Believe in Serverless

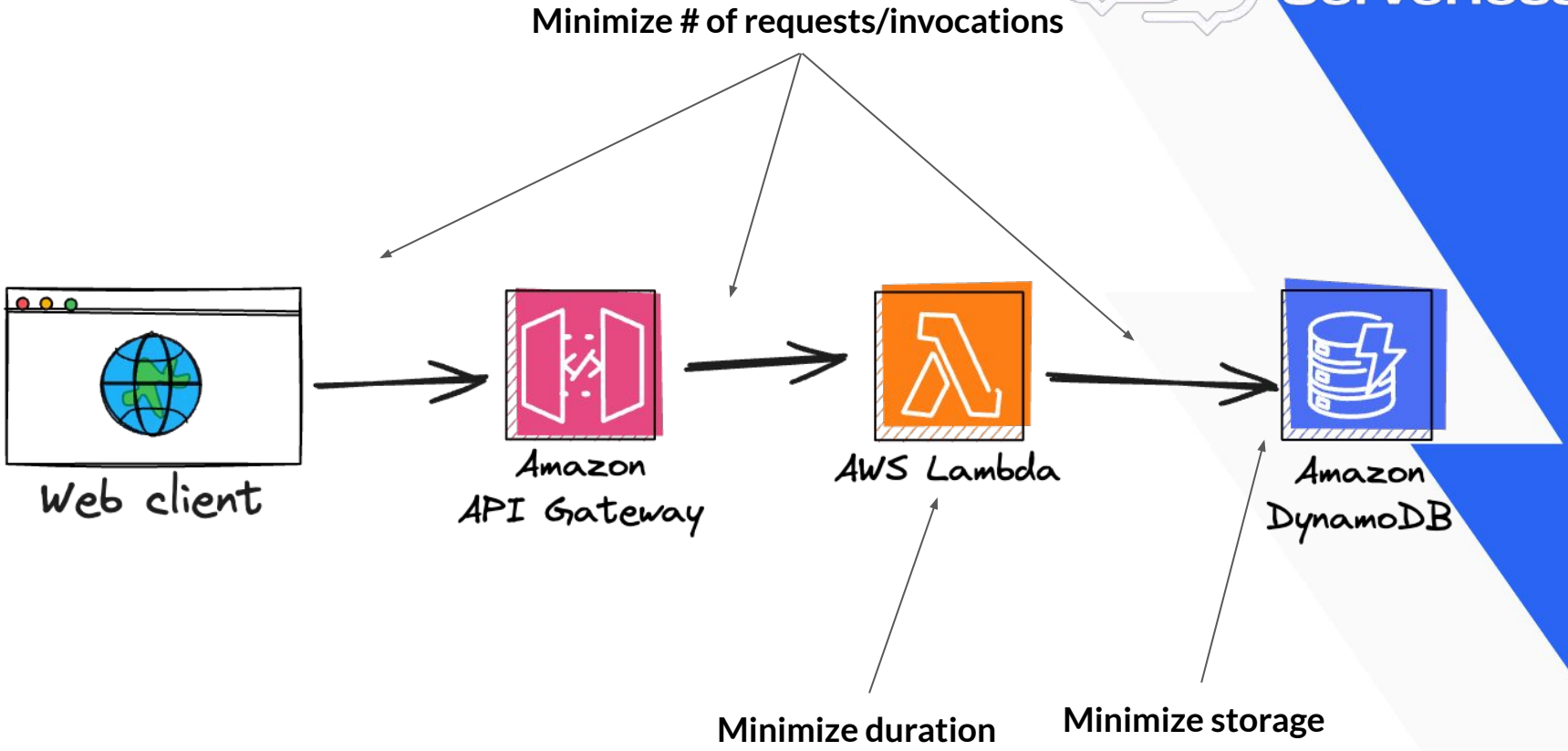
Serverless Cost Optimizations:

Theory

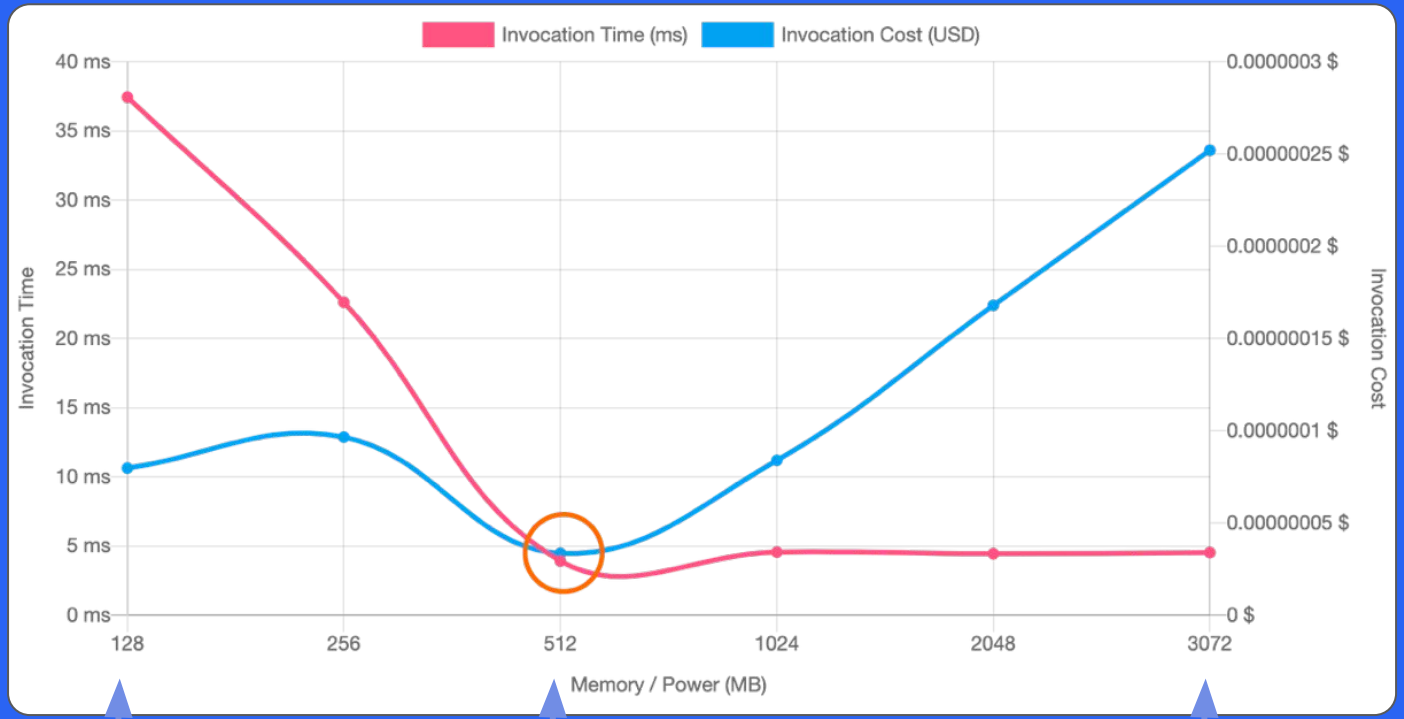
Believe in Serverless

The Basics

World-class AWS experts


Minimize # of requests/invocations
Web client
Amazon API Gateway
AWS Lambda
Amazon DynamoDB
Minimize duration
Minimize storage

Believe in Serverless



Original

Option A

Option B

Original

- Memory: 128MB
- Duration: 36ms

Option A (👍):

- Memory: 512MB
- Duration: 4ms
- Savings: +36%
- Speed: +88%

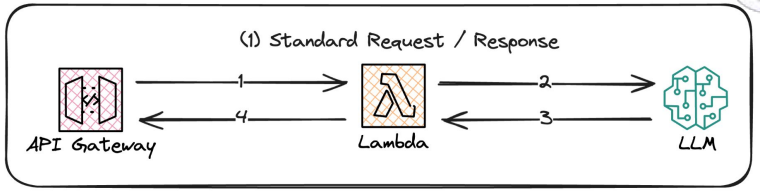
Option B (👎):

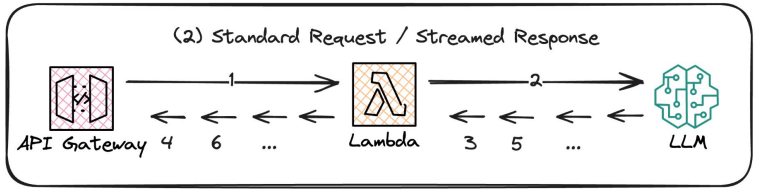
- Memory: 3072MB
- Duration: 4.5ms
- Savings: -257%
- Speed: +87%

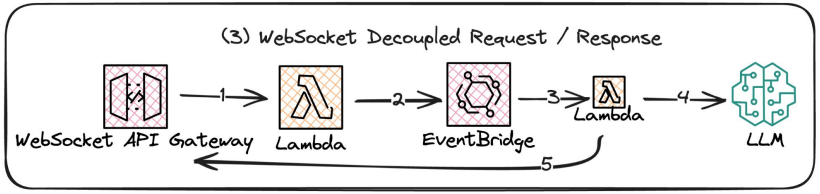
World-class AWS experts

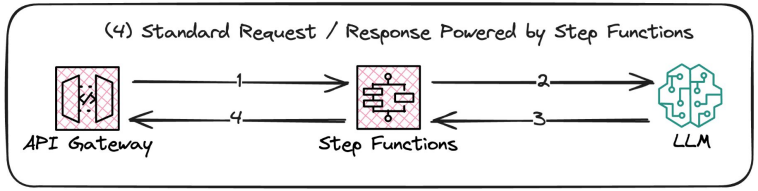
https://aws.amazon.com/blogs/compute/optimizing-your-aws-lambda-costs-part-1/

LLM Access Patterns







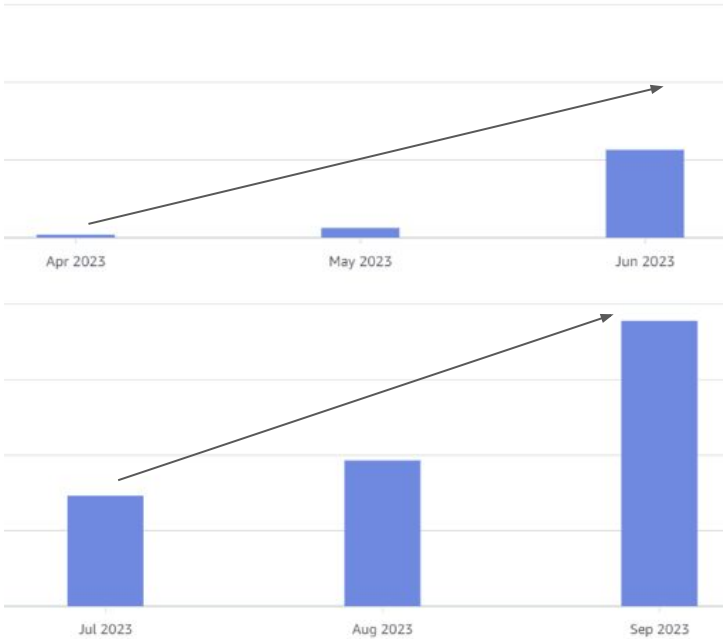


Believe in Serverless

Serverless Cost Optimizations:

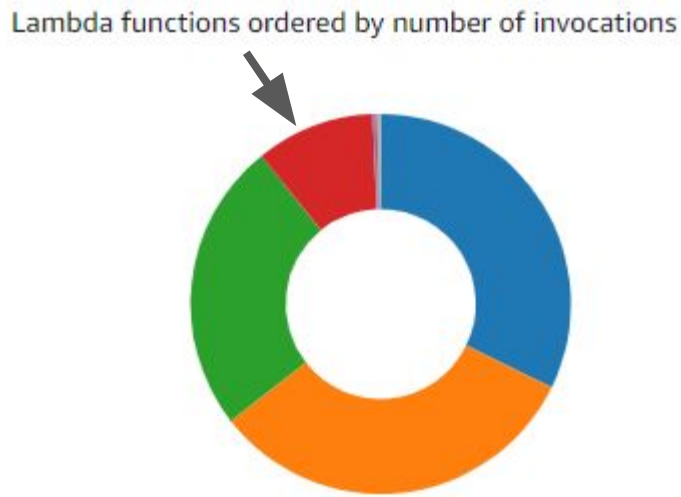
An Example

The 100x Lambda



Looking at the AWS Cost Dashboard we realized there was an 100x increase in the Lambda costs in only 6 months

The 100x Lambda

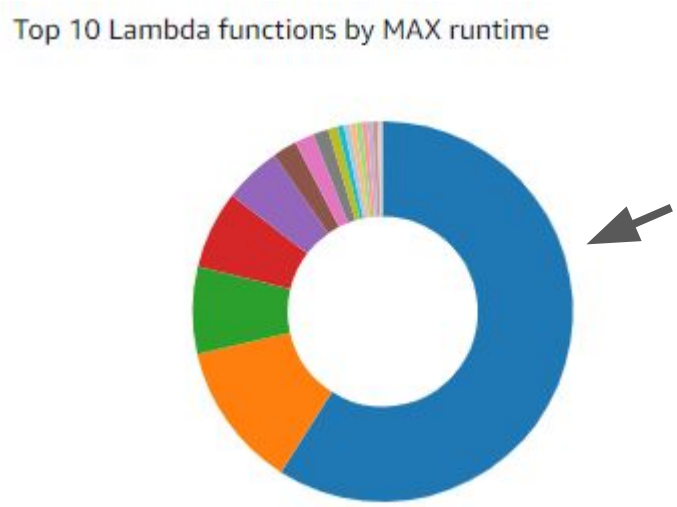


A single lambda at fault

Using CloudWatch Metrics Dashboards we confirmed the red slice was our costly lambda and only accounted for **10% of invocations**

Believe in Serverless

The 100x Lambda



Max execution time is long ⏰

Here we can see the same Lambda (now shown in BLUE) is by far the one that has the longest running request. But what is causing the long running times?

The 100x Lambda

Believe in Serverless

Percentage Distrib

duration_bucket	0-10	10-20	20-30	30-40	40-50	50-60	60-70	70-80	80-90	90-100	100-110
path											
Path1	88.45	9.64	0.22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Path2	97.99	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Path3	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Path4	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Path5	17.65	82.35	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Path6	44.44	0.00	33.33	22.22	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Path7	97.32	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Path8	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Path9	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Path10	89.65	8.77	0.22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Path11	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Path12	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Path13	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Analyzing Lambda Durations

We categorized the request / response times by path e.g. Path 1, Path 2, etc with a duration between 0-10, 10-20, etc

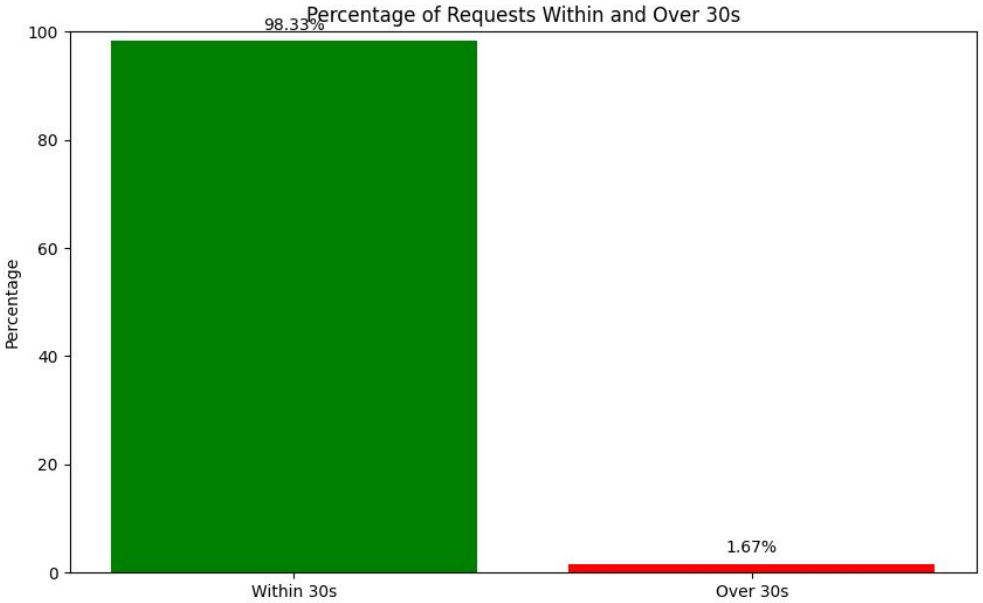
It's interesting to see that there's a gap between ~40s and ~110s duration time. That suggests that there's something wrong with those requests that are potentially causing it to hang.

The 100x Lambda

110-120	120-130	130-140	140-150	150-160	160-170	170-180	180-190	190-200	200-210	210-220	220-230	230-240	240-250
0.04	0.11	0.04	0.00	0.11	0.04	0.07	0.04	0.07	0.15	0.04	0.07	0.07	0.82
0.00	0.13	0.00	0.07	0.00	0.10	0.10	0.20	0.03	0.00	0.03	0.03	0.07	1.25
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.89	0.00	0.00	0.00	0.89	0.00	0.00	0.00	0.89
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.02	0.00	0.05	0.07	0.10	0.02	0.05	0.02	0.02	0.07	0.00	0.05	0.07	0.79
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Anything highlighted in red is longer than 30s. We found out that a small percentage of requests in only a few paths were taking a long time to respond.

The 100x Lambda



Percentage of Requests

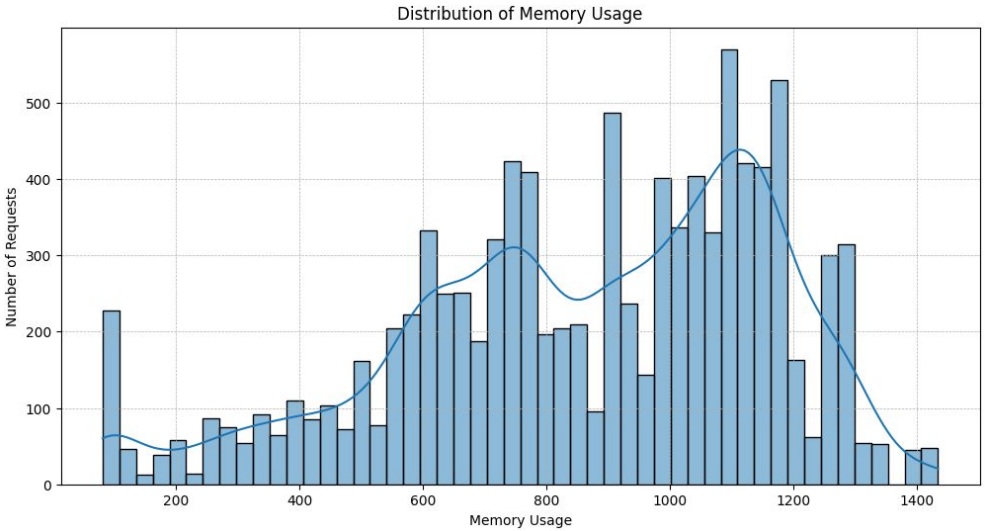
Only 1.67% of requests took longer than 30s.

Waste Analysis

- $10k/mo in waste
- < 2% responsible for 1/3rd cost

Believe in Serverless

The 100x Lambda



Lambda was set to 4096 MB which was very over provisioned.

An adjustment of memory to 1536mb would be a good first step, and it's 62.5% cheaper or $6k in savings off memory alone.

Reducing timeout + memory, saved roughly $15k/mo.

Believe in Serverless

Believe in Serverless

The 100x Lambda: Recap

- Looking at the AWS Cost Dashboard we realized there was an 100x increase in the Lambda costs in only 6 months
- Upon further investigation it became clear that the increase in cost was driven mostly by a single Lambda
- Analyzing the Lambda Logs we realized that this particular Lambda had a few requests taking up to 4 minutes to respond

Believe in Serverless

The 100x Lambda: Recap

- Long requests were consuming a lot of Lambda Concurrency up to the Max Concurrency Level, causing new invocations to throttle on every service on the same account.
- Realized these requests were hanging and after 120 seconds even if a valid response comes back it would no longer be useful since the connection was already closed by the CDN provider.
- Thus it was decided we should reduce the Lambda Max Timeout to reduce unnecessary costs and throttling. Which led to $15k/mo in savings!

Believe in Serverless

That’s it folks!

Thank you!