Vincenzo Norman Vitale

Tutor: Professor Sergio Di Martino

XXXIV Cycle - III year presentation

# Towards Cost-Performance Awareness in Fog-based Data Management Architectures

Università degli Studi di Napoli Federico II

# Template

- Presentation CONTENT (20 minutes)
  - Overall perspective (10 minutes, dissemination style for the general audience)
    - Cover
    - Your background
      - Graduation MS, DIETI group, cooperations (mostly written)
      - Type of fellowship
    - Credits summary
      - Credits (table, mark in red if discrepancies occurs with PhD web site table)
      - Specific objects (say)
      - Experience abroad
      - Table for training (credits) no words
    - Your problem (general perspective)
      - Relevance
      - Approach
  - Specific activity (10 minutes, conference style for the experts)
    - Your problem (your specific activity)
      - idea, methodology, developments, results, validation
    - Your products
      - Pubblications, Patents, List and mention

- Question time (10 minutes)
  - To support your replies, prepare some spare, very specific, slides

# Background

- MS degree in Computer Science, University of Naples Federico II

- Fellowship: “Fellowship: “*Industry 4.0: Storing, Retrieving and Mining sensor data for Predictive Maintenance*” supported by AvioAero a GE Aviation Business

# Credits Earned

| | Credits year 1 | | | | | | | | Credits year 2 | | | | | | | | Credits year 3 | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | | | 1 | 2 | 3 | 4 | 5 | 6 | | | 1 | 2 | 3 | 4 | 5 | 6 | | | |
| | **Estimated** | bimonth | bimonth | bimonth | bimonth | bimonth | bimonth | **Summary** | **Estimated** | bimonth | bimonth | bimonth | bimonth | bimonth | bimonth | **Summary** | **Estimated** | bimonth | bimonth | bimonth | bimonth | bimonth | bimonth | **Summary** | **Total** | **Check** |
| **Modules** | **20** | | 1,2 | 3 | 3 | 0 | 6 | **13,2** | **15** | 1,8 | 0 | 0 | 13 | 4 | 0 | **19** | **5** | 0 | 3 | 0 | 0 | 0 | 0 | **3** | **35** | **30-70** |
| **Seminars** | **5** | 0,4 | 0,2 | 0 | 0,8 | 0 | 1 | **2,4** | **10** | 0 | 0 | 0 | 2 | 0 | 0 | **2** | **6** | 1,9 | 1,3 | 0,2 | 0 | 0,8 | 0,6 | **4,8** | **9,2** | **10-30** |
| **Research** | **35** | 8 | 8 | 6 | 10 | 6 | 8 | **46** | **50** | 8 | 10 | 9 | 2 | 10 | 10 | **49** | **50** | 8 | 5,7 | 10 | 10 | 9,2 | 9,4 | **52** | **147** | **80-140** |
| | **60** | 8,4 | 9,4 | 9 | 13,8 | 6 | 15 | **62** | **75** | 9,8 | 10 | 9 | 17 | 14 | 10 | **70** | **61** | 9,9 | 10 | 10 | 10 | 10 | 10 | **60** | **192** | **180** |

# Collaborations

- AVIO AERO
- IVM
  - Octopus Project

# Experience Abroad

- L3S research center, Leibnitz University Hannover(Germany), interrupted because of CoViD-19.

# Training Activities

- Tutoring and supervision of Bachelor's Master's degree thesis activities in the Degree Commissions.
- Teaching activity:
  - Ingegneria del Software I
  - Ingegneria del Software II
  - Object Orientation
  - Basi Di Dati II
  - Tecnologie Web

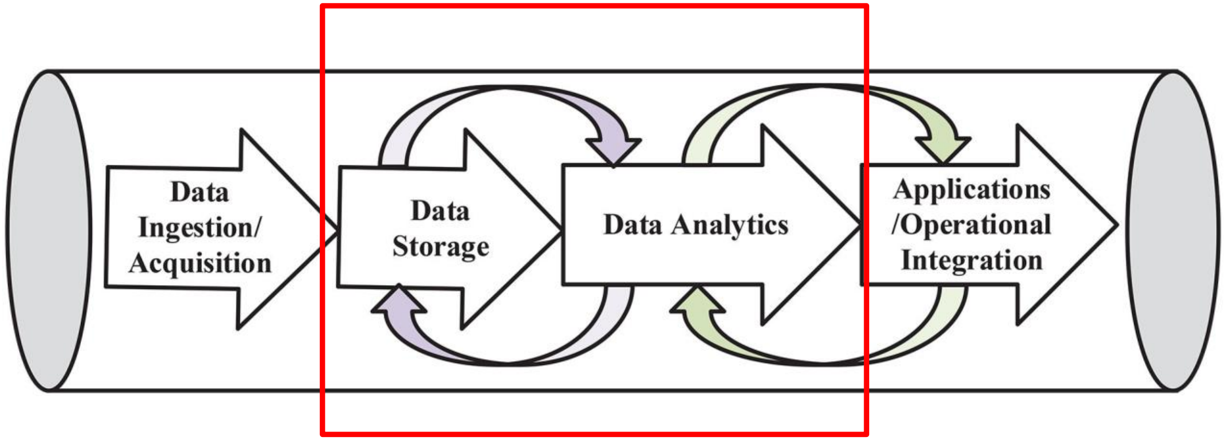
# The Approached Problem

# Massive IoT Data Management

- According to various studies the number of IoT devices will dramatically increase in the next decade.
- Cisco estimated that in 2023 more than 5 billion people will use an internet connection, with an average of 3.6 connected devices per person that could potentially produce and consume data [1].
- IoT has been identified as one of the main sources of Big Data.

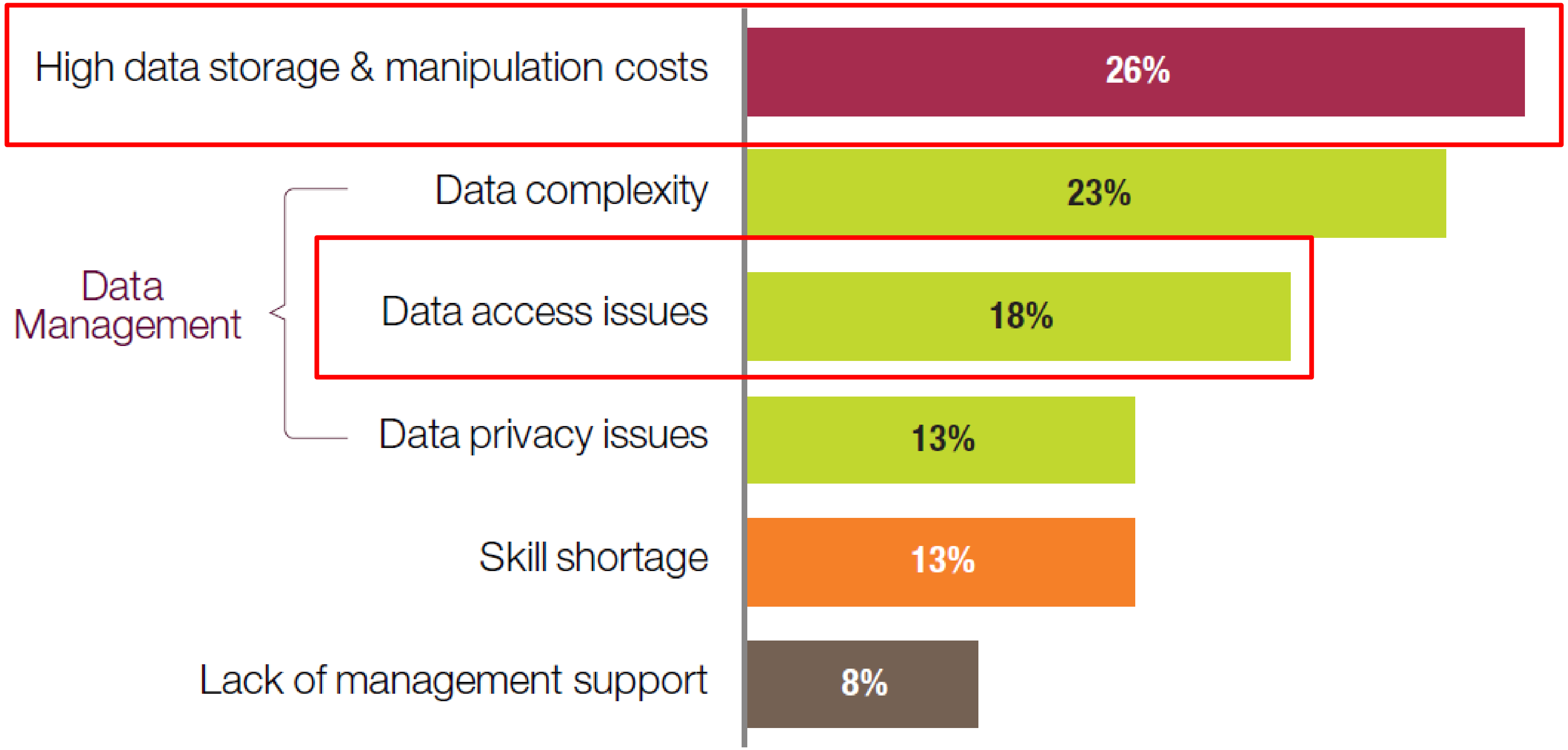
# Big Data & IoT

- IoT devices small or no computing capabilities.
- Storage and Analytics mostly on Cloud.




Key Stages of Big Data Analytics [22]

# Issues Limiting Big Data and Analytics




Top challenges in implementing IoT and Big Data Analytics in Industry 4.0 [2]

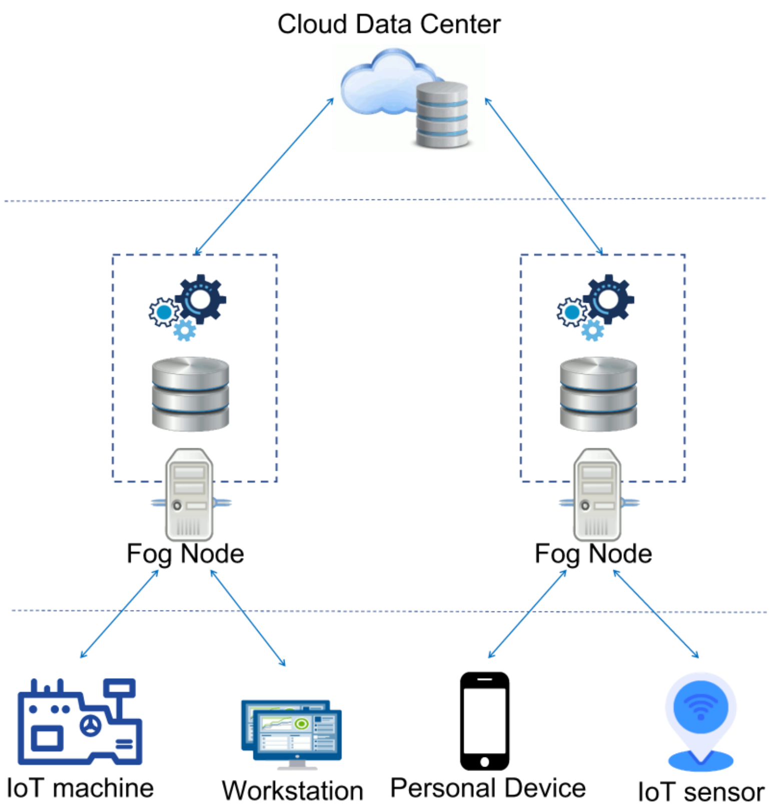
# The Approached Problem

Cost-effective data management in Cloud-centric architectures for massive (spatial) time-series.

- Very high frequency Time Series from Industrial IoT (IIoT).
- High frequency Spatial Time Series from Smart Cities.

# Actual Data Management Architectures

- **Cloud** centralized storage hosting production DBMSs and long-term backup.
- **Fog** support layer(s) to overcome Cloud limits, i.e., response delay [20].
- **IoT/Edge** producing and consuming data streams, i.e., Time-series.



# Research Questions & Contributions

# Research Questions

1. Are COTS DBMSs, on single nodes, capable of providing adequate and reliable performance for the ingestion and recovery of massive data streams?
2. How much do the characteristics of the Fog node impact of the performance of data management architecture?
3. Can Fog layer’s location-awareness and analytical workloads’ knowledge improve the cost-performance ratio of a multilayered data management architecture?

# Contributions

- An extended study on Commercial Off The Shelf (COTS) DBMSs in managing IoT multidimensional data
- Assessment of  COTS DBMSs combined with an empirical assessment through real-world datasets.
- A hybrid Cloud-Fog architecture focused on both cost and performance effectiveness.

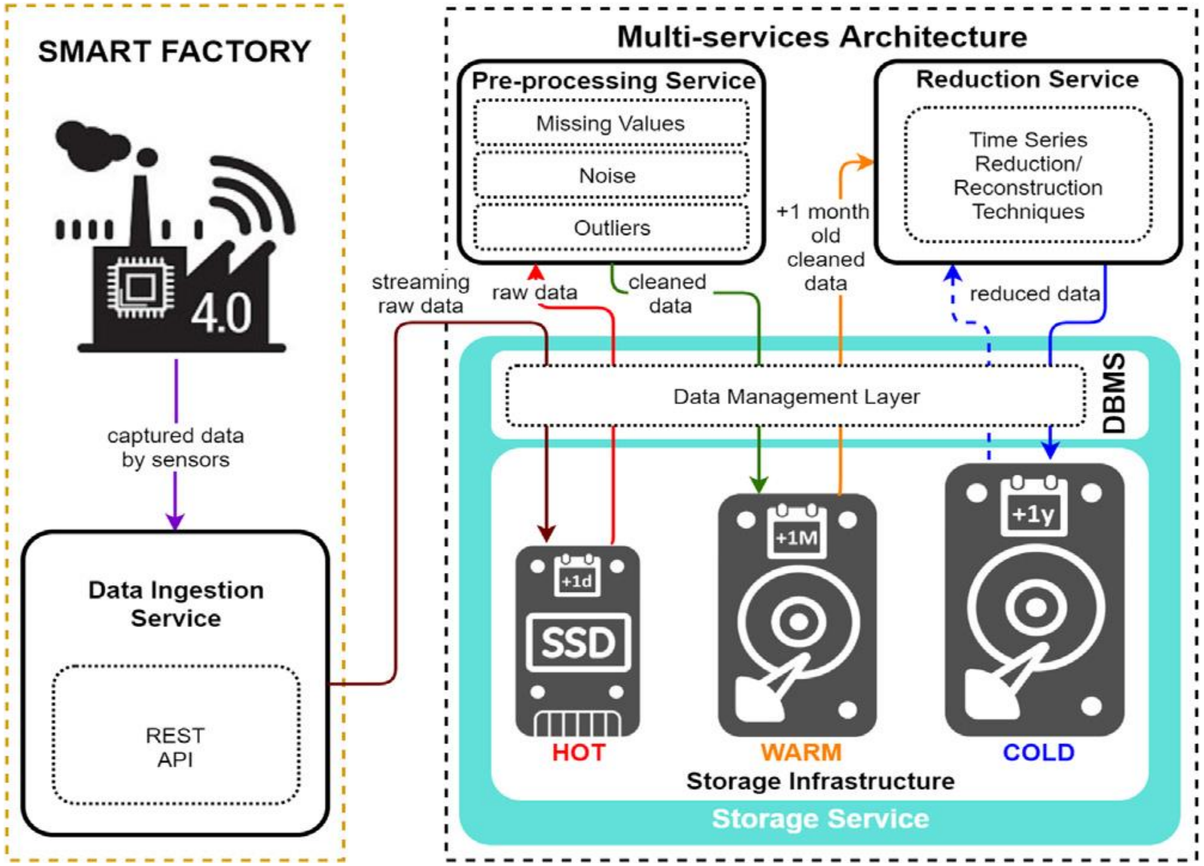
# Contributions

- An extensible tool aimed at evaluating Fog Storage Offloading approaches in hybrid Cloud-Fog environments.
- A metric that integrates those currently used for the cost-benefit evaluation of a multilevel architecture.

# Background & Related Works

# Smart Factories

- Cloud Central Storage[4]
- Fog:
  - Local storage and processing [3]
  - Transparent caching [5]

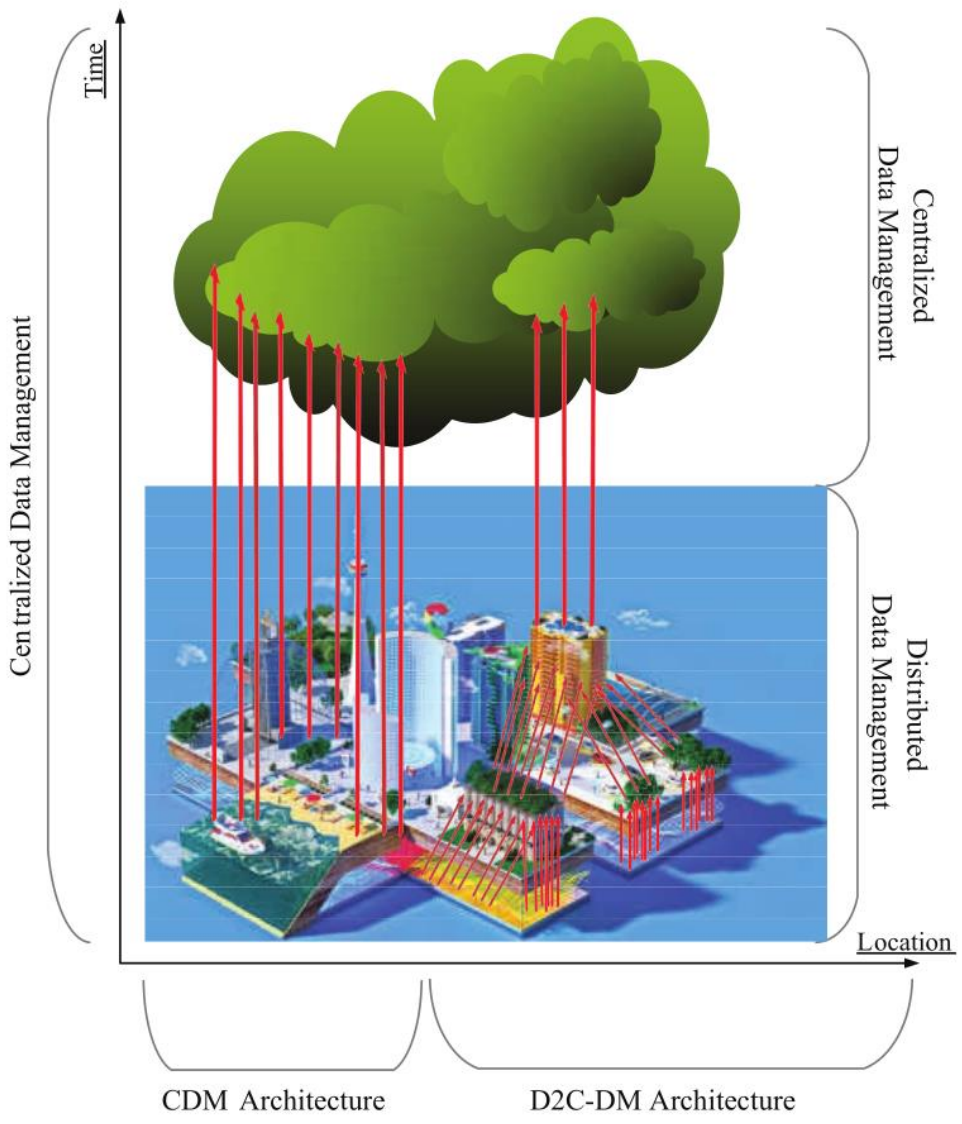



Cloud-centered storage for cost savings [4]

# Smart Cities

Deployed Architectures[6,7,8]

- Centralized Data Management (CDM) [6]
- Distributed to CDM (D2CDM):
  - Fog to cloudlet to CDM (F2c2CDM)[7,8]

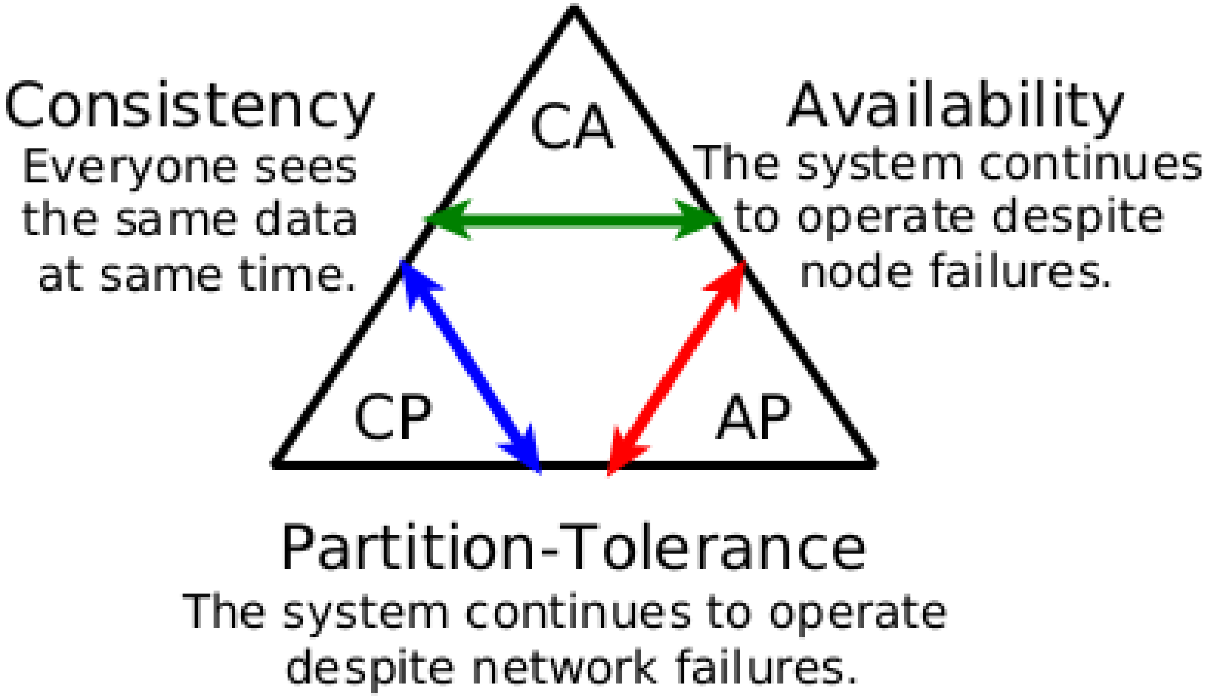



CDM and D2CDM overview [7]

# Storing & Retrieving (Spatial) Time-Series

DBMSs employed in (spatial) time-series storage:

- NoSQL (mostly CP,AP)[9]
- TSMS (any) [10]
- RDBMS (CA)

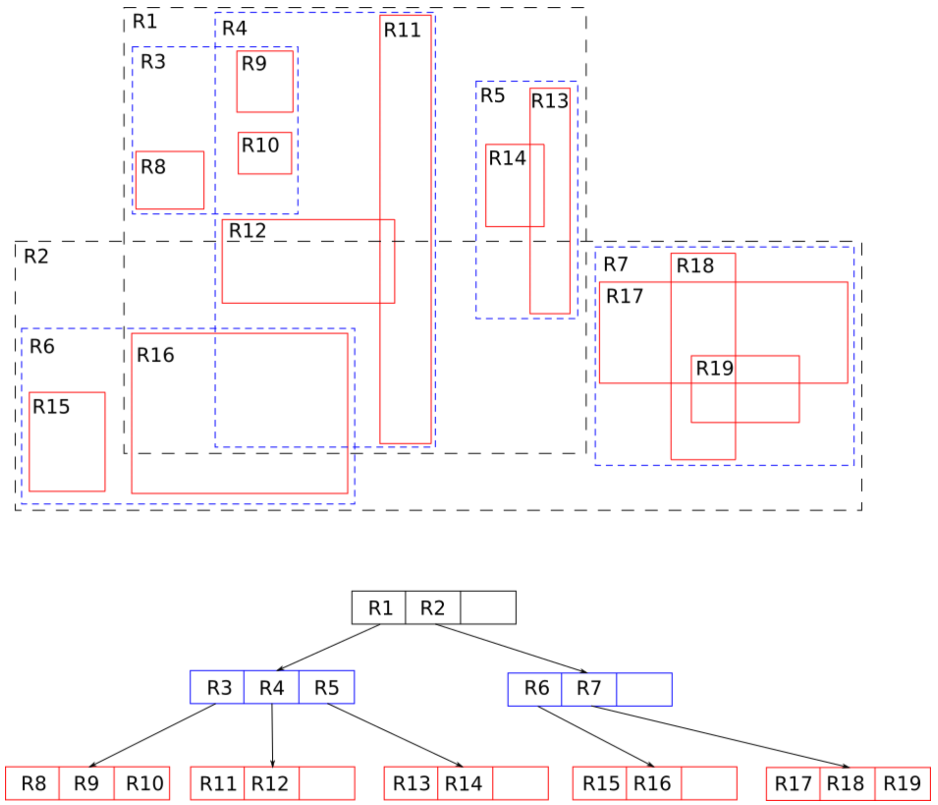



Brewer's CAP theorem[11] DBMS classification

# Storing & Retrieving (Spatial) Time-Series

Choosing based on indexing capabilities:

- Indexes types?
  - Mono/multi-dim, tree-structured, inverted-list
- Spatial indexing?
  - R-tree, BRIN, etc.
- Secondary indexing?
  - Native,



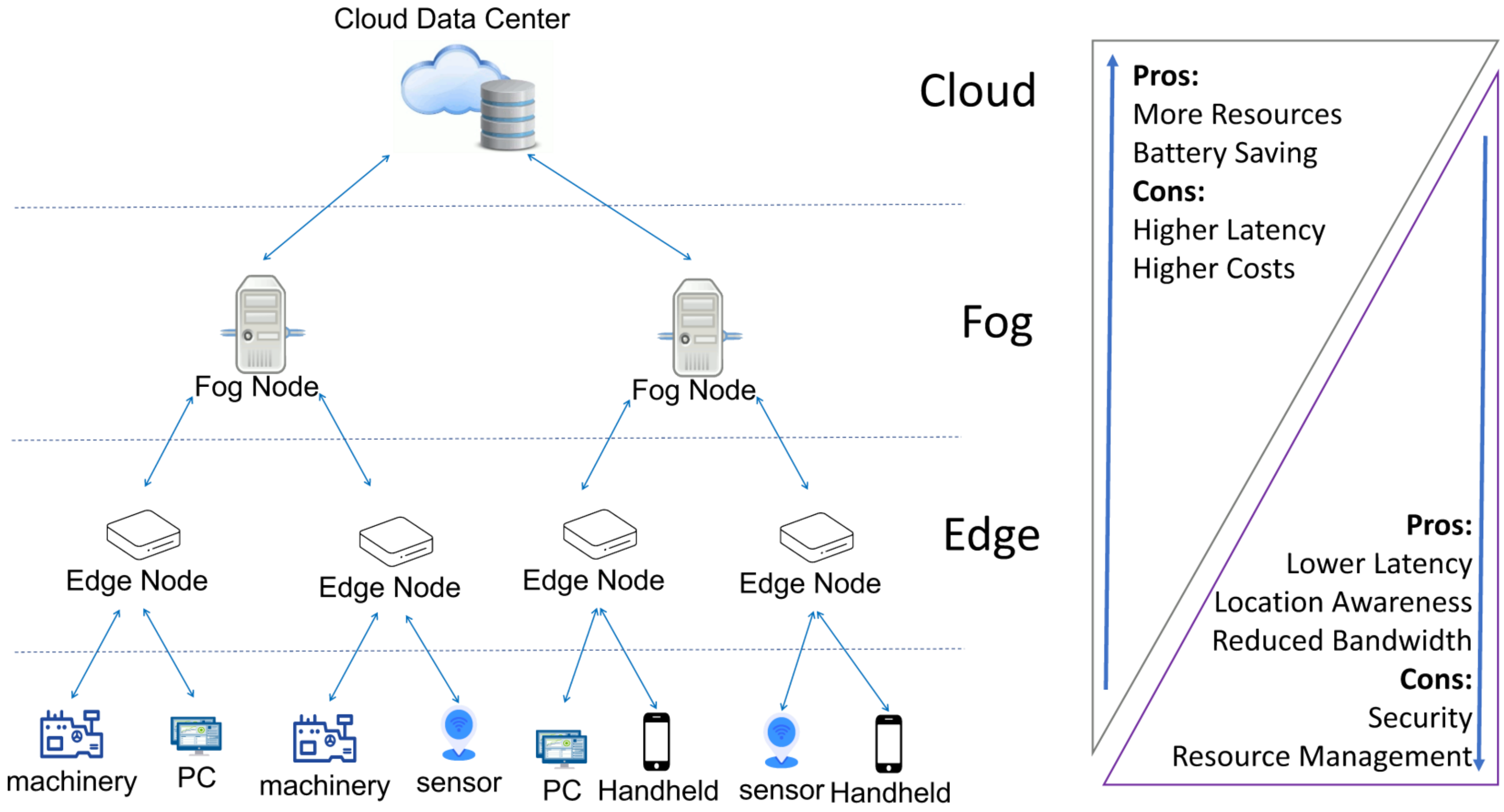

R-tree for spatial indexing []

# Task Offloading

The offloading practice mainly focuses on computation.

Offloading types[12]:

- Full
- Partial
- Mixed



# A Recent Trend: Fog Storage Offloading

Storage as a Service (SaaS) on Fog is a recent trend [13].

Identified Storage approaches:

- Caching [14]
- Pre-fetching [15]
- Local-storage [16]

Possible behaviours:

- Passive
- Active
- Pro-Active

# Benchmarking Tools

- Yahoo Cloud Serving Benchmark (YCSB) [17]
  - Allows for CRUD traffic tuning (op. mix, no parameters).
  - Focused on delay reduction in Cloud storage, <u>neglects costs</u>.
- TPCx-IoT [18] industrial-grade benchmark based on YCSB
  - Metrics for <u>cost-performance($/time)</u> evaluation of IoT gateway systems.
  - Allows inbound traffic tuning, NO analytic workload tuning.

# Simulation Tools

Cutting edge Fog simulation tools[20] iFogSim, FogTorch, FogDirMine, FogNetSim:

- Mostly focused on infrastructure simulation.
- Non-trivial effort for extension and simulation of specific contexts.

# Cost Modeling

Most commonly charged aspects

- Up-time
- Disk-space
- Bandwidth

Other depending on CSP

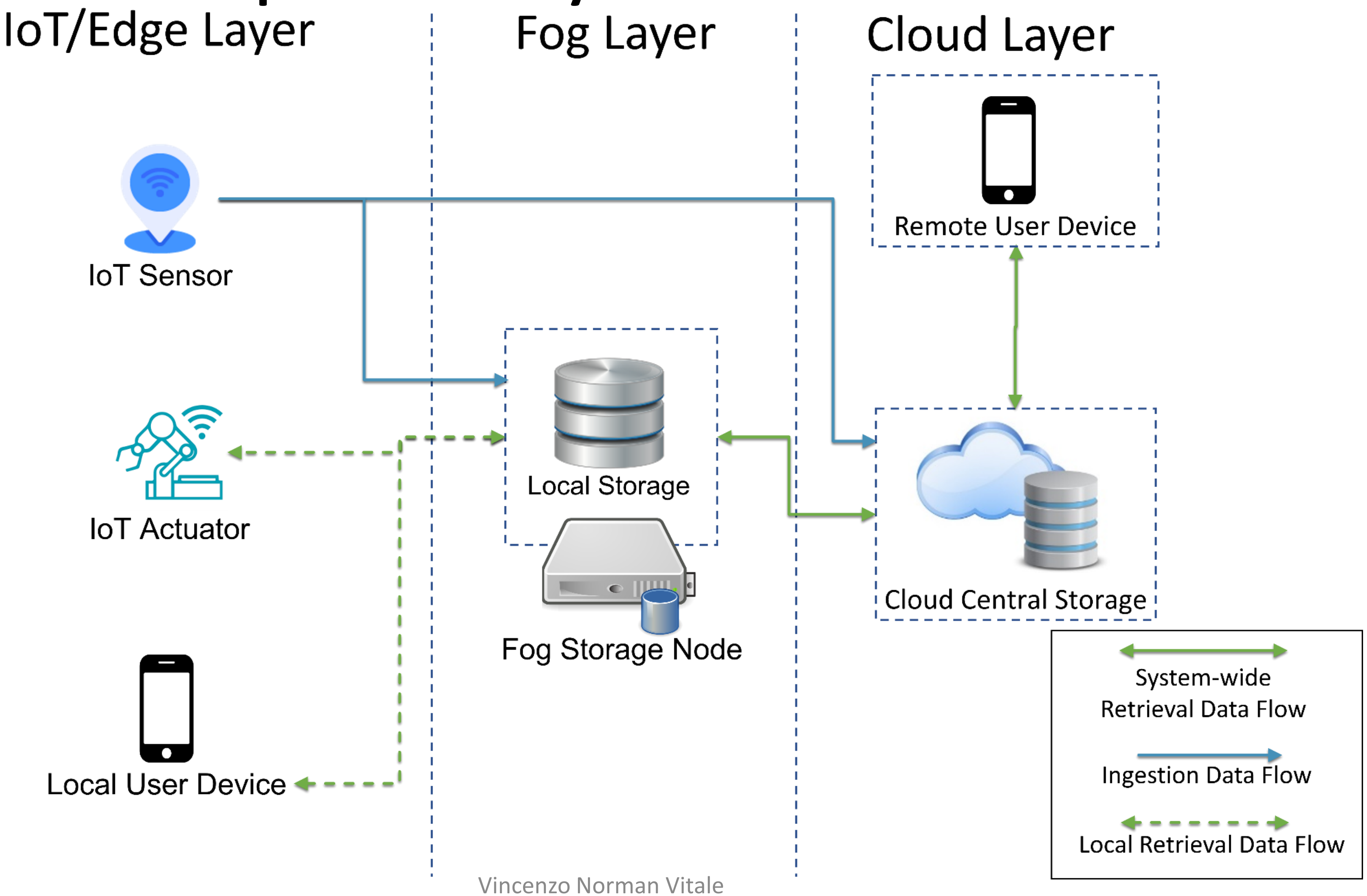
Most common cost factors in View Materialization[20]:

$$Cost = C_{computing} + C_{storage} + C_{transfers}$$

- Time to compute views
- Storage space to store them
- Transfer data amount

# The Proposed Hybrid Architecture

# The Proposed Hybrid Architecture

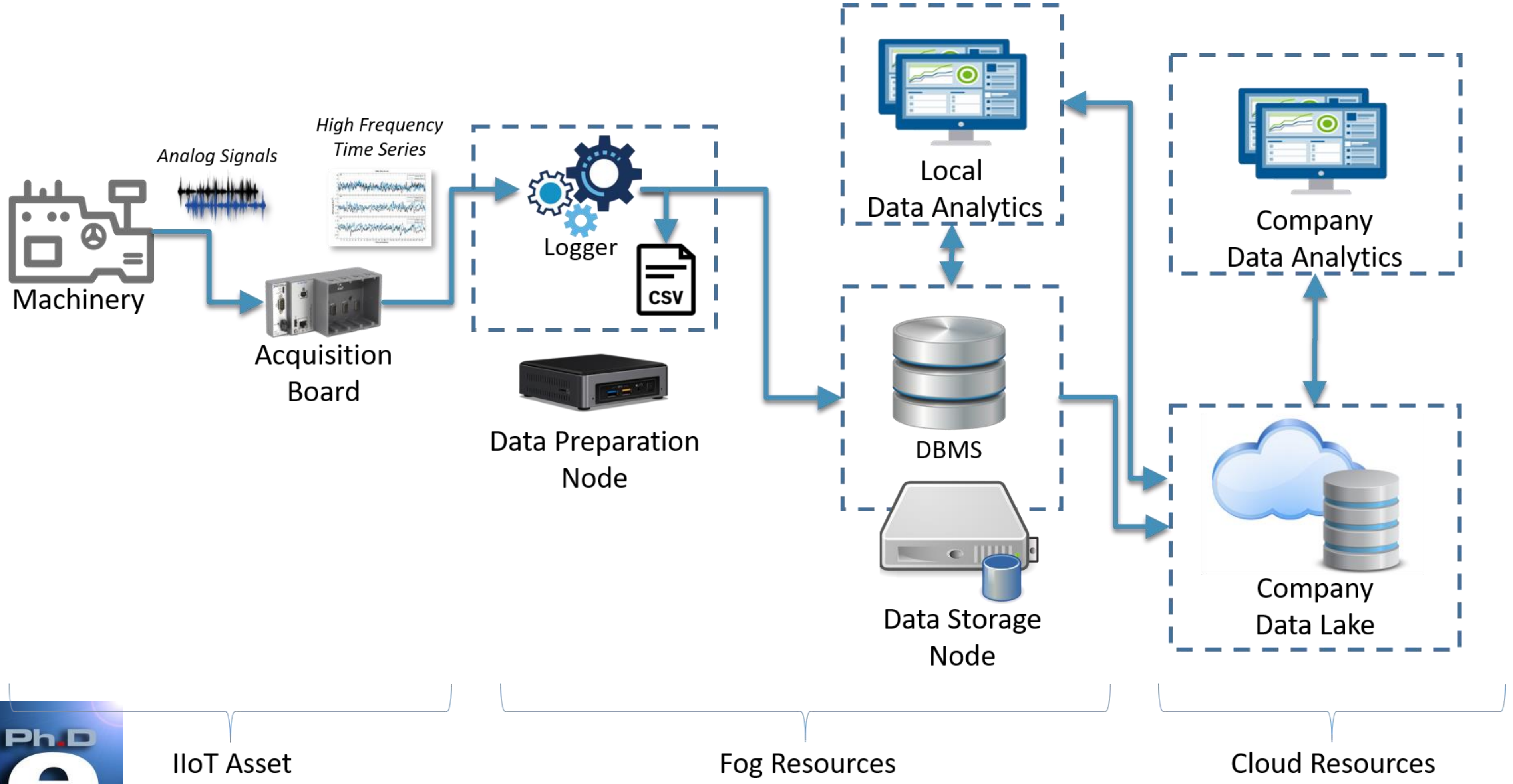


# The Proposed Hybrid Architecture

- Focused on Costs
- Fog is an active part of Data Management Architecture
- Combine Fog Location-Awareness with Workload Knowledge

# Assessing IoT Data Management In Hybrid Environments

# The Deployed IIoT Architecture



# Handling High Frequency IIoT

Investigated DBMS on single node:

- Cassandra
- MongoDB
- InfluxDB
- PostgreSQL
- Clustered PostgreSQL

Company Constraints:

- Local and centralized analytics.
- Storage redundancy.
- IIoT data stored locally.
- Off-line data replication in Data Lake.

# Handling High Frequency IIoT

Considered dataset:

- 600 million points over two days.
- 1 query with 3 different filters

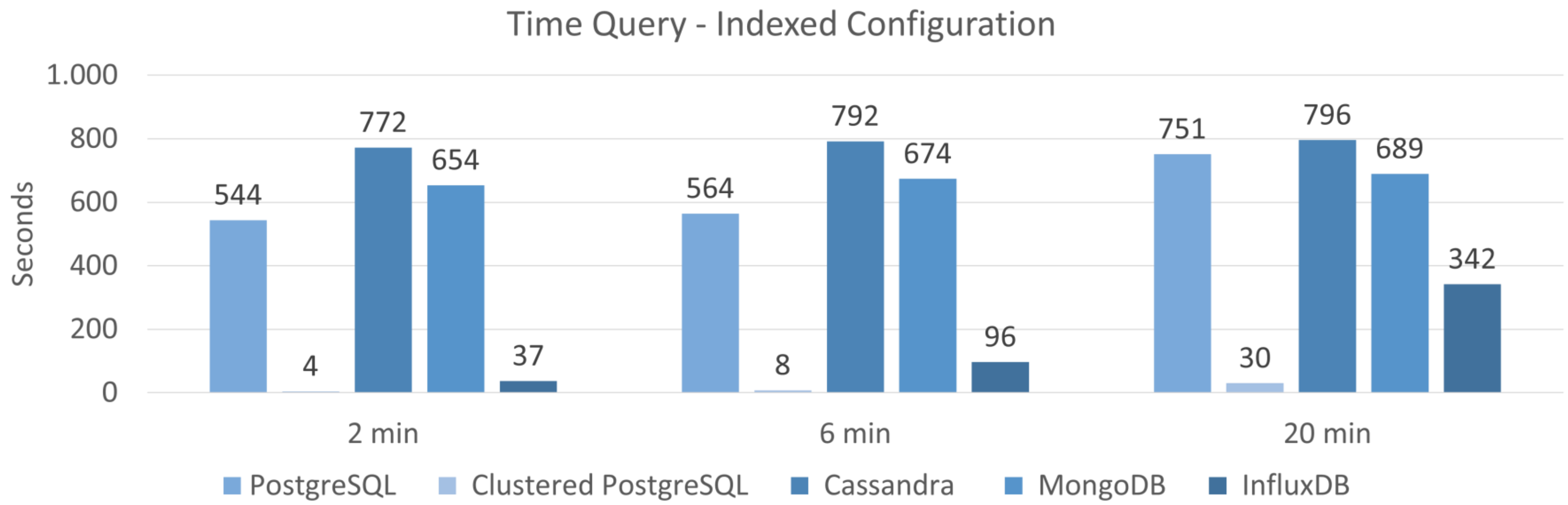
Considered configurations:

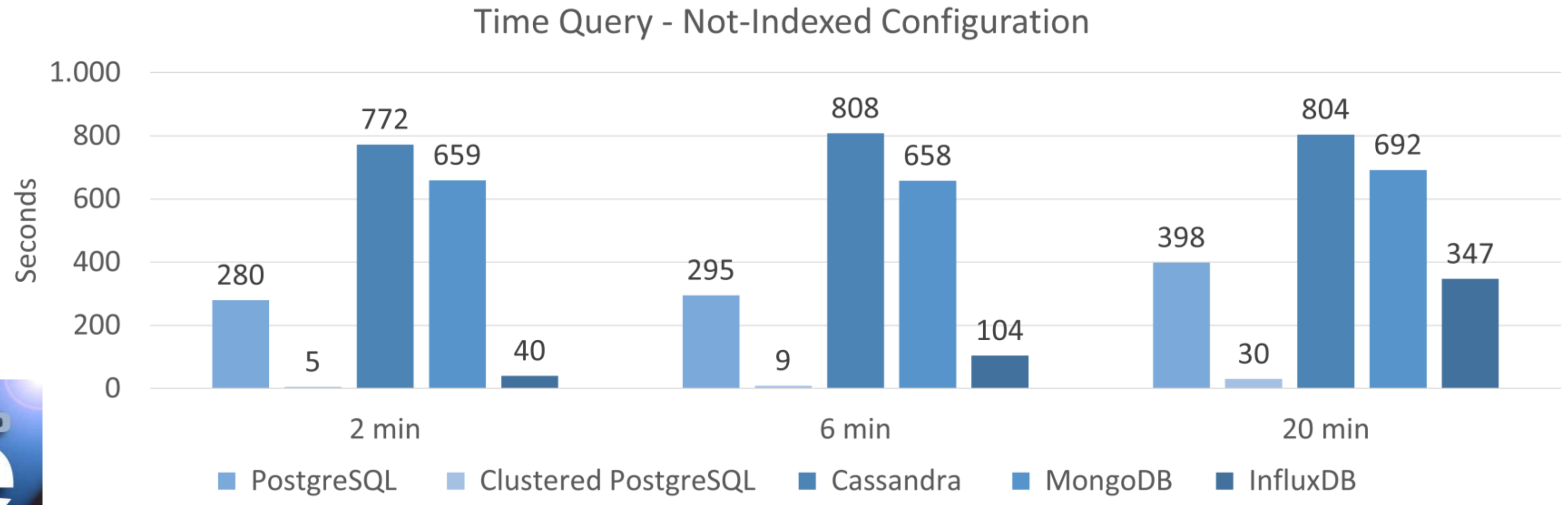
- With secondary indexes
- Without secondary indexes

KPI

- Ingestion Time
- Retrieval Time
  - Primary index
  - Secondary index
- Disk Usage

# Retrieval on Massive Time Dimension



# Results

1. **InfluxDB**: low performance without secondary indexes.
2. **Clustered PostgreSQL:** better performances with indexes , require additional efforts after ingestion.
3. **PostgreSQL**: better performance with secondary indexes
4. **MongoDB:** good performance and disk usage.
5. **Cassandra**: worst performance, unreliable.

# Spatial Time-Series

# Handling Spatial Time-Series

Investigated DBMS on single node:

- PostgreSQL+PostGIS
- Clustered PostgreSQL+PostGIS
- TimescaleDB

KPI

- Retrieval Time
- Disk Usage

# Stationary Time-Series

Parking Monitoring System, 275 million points,2011 to 2017.

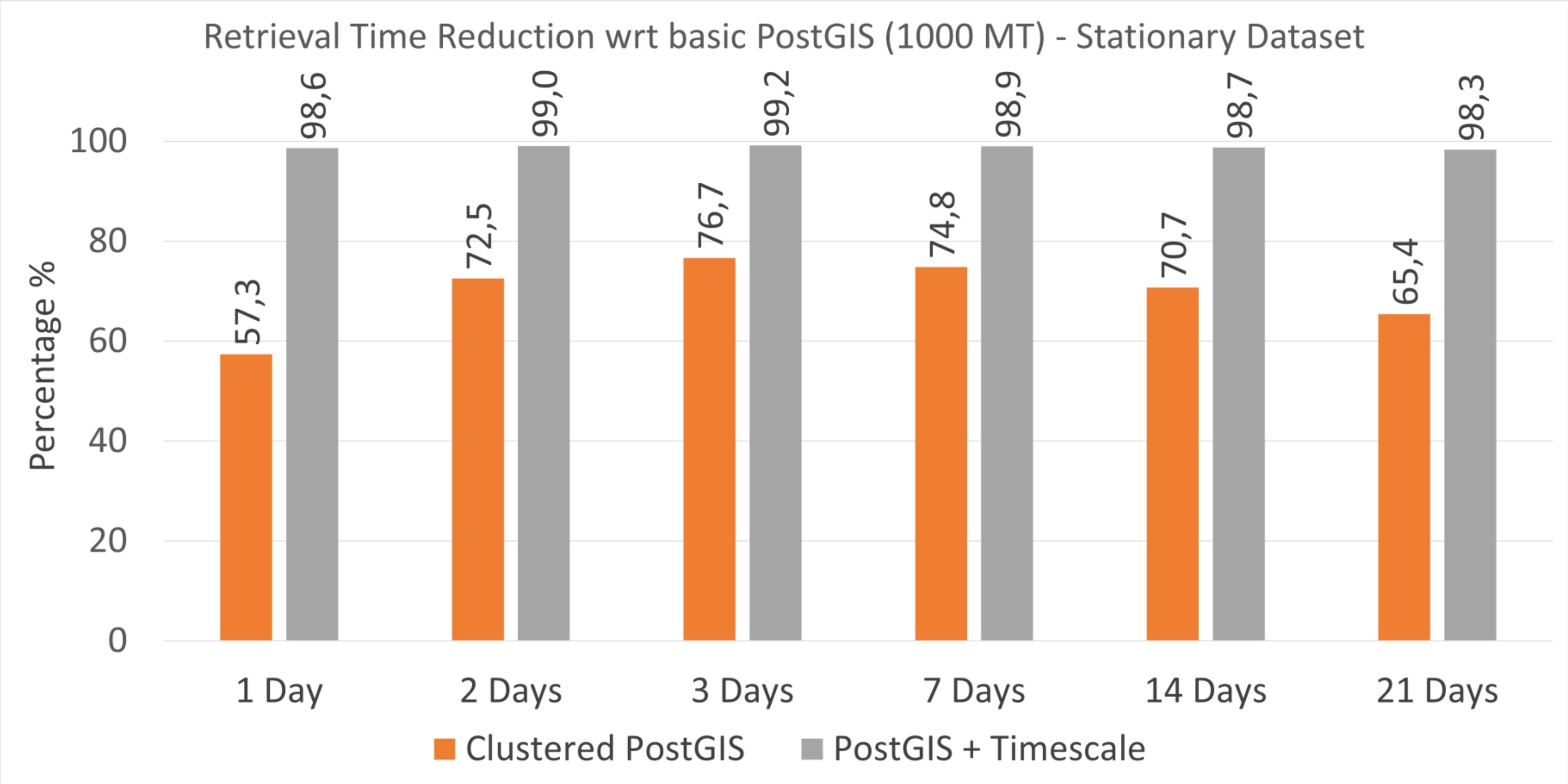
Indexing Configurations:

- Temporal: B-tree
- Spatial: R-tree

Filters:

- 1,3,7,14,21 days
- 100,500,1000 meters

# Retrieval Time Reduction



# Non-Stationary Time-Series

Railway Monitoring System, 58 million points, 30 days.

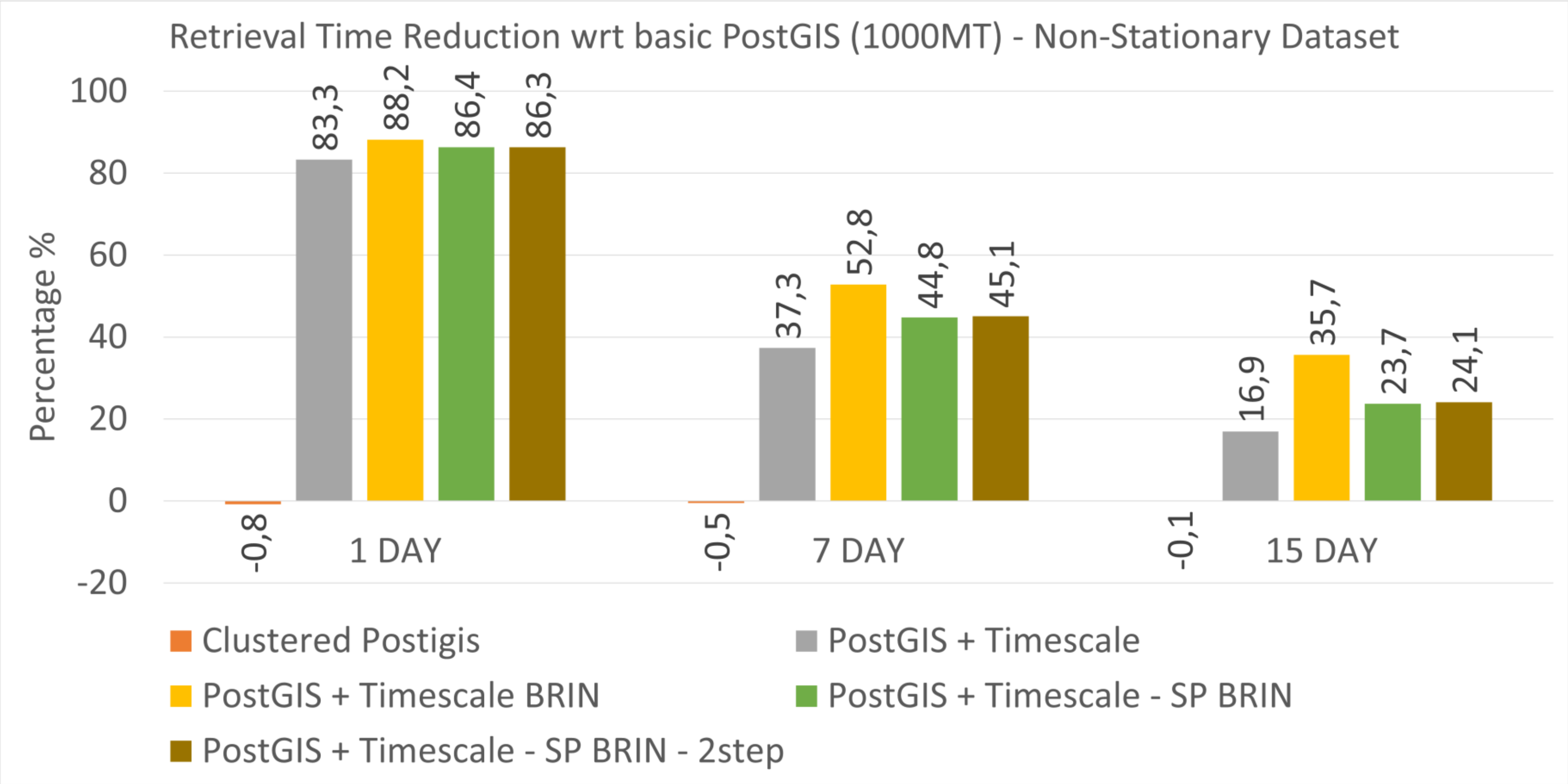
Indexing Configurations:

- Temporal: B-tree
- Spatial: R-tree, Brin, R-tree Secondary Partitioning (SP), Brin Secondary Partitioning (SP)

Filters:

- 1,3,7,14,21 days
- 100,500,1000 meters

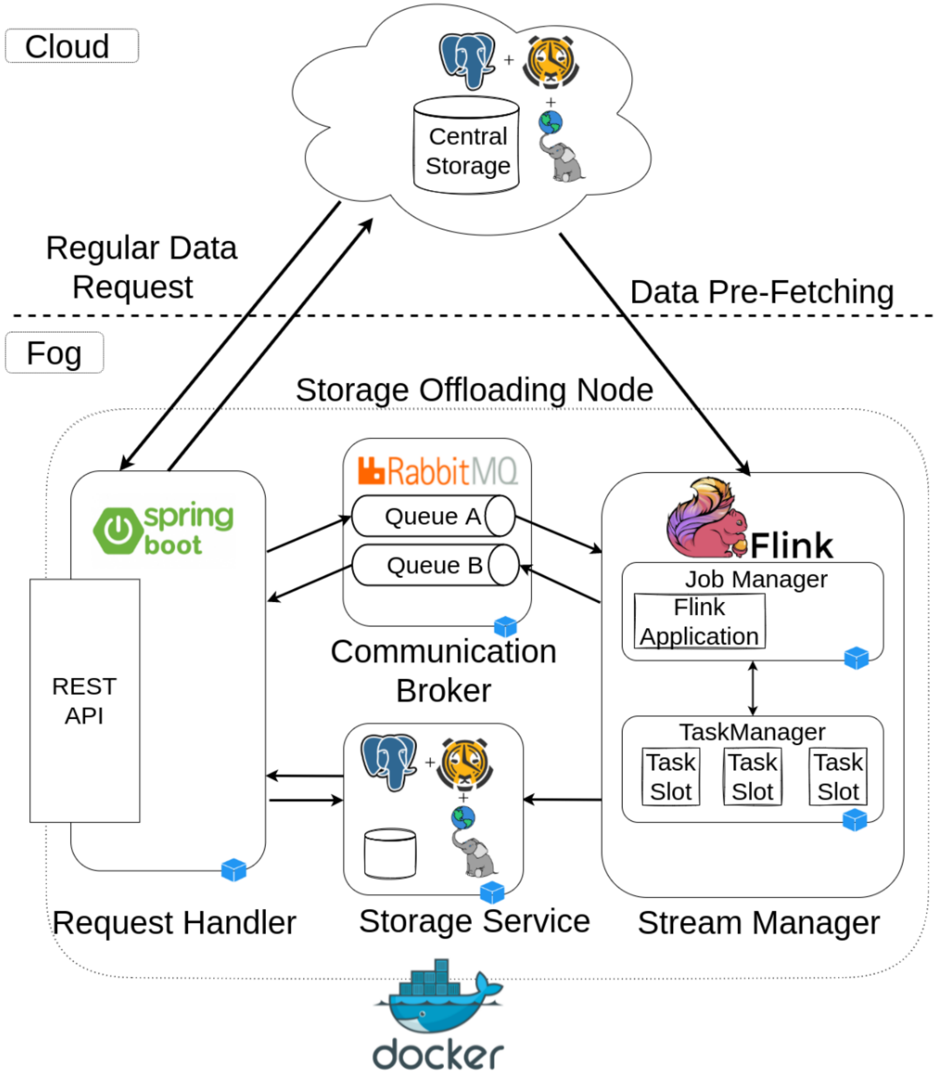
# Non-Stationary Time-Series



# Assessing Fog Performances with Massive IoT Data

# The Considered Architecture

- Cloud: Microsoft Azure
- Fog: General-purpose Dell server in Via Claudio



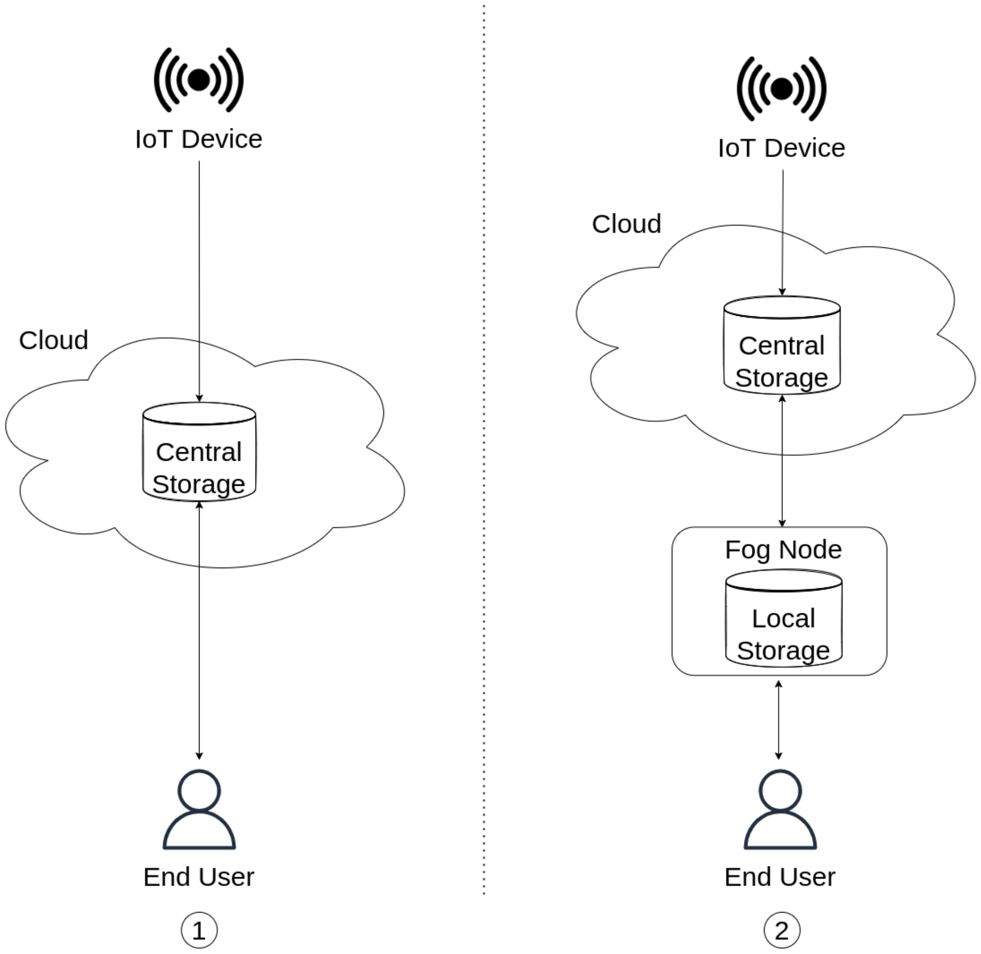
# Considered Configurations

Different Traffic Conditions:

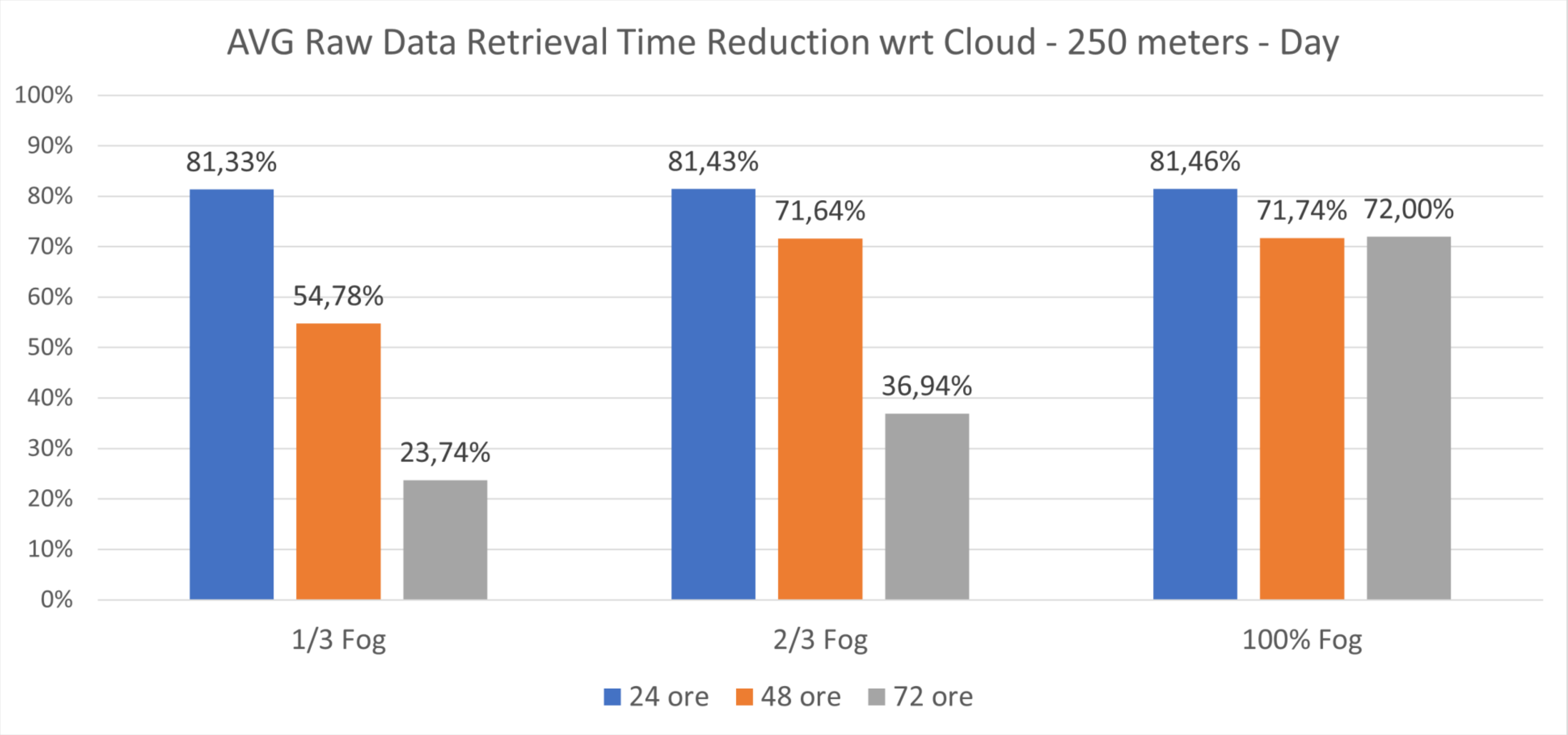
- High Traffic (Daily)
- Low Traffic (Nightly)

Filters:

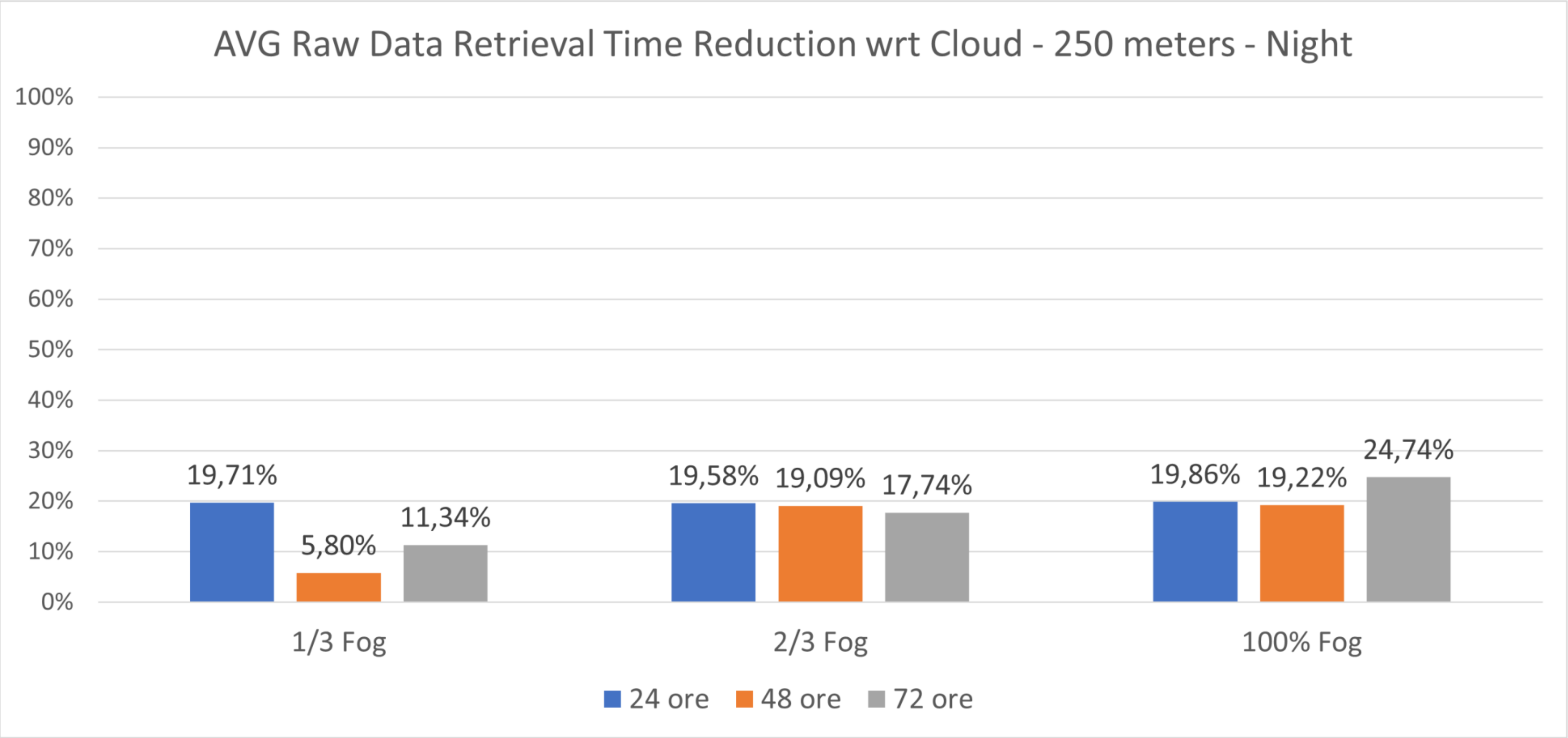
- 1,2,3 days
- 25,50,100,250 meters



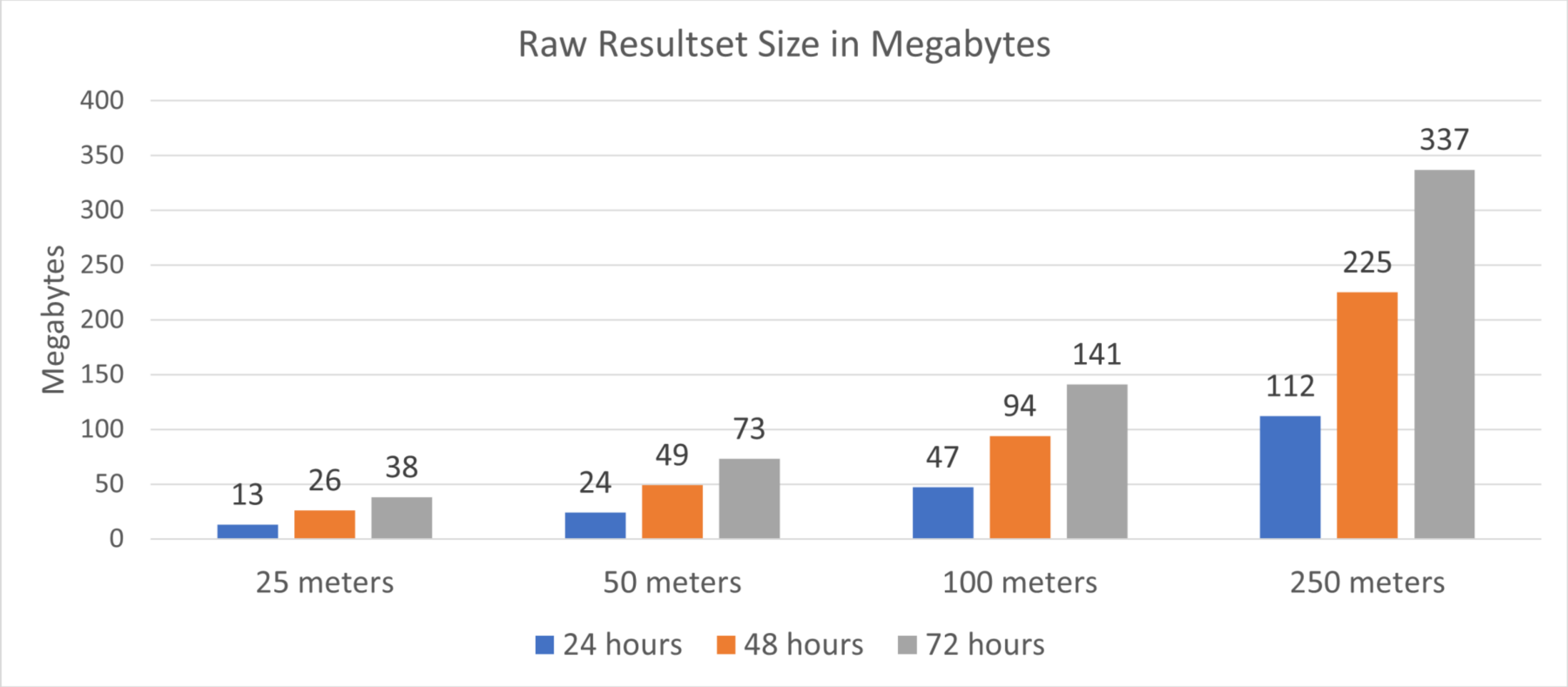
# High Traffic (Day)



# High Traffic (Night)



# Raw Resultset size

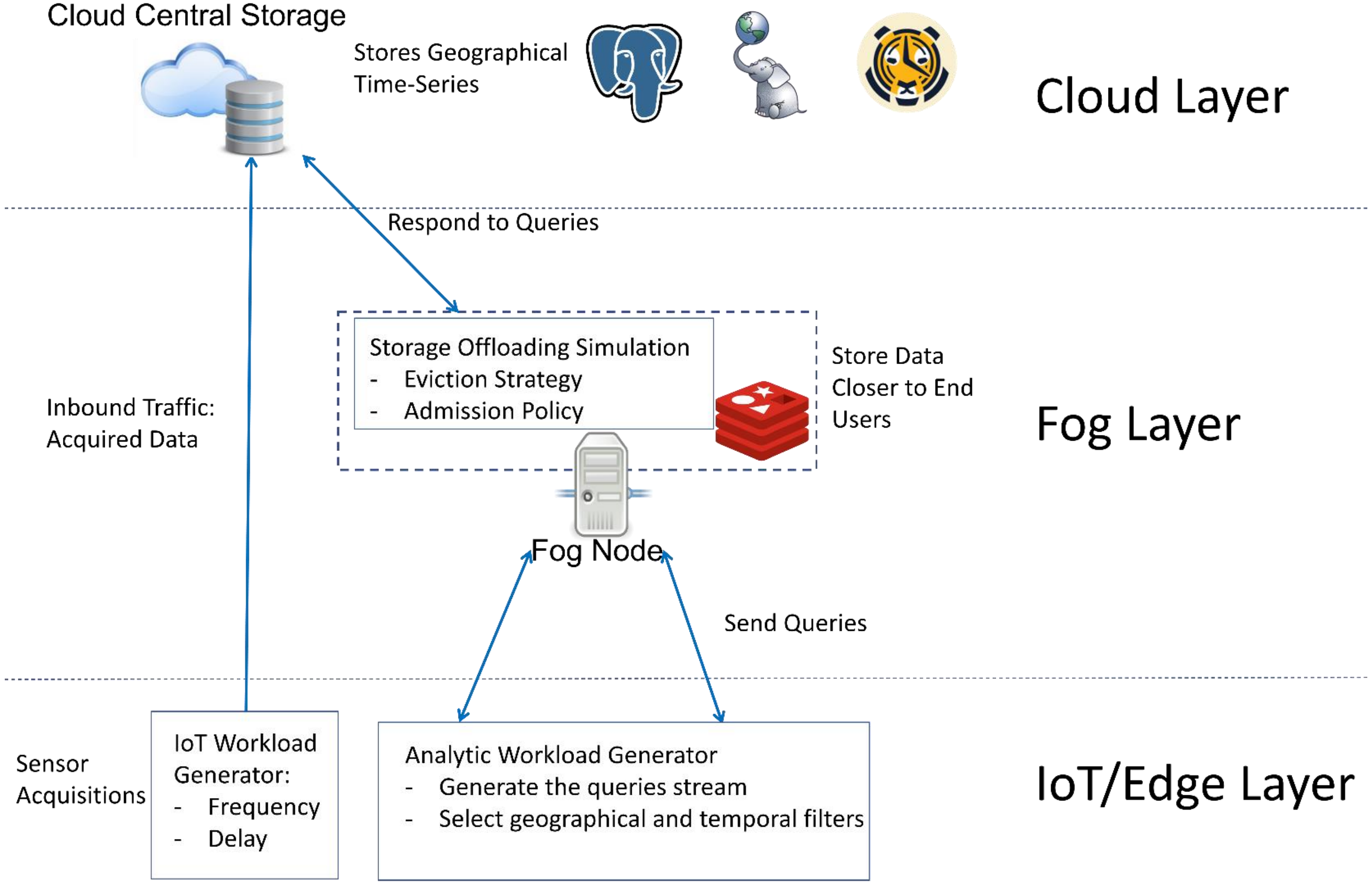


# The Proposed Tool for Fog Storage Offloading Assessment

# The Simulated Architecture

- Emulate IoT Traffic
- Generate Custom Analytic Workload
- Support Different Fog Storage Offloading Approaches

- 7 Query Templates



# Requirements

*Fog Storage Offloading*

- Flexible behaviour simulation.
- Restrict resources.
- Traffic monitoring.

*Analytic Workload*

- Control Access pattern
- Customize Temporal access complexity
- Customize Spatial access complexity

# Modeling Analytic Workload

Query Stream Syntax

- Query Template and Patterns.
- Changes over time[23].

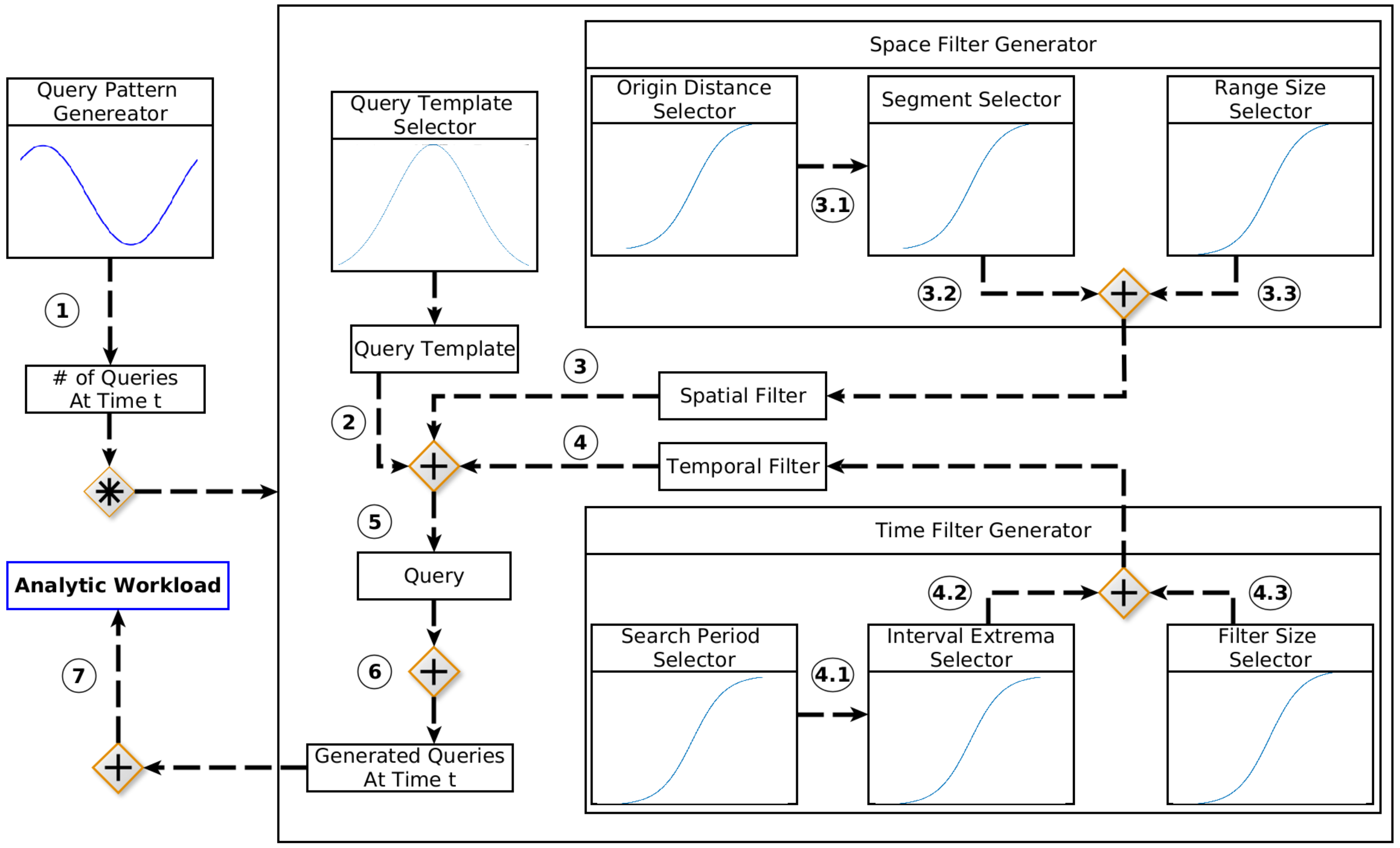
Query Stream Semantic

- Actual spatial and temporal filters.
- Represents users' information need[23].

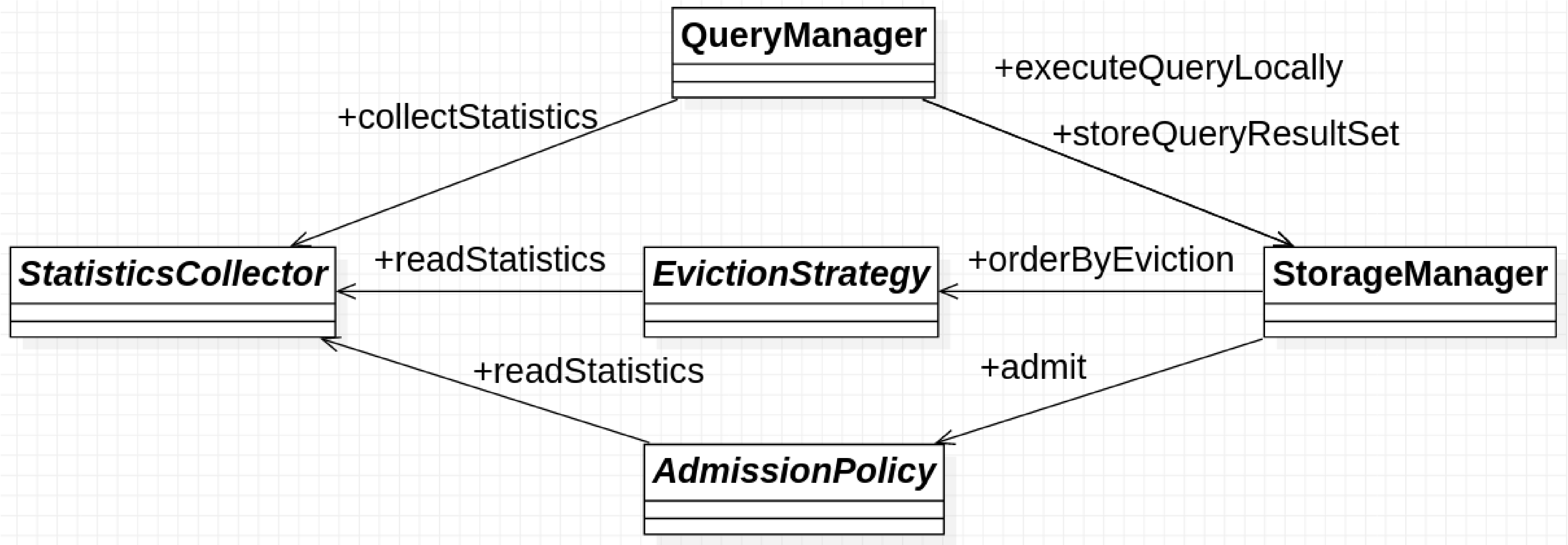
# Analytic Workload Generation

Workload tuning:

- Built-in functions (Sigmoid, Constant, Gaussian)
- Custom Functions
- Custom Model



# Fog Storage Offloading Core Functionalities



# Assessing the Efficiency of Fog Storage Offloading

# The Considered Approach

## Lazy Caching

- Admit all
- Eviction Strategy:
  - LRU
  - LFU

Standard KPI:

- Request hit-rate
- Byte hit-rate

Proposed KPI:

- Byte Information Rate (BIR)

# Byte Information Rate (BIR)

Provides information depending on the architecture layer:

- **At Cloud**: gives an estimation of cost ratio
- **At Fog**: gives an estimation of efficiency and saings

$$BIR = \frac{InformationStream(T)}{PersistedStream(T)}$$

- InformationStream(T): byte retrieved during T
- PersistedStream(T): byte stored during T

# The Considered Approach

## Lazy Caching

- Cache size: 256MB, 512MB, 1GB.
- Admission policy: passively admit all.
- Eviction Strategy:
  - LRU
  - LFU

## Experimental Settings

- 2 Days
- Daily Cyclic queries [23]
- IoT workload
  - 1 reading per minute
  - 12843 sensors
- Analytic Workload:
  - Temporal Filter: 30min
  - Spatial Filter: 125,250,500 meters

# Workload Generation Parameters

Workload Parameters

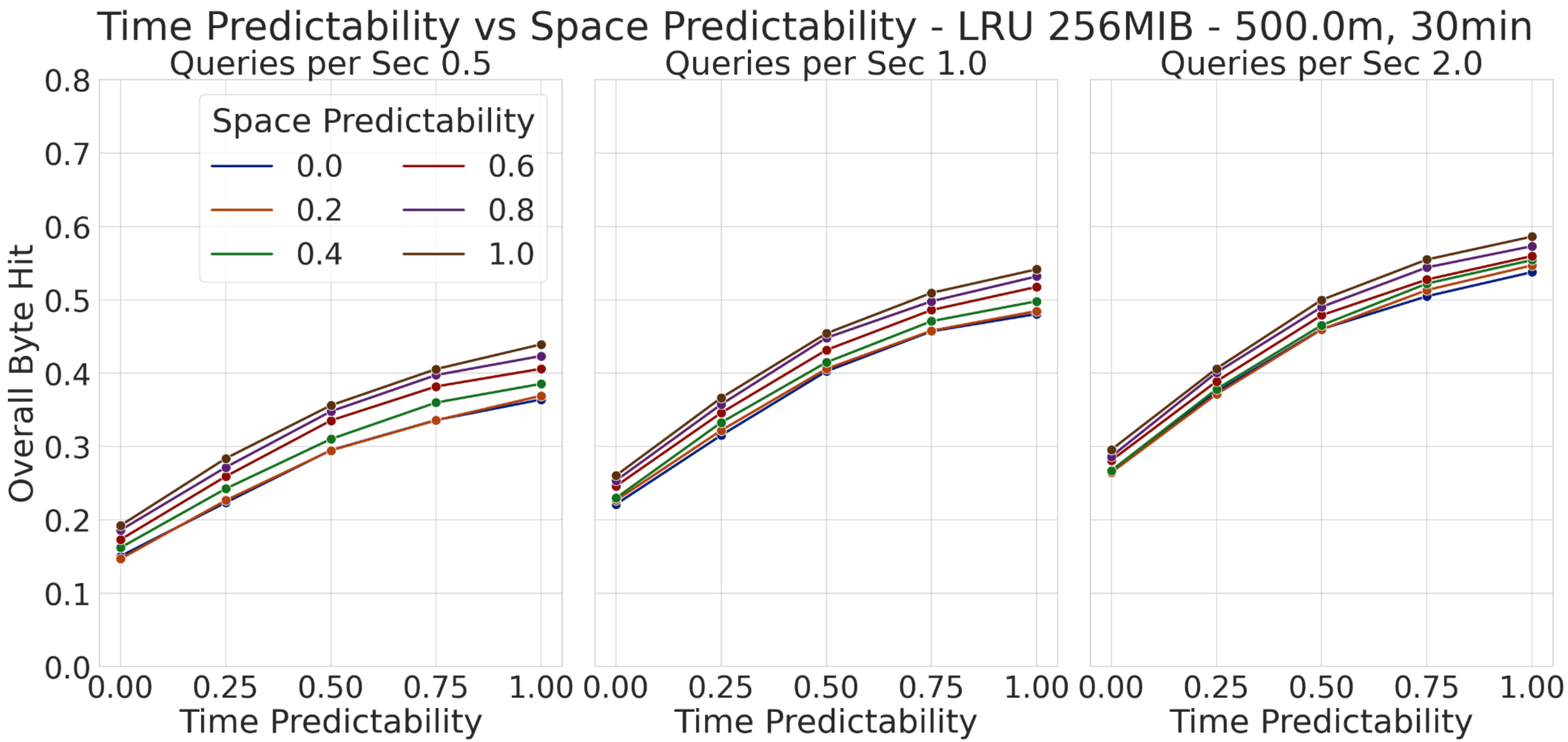
- Query-generator Gaussian with 5 predictability grades
- Temporal filter Sigmoid 5 predictability grades.
- Spatial filter Sigmoid 6 predictability grades.
- Queries Per Second 0,5/1/2.

1350 generated workloads

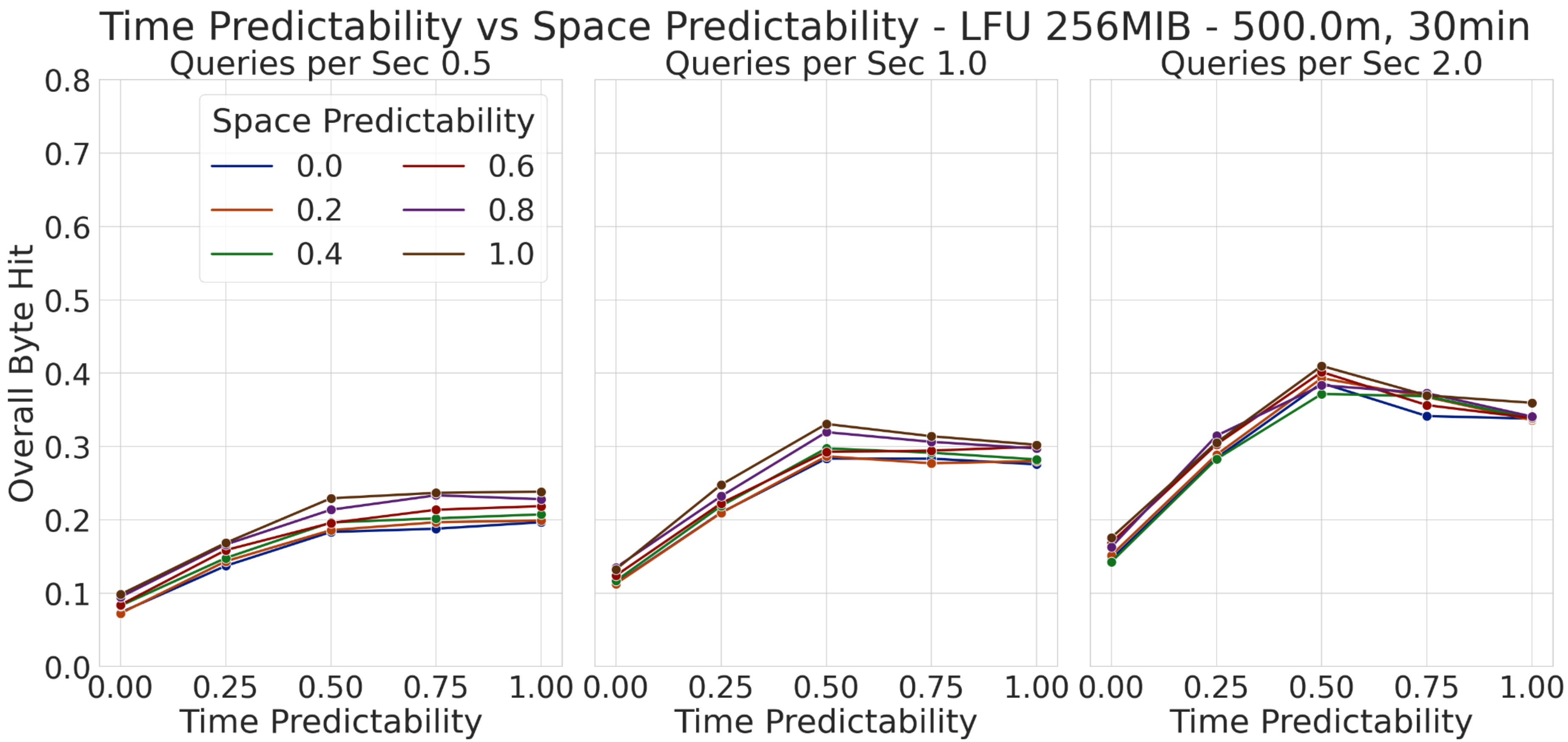
# Workload Observable Aspects

- Query Volume
- Query Pattern Predictability
- Temporal Access Pattern Predictability
- Spatial Access Pattern Predictability

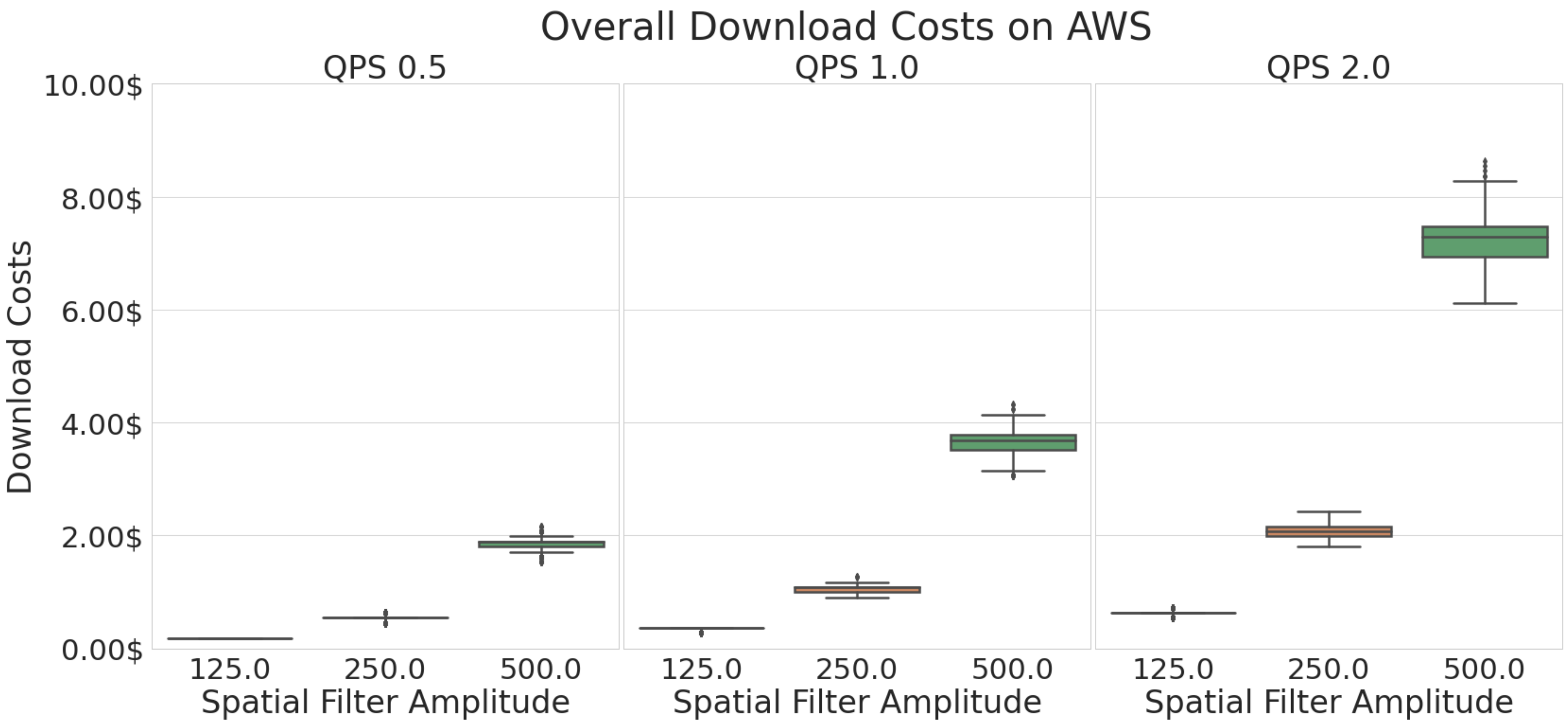
# Byte Hit-Rate - LRU – Most Challenging Scenario



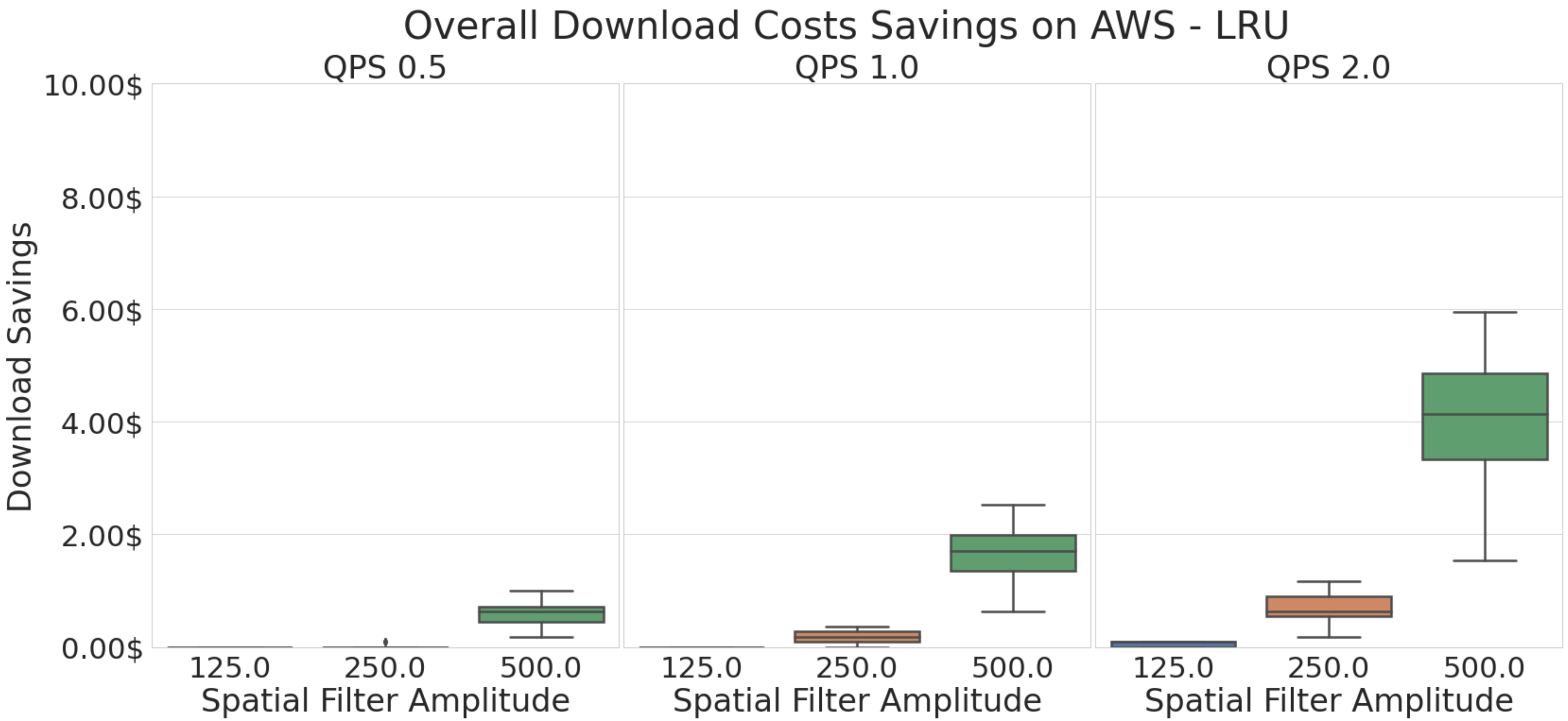
# Byte Hit-Rate - LFU – Most Challenging Scenario



# Download Costs with AWS



# Savings with LRU



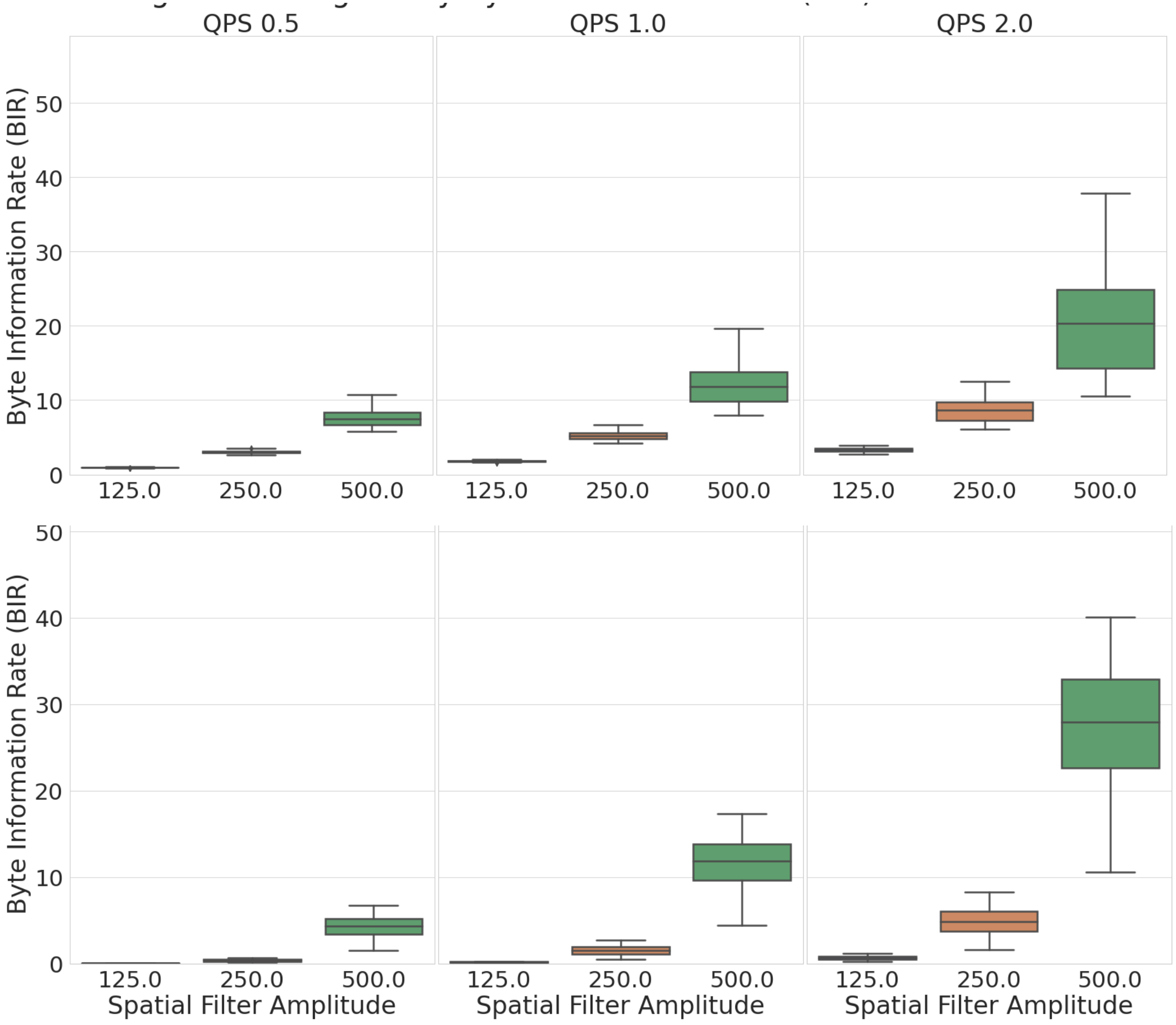
# BIR –LRU

## At Cloud

(lower is better)

## At Fog

(higher is better)



# Conclusions

# RQ1

*Are COTS DBMSs, on single nodes, capable of providing adequate and reliable performance for the ingestion and recovery of massive data streams?*

- Native TSMS like InfluxDB (performance deterioration as filter amplitude increase)
- Releational TSMS like TimescaleDB (high disk requirements)

# RQ2

*How much do the characteristics of the Fog node impact of the performance of data management architecture?*

- Impact on response delay reduction is strongly influenced by traffic conditions (20% in worst case, 80% in best case).
- Impact on bandwidth depends only on the available resources and on Storage Offloading’s approach efficiency (with the wider filter the result is 327MB).

# RQ3

*Can Fog layer's location-awareness and analytical workloads' knowledge improve the cost-performance ratio of a multilayered data management architecture?*

Analytic workload knowledge demonstrated to be

Thank You for the attention!

# References


- [1]Cisco. Internet adoption and network performance. (2021)
- [2]Capgemini Consulting. “Big data blackout: are utilities powering up their data analytics?”.(2015)
- [3] Firat Karatas and Ibrahim Korpeoglu. “Fog-based data distribution service (F-DAD) for internet of things (IoT) applications”. In: Future Generation Computer Systems 93 (2019)
- [4] K Villalobos et al. “A three level hierarchical architecture for an efficient storage of industry 4.0 data”. In: Computers in Industry 121 (2020), p. 103257.
- [5] Xiaomin Li and Jiafu Wan. “Proactive caching for edge computingenabled industrial mobile wireless networks”. In: Future Generation Computer Systems 89 (2018), pp. 89–97.
- [6] Kanishk Chaturvedi and Thomas H Kolbe. “Towards establishing cross-platform interoperability for sensors in smart cities”. In: Sensors 19.3 (2019), p. 562.
- [7] Amir Sinaeepourfard, John Krogstie, and Souvik Sengupta. “Distributed-to-Centralized Data Management: A New Sense of Large-Scale ICT Management of Smart City IoT Networks”. In: IEEE Internet of Things Magazine 3.3 (2020), pp. 76–82.

# References


- [8] Amir Sinaeepourfard et al. “F2c2C-DM: A Fog-to-cloudlet-to-Cloud Data Management architecture in smart city”. In: 2019 IEEE 5thWorld Forum on Internet of Things (WF-IoT). IEEE. 2019, pp. 590–595.
- [9] Dharavath Ramesh, Ashay Sinha, and Suraj Singh. “Data modelling for discrete time series data using Cassandra and MongoDB”. In: 2016 3rd International Conference on Recent Advances in Information Technology (RAIT). IEEE. 2016, pp. 598–601.
- [10] Søren Kejser Jensen, Torben Bach Pedersen, and Christian Thomsen. “Time series management systems: A survey”. In: IEEE Transactions on Knowledge and Data Engineering 29.11 (2017), pp. 2581–2600.
- [11] Eric A Brewer. “Towards robust distributed systems”. In: PODC. Vol. 7. Portland, OR. 2000.
- [12] Firdose Saeik et al. “Task offloading in Edge and Cloud Computing: A survey on mathematical, artificial intelligence and control theory solutions”. In: Computer Networks 195 (2021), p. 108177.
- [13] Ridhima Rani et al. “Storage as a service in fog computing: A systematic review”. In: Journal of Systems Architecture (2021), p. 102033.
- [14] Feng Lu et al. “An adaptive multi-level caching strategy for Distributed Database System”. In: Future Generation Computer Systems 97 (2019), pp. 61–68.

# References

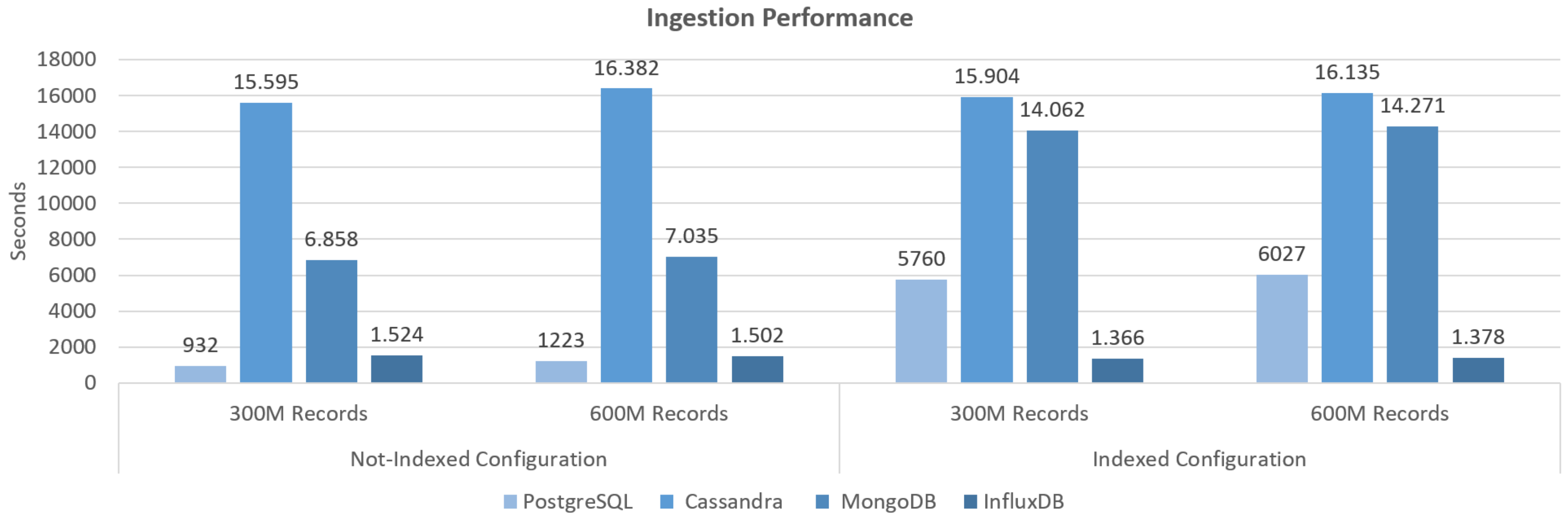

- [15] Brad Glasbergen et al. “Chronocache: Predictive and adaptive midtier query result caching”. In: Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data. 2020, pp. 2391–2406.
- [16] Firat Karatas and Ibrahim Korpeoglu. “Fog-based data distribution service (F-DAD) for internet of things (IoT) applications”. In: Future Generation Computer Systems 93 (2019), pp. 156–169.
- [17] Brian F Cooper et al. “Benchmarking cloud serving systems with YCSB”. In: Proceedings of the 1st ACM symposium on Cloud computing. ACM. 2010, pp. 143–154.
- [18] Meikel Poess et al. “Analysis of tpcx-iot: The first industry standard benchmark for iot gateway systems”. In: 2018 IEEE 34th International Conference on Data Engineering (ICDE). IEEE. 2018, pp. 1519–1530.
- [20] Spiridoula V Margariti, Vassilios V Dimakopoulos, and Georgios Tsoumanis. “Modeling and simulation tools for fog computing—a comprehensive survey from a cost perspective”. In: Future Internet 12.5 (2020), p. 89.
- [21] Thi-Van-Anh Nguyen et al. “Cost models for view materialization in the cloud”. In: Proceedings of the 2012 Joint EDBT/ICDT Workshops.2012, pp. 47–54.
- [22] Bishnu P Bhattarai et al. “Big data analytics in smart grids: State-of-the-art, challenges, opportunities, and future directions”. In: IET Smart Grid 2.2 (2019), pp. 141–154.

# References

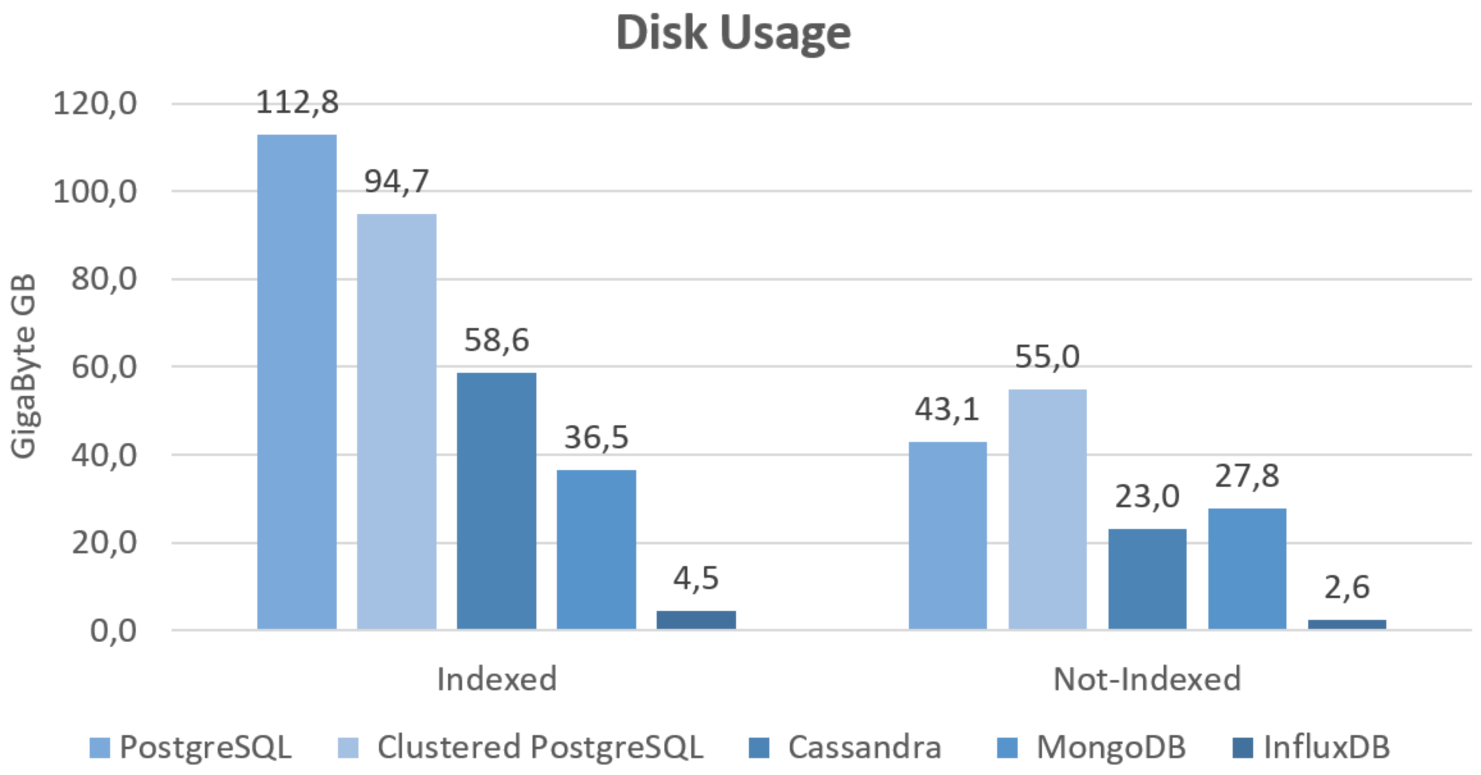

- [23] Lin Ma et al. “Query-based workload forecasting for self-driving database management systems”. In: Proceedings of the 2018 International Conference on Management of Data. 2018, pp. 631–645.
- [24] Eduard Hoenkamp. “On the notion of “an Information Need””. In: Conference on the theory of information retrieval. Springer. 2009, pp. 354–357.

# BACKUP

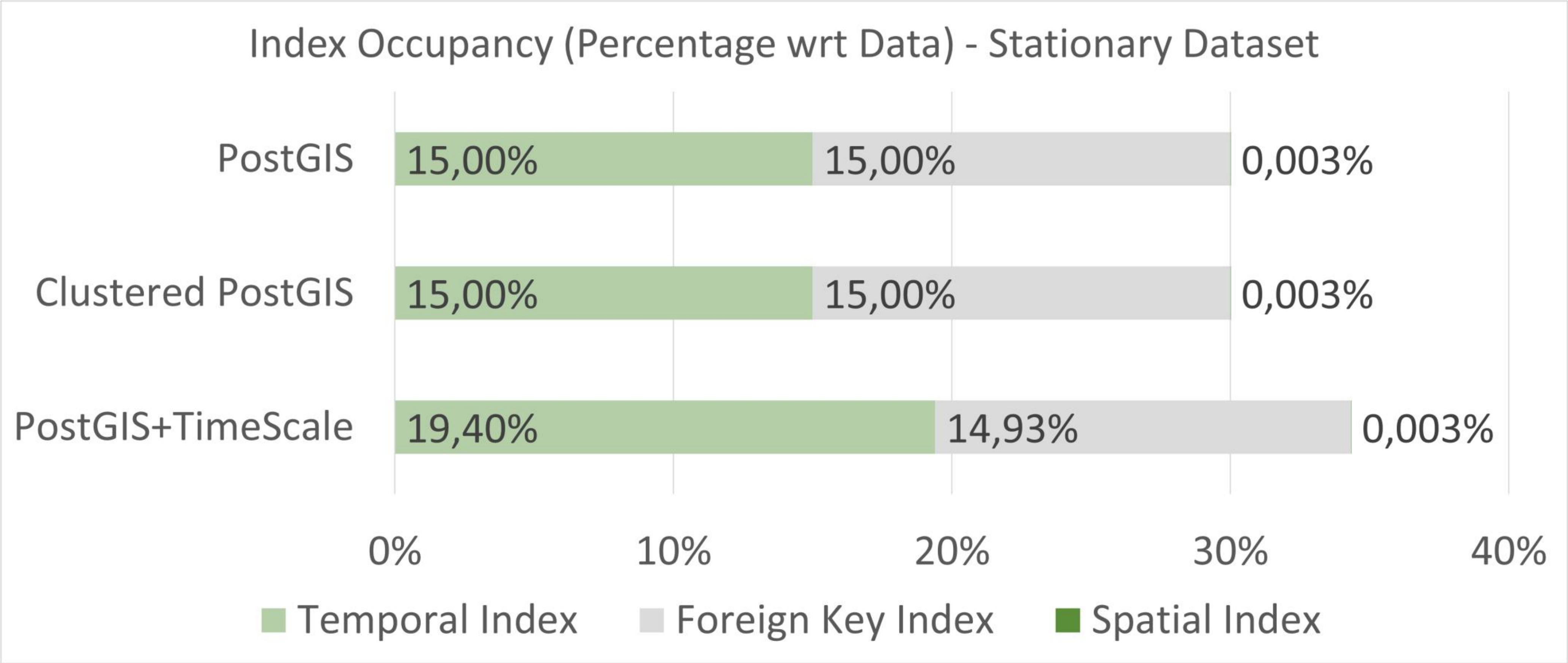
# Handling High Frequency IIoT



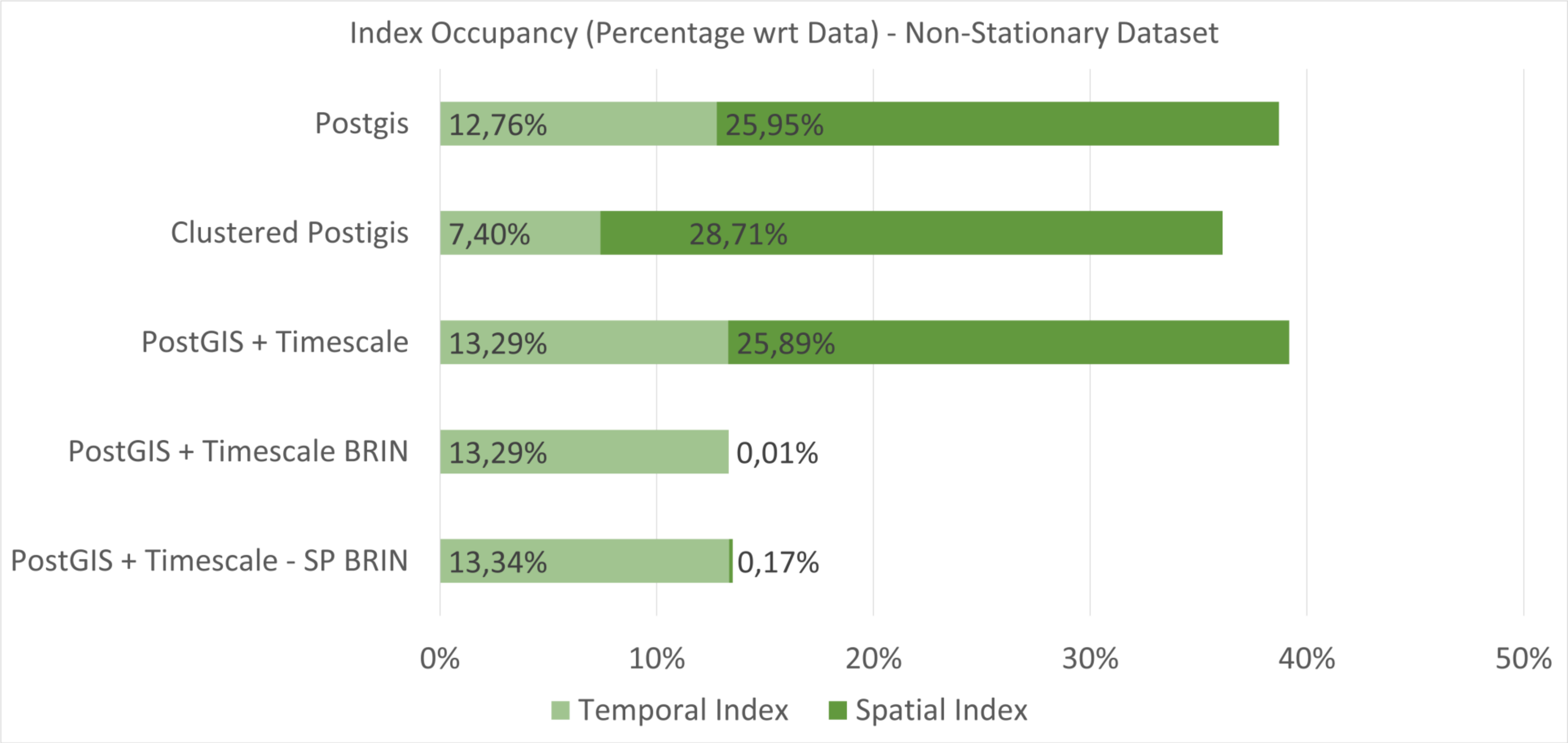
# Handling High Frequency IIoT



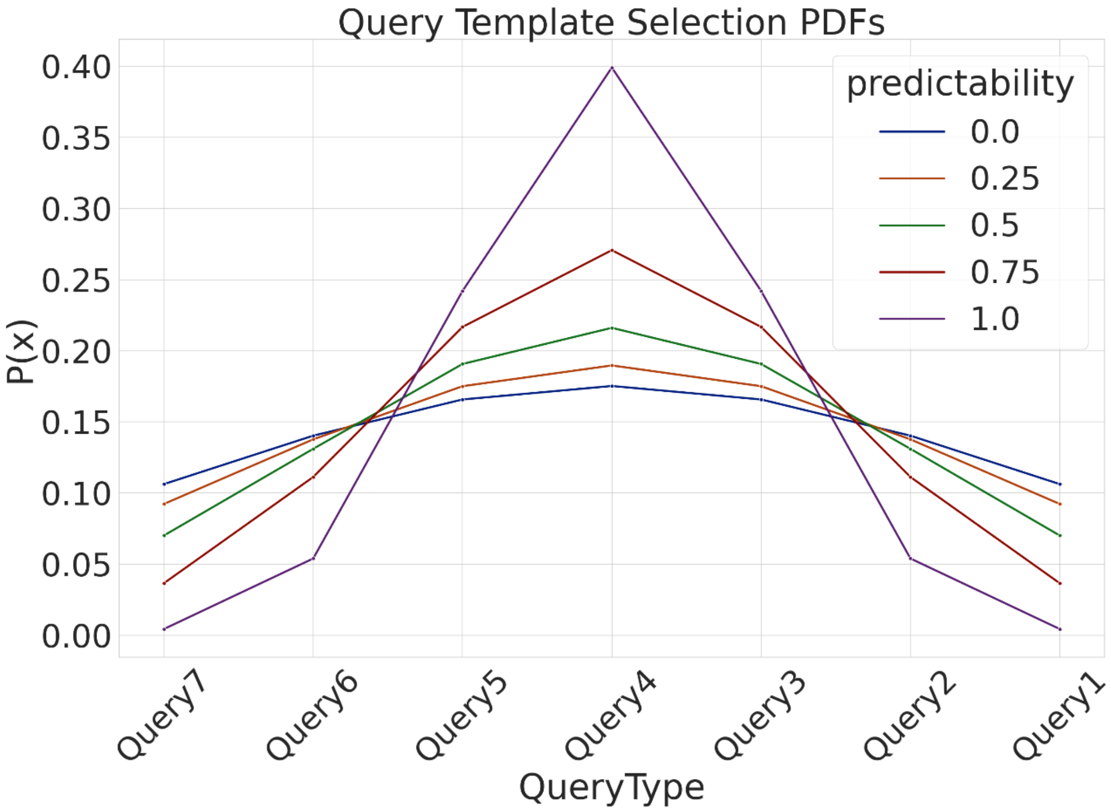
# Stationary Time-Series

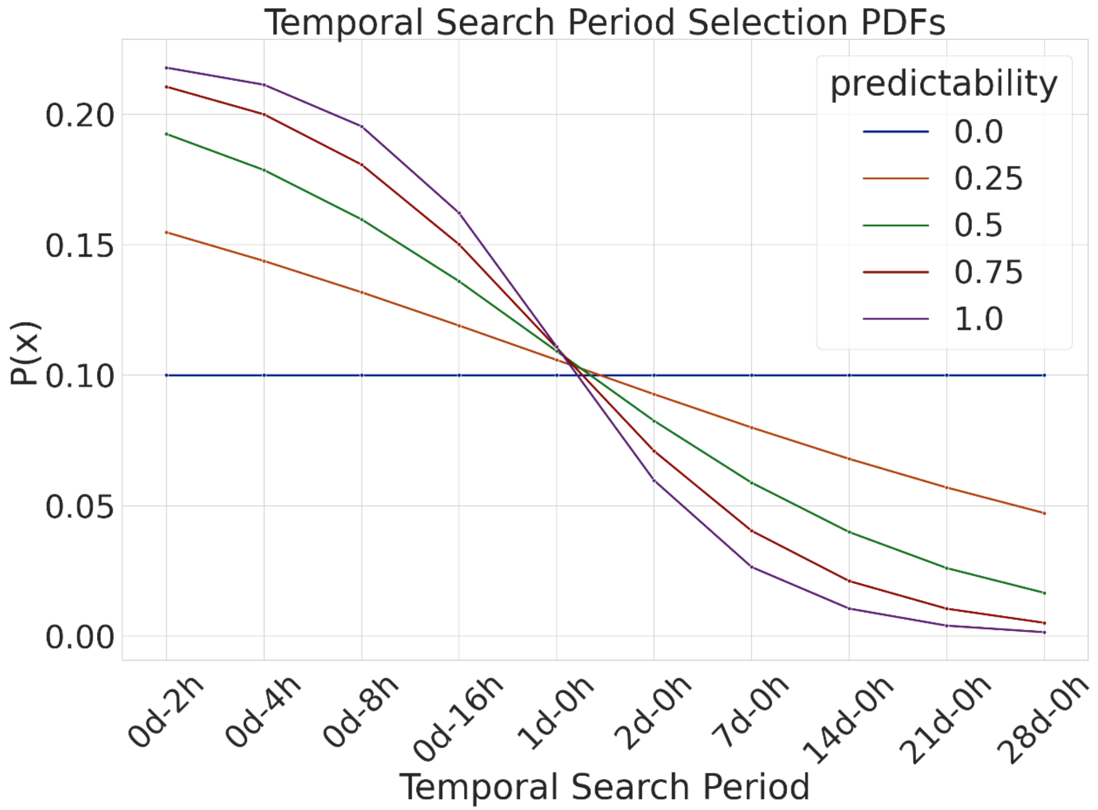


# Non-Stationary Time-Series

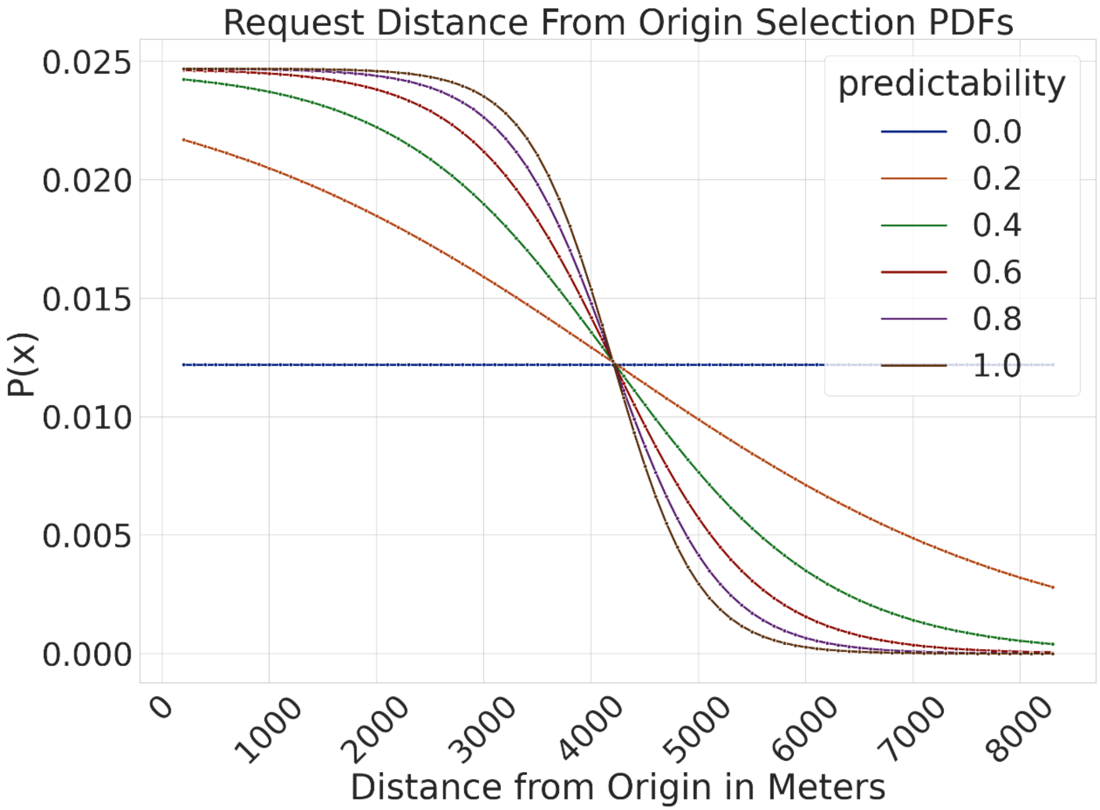


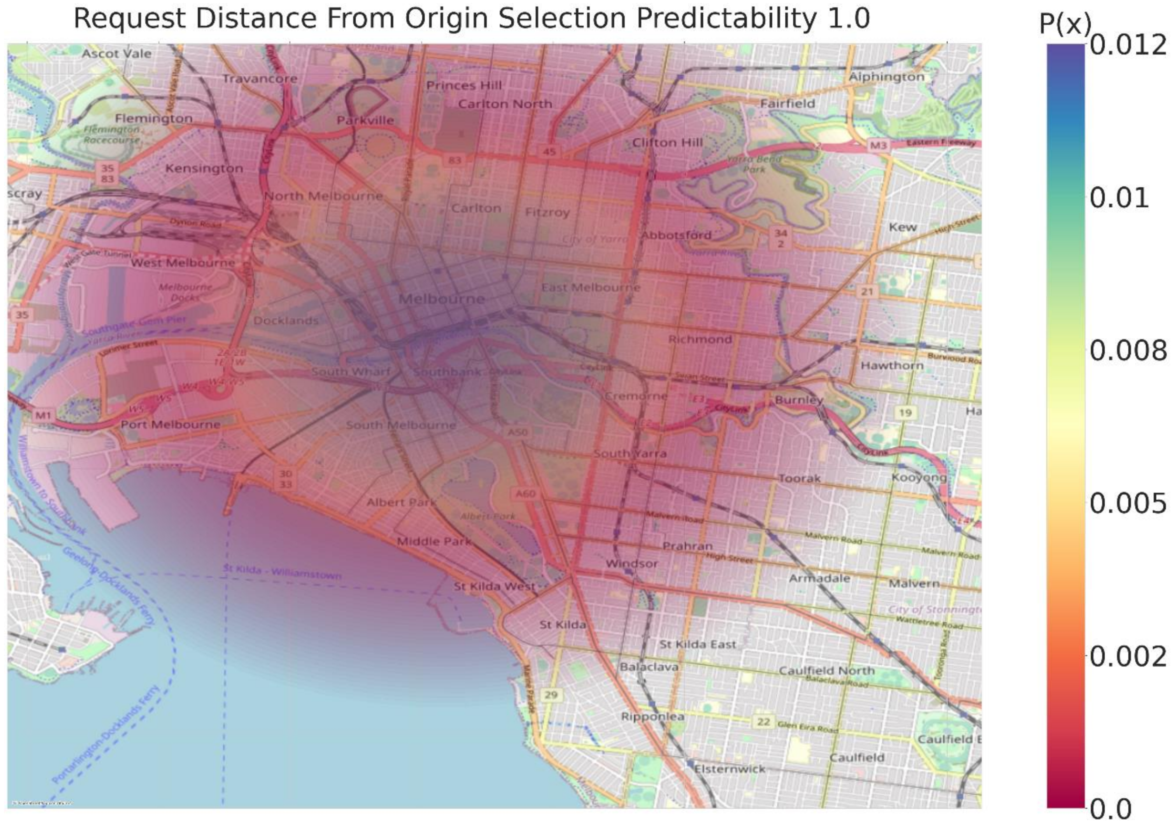
# Generating Analytic Workloads



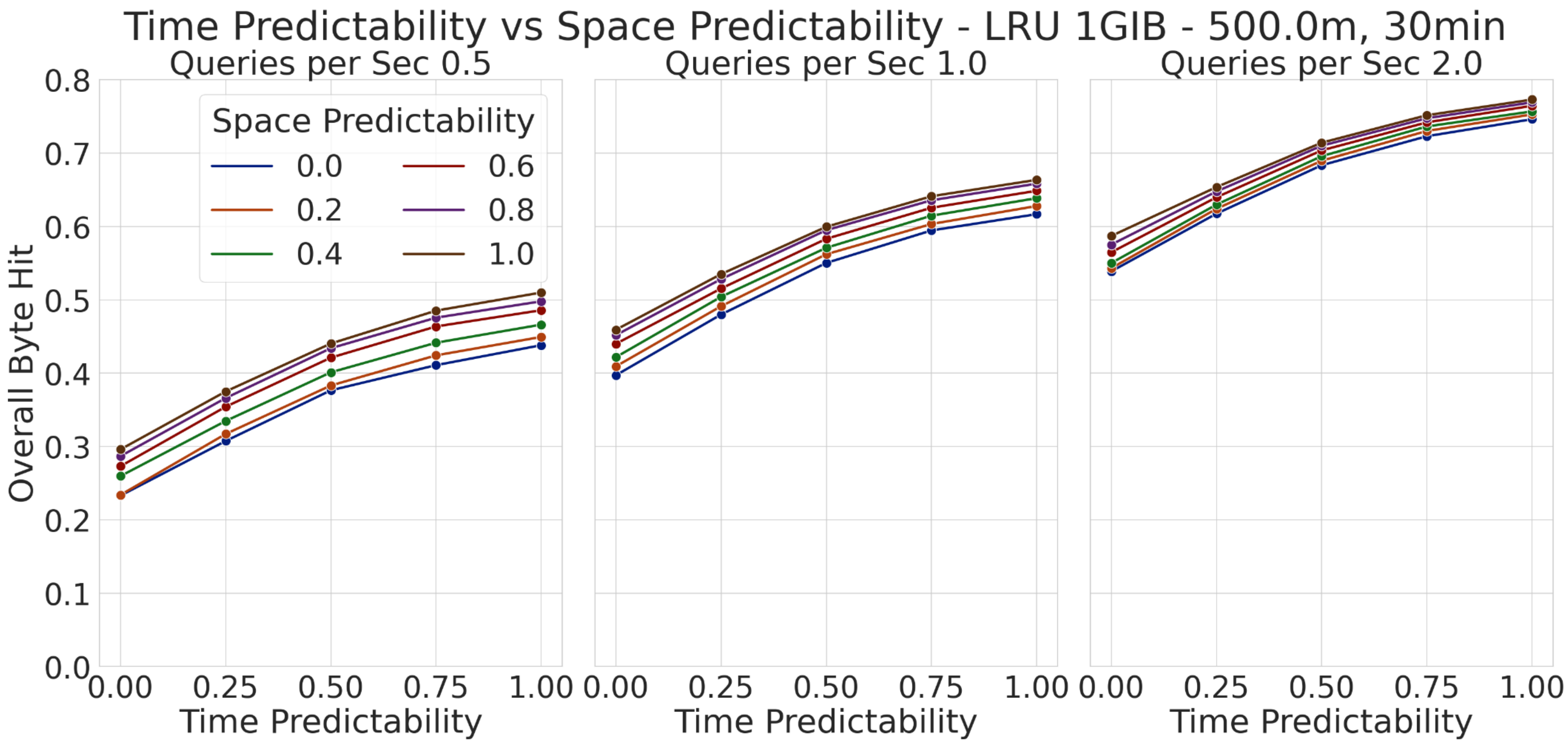



# Generating Analytic Workloads

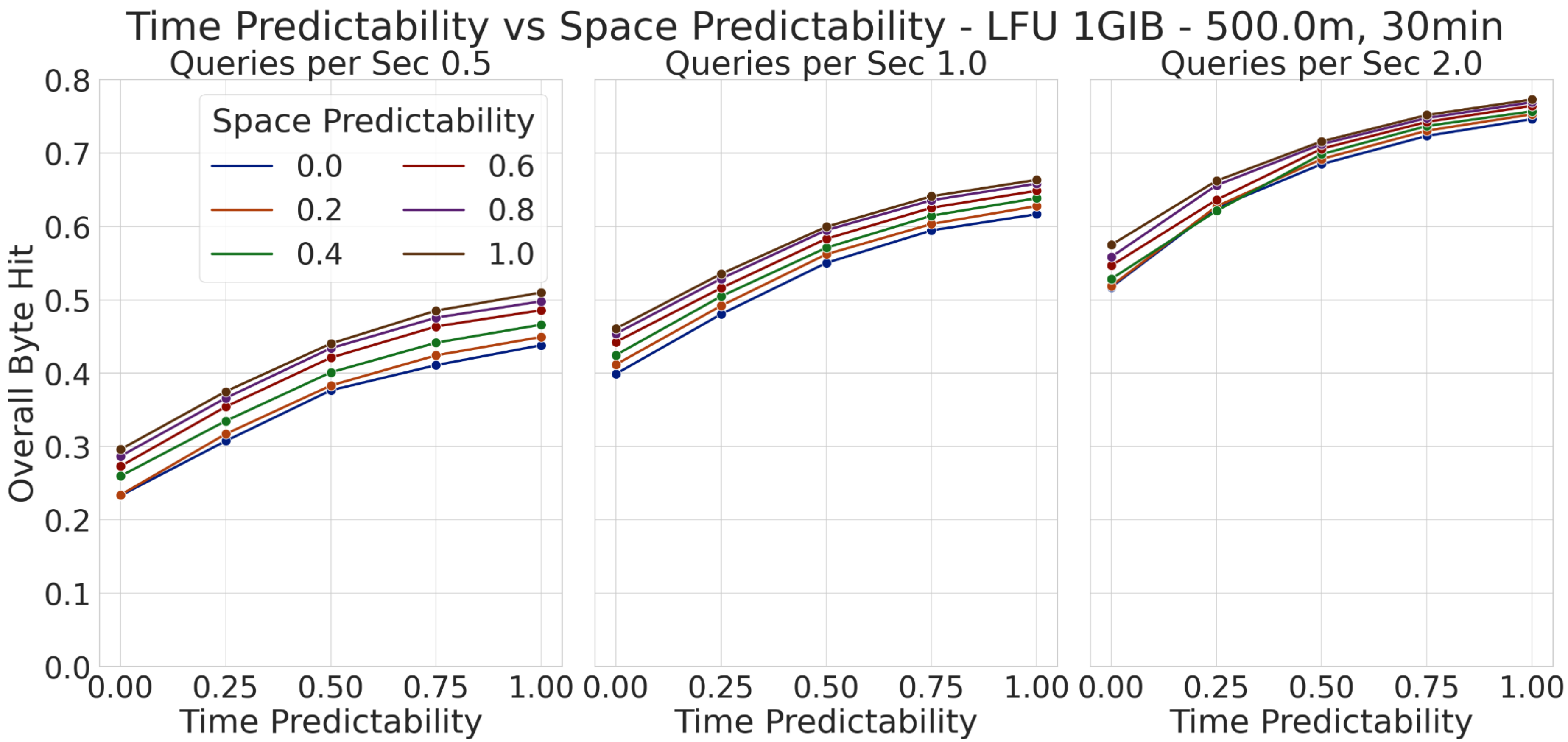


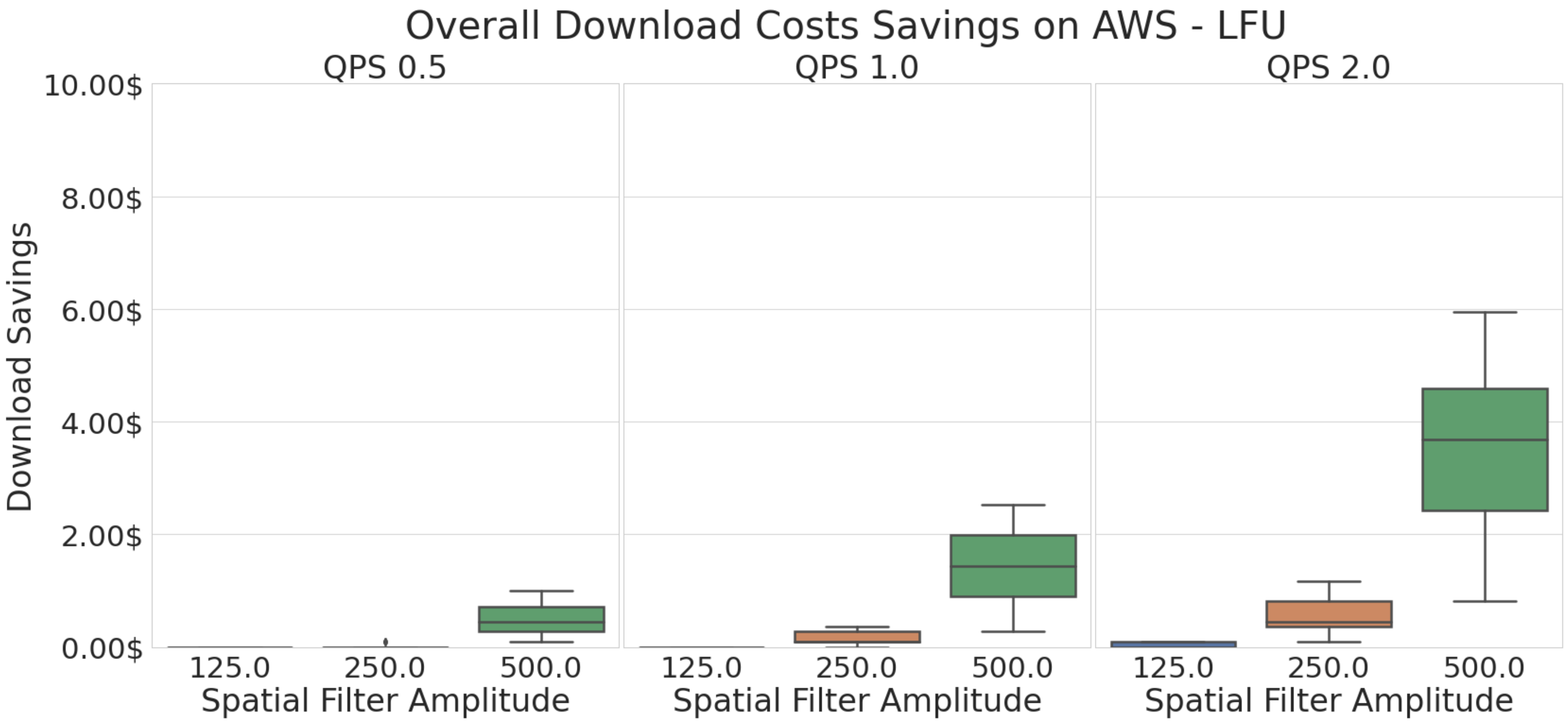
# LRU - Least Challenging Scenario



# LFU – Lest Challenging Scenario



# Savings with LFU



# BIR –LFU

At Cloud
(lower is better)

At Fog
(higher is better)