logic
technology

DATA MANAGEMENT

in AI driven edge applications

Logic Technology
2024

1. Introduction

Explaining Edge AI

Understanding Edge AI

To understand what Edge AI is, we need to look at the technological trends that create the need to move AI to the Edge. Edge AI is the combination of Edge Computing and Artificial Intelligence to perform machine learning tasks directly on Edge devices.

The fusion of AI and edge computing is natural, since there is a clear intersection between them. Data generated at the network edge depends on AI to fully unlock its full potential. And edge computing is able to prosper with richer data and application scenarios. Edge intelligence is expected to push deep learning computations from the cloud to the edge as much as possible. This enables the development of various distributed, low-latency, and reliable, intelligent services. Today, in the era of IoT, a gigantic amount of data generated by countless connected devices needs to be collected and analyzed. This leads to large amounts of data being generated in real time, which at the same time requires AI systems to give meaning to this data.

The benefits of Edge AI

- Reduced latency

AI Deep Learning is deployed close to the requesting users. This significantly reduces the latency and cost of sending data to the cloud for processing. This is important for applications where real-time response is critical, such as autonomous vehicles and industrial automation.

- Improved security

Edge AI can help to improve the security of AI applications by keeping data local. This can help to protect sensitive data from being intercepted or hacked.

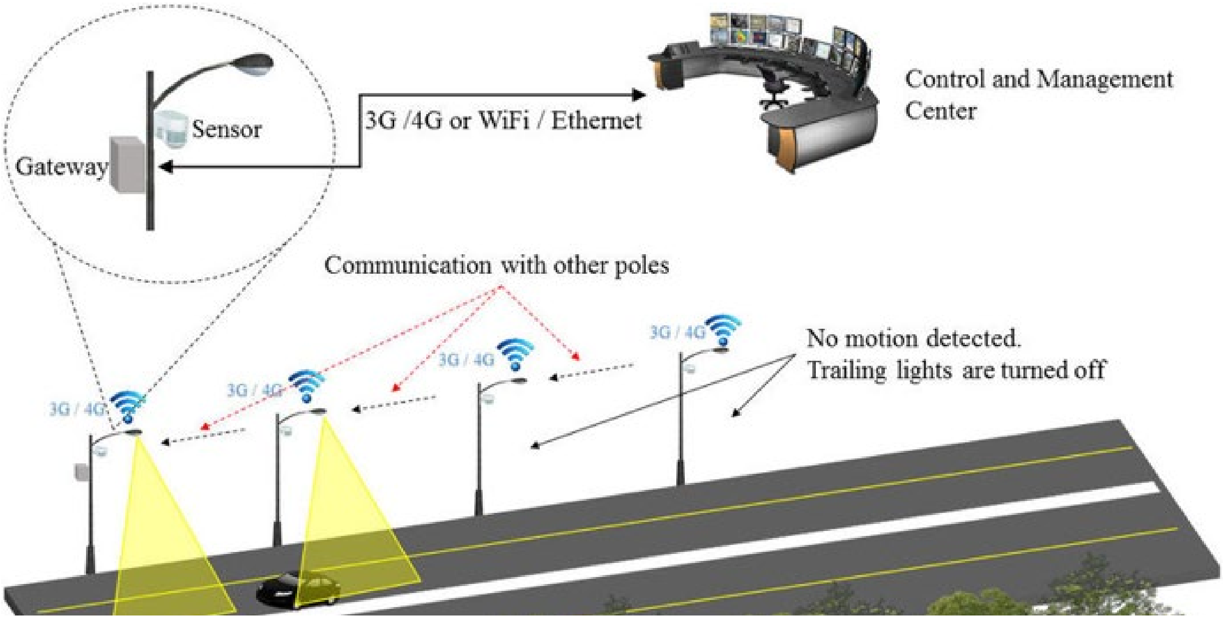
- Lower data transfer volume

One of the main benefits of edge AI is that data is processed by the device, and only a significantly lower amount of processed data is sent to the cloud. By reducing the traffic amount across the connection between a small cell and the core network, the bandwidth of the connection can be increased to prevent bottlenecks, and the traffic amount in the core network is reduced.

1. Introduction

Traditional Edge vs. Edge AI

Traditional Edge Devices
The illustration below shows a well-known example of edge computing in the context of street lighting. The sensors communicate with the gateway in the vicinity of the lamppost. The data is then sent from the gateway to a central control point where the actual analysis and decision-making takes place. This is then communicated back to the gateway and lampposts in the event of possible adjustments that need to be made.


Sensor
Gateway
3G /4G or WiFi / Ethernet
Control and Management Center
Communication with other poles
3G / 4G
3G / 4G
3G / 4G
3G / 4G
No motion detected.
Trailing lights are turned off

Edge AI
Pictured below is the application called ET city brain. This is a software system that uses artificial intelligence (AI) and data collection to better manage cities. The system collects data from various sources, such as cameras, sensors, and mobile apps, and uses AI to analyze this data and make real-time decisions. For example, ET City Brain can adjust traffic lights to improve traffic flow, optimize bus routes, and identify incidents faster.

2. Designing for Edge AI

1. Fragmentation

Fragmentation

Fragmentation in flash memory occurs when data is not stored in contiguous memory cells. Instead, it is spread over different non-contiguous memory areas. This can happen when data is written, erased, and rewritten on flash memory.

Recent research shows that as flash memory becomes faster, the more likely the software I/O stack is to create overhead, causing performance bottlenecks.

Fragmentation is currently seen as one of the main causes of poor flash memory performance. Another interesting finding is that the location of the I/Os plays a very important role. Both write and read latency increases as I/Os are more spread out in flash memory.

In applications where data is intensively read, written, and rewritten (such as cameras for autonomous driving), fragmentation can lead to critical system errors.

Solving Fragmentation Issues

Fragmentation is a serious problem that can lead to poor flash memory performance and critical system errors. Preventing fragmentation completely is impossible, but there are ways to avoid it as much as possible.

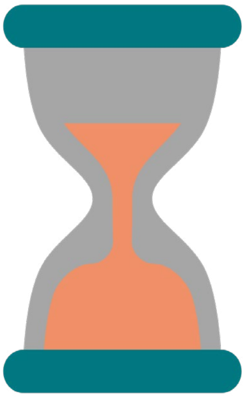
Dynamic transaction point settings

However, there are ways to avoid fragmentation, such as using a file system with dynamic transaction point settings. The only file system on the market that uses dynamic transaction point settings is Reliance Nitro by Tuxera.

Dynamic transaction point settings allow you to dynamically change transaction points during runtime. It groups related write operations into 1 atomic write operation. In short, the transaction points mark the beginning and end of a series of related write operations. This reduces the chance of having different fragmented data blocks. You can compare it to a delivery person who performs 4 different scans for 4 packages that need to go to the same location, while he could also perform 1 scan to group all 4 packages.

Wear leveling

Wear leveling is directly related to fragmentation because wear leveling is one of the solutions to the fragmentation issue. In a nutshell, wear leveling is a technique for extending the life of flash memory by evenly distributing wear across the memory cells.



However not all wear leveling algorithms are the same. We recommend using static wear leveling because it makes optimal use of all memory cells and it’s available P/E cycles. Dynamic wear-leveling only involves free blocks into the wear leveling algorithm(with possible high P/E cycles), whereas static wear-leveling also involves static blocks into the game with very low P/E cycles where data has been stored for a long while. This method ensures a much longer flash lifetime as all blocks are considered in the wear leveling algorithm.

2. Designing for Edge AI

2. Flash Memory Management

Flash memory management in Edge applications

Flash memory is an important aspect to consider when you're developing an edge application with AI. Adding AI to your Edge application increases the chance that significantly more data will be written to the flash, or at least that more data will be read from the flash.

Write amplification

One of the consequences this can have is write amplification.This occurs when the amount of information written to the flash is a multiple of the logical amount that is intended to be written. The solution we recommend for this is to use a transactional file system.

In transactional file systems, data is stored in blocks. With traditional file systems, when the data to be written is larger than the data block, there is a spillover effect to the next block. What happens, when writing to flash memory, is that the next write operation searches for the next free data block. In the meantime, you have a data block that is only half used.

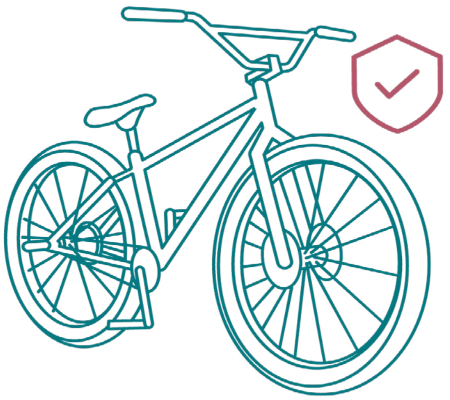
The solution to this is to completely fill the next block during a spillover, instead of leaving it half empty. Ideally write operations are matched to the size of the available data block whenever possible.

Secure erase

Secure erase is a process used to permanently erase data from a storage device with the aim of ensuring that the erased data can no longer be recovered, even with special recovery software or other techniques.

The way to minimize the impact of secure erase on flash wear depends on good interaction between the file system and the flash driver. With the right file system, the user's critical data can be marked "secure", and the file system can then notify the flash driver to use the proper secure operations while erasing any copies of that data.

This means immediate secure erasure of second copies of data blocks, and similar secure erasure of the blocks when the file itself is removed. Reliance Nitro has all of these secure options built in, and will flag the media driver to use those secure commands when necessary, and only when necessary.



Use Case: e-bike OEM

The largest e-bike system OEM chose our software specifically for the secure erase aspect.

In order to comply with certain regulations, they needed to be assured that all deleted data in their NAND was actually erased and no longer accessible or recoverable.

2. Designing for Edge AI

3. Database Management

When Edge AI applications process large amounts of data such as sensor data or image information, this also means that real-time decisions have to be made at the database level.

If performance is a top priority, then you should consider a hybrid or in-memory database. These are real-time. But if it has to be hard real time, then there is only one DBMS and that is eXtremedb/rt from McObject

What distinguishes eXtremeDB/rt from its competitors is that it is the first commercially available deterministic embedded database. This database guarantees that data processing and data access are predictable and meet strict deadlines.

RAM and CPU

In Edge AI, it is crucial to pay attention to components such as RAM and CPU. These are often small devices that have to achieve a lot with limited resources.

An in-memory database is optimized to be as fast as possible and to take up as little storage space as possible. In addition to using as few CPU cycles as possible, the database must also be designed to store data in a compact way.

Indexes take up extra disk space. This is often the case within the SQL database. It's not designed for performance but more for retrieving records and reporting. Usually it's too slow for fast throughput and data processing. The consequence of having too many indexes is index fragmentation. You need a DBMS with as little indexes as possible. Less indexes mean less data transactions.

Data compression

Implementing data compression algorithms within the database can reduce storage space and reduce I/O operations, which reduces energy consumption. This is especially relevant for edge AI devices with limited storage and processing capacity.

3. The Platform Approach

Best Practices

The balance between fail-safety and lifetime
All the principles I have discussed before are ultimately dependant on what is most important for your use case. It’s a matter of making choices between data integrity, fail-safety, performance and flash lifetime.

By following these best practices, you can design and develop edge devices that integrate AI, as well as the latest and greatest soft- and hardware. Leveraging the benefits of on-device processing while ensuring efficiency, security and a future proof design.

Platform approach
Not all edge devices are created equal. Depending on your use case, you may need different hardware and software capabilities, such as memory, processing power, battery life, connectivity, and security. Approach it as a cohesive platform rather than isolated stacks that remain the same regardless of the addition of AI models.

For example:
- If you are running a facial recognition model on a smart camera, you may need a powerful GPU and a reliable networking stack.
- If you are running a speech recognition model on a smart speaker, you may need an energy-efficient CPU and a robust offline mode.

You should also consider the compatibility and scalability of your edge device with your cloud platform and your AI framework.

Optimize your AI model
Running a complex AI model on an edge device can be challenging, as you may face trade-offs between accuracy, speed, and size. To optimize your AI model for edge computing, you should apply some techniques, such as:
- Quantization
- Pruning
- Compression
- Distillation

These techniques can help you reduce the number of parameters, the computational complexity, and the storage space of your model, while maintaining or improving performance. You should also test your model on different edge devices and scenarios, and monitor its behavior and results.

Secure your AI model
Securing your AI model on the edge is essential for its safety and integrity. However, securing your AI model on the edge can be risky, as you may face threats such as malware, hacking, manipulation, or counterfeiting.

To secure your AI model on the edge, you should use some strategies, such as:
- Encryption
- Authentication
- Authorization

4. Next Steps

You've reached the end of this journey into data management. We've delved into the complexities of data management at the edge and discovered best practices and a first use case.

What's Next for AI at the Edge

Edge AI applications can potentially transform and improve businesses and their services. The combined usage of AI-based algorithms and IoT-based devices and sensors is bringing innovative solutions to businesses across various industries.

However, the successful implementation of Edge AI requires careful consideration of different practices such as scalability, security, privacy and other ethical considerations.

Contact us to learn more about how we can help you with Edge AI application development.

Read more on:
File Systems
Database Management Systems

Embedded Tools & Software
30 years of embedded expertise

☎ +31 (0)77 3078438 (EN)
+49 89 2152 6995 (GE)

🌐 www.logic.nl
info@logic.nl

📍 Van Horneplein 6
6019 BW Wessem

logic technology