DisGB: Using Geo-Context Information for Efficient Routing in Geo-Distributed Pub/Sub Systems

Jonathan Hasenburg and David Bermbach | UCC 2020
Motivating example
With time, our system evolves

- More users + global rollout
- New functionality
  - Real-time data exchange between cars
  - Include data from roadside equipment such as traffic lights

- More data is transmitted to the cloud
- Stricter latency / privacy requirements
- Users are distributed across the globe

A centralized cloud setup might not be the best option anymore
We need to distribute brokers across cloud regions.

High Latency

Low Latency

Broker

Transform
Open questions

• How to distribute events and subscriptions fast (low latency) and efficiently (low excess data) between cloud regions?
Open questions (cont.)

• How to distribute events and subscriptions fast (low \textit{latency}) and efficiently (low \textit{excess data}) between cloud regions?

• Can we use IoT specific domain knowledge for optimizing message flooding?
  
  ➢ IoT devices have a location
  
  ➢ IoT devices often know where their data is relevant / should be accessed
  
  ➢ IoT devices often know where relevant data comes from

Geo-context
Clients publish events with different content
Subscribers have different content interests
But: There is also a geo-context
There can be multiple area of interests
Subscription geofence & publisher location

• Publisher location
  • Different for each publisher
  • Not related to the content of published data
• Subscription geofence (area of interest)
  • Distinct for each subscriber and content interest

➢ Subscription GeoCheck: subscriber limits data distribution
Event geofence & subscriber location
Event geofence & subscriber location (cont.)

- Subscriber location
  - Different for each subscriber
  - Not related to the content of received data
- Event geofence (area of relevance)
  - Distinct for each publisher and content interest

- Event GeoCheck: publisher limits data distribution
Using geo-contexts for matching events

Event is delivered, if:

- It has a certain type of content (Content Check)
- Respective publisher is located inside an area defined by subscriber

  ➢ (Subscription GeoCheck)
Using geo-contexts for matching events (cont.)

Event is delivered, if:

- It has a certain type of content (ContentCheck)
- Respective publisher is located inside an area defined by subscriber
  - (Subscription GeoCheck)
- Respective subscriber is located inside an area defined by publisher
  - (Event GeoCheck)
GeoBroker

- Open-source pub/sub broker implementation
- Combines a topic-based ContentCheck with the two GeoChecks
- There is a paper
  - More details on approach, incl. efficient subscription indexing structure
  - Evaluation of GeoCheck overheads and benefits
Distributed GeoBroker (DisGB)
Execution Environment

Broker $B_1$

Broker $B_2$

Broker $B_3$

Cloud

Edge

B$_1$ Broker Area

B$_2$ Broker Area

B$_3$ Broker Area
Overview

- DisGB extends the GeoBroker approach to support a distributed execution environment
- Key idea:
  - Distribute messages to all brokers to which a matching client can be connected
  - Do not send messages to brokers that cannot pass the GeoChecks
- Two routing strategies that …
  - Optimize event flooding (DisGB_E)
  - Optimize subscription flooding (DisGB_S)
DisGB_E: Optimizing event flooding

- We only distribute events to brokers to which a matching subscriber can be connected
- We only distribute events to brokers whose broker area intersects with the event geofence
  - Only here can subscribers be located that pass the event GeoCheck
DisGB_S: Optimizing subscription flooding

- We only distribute subscriptions to brokers to which a matching publisher can be connected.
- We only distribute subscriptions to brokers whose broker area intersects with the subscription geofence.
  - Only here can publishers be located that pass the subscription GeoCheck.

![Diagram of broker areas and subscription geofences](image-url)
Evaluation
Evaluation

We evaluated DisGB

• Through experiments
  - Attaching geo-contexts to events and subscriptions is feasible in practice
  - Compare effects on latency and excess data dissemination for three IoT scenarios

• Through simulation
  - Compare our strategies to strategies from related work
  - Based on one of the three IoT scenarios from the experiments
Simulation Design

https://moewex.github.io/DisGB-Simulation/
Strategies

- Flooding Events (Flood_E)
- Flooding Subscriptions (Flood_S)
- Consistent Hashing (DHT)
- Grid Quorum (GQPS)
- Broadcast Groups (BG)
Number of inter-broker messages

25 brokers
100,000 clients

256 brokers
100,000 clients
Latency

25 brokers
100,000 clients

256 brokers
100,000 clients
Conclusion

- DisGB uses geo-context information to optimize inter-broker routing
- We proposed two inter-broker routing strategies
  - Achieve the same latency as flooding
  - Require significantly less inter-broker messages.
  - Can only be used when geo-context information is available

When no geo-context information is available, we recommend to
- Use Consistent Hashing to minimize the total number of inter-broker messages
- Use Broadcast Groups to minimize event delivery latency
Mail: jh@mcc.tu-berlin.de