July 2025

LightGBM Inference

Benchmark Report

Blackcore ACE 3100-RZ Server
AMD Alveo U50 Accelerator Card

Contents

1 Xelera Silva

Gradient Boosting Tree frameworks, such as XGBoost, LightGBM, and CatBoost, are widely used in financial trading systems, ransomware and DDOS detection systems, and recommender systems. Xelera Silva achieves best-in-class inference latency and throughput by leveraging commercial off-the-shelf data-centre grade PCIe-based accelerators.

The Xelera Silva software loads Machine Learning models in XGBoost, LightGBM, and ONNXMLTools formats and executes them for inference on AMD Alveo platforms. The user application interacts with the accelerator software via a C/C++ or Python API.

The benchmarking was conducted across the following tests:

1. **PCIe Access:** This test measures the latency of a single 32-bit register read from the accelerator card by the host system. The results in this mode are primarily influenced by the server's PCIe architecture and the specific slot configuration used.
2. **Data Transfer**: This benchmark evaluates the latency of multiple data transfer sizes to and from the accelerator card, comparing single-process and multi-process scenarios to assess the impact of concurrent access.
3. **Inference Software Comparison**: This test measures the end-to-end latency of running inference using Gradient Boosting Tree models. It provides a performance comparison between a standard CPU-based implementation and the Xelera Silva solution running on the accelerator. Benchmarks were conducted using a single model to assess baseline inference performance.
4. **Concurrent Inferences:** This test evaluates the end-to-end latency of Gradient Boosting Tree model inference under concurrent execution conditions. It compares multi-model scenarios to assess Xelera Silva impact of parallel inference workloads running on the accelerator.

These results offer insight into the performance characteristics of Xelera Silva software.

1 Test Setup

This document presents latency benchmark results for the Xelera Silva software executed on a Blackcore ACE 3100-RZ server equipped with an AMD Alveo U50 PCIe-based accelerator card. Table 1 shows the system specification.

The measured latency reflects the combined duration of data transfer between the host and accelerator, as well as the computation time on the accelerator itself.

Table 1 System under test

Server	ACE 3100-RZ CPU: AMD Ryzen 9950X CPU Frequency: 16 Cores @ 5.4GHz (SSE) CPU Cache: 64MB Memory: 4x 32GB DDR5 ECC UDIMM
OS	Linux Rocky 9.4
PCIe interface	Gen4 x8
AMD Alveo Card	U50 with Xelera PCIe ULL shell 1.2.0.0
Driver	Xelera PCIe ULL 2.13.5 or pcie-lat for PCIe Access test only
ML Inference Software	Xelera Silva 7.13.0 or pcie-lat for PCIe Access test only

2 PCIe Access

The PCIe access latency 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.

The process is pinned to CPU core 1, which has also been isolated. The test is conducted 1,000,000 times.

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.

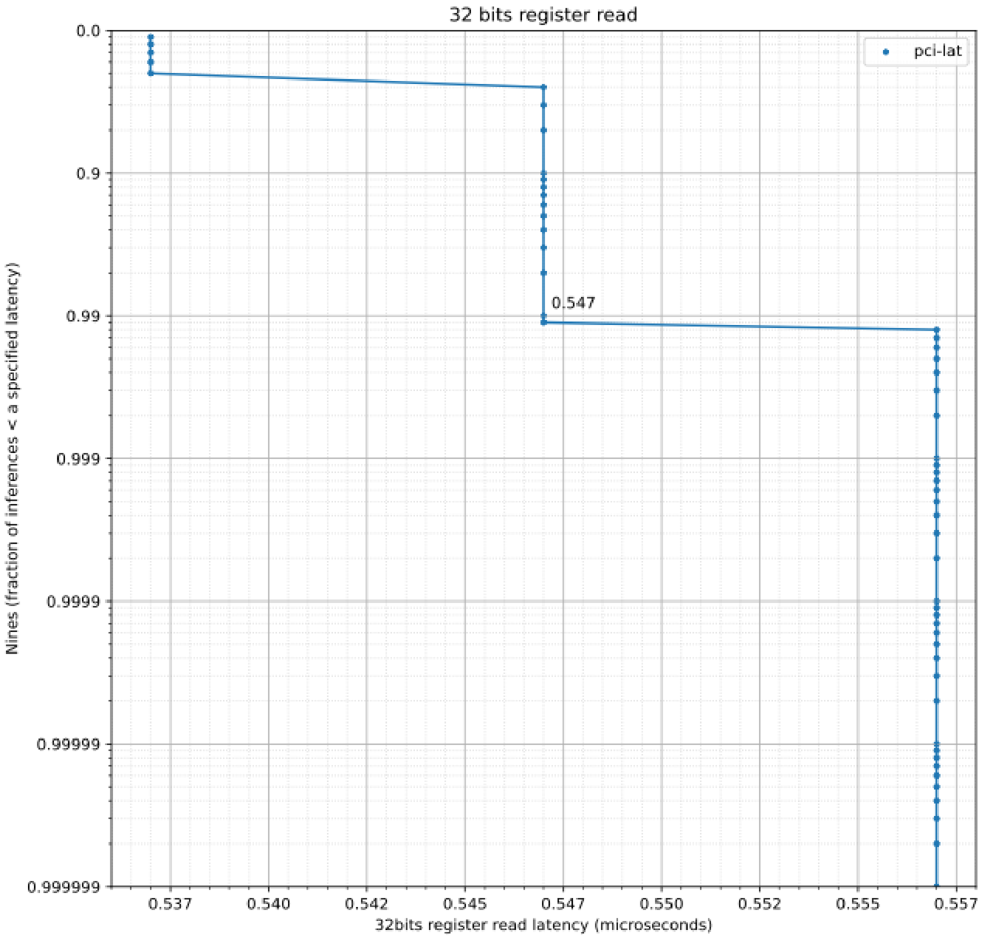

Figure 1: Latency statistic 32-bit word read form PCIe device

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 (microseconds)

Minimum	Maximum	50th percentile	99th percentile
0.527	0.557	0.537	0.547

2.2 Key Findings

The benchmark demonstrates a highly stable server configuration, with minimal variation in PCIe access latency—showing a maximum difference of only 30 nanoseconds between the minimum and maximum measurements. This test is also effective for identifying the PCIe slot with the lowest latency, which is typically the one with lanes directly connected to the CPU.

3 Data Transfer

This benchmark evaluates the latency of data transfers of various sizes between the host system and the accelerator card.

Each data packet is transmitted to the accelerator, internally looped back, and returned to the host, with roundtrip latency measured at the API interface (Tout – Tin) to capture the full transfer cycle. Packet sizes tested include 16, 32, 64, 128, 256, 512, and 1024 bytes.

Two test configurations are employed (Figure 2): the first involves single-core execution to assess latency across multiple packet sizes; the second performs four concurrent, asynchronous transfers using 1024-byte packets.

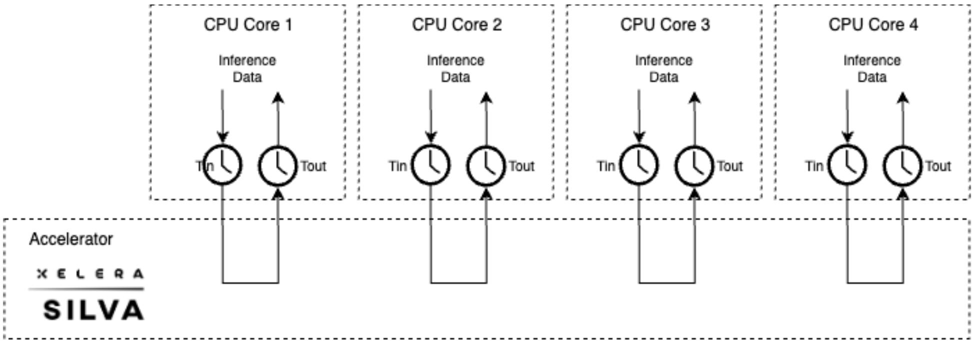

Figure 2 Data transfer test configuration

To ensure consistent and reliable measurements, each process is pinned to an isolated CPU core (cores 1 through 4). All tests are repeated 1,000,000 times to ensure statistical robustness.

3.1 Results Single Transfer

Figure 3 shows the latency statistics of single data transfer for multiple packet sizes using only CPU core 1. The graphs show the fraction of inference measurements (y-axis) below a specified latency (x-axis).

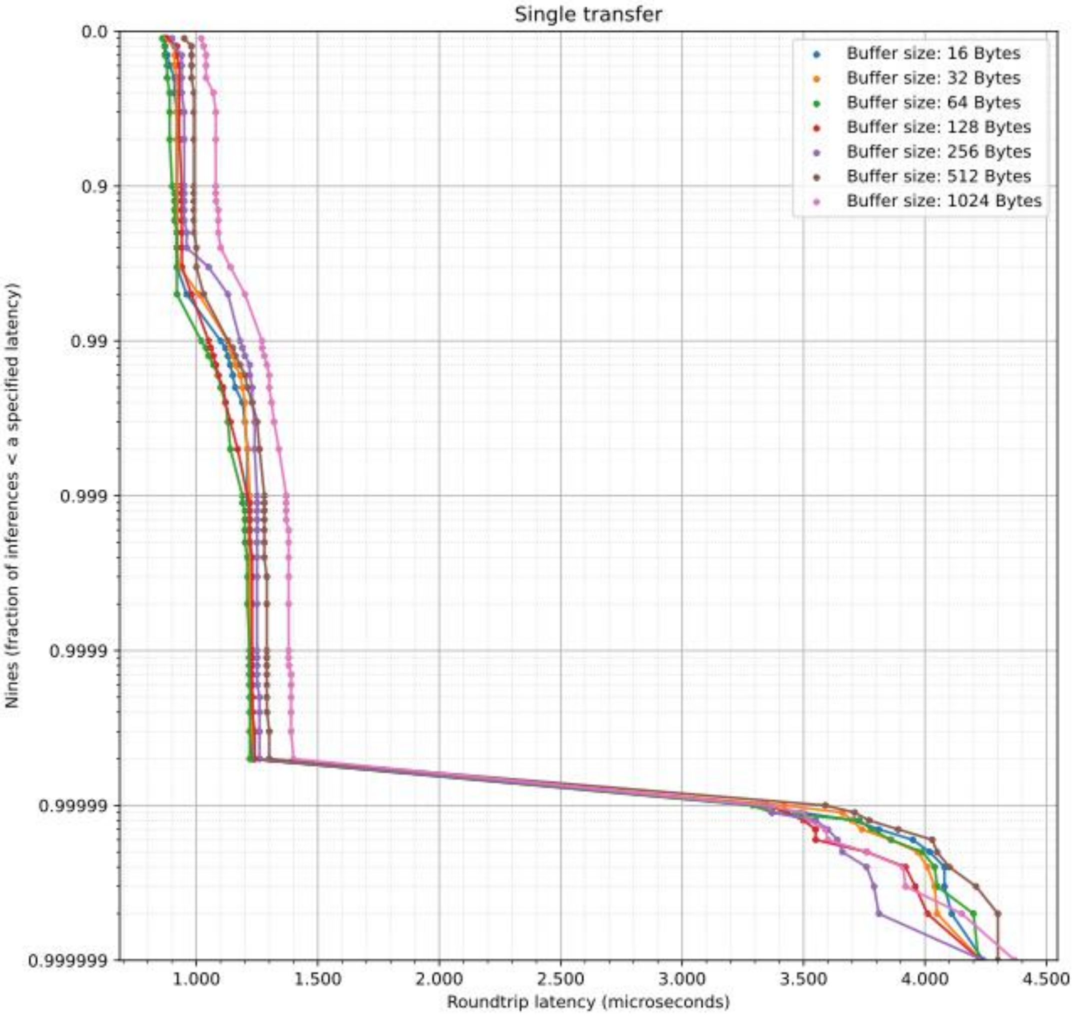

Figure 3 Latency statistic single transfer

Table 3 compares the minimum, maximum, median (50th percentile) and the 99th percentile latency (microseconds) of the graphs above.

Table 3 Latency statistics single transfer (microseconds)

Data Packet Size (Bytes)	Minimum	Maximum	50th percentile	99th percentile
16	0.830	4.240	0.910	1.100
32	0.830	4.230	0.920	1.130
64	0.830	4.220	0.880	1.020
128	0.850	4.230	0.930	1.050
256	0.860	4.240	0.940	1.180
512	0.910	4.300	0.980	1.130
1024	1.000	4.370	1.040	1.270

3.2 Results Parallel Transfers

Figure 4 shows the latency statistics of four concurrent, asynchronous data transfers using 1024-byte packets on CPU cores 1-4. The graphs show the fraction of inference measurements (y-axis) below a specified latency (x-axis).

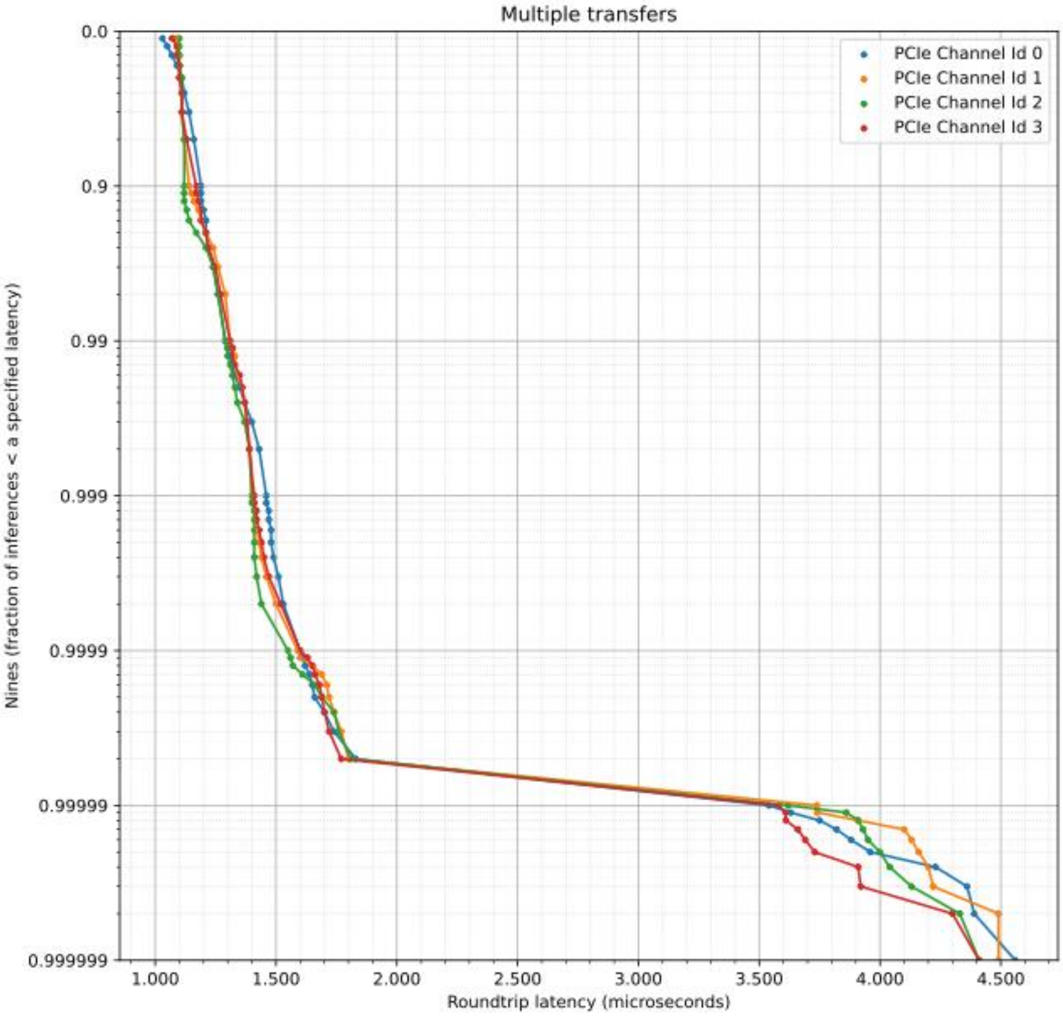

Figure 4 Latency statistic multiple transfers

Table 4 compares the minimum, maximum, median (50th percentile) and the 99th percentile latency (microseconds) of the graphs above.

Table 4 Latency Statistics Parallel Transfers (microseconds)

Data Transfer ID	Minimum	Maximum	50th percentile	99th percentile
0	1.000	4.560	1.110	1.290
1	1.000	4.490	1.100	1.310
2	1.010	4.410	1.110	1.290
3	0.990	4.410	1.100	1.310

3.1 Key Findings

The data transfer benchmarks demonstrate that **Xelera Silva maintains consistently low latency** across a wide range of packet sizes and concurrent access scenarios.

For single-process transfers, median latency remained under **1.05 microseconds** for all tested packet sizes up to 1024 bytes, with the 99th percentile never exceeding **1.27 microseconds**.

In the concurrent test—where four 1024-byte transfers were executed asynchronously on isolated CPU cores—the **median latency ranged from 1.10 to 1.11 microseconds**, and the **99th percentile remained below 1.32 microseconds** across all processes.

These results confirm that the system handles both **single and multi-process data transfers with minimal variability**, making it highly suitable for real-time applications that demand **predictable, low-latency communication between the host and accelerator**.

4 Inference Software Comparison

The Xelera Silva software was compared against other software frameworks for the acceleration of Gradient Boosting Tree Machine Learning models. The compared software frameworks are listed in Table 5 below.

Table 5: Compared software frameworks

ML Inference Software	Version	Description
Intel oneDAL	2024.5.0	Intel CPU-optimized ML inference software
Xelera Silva	7.13.0	FPGA-accelerated ML inference software

Xelera Silva is the only FPGA-accelerated ML inference software in this comparison. Intel oneDAL framework uses only CPU optimizations to accelerate the inference of gradient boosting models, such as the use of vector extension instructions, branch prediction and integer comparisons.

The roundtrip latency at the API interface (Tout – Tin) is measured when running the inference for a small model configuration (Table 6) and a big model configuration (Table 7).

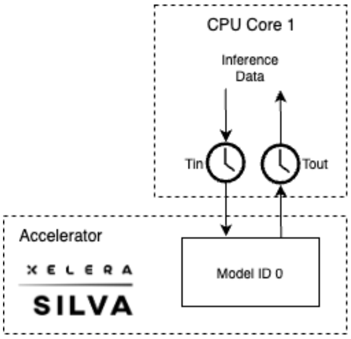

Table 6: Small model configuration

Model Type	LightGBM regression
Dataset	Synthetic Random
Number of Features	64
Number of Trees	200
Number of Levels	5
Batch Size	1
Numerical Features	Yes
Categorical Features	No

Table 7: 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 the two models. The process is pinned to CPU core 1, which has also been isolated. The test is conducted 1,000,000 times.

4.1 Results Small Model

Figure 5 shows the latency statistics of Xelera Silva in comparison to the third-party software frameworks when running the small model (Table 6). The graphs show the fraction of inference measurements (y-axis) below a specified latency (x-axis).

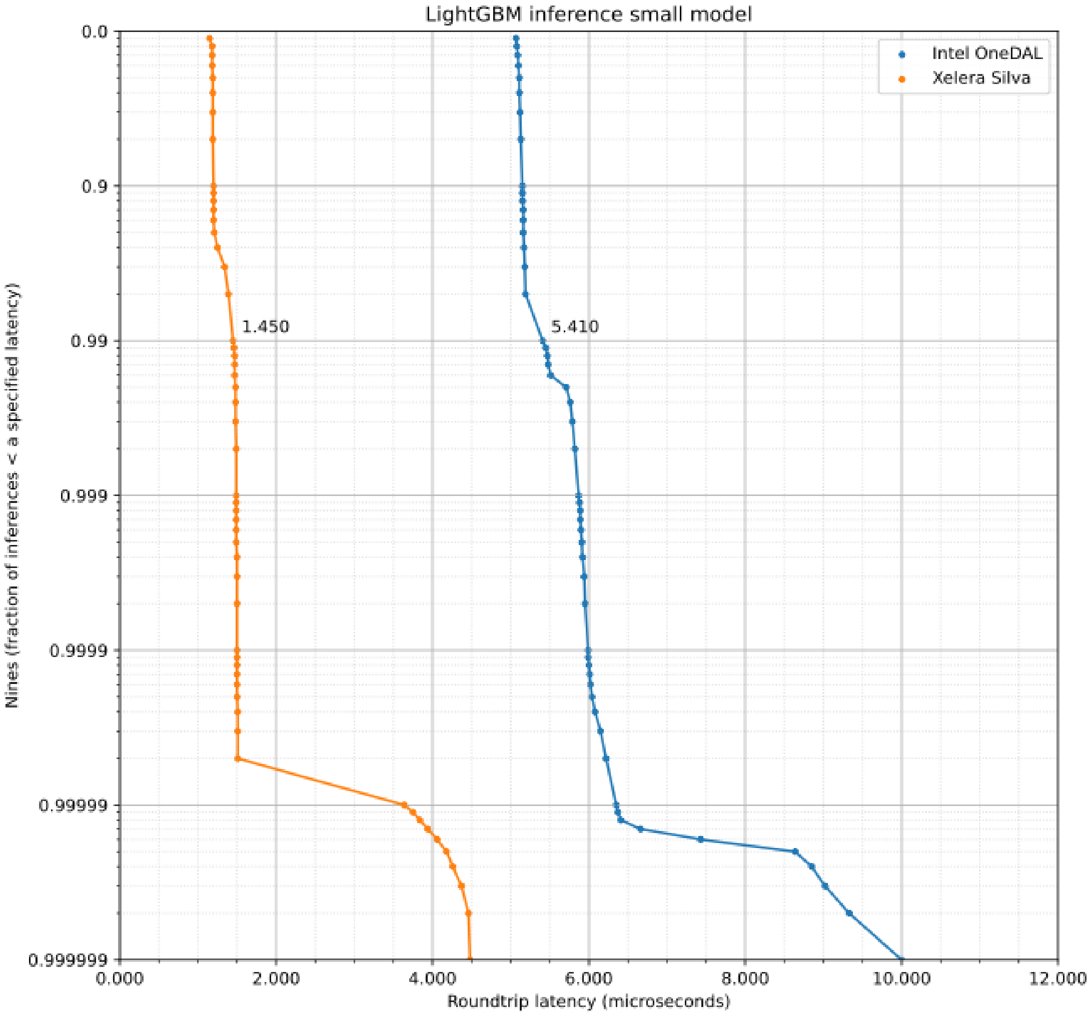

Figure 5 Latency statistic small model

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

Table 8: Latency statistics small model (microseconds)

ML Inference Software	Minimum	Maximum	50th percentile	99th percentile
Intel oneDAL	4.960	19.781	5.110	5.410
Xelera Silva	1.110	4.550	1.190	1.450

4.2 Results Big Model

Figure 6 shows the latency statistics of Xelera Silva in comparison to the third-party software frameworks when running the small model (Table 7). The graphs show the fraction of inference measurements (y-axis) below a specified latency (x-axis).

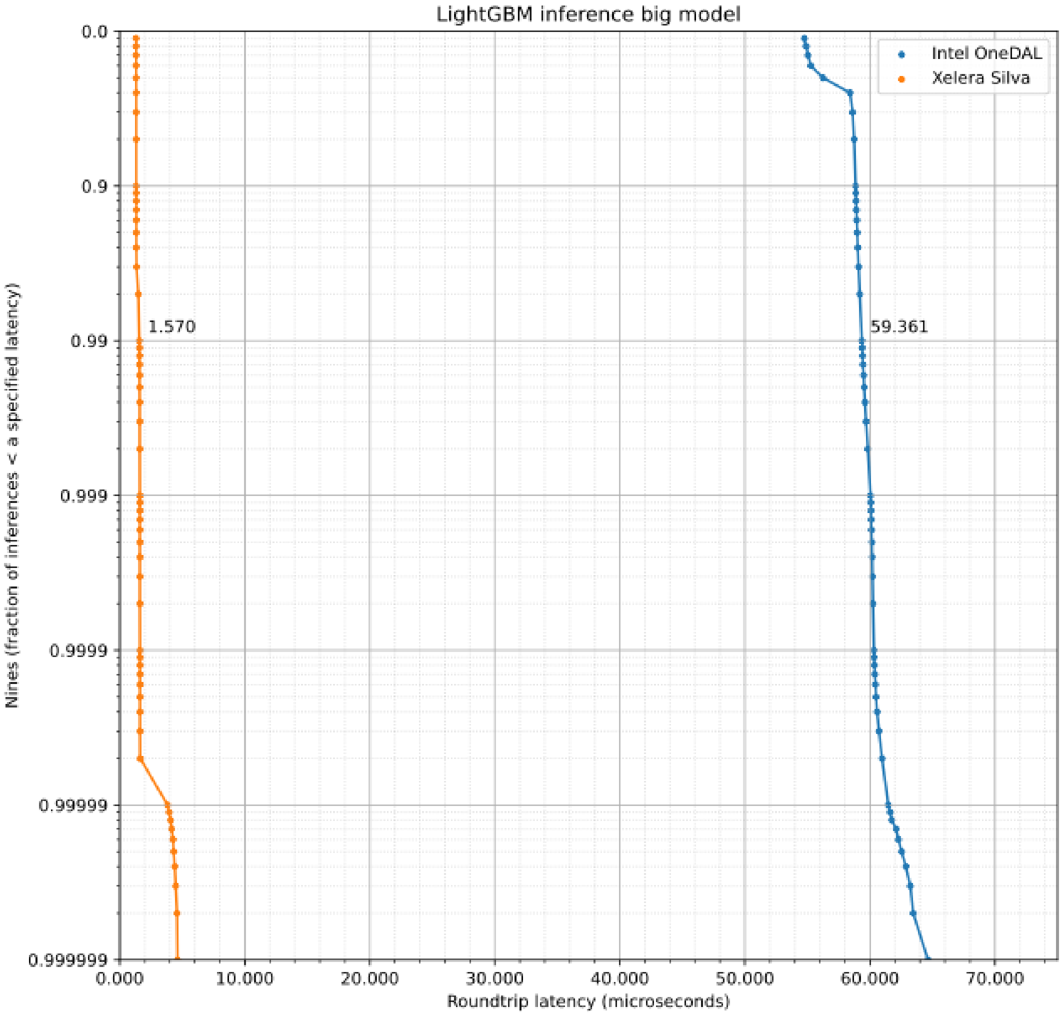

Figure 6 Latency statistics big model

Table 9 compares the minimum, maximum, median (50th percentile) and the 99th percentile latency (microseconds) of the graphs above.

Table 9: Latency statistics big model (microseconds)

ML Inference Software	Minimum	Maximum	50th percentile	99th percentile
Intel oneDAL	50.291	199.412	56.250	59.361
Xelera Silva	1.230	4.670	1.310	1.570

4.3 Key Findings

The benchmark results demonstrate that Xelera Silva delivers a significant performance advantage over Intel oneDAL across both small and large LightGBM model configurations.
In particular, Xelera Silva achieves up to **44 times lower median inference latency** on large models, with just **1.31 microseconds** compared to **56.25 microseconds** for Intel oneDAL. Moreover, while Intel oneDAL shows a **maximum latency of nearly 200 microseconds**, this upper bound can be **prohibitive in high-frequency applications** such as trading or real-time detection systems, where strict latency ceilings are critical. In contrast, Xelera Silva not only delivers faster inference but also maintains **very stable latency characteristics**, with minimal variance even at the 99th percentile—making it highly suitable for deterministic, ultra-low-latency environments.

5 Concurrent Inferences

In this benchmark, 4 models are executed simultaneously on the FPGA accelerator. Each model is accessed by the host software via an individual process. The processes are pinned to CPU core 1, 2, 3, 4 respectively. These cores have also been isolated. The test is conducted 1,000,000 times.

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).

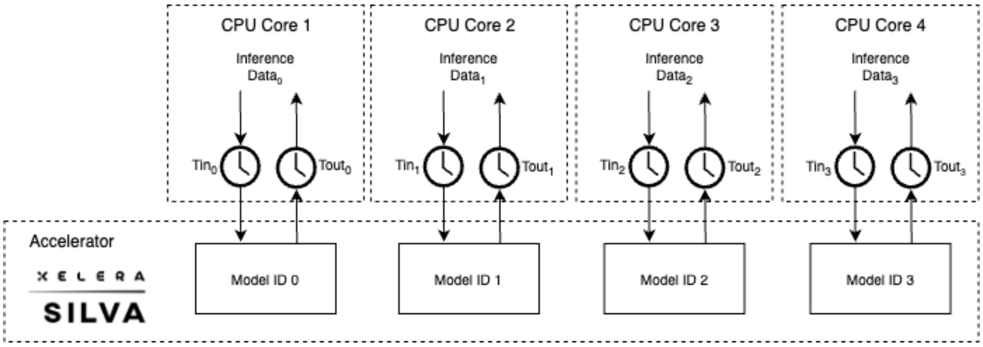

Table 10: Small model configuration

Model Type	LightGBM regression
Dataset	Synthetic Random
Number of Features	64
Number of Trees	200
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 running inference with four models (IDs from 0 to 3) simultaneously in an **asynchronous** mode (independent processes accessing the models). Each process is pinned to a CPU core (0 to 3). The test is conducted 1,000,000 times.

5.1 Results Small Model

Figure 7Figure 7: Latency statistic multi-model shows the latency statistics of Xelera Silva when running inference with 4 small models at the same time. The graphs show the fraction of inference measurements (y-axis) below a specified latency (x-axis) for each of the 4 concurrent model inferences.

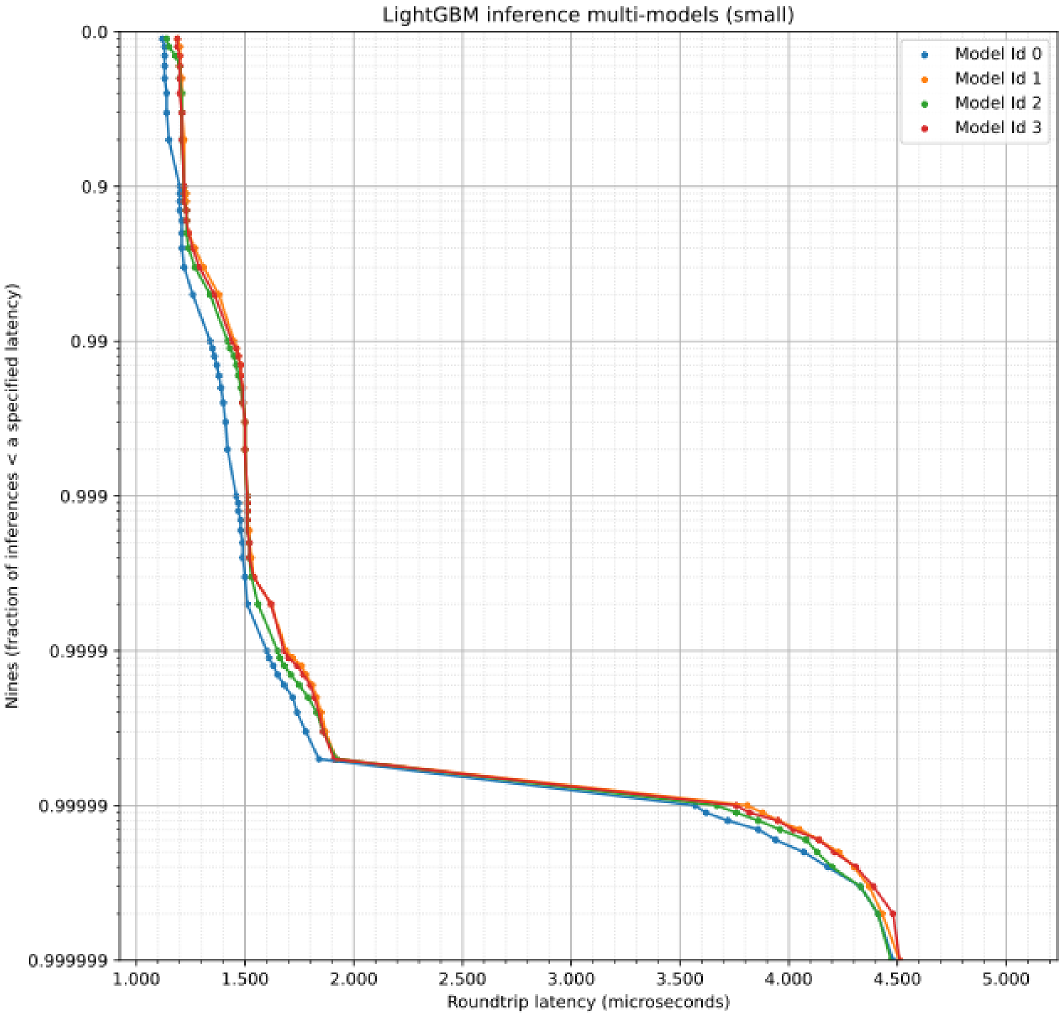

Figure 7: Latency statistic multi-model (small)

Table 12 Table 12 : Latency statistics small modelcompares the minimum, maximum, median (50th percentile) and the 99th percentile latency (microseconds) of the graphs above.

Table 12 : Latency statistics small model (microseconds)

Model ID	Minimum	Maximum	50th percentile	99th percentile
0	1.080	5.040	1.130	1.340
1	1.100	4.570	1.210	1.450
2	1.100	4.580	1.200	1.420
3	1.090	4.650	1.200	1.440

5.2 Results Big Model

Figure 8 shows the latency statistics of Xelera Silva when running inference with 4 big models at the same time. The graphs show the fraction of inference measurements (y-axis) below a specified latency (x-axis) for each of the 4 concurrent model inferences.

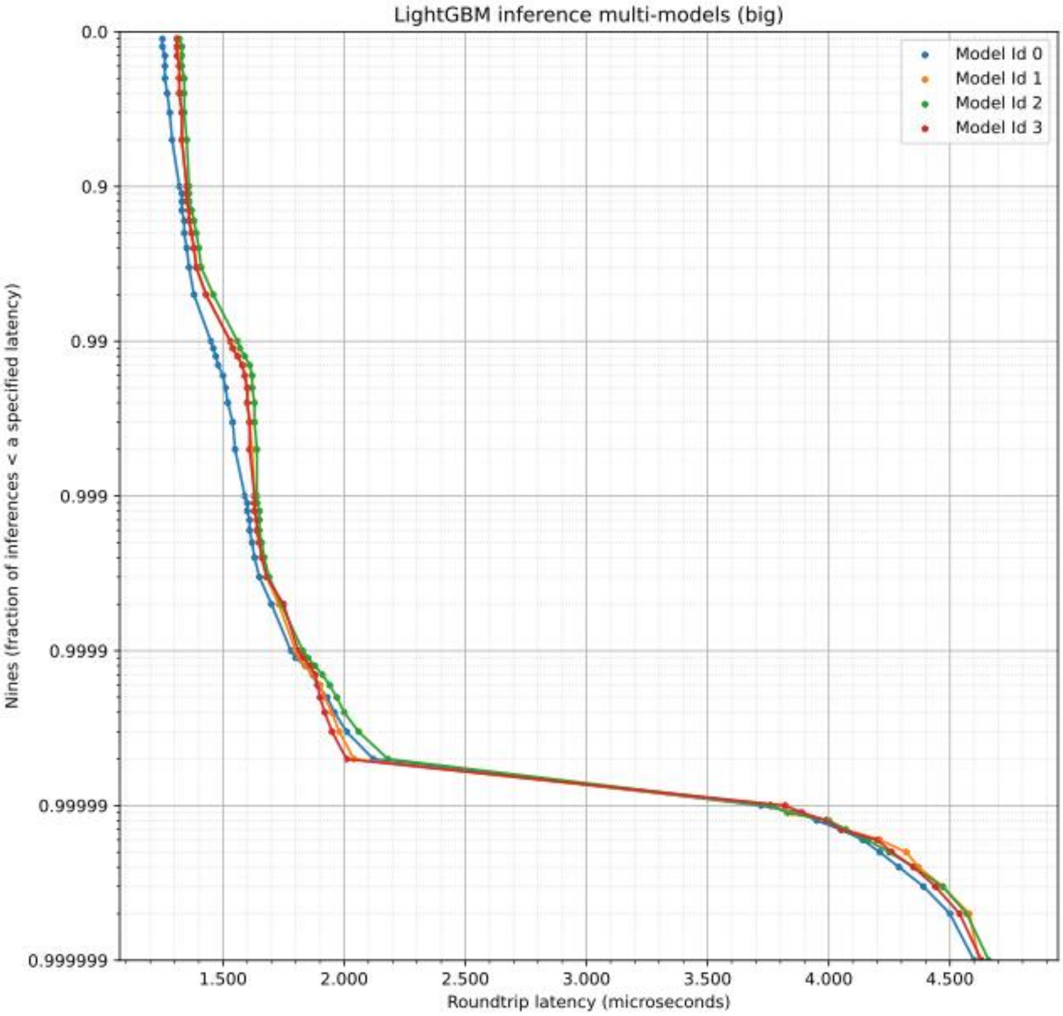

Figure 8: Latency statistic multi-model (big)

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

Table 13 : Latency statistics big model (microseconds)

Model ID	Minimum	Maximum	50th percentile	99th percentile
0	1.210	4.770	1.260	1.450
1	1.220	4.700	1.320	1.530
2	1.240	4.760	1.340	1.560
3	1.220	4.770	1.320	1.530

5.1 Key Findings

The concurrent inference benchmarks show that **Xelera Silva maintains ultra-low and highly consistent latency** even when running four models simultaneously in an asynchronous, multi-process setup. For both small and large model configurations, the **median latency across all models remains below 1.35 microseconds**, and the **99th percentile stays under 1.6 microseconds**, demonstrating minimal performance degradation under parallel workloads. Notably, when compared to **single model execution**, where the 99th percentile latency was **1.450 µs for the small model** and **1.570 µs for the big model**, the concurrent inference results remain remarkably close: **1.340–1.450 µs** for small models and **1.450–1.560 µs** for big models. This shows that even under **fully parallel and asynchronous execution**, Xelera Silva maintains **virtually the same tail latency** as in isolated scenarios, highlighting its **excellent scalability and resource efficiency**. Additionally, the **maximum observed latencies remain well below 5 microseconds**, reinforcing the platform's ability to deliver **scalable, deterministic performance**. These results confirm Xelera Silva's suitability for **real-time, high-frequency applications** where predictable latency across concurrent inference tasks is critical.

6 Summary and Conclusions

This benchmark report evaluates the performance of the Xelera Silva machine learning inference software running on the Blackcore ACE 3100-RZ server equipped with an AMD Alveo U50 accelerator card.
The evaluation encompasses a comprehensive set of tests, including PCIe access latency, data transfer performance, inference latency comparison, and concurrent inference scalability.

The results demonstrate that the combined Xelera Silva and Alveo U50 platform delivers exceptional performance across all test categories:

- **PCIe Access Latency**: Ultra-stable latencies with minimal jitter (only 30 nanoseconds between min and max) confirm a highly optimized PCIe topology, critical for consistent low-latency operation.
- **Data Transfer Performance**: Single and multi-process data transfers maintained sub-1.3 microsecond 99th percentile latency across all packet sizes tested, demonstrating efficient and predictable host-to-accelerator communication.
- **Inference Acceleration**: Xelera Silva outperformed Intel oneDAL by up to 44x in median inference latency. For large LightGBM models, median latency was reduced from 56.25 µs (CPU) to just 1.31 µs (FPGA), while significantly narrowing the variance (99th percentile remained under 1.6 µs).
- **Concurrent Inference Scalability**: Even under four fully parallel inference streams, latency remained consistently low and tightly bounded, with virtually no degradation compared to single-model execution. This highlights the solution’s excellent scalability and suitability for high-throughput, real-time workloads.

Xelera Silva is well-suited for real-time systems with stringent latency and consistency requirements, including financial trading, cyber defense, and high-frequency industrial applications. Its ability to scale to multiple parallel inferences without impacting tail latency further underscores its architectural efficiency and production-readiness for mission-critical deployments.