STAC

BENCHMARK COUNCIL

# Introducing STAC-ML

Bishop Brock
Head of Research, STAC

bishop.brock@STACresearch.com

# History

- Driven by financial firms
  - Motivation: market making, hedging, customer pricing, etc.
- STAC-ML working group has refined the original POC idea into a finished benchmark specification
- Tech vendors provided crucial input
- But control ultimately rests with users – i.e., those who must deliver business value from technology in the real world
  - Like all STAC Benchmarks

# Latest Status

- Off to the races!
- The benchmark specifications, test harness, reference implementation, and documentation are released
- 5 vendor implementations are currently underway
- Test Harness engineered to allow end-users to mark their own stacks to market

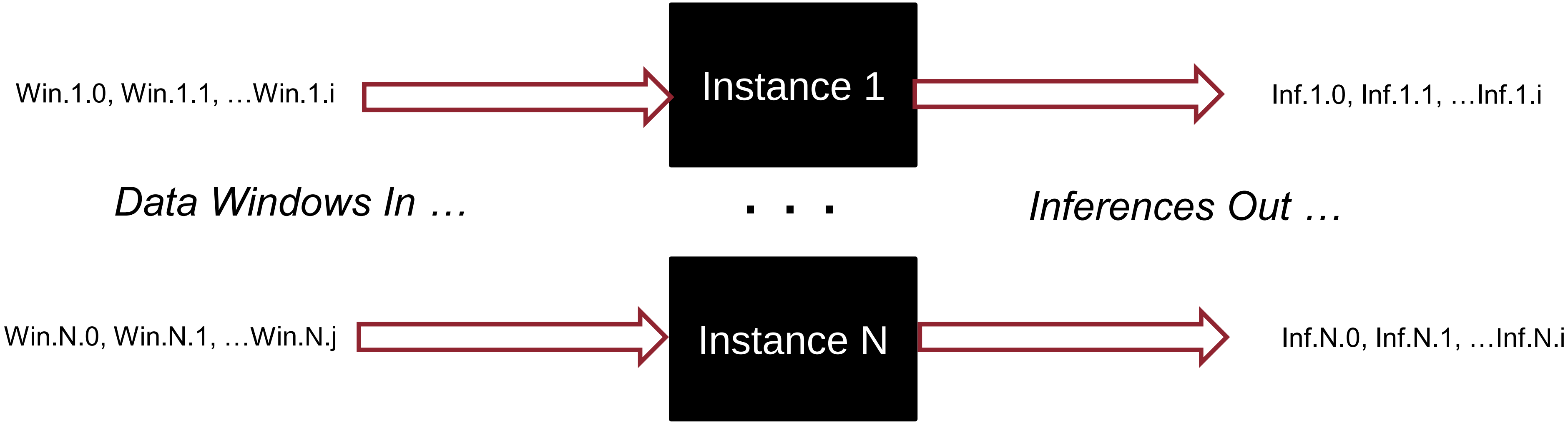
**http://www.STACresearch.com/ml**

# Basics

- LSTM models that simulate real models derived from market data
- 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, power efficiency, space efficiency, error
- Benchmarks allow any level of precision (including mixed-precision)

# Scale Dimensions

- 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


Win.1.0, Win.1.1, …Win.1.i
Instance 1
Inf.1.0, Inf.1.1, …Inf.1.i
*Data Windows In …*
. . .
*Inferences Out …*
Win.N.0, Win.N.1, …Win.N.j
Instance N
Inf.N.0, Inf.N.1, …Inf.N.i

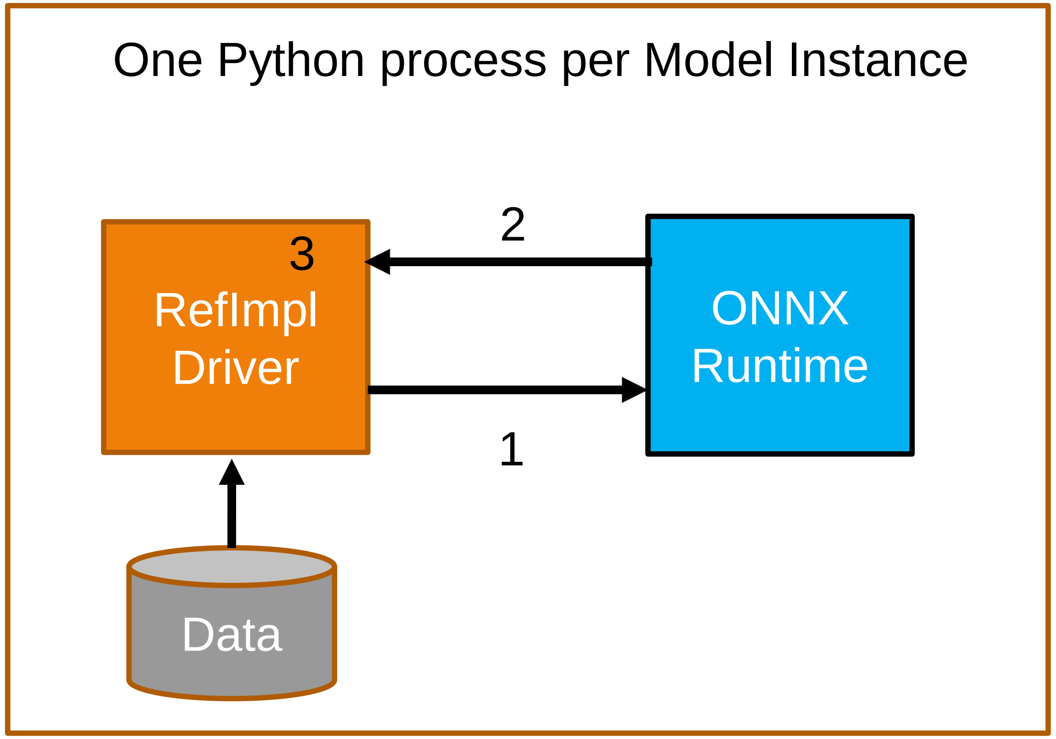
# Use Cases and Optimizations

- 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

# Speaking of Tradeoffs…

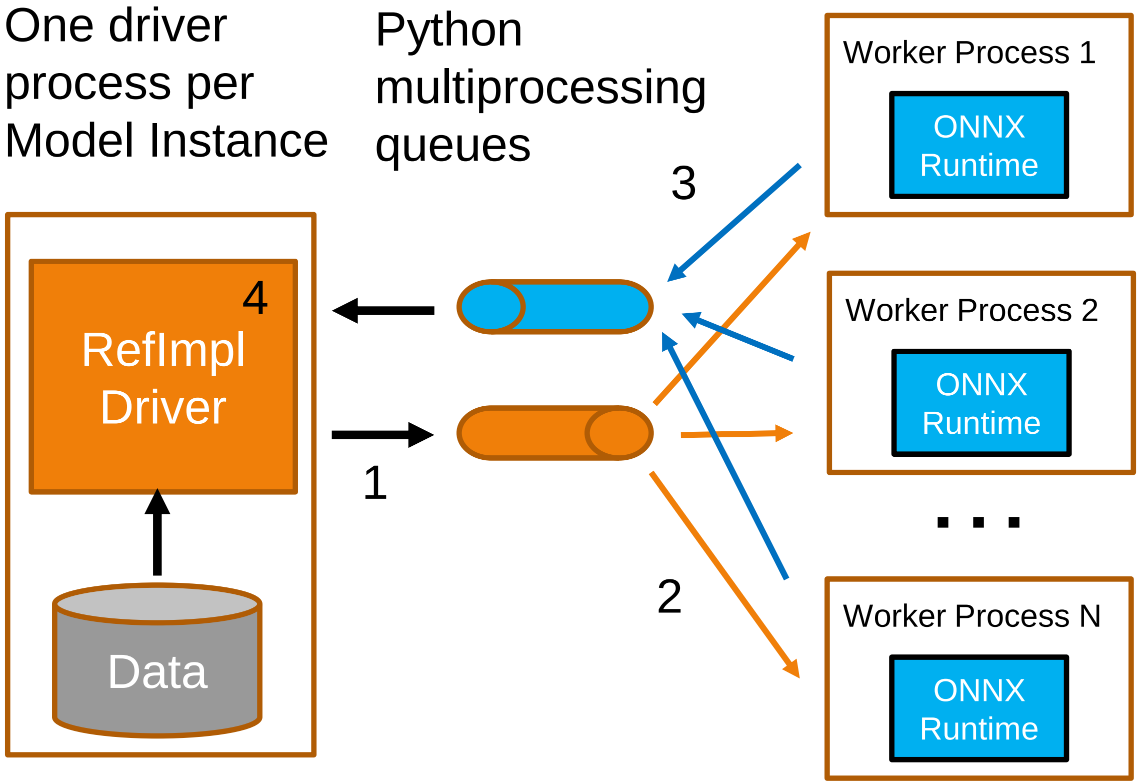
- STAC has just published the first audit reports, representing internal research
- 2 SUTs:
  - Latency Optimized
  - Throughput Optimized
- Same Software (different tuning):
  - Pure Python Implementation
  - STAC-ML Markets (Inference) Naive Implementation
  - Unmodified ONNX 1.11.0 runtime
- Same Hardware:
  - 60-vCPU @ 3.1Ghz, Sole-Tenant cloud instance, 240GiB Memory
- Models: Standard benchmark models at default FP32 precision

# Serial Implementation. | Parallel Implementation



1. Driver calls ONNX runtime with data window; waits for response
2. ONNX returns inference value
3. Driver stores value in memory

Repeat…



1. Driver inserts data window(s) into queue; waits for response
2. Workers take windows from queue
3. Workers store results in queue
4. Driver gets result from queue and orders it in memory

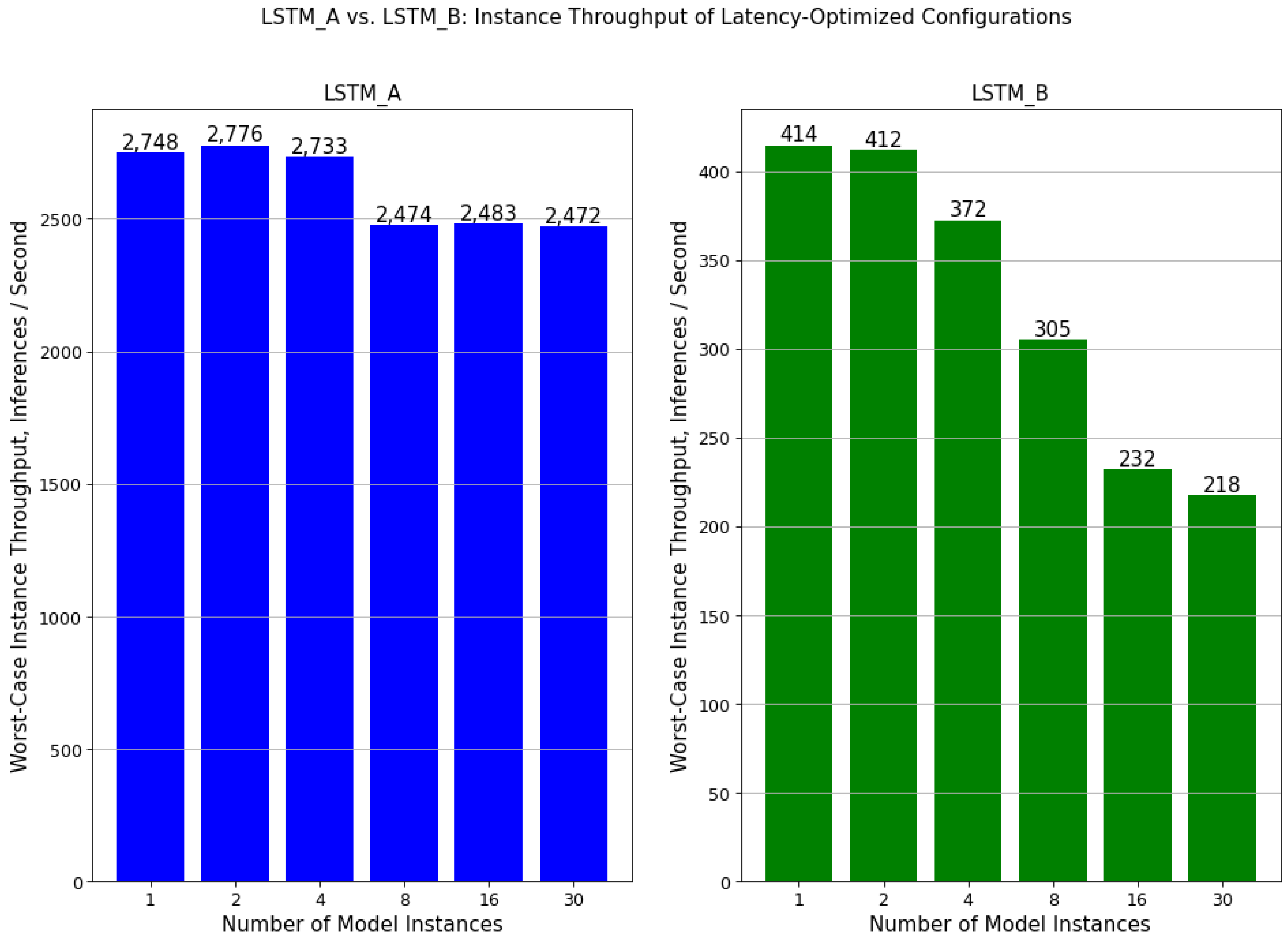
Repeat…

Multiple worker processes per Model Instance

# Optimization ‘Knobs’

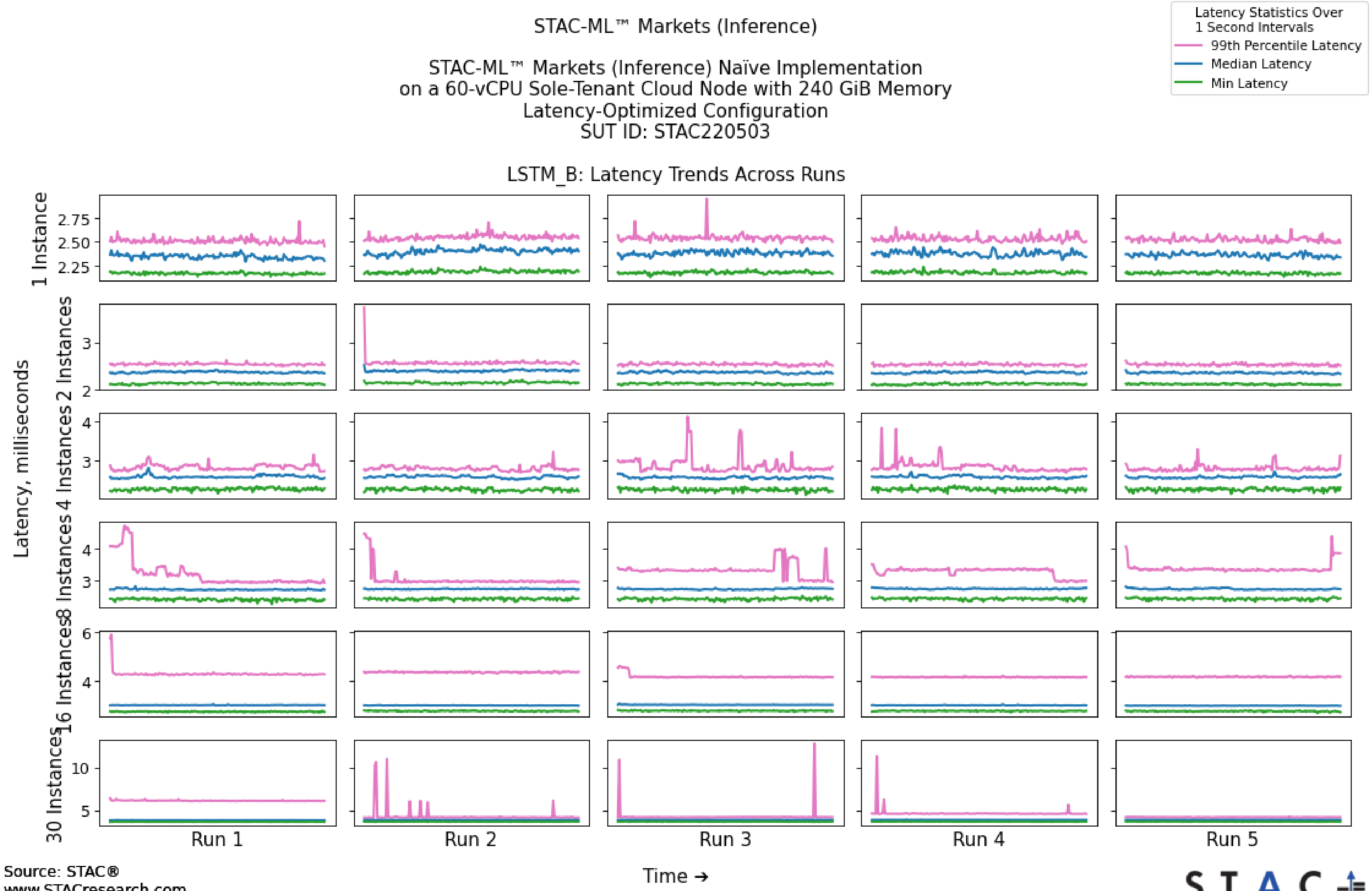
- Number of Model Instances
  - We looked at 1, 2, 4, 8, 16 and 30 instances
- Number of ONNX threads
  - ONNX can (sometimes) effectively utilize multiple threads per model
- Number of Parallel Instances
  - Note: 1 Parallel Instance == Serial Model
  - We have a choice of allocating a HW thread to either an ONNX thread or a Python process
  - The optimal choice again varies by model and optimization goal
- *A Research Note describing the optimization experiment is available in the STAC Vault*.

# Worst-Case Instance Throughput Comparison, LSTM_A vs LSTM_B



STAC®
SECURITIES TECHNOLOGY ANALYSIS CENTER

# Latency Trends



STAC-ML™ Markets (Inference)

STAC-ML™ Markets (Inference) Naïve Implementation
on a 60-vCPU Sole-Tenant Cloud Node with 240 GiB Memory
Latency-Optimized Configuration
SUT ID: STAC220503

LSTM_B: Latency Trends Across Runs

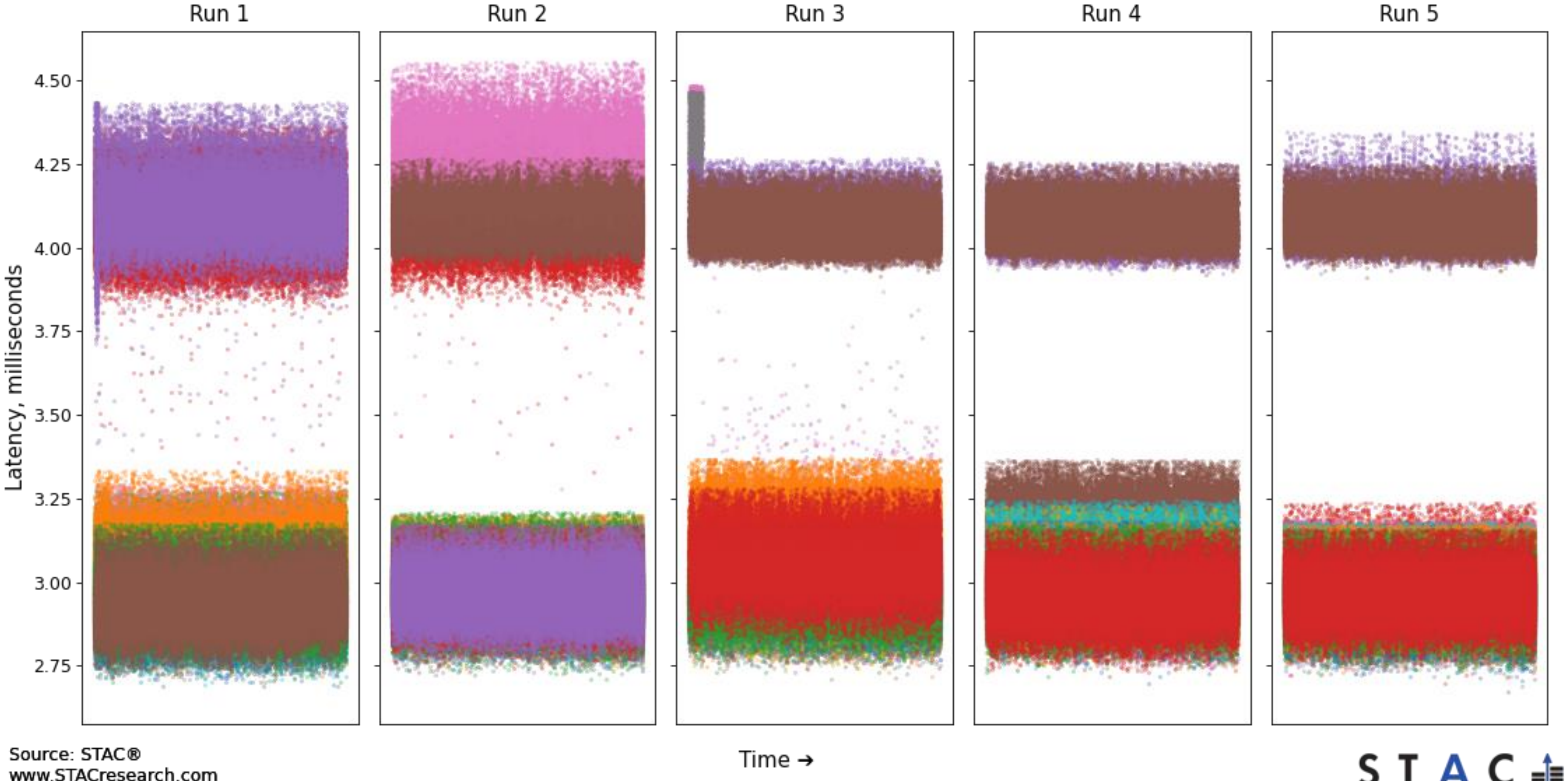
Source: STAC®
www.STACresearch.com
Copyright © 2022 STAC

# Latency Details

STAC-ML™ Markets (Inference)

STAC-ML™ Markets (Inference) Naïve Implementation
on a 60-vCPU Sole-Tenant Cloud Node with 240 GiB Memory
Latency-Optimized Configuration
SUT ID: STAC220503

LSTM_B, 16 Model Instances
Detailed Latency over Time, Samples Above the 99.0th% Quantiles per Model Instance/Run Omitted



Source: STAC®
www.STACresearch.com
Copyright © 2022 STAC

# How to get Access

- All STAC subscribers can access
  - Audited and publicly published STAC Reports
- Premium Subscribers can access
  - Benchmark Specifications
  - Highly detailed configuration information
  - Extensive, detailed visualizations and tables on Performance, Efficiency, and Error
  - Code for test harness, generating post-test visualizations, and STAC Packs
  - Additional reports and research in the confidential STAC Vault™

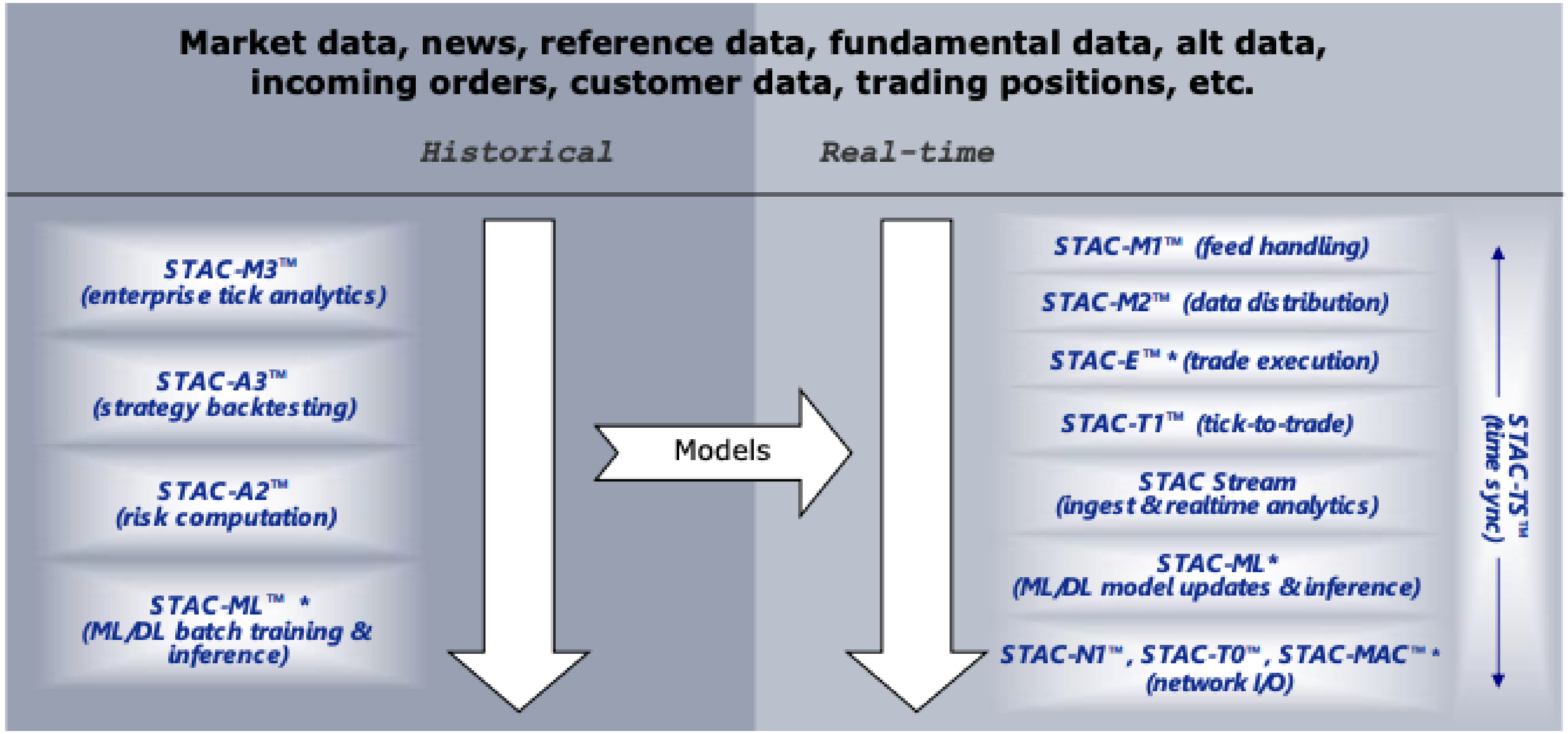
**http://www.STACresearch.com/ml**

# STAC Tracks

## Tradeflow STAC Track    Analytics STAC Track

Included in both previously existing STAC Tracks

No need to do anything different



Market data, news, reference data, fundamental data, alt data, incoming orders, customer data, trading positions, etc.

Historical

Real-time

STAC-M3™ (enterprise tick analytics)

STAC-A3™ (strategy backtesting)

STAC-A2™ (risk computation)

STAC-ML™ * (ML/DL batch training & inference)

Models

STAC-M1™ (feed handling)

STAC-M2™ (data distribution)

STAC-E™ * (trade execution)

STAC-T1™ (tick-to-trade)

STAC Stream (ingest & realtime analytics)

STAC-ML* (ML/DL model updates & inference)

STAC-N1™, STAC-T0™, STAC-MAC™ * (network I/O)

STAC-TS™ (time sync)

STAC® SECURITIES TECHNOLOGY ANALYSIS CENTER

# Machine Learning STAC Track

- New STAC Track that includes
  - STAC-ML Markets (Inference)
  - Future STAC-ML benchmarks
- Free trial for the remainder of 2022
  - For those responsible for ML research and infrastructure
  - Full access, including STAC Vault content
  - To request the trial:

**council@STACresearch.com**