Benchmarking realtime LSTM inference on time series

Michel Debiche
Director of Analytics Research, STAC

michel.debiche@STACresearch.com

Business background

- ML models are being applied to incoming market data in real time. E.g.:
  - Trading
  - Algorithm parameter tuning
- A significant question about a given technique and technology stack:
  - How fast are inferences?
  - What kind of throughput is possible with acceptable latency and accuracy?

STAC®
SECURITIES TECHNOLOGY ANALYSIS CENTER

Benchmark development

- Some user firms on the STAC Benchmark Council have asked for/provided input to proposed inference benchmarks
- STAC has now proposed a first version of specs
- Demonstrated those specs via a POC (described in this talk)
- Now calling for interest to finalize the specs and specify enhancements

www.STACresearch.com/AI

STAC®
SECURITIES TECHNOLOGY ANALYSIS CENTER

Benchmark objectives

- Many things affect inference latency using a given model:
  - Event handling
  - Data prep
  - Feature engineering
  - Getting features to the inference engine
  - Inference engine performance
  - Accuracy tolerance (choices regarding numerical precision)

- Initial benchmark specs seek to isolate inference engine performance

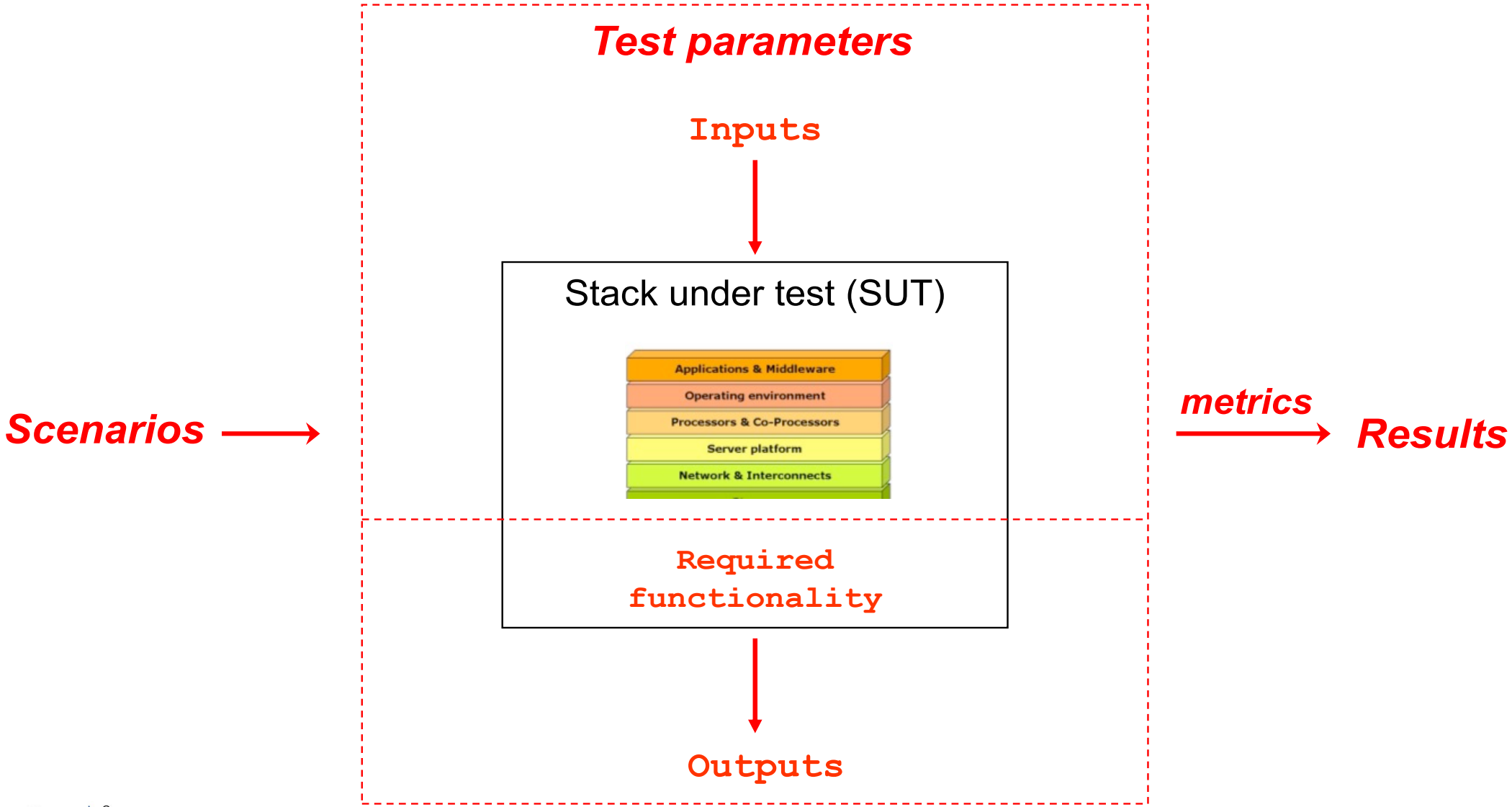
STAC®
SECURITIES TECHNOLOGY ANALYSIS CENTER

Benchmark objectives

- Isolating performance of the inferencing stack means involving only:
  - Inference engine
  - Underlying processors, memory, etc.
  - Anything required to optimally use the former with the latter (e.g., data transfer to processor memory)
- Measure the upper bound performance of the SUT
  - Eliminate queuing of inbound data from the latency measurement. That means this should be a pull-based workload, not push.
  - Maximize the resources available for inferencing. That is, the SUT should not have to handle:
    - Network IO
    - Market data processing
    - Feature engineering

STAC®
SECURITIES TECHNOLOGY ANALYSIS CENTER

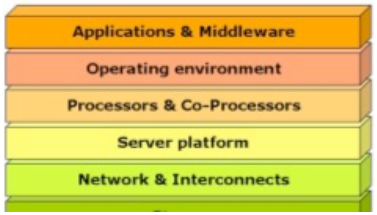
Benchmark construction



Benchmark construction

1. Model in ONNX format
2. Features pre-loaded into main memory

Stack under test (SUT)



Required functionality

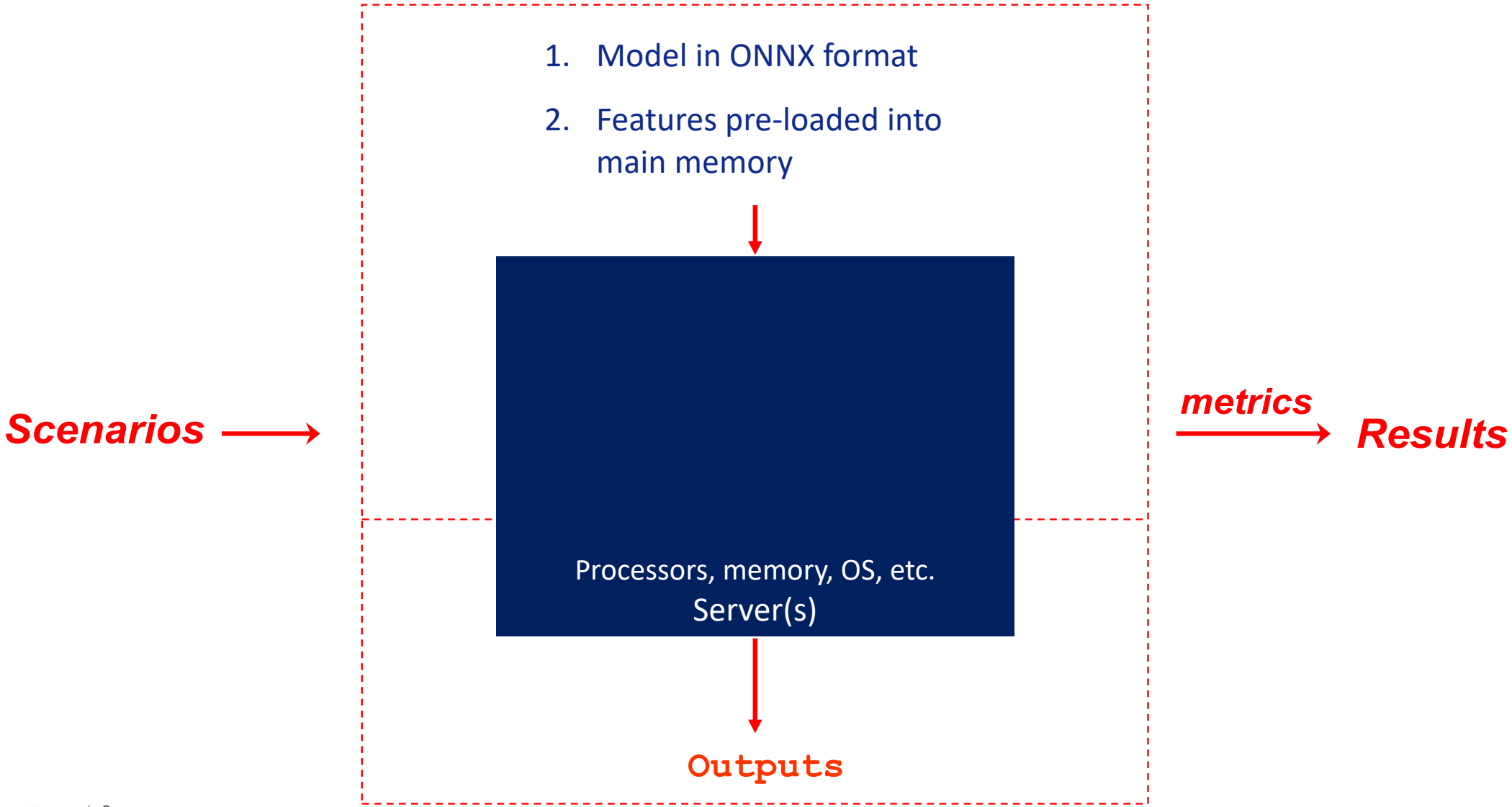
Outputs

Scenarios →

metrics → ***Results***

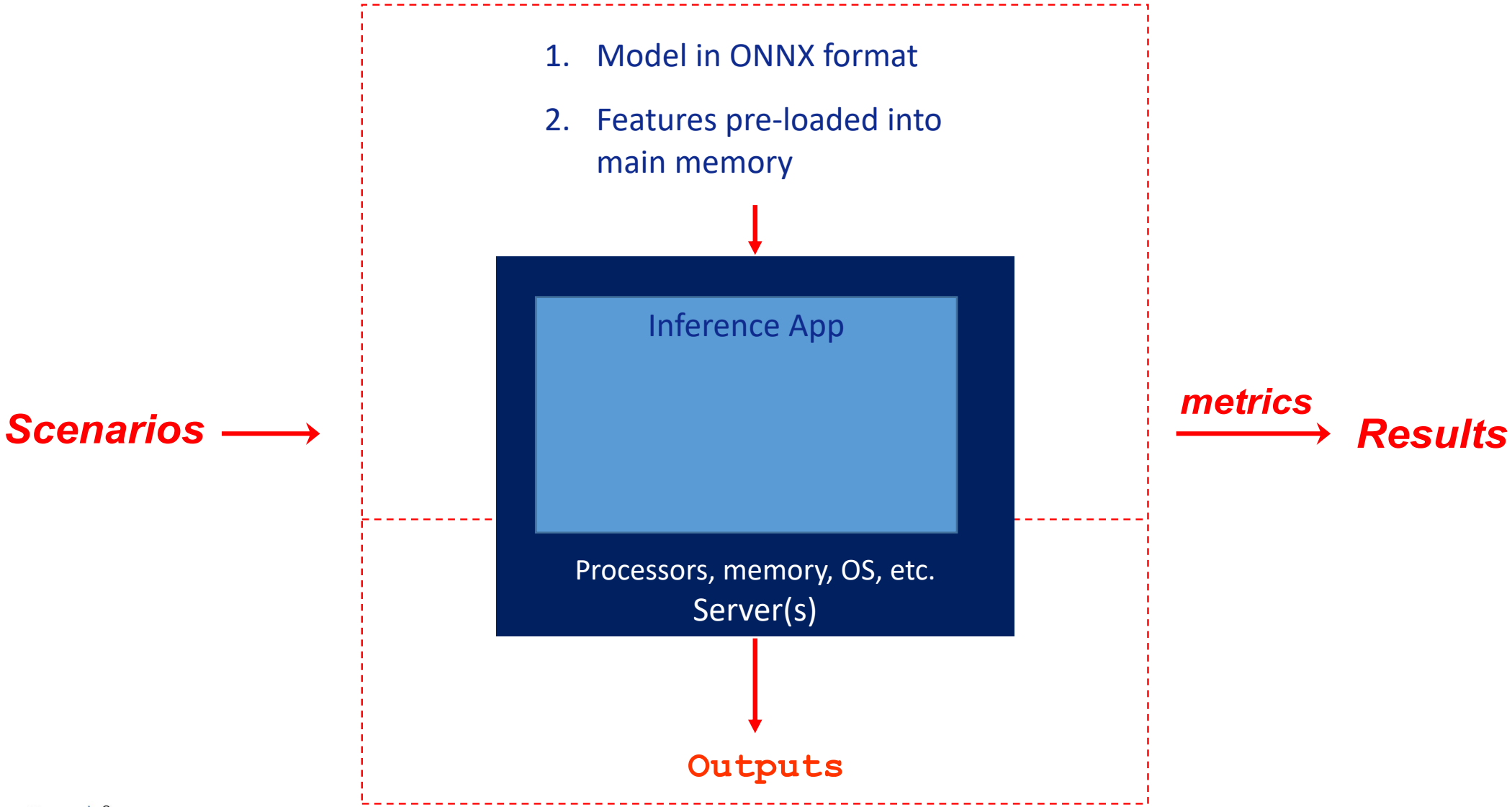
STAC® SECURITIES TECHNOLOGY ANALYSIS CENTER

Benchmark construction

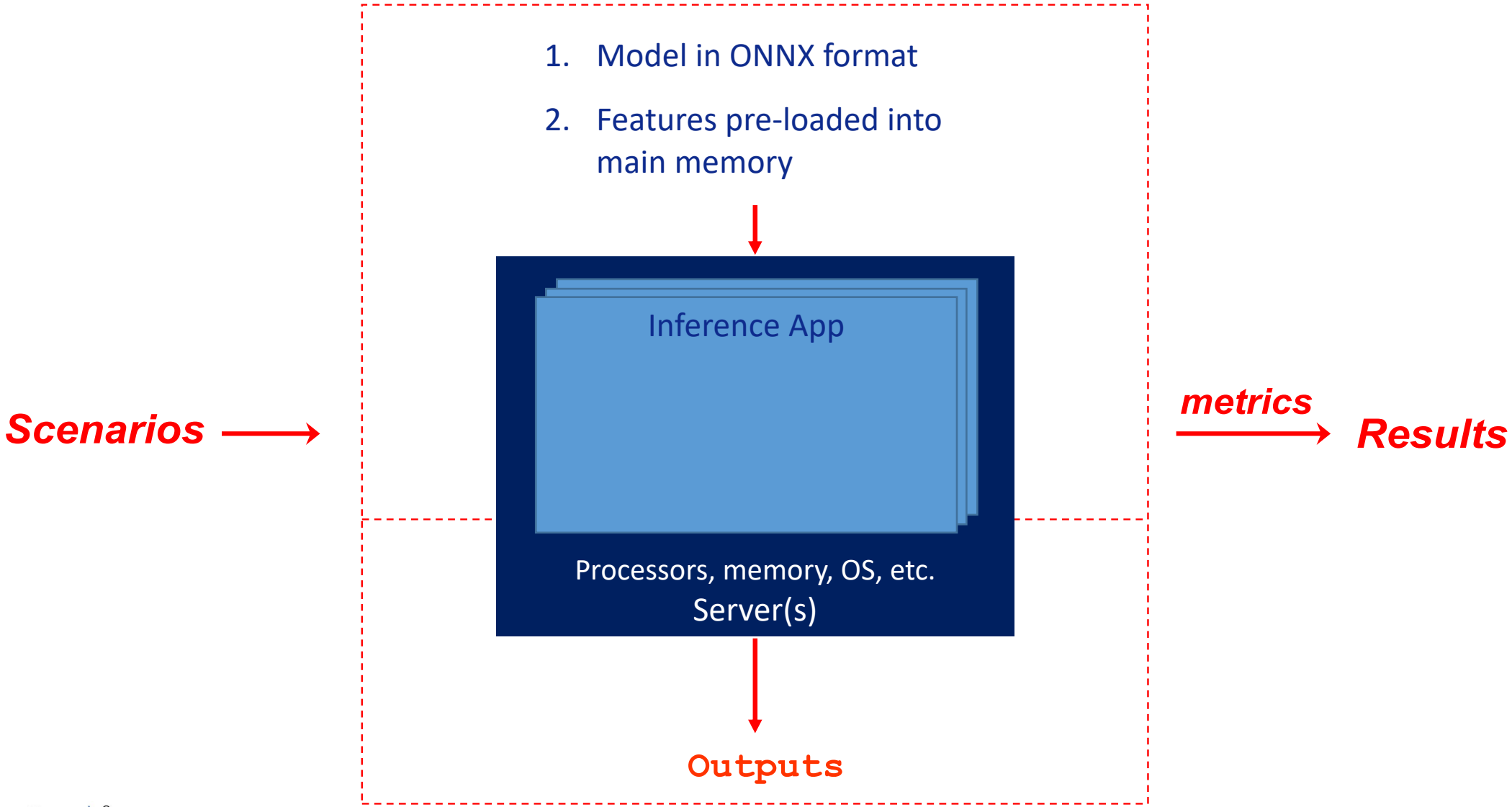


STAC®
SECURITIES TECHNOLOGY ANALYSIS CENTER

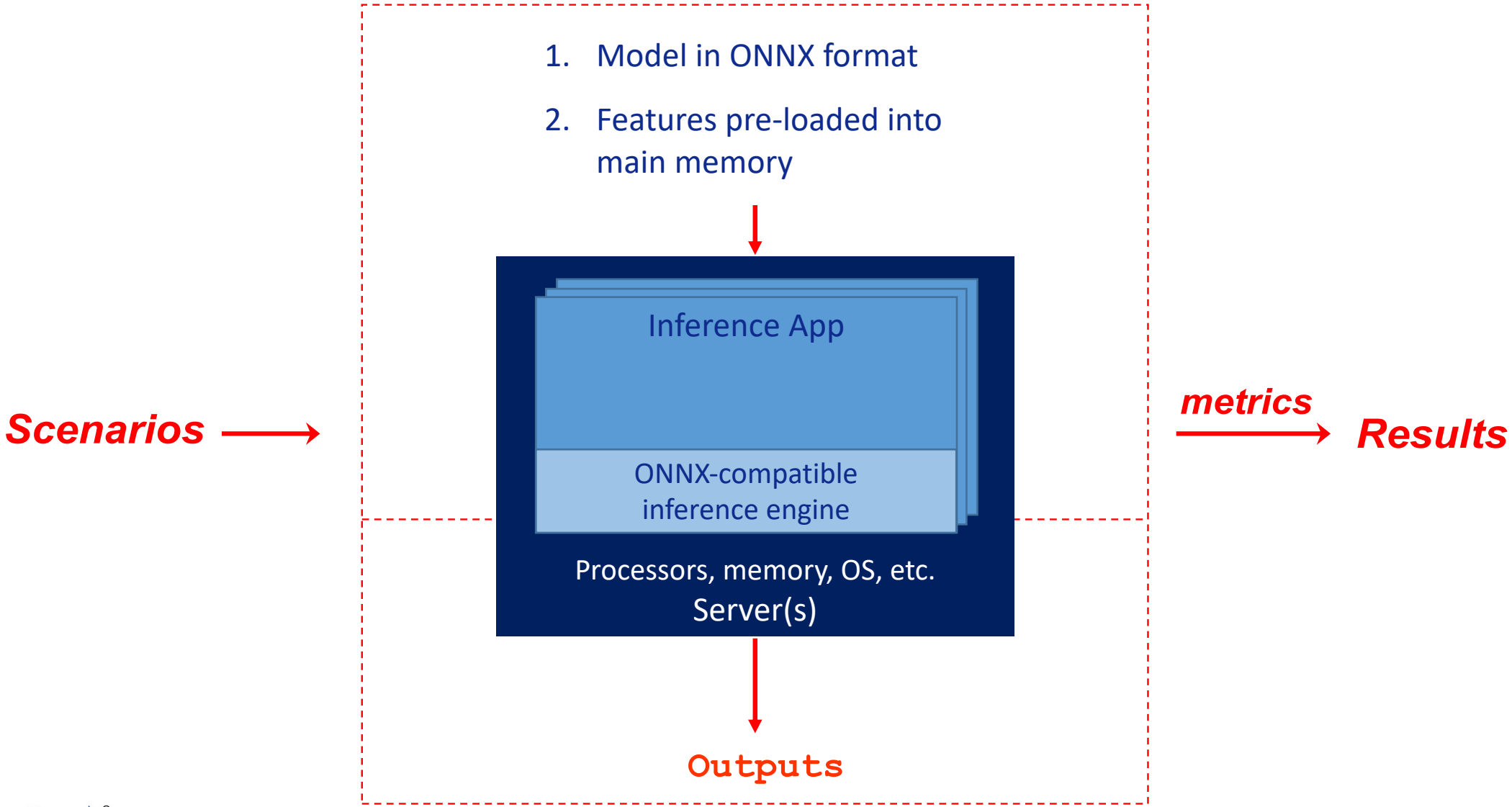
Benchmark construction



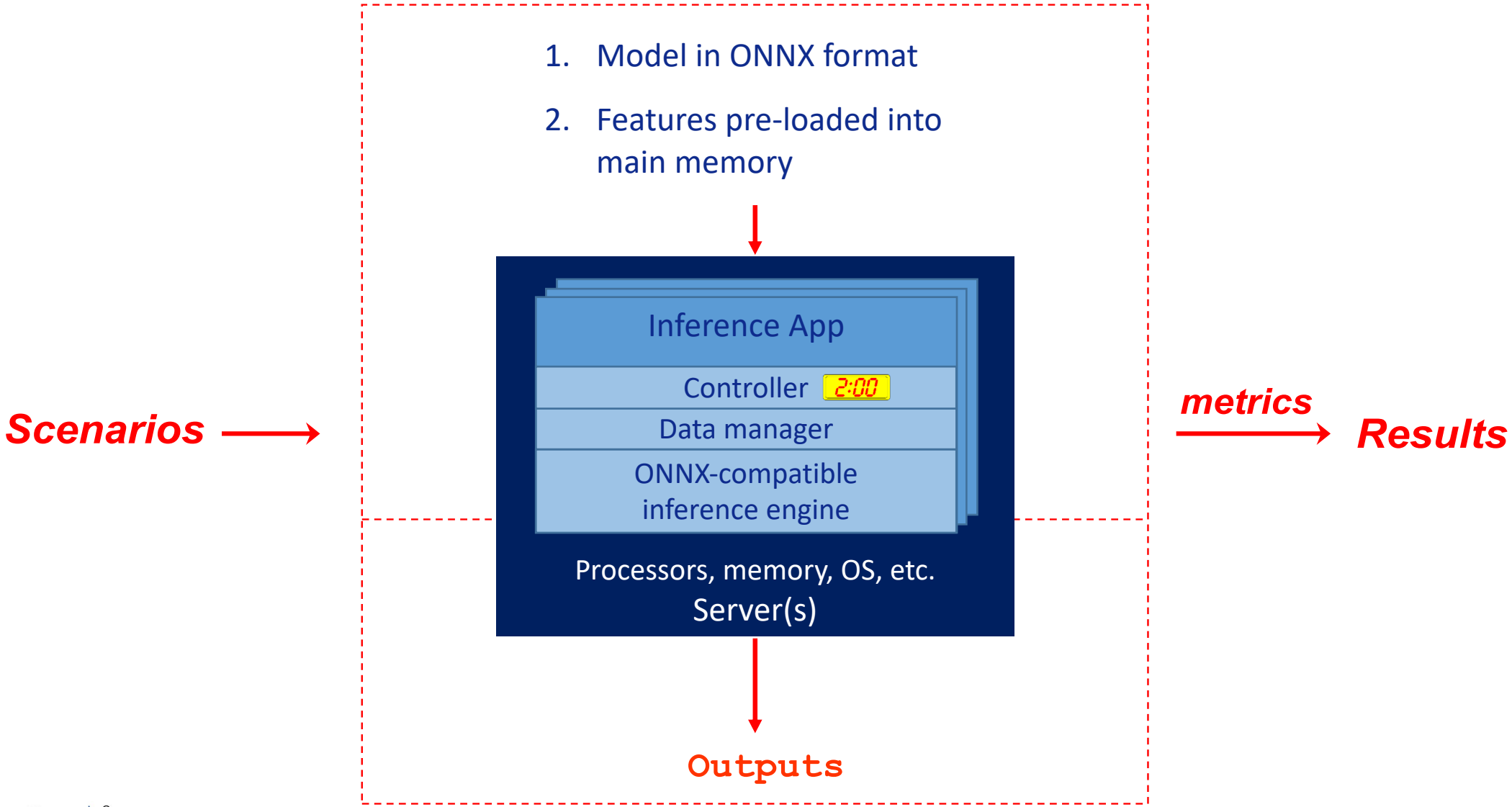
Benchmark construction



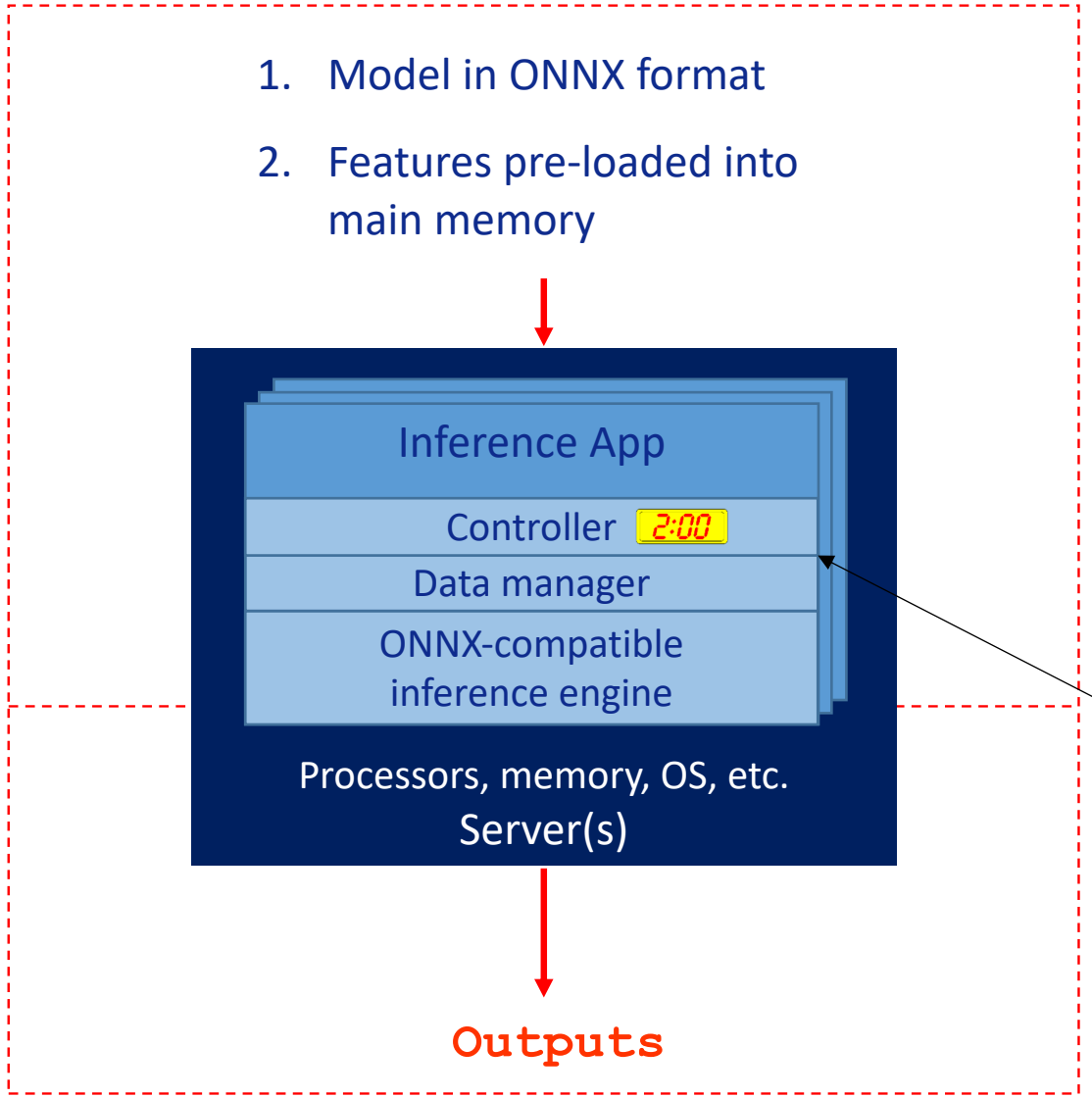
Benchmark construction



Benchmark construction



Benchmark construction



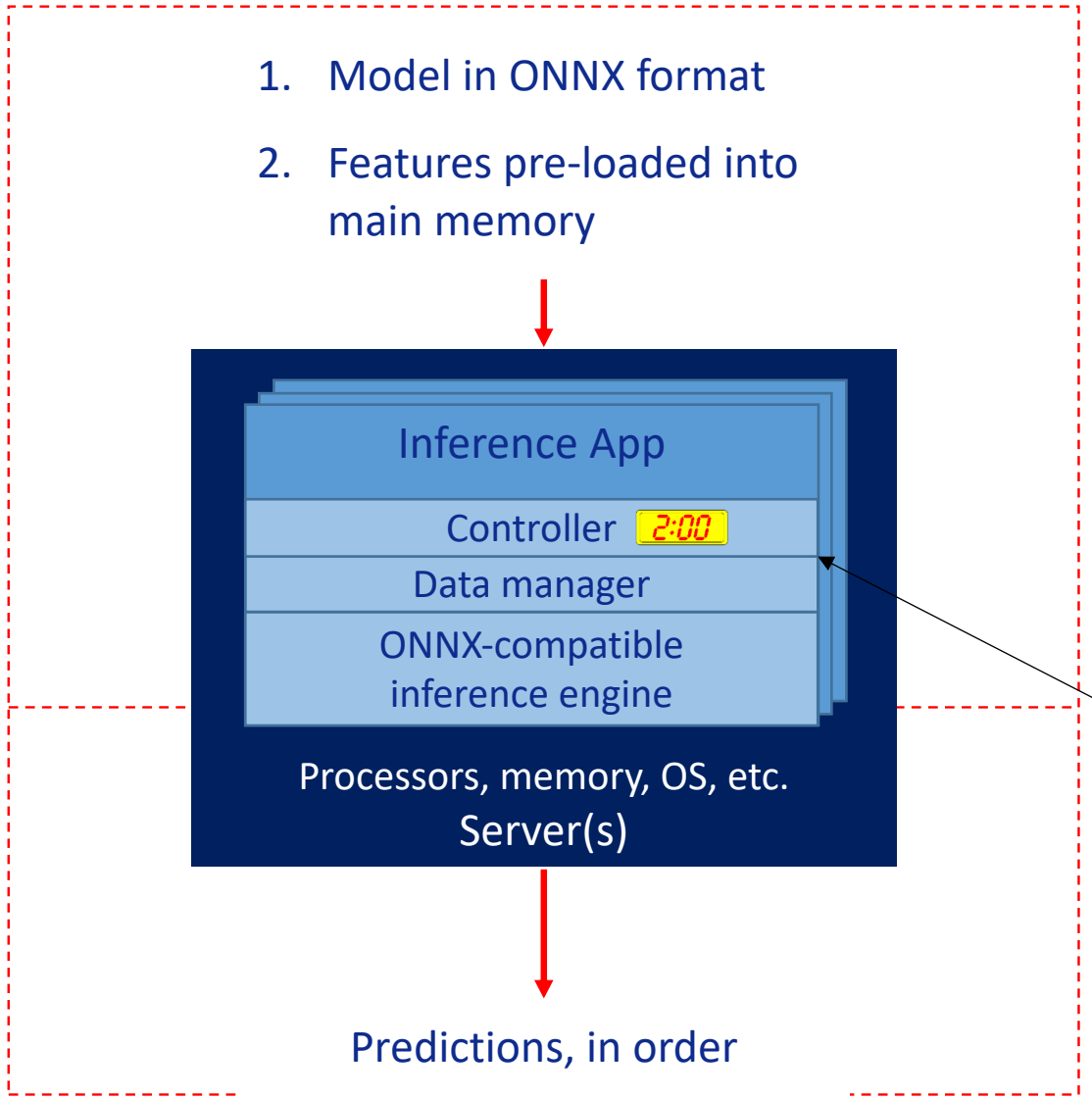
Scenarios →

metrics → **Results**

- Controller releases one timestep at a time to Data Manager. Features are windows, which means Data Manager does a little feature engineering.
- Data Manager decides how to marshall/transfer data for Inference Engine.

STAC®
SECURITIES TECHNOLOGY ANALYSIS CENTER

Benchmark construction



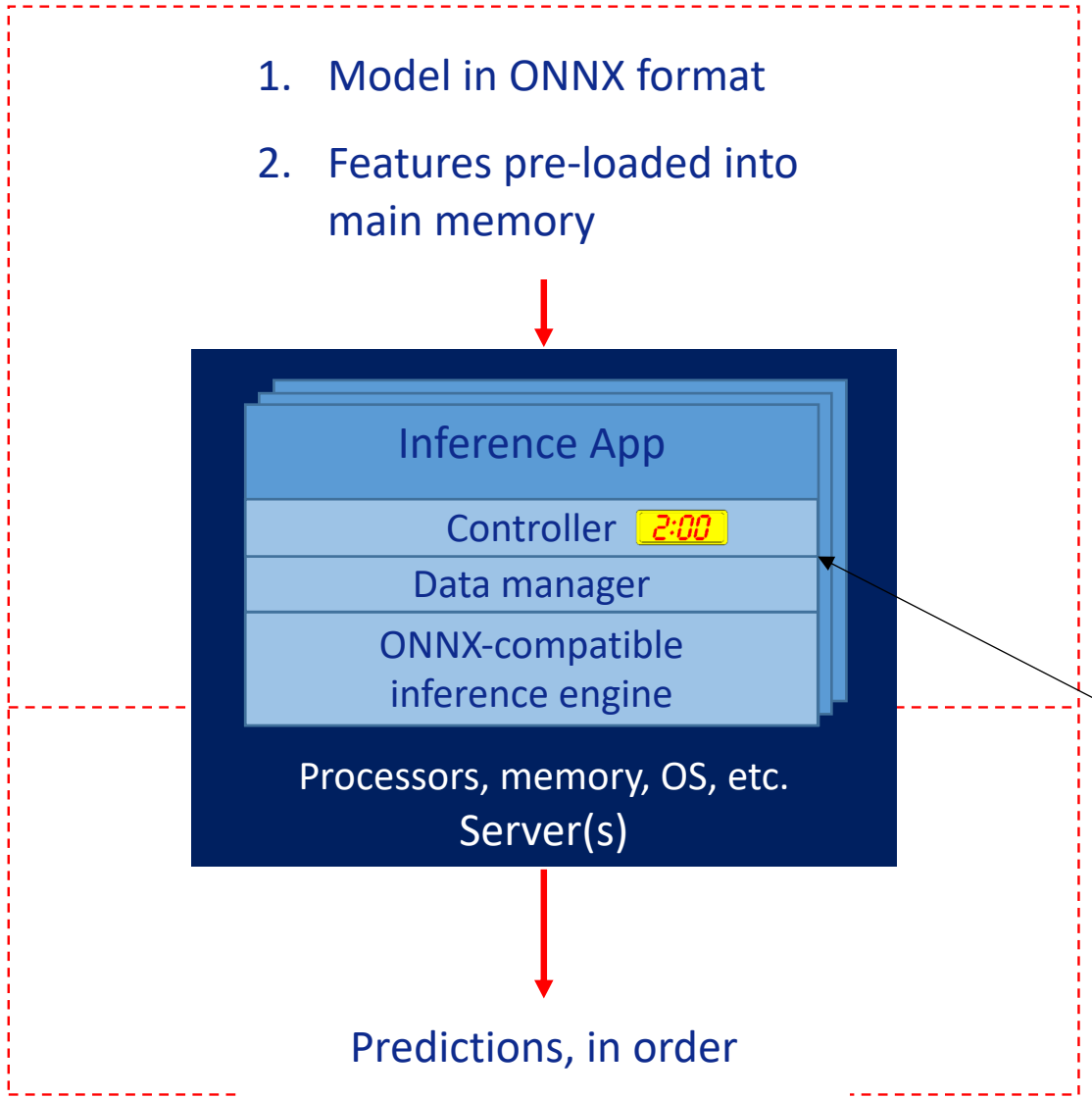
Scenarios →

metrics → **Results**

- Controller releases one timestep at a time to Data Manager. Features are windows, which means Data Manager does a little feature engineering.
- Data Manager decides how to marshall/transfer data for Inference Engine.

STAC®
SECURITIES TECHNOLOGY ANALYSIS CENTER

Benchmark construction



Scenarios →

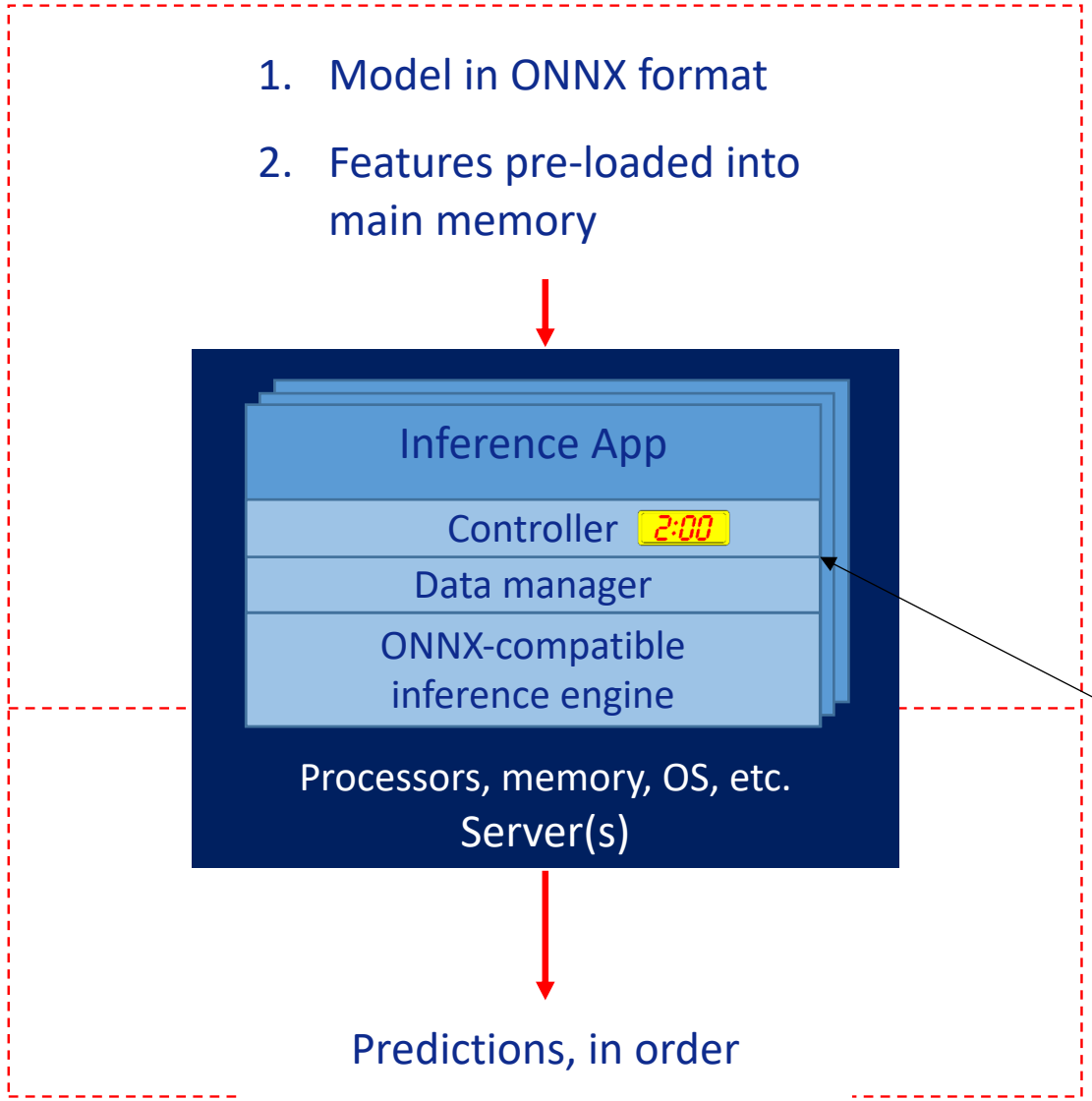
- Per-inference latency
- Throughput
- Validation (Pass/Fail)

→ **Results**

- Controller releases one timestep at a time to Data Manager. Features are windows, which means Data Manager does a little feature engineering.
- Data Manager decides how to marshall/transfer data for Inference Engine.

STAC®
SECURITIES TECHNOLOGY ANALYSIS CENTER

Benchmark construction



Otherwise quiet system

- Per-inference latency
- Throughput
- Validation (Pass/Fail)

Results

- Controller releases one timestep at a time to Data Manager. Features are windows, which means Data Manager does a little feature engineering.
- Data Manager decides how to marshall/transfer data for Inference Engine.

STAC®
SECURITIES TECHNOLOGY ANALYSIS CENTER

SUT design goals

- SUT designers could have various goals:
  - Minimize latency
  - Maximize throughput subject to a latency constraint
    - Maximize for a single model instance
    - Maximize across multiple instances
  - Maximize throughput irrespective of latency (unlikely goal)
  - Maximize number of model instances subject to throughput and latency constraints
  - Maximize resource efficiency (e.g., power efficiency) subject to throughput and latency constraints

STAC
SECURITIES TECHNOLOGY ANALYSIS CENTER

SUTs and results

Four results that are potentially interesting:

- number model instances (= number input streams)
- throughput per model
- total throughput
- latency stats across all inferences

STAC
SECURITIES TECHNOLOGY ANALYSIS CENTER

Comparing results

- A SUT provider can claim superiority of a benchmark result over that of another SUT if the result is a “Pareto improvement”:
  - Result is better and no workload dimension is less demanding
  - Result is the same but at least one workload dimension is more demanding while no other dimension is less demanding.
- Examples:
  - SUT X had 10% lower latency at 100K inf/sec than SUT Y
  - SUT X had 5% higher throughput at 250 μsec latency than SUT Y
  - SUT X had 50% higher throughput because it ran 3 model instances at 50K inf/sec/model whereas SUT Y it could only run 2 model instances at 50K inf/sec/model

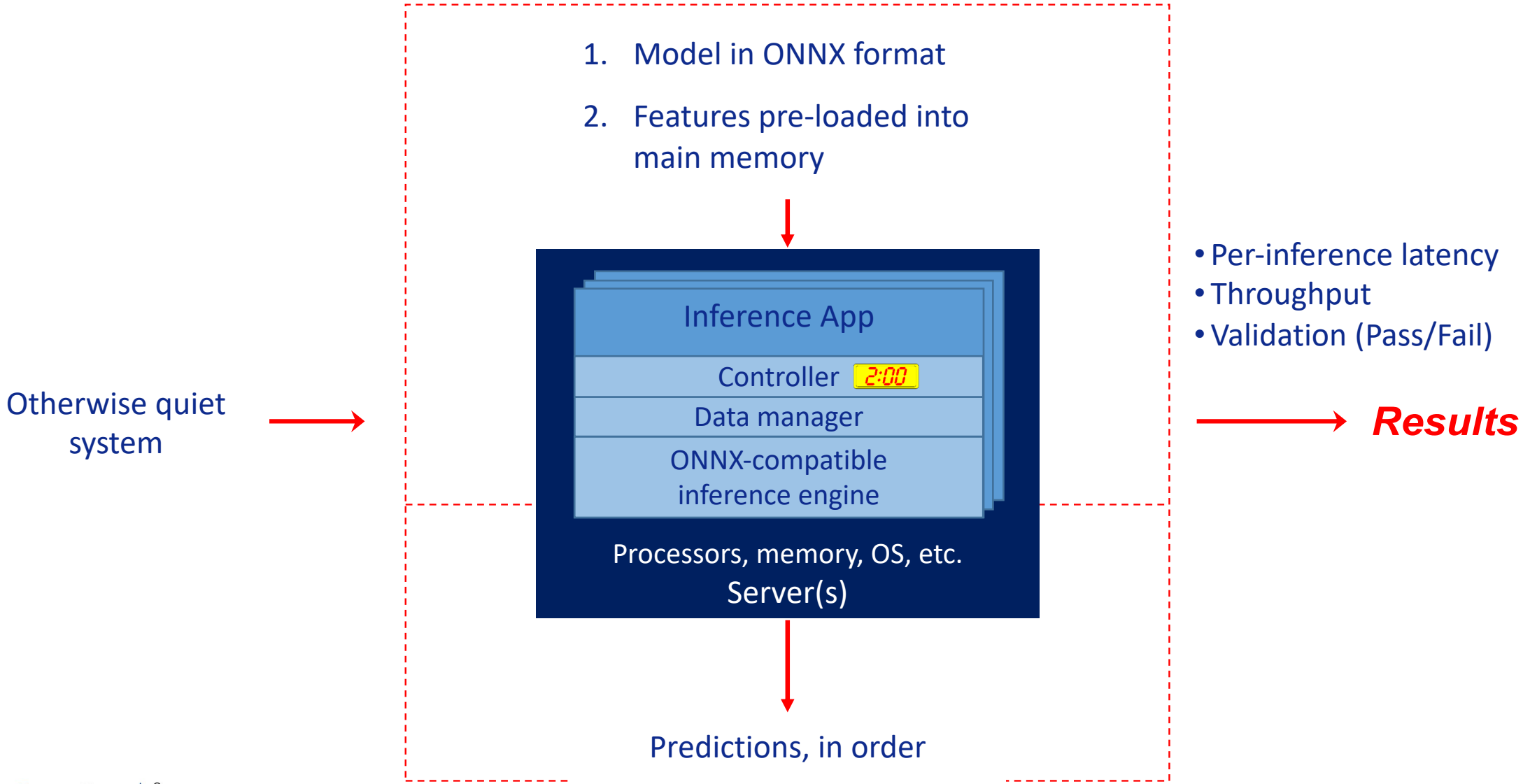
STAC®
SECURITIES TECHNOLOGY ANALYSIS CENTER

Proof of concept

- POC motivations
- Neural network models are now being used or evaluated for market data time series
- Latency-sensitive use cases tend to be exchange-collocated datacenters where power and space are at a premium
- CPUs are commonly used for inference

- So…

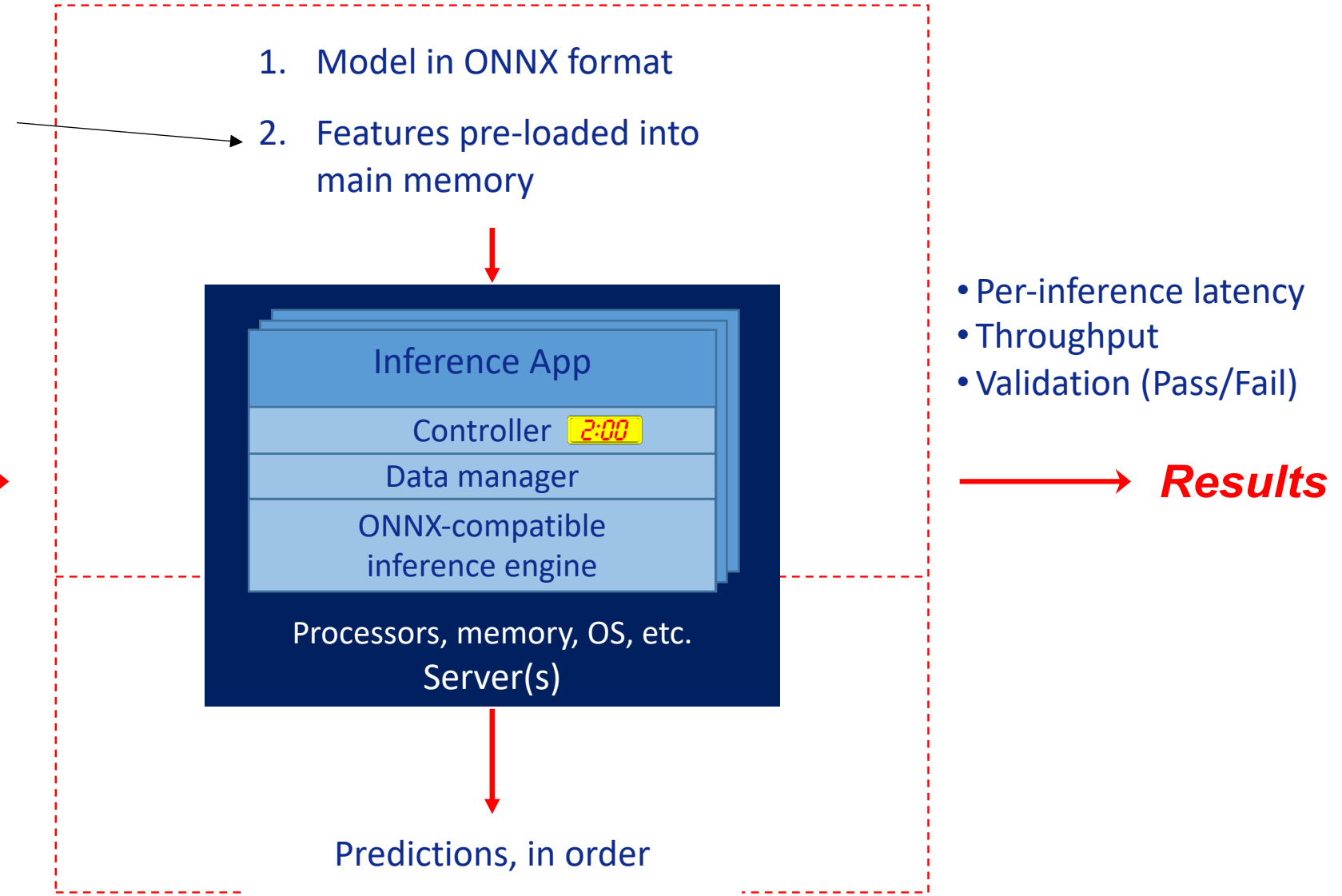
POC benchmark construction



STAC®
SECURITIES TECHNOLOGY ANALYSIS CENTER

POC benchmark construction

- Log return time series for multiple instruments & timesteps
- Signals superimposed on market noise generated by STAC-A2 PATHGEN (Heston stochastic market volatility).



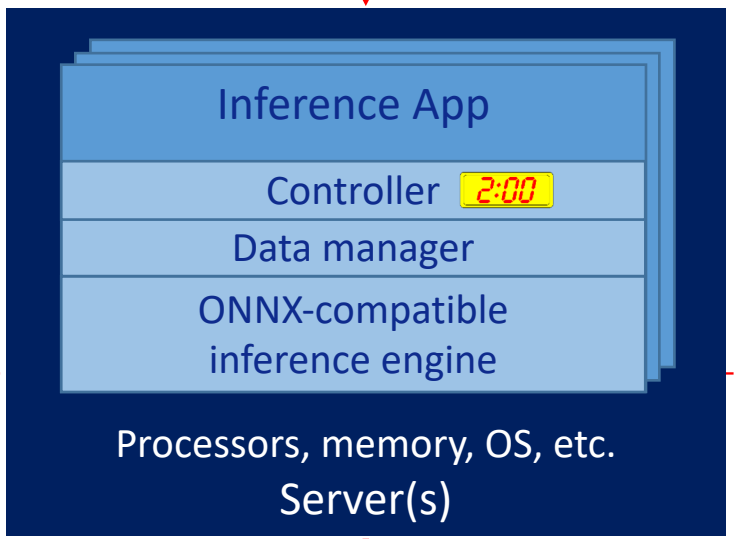
STAC®
SECURITIES TECHNOLOGY ANALYSIS CENTER

POC benchmark construction

- Log return time series for multiple instruments & timesteps
- Signals superimposed on market noise generated by STAC-A2 PATHGEN (Heston stochastic market volatility).

Otherwise quiet system →

1. Model in ONNX format
2. Features pre-loaded into main memory



Predictions, in order

- LSTM model (stateless) that operates on a fixed window size.
- No online retraining.

- Per-inference latency
- Throughput
- Validation (Pass/Fail)

→ ***Results***

STAC®
SECURITIES TECHNOLOGY ANALYSIS CENTER

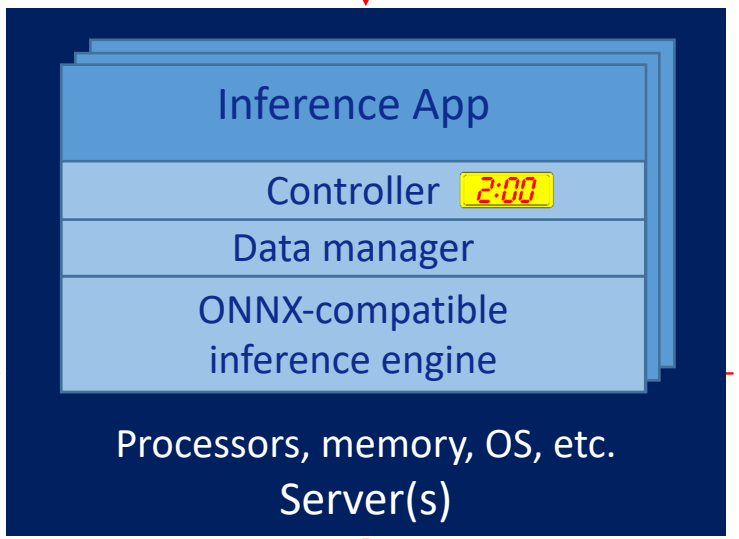
POC benchmark construction

- Log return time series for multiple instruments & timesteps
- Signals superimposed on market noise generated by STAC-A2 PATHGEN (Heston stochastic market volatility).

Otherwise quiet system

1. Model in ONNX format
2. Features pre-loaded into main memory

- LSTM model (stateless) that operates on a fixed window size.
- No online retraining.



- Per-inference latency
- Throughput
- Validation (Pass/Fail)

Results

- Predict a single weighted index value one timestep ahead

Predictions, in order

STAC
SECURITIES TECHNOLOGY ANALYSIS CENTER

POC benchmark construction

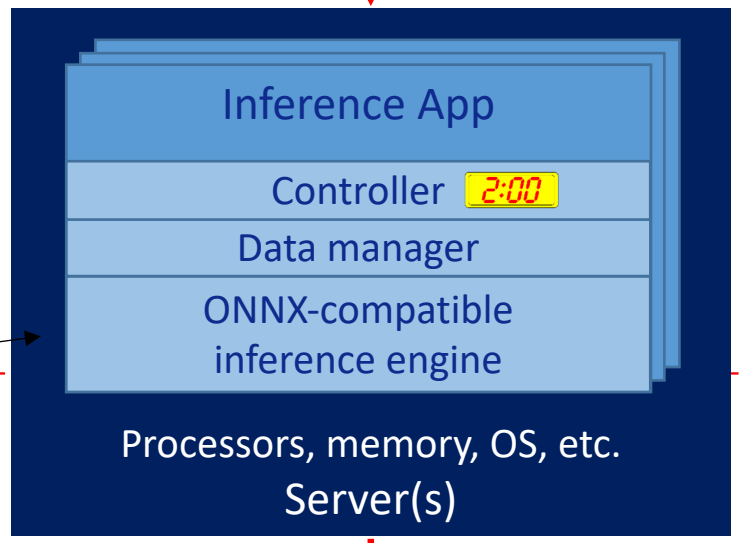
- Log return time series for multiple instruments & timesteps
- Signals superimposed on market noise generated by STAC-A2 PATHGEN (Heston stochastic market volatility).

Otherwise quiet system

- Chose the ONNX runtime with default network configuration

- Predict a single weighted index value one timestep ahead

1. Model in ONNX format
2. Features pre-loaded into main memory



Predictions, in order

- LSTM model (stateless) that operates on a fixed window size.
- No online retraining.

- Per-inference latency
- Throughput
- Validation (Pass/Fail)

Results

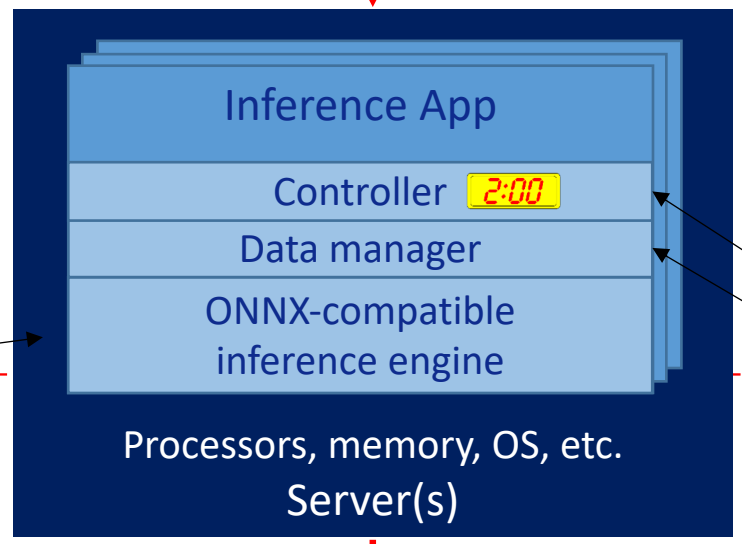
STAC
SECURITIES TECHNOLOGY ANALYSIS CENTER

POC benchmark construction

- Log return time series for multiple instruments & timesteps
- Signals superimposed on market noise generated by STAC-A2 PATHGEN (Heston stochastic market volatility).

1. Model in ONNX format
2. Features pre-loaded into main memory

- LSTM model (stateless) that operates on a fixed window size.
- No online retraining.



- Per-inference latency
- Throughput
- Validation (Pass/Fail)

Otherwise quiet system

Results

- Chose the ONNX runtime with default network configuration

- Rudimentary, single-threaded Python, developed by STAC

- Predict a single weighted index value one timestep ahead

Predictions, in order

STAC
SECURITIES TECHNOLOGY ANALYSIS CENTER

POC SUT platforms

Tested several 2-socket servers with different CPU characteristics (frequency, cores, etc.)

Thanks to the Dell Technologies HPC and AI Innovation Lab for their partnership in providing access to PowerEdge servers for these experiments!

For each platform, created three SUT configurations:

- Config 1 (Min latency): Minimize 99P latency
- Config 2 (Max TP near min): By increasing the number of models, maximize throughput such that 99P latency <= 110% that of Config 1
- Config 3 (Max TP <= 1ms): By increasing the number of models, maximize throughput such that 99P latency <= 1 ms.

STAC
SECURITIES TECHNOLOGY ANALYSIS CENTER

POC results

- Will put report in the STAC Vault.
- For this presentation, the SUT and results are de-identified because:
  - The benchmarks aren't official
  - No vendor had an opportunity to optimize
  - Other than Dell providing access to hardware, no vendor was involved in this project at all

STAC
SECURITIES TECHNOLOGY ANALYSIS CENTER

Normalizing results

- In order to obscure the latency and throughput values, we divide them by a constant
- Latencies are expressed relative to the lowest min
- Throughputs are relative to the per-instance throughput of the configuration with the lowest min latency
  - i.e., the same configuration above
- Bottom line: latencies and throughputs are all divided by the same respective constant so that you can compare magnitudes

STAC
SECURITIES TECHNOLOGY ANALYSIS CENTER

POC results: first takeaway

- Remind your data scientists: tuning matters!
- Data scientists usually don’t have HPC engineering skills
- Even fewer have low-latency skills
- So it’s worth making sure they know that inference performance is quite sensitive to many tuning parameters
- Here are a few basic examples

STAC
SECURITIES TECHNOLOGY ANALYSIS CENTER

Sensitivity to hardware threading

N is the number of physical cores in the SUT

SMT disabled: one hardware thread per physical core

SMT enabled: two hardware threads per physical core

	Throughput	Latency	
SMT disabled		99%ile	max
N models	1.00	1.00	47.55
2N models	0.98	24.27	495.86
SMT enabled			
N models	0.97	1.10	12.27
2N models	1.06	1.96	84.21

N models: SMT disabled slightly higher throughput, noticeably lower latency, but more outliers

2N models: SMT enabled slightly higher throughput, lower latency, and much more consistent.
However, small gain in total throughput vs. N models on same SUT.

STAC
SECURITIES TECHNOLOGY ANALYSIS CENTER

Sensitivity to number of OpenMP threads

#OpenMP threads	Throughput	Latency			
		min	median	99%ile	max
1	1.00	1.00	1.02	1.07	11.07
2	0.79	1.06	1.32	1.41	7.80
4	0.88	1.08	1.16	1.23	7.74
8	0.69	1.17	1.44	1.77	17.94
Default	0.63	1.29	1.58	1.73	68.89

STAC
SECURITIES TECHNOLOGY ANALYSIS CENTER

Comparison of two platforms

To illustrate the kind of comparisons the proposed benchmarks enable, we chose two 2-socket x86 servers running Linux

- Platform A:
  - Lower frequency
  - Higher core count (several times Platform B)
- Platform B:
  - Higher frequency (about double Platform A)
  - Lower core count

STAC
SECURITIES TECHNOLOGY ANALYSIS CENTER

Comparison of two platforms: normalization

Min latency config	Throughput	Latency			
		min	median	99%ile	max
Platform A	0.1	1.1	1.1	1.2	14.8
Platform B	1.0	1.0	1.0	1.1	8.7
Max TP <= 1msec					
Platform A	3.2	1.4	1.7	1.8	99.1
Platform B	1.9	1.0	1.1	1.1	86.3
Max TP near min					
Platform A	1.4	1.1	1.2	1.3	13.0
Platform B	1.9	1.0	1.1	1.1	86.3

All latencies (in blue) divided by the smallest min latency.

All throughputs (in green) divided by the throughput for the config with the smallest min latency.

STAC
SECURITIES TECHNOLOGY ANALYSIS CENTER

Goal: Minimize latency. Use: Platform B

Min latency config	Throughput	Latency			
		min	median	99%ile	max
Platform A	0.1	1.1	1.1	1.2	14.8
Platform B	1.0	1.0	1.0	1.1	8.7
Max TP <= 1msec					
Platform A	3.2	1.4	1.7	1.8	99.1
Platform B	1.9	1.0	1.1	1.1	86.3
Max TP near min					
Platform A	1.4	1.1	1.2	1.3	13.0
Platform B	1.9	1.0	1.1	1.1	86.3

Platform B has the lowest 99P latency in the min latency config

STAC
SECURITIES TECHNOLOGY ANALYSIS CENTER

Goal: Maximize throughput for latency <1ms. Use: Platform A

	Throughput	Latency			
Min latency config		min	median	99%ile	max
Platform A	0.1	1.1	1.1	1.2	14.8
Platform B	1.0	1.0	1.0	1.1	8.7
Max TP <= 1msec					
Platform A	3.2	1.4	1.7	1.8	99.1
Platform B	1.9	1.0	1.1	1.1	86.3
Max TP near min					
Platform A	1.4	1.1	1.2	1.3	13.0
Platform B	1.9	1.0	1.1	1.1	86.3

Platform A has the highest throughput for 99P latency < 1ms

STAC
SECURITIES TECHNOLOGY ANALYSIS CENTER

Goal: Maximize throughput for near-min latency

Min latency config	Throughput	Latency			
		min	median	99%ile	max
Platform A	0.1	1.1	1.1	1.2	14.8
Platform B	1.0	1.0	1.0	1.1	8.7
Max TP <= 1msec					
Platform A	3.2	1.4	1.7	1.8	99.1
Platform B	1.9	1.0	1.1	1.1	86.3
Max TP near min					
Platform A	1.4	1.1	1.2	1.3	13.0
Platform B	1.9	1.0	1.1	1.1	86.3

In their “near min latency” configurations, Platform B has higher throughput and lower latency, but bigger outliers.

Summary

- STAC has proposed specs for real-time inference on time series in response to requests
- The benchmark framework appears to satisfy most of the requirements
- We're looking for more firms to give feedback
- Please visit:

www.STACresearch.com/AI

STAC
SECURITIES TECHNOLOGY ANALYSIS CENTER