ECOLE
POLYTECHNIQUE
DE BRUXELLES

ULB UNIVERSITÉ LIBRE DE BRUXELLES

MobilityDB on Google Cloud Platform

Sid-Ahmed Bouzouidja

Supervisor: Esteban Zimányi

Academic year:
2022-2023

Presentation Outline

- Motivation and Context
- Research and Reviews
- Distributed Database on Google Kubernetes Engine
- Experimental Evaluation
- Future Perspectives
- Discussion

Motivation and Context

Moving Object Data

Trajectory application use cases:

- Trip planning
- Route optimization
- Track pattern identification
- Tracking virus spreading
- …and more.

General format

```
{
  "timestamp": "2023-08-28T15:30:00Z",
  "latitude": 37.7749,
  "longitude": -122.4194,
  "altitude": 10.0,
  "accuracy": 5.0,
  "speed": 20.0,
  "heading": 45.0,
  "device_id": "ABC123"
}
```

Moving Object Data Challenges

Advanced technology

- GPS
- 5G
- IoT sensors
- Edge computing

Unsuitable existing RDBMS:

- No temporal evolution on time
- Inadequate trajectory handling
- No temporal aggregation support

Fast data collection

Historical data keeps increasing

Storage and processing challenges:

- How do we store and process moving object data efficiently ?
- How do we align resources capacity with the increased volume ?
- How do we maintain ideal performance ?

MobilityDB [1] Features

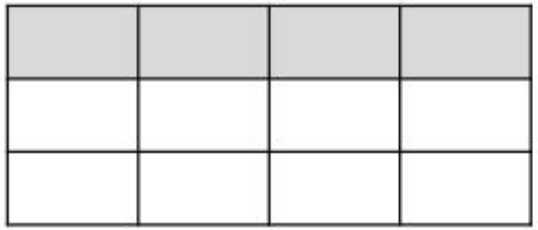
New abstractions:

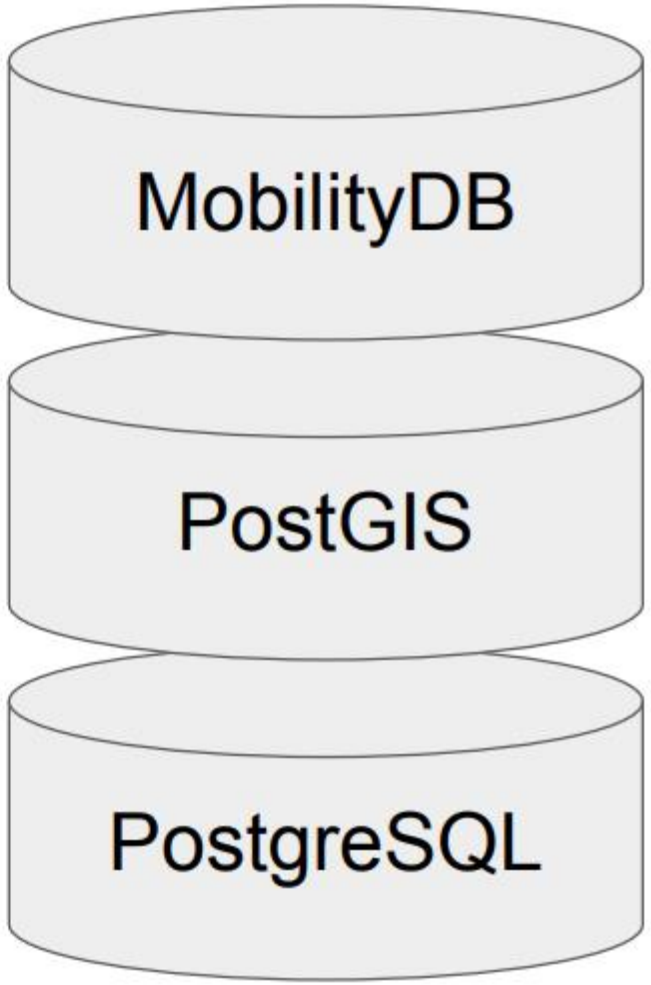
- tint, tfloat, tbool, ttext
- tgeompoint
- timestampset
- periodset
- …and more

New operations:

- Trajectories manipulation
- Temporal properties
- Temporal aggregations

Architecture:





Our missions

Establish a Cloud-Native Environment for MobilityDB

Enable Scalable Distributed Moving Object Database in GCP

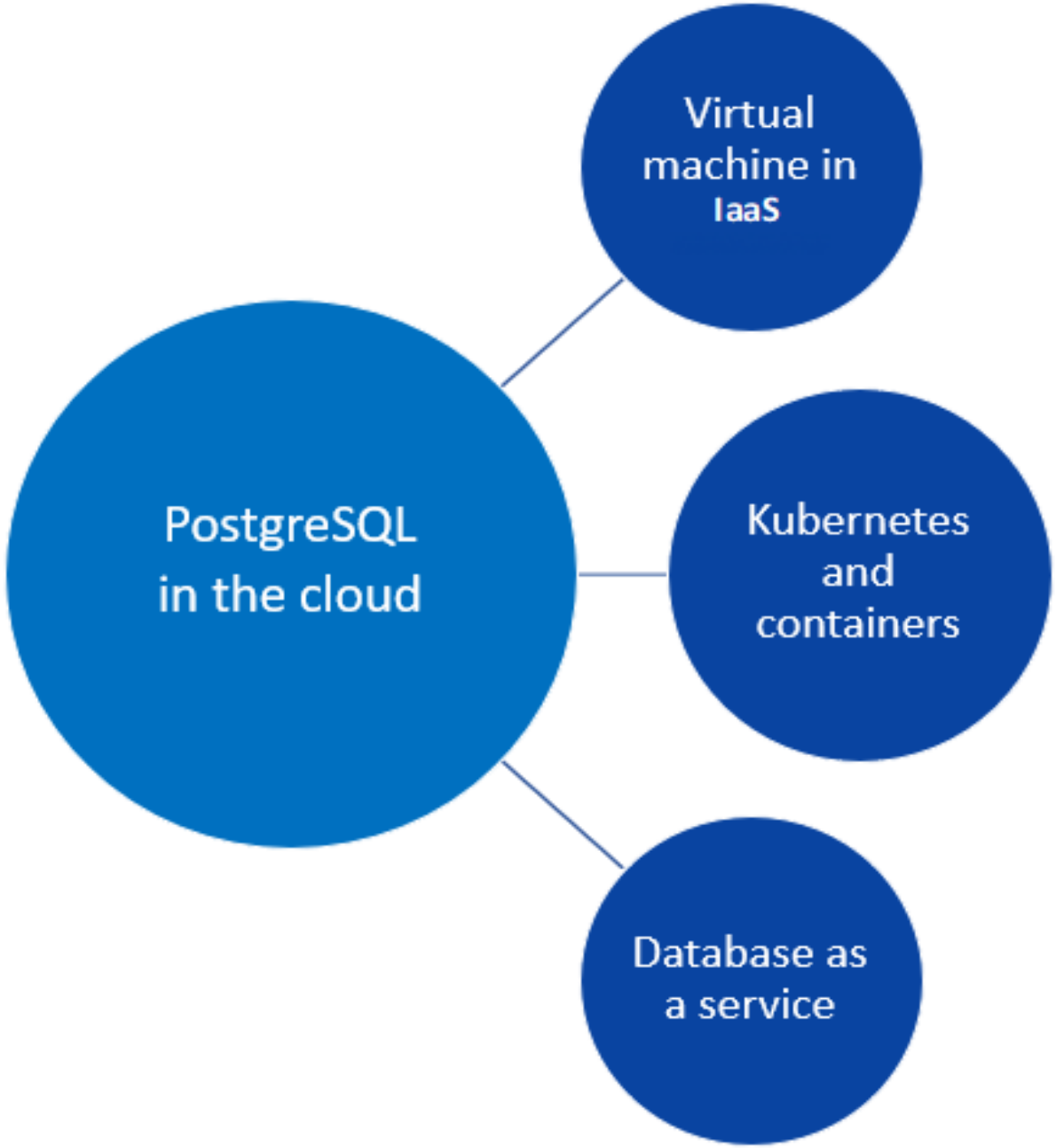
Assessing Feature Compatibility Across AWS, Azure and GCP for MobilityDB

Implementing an API Client that Manages Kubernetes Cluster within GCP

Ensuring Database Integrity During Horizontal Scaling

Research and Reviews

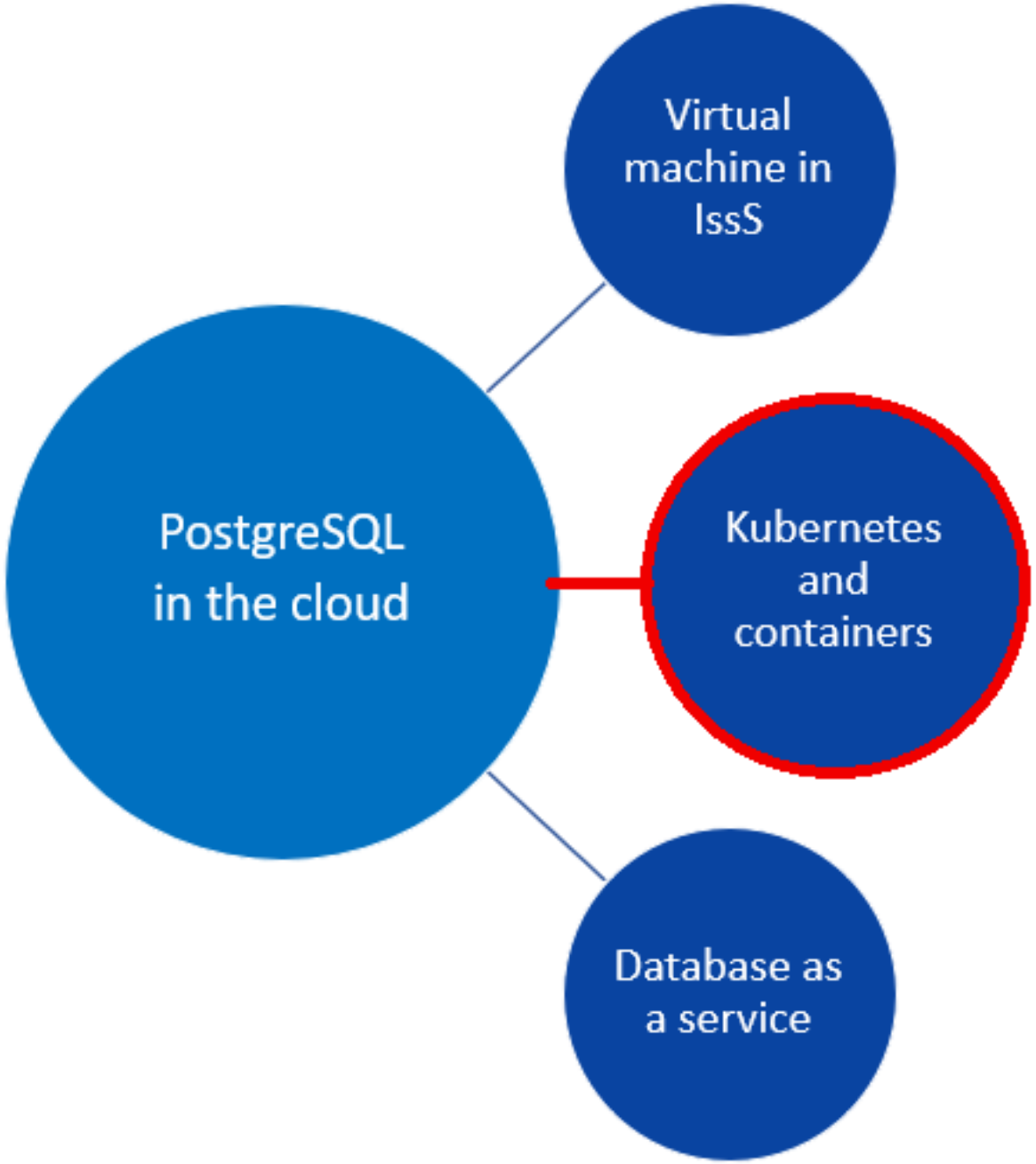
Possible Distributed Environment for PostgreSQL Database in the Cloud

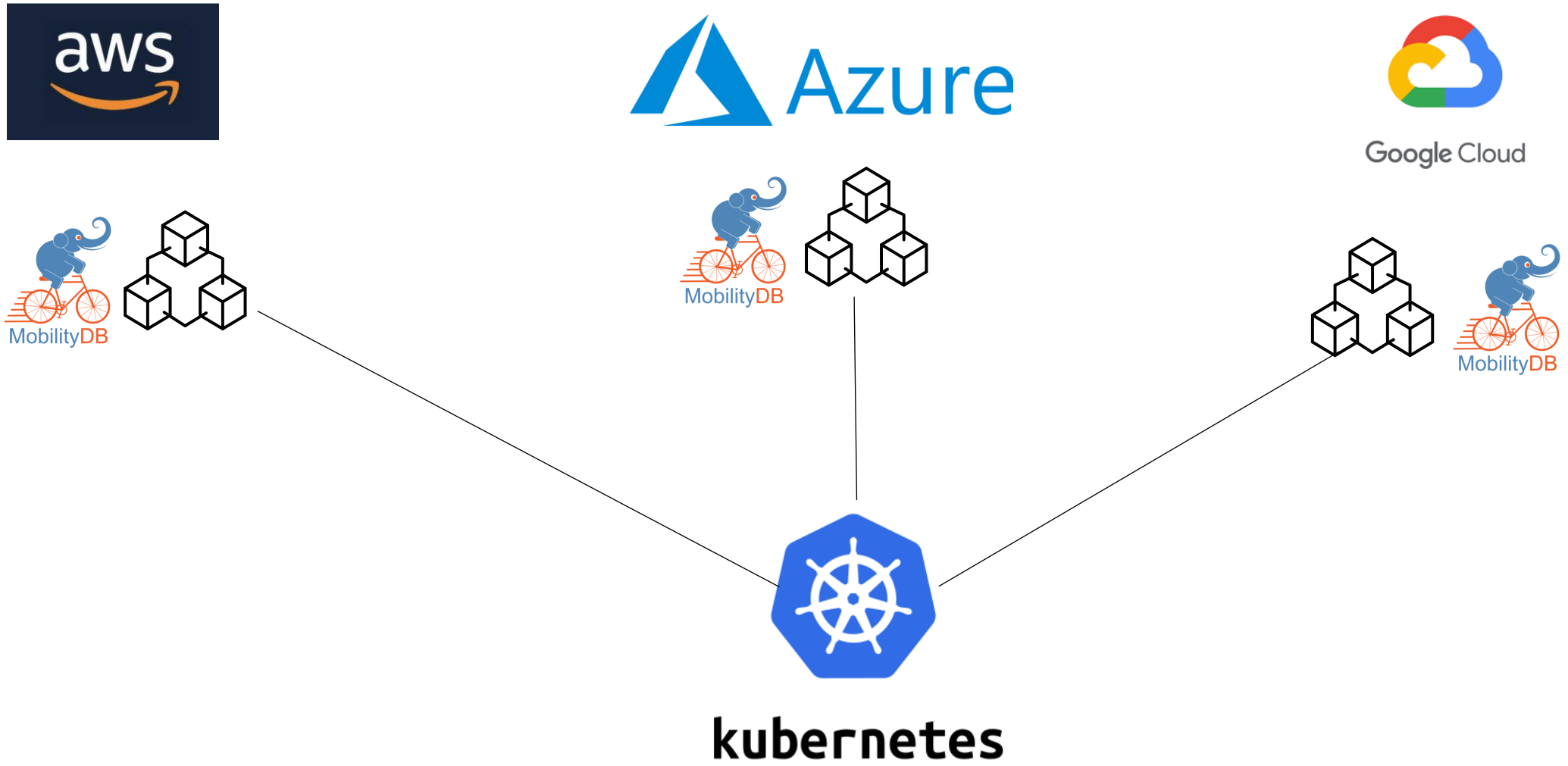


Kubernetes System is the Most Suitable

Reasons why:

- Cluster elasticity
- Features extensibility
- Deployment portability
- Infrastructure as a code

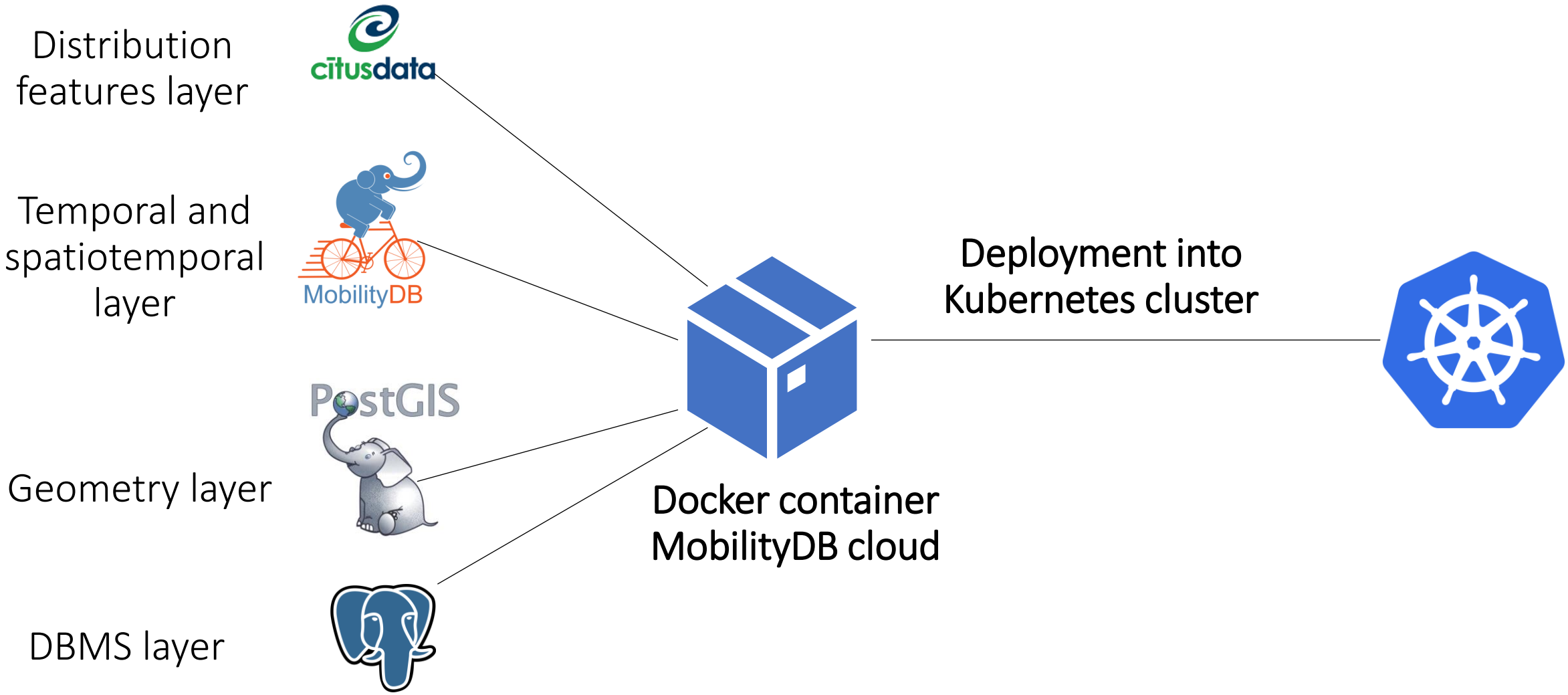




Kubernetes Enables Portable MobilityDB Deployment Across AWS, Azure and GCP

Distributed Database on Google Kubernetes Engine

Distributed Moving Object Database Components



Infrastructure as a Code: examples

```yaml
apiVersion: v1
kind: Service
metadata:
  name: citus-coordinator
  labels:
    app: citus-coordinator
spec:
  selector:
    app: citus-coordinator
  type: NodePort
  ports:
    - port: 5432
      nodePort: 30001
```

Citus coordinator service

```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: postgres-pv-claim-coordinator
  labels:
    app: citus-coordinator
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 5Gi
  storageClassName: standard-rwo
```

Citus coordinator volume

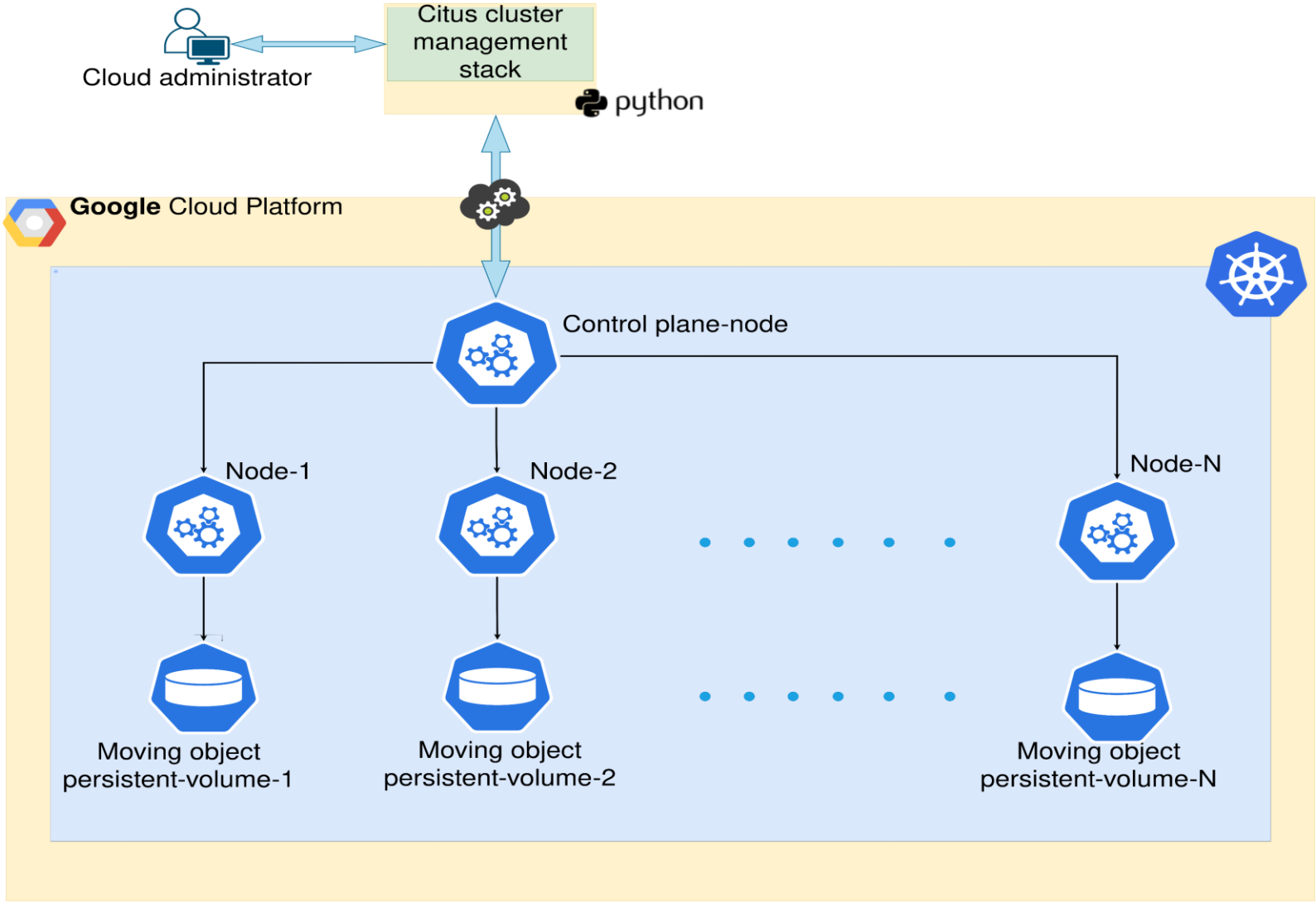
```yaml
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: citus-workers
spec:
  serviceName: "citus-workers"
  replicas: 3
  selector:
    matchLabels:
      app: citus-workers
  template:
    metadata:
      labels:
        app: citus-workers
    spec:
      containers:
      - name: mobilitydb-cloud-worker
        image: bouzouidja/mobilitydb-cloud:latest
        ports:
        - containerPort: 5432
```

Citus worker StateFulSet

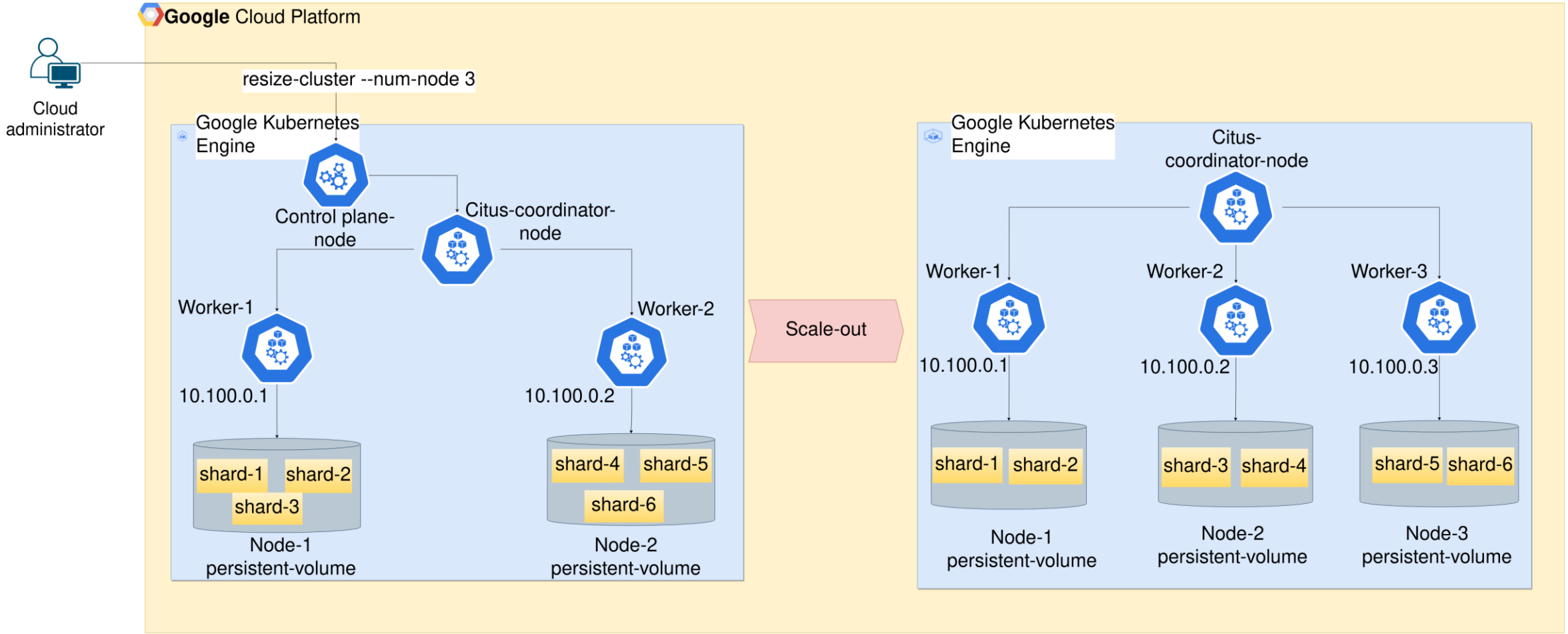
Google Kubernetes Cluster

Citus cluster management stack Key roles:

- Automating deployment tasks
- Allowing horizontal scaling (scale-out/scale-in)
- Ensure distributed database consistency after (scale-out/scale-in)



Scale-out Operation in Citus Cluster



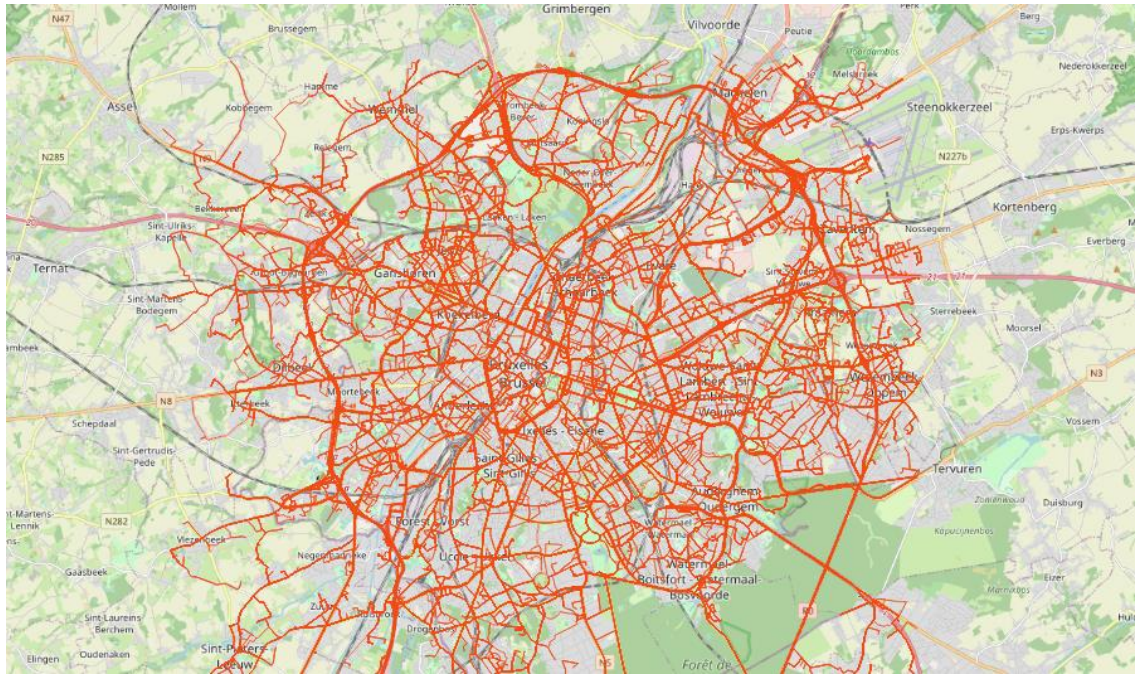
Experimental Evaluation

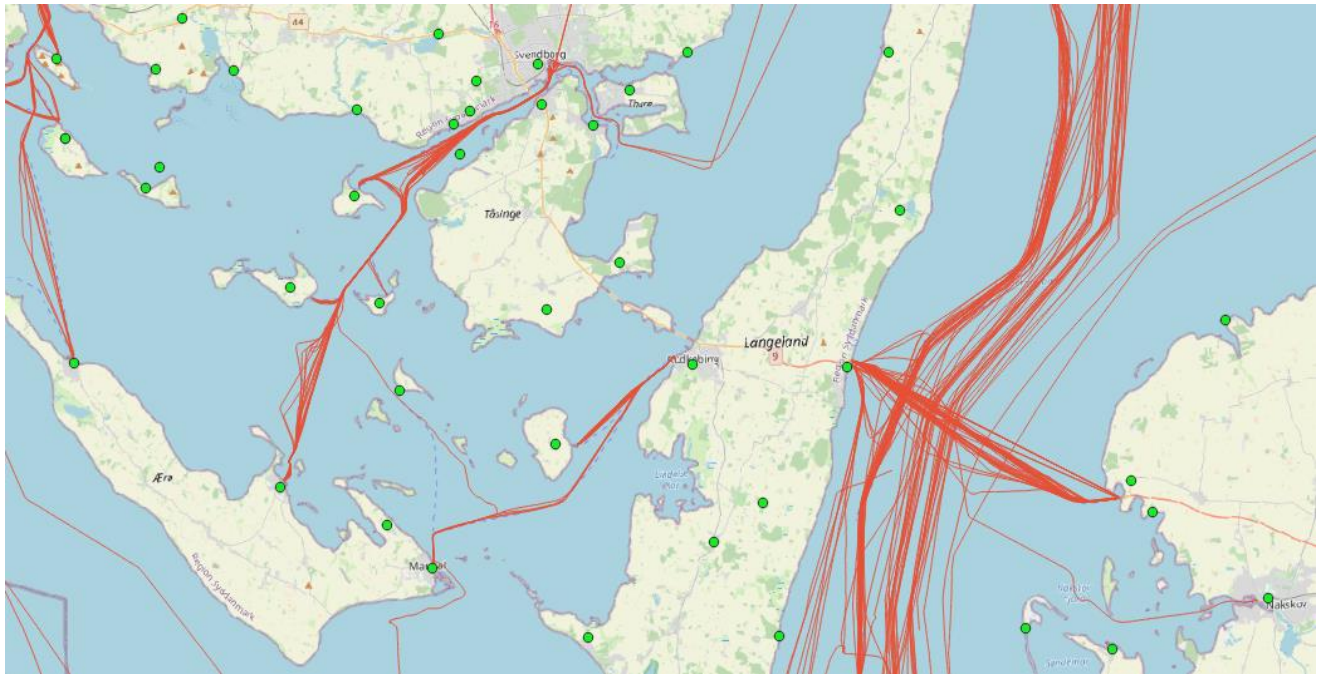
BerlinMOD Benchmark [2]

- Generates scaled moving object data
- Cover diverse queries scenarios
- Assesses varied query complexities

Real-World Benchmark: AIS Data from Danish Maritime Authority [3]

- More realistic scenarios
- Data generated from live application
- Reflects dynamic and authentic movement patterns

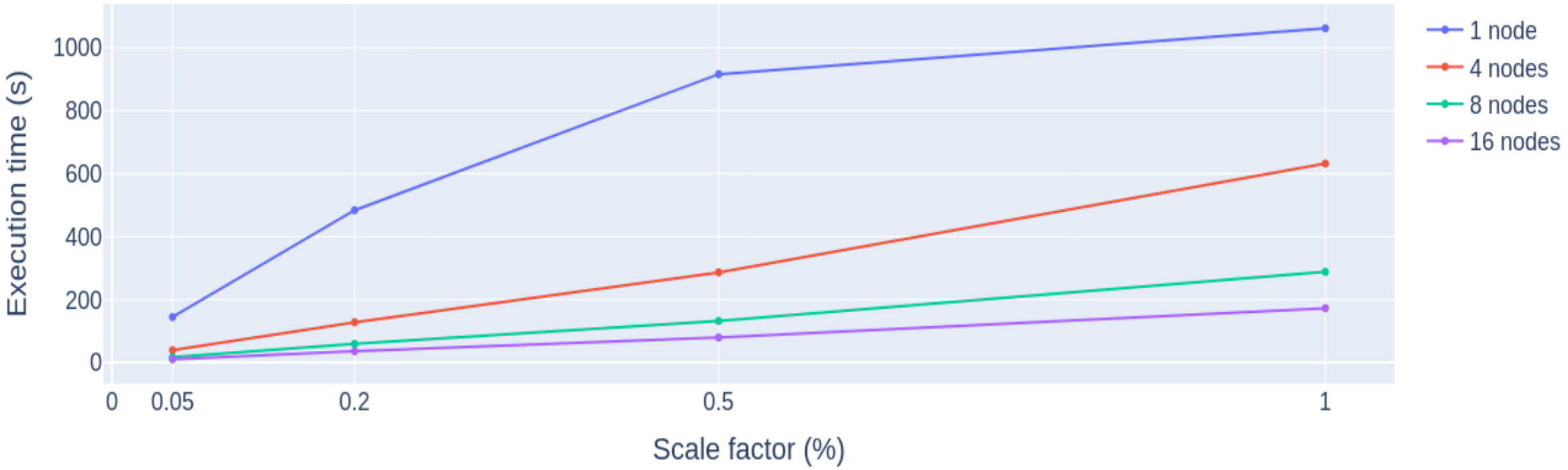




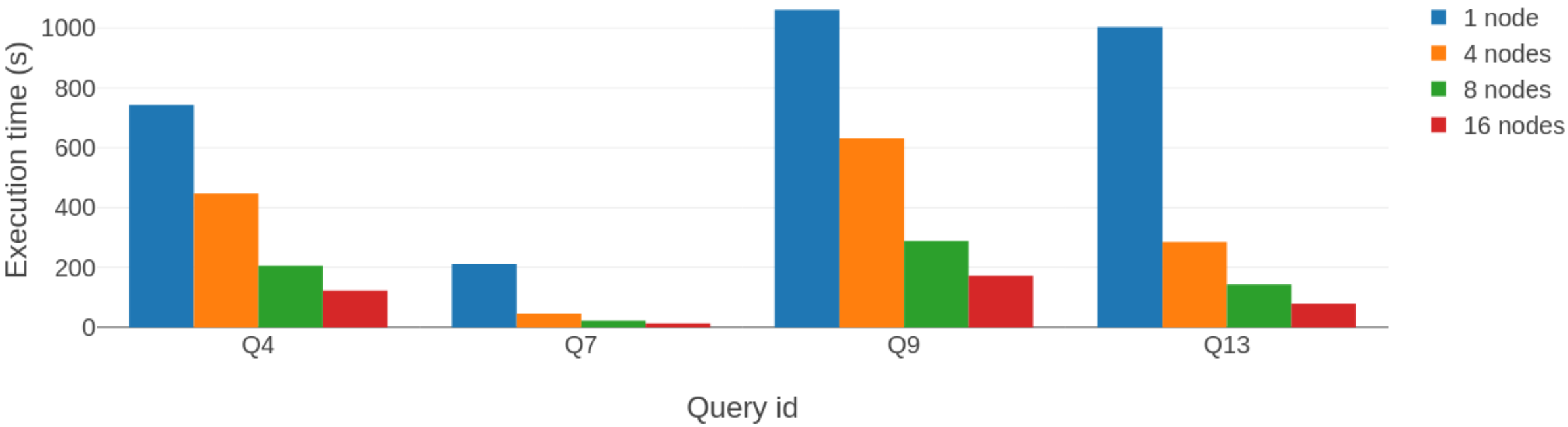
Benchmarks Environments

	VM type	CPU	RAM	Disk	Cluster size	Region/Zone
cluster-config1	Intel(R) Core(TM)	5 cores	6 GB	SSD	Single	Local machine
cluster-config2	e2-standard-4	4 cores	8 GB	Balanced	4 nodes	europe-west1-c
cluster-config3	e2-standard-4	4 cores	8 GB	Balanced	8 nodes	europe-west1-d
cluster-config4	e2-standard-4	4 cores	8 GB	Balanced	16 nodes	europe-west1-c

BerlinMOD Query 9 [4] Execution Time Grouped by Database Scale and Cluster Size (scale-up)



BerlinMOD Query 4, 7, 9 and 13 [4] Execution Time Grouped by Cluster Size for Scale Factor 1



AIS Query Example

Schema

```
Trips(mmsi integer, trip tgeompoint,
    SOG tfloat, COG tfloat)
Periods(periodId integer, period tstzspan)
Ports(pid integer,latitude double precision,
    longitude double precision, Geom Geometry)
```

Query 1 description

Identifying nearby ships within a 500-meter radius of a port

```
SELECT T.mmsi, P.code, P.description
FROM ports P, trips T
WHERE ST_Intersects(trajectory(T.trip),
    ST_Transform(ST_MakeEnvelope(P.longitude,
    P.latitude, P.longitude+0.001409,
    P.latitude+0.001409, 4326), 4326));
```

AIS queries execution time grouped by cluster size (speed-up)

Future Perspectives

- Improve the capabilities for Multi-cloud integration
- Implementing a robust scaling mechanism
- Enhancing the Python API client
- Engage in extensive large-scale experimentation

Video Tutorial

Discussion

References

[1] Esteban Zimanyi, Mahmoud Sakr, and Arthur Lesuisse. Mobilitydb: A mobility database based on postgresql and postgis. ACM Trans. Database Syst., 45(4), dec 2020.

[2] Christian Duntgen, Thomas Behr, and Ralf Hartmut Guting. Berlinmod: a benchmark

for moving object databases. The VLDB journal, 18(6):1335-1368, 2009.

[3] Danish Maritime Authority, AIS Data, https://dma.dk/safety-at-sea/navigational-information/ais-data

[4] Esteban Zimanyi. Berlinmod benchmark on mobilitydb. https://docs.mobilitydb.com/MobilityDB BerlinMOD/master/mobilitydb-berlinmod.pdf, 2022.