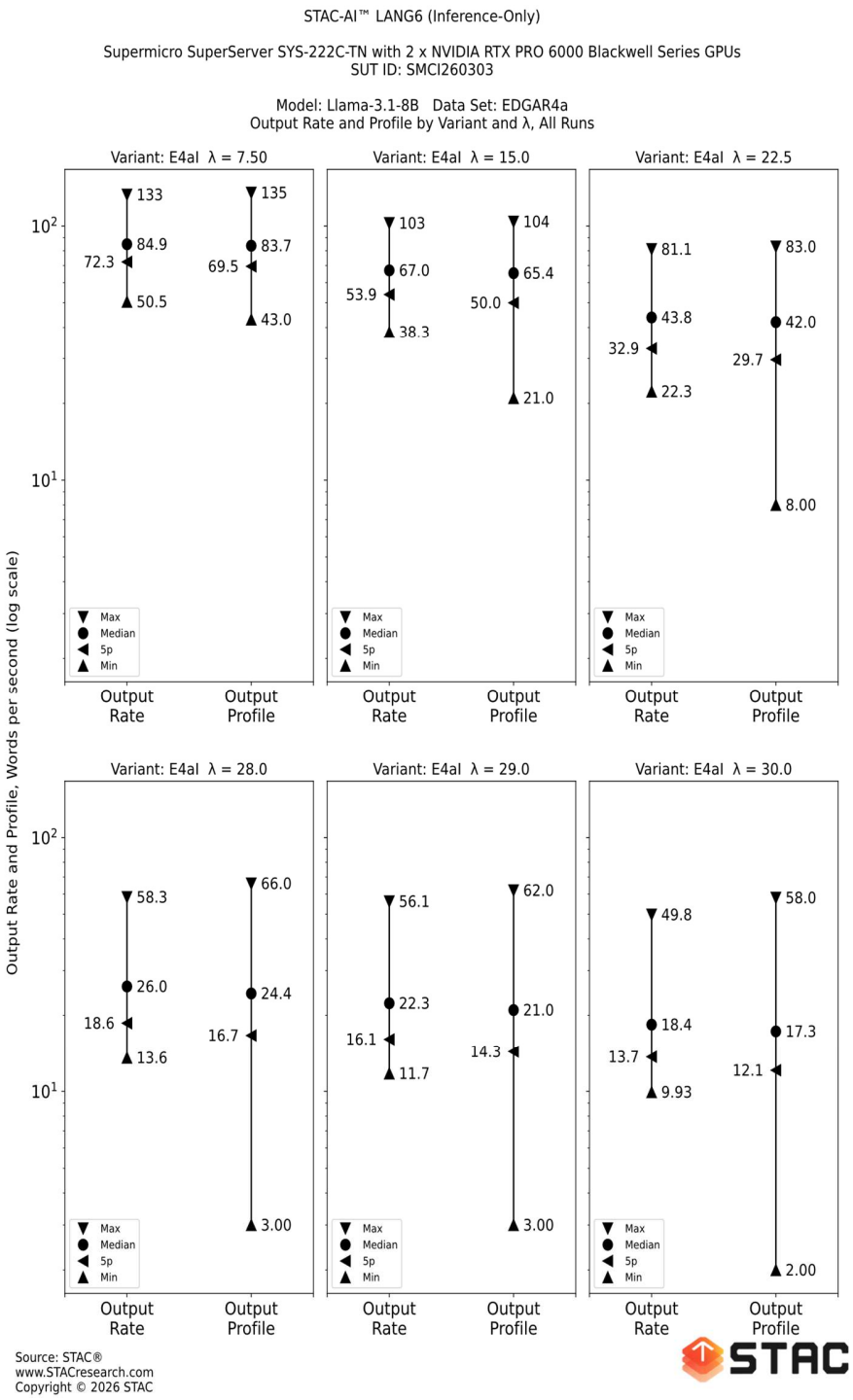
Below are graphics that outline the distributions of selected latency and throughput metrics for Interactive-mode workloads. Pairs of graphics are provided for each workload tested. The first plot for each workload details Reaction Time and Response Time. The second plot details Output Rate and Output Profile.

Note that it is possible to obtain individual Output Profile readings of 0.0. In these cases the minima of the Output Profile distributions cannot be represented on the logarithmically-scaled Y-axes of the plots.

Llama-3.1-8B + EDGAR4a

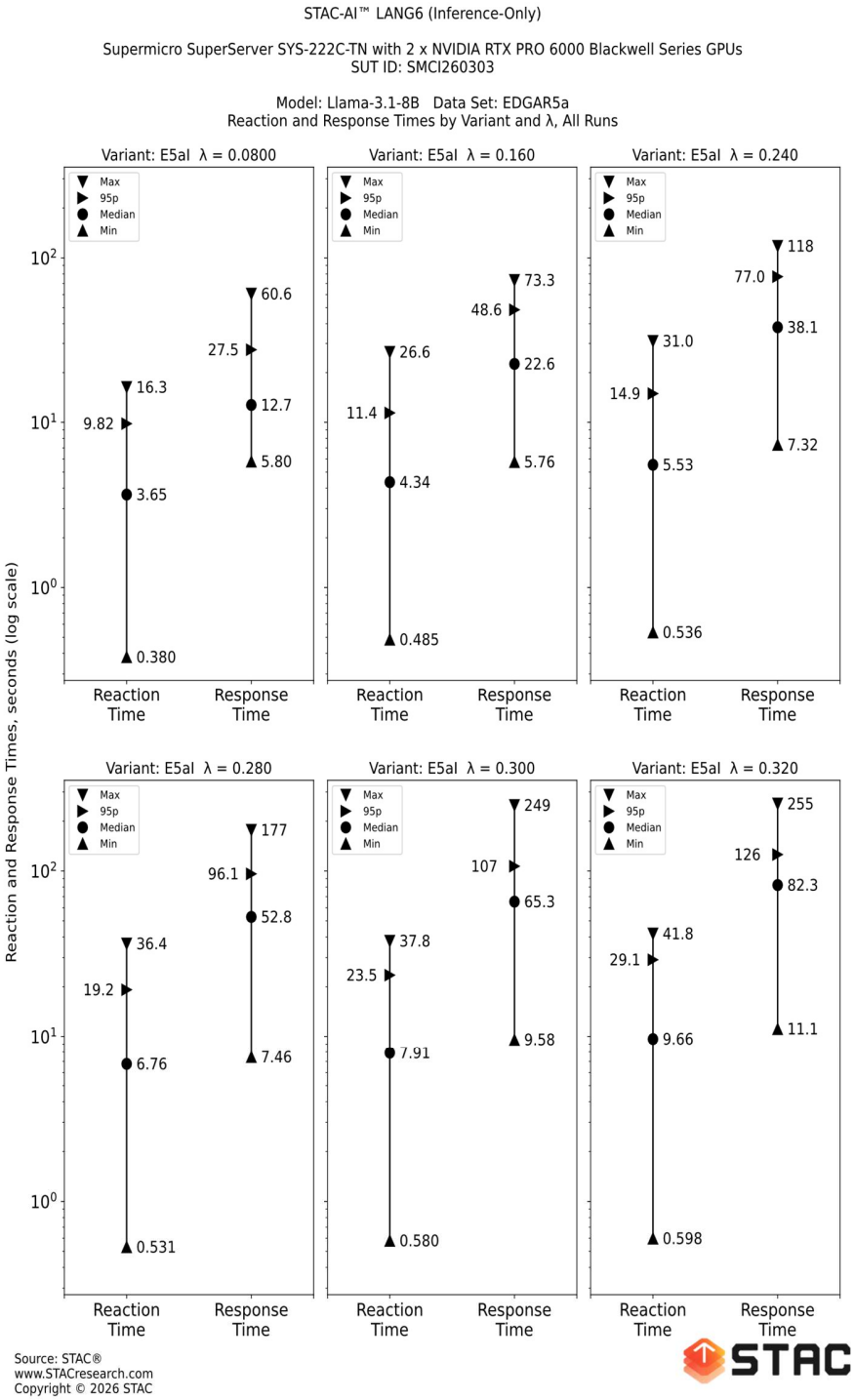


Llama-3.1-8B + EDGAR4a: Reaction and Response Times

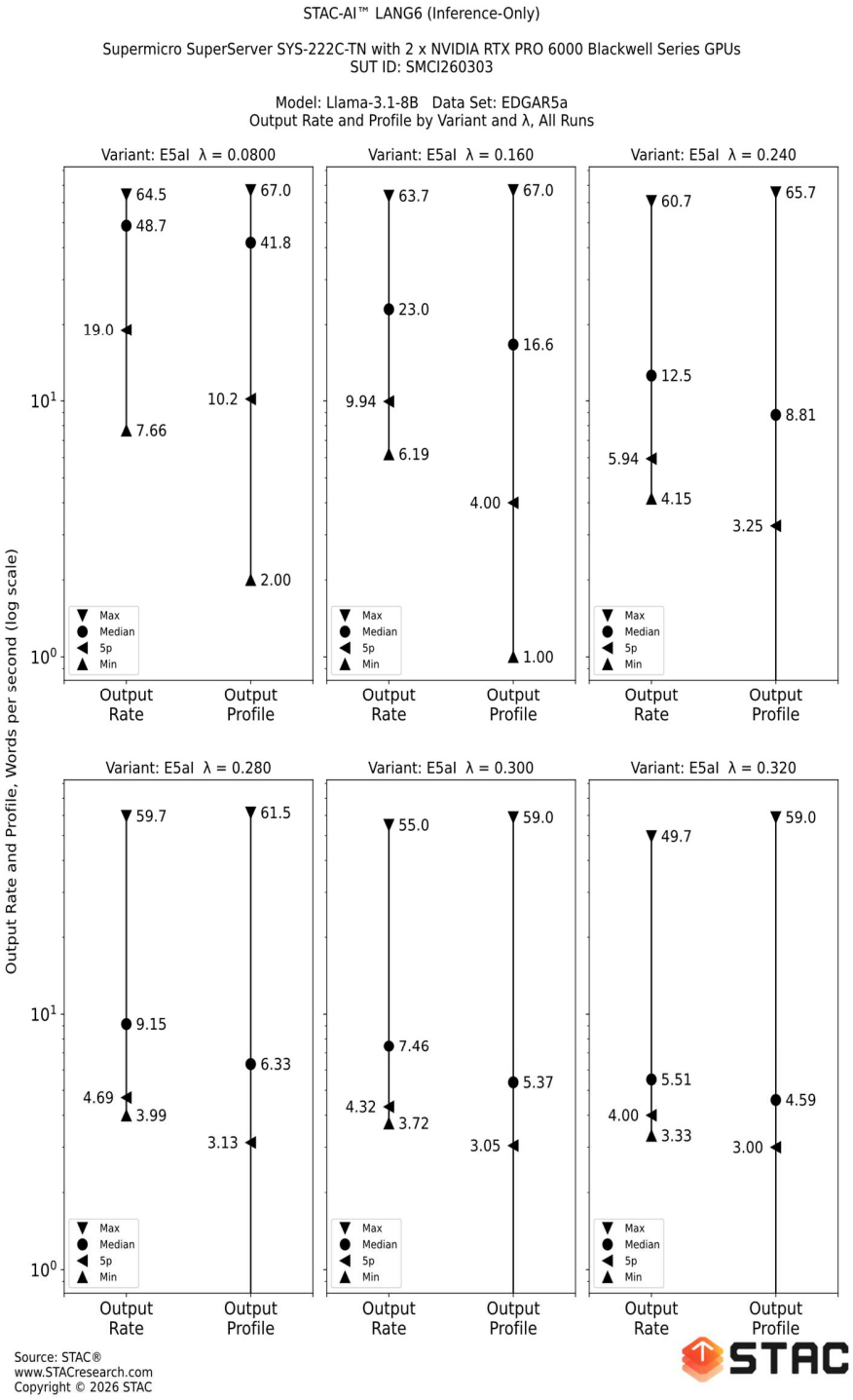


Llama-3.1-8B + EDGAR4a: Output Rate and Profile

Llama-3.1-8B + EDGAR5a

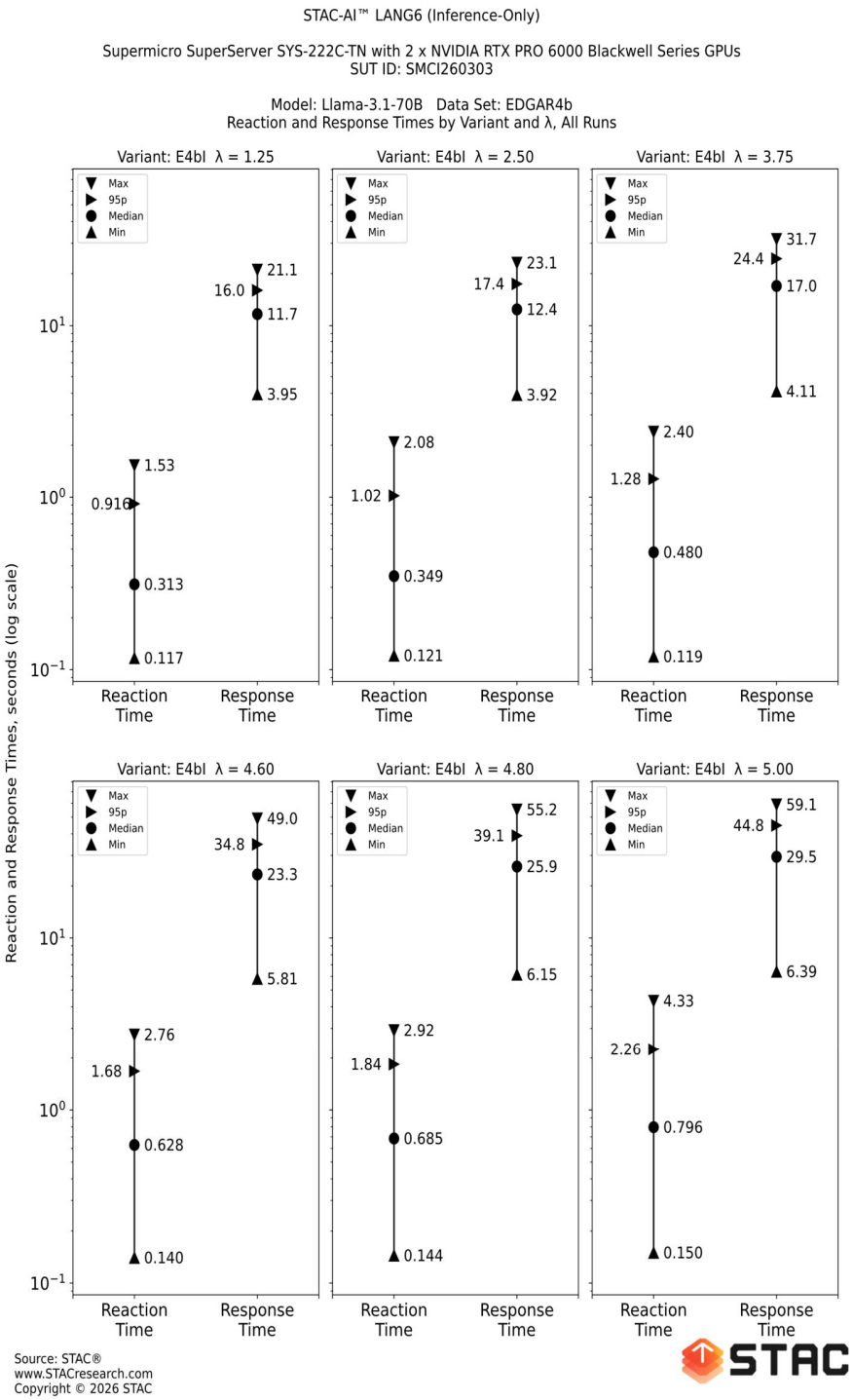


Llama-3.1-8B + EDGAR5a: Reaction and Response Times

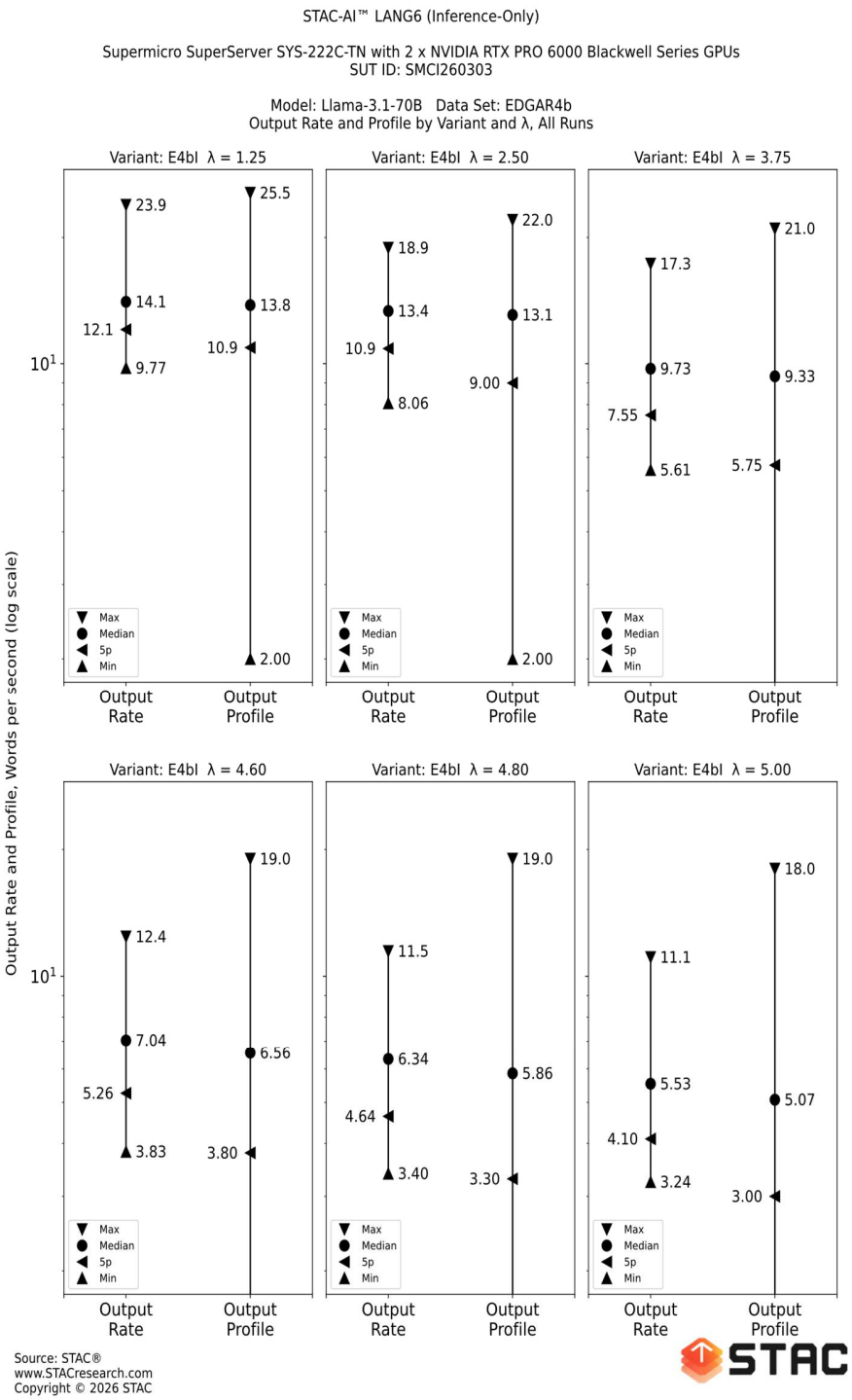


Llama-3.1-8B + EDGAR5a: Output Rate and Profile

Llama-3.1-70B + EDGAR4b



Llama-3.1-70B + EDGAR4b: Reaction and Response Times



Llama-3.1-70B + EDGAR4b: Output Rate and Profile

Efficiency Results

For on-prem systems, power consumption and space are the basis for efficiency metrics. Efficiency summaries are reported separately for Batch and Interactive Workloads (if any).

STAC-AI™ LANG6 (Inference-Only)

Supermicro SuperServer SYS-222C-TN with 2 x NVIDIA RTX PRO 6000 Blackwell Series GPUs

SUT ID: SMCI260303

Batch Efficiency Summary
Batch-Mode Setups

* = STAC-AI.LANG6.[Model].[Data Set]

Model	Llama-3.1-8B		Llama-3.1-70B	
Data Set	EDGAR4a	EDGAR5a	EDGAR4b	EDGAR5b
SUT Variant	TLLM	TLLM	TLLM	TLLM
***.Batch.TPUT.v1* Throughput Words / sec	5,549	139	834	13.2
Setup Power kW	2.14	2.13	2.21	2.13
***.Batch.ENERG_EFF.v1* Energy Efficiency Words / kWh	9.320M	234.6K	1.358M	22.34K
Effective Volume ft³	2.1025	2.1025	2.1025	2.1025
***.Batch.SPACE_EFF.v1* Space Efficiency Words / (ft³·hour)	9.501M	237.4K	1.428M	22.61K

Source: STAC
www.STACresearch.com

Efficiency: Batch-Mode Setups

STAC-AI™ LANG6 (Inference-Only)

Supermicro SuperServer SYS-222C-TN with 2 x NVIDIA RTX PRO 6000 Blackwell Series GPUs

SUT ID: SMCI260303

Interactive Efficiency Summary
Interactive-Mode Setups

* = STAC-AI.LANG6.[Model].[Data Set]

Model	Llama-3.1-8B												Llama-3.1-70B					
Data Set	EDGAR4a						EDGAR5a						EDGAR4b					
SUT Variant	E4aI	E4aI	E4aI	E4aI	E4aI	E4aI	E5aI	E5aI	E5aI	E5aI	E5aI	E5aI	E4bI	E4bI	E4bI	E4bI	E4bI	E4bI
Poisson Arrival Rate (λ) Inferencess / sec	7.50	15.0	22.5	28.0	29.0	30.0	0.0800	0.160	0.240	0.280	0.300	0.320	1.25	2.50	3.75	4.60	4.80	5.00
*.Interactive.TPUT.v1 Throughput Words / sec	1,263	2,522	3,775	4,686	4,851	5,013	31.9	64.2	96.1	113	121	128	196	391	578	696	717	743
Setup Power kW	1.83	1.64	1.84	2.02	2.06	2.10	1.26	1.74	1.98	2.02	2.03	2.07	2.21	1.96	2.11	2.18	2.20	2.22
*.Interactive.ENERG_EFF.v1 Energy Efficiency Words / kWh	2.478M	5.551M	7.392M	8.343M	8.465M	8.579M	90.83K	132.6K	174.5K	200.6K	214.3K	221.6K	319.4K	719.1K	987.1K	1.147M	1.173M	1.206M
Effective Volume ft³	2.1025	2.1025	2.1025	2.1025	2.1025	2.1025	2.1025	2.1025	2.1025	2.1025	2.1025	2.1025	2.1025	2.1025	2.1025	2.1025	2.1025	2.1025
*.Interactive.SPACE_EFF.v1 Space Efficiency Words / (ft³·hour)	2.162M	4.318M	6.464M	8.023M	8.306M	8.584M	54.60K	110.0K	164.5K	192.9K	207.3K	218.7K	336.1K	668.8K	989.4K	1.191M	1.228M	1.271M

Source: STAC
www.STACresearch.com

Efficiency: Interactive-Mode Setups

Benchmark Overview

Business Context

Large Language Models (LLMs) have taken the world by storm, revolutionizing knowledge work by enabling machines to understand and generate human-like text. Their ability to process and analyze vast amounts of data allows financial firms to automate tasks, generate insights, and interact with customers in more personalized ways.

However, constructing the hardware and software infrastructure to support LLMs in a way that achieves the quality, responsiveness, and cost objectives of a business is a significant technical challenge. LLMs and related technology are advancing at a breathtaking pace, so understanding which components to choose and how to configure them for a desired set of tradeoffs requires rigorous testing.

The STAC-AI™ Working Group seeks to simplify this testing challenge by defining benchmarks for LLM solutions that provide information that engineers will find valuable. While the word "benchmark" in the context of LLMs often refers to how well an LLM can answer a class of questions or create human-quality responses, the initial goal for STAC-AI is to be an infrastructure performance benchmark, not a data science challenge. STAC-AI begins at the point at which a hypothetical data science team has identified the LLM they wish to use for a given dataset and type of query, and it is now the job of IT to construct infrastructure in which to host that model that will deliver the required performance and efficiency without sacrificing the quality that the data scientists established.

Some use cases call for minimum latency, while others call for maximum efficiency at an acceptable latency. A single solution may be configurable to provide different tradeoffs. For a given configuration, the goal of the benchmarks is to measure the upper-bound of performance and efficiency, thus elucidating the theoretical limits that a solution involving more functions could provide in the real world.

The first solution pattern the Working Group chose to tackle is retrieval-augmented generation (RAG). By supplying an LLM with pertinent information retrieved from a database or other authoritative source, RAG enables the LLM to generate more accurate, contextually relevant, and factually grounded outputs. RAG is particularly useful in areas requiring precise or up-to-date information.

The Working Group identified several measurable steps of an end-to-end RAG workflow, ranging from document embedding in isolation to a full end-to-end RAG pipeline with online document ingestion. While RAG performance in the real world depends on several steps, the group decided to focus initially on the sixth step identified: LLM inference. This step is typically the slowest and most costly. The benchmarks measure performance of inference after earlier steps in a RAG pipeline have retrieved documents and constructed the input to the LLM (loosely speaking, the *prompt*; more precisely, the *context*) and before delivery to any downstream components such as post-processors or client displays. This scope explains the full name of the benchmark suite: STAC-AI™ LANG6 (Inference-Only). For brevity, the rest of this overview will refer to the benchmark simply as STAC-AI.

STAC-AI is simple to implement and run. The STAC-AI Test Harness software handles the supply of requests, as well as all timing logic and file I/O. It uses off-the-shelf open-source LLMs and simple integration of the open-source or proprietary inference engine of choice via a small Python module.

The SUT (Stack Under Test)

Like other STAC Benchmarks, STAC-AI is conceived as specific combinations of workloads and metrics applied to a "stack under test" (SUT). A *SUT* comprises all hardware and software used to execute the benchmarks with the STAC Test Harness software. In addition to an inference engine and computing resources, this includes a *SUT adapter,* which is a small Python module that translates the protocols of the Test Harness into the protocols of the SUT's inference engine. For published results, STAC performs a code review of the SUT adapter code to ensure conformance with the specs. Given the nature of LLM inference and of the Benchmark tests, we would not expect the SUT adapter to have a material effect on the performance or efficiency of the SUT.

A *SUT variant* denotes a particular software configuration of the SUT required to execute a unique model configuration or chosen to illustrate one possible set of tradeoffs among latency, throughput, and/or efficiency. For example, one variant of a SUT may use one accelerator per model instance, while another variant may use two accelerators per model instance. All variants of a given SUT must use substantially the same types and amounts of hardware resources. For example, if a report includes a SUT variant with 8 accelerators, then all SUT variants in the report must use 8 accelerators. Details of the SUT variants named in this report appear in the associated Configuration Disclosure.

Models

A *model* refers to a particular LLM. For the initial STAC-AI, the Working Group chose 2 models whose size, performance, and quality are thought to be a good match for typical workloads in finance: Llama-3.1-8B-Instruct and Llama-3.1-70B-Instruct. We expect the working group to introduce new models on a periodic basis as needs and technology change.

Model	Reference	Reference Data Format	Model Size	Context
Llama-3.1-8B-Instruct	https://huggingface.co/meta-llama/Meta-Llama-3.1-8B-Instruct	BF16	8.03B parameters	128K
Llama-3.1-70B-Instruct	https://huggingface.co/meta-llama/Meta-Llama-3.1-70B-Instruct	BF16	70.6B parameters	128K

Models are typically delivered with relatively high-precision data formats, such as Bfloat16. However, many novel reduced-precision schemes have been developed to reduce memory requirements and/or improve performance. STAC-AI allows any quantization scheme to be used to modify the model parameters as long as:

- The quantization scheme is fully disclosed to STAC;
- The number of model parameters is not modified (e.g., no pruning); and
- The quantized model meets some minimum quality standards discussed under Fidelity, below.

Data Set

A *data set* pairs prompt templates with RAG documents that are inserted into the prompt template to form the final prompt. The following table summarizes the current supported data sets.

Name	Prompt Type	Document Type
EDGAR4	Summarization of the relationship of a company to one of various physical and financial concepts such as commodities, currencies, interest rates and real estate sectors.	EDGAR 10-K paragraphs from a single security filing for a single year
EDGAR5	A set of questions covering several different aspects of a complete 10-K filing	Complete text of a single EDGAR 10-K filing

These data sets are based on content obtained from EDGAR, the US Securities and Exchange Commission's public database of securities filings. The prompts involve analysis and summarization of annual reports (10-K filings) for thousands of public companies going back 5 years, representing common tasks in financial firms. New workloads may be introduced in the future as well.

Each dataset comes in two variants, one for the 8B model and one with far fewer prompts for the 70B model, based on performance differences STAC observed on a reference SUT.

Each data set uses a static set of prompt templates. However, the Benchmark includes some tests that utilize only a fraction of a model's context size (input capacity), as well as others that use nearly the full context. Moreover, since the numbers and sizes of the documents vary a great deal, as do the sizes of generated responses, the Benchmark data sets still exhibit a range of performance behaviors.

Data Set	Class	Prompts	Model Target	Median Prompt Words	Median Response Words	Request Mode(s) Supported
EDGAR4	a	21,816	Llama-3.1-8B-Instruct	1197	167	Batch Interactive
	b	3,500	Llama-3.1-70B-Instruct	1177	163	Batch Interactive
EDGAR5	a	898	Llama-3.1-8B-Instruct	44029	370	Batch Interactive
	b	50	Llama-3.1-70B-Instruct	44288	373	Batch

The STAC-AI specs require a minimum run time to guarantee accurate efficiency metrics. All SUTs run a data set in its entirety; fast SUTs may run a data set an integral number of times.

Request Modes

STAC-AI supports two distinct inference processing modes, corresponding to the most common use cases identified by the working group:

- Batch Mode, in which all inference requests for a test run are provided to the SUT with a single API call, and all inference results are returned from that single API call.

- Interactive Mode, in which inference requests arrive at pseudo-random times. Each inference response is recorded individually (sometimes even token by token). The working group specified that interactive requests should follow a Poisson arrival process, since this mathematical model accurately represents the random nature of user-initiated requests in many real-world systems. The key parameter of this process is λ, which corresponds to the mean arrival rate, denominated in inference requests/second.

  The SUT must not drop requests and must internally retry failed requests an unlimited number of times. Unlike LLM benchmarks that require the SUT to meet an arbitrary latency bound for interactive tests, no artificial bounds exist in this Benchmark. The SUT provider is free to test any set of λ parameters, and the Benchmark simply reports the performance, efficiency, and quality results for each λ. However, the specs require that it be obvious from the results that the SUT is capable of processing requests at the arrival rate λ indefinitely.

Workloads and Setups

Each STAC-AI *workload* consists of a model, data set, and request mode. A *setup* is defined as a workload running on a SUT variant, plus an interactive λ if appropriate. Due to the wide variety of SUTs and target use cases—and the large performance differences between models—SUT providers are free to test any setup. However, competitive comparisons are only allowed between SUTs running the same workload (e.g., Llama-3.1-8B-Instruct with the EDGAR4a data set in batch mode).

Metrics

STAC-AI measures load time, latency, throughput, efficiency, and fidelity. The *measurement set* refers to all inferences that are measured. Interactive-mode runs include warmup inferences and other inferences that are not measured for performance and efficiency. The *measurement interval* is the amount of time taken by the SUT to process the measurement set. The *measurement period* is the period encompassing the measurement interval.

Load Time

Load Time is the mean across test runs of the time it takes for the SUT to load model parameters and signal that it is ready to begin inference (batch mode) or return the first token of an inference request (interactive mode). It is not feasible to reboot the system before each measurement of the Load Time. Instead, the specs detail how the SUT and any accelerators must not cache model parameters between runs, and the Load Time is always measured after flushing the file system cache. A Load Time is computed for each unique combination of model, data set, request mode and SUT variant.

Latency

Two latency metrics are reported for interactive-mode tests. These metrics are measured once per inference, are reported as quantiles and may be individually plotted.

- Reaction Time (seconds): The time that elapses between the submission of an inference request to the SUT and the SUT responding with the first character of the response (aka time to first token).

- Response Time (seconds): The time that elapses between the submission of an inference request to the SUT and the completion of the SUT's response.

Throughput

Several throughput metrics are computed.

- *Inference Rate* (inferences per second): The number of inferences in the measurement set divided by the measurement interval. This metric is only computed for batch-mode runs, as the interactive λ is the effective inference rate for interactive runs.

- *Throughput* (words per second): The total number of words generated by all inference responses in the measurement set, divided by the measurement interval.

- *Output Rate* (words per second): The number of words in a response divided by the time from the first character to response completion.

- *Output Profile (words per second): The number of words in a partial response divided by the time elapsed from the first character of the response to the end of the partial response, measured multiple times per inference.*

- The Reaction Time, Response Time and Output Rate are measured once per interactive inference, are reported as quantiles and may be individually plotted. The Output Profile is only reported as a quantile over all measurements for all measured inferences. A *word* corresponds to any whitespace-delimited string of 1 or more characters (as with the Linux `wc` command). The Working Group believed that reporting word-based metrics would be more meaningful to a businessperson than reporting in tokens, as is common in other LLM benchmarks.

- The Output Profile is designed to convey an idea of how well the SUT keeps pace during the whole output interval (as opposed to having sluggish periods). Output Profile is most relevant to human readers and other consumers that require individual responses to be delivered at some minimum rate, such as converting LLM responses into speech. An ideal SUT for such use cases might couple a fast Reaction Time with an Output

Profile whose 5th percentile was significantly faster than a reader's (or consuming application's) ability to consume content, meaning that the consumer would not be waiting for the LLM inference results.

Efficiency Metrics

LLM solutions tend to consume expensive resources, so minimizing cost for a given output (that is, maximizing efficiency) is a huge focus for solution architects.

For on-prem systems, efficiency consists of:

- *Energy Efficiency* (words per kWh): Computed from accurate power measurements sampled frequently over the course of the measurement period.
- *Space Efficiency* (words per ft^3·hour): Computed from the rack volume rendered unavailable for other equipment by the SUT.

For SUTs based on a public cloud, efficiency consists of a single metric:

- *Price-Performance* (words per USD): Throughput divided by published retail prices at the time of the test under three pricing scenarios, viz. uncommitted use by the hour, and 24x5 and 24x7 usage/commitment over a year.

Quality Metrics

Fidelity is the main quality metric in STAC-AI. Rather than assessing inference responses against a putative source of ground truth (as in most data-science oriented benchmarks), Fidelity assesses responses against the responses of a *Fidelity Reference*, which is the same model running at the precision defined by its developers on a system selected by STAC.

Fidelity is important because implementation choices in the SUT such as quantization or prompt compression can cause inference responses to differ across SUTs. A reader of a STAC Report will want to know whether a SUT achieved great physical performance by compromising the model's output.

However, a Fidelity standard cannot be absolute. Even though STAC-AI requires the SUT to generate output that is as deterministic as possible (e.g., temperature zero), LLM model-serving frameworks tested by STAC are observed to always entail some non-determinism. So STAC-AI Fidelity ranges from 0% to 100% using an algorithm approved by the Working Group. Fidelity is measured across *all* inferences, including those that are not part of the measurement set.

It is important to keep in mind that the reference for STAC-AI Fidelity is the output of the same model on a different SUT rather than something intended to reflect acceptability of the output to a human. That is, low Fidelity does not necessarily mean low acceptability.

STAC-AI uses the output of the Fidelity Reference in one other way: to calculate *Word Count* Ratio metric. This metric is the total number of words produced by the Fidelity Reference when running the test set (in aggregate, not per inference); hence represents response length similarity between SUT and Fidelity Reference. Due to the non-determinism described earlier, the Word Count Ratio between two sets of responses generated by the same identical systems is typically between 90-110%.

Interpreting and Comparing Results

The STAC-AI benchmark was designed with different real-world deployments and workloads; consequently, comparison of SUTs results may not be straightforward. We therefore present a few useful observations to help read interpret and compare results.

Given the architectural constraints of 2026-era hardware and software for LLM inference, there is a verifiable inverse relationship between Inference Rate (or Lambda) and aggregate Throughput that manifests when LLM output response lengths are varied (all else equal). When comparing results of two similar SUTs, but ones produced shorter response lengths, we expect the shorter response length SUT to have higher Inference Rate (or Lambda) but lower

Throughput benchmark results. Thus, when comparing SUTs results it is important to compare both aggregate Throughput and Inference Rate (or Lambda) alongside Word Count Ratio (a measure of response length) for unbiased comparison.

Similarly, when evaluating SUTs where one employs quantization (e.g., FP4) against the native precision model, we expect the quantized SUT to demonstrate superior figures for **both** Inference Rate and aggregate Throughput. On the other hand, quantization typically results model deviation which can be observed through lower Fidelity metric score. As previously mentioned, model deviation does not necessarily equate to degradation of responses quality. Response length may also vary between quantized and native precision models. Consequently, an equitable performance comparison of SUTs must explicitly account for the precision level used and appropriate quality metrics.