STAC-ML Introduction and Update

Bishop Brock
Head of Research, STAC
bishop.brock@STACresearch.com

Peter Nabicht
President, STAC
peter.nabicht@STACresearch.com

STAC-ML Markets (Training) Benchmark : Underway

- Existing ML training benchmarks are not *specific* to Finance:
  - They typically focus on <u>categorical</u> decisions (e.g., most probable next word)
  - Finance requires good <u>quantitative</u> models (e.g., fair value of a derivative)
- Many use cases have been proposed and discussed, but may not satisfy all high-level requirements:
  - Is this an ongoing concern for many end-users?
  - Can performance and quality be reliably measured and compared?
  - Can we validate that the implementation conforms to the specifications?

STAC® SECURITIES TECHNOLOGY ANALYSIS CENTER

Some ML Training Use Cases Being Considered

Model Type / Use case	Issues / Notes
Predict prices/returns/portfolio-weights from market data	• Obviously interesting use cases • Training / re-training very important ! Low signal-noise means models learn quickly and erratically – difficult to benchmark
Complex multi-dimensional functions (Derivative valuation, Model Calibration PDE solving)	• Also sees much current interest ! Not clear if training is the bottleneck for most use cases (train once and done?)
Synthetic market data generation	• Useful research and risk testing tool ! Quality evaluation may be difficult ! Again, not clear training is bottleneck
Reinforcement learning for (hedging, trading, …)	• Under investigation

STAC
SECURITIES TECHNOLOGY ANALYSIS CENTER

Training: Searching for the right workload

- STAC Benchmarks are defined by financial firms to reflect their needs
- What training workloads give you the insight you need?
- **Join us!**

www.STACresearch.com/ML

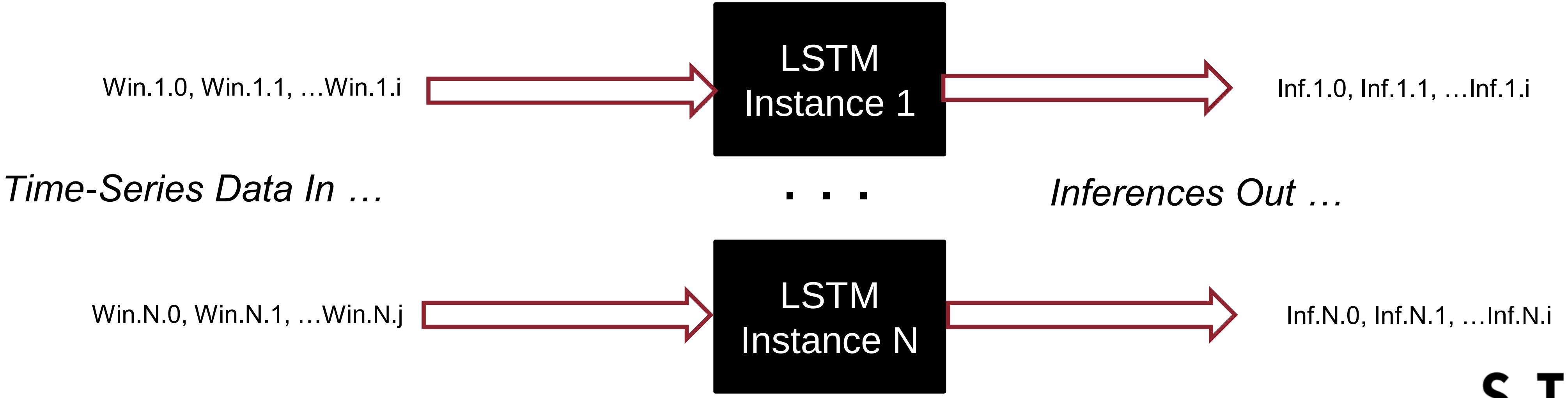
STAC-ML Markets (Inference) : Basics

- LSTM models inferring on simulated market data features
- Goal: isolate inference performance
  - Inference engine software
  - Underlying processors, memory, accelerators, etc.
  - Anything required to optimally use the former with the latter (e.g., data transfer to processor memory)
- Metrics:
  - Latency, throughput, error, power efficiency, space efficiency, cost
- Benchmarks allow any level of precision (including mixed-precision)

STAC® SECURITIES TECHNOLOGY ANALYSIS CENTER

Scale Dimensions; Benchmark Schematic

- Model size
  - Three are currently specified
  - Input data window scales with model size
- Number of Model Instances running in parallel
  - As specified by the SUT provider
  - Performance / efficiency per model instance is key for co-located inference

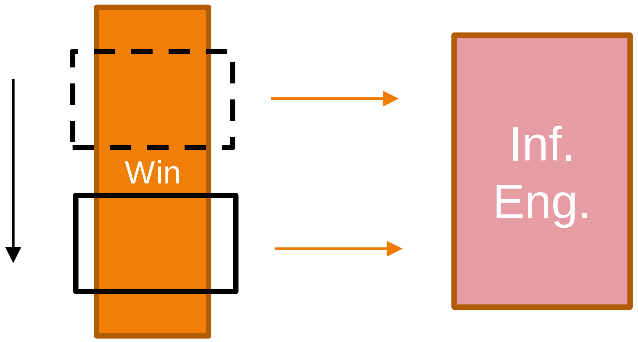


Use Cases and Optimizations; Suites

- Different Use Cases:
  - Trading – Latency Optimization
  - Backtesting – Throughput Optimization
- Optimization tradeoffs (latency vs throughput vs efficiency vs error) are up to the SUT provider
  - The tests collect all metrics every time, no matter the optimization goal
  - Any quantization scheme allowed, if used consistently

Sumaco – Fixed, Unique Window

Tacana - Sliding Window (Streaming)



STAC® SECURITIES TECHNOLOGY ANALYSIS CENTER

Uses - STAC-ML Markets (Inference)

- Three users of STAC-ML
  - STAC
  - Vendors
  - Financial firms
- I will talk about all three

Research Available to ML STAC Track Subscribers

- GCP Cloud SUT
  - Latency- and Throughput-optimized configurations for ONNX inference
- TensorFlow Performance (on CPU)
  - Looked at different ways to configure TensorFlow for inference
- Azure Cloud-SUT Comparison
  - Looked at latency and throughput on 3 different CPU architectures
  - Report includes a detailed business use-case analysis
- For access:

council@STACresearch.com

TensorFlow Optimization Note

- TensorFlow is becoming widely used as a general-purpose computing environment
- We explored optimization of LSTM model single-inference in TensorFlow
  - This report may be a good place to begin your own optimization research
- Some of what we found:
  - ONNX was always faster than TensorFlow for LSTM single-inference on our CPU-based test system
  - XLA compilation often - but not always - yields the most performant TensorFlow models
    - … and we explain why

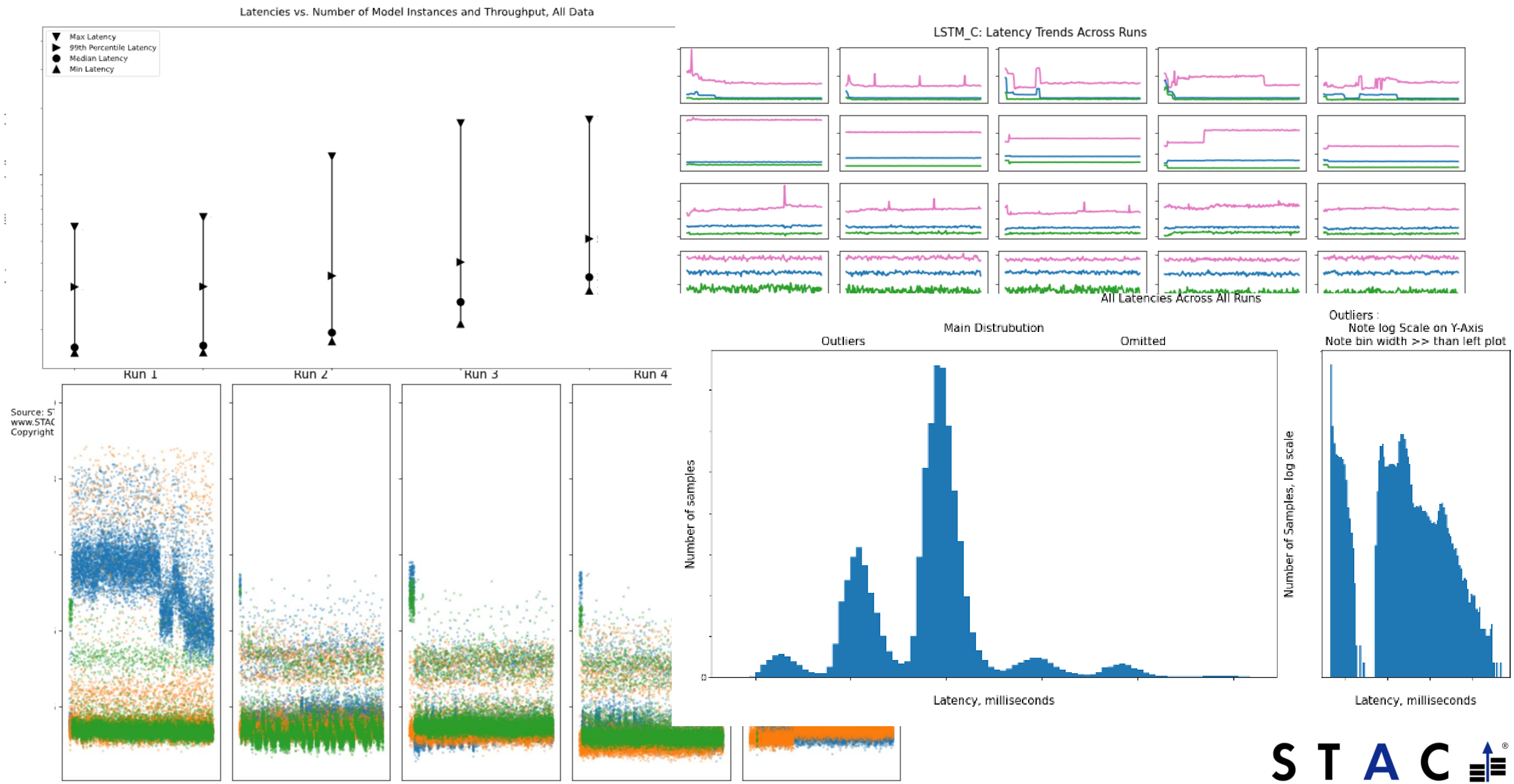
STAC-ML Markets (Inference) Azure Cloud-SUT Comparison

- Goal: compare 3 CPU architectures for inference
  - Intel, AMD, Ampere (ARM)
- Used the STAC "Naive" Python implementation with ONNX
- Tested on Microsoft Azure

Thanks to Microsoft for supporting the STAC community by providing credits for this research!

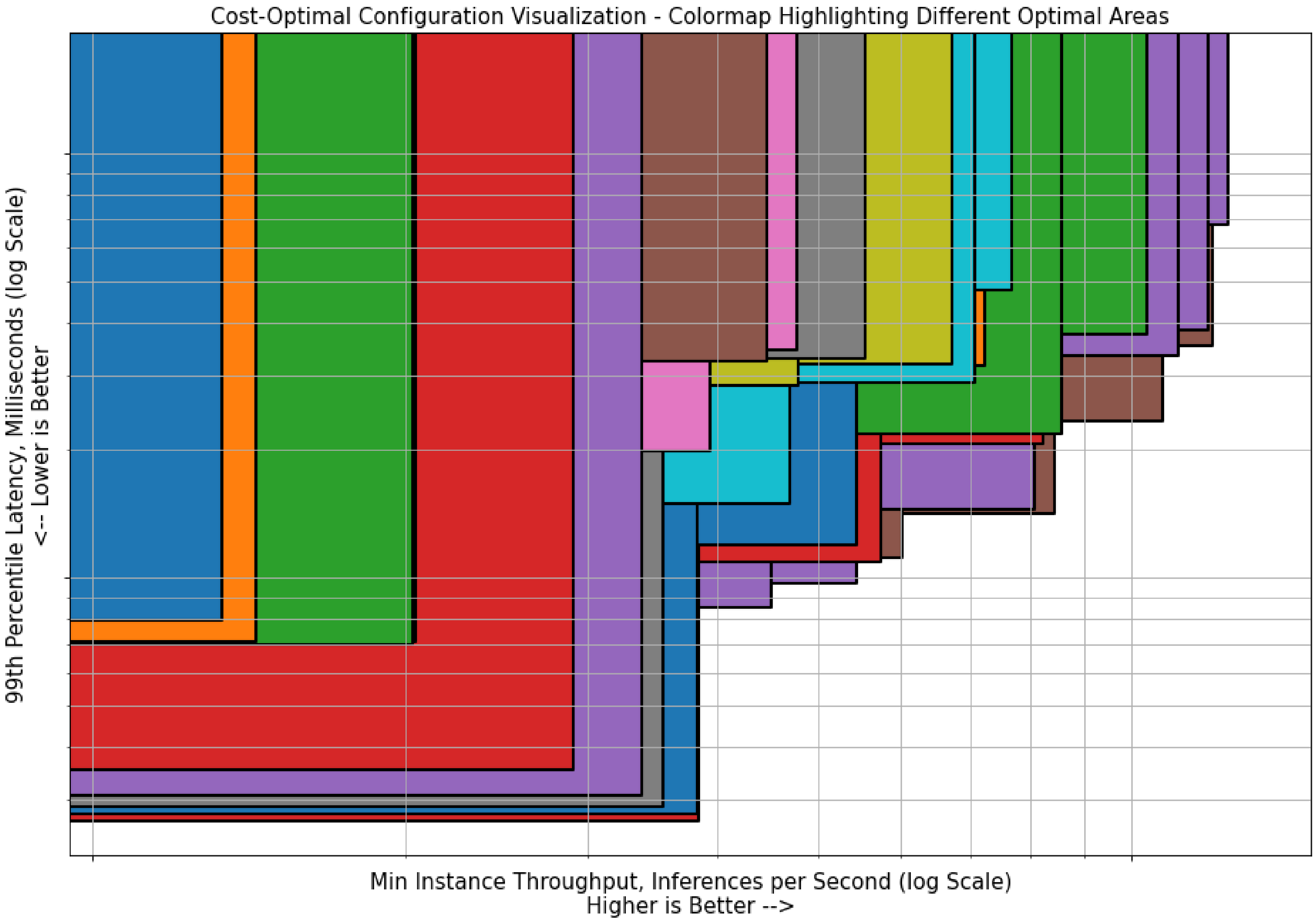
- Tested two configs for each VM (latency opt., throughput opt.)
- All 6 reports are in the STAC Vault with a comparison report
- No vendors participated in the setup and optimization of the SUTs

Audit Reporting: Detailed analysis available for each SUT



SECURITIES TECHNOLOGY ANALYSIS CENTER

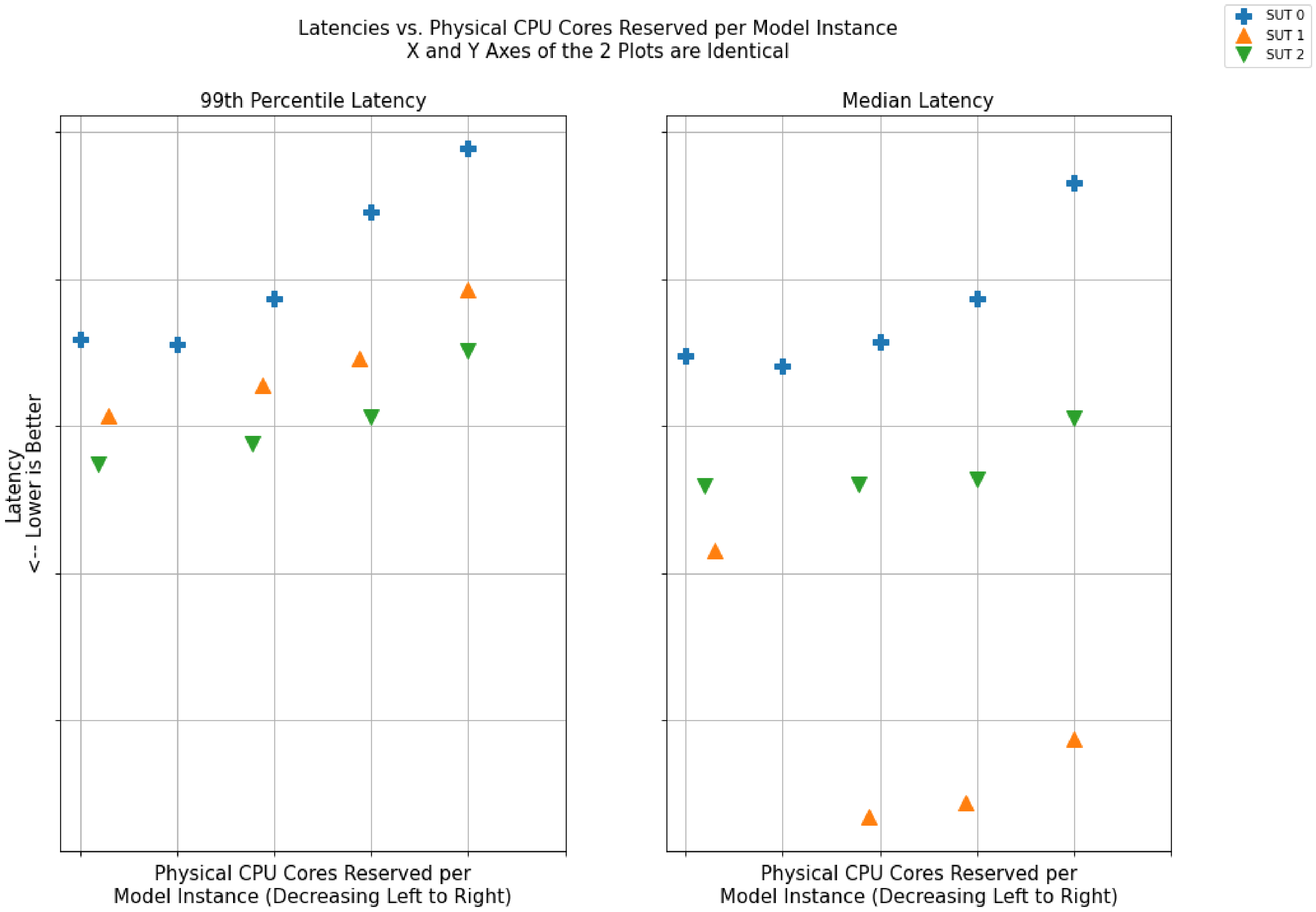
Comparison Report: Business Use-Case Analysis



Each colored area represents the most cost-efficient way to achieve any latency/throughput contained in the area.

All processors (AMD, Ampere, Intel) are represented multiple times here.

Comparison Report: Performance Analysis



In the report we use benchmark visualizations to explain why SUT 2 demonstrates lower 99th percentile latencies but higher median latencies than SUT1

STAC® SECURITIES TECHNOLOGY ANALYSIS CENTER

Groq was first public tested SUT!

- STAC-ML Pack for GroqWare™ (Rev A)
  - Version of STAC “Naive” implementation adapted for GroqWare™ APIs
  - Effectively FP16
- GroqWare™ SDK 0.9.0.5 devtools and runtime
- Python 3.8.15; NumPy 1.23.4
- Ubuntu Linux 22.04.1 LTS
- GroqNode™ GN1-B8C-ES:
  - 8 x GroqCard™ 1 Accelerators (GC1-010B)
  - 2 x AMD EPYC™ 7413 24-core CPUs @ 2650 MHz
  - 16 slots x 64GiB DDR4 - 1024GiB Total

www.STACresearch.com/GROQ221014

STAC SECURITIES TECHNOLOGY ANALYSIS CENTER

Result highlights – Groq

- For small model LSTM_A, across 1, 2 and 4 simultaneously running model instances (NMI):
  - Worst case 99th percentile latency was 56.4 µsec[1]
  - 99th percentile latencies varied 1% (55.9 to 56.4 µsec) [2]
  - The widest spread from minimum to 99th percentile latency was 6% (53.4 to 56.4 µsec) [3]

www.STACresearch.com/GROQ221014

1. STAC-ML.Markets.Inf.S.LSTM_A.4.LAT.v1
2. STAC-ML.Markets.Inf.S.LSTM_A.[1,2,4].LAT.v1
3. STAC-ML.Markets.Inf.S.LSTM_A.4.LAT.v1

Result highlights – Groq

- For small model LSTM_A, across 1, 2 and 4 simultaneously running model instances (NMI):
  - Worst case 99th percentile latency was 56.4 μsec[1]
  - 99th percentile latencies varied 1% (55.9 to 56.4 μsec) [2]
  - The widest spread from minimum to 99th percentile latency was 6% (53.4 to 56.4 μsec) [3]

www.STACresearch.com/GROQ221014

1. STAC-ML.Markets.Inf.S.LSTM_A.4.LAT.v1
2. STAC-ML.Markets.Inf.S.LSTM_A.[1,2,4].LAT.v1
3. STAC-ML.Markets.Inf.S.LSTM_A.4.LAT.v1

Results highlights – Groq

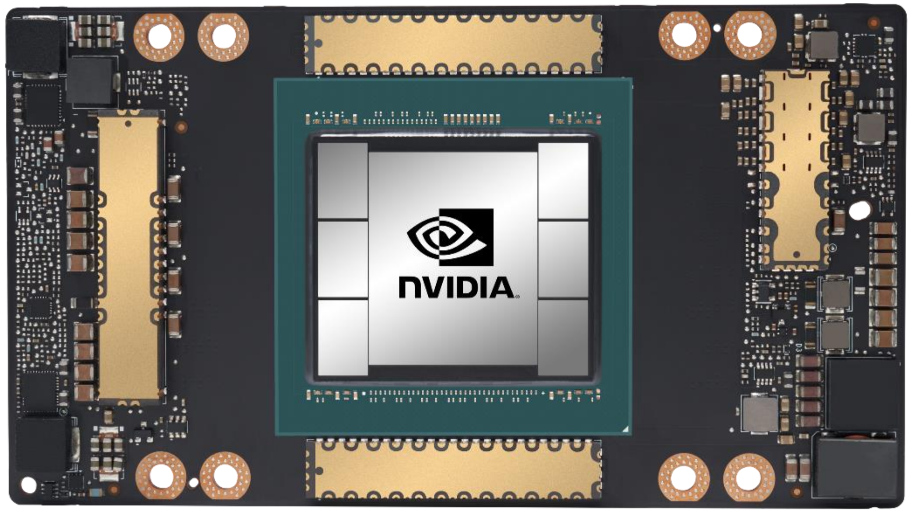
- For large model LSTM_C, across all NMI tested:
  - Worst case 99th percentile latency was 2.27 ms[1]
  - 99th percentile latencies varied by 2% (2.72 to 2.77 ms) [2]
  - The widest spread from minimum to 99th percentile latency was 3% (2.68 to 2.77 ms) [3]

www.STACresearch.com/GROQ221014

1. STAC-ML.Markets.Inf.S.LSTM_C.8.LAT.v1
2. STAC-ML.Markets.Inf.S.LSTM_C.[1,2,4,8].LAT.v1
3. STAC-ML.Markets.Inf.S.LSTM_C.8.LAT.v1

NVIDIA – 3 SUTs with same GPU-based stack

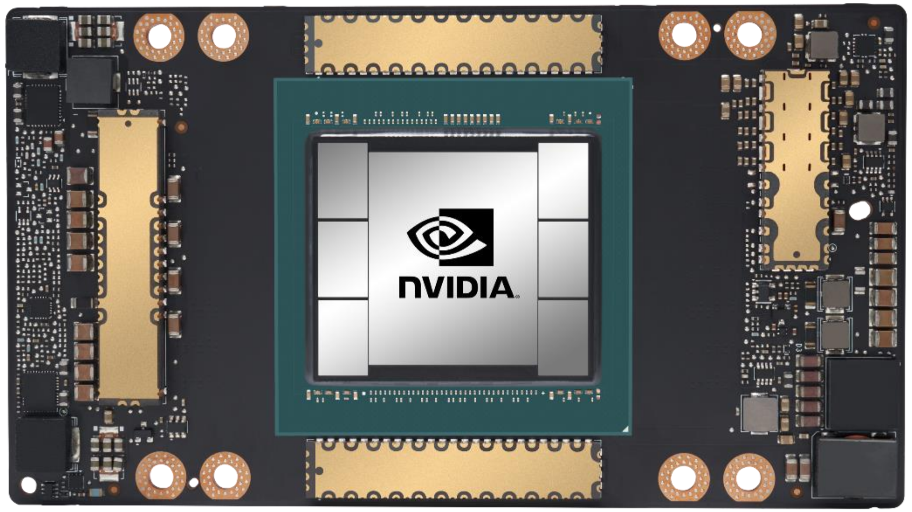
- STAC-ML Pack for CUDA and cuDDN (Rev A)
- NVIDIA CUDA Toolkit 11.7
- NVIDIA CUDA Deep Neural Network library (cuDNN) 8.4.1.50
- Ubuntu 20.04.5 LTS
- SuperMicro Ultra SuperServer SYS-620U-TNR
  - NVIDIA A100 80GB PCIe Tensor Core GPU
  - 2 x Intel Xeon Gold 6354 CPU @ 3.00GHz
  - 512GiB of memory
- Published results on two SUTs
  - Throughput optimized, Sumaco suite, FP16
  - Latency optimized, Tacana suite, FP32
- Vault Report for 3rd SUT
  - Throughput optimized, Tacana suite, FP16



www.STACresearch.com/NVDA221118a
www.STACresearch.com/NVDA221118b
www.STACresearch.com/NVDA221118c

Throughput optimized, Sumaco suite, FP16

- Same stack configured to
  - Operate on a fixed window of unique updates (Sumaco)
  - Maximize throughput
  - Use FP16
- For LSTM_A, across all NMI tested:
  - Total throughput ranged from 1.63 to 1.71 M inf/sec[1]
  - Energy efficiency ranged from 1.72 to 1.8 M inf/sec/kW[2]



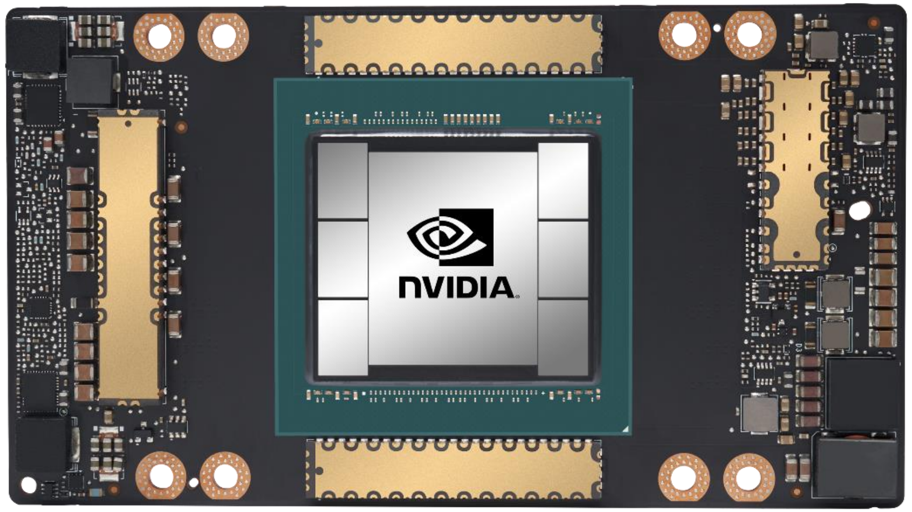
www.STACresearch.com/NVDA221118a

1. STAC-ML.Markets.Inf.S.LSTM_A.[1,2,4].TPUT.v1
2. STAC-ML.Markets.Inf.S.LSTM_A.2.ENERG_EFF.v1

STAC SECURITIES TECHNOLOGY ANALYSIS CENTER

Throughput optimized, Sumaco suite, FP16

- For LSTM_B, across all NMI tested:
  - Total throughput was 191 K inf/sec[1]
  - Energy efficiency was 206 K inf/sec/kW[2]
- For LSTM_C, across all NMI tested:
  - Total throughput was 12.8 K inf/sec[3]
  - Energy efficiency was 17.7 K inf/sec/kW[4]



www.STACresearch.com/NVDA221118a

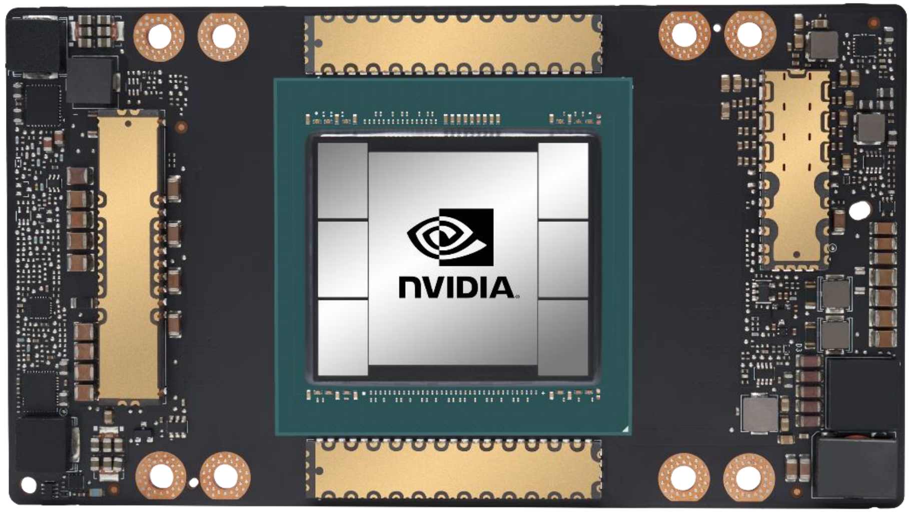
1. STAC-ML.Markets.Inf.S.LSTM_B.[1,2,4].TPUT.v1
2. STAC-ML.Markets.Inf.S.LSTM_B.[1,2,4]. ENERG_EFF.v1
3. STAC-ML.Markets.Inf.S.LSTM_C.[1,2,4].TPUT.v1
4. STAC-ML.Markets.Inf.S.LSTM_C.[1,2,4]. ENERG_EFF.v1

Latency optimized, Tacana suite, FP32

- Same stack configured to
  - Operate on a sliding window of updates (Tacana)
  - Minimize latency
  - Use FP32

- For LSTM_A the 99p latency :
  - With 1 NMI was 35.2 µsec[1]
  - With 32 NMI was 58.8 µsec[2]

- For LSTM_B the 99p latency:
  - With 1 NMI was 68.5 µsec[3]
  - With 32 NMI was 149 µsec[4]



www.STACresearch.com/NVDA221118b

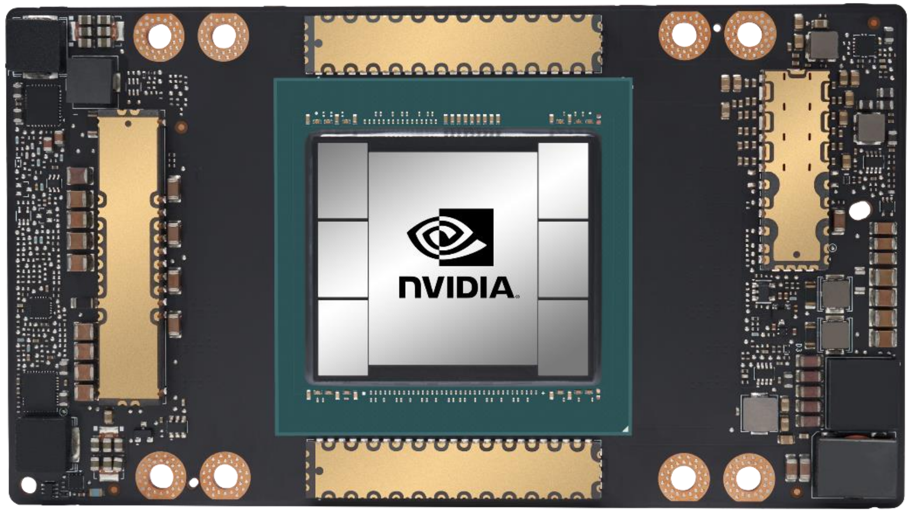
1. STAC-ML.Markets.Inf.T.LSTM_A.1.LAT.v1
2. STAC-ML.Markets.Inf.T.LSTM_A.32.LAT.v1
3. STAC-ML.Markets.Inf.T.LSTM_B.1.LAT.v1
4. STAC-ML.Markets.Inf.T.LSTM_B.32.LAT.v1

STAC®
SECURITIES TECHNOLOGY ANALYSIS CENTER

Latency optimized, Tacana suite, FP32

- For LSTM_C the 99p latency:
  - With 1 NMI was 640 µsec[1]
  - With 16 NMI was 748 µsec[2]

- Across all tested LSTM models and NMI, the largest outlier was 2.3x the median latency
  - Median latency 35 µsec, max latency 81 µsec[3]



www.STACresearch.com/NVDA221118b

1. STAC-ML.Markets.Inf.T.LSTM_C.1.LAT.v1
2. STAC-ML.Markets.Inf.T.LSTM_C.16.LAT.v1
3. STAC-ML.Markets.Inf.T.LSTM_A.2.LAT.v1

STAC® SECURITIES TECHNOLOGY ANALYSIS CENTER

Myrtle.ai tested with FPGA as accelerator

- STAC-ML Pack for Myrtle.ai VOLLO™ (Rev A)
  - bfloat16 precision
- VOLLO SDK 0.1.0
- VOLLO Accelerator 0.1.0
- Ubuntu Linux 22.04.1 LTS
- BittWare TeraBox™ 1402B (1U)
  - 4 x BittWare IA-840f-0001 each with
    - Intel® Agilex™ AGF027 FPGA
    - 4 x 16 GiB DDR4 @ 2666 MHz
  - 1 x Intel® Xeon® Platinum 8351N CPU @ 2.40 GHz
  - 4 x 8 GiB Micron DDR4 @ 2933 MHz (32GiB total)

Myrtle.ai

www.STACresearch.com/MRTL221125

STAC®
SECURITIES TECHNOLOGY ANALYSIS CENTER

Results highlights – Myrtle.ai

- 99p latencies across 1, 2, 3 & 4 NMI for:
  - LSTM_A were 24.0 – 24.1 μsec[1]
  - LSTM_B were 64.8 μsec[2]
  - LSTM_C were 1.35 ms[3]

- For LSTM_A with 48 NMI:
  - Total throughput was 651 K inf/sec[4]
  - Space eff. was 647 K inf/sec/cubic foot[5]
  - Energy eff. was 1.2 M inf / sec/ kW[6]
  - The 99p latency was 73.9 μsec, which was 3.1x the 99th percentile latency of 1 NMI[7]

Myrtle.ai

www.STACresearch.com/MRTL221125

1. STAC-ML.Markets.Inf.S.LSTM_A.[1,2,3,4].LAT.v1
2. STAC-ML.Markets.Inf.S.LSTM_B.[1,2,3,4].LAT.v1
3. STAC-ML.Markets.Inf.S.LSTM_C.[1,2,3,4].LAT.v1
4. STAC-ML.Markets.Inf.S.LSTM_A.48.TPUT.v1
5. STAC-ML.Markets.Inf.S.LSTM_A.48. SPACE_EFF.v1
6. STAC-ML.Markets.Inf.S.LSTM_A.48. ENERG_EFF.v1
7. STAC-ML.Markets.Inf.S.LSTM_A.[1, 48].LAT.v1

Results highlights – Myrtle.ai

- For LSTM_B with 16 NMI:
  - The 99p latency was 147 µsec, which was 2.3x the 99p latency of 1 NMI[1]
- Across all Models and NMI:
  - The widest percentage spread from median to 99p latencies was 7% (26.5 µsec to 28.4 µsec) [2]

Myrtle.ai

www.STACresearch.com/MRTL221125

1. STAC-ML.Markets.Inf.S.LSTM_B.[1, 16].LAT.v1
2. STAC-ML.Markets.Inf.S.LSTM_A.12.LAT.v1

STAC SECURITIES TECHNOLOGY ANALYSIS CENTER

STAC-ML tools are ready for you, too

Machine Learning

Inference (STAC-ML)

STAC-ML Markets (Inference) Test Harness

STAC-ML Markets (Inference) Reference Implementation (ONNX & TensorFlow)

STAC-ML Pack for CUDA and cuDNN

STAC-ML Pack for Myrtle.ai VOLLO

STAC-ML Pack for GroqWare

- Vendor implementations – See how it works
- Test harness software and analysis tools – Test your own stacks
  - In fact, test your own models!

www.STACresearch.com/ML

STAC® SECURITIES TECHNOLOGY ANALYSIS CENTER