Out of the Box and at the Edge

An informal chat by Jose Alvarez (20250213)

In recent years, the term “edge computing” has come to refer to the concept of processing data at its source of creation in order to free server resources and reduce bandwidth. This concept has been widely applied in Internet of Things (IoT) devices and other automated data gathering systems. However, this concept can be further extended to include processing data at the points where it is consumed in order to offload even further server side processing. This idea is similar to a concept proposed by CISCO in the first half of the last decade, known as “fog computing”.

Cloud computing is a convenient and powerful solution for many applications, but it can quickly become costly. To address this issue, edge computing can be used to keep the more affordable cloud resources while pushing the more expensive ones to the edge. This can help to reduce costs significantly while still providing a reliable and secure solution.

In this chat we will discuss the implications of this extended concept of edge computing and explore its potential applications in the future through a set of practical examples ranging from the most simple ones to the machine learning enabled.

Example 1: The standalone URL shortener.

How the (failed) introduction of new cloud fees prompted a shift from an open source URL shortener running on a cheap cloud instance to a fully distributed system hosted statically in a git service provider.
Technologies: Javascript, YoURLs, GCP.

Example 2: A full-text search database at the edge.

CDNs are great for distributing multimedia content and even better to hold properly partitioned indices that can be combined and queried on the client side for a search-as-you-type query response.
Technologies: Golang, WebASM.

Example 3: Traditional tune title teller.

Combining machine learning and the shards of an index to recognise Irish traditional tunes as they are being played.
Technologies: TensorFlow, Javascript.

Example 4: You only look once, “you” being the keyword.

Using YOLO inference at the edge to turn a research project into a commercial application without venture capital or going broke in the process.
Technologies: Yolo, ONNX, AWS.

Example 5: WTF is morphology, syntax and semantics.

A proposal on how applying well established natural language rules could help making large language models a little less large, a little more efficient and much more edgy.
Technologies: Wide Token Format, Birds Eye View, Universal Dependencies, Golang, WebASM.

Example 6: Large language models, I had to cheat on this one but I will get it right eventually.

An application of of the previous example using limited versions of the current models run locally.
Technologies: Dart, Flutter, Golang, llama.cpp.

1. The Standalone URL Shortener

Niuter was a small news aggregator running in Twitter that heavily relied in URL shortening for its operation. At the time, Bitly was the service of choice for this task but in March 2015 they announced they were starting to charge for the amount of links being generated. This prompted the creation of an in-house shortening service consisting in a single nano instance in the Google Compute Engine service for an approximate price of 5 Euros per month. The instance had installed YoURLs open source shortener based on PHP and an SQL database. On top of that, three monitoring alerts were set to check on login attempts, network traffic and budget overruns. In July 2024 Google announced it would be charging 1.5 Euros monthly for each one of the alerts. This would mean doubling the monthly charges. Enter the edge. Traditionally, URL shorteners rely on the HTTP protocol 300 range messages managed at server level, but from a very long time now redirections can be accomplished programatically at browser level through a very simple Javascript function. This allows for a URL shortener that is fully static at server level and therefore free of charge through the use of services like GitLab pages.
Total savings: around 10 Euros per month.

2. A full-text search database at the edge

ErrantDB is the perfect hybrid between Cloud and Edge computing. It generates a static index broken down into partitions (or shards) that can be stored and distributed through a CDN. It uses a high speed inference engine written in Go and compiled to Web Assembly code that runs on the client side retrieving and combining the shards relevant to the user’s query. In this manner, it takes advantage of the most affordable cloud technology, storage, and the computing power of the customer's computer to create a fast and infinitely scalable information retrieval system.
For demonstration purposes all the metadata from Project Gutenberg was map-reduced using an n-gram method, indexed through ErrantDB and made available for general use.
Monthly cost: 0 Euros per month.

3. Traditional Tune Title Teller

There are several tools that help bring Machine Learning (ML) to the edge and the Javascript version of TensorFlow is one of them. In 2021 a colleague from Avaya, and fiddle player, noticed a lack of relevant results in Shazam regarding Irish Traditional Music. TraDSP was then created to fill in this gap: an Irish traditional music search engine. It is an exercise on the importance of multi-disciplinary knowledge and pipelines, both at the time of training and at the time of decision making. Many different techniques were necessary to bring it to fruition including search engines, digital signal processing DSP and fuzzy logic: As a search engine, ErrantDB was used but applying n-gram techniques to music notation instead of text; this was combined with a TensorFlow model doing some DSP for note recognition; and finally a fuzzy logic algorithm compensated for the different styles of playing the same tune. Although it is mostly a work of engineering, it wouldn't have been possible without the incredible work of ITMA and the contributions from all the artists and editors involved with the institution.
Monthly cost: 0 Euros per month.

4. You Only Look Once, “You” Being the Keyword

The BLUEPOINT project is an initiative to address the problem of marine plastic waste through sustainable and circular solutions. At the Centre for Robotics and Intelligent Systems (CRIS) in University of Limerick a system was develop to retrieve waste plastic from natural coastal areas. The first step consists in locating the waste plastics by surveying the area using drones. The drones take high resolution pictures of the ground which are subsequently processed using a fine tuned model derived from the open source image classification system YOLO. One of the biggest concerns was having the project die of its own success because of not having enough resources to process large amounts of high resolution images. An ONNX based system was devised to allow to do the image processing straight in the browser of the drone remote pilots taking advantage of the direct access to the local GPU through WebGL and WebGPU extensions. This approach off-loads from the server the most processing intensive tasks.
Monthly cost: 0 Euros per month.

5. WTF is morphology, syntax and semantics

The Wide Token Format and the Bird’s Eye View method provide a means of representing a document concurrently in an abstract and a concrete way.

The document as a whole is first set in a full circle (shown in blue). Then, every word is lemmatised and accounted for using vectorial map-reduction in such a way that each lemma is placed in the X-Y plane as the average position of every inflected appearance in the circle and vertically as a normalised function of the number of times that it occurs (shown in green).

The map-reduced version of the document is then truncated using a conic pattern (shown in red) and

used as input for the global attention layers while the words in the circle are divided using fixed sized arcs and used as inputs for the local attention layers.

The ultimate goal of this process was to create a tokenisation and embedding system for Large Language Models based on well defined grammatical rules instead of forcing the models to learn abstract grammars ultimately disconnected from the original natural languages.

The analysis and visual representation of the documents can be performed at the edge, again using WebASM and WebGL technologies.

Monthly cost: 0 Euros per month.

6: Large language models, I had to cheat on this one but I will get it right eventually.

Having failed to secure GPU hours for training a model using BEV and WTF technology we are currently developing pipelines and tools for using these methods for enhancing LLM prompt generation and boosting the performance of smaller models that can be reasonably run in personal computers.