2.3 Cost Breakdown Analysis

One of the main reasons companies decide to adopt serverless functions is because of its pay-per-use model offering which merits reduced hosting cost compared to traditional hosting. In this section, this pricing model is investigated to identify the cost implications of hosting serverless functions. Since each CSP has slightly different pricing models, each is outlined individually. All four of the major cloud service providers of FaaS provide a pay per execute model [8, 10–12]. An example use case is applied to each of the services to demonstrate the slight variations in the pricing models. The formula for calculating cost varies from different CSPs. When calculating the price, any free tier offering was ignored for this analysis.

A function with 128MB of memory, invoked 8 million times per month and running for 300ms each time.

This use case is a simple example that was applied to each pricing model. In a real-world scenario, calculating costs would not be as straightforward as the running time may vary. A real-world scenario will be presented in the experiments stage of this thesis.

2.3.1 AWS Lambda

With AWS Lambda the pricing model depends on the number of requests, the duration of each of the requests and the allocated memory. The duration of the request is calculated from the time the request begins executing to the time it returns or terminates. The duration price is calculated based on the number of resources that have been reserved to run the instance, this is a preassigned value [8].

The number of requests is charged at $0.20 per 1 million requests. The duration pricing is charged at $0.0000166667 for every GB-second. The charge for duration depends on the amount of memory allocated to the lambda function.

Table 2.1 presents the formulas used to calculate the monthly charges of the serverless functions.

TABLE 2.1: AWS Lambda Pricing Formula

Metric	**Formula**
Total Compute Seconds	Request Count x Execution Duration (seconds)
Total Compute GB-s	Total Compute Seconds x Allocated Memory / 1024
Monthly Compute Charge	Total Compute GB-s x $0.0000166667
Monthly Request Charge	Request Count ÷ 1,000,000 x $0.20
Total Charge Per Month	Monthly Compute Charge + Monthly Request Charges

Using the formulas outlined in Table 2.1 the use-case can be calculated. These calculations are shown in Table 2.2

TABLE 2.2: AWS Lambda Use-Case Calculation

Metric	**Calculation**	**Result**
Total Compute Seconds	8,000,000 x 0.3	2,400,000
Total Compute GB-s	2,400,000 x 128MB / 1024	300,000
Monthly Request Charge	8 x $0.20	$1.60
Monthly Request Charge	300,000 x $0.000016674	$5.00
Total Cost Per Month	$1.60 + $5.00	$6.60

2.3.2 Azure Functions

Azure Functions has a very similar pricing model to AWS Lambda described in section 2.3.1. The charges for Azure Functions are also based on the number of requests and the execution time in GB-seconds [10].

Total Executions variable is the total number of requests made per month for all functions. Executions are counted each time a function is triggered by an event. This does not include any errors in which the function did not execute such as 401 errors. The customer gets billed $0.20 per million executions [10].

Execution time is calculated in gigabyte seconds (GB-s). Calculated by multiplying the average memory size in gigabytes by the time in milliseconds it takes to execute the function. The memory used by the function is rounded up to the nearest 128 MB and execution time rounded up to the nearest 1 millisecond. The total cost to the customer is $16 per million GB-seconds. This is based on a Function executing 1 million times, consuming 1 GB of memory at each execution and completes in 1 second [10].

Table 2.3 presents the formulas used to calculate the monthly charges of the serverless functions.

TABLE 2.3: Azure Functions Pricing Formula

Metric	**Formula**
Total Compute Seconds	Request Count x Execution Duration (seconds)
Total Compute GB-s	Total Compute Seconds x Memory Usage / 1024
Monthly Compute Charge	Total Compute GB-s x $0.000016
Monthly Request Charge	Request Count ÷ 1,000,000 x $0.20
Total Charge Per Month	Monthly Compute Charge + Monthly Request Charges

Using the formulas in Table 2.3 the use-case can be calculated. These calculations are shown in Table 2.4

TABLE 2.4: Azure Functions Use-Case Calculation

Metric	**Calculation**	**Result**
Total Compute Seconds	8,000,000 x 0.3	2,400,000
Total Compute GB-s	2,400,000 x 128MB / 1024	300,000
Monthly Compute Charge	300,000 x $0.000016	$4.80
Monthly Request Charge	8 x $0.20	$1.60
Total Cost Per Month	$1.60 + $4.80	$6.40

2.3.3 Google Fuctions

Google Functions also calculates charges based on request count and execution time measured in GB-Seconds. Compute time is measured in 100ms increments, rounded up to the nearest increment. Google Functions also charge for CPU provisioned in addition to the other variants. This is where Google Functions differs from AWS Lambda and Azure Functions. CPU charge is calculated in GHz-Seconds where 1 GHz-second is 1 second of wallclock time with a 1GHz CPU provisioned [11].

Table 2.5 presents the formulas used to calculate the monthly charges of the serverless functions.

TABLE 2.5: Google Functions Pricing Formula

Metric	**Formula**
Total Compute Seconds	Request Count x Execution Duration (seconds)
Total Compute GB-s	(Total Compute Seconds x Memory Usage / 1024) x $0.0000025
Total Compute GHz-s	(Total Compute Seconds x CPU Usage / 1000) x $0.0000100
Monthly Compute Charge	Total Compute GB-s + Total Compute GHz-s
Monthly Request Charge	Request Count / 1,000,000 x $0.40
Total Charge Per Month	Monthly Compute Charge + Monthly Request Charges

Using the formulas in Table 2.5 the use-case can be calculated. These calculations are shown in Table 2.6. It is assumed for this calculation that the function uses 200MHz of CPU.

TABLE 2.6: Azure Functions Use-Case Calculation

Metric	**Calculation**	**Result**
Total Compute Seconds	8,000,000 x 0.3	2,400,000
Total Compute GB-s	2,400,000 x 128MB / 1024	300,000
Total Compute GHz-s	2,400,000 x 200 / 1000	480,000
Monthly Compute Charge	(300,000 x $0.0000025) + (480,000 x $0.0000100)	$5.55
Monthly Request Charge	8 x $0.40	$3.20
Total Cost Per Month	$3.20 + $5.55	$8.75

2.3.4 IBM Functions

Unlike the other Cloud Services Providers pricing model investigated in this section. IBM cloud functions pricing is based solely on the execution time of the function in gigabyte seconds and does not charge for function invocation. Therefore, the price of running IBM functions is based on the time executed and the previsioned memory of the function. The IBM cloud function rate is charged at $0.000017 per second of execution, per GB of memory allocated [12].

Table 2.7 presents the formulas used to calculate the monthly charges of the serverless functions.

Table 2.7: IBM Functions Pricing Formula

Metric	**Formula**
Total Compute Seconds	Request Count x Execution Duration (seconds)
Total Compute GB-s	(Total Compute Seconds x Memory Usage / 1024) x $0.000017
Total Charge Per Month	Total Compute GB-s

Using the formulas in Table 2.7 the use-case can be calculated. These calculations are shown in Table 2.8.

Table 2.8: IBM Pricing Breakdown

Attribute	**Calculation**	**Result**
Execution Count	Not Charged	$0.00
Total compute (seconds)	8,000,000 x 0.3	2,400,000
Total compute (GB-s)	2,400,000 x 128MB / 1024	300,000
Total Cost Per Month	300,000 x $0.000017	$5.10

2.3.5 Summary

In this section, a common use-case was applied to each of the leading cloud service providers to demonstrate their pricing models and the differences. The results of the use-case cost analysis are displayed in Use-Case Cost Summary. As shown, Google Functions had the highest running costs where IBM functions had the lowest.

Table 2.9: Use-Case Cost Summary

Cloud Service Provider	**Total Cost**
AWS Lambda	$6.60
Azure Function	$6.40
Google Function	$8.75
IBM Functions	$5.10

However, the purpose of this exercise was to demonstrate how each of the cloud service providers calculated cost. In each case, there is a common factor that affects the running costs, memory consumption and execution duration. Therefore it was demonstrated how

reducing the memory consumption and execution duration of a serverless function has a direct impact on hosting cost. This element of the pay-per-use model is the basis for the migration technique proposed in this research. By improving the performance and reducing memory consumption the serverless functions will run at a reduced cost compared to functions running without the implementation of the migration technique.