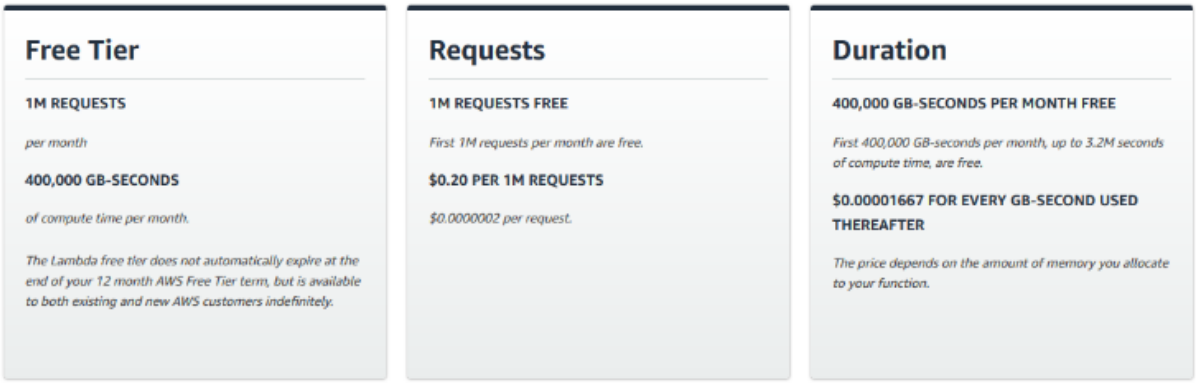
Economic Breakdown for the Short-Term Model

As part of the model's development, the system's implementation needs to be financially evaluated to determine the cost of its deployment on an industrial level. The Terms of Reference outlines the project's main finances, but here will be a more in-depth exploration of the financial aspects behind the model specifically. There will also be a brief look at the potential finances behind developing a more sophisticated model.

Current Model

If the model was to continue being used in its current form, this would involve it continuing to retrieve device data from the system's database via the use of an endpoint. This endpoint takes the user's id then returns all associated device information, allowing it to be searched for relevant information that can be used for the short-term decision-making process.

The primary cost consideration here would be the continued and increasing usage of AWS Lambda, an Amazon service that automatically runs associated code for various events. In the case of our system, the event is when an http endpoint is triggered. Usage of Lambda remained free during the prototype stage thanks to the relatively low amount of requests and data use. Moving forward however, possible costs will need to be considered. Lambda's usage is based on two things, requests and compute time. Requests pertains to simple requests that call Lambda code, and compute time is the time it takes for the code to execute and terminate. The costs (shown below in dollars) can be viewed on AWS' Lambda page.



Requests

Lambda offers the first million requests each month for free. First how many users the system in its current implementation can support whilst remaining in this free tier will be calculated, then a rough cost per user in request fees will be determined.

The model sends a request to the endpoint that returns device data approximately every four seconds. To be completely prepared for financial ramifications, the most intensive scenario will be assumed where the model runs constantly and is at no point deactivated for maintenance or updates.

With 86400 seconds being in a day, scheduled retrieval of a single user's data will use 21600 (86400/4) requests each day. That means over the course of 30 days, a single user's model will use 648000 requests. Dividing a million by this figure yields a little over 1.5, meaning in its current state

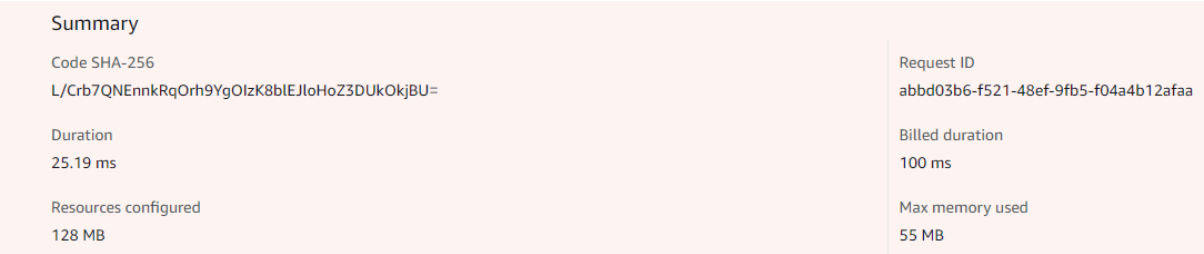
the system can only support one user full time without incurring fees. Beyond this point, requests would need to be paid for.

Lambda offers 1 million requests for $0.20 (AWS Lambda, 2018), which is equivalent to £0.15 at the time of writing. We can work out the cost of a single request by diving this cost by 1 million, then by multiplying this by the number of requests needed for a single user's model we can determine how much in requests a single user will cost. By doing this, we arrive at the figure of £0.0972 per user. So, for each user that uses the product that the current model needs to accommodate for, an additional £0.0972 per month will need to be spent on requests from AWS Lambda.

Compute Time

Lambda offers 400,000 gb-seconds per month free per month. First it will be determined how many users can be supporter by the system in its current implementation without incurring costs. Then a cost per user in compute time fees will be determined.

The screenshot below shows the Lambda details for the endpoint that returns user device data once supplied with a user ID.



The memory used by the endpoint doesn't always equal 55mb, but to ensure that the worst-case scenario has been prepared for it will be assumed that every request uses this much data. This amount of data can increase if more device data is returned but this increase is ultimately negligible, nearly all the computing time used is from processing the respective scripts and code that is used for the endpoint's function rather than the actual bandwidth used returning the data.

400,000 GB-seconds per month equates to around 400000000 megabytes. This is enough to process around 7272727 requests. It was earlier established that a single user will (in a worst-case scenario) require 648000 requests to retrieve up to date model information. This means that from a free perspective, the current system's usage of Lambda can support around 11 users (7272727/648000 is about 11) before costs are incurred from the usage of the endpoints. After this point, a single user's model requests will use around 35640000 megabytes of compute-time, equating to around $0.59 in monthly costs. At the time of writing, this translates into around £0.45.

In summary, the free tier will be exceeded by just two users. Even though Lambda's memory allowance offers support for many more, the sheer number of requests required to keep the model up to date burns through the free allowance quite quickly. Only the respective free allowances have been exceeded, each user will cost £0.0972 in request fees and £0.45 in compute-time fees, giving a total per-user cost of roughly £0.54.

Improved Model

Rather than continuing with the current model's implementation, a more sophisticated model could be created that takes advantage of true deep learning technologies. This will allow its behaviour to be more accurate, as it will be capable of picking up on small nuances in user behaviour and distinguishing them from real problems. For example if a user frequently has parties at certain times

of the year, the current implementation might flag this activity as a high temperature whereas a true machine learned model might recognise it and understand that it isn't a problem.

AWS

AWS offers technologies capable of deep learning. Whilst its lack of prior use in the project makes it difficult to estimate the cost per user as was done with the current model's implementation, instance pricing can still be reviewed.

The three main rentable instances in order of power are P2, P3 and G3 (AWS Deep Learning Developer Guide, 2018). To begin with, it would be best to start with the cheapest option. This would prevent money being wasted should this foray into deep learning prove to not be useful, but should more power be required it will be easy to scale up compared to wasting money and realizing what was purchased was overkill.

P2 instances are categorised by their specifications, with more expensive instances containing additional GPUs, vCPUs and more RAM, among other things. The closest P2 instances that can be obtained are based in Ireland, below is a pricing breakdown taken from the EC2 instance documentation of how much these instances would cost.

p2.xlarge	4	12	61 GiB	EBS Only	$0.972 per Hour
p2.8xlarge	32	94	488 GiB	EBS Only	$7.776 per Hour
p2.16xlarge	64	188	768 GiB	EBS Only	$15.552 per Hour

How much this will cost is something that can only be properly determined once they have been used, as how many computing hours are needed to train the model is unknown. As well, the model may not be usable during its training, but frequent training will likely be needed in order to keep up with user habits as they change and nuances occur. Two different instances could be obtained as a work around for this. As one model trains, the other performs the necessary short term decision making. Once this first model is done training it takes over and the second model begins training. This could be repeated frequently to allow for the models to both stay up to date and current, but alternating deep learning instances would mean this could happen with any service interruption.

Sources

- Amazon. AWS Lambda Documentation. Available at: https://aws.amazon.com/lambda/ (Accessed (03/05/19).
- Amazon. Recommended Deep Learning Instances. Available at: https://docs.aws.amazon.com/dlami/latest/devguide/gpu.html (Accessed (03/05/19).
- Amazon. EC2 Pricing. Available at: https://aws.amazon.com/ec2/pricing/ (Accessed 03/05/19).
- Amazon. P2 Instance Details. Available at: https://aws.amazon.com/ec2/instance-types/p2/ (Accessed 03/05/19).