Xelera Silva

Benchmark Report

Xelera Technologies GmbH
Rheinstr. 40-42
64283 Darmstadt

Registergericht (Legal Domicile):
Darmstadt HRB 97805
USt.-IdNr. DE318360521

bank details
IBAN: DE 17 5085 0150 0000 7709 90
BIC: HELADEF1DAS

info@xelera.io
www.xelera.io
phone: +49 6151 6290900

Contents

References

[1] Blackcore Technologies, "SPR-M High Performance Server," [Online]. Available: https://blackcoretech.com/firefly/file/get?id=qrqhCt3deZacM5MeCtx0fw. [Accessed 10 04 2024].

[2] Xelera Technologies, "Xelera Silva," [Online]. Available: https://www.xelera.io/products/silva. [Accessed 10 4 2024].

1 Xelera Silva

Gradient Boosting frameworks such as XGBoost and LightGBM are widely used in financial trading systems, ransomware and DDOS detection systems, and recommender systems. Xelera Silva provides best-in-class latency and throughput for XGBoost and LightGBM and Random Forest inference by leveraging commercial off-the-shelf data-center grade FPGA accelerators.

The Xelera Silva software [1] loads machine learning models from XGBoost, LightGBM, ONNX ML Tools. The models are executed for inference on AMD Alveo platforms. The user application interacts with the accelerator software via a C/C++, C# or Python API. This document describes the latency benchmark tests on a Blackcore Technologies SPR-M server [2]. The tests are done on the upcoming Xelera Silva 7.0.0 release:

- Test 1: single model inference
- Test 2: simultaneous and synchronous inference with 4 models
- Test 3: simultaneous and asynchronous inference with 4 models

2 Test 1: single model inference

This benchmark validates the LightGBM model inference latency on the system specified in Table below.

Table 1: System-under-test

Server	"SPR-M" – Sapphire Rapids [2] CPU: Intel® Xeon® w7-2495x CPU Frequency: Up to 24 Cores @ 4.8GHz (all-core) SSE CPU Cache: 45MB @ 3.2GHz Chipset: W790 Memory: 128GB Overclocked ECC RDIMM
OS	Linux Rapids 9.3
PCIe interface	Gen4 x8
AMD Alveo Card	U50 with Xelera PCIe ULL shell
Driver	Xelera PCIe ULL 1.0.0
ML Inference Software	Xelera Silva 7.0.0

2.1 Test Description

The roundtrip latency at the API interface (Tout – Tin) is measured when running the inference for the model configuration in Table 2.

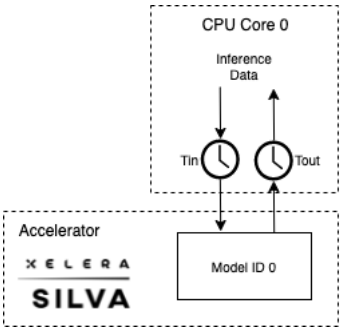


Table 2: Model configuration

Model Type	LightGBM regression
Dataset	Synthetic Random
Number of Features	128
Number of Trees	1000
Number of Levels*	5,6,7,8
Batch Size	1
Numerical Features	Yes
Categorical Features	No

* It should be noted that increasing the number of levels of the models increases the amount of inference computation.

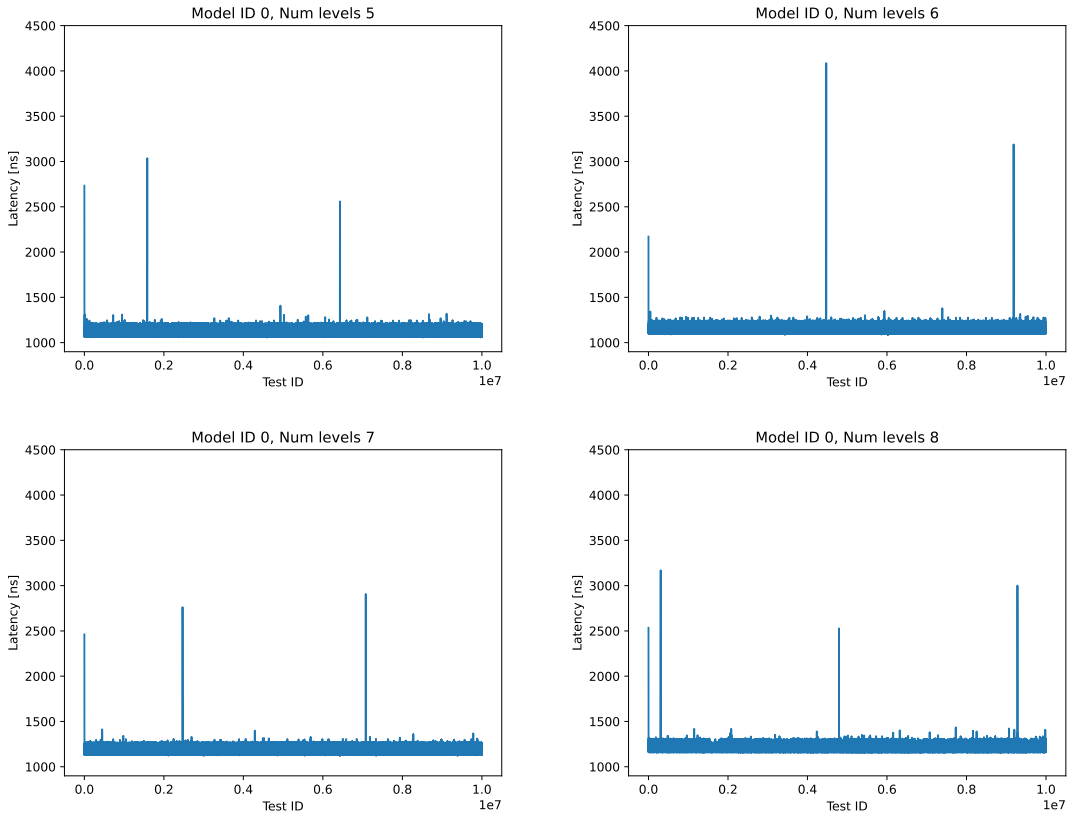
For each model configuration, the test involves running inference on one model. Each process is pinned to a CPU core 0. The test is conducted 10,000,000 times.

2.2 Results

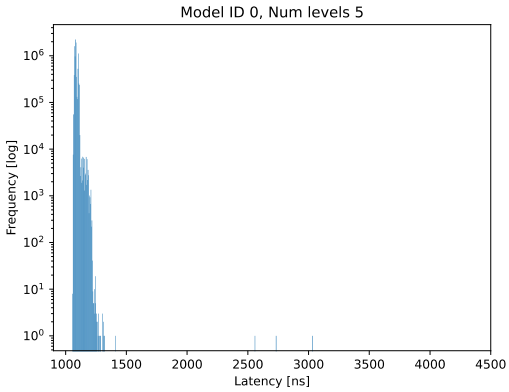
Table 3: Latency measurements

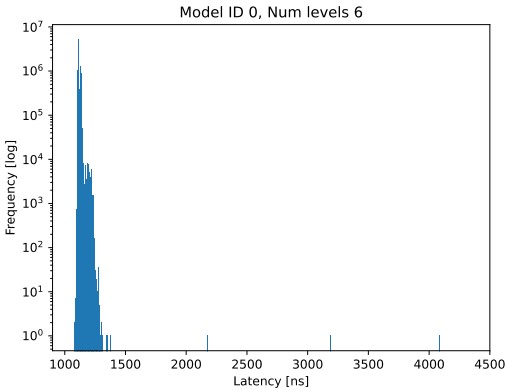
Number of tree levels in model	Model ID / CPU ID	Min Latency [ns]	Max Latency [ns]	Median Latency [ns]	99th Percentile Latency [ns]
5	0	1056	3036	1083	1115
6	0	1081	4086	1113	1143
7	0	1119	2904	1143	1170
8	0	1155	3169	1177	1207

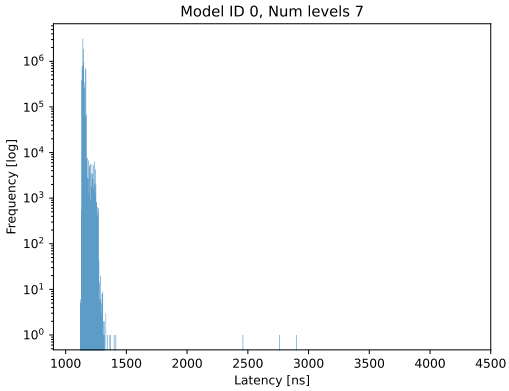
The following figures show the latency graph (the measured latency per test run) for the above tests.

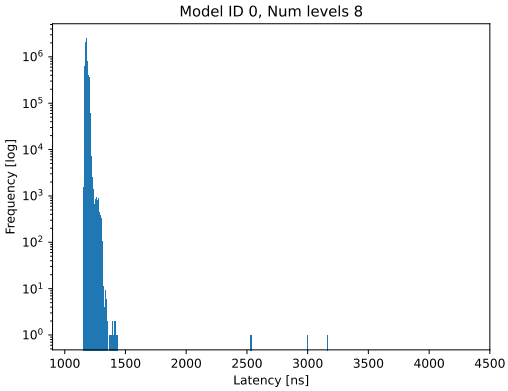


The following figures show the histograms for the above tests.


Model ID 0, Num levels 5
Frequency [log]
Latency [ns]


Model ID 0, Num levels 6
Frequency [log]
Latency [ns]


Model ID 0, Num levels 7
Frequency [log]
Latency [ns]


Model ID 0, Num levels 8
Frequency [log]
Latency [ns]

3 Test 2: simultaneous and synchronous inference with 4 models

This benchmark validates the LightGBM model inference latency on the system specified in the Table 1 below.

Table 4: System-under-test

Server	"SPR-M" – Sapphire Rapids [2] CPU: Intel® Xeon® w7-2495x CPU Frequency: Up to 24 Cores @ 4.8GHz (all-core) SSE CPU Cache: 45MB @ 3.2GHz Chipset: W790 Memory: 128GB Overclocked ECC RDIMM
OS	Linux Rapids 9.3
PCIe interface	Gen4 x8
AMD Alveo Card	U50 with Xelera PCIe ULL shell
Driver	Xelera PCIe ULL 1.0.0
ML Inference Software	Xelera Silva 7.0.0

3.1 Test Description

The roundtrip latency at the API interface ($Tout_x$ - Tin) is measured when running the inference for the model configuration in 5.

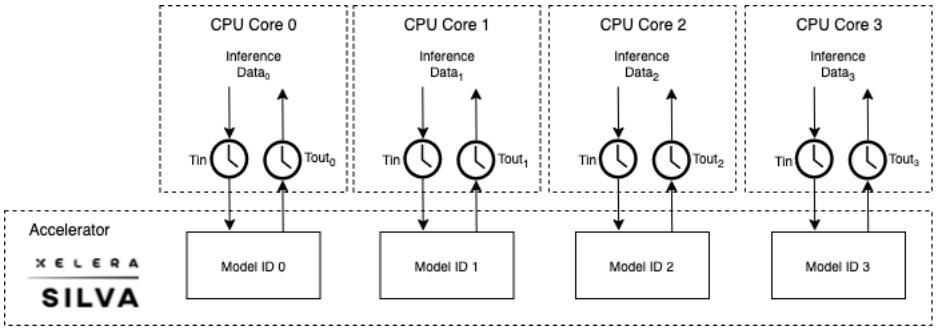


Table 5: Model configuration

Model Type	LightGBM regression
Dataset	Synthetic Random
Number of Features	100
Number of Trees	1000
Number of Levels*	5,6,7,8
Batch Size	1
Numerical Features	Yes
Categorical Features	No

* It should be noted that increasing the number of levels of the models increases the amount of inference computation.

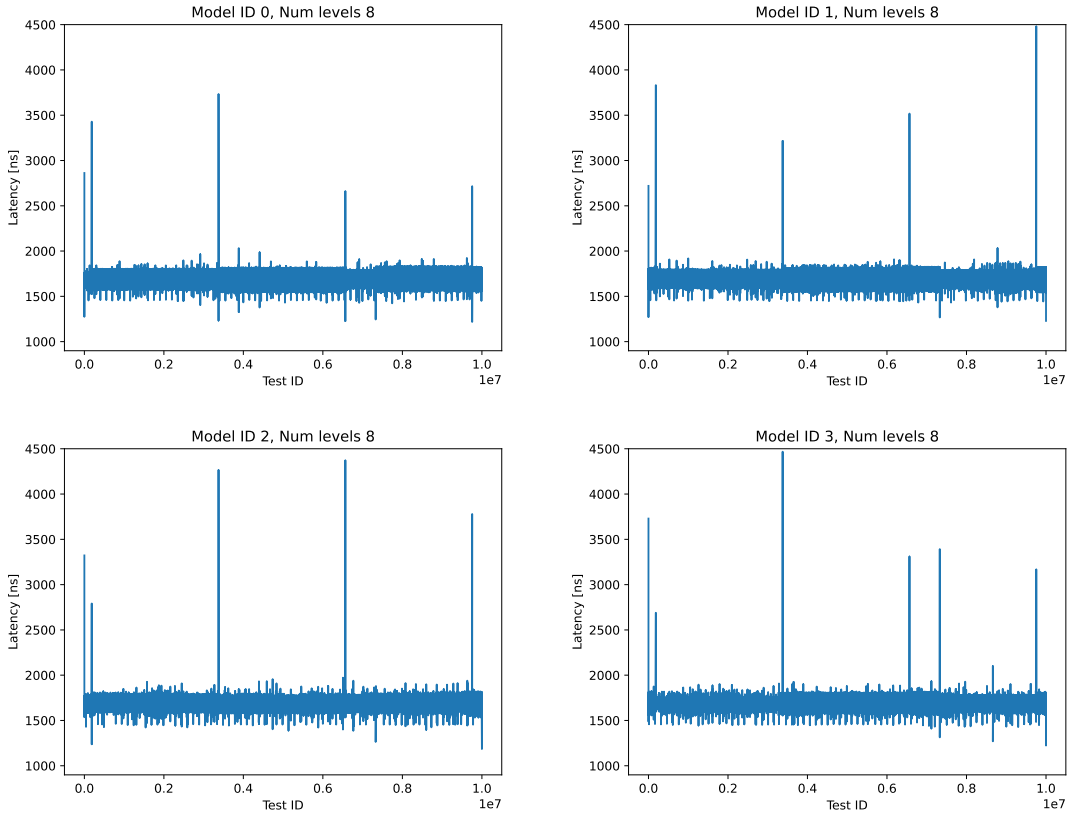
For each model configuration, the test involves running inference with four models (IDs from 0 to 3) simultaneously in the **synchronous** mode (4 independent processes, each of them accessing exclusively one model). Each process is pinned to a CPU core (0 to 3). The test is conducted 10,000,000 times.

3.2 Results

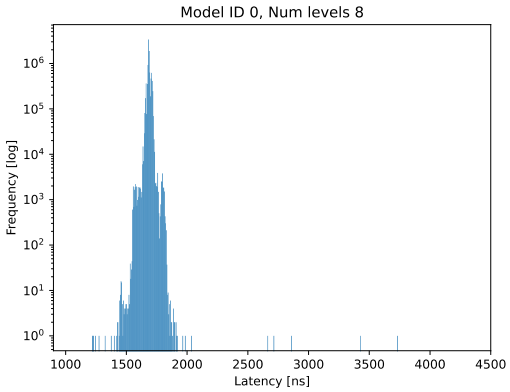
Table 6: Latency measurements

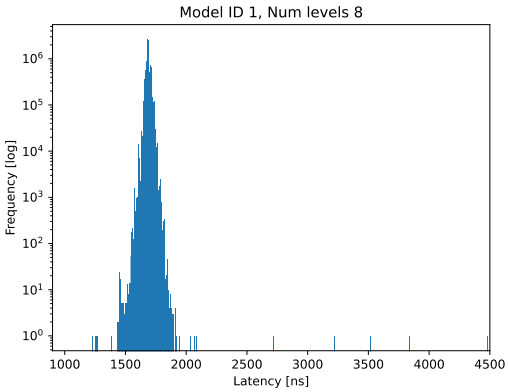
Number of tree levels in model	Model ID / CPU ID	Min Latency [ns]	Max Latency [ns]	Median Latency [ns]	99th Percentile Latency [ns]
5	0	1063	5164	1303	1339
5	1	1061	5434	1302	1343
5	2	1101	4653	1302	1340
5	3	1089	3701	1302	1346
6	0	1116	4088	1429	1471
6	1	1117	3345	1430	1470
6	2	1171	3893	1429	1471
6	3	1125	3608	1430	1467
7	0	1196	4052	1557	1597
7	1	1176	3189	1557	1595
7	2	1257	4248	1556	1597
7	3	1237	4323	1557	1600
8	0	1217	3734	1685	1722
8	1	1229	4483	1685	1742
8	2	1185	4374	1684	1742
8	3	1226	4468	1685	1736

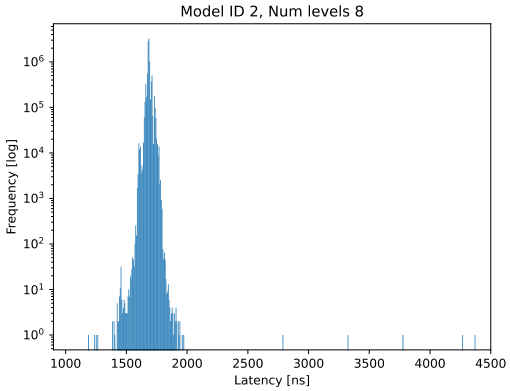
The following figures show the latency graph (the measured latency per test run) for the above tests when the number of levels is 8 for the four models running simultaneously in asynchronous mode.

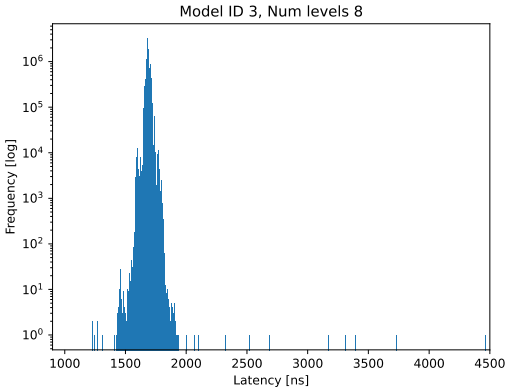


The following figures show the histograms for the above tests when the number of levels is 8 for the four models running simultaneously in asynchronous mode.


Model ID 0, Num levels 8
Frequency [log]
Latency [ns]


Model ID 1, Num levels 8
Frequency [log]
Latency [ns]


Model ID 2, Num levels 8
Frequency [log]
Latency [ns]


Model ID 3, Num levels 8
Frequency [log]
Latency [ns]

4 Test 3: simultaneous and asynchronous inference with 4 models

This benchmark validates the LightGBM model inference latency on the system specified in the Table 1 below.

Table 7: System-under-test

Server	"SPR-M" – Sapphire Rapids [2] CPU: Intel® Xeon® w7-2495x CPU Frequency: Up to 24 Cores @ 4.8GHz (all-core) SSE CPU Cache: 45MB @ 3.2GHz Chipset: W790 Memory: 128GB Overclocked ECC RDIMM
OS	Linux Rapids 9.3
PCIe interface	Gen4 x8
AMD Alveo Card	U50 with Xelera PCIe ULL shell
Driver	Xelera PCIe ULL 1.0.0
ML Inference Software	Xelera Silva 7.0.0

4.1 Test Description

The roundtrip latency at the API interface ($Tout_x$ – Tin_x) is measured when running the inference for the model configuration in Table 8.

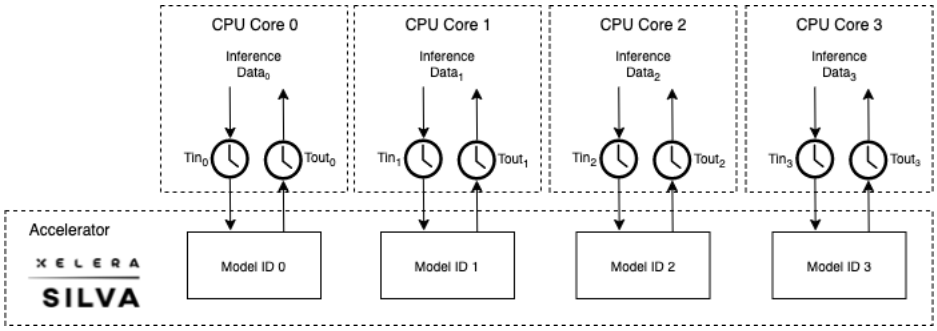


Table 8: Model configuration

Model Type	LightGBM regression
Dataset	Synthetic Random
Number of Features	100
Number of Trees	1000
Number of Levels*	5,6,7,8
Batch Size	1
Numerical Features	Yes
Categorical Features	No

* It should be noted that increasing the number of levels of the models increases the amount of inference computation.

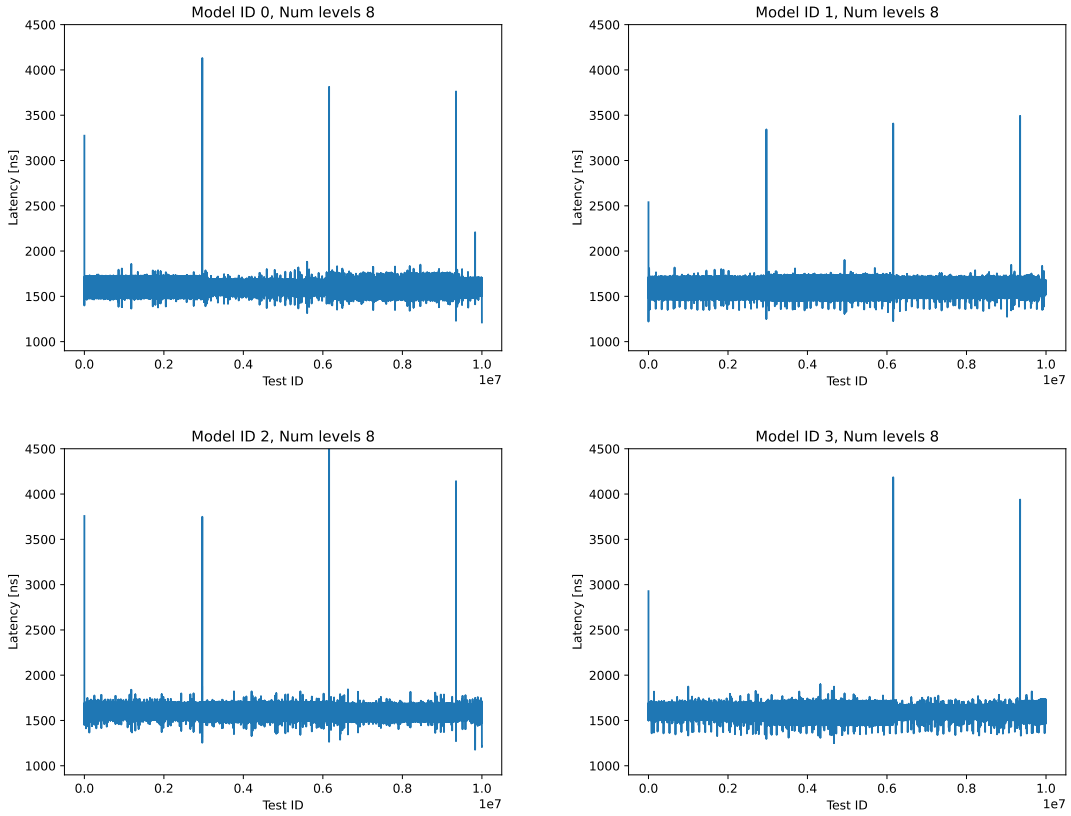
For each model configuration, the test involves running inference with four models (IDs from 0 to 3) simultaneously in the **asynchronous** mode (4 independent processes, each of them accessing exclusively one model). Each process is pinned to a CPU core (0 to 3). The test is conducted 10,000,000 times.

4.2 Results

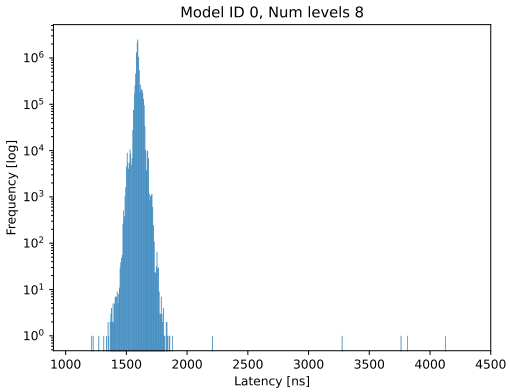
Table 9: Latency measurements

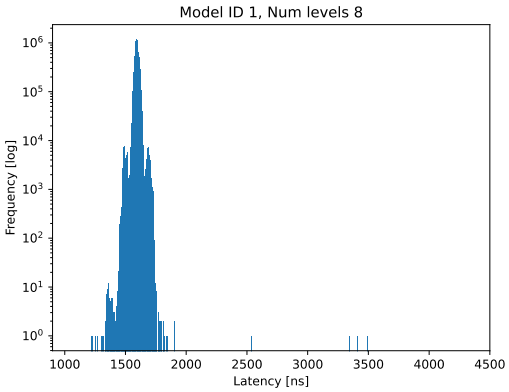
Number of tree levels in model	Model ID / CPU ID	Min Latency [ns]	Max Latency [ns]	Median Latency [ns]	99th Percentile Latency [ns]
5	0	1068	29721	1211	1254
5	1	1048	30142	1209	1250
5	2	1078	31005	1206	1245
5	3	1080	30660	1207	1254
6	0	1119	3195	1337	1374
6	1	1123	3933	1336	1375
6	2	1134	5975	1334	1370
6	3	1116	3547	1334	1376
7	0	1142	3733	1467	1504
7	1	1146	3426	1465	1502
7	2	1144	4593	1461	1499
7	3	1138	4117	1463	1503
8	0	1212	4132	1592	1648
8	1	1221	3491	1593	1638
8	2	1177	4682	1589	1650
8	3	1250	4186	1591	1633

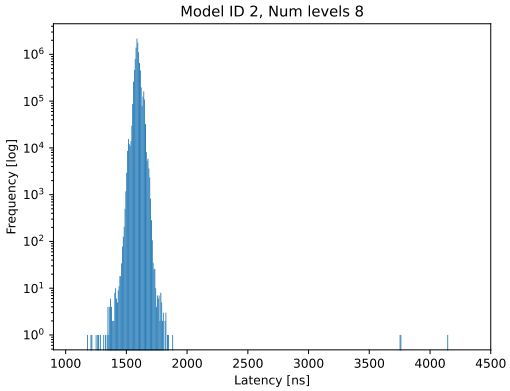
The following figures show the latency graph (the measured latency per test run) for the above tests when the number of levels is 8 for the four models running simultaneously in asynchronous mode.

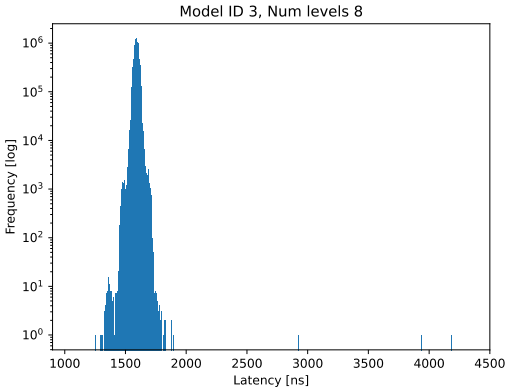


The following figures show the histograms for the above tests when the number of levels is 8 for the four models running simultaneously in asynchronous mode.


Model ID 0, Num levels 8
Frequency [log]
Latency [ns]


Model ID 1, Num levels 8
Frequency [log]
Latency [ns]


Model ID 2, Num levels 8
Frequency [log]
Latency [ns]


Model ID 3, Num levels 8
Frequency [log]
Latency [ns]