Journal of Embedded Systems & Its Applications
Volume 2, Issue 2, July-December 2017

Embedded System Architecture for High-Performance Edge Computing

Amit Bansode
Department of Electronics & Computer Engineering,
Pune Institute of Computer Technology, Pune, Maharashtra, India
***Email**: amit.bansode@pict.edu*

Neha Kulkarni
Department of Electronics Engineering,
Walchand College of Engineering, Sangli, Maharashtra, India
***Email**: neha.kulkarni@walchandsangli.ac.in*

Kunal Jadhav
Centre for Embedded & Edge Computing Research,
MIT World Peace University, Pune, Maharashtra, India
***Email**: kunal.jadhav@mitwpu.edu.in*

Abstract

The exponential growth of data generated by Internet of Things (IoT) devices has exposed the limitations of cloud-centric computing models, particularly in terms of latency, bandwidth consumption, privacy, and reliability. Edge computing has emerged as a promising paradigm that brings computation closer to data sources, enabling real-time analytics and intelligent decision-making. Embedded systems form the core of edge nodes, providing computing, storage, and communication capabilities under strict power and resource constraints. This paper presents a detailed study of embedded system architectures designed for high-performance edge computing. It discusses hardware accelerators, heterogeneous processing platforms, memory hierarchies, and software frameworks that enable low-latency and high-throughput edge applications. A reference architecture is proposed and evaluated using a real-time video analytics use case. Experimental analysis demonstrates that optimized embedded edge architectures significantly reduce latency and network load compared to cloud-only approaches.

Keywords: *Edge computing, Embedded systems, Heterogeneous architecture, Hardware accelerators, IoT*

1. INTRODUCTION

The rapid expansion of IoT, cyber-physical systems, and intelligent applications has resulted in massive volumes of data generated at the network edge. Traditional cloud computing architectures require data to be transmitted to centralized servers for processing, leading to increased latency, bandwidth congestion, and dependency on network availability. These limitations are unacceptable for time-sensitive applications such as autonomous vehicles, industrial automation, smart surveillance, and healthcare monitoring.

Edge computing addresses these challenges by enabling data processing and analytics closer to the source of data generation. Embedded systems play a crucial role in edge computing, acting as edge nodes that perform real-time computation, data filtering, and local decision-making. However, designing embedded architectures that deliver high performance while maintaining energy efficiency and cost-effectiveness remains a significant challenge.

This paper focuses on embedded system architectures optimized for high-performance edge computing. The objectives are to analyze architectural requirements, explore hardware and software optimization techniques, propose a scalable reference architecture, and evaluate its performance through a representative application.

2. REQUIREMENTS OF HIGH-PERFORMANCE EDGE COMPUTING

Edge computing applications impose stringent requirements on embedded system architectures.

2.1 Low Latency

Real-time decision-making demands minimal processing and communication delays.

2.2 High Throughput

Edge nodes must process large volumes of data streams such as video, audio, and sensor data.

2.3 Energy Efficiency

Despite increased performance demands, edge devices often operate under power constraints.

2.4 Scalability and Reliability

Architectures should support scalability and tolerate failures without disrupting system operation.

3. EMBEDDED SYSTEM ARCHITECTURE FOR EDGE COMPUTING

Modern embedded edge systems employ heterogeneous architectures combining multiple processing elements.

3.1 Heterogeneous Processing Platforms

Typical edge nodes integrate CPUs, GPUs, DSPs, and AI accelerators to handle diverse workloads efficiently.

3.2 Memory Hierarchy

Efficient memory architecture, including cache hierarchies and shared memory, is critical for reducing data access latency.

3.3 Hardware Accelerators

Dedicated accelerators for machine learning, encryption, and video processing significantly enhance performance per watt.

Table 1: compares common processing elements used in edge embedded systems.

Processing Unit	Strengths	Typical Applications
CPU	General-purpose control	System management
GPU	Parallel processing	Video analytics
DSP	Signal processing	Audio and sensor data
AI Accelerator	Neural networks	Object detection

4. SOFTWARE STACK FOR EMBEDDED EDGE SYSTEMS

A robust software stack is essential for managing complex edge workloads.

4.1 Operating Systems and Middleware

Embedded Linux and real-time extensions are commonly used for high-performance edge nodes. Middleware platforms enable device management and data orchestration.

4.2 Containerization and Virtualization

Lightweight containers allow efficient deployment and isolation of edge applications.

4.3 Edge AI Frameworks

Frameworks such as TensorFlow Lite and ONNX Runtime enable deployment of machine learning models on embedded devices.

5. PROPOSED REFERENCE ARCHITECTURE

A scalable embedded architecture for high-performance edge computing is proposed.

5.1 Architecture Overview

The system integrates a multi-core ARM processor, GPU, AI accelerator, high-speed memory, and network interfaces.

```
+-----------------------------+
| Edge Applications           |
| (AI, Analytics, Control)    |
+-------------+---------------+
              |
+-------------v---------------+
| Middleware & Containers     |
+-------------+---------------+
              |
+-------------v---------------+
| OS & Resource Manager       |
+------+------+------+--------+
       |      |      |
+------v+ +---v---+ +v--------+
| CPU   | | GPU   | | AI Acc  |
+--------+ +-------+ +---------+
              |
+-------------v---------------+
| Memory & Network Interfaces |
+-----------------------------+
```

Figure 1: shows the proposed embedded edge computing architecture.

6. CASE STUDY: REAL-TIME VIDEO ANALYTICS AT THE EDGE

To evaluate the proposed architecture, a real-time video analytics application was implemented.

6.1 Application Description

The system performs object detection on video streams captured by edge cameras.

6.2 Experimental Setup

- ARM-based embedded edge platform
- AI accelerator for inference
- Video input: 1080p at 30 fps

6.3 Performance Evaluation

Latency, throughput, and network usage were measured.

Table 2: presents comparative results.

Processing Model	Average Latency (ms)	Network Bandwidth Usage
Cloud-Based	180	High
Edge-Based	45	Low

7. DISCUSSION

The results demonstrate that embedded edge architectures significantly improve application responsiveness and reduce network dependency. Heterogeneous processing and hardware acceleration are key enablers of high-performance edge computing.

8. CHALLENGES AND FUTURE TRENDS

Despite its advantages, edge computing faces challenges such as programming complexity, security risks, and resource management. Future trends include federated learning, AI-driven resource orchestration, and tighter cloud–edge integration.

9. CONCLUSION

This paper presented a comprehensive analysis of embedded system architectures for high-performance edge computing. By leveraging heterogeneous processing platforms, efficient memory hierarchies, and optimized software stacks, embedded edge systems can deliver low-latency and high-throughput performance. The proposed architecture and case study validate the effectiveness of edge-based embedded solutions for next-generation intelligent applications.

REFERENCES

1. Shi, W., et al., "Edge Computing: Vision and Challenges," *IEEE Internet of Things Journal*.
2. Satyanarayanan, M., "The Emergence of Edge Computing," *Computer*, IEEE.
3. NVIDIA Jetson Platform Documentation.
4. TensorFlow Lite Developer Guide.
5. OpenFog Consortium Architecture Framework.

