Session CTD405

Optimizing Lambda@Edge for Performance and Cost Efficiency

Achraf Souk
Solutions Architect
Amazon Web Services

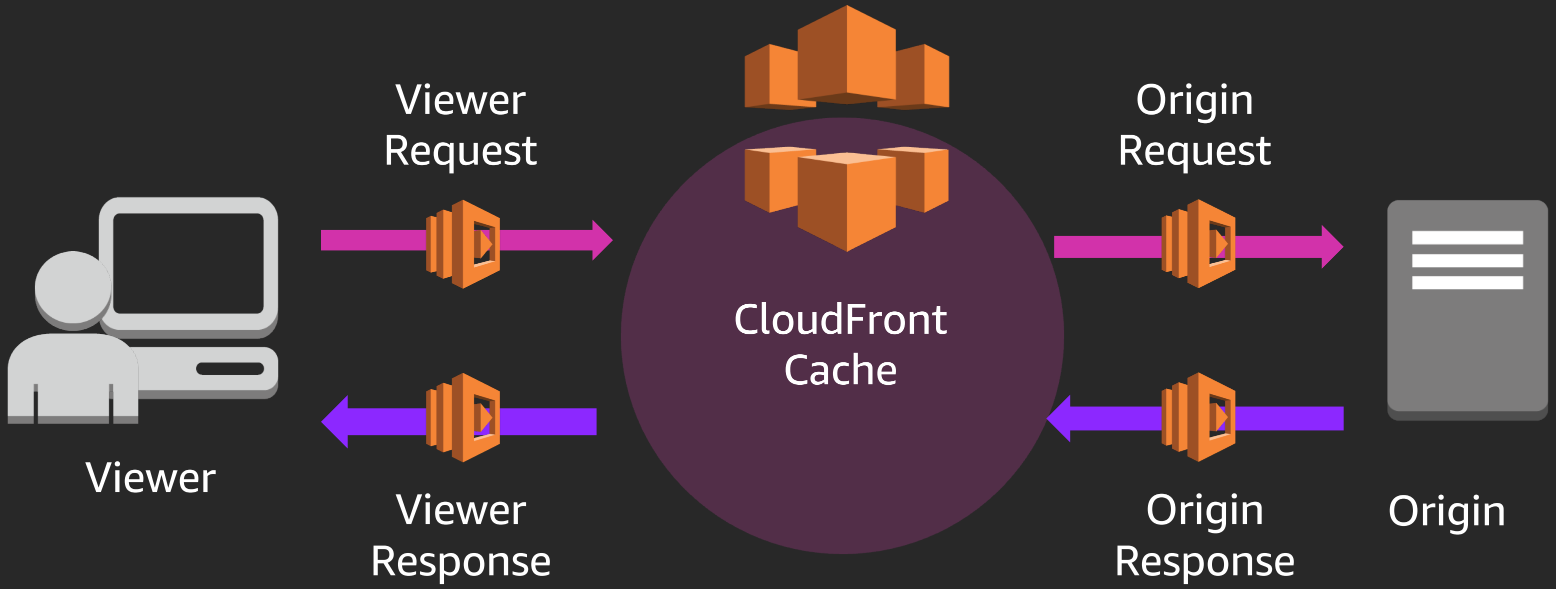
Alexander Korobeynikov
Software Dev Engineer
Amazon Web Services

AWS re:Invent

Lambda@Edge use cases

Simple HTTP manipulations	Dynamic content generation	Origin independence
User-Agent header normalization	Image manipulation	Pretty URLs
Adding HSTS security headers	Render pages	API wrapper
Enforcing Cache-Control headers	Redirections	Authorization
A/B testing	SEO optimization	Bot mitigation

CloudFront and Lambda@Edge



AWS re:Invent

aws

How much does it cost?

Consider an API with 15M requests/month & 128MB Lambda@Edge function executing in 2ms. Viewer Request event is configured on CloudFront.

Lambda@Edge is charged based on:

- Number of requests: 15M*,6$/1M = 9$
- Memory*Duration resource usage: 15M * 50ms * 128MB * 0,00005001$/GBS = 4,7$

Total cost is 13,7$/month

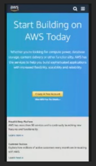
Why bother optimizing?

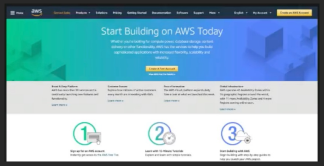
Is Lambda@Edge the right solution for you?

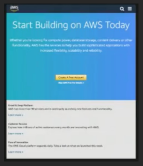
#1 Consider all the available options

- CloudFront already provide native features:
  - Device identification: CloudFront-Is-Mobile-Viewer headers
  - Analytics: CloudFront Access Logs delivered to S3 & WAF logs
  - Access Control: CloudFront signed URLs/Cookies, Geoblocking, WAF

- Leverage responsive web design







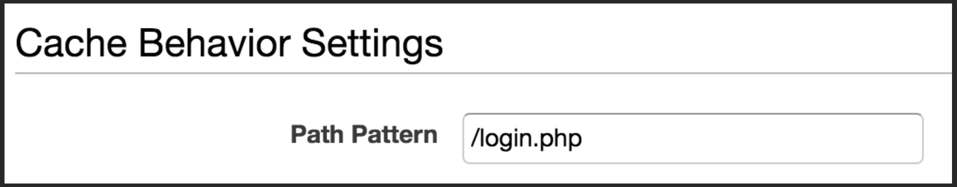
- Some logic is better off on the origin!

Optimizing Lambda@Edge configuration

#2 Invoke Lambda@Edge only when you need it

- For every request or only on cache misses?

- Use the most specific CloudFront behavior:

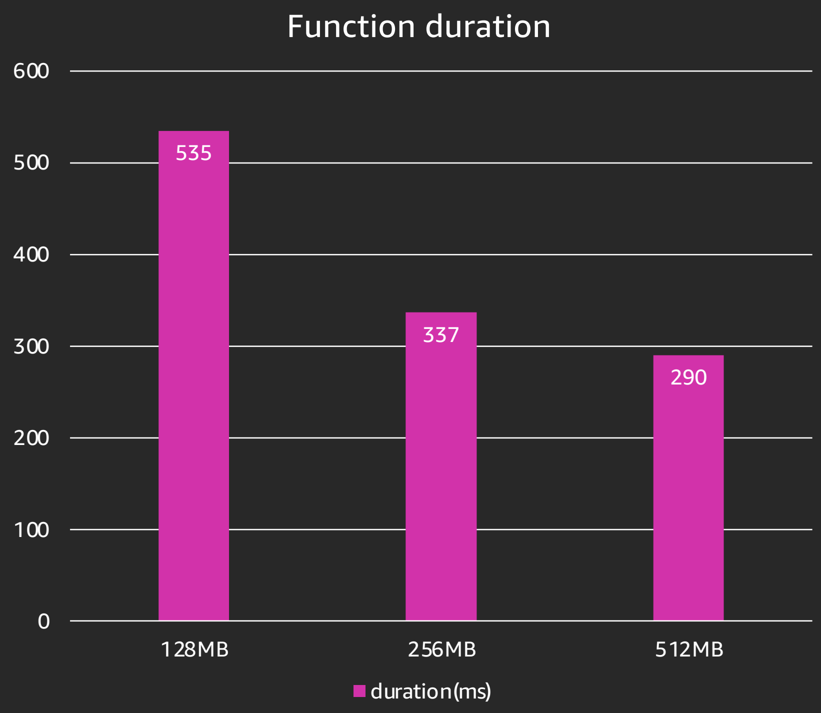


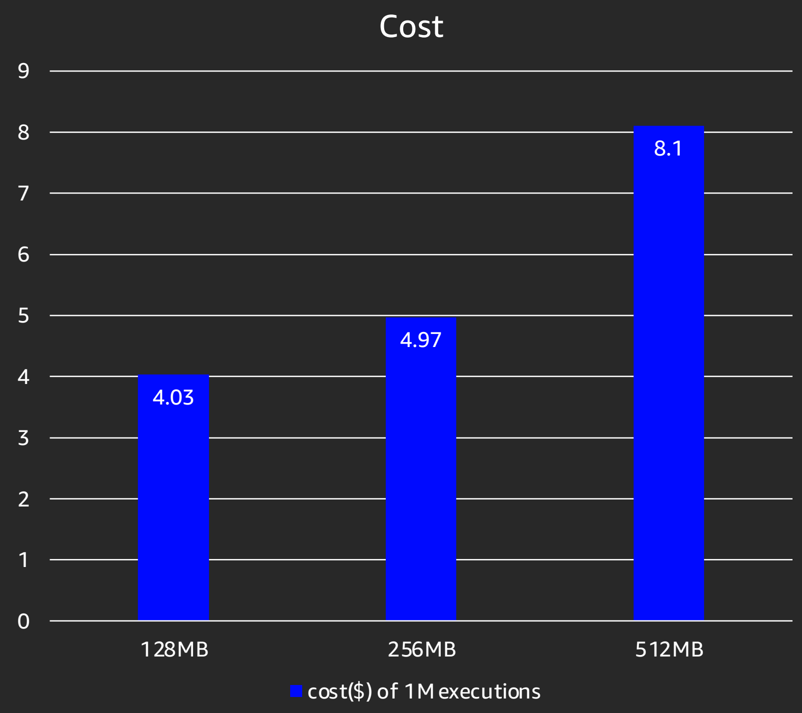
Cache Behavior Settings

Path Pattern: /login.php

- Remove it when it’s not used any more

#3 Choose the optimal memory configuration





AWS re:Invent

Optimizing Lambda@Edge code

#4 Optimize function code for Node.js

Node.js 8.10 runtime

Reduce deployment package size: external dependencies, lightweight packages, minify, browserfy…

Parallelism using async calls

```
let responses = await Promise.all([

      httpGet({ hostname: ”HTML template", path: ""}),

      ddbGet({ TableName: ddbTableName, Key: { name: ”achraf" } }),

]);
```

#5 Leverage global variables (1/2)

```
const dns = require("dns");
let bestOrigin;
let expires = 0;
exports.handler = (event, context, callback) => {
       const request = event.Records[0].cf.request;
       getBestOrigin().then((origin) => {
              request.origin.custom.domainName = origin;
              headers.host[0].value = origin;
              callback(null, request);
       });
}
```

#5 Leverage global variables (2/2)

```
function getBestOrigin() {
        const now = Date.now();
        if (now < expires) return Promise.resolve(bestOrigin);
        return new Promise((resolve, reject) => {
                dns.resolveCname(DNS_HOST, (err, addr) => {
                        bestOrigin = addr[0];
                        expires = now + TTL;
                        resolve(bestOrigin);
                });
        });
}
```

#6 Optimize external network calls

```
const http = require('https’);

const keepAliveAgent = new http.Agent({ keepAlive: true, keepAliveMsecs: 2000 });

exports.handler = (event, context, callback) => {

    http.get({ hostname: ”hello.com", path: ”/”, agent: keepAliveAgent}, (resp) => {

        let data = ’’;

        resp.on('data', (chunk) => { data += chunk; });

        resp.on('end', () => { resolve(data); });

    });

}
```

Know the limits!

#7 Know the limits!

Functional:

Blacklisted/ Read only headers

Function size – 1MB vs 50MB

Response size – 40K vs 1MB

Resource allocation

Memory – 128M vs 3G

Timeout – 5s vs 30s

1K concurrent execution region

Scaling mechanism

Thank you!

Achraf Souk
achrsouk@amazon.com

Alexander Korobeynikov
akorob@amazon.com

Additional resources

- https://aws.amazon.com/lambda/edge/
- https://aws.amazon.com/blogs/networking-and-content-delivery/lambdaedge-design-best-practices/

- https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/lambda-edge-testing-debugging.html

- AWS Blog: Cookie Synking for AdTech, Visitor prioritization for e-commerce, Paywall for publishers

!

Please complete the session survey in the mobile app.

AWS re:Invent

aws

AWS re:Invent