Data management over geo-distributed edge computing infrastructure

Embedded Pervasive Lab
Georgia Tech

Novel applications necessitate edge computing

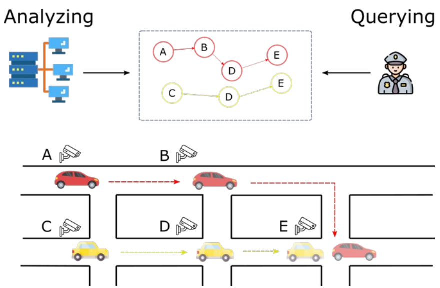
1. Collaborative driving assistance
2. Distributed camera networks
3. Autonomous drone swarms
4. Cognitive assistance

Application requirements

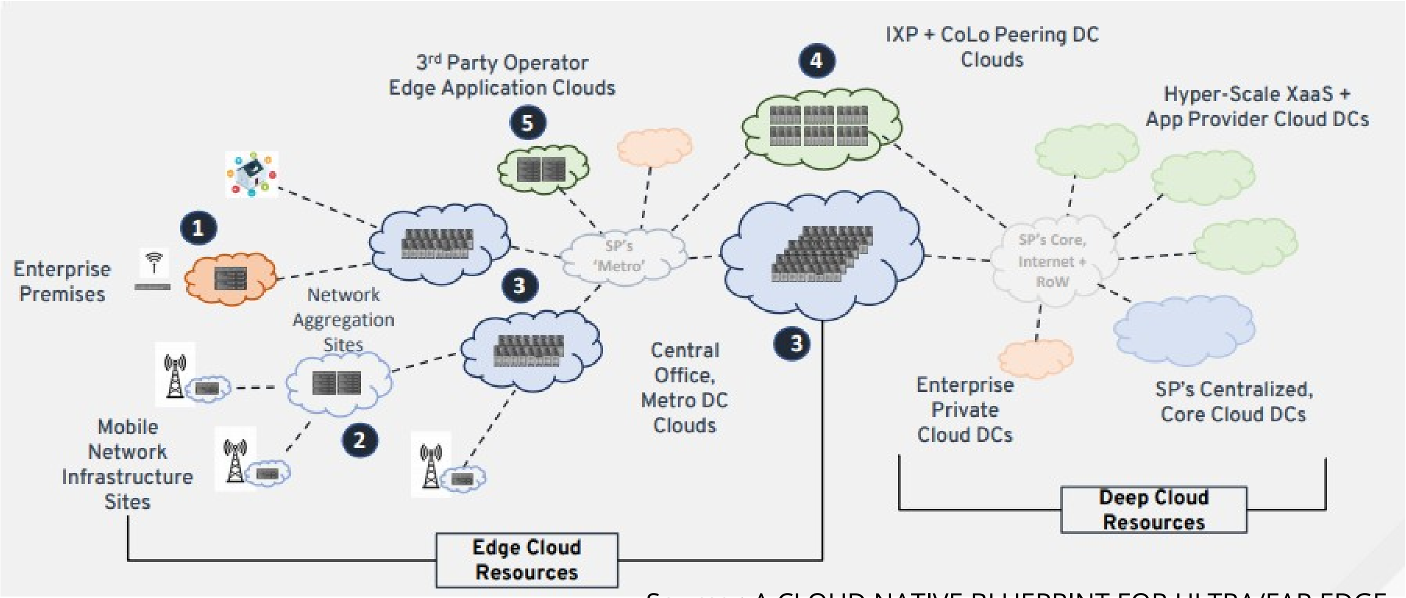
- Geo-distributed clients
- Require low-latency access to compute services
- High-data rate
- Continuous mobility

Source : Qiu, Hang, et al. "Avr: Augmented vehicular reality." Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services. 2018.

http://www.uasmagazine.com/articles/2028/drone-swarm-operations-gaining-traction



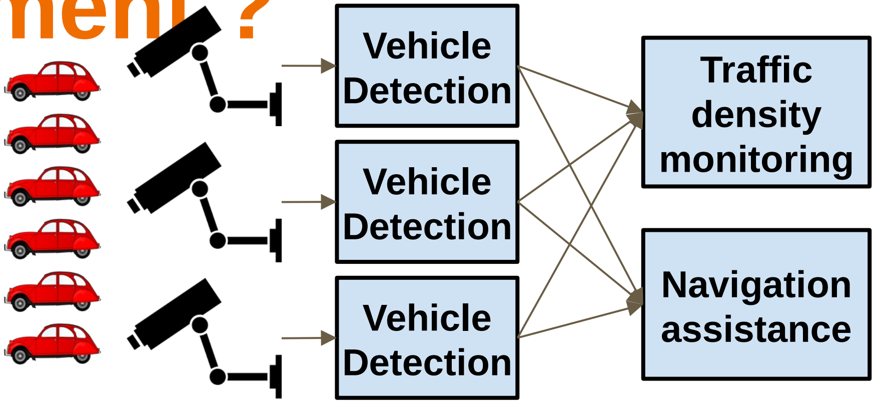
Vision of edge computing



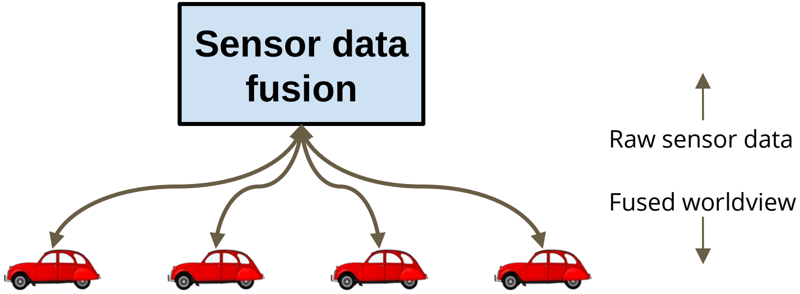
Source : A CLOUD NATIVE BLUEPRINT FOR ULTRA/FAR EDGE
Open Networking Summit

- Continuum of resources
- Could be owned by multiple providers
- Dense geo-distribution
- Limited capacity at each site

What is data management ?



- Situation-awareness applications generate large amounts of data
  - Can be reused by other apps (Nokia streaming as a service)
  - Stored for later retrieval (akin timeseries DBMS)
  - State of processing (as in Flink)
- Data exchange for coordination between connected entities
  - Drone swarms, collaborative driving



Contemporary solutions

Key-value stores

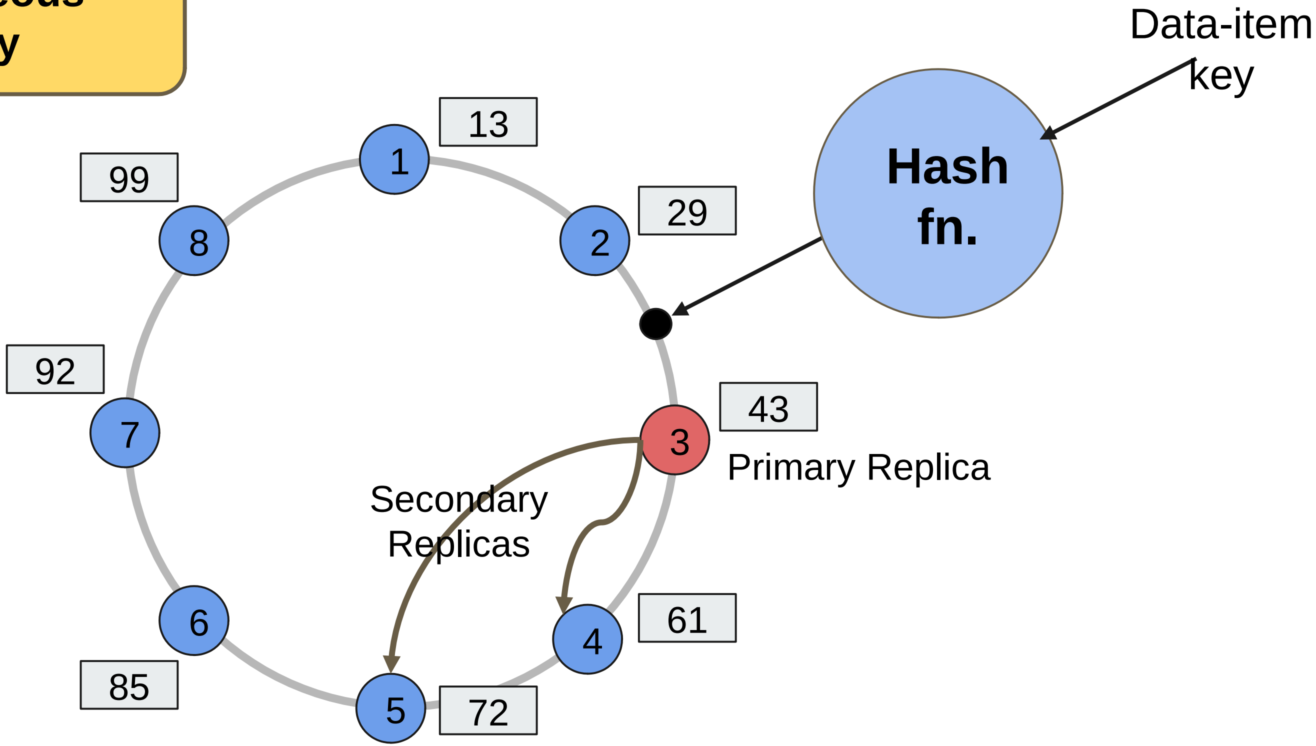
Publish-subscribe systems

Requirements of data management solutions

- Low-latency & high-throughput access to data
- Agility to handle workload dynamism (e.g. client mobility)
- Expose well-known semantics to clients

Peculiar challenges of edge computing infrastructures

Heterogeneous topology



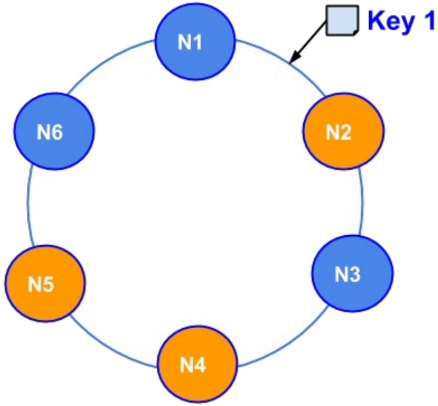
Peculiar challenges of edge computing infrastructures

Heterogeneous topology

Typical data-distribution is topology-agnostic

Dense geo-distribution

Node	DC	RACK
N1	DC1	RACK2
N3	DC1	RACK1
N6	DC1	RACK1
N2	DC2	RACK1
N4	DC2	RACK1
N5	DC2	RACK2



Source : http://distributeddatastore.blogspot.com/2015/08/cassandra-replication.html

Peculiar challenges of edge computing infrastructures

Heterogeneous topology

Typical data-distribution is topology-agnostic

Dense geo-distribution

Coarse-grained geo-distribution support

Client mobility

No support for monitoring client mobility

Edge-friendly data management solutions

We present two systems that meet the above requirements of edge-friendly data management systems

1. Distributed spatio-temporal data store
2. Distributed publish-subscribe system

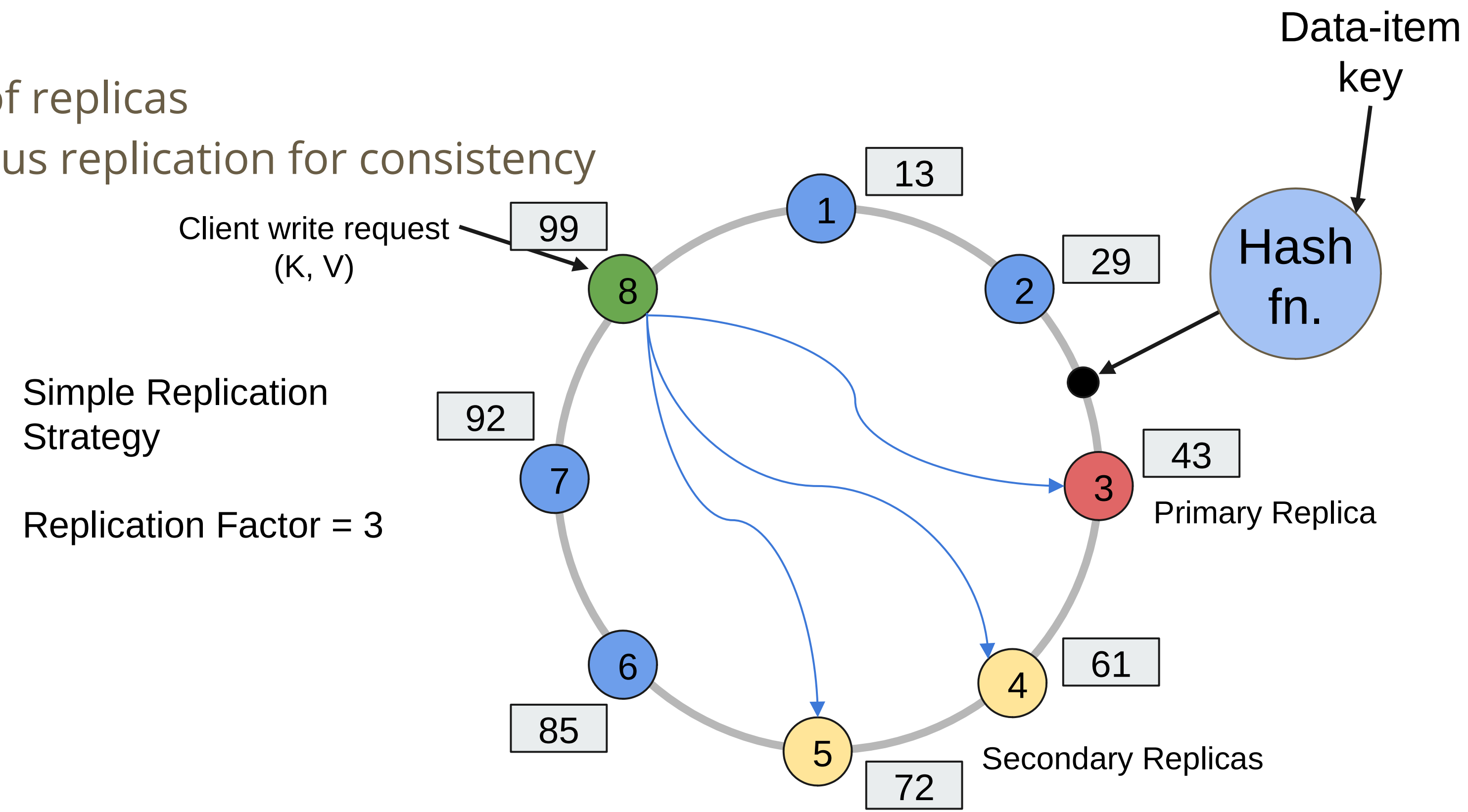
In both cases, data mgmt nodes are deployed on multiple edge sites

DataFog : geo-distributed spatio-temporal K-V store

- User interface
  {	K1 : V1, K2 : V2, ...,
  	loc : (lat, lng),
  	timestamp : ts
  }
- Assumption : Situation-awareness applications
  Low-latency access to data is required by clients in proximity to the location of data-item

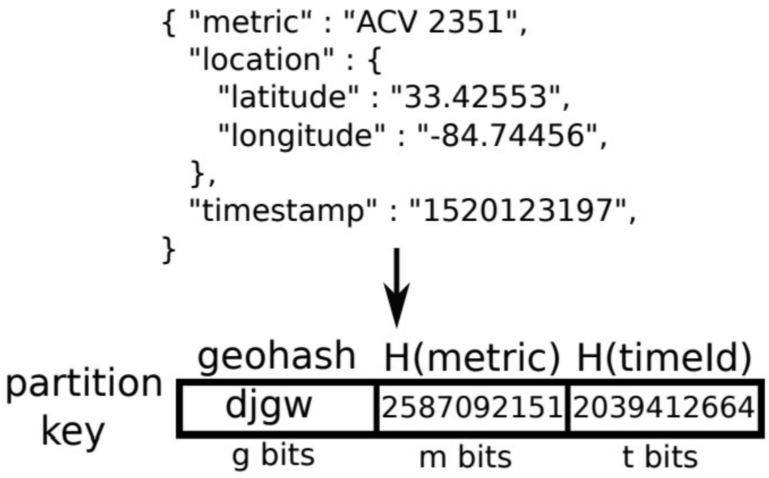
Source of latency in geo-distributed key-value stores

1. Selection of replicas
2. Synchronous replication for consistency



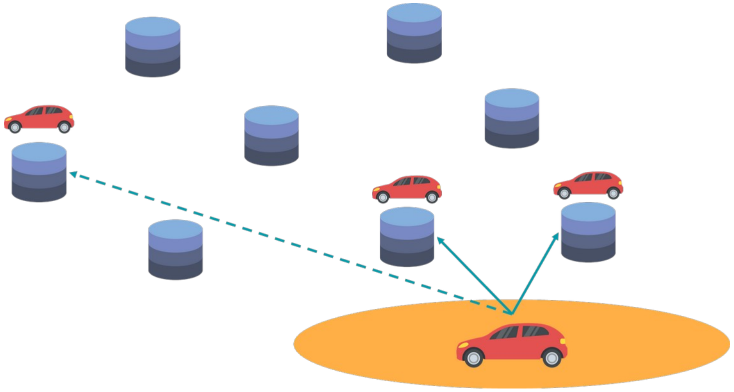
Location-based data partitioning

- Objective : Keep data close to potential users
  - Smart-car interested in congestion information of the same city
- Use of spatial attributes to form data-item's partition key
  - Partition key is used for distributing data items



Tradeoff between latency and fault tolerance

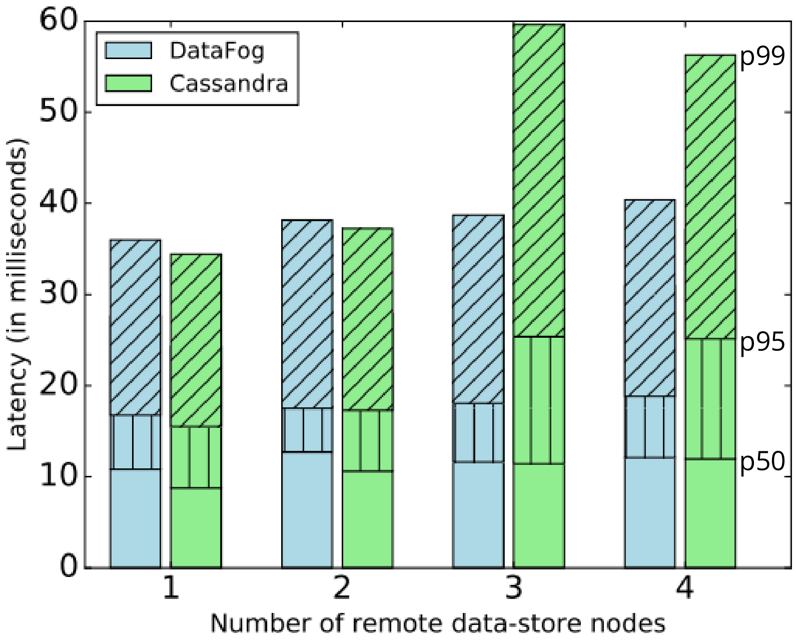
- Utilize spatio-temporal locality in queries
  - E.g. smart car interested in recent events in its vicinity
  - Provide consistent access to spatio-temporally relevant items
- Relevance specified by developer
  - Area-of-interest
  - Specified in terms of geographical distance
- Maintain two types of replicas
  - In-vicinity : Kept strongly consistent
    ⇒ Low latency access
  - Remote : Eventually consistent
    ⇒ Provide fault-tolerance



Area-of-interest size
specified by developer

Performance improvements by locality optimizations

- Infrastructure
  - 4 local nodes within Atlanta, US
  - Variable no. of remote nodes in USA
  - Replication factor = 3
  - Emulation using Containernet
- Query : select all vehicle detections of vehicle X within a 5 kilometre radius that happened less than 10 minutes ago
- Increasing scale of data store results in higher query latencies
  - Data is stored on remote DB nodes
  - For < 3 remote nodes
    - At least 1 (out of 3 total) replicas is local



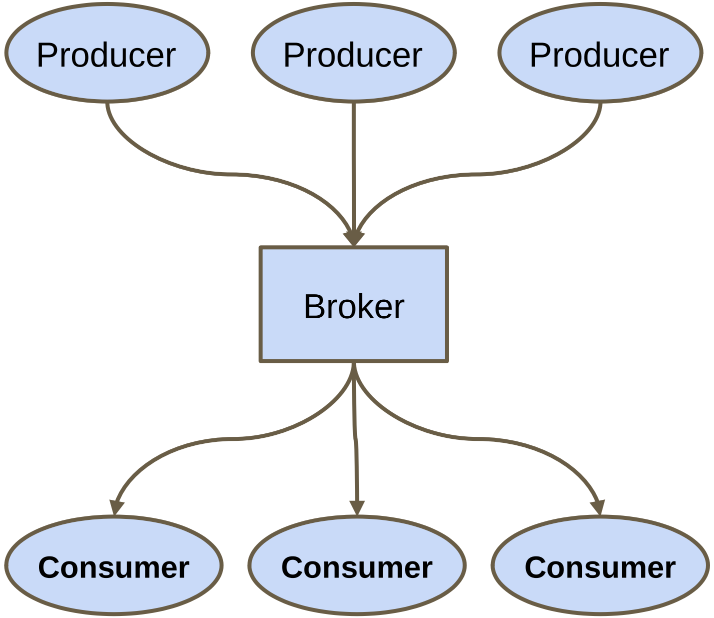
Inter-city latency estimated using
WAN Latency Estimator (https://wintelguy.com/wanlat.html)

ePulsar : Topic-based publish-subscribe system

- Geo-distributed publish-subscribe system
- Follows similar interface as Kafka and Pulsar
  - Producer clients publish to topics
  - Consumer clients consume from topics
  - Brokers host topics

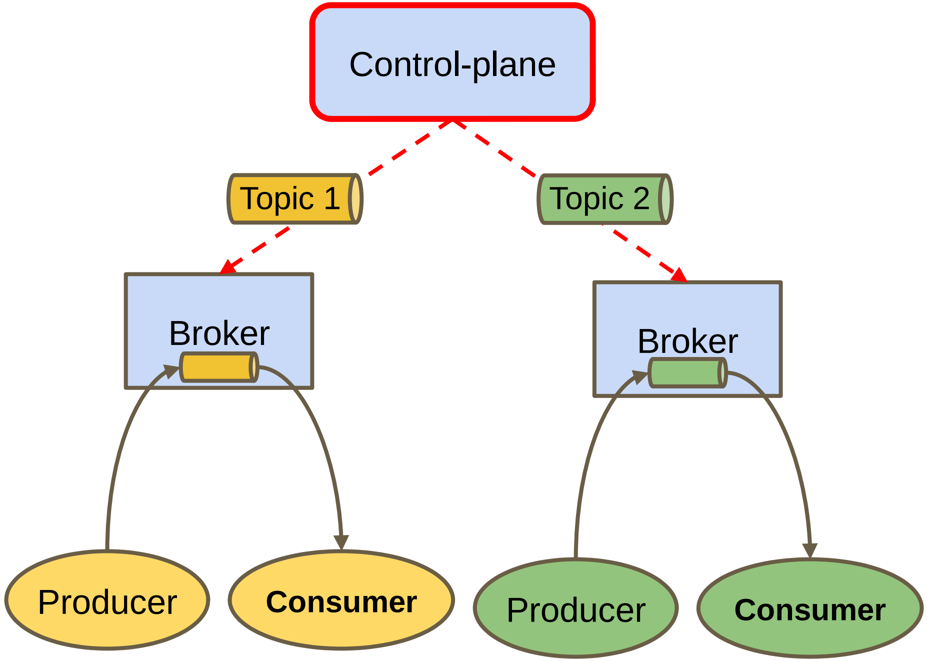
Source of latency in publish-subscribe systems

- Broker selection : topic → broker
- Broker selection done using **consistent hashing**
  - Network topology is not taken into account
- Results in high client-broker network latency



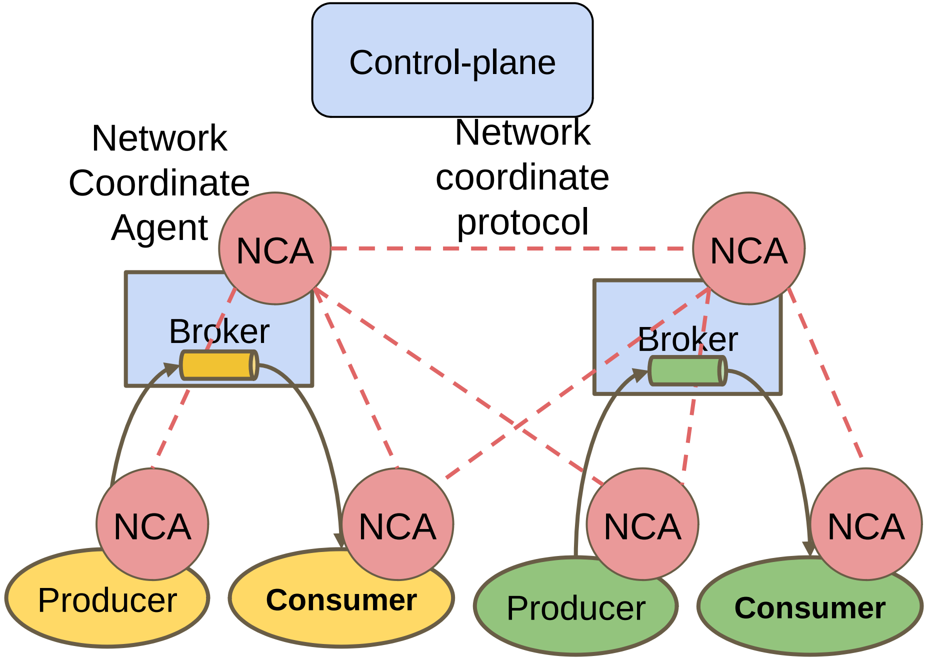
ePulsar's contribution : edge-friendly control-plane

1. Scalable client-broker latency estimation
   a. Network coordinates protocol
2. Latency-aware adaptive broker selection
   a. Aim to minimize worst-case end-to-end latency
   b. Continuous monitoring of broker and clients' network coordinates



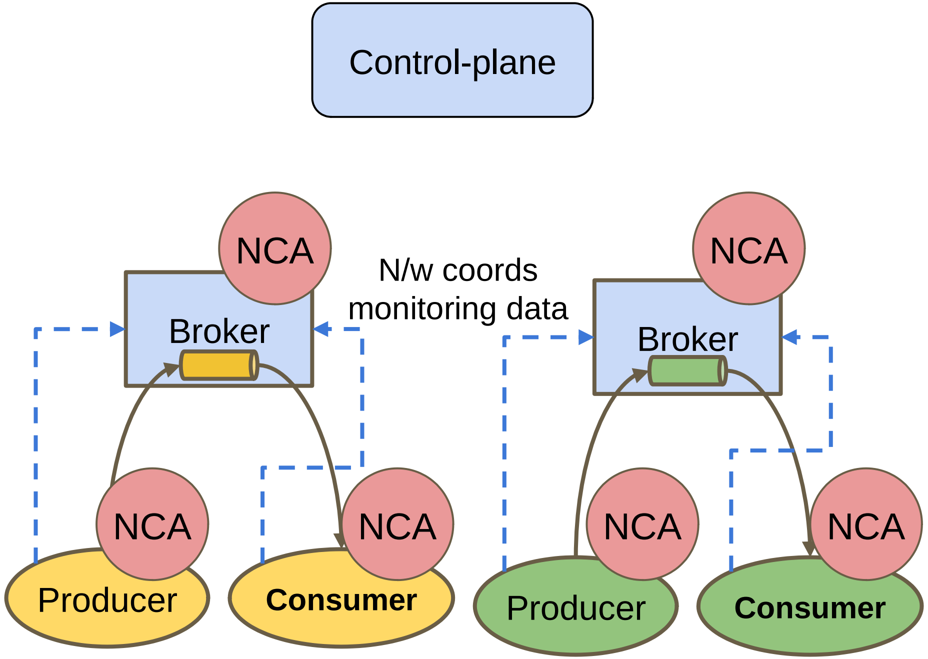
Scalable client-broker latency estimation

- Vivaldi network coordinate agent runs on each pub-sub entity
  - Brokers
  - Clients
  - We use Serf service orchestration tool to provide network coordinates



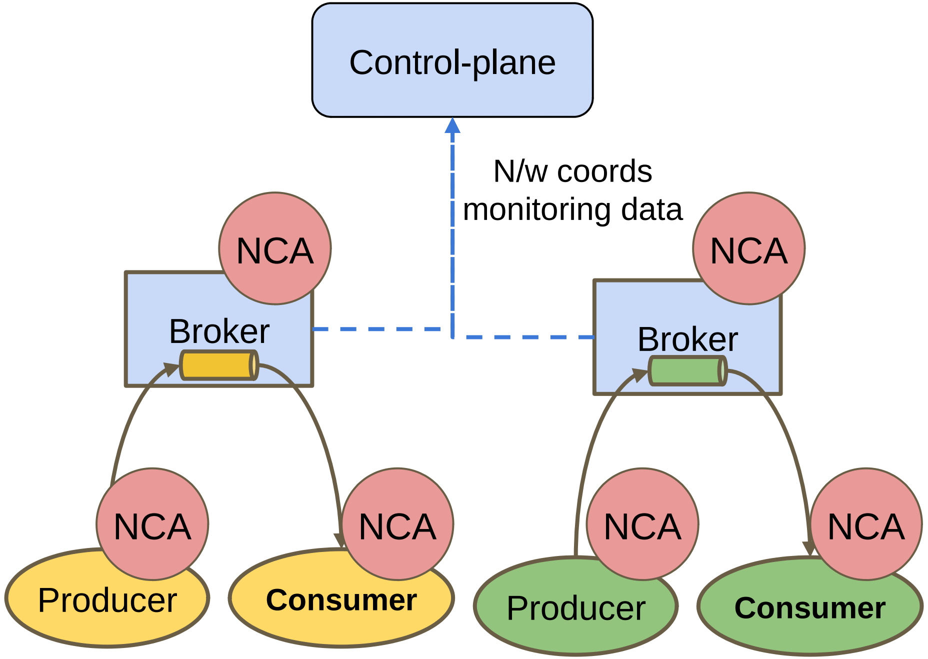
Scalable client-broker latency estimation

- Vivaldi network coordinate agent runs on each pub-sub entity
  - Brokers
  - Clients
  - We use Serf service orchestration tool to provide network coordinates



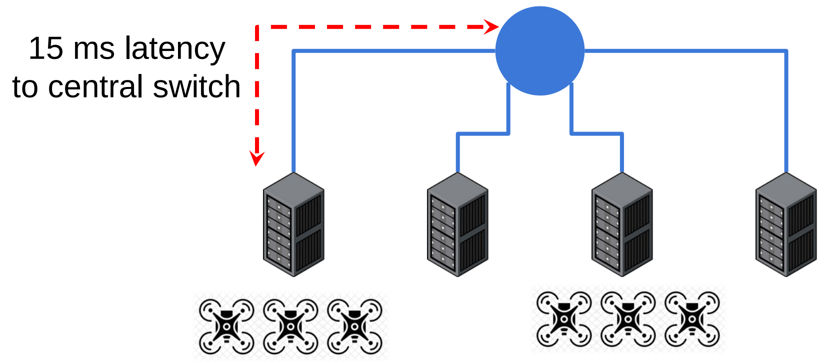
Scalable client-broker latency estimation

- Brokers report monitoring data to control-plane per-topic
  - Worst-case end-to-end latency
  - Aggregate of clients’ network coordinates
  - Traffic load (message rate, size)

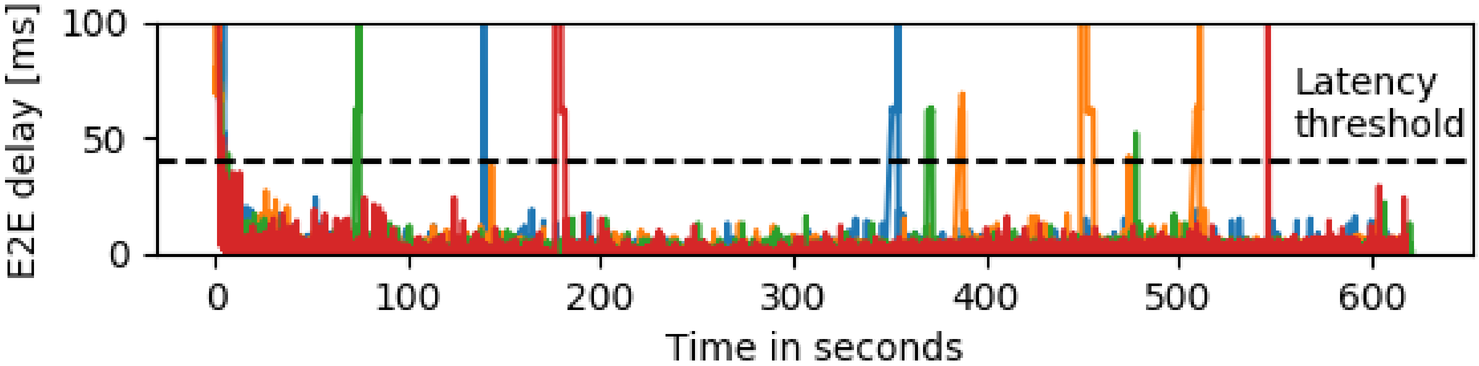


ePulsar performance

- UAV swarm scenario
- Topic 1 : Leader UAV → follower
- Topic 2 : Follower → Leader
- 8 swarms each with 8 UAVs



- City-scale edge infrastructure
  - Multiple sites connected via backbone
- Each swarm moves independently
  - Following Random Waypoint Model
- All UAVs in a swarm move together



Conclusion

- Data management at edge poses peculiar challenges
- We design edge-friendly data mgmt solutions
  - Leverage high-performance data-plane of cloud-based systems
  - Implement edge-friendly control-plane
- We show main design decisions for two kinds of systems
  - Key-value stores
  - Publish-subscribe systems

References

1. **Datafog: Towards a holistic data management platform for the iot age at the network edge.**
   Harshit Gupta, Zhuangdi Xu, and Umakishore Ramachandran
   USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18) 2018

2. **Fogstore: A geo-distributed key-value store guaranteeing low latency for strongly consistent access.**
   Harshit Gupta and Umakishore Ramachandran
   12th ACM International Conference on Distributed and Event-based Systems (DEBS) 2018

3. **STTR: A System for Tracking All Vehicles All the Time At the Edge of the Network**
   Zhuangdi Xu, Harshit Gupta and Umakishore Ramachandran
   12th ACM International Conference on Distributed and Event-based Systems (DEBS) 2018

4. [Under review] **ePulsar: Control Plane for Publish-Subscribe Systems on Geo-Distributed Edge Infrastructure**