GeoFaaS: An Edge-to-Cloud FaaS Platform

M. Malekabbasi, T. Pfandzelter, T. Schirmer, D. Bermbach | Scalable Software Systems
Edge-to-Cloud FaaS & Serverless

- Imagine (near) future with mobile clients
- Their characteristics:
  - Geographically-distributed
  - Limited battery and computation power
  - many latency-sensitive applications
- Problem: edge-to-cloud systems complex dev & management.

- BUT, we are not considering client’s location!
  - In a changing network, physical distance effectively approximates latency [13], [14].
  - Client location measurement is cheap (no interaction with the system)
Geo-aware Function-as-a-Service

- A geo-distributed FaaS platform, across e2c continuum
- Transparency for end clients
  - i.e. publish on “f1/call” to call $f_1$
  - Through a client library for the common serverless abstraction
- GeoFaaS node has three key elements
  - (Distributed) Geo-aware Message Broker (“DisGB”)
  - FaaS server(s) (local or in the same data center)
  - Bridge, a middleware between the other two
Evaluation

- Prototype:
  - Kotlin prototype and client library.
  - Topics for a sample function $f_1$ (Table I)

- Experiment setup:
  - 2 $RPi^1$ Edge nodes, and one $GCP^2$ Cloud node (Figure)
  - Each node, running tinyFaaS, GeoBroker and GeoFaaS Bridge instances
  - All clients on a $Rpi$, in the same network as the edges

- Scenarios:
  1) “Distance/Latency Change”
  2) “High Load”, transparent offloading for uninterrupted service
  3) “Outage”, system’s resilience against GeoFaaS Bridge failure

<table>
<thead>
<tr>
<th>#</th>
<th>Topic</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>/f1/call</td>
<td>Client calls function (Bridge subscribes)</td>
</tr>
<tr>
<td>2</td>
<td>/f1/ack</td>
<td>Bridge acknowledges call (client subscribes)</td>
</tr>
<tr>
<td>3</td>
<td>/f1/result</td>
<td>Client subscribes for result (Bridge publishes)</td>
</tr>
<tr>
<td>4</td>
<td>/f1/naack</td>
<td>Edge Bridge offloads call (Cloud subscribes)</td>
</tr>
<tr>
<td>5</td>
<td>/f1/call/retry</td>
<td>Client direct cloud call (Cloud subscribes)</td>
</tr>
</tbody>
</table>

1. Raspberry Pi 4 B
2. Google Cloud Platform VM

Source & Experiments: 
Github.com/OpenFogStack/GeoFaaS
Results

- GeoFaaS’s physical distance impacts response times. GeoFaaS effectively routes clients to the nearest servers.
- GeoFaaS offloads requests for transparent client responses under high load.
- GeoFaaS with reliable DisGB routes to cloud in case of edge (internal) failure.

- Full results in our paper (under review)
References


