LightGBM Inference

Benchmark Report

Accelerator Card: Napatech NT200A02 SmartNIC

Server: AMD Ryzen 9 7950X3D CPU

Contents

1 Xelera Silva

Gradient Boosting frameworks like XGBoost, LightGBM, and CatBoost are widely employed in financial trading systems, ransomware and DDOS detection systems, and recommender systems. Xelera Silva offers best-in-class latency and throughput inference by utilising commercial off-the-shelf data centre-grade FPGA accelerators.

The Xelera Silva software, developed by Xelera Technologies, enables the loading of machine learning models from various frameworks, including XGBoost, LightGBM, CatBoost, and ONNX ML Tools. These models are then executed for inference on Napatech accelerator platforms. The user application communicates with the accelerator through either a C/C++ or Python API.

This document presents the latency benchmark tests conducted on a Napatech NT200A02 SmartNIC.

2 Test 1: Profile PCIe latency

The PCIe latency for the Napatech NT200A02 SmartNIC has been measured with the open-source tool pcie-lat.
The latencies are measured by calculating the time taken to read a 32-bit word from a PCIe device using a Linux kernel module on the system specified in the Table 1 below.

Table 1: System under test

Server	CPU: AMD Ryzen 9 7950X3D CPU Frequency: 16-Core Processor @ 4.20-5.70GH CPU Cache: 128 MiB (L3) Memory: 4 x 32GiB @4.8GHz
OS	Ubuntu 22.04
Driver	pcie-lat
Software	pcie-lat

2.1 Results

Figure 1 displays the latency statistics for reading a 32-bit word from a PCIe device. The y-axis represents the fraction of inference measurements that fall below a specified latency on the x-axis.

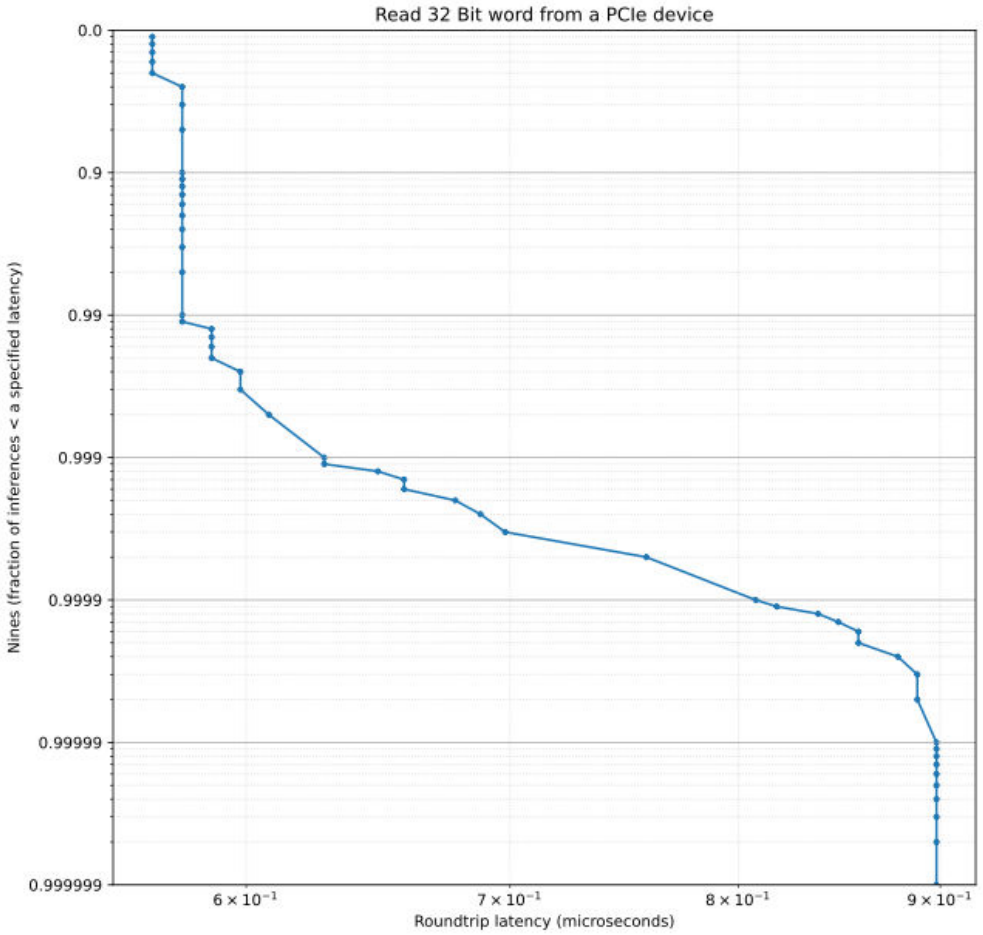

Figure 1: Latency statistic 32-bit word read form PCIe device

Table 2 compares the minimum, maximum, median (50th percentile) and the 99th percentile latency (in microseconds) of the graphs above.

Table 2 : Latency statistics 32-bit word PCIe read

Model ID	**Minimum latency**	**Maximum latency**	**Median latency (50th percentile)**	**99th percentile latency**
0	0.558	0.898	0.568	0.578

3 Test 2: Single Model Inference

This benchmark validates the LightGBM model inference latency on the system specified in Table 3 below.

Table 3: System-under-test

Server	CPU: AMD Ryzen 9 7950X3D CPU Frequency: 16-Core Processor @ 4.20-5.70GH CPU Cache: 128 MiB (L3) Memory: 4 x 32GiB @4.8GHz
OS	Ubuntu 22.04
PCIe interface	Gen3 x16
Accelerator Card	Napatech NT200A02 SmartNIC with Xelera PCIe ULL shell
Driver	Xelera PCIe ULL 2.13.0
ML Inference Software	Xelera Silva 7.10.0

The Xelera Silva ML Inference software was compared against software frameworks designed to accelerate gradient boosting machine learning models inference. The software frameworks under comparison are listed in Table 4 below.

Table 4: Compared software frameworks

ML Inference Software	**Version**	**Description**
Intel oneDAL	2024.5.0	Intel CPU-optimized ML inference software
Xelera Silva	7.4.0	FPGA-accelerated ML inference software

Xelera Silva is the sole FPGA-accelerated ML inference software in this comparison. In contrast, the other framework relies solely on CPU optimisations to enhance the inference speed of gradient boosting models inference. These optimisations include utilising vector extension instructions, branch prediction, and integer comparisons.

3.1 Test Description

The roundtrip latency at the API interface (Tout – Tin) is measured when running the inference for a small model configuration (Table 5) and a big model configuration (Table 6).

Table 5: Small model configuration

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

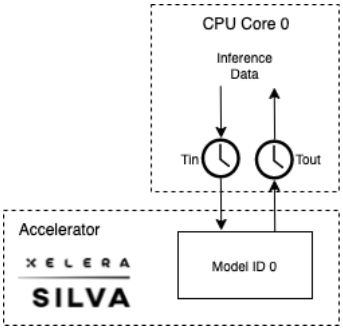

Table 6: Big model configuration

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

For each software framework configuration, the test involves running inference on both models. Each process is assigned to a specific CPU core. The test is conducted a million times.

3.2 Results Small Model

Figure 2 shows the latency statistics of Xelera Silva with third-party software frameworks when running the small model. The y-axis represents the fraction of inference measurements that fall below a specified latency on the x-axis.

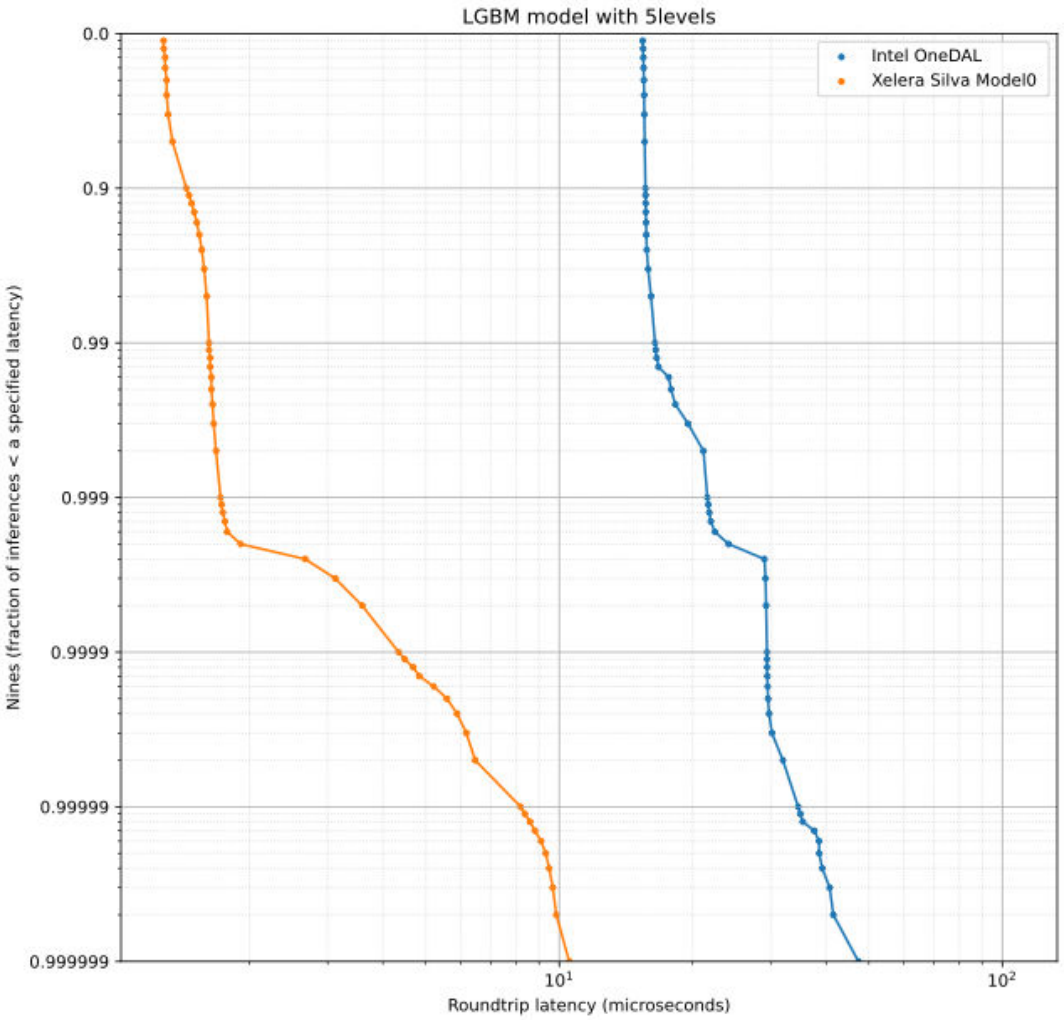

Figure 2 : Latency comparison for single-model inference

Table 7 compares the minimum, maximum, median (50th percentile) and the 99th percentile latency (in microseconds) of the graphs above.

Table 7: Latency statistics small model

ML Inference Software	Minimum latency	Maximum latency	Median latency (50th percentile)	99th percentile latency
Intel oneDAL	15.159	47.340	15.520	16.450
Xelera Silva	1.219	106.589	1.300	1.620

3.1 Results Big Model

Figure 3 shows the latency statistics of Xelera Silva with third-party software frameworks when running the big model. The y-axis represents the fraction of inference measurements that fall below a specified latency on the x-axis.

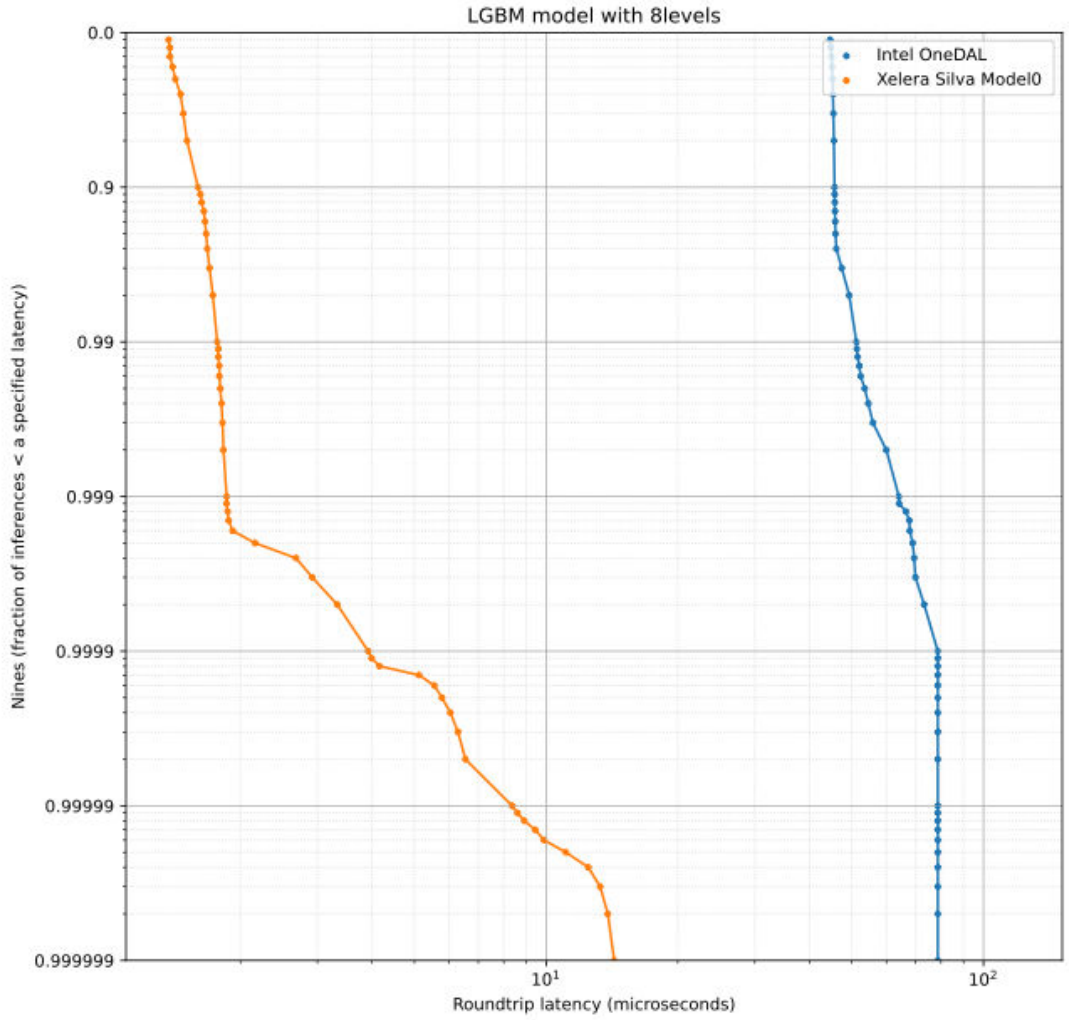

Figure 3 : Latency comparison for single-model inference

Table 8 compares the minimum, maximum, median (50th percentile) and the 99th percentile latency (in microseconds) of the graphs above.

Table 8: Latency statistics big model

ML Inference Software	Minimum latency	Maximum latency	Median latency (50th percentile)	99th percentile latency
Intel oneDAL	42.840	78.630	45.170	51.212
Xelera Silva	1.320	120.969	1.420	1.770

4 Test 3: Simultaneous and Asynchronous Inference with 4 Models

This benchmark validates the LightGBM model inference latency on the system specified in Table 9. In this test, four models are executed simultaneously on the accelerator. Each model is accessed by the host software through an individual process.

Table 9 : System-under-test

Server	CPU: AMD Ryzen 9 7950X3D CPU Frequency: 16-Core Processor @ 4.20-5.70GH CPU Cache: 128 MiB (L3) Memory: 4 x 32GiB @4.8GHz
OS	Ubuntu 22.04
PCIe interface	Gen3 x16
Accelerator Card	Napatech NT200A02 SmartNIC with Xelera PCIe ULL shell
Driver	Xelera PCIe ULL 2.13.0
ML Inference Software	Xelera Silva 7.10.0

4.1 Test Description

The roundtrip latency at the API interface ($Tout_x$ – Tin_x) is measured when running the inference for a small model configuration (Table 10) and a big model configuration (Table 11).

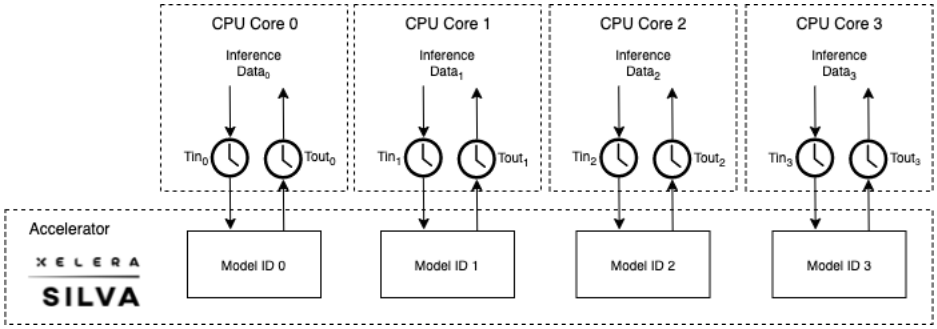

Table 10: Small model configuration

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

Table 11: Big model configuration

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

For each model configuration, the test involves executing inference simultaneously with four models (IDs from 0 to 3) in an **asynchronous** manner. This means that the processes accessing the models are independent and run on different CPU cores (0 to 3). The test is repeated 1,000,000 times.

4.2 Results Small Model

Figure 4 presents the latency statistics of Xelera Silva when executing inference simultaneously with four small models. The graphs illustrate the proportion of inference measurements (y-axis) that fall below a predetermined latency (x-axis) for each of the four concurrent model inferences.

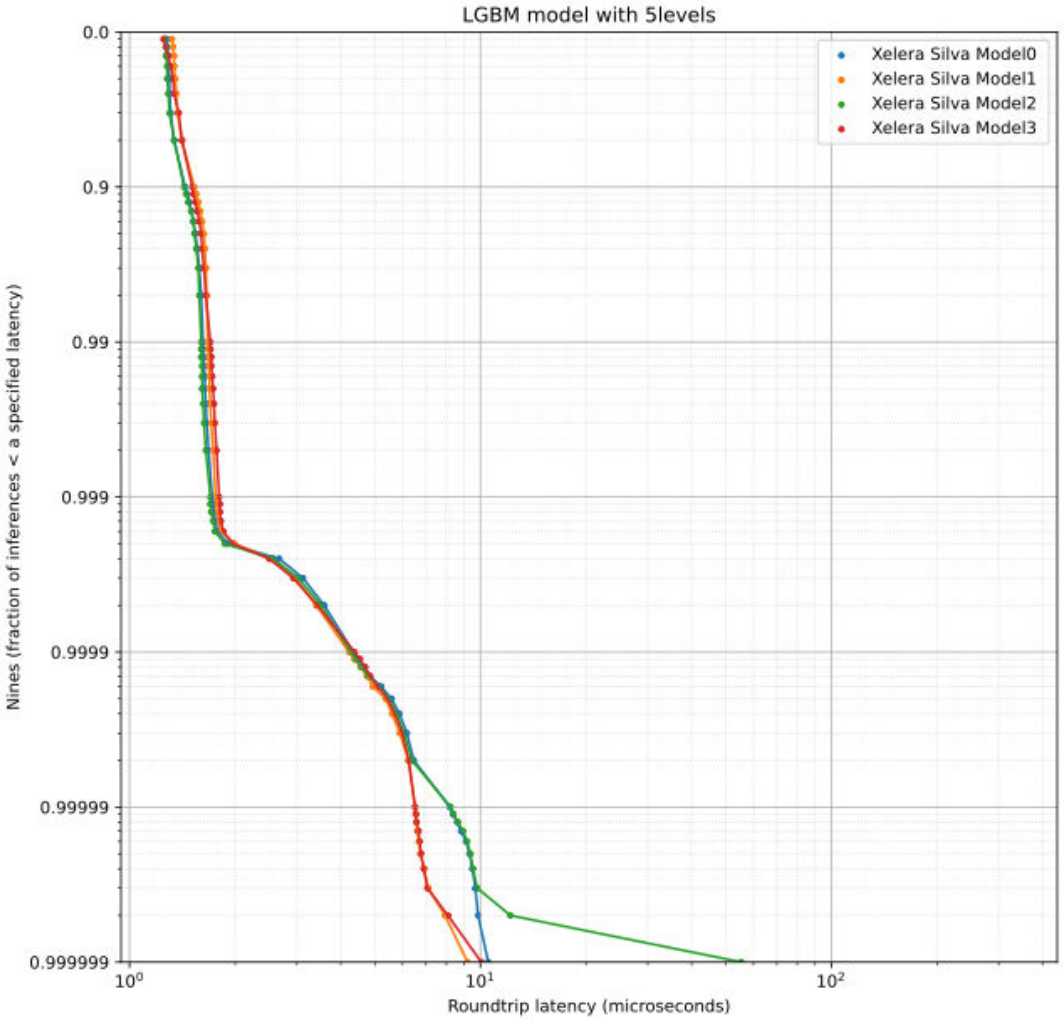

Figure 4 : Latency statistic of Xelera Silva in multi-model execution

Table 12 compares the minimum, maximum, median (50th percentile) and the 99th percentile latency (in microseconds) of the graphs above.

Table 12 : Latency statistics small model

Model ID	**Minimum latency**	**Maximum latency**	**Median latency (50th percentile)**	**99th percentile latency**
0	1.219	106.589	1.300	1.620
1	1.220	13.770	1.350	1.670
2	1.220	333.148	1.280	1.600
3	1.210	24.420	1.330	1.699

4.3 Results Big Model

Figure 5 presents the latency statistics of Xelera Silva when executing inference simultaneously with four large models. The graphs illustrate the proportion of inference measurements (y-axis) that fall below a predetermined latency (x-axis) for each of the four concurrent model inferences.

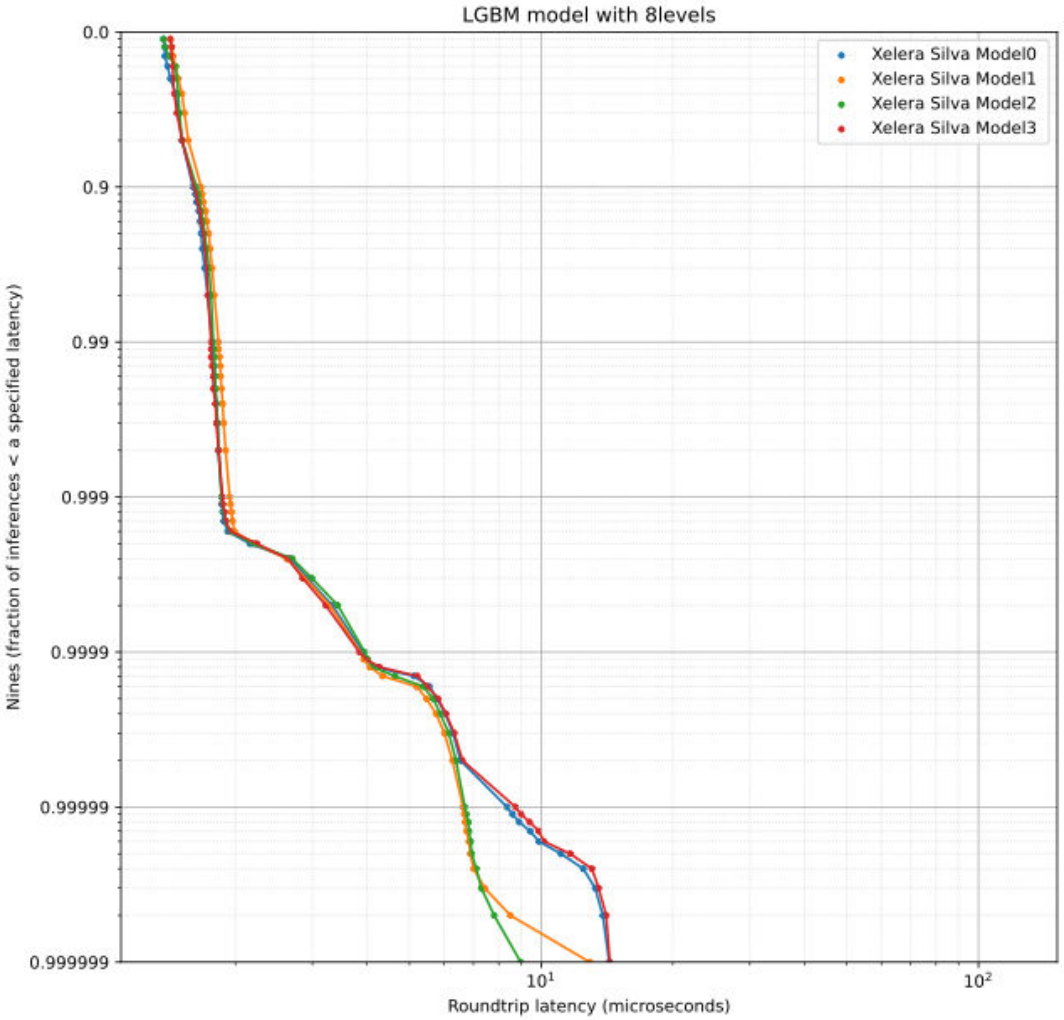

Figure 5 : Latency statistic of Xelera Silva in multi-model execution

Table 13 compares the minimum, maximum, median (50th percentile) and the 99th percentile latency (in microseconds) of the graphs above.

Table 13: Latency statistics big model

Model ID	**Minimum latency**	**Maximum latency**	**Median latency (50th percentile)**	**99th percentile latency**
0	1.320	120.969	1.420	1.770
1	1.340	34.820	1.480	1.830
2	1.320	57.069	1.470	1.780
3	1.320	81.189	1.440	1.760