IEEE INTERNATIONAL CONFERENCE ON
COMMUNICATIONS
9–13 June 2024 // Denver, CO, USA
Scaling the Peaks of Global Communications

Dynamic Edge/Cloud Resource Allocation for Distributed Computation under Semi-Static Demands

AUTHORS: IPPOKRATIS SARTZETAKIS, PANAGIOTIS PANTAZOPOULOS*, KONSTANTINOS V. KATSAROS*, VASILIS SOURLAS*, EMMANOUEL (MANOS) VARVARIGOS*‡

SUBSTITUTE PRESENTER: GEORGIOS DRAINAKIS*

*INSTITUTE OF COMMUNICATION AND COMPUTER SYSTEMS (ICCS), ATHENS, GREECE
‡NATIONAL TECHNICAL UNIVERSITY OF ATHENS (NTUA), ATHENS, GREECE

IEEE ICC®
IEEE ComSoc
IEEE Communications Society
IEEE

Outline

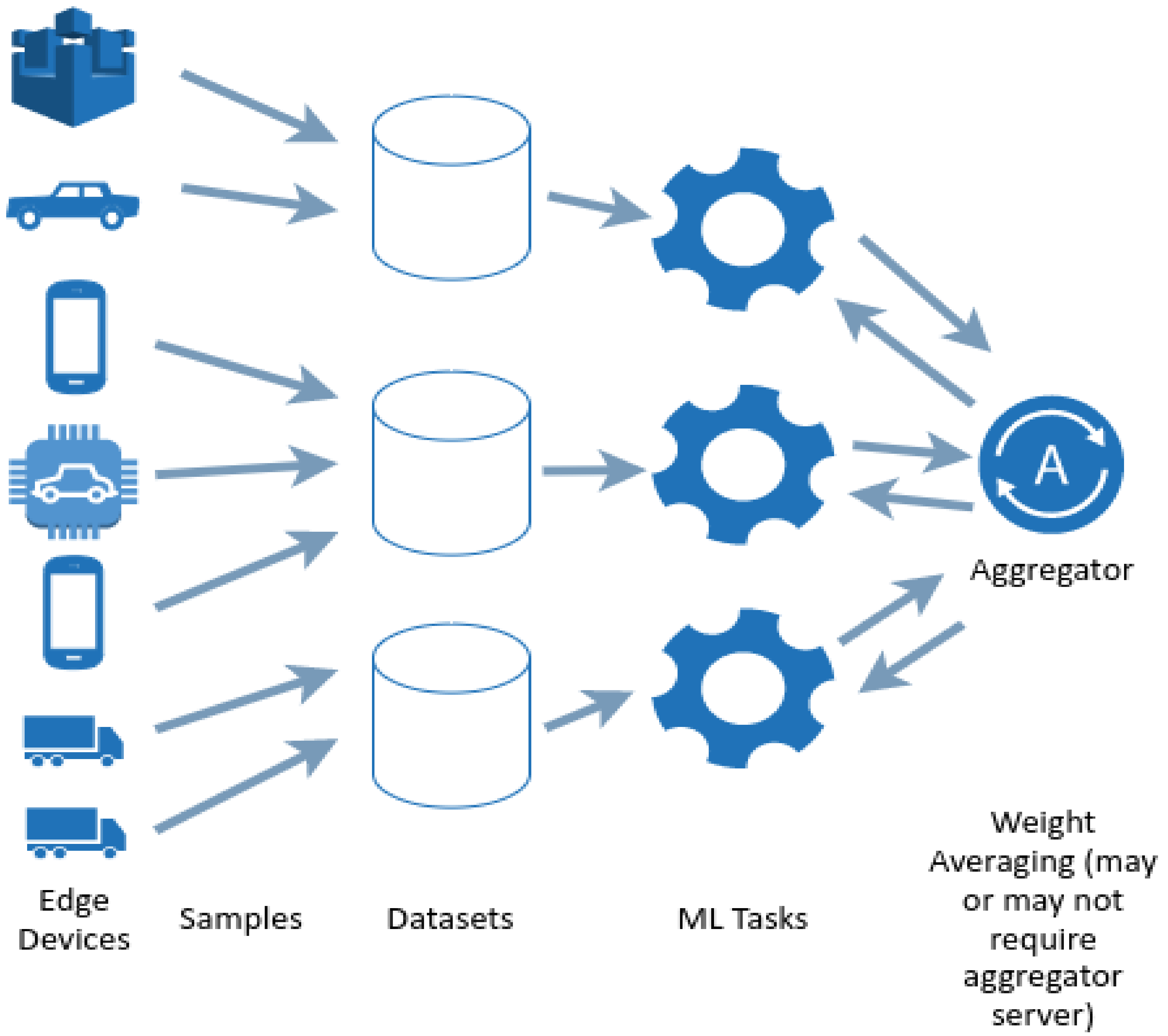
- ❑Motivation
- ❑Distributed Computation
- ❑Network Scenario
- ❑Resource Allocation and Prediction Algorithms
- ❑Results
- ❑Conclusions

Motivation

- New services utilize edge-device data
  - Automotive
  - Industry 4.0+
  - 5G and beyond
- Enormous amount of **time-varying data** with **various processing requirements**
- Centralized processing
  - Processing delays (ML training times)
  - Transmission costs
  - Storage
- **New computing paradigms** e.g., edge-cloud computing
  - An opportunity for **distributed computation** arises

Distributed Computation

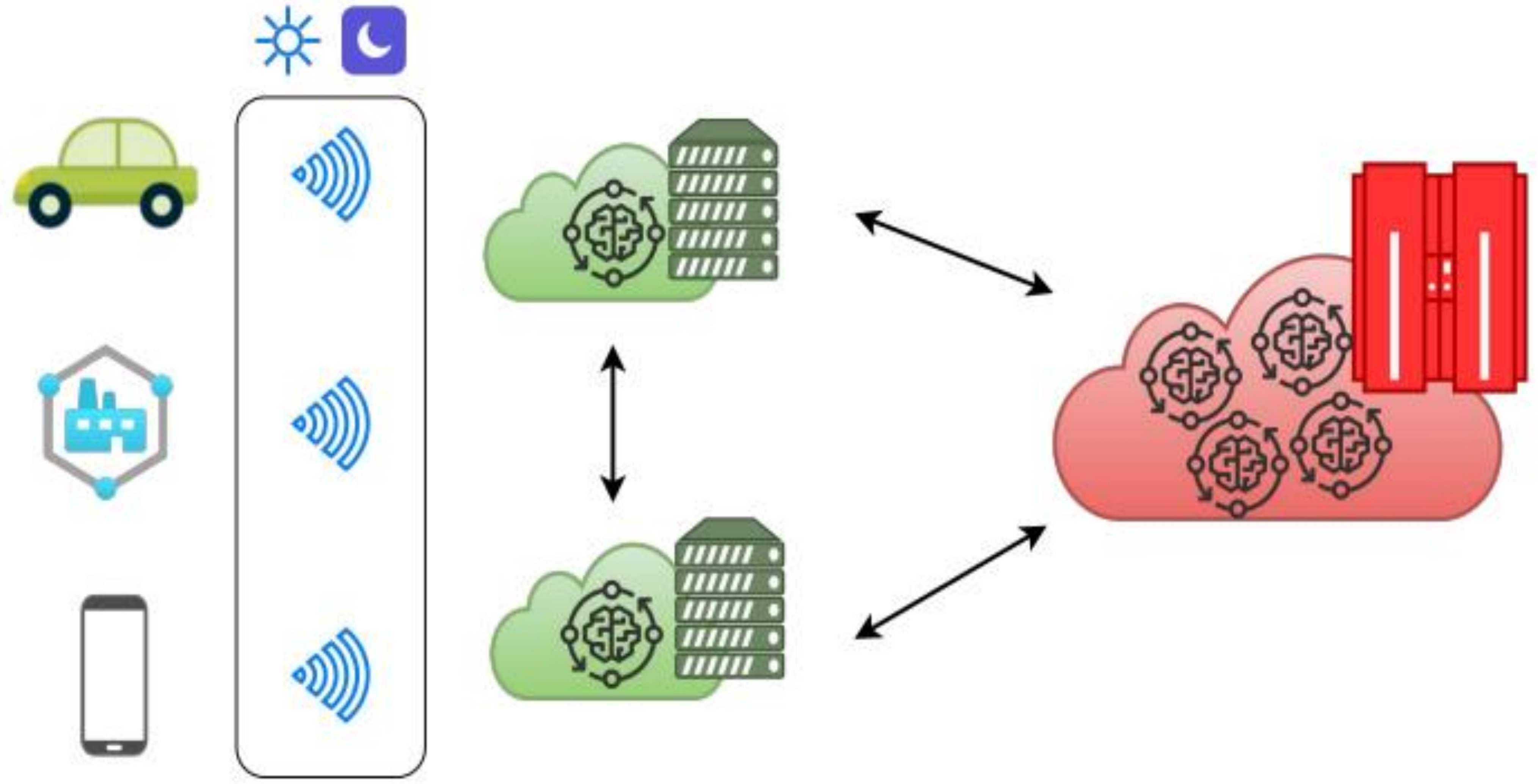
- Processing is performed on dedicated edge/cloud resources
- A job breaks down to several tasks served in a distributed manner
- Advantages:
  - Make use of powerful computation resources
  - Parallelism
- Challenges
  - Allocate the appropriate network resources
  - Specific architectures e.g., distributed ML tasks
  - Job requirements
    - Bandwidth
    - Processing cost
  - The formulation is more complicated assuming **time-varying data generation**



Contribution

- ❑ Developed resource allocation model for Distributed Computation jobs assuming time-varying demands
  - Jointly considered
    - edge and cloud resources
    - their performance
    - bandwidth and processing monetary costs
- ❑ We consider:
  - a multi-period Integer-Linear-Programming (ILP) algorithm to plan periodic demands
  - a predictor that estimates temporary data volume fluctuations
  - a suitable dynamic reconfiguration algorithm
- ❑ Performed realistic simulations and compared to alternative solutions

Network Scenario



- ❑ Edge devices continuously produce data
- ❑ Data generation is time-varying:
  - periodic/expected (e.g., during a day)
  - Or unexpected due to (a sequence of) certain events
- ❑ Edge and cloud network
- ❑ The edge network consists of a set of nodes N with finite resources
- ❑ Edge and cloud have different b/w and processing costs
  - Edge has inexpensive b/w and expensive proc.
  - Cloud has expensive b/w and inexpensive proc.
- ❑ Resources to be assigned:
  - CPU/GPU, b/w for specific computation accuracies

Resource allocation for periodic demands (1)

- ❑ Assumptions
  - Each **device continuously produces data** at an average rate measured in samples/sec
  - The **average rate remains stable** (or constrained by a max value known beforehand) during a number of periods (e.g., three periods during a day)
  - Each task has to **process all the samples** from its devices
  - Each task **requires specific processing and b/w**
    - depending on the number of its samples and the requested accuracy
- ❑ Resource allocation objective:
  - Allocate the appropriate **resources**
    - for all the jobs
    - for all the assumed time periods
  - **Minimize** the total (b/w and processing) **cost** of edge and cloud to serve all the jobs
  - **Maximize** the computation **accuracy**

Resource allocation for periodic demands (2)

- The resource requirements of the jobs are not constant
  - Periodic changes throughout the day
  - Non-periodic fluctuations
- Periodic changes are not very large and frequent.
  - During a 24-h period we can have 2-3 time re-configuration sub-periods
  - ILP resource allocation during sub-periods
- Short-term fluctuations due to special circumstances, e.g., a football game.
  - Short term predictor for bursty changes
  - A heuristic algorithm that reconfigures the demands based on the prediction

ILP Resource allocation algorithm

Symbol	Description
J	Set of jobs
T_j	Set of tasks of job j
λ_{je}	Production rate of task je in samples/sec j
N	Set of node of edge network
$R_n^G, R_n^B, R_n^{\Theta}$	Set of processing, b/w, aggregation resources of edge node n
$C_E^G, C_E^{bw}, C_C^G, C_C^{bw}$	Processing and b/w costs at the edge and cloud respectively
δ_c	Propagation delay of cloud
Δ_j	Acceptable prop. delay of job j
W	Weight to control optimization objective
A	Set of possible accuracies of ML jobs
a_j	An accuracy of a job j ranging from 0 to 1
a_j^{min}	The minimum acceptable accuracy of a job j
ξ_n^{pjea}	Binary variable equal to 1 if task je is served at node n, period p, accuracy a
ξ_c^{pjea}	Binary variable equal to 1 if task je is served at period p, accuracy a
k	The total monetary cost to serve all jobs
S	A set of jobs that must not migrate locations from one period to another
PC	A set of all possible combinations of successive periods p, p'
$m_{pp'}^{je}$	The migration cost of each task je from a period p to a period p'

- Objective:

$$\min\left(w_1 k - w_2 a + w_3 \sum m_{pp'}^{je}\right)$$

  - multi-criterion optimization problem
  - minimize the **total cost** to serve the jobs
  - minimize the **migration cost** (tasks moving from one location)
  - maximize **accuracy**

- Monetary cost – sum of:

$$k = \sum_j \sum_{t_{je}} \sum_a \sum_p (\sum_n \xi_n^{pjea} \lambda_{je}(C_E^{bw} B^{pjea} + C_E^G G^{pjea}) + \xi_c^{pjea} \lambda_{je}(C_C^{bw} B^{pjea} + C_C^G G^{pjea}))$$

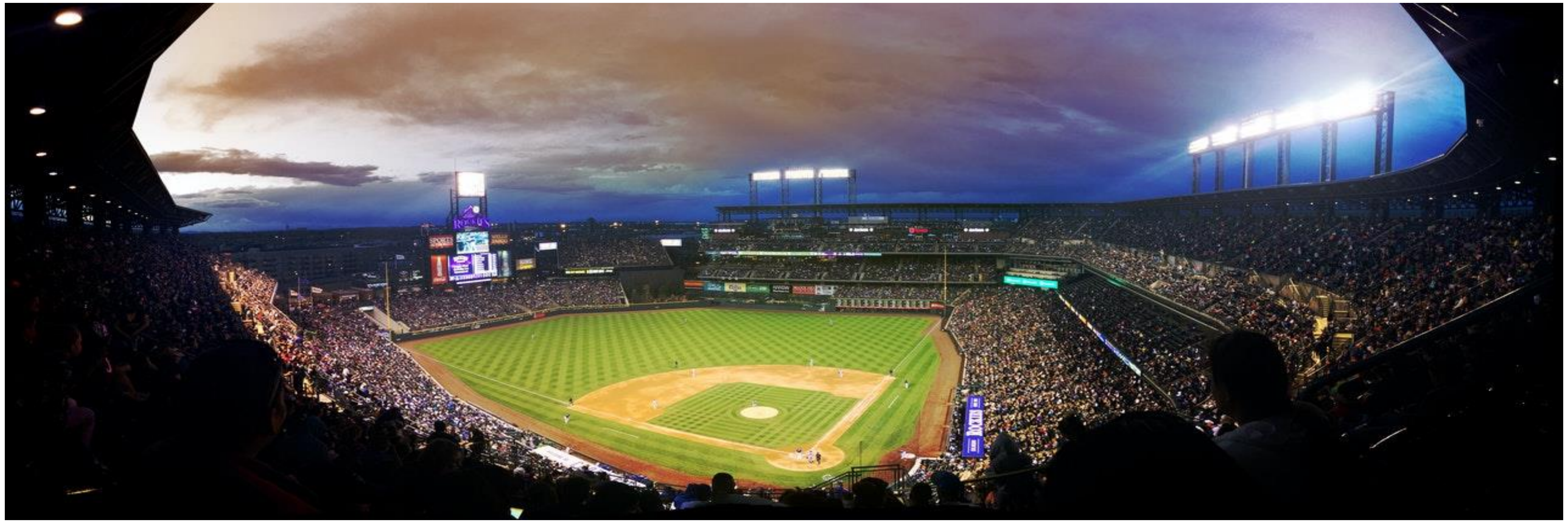
  - edge and cloud **bandwidth** (b/w)
  - plus the edge and cloud **processing cost**
  - for all the task jobs, for all the accuracy options and for all the periods

- Accuracy – mean accuracy of all tasks

$$a = \sum_j \sum_{t_{je}} (\sum_n \xi_n^{pjea} \alpha_j + \xi_c^{pjea} \alpha_j)$$

Traffic prediction algorithm (1)

- Data generation can have unplanned variations due to special events e.g., a football match
- We employ a traffic predictor
  - Input: historical data
  - Output: estimates a number of future time steps



Traffic prediction algorithm (2)

- Prediction objective
  - Data generation rate
  - Required resources for each task
- Several prediction algorithms
  - Auto-regression
  - Traditional ML techniques e.g., random forest
  - Deep NNs e.g., LSTM
- Refs

  - A. S. Weigend, "Time series prediction: forecasting the future and understanding the past," Routledge, 2018.
  - N. I. Sapankevych, S. Ravi, "Time series prediction using support vector machines: a survey," IEEE Computational Intellig. Mag., 4(2), 2009.
  - Y. Hua, et al., "Deep learning with long short-term memory for time series prediction," IEEE Comm. Mag., 57(6), 114-119, 2019.

Reconfiguration Algorithm

- Uses the estimated (future/projected) requirements as input
- If the allocated resources are **not** sufficient -> reallocates the resources w.r.t.
  - Heuristic approach - Avoid moving tasks to different locations
    - Unless necessary (according to the SLAs)
    - and/or reconfiguration costs (e.g., % change of resources, additional monetary cost etc.)
- When the requirements return to the normal planned values
  - the algorithm releases the additional resources
  - preserves the location of the tasks

Results

IEEE ICC IEEE ComSoc

❑Setup

- We assumed **a 10-node edge network** with finite resources
- Two scenarios: [400, 600] **image recognition ML jobs** with varying image size
- Modelling
  - Realistic training performance (NVIDIA MLPERF benchmarks)
  - Realistic cloud processing and b/w costs (AMAZON EC2)
- Two accuracies (good, low), three time periods for ILP to plan with varied traffic
- The unplanned variations result in 20% traffic increase

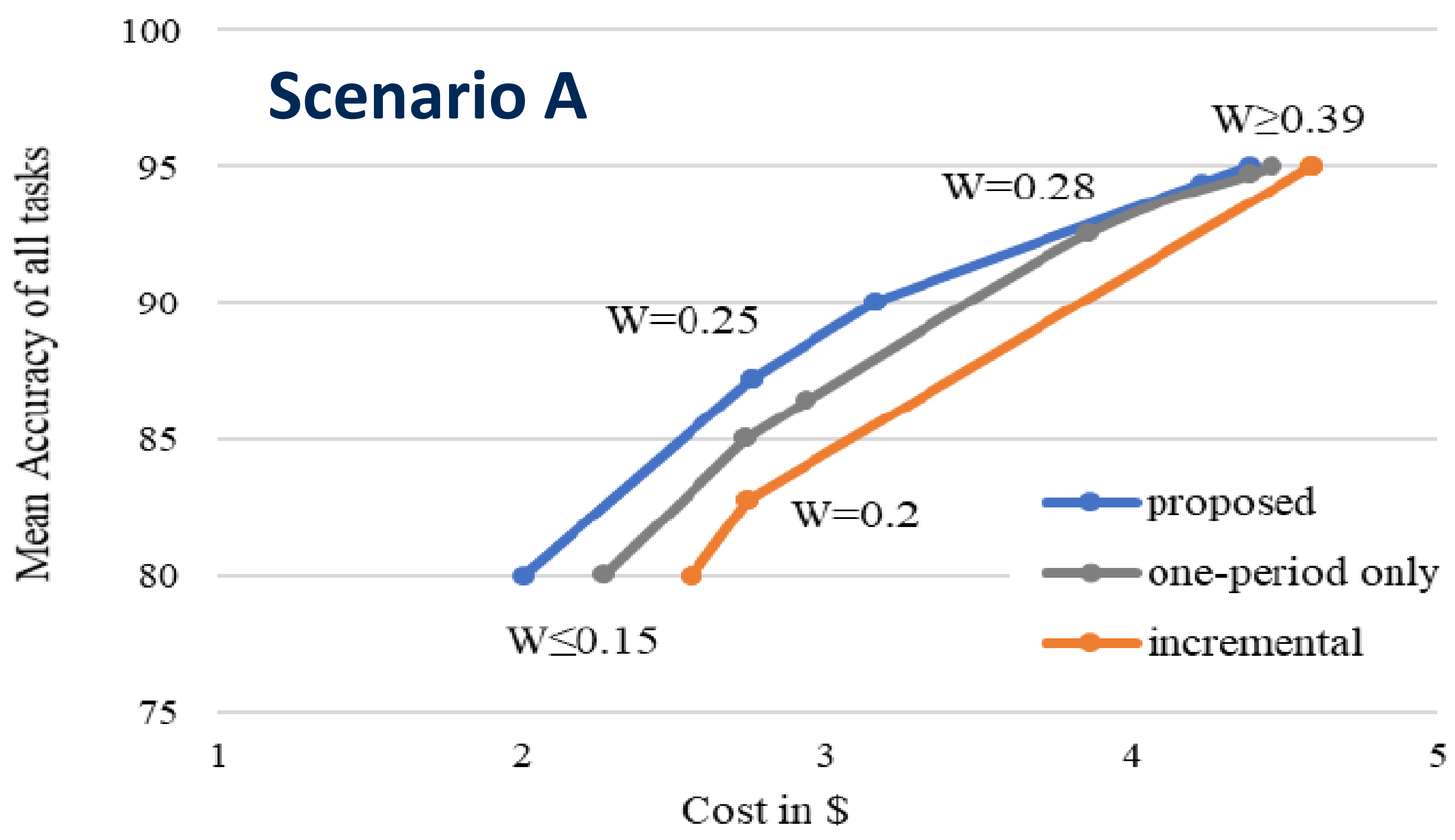
❑Simulation environment:

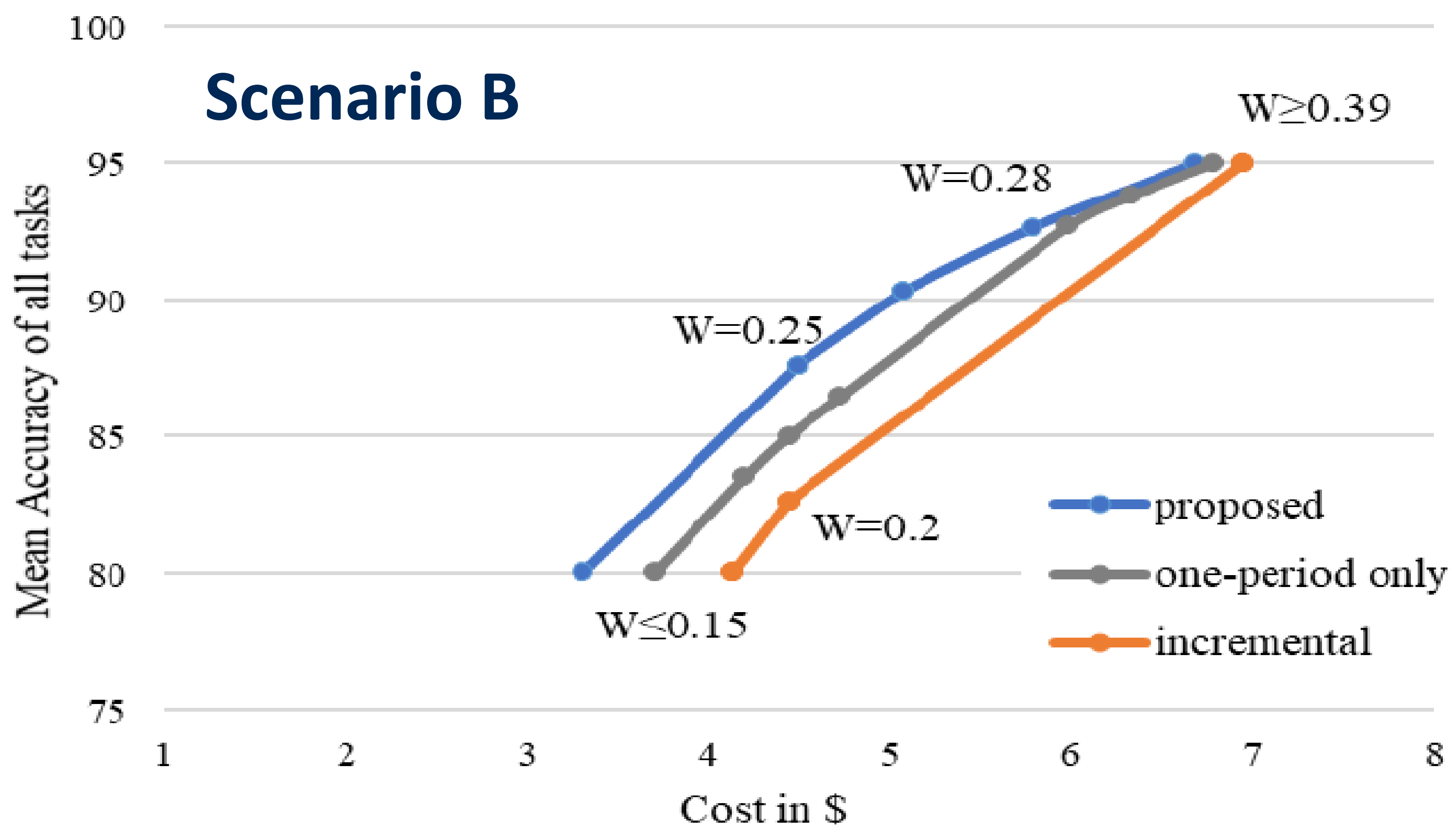
- Pyomo (Python) and IBM CPLEX: **2 secs to solve ILP** on a quad core CPU@4GHz

❑Compare against SotA

- An algorithm assuming only one period planning; the rest of the demands are incrementally served
- An algorithm that incrementally and greedily serves demands one-by-one

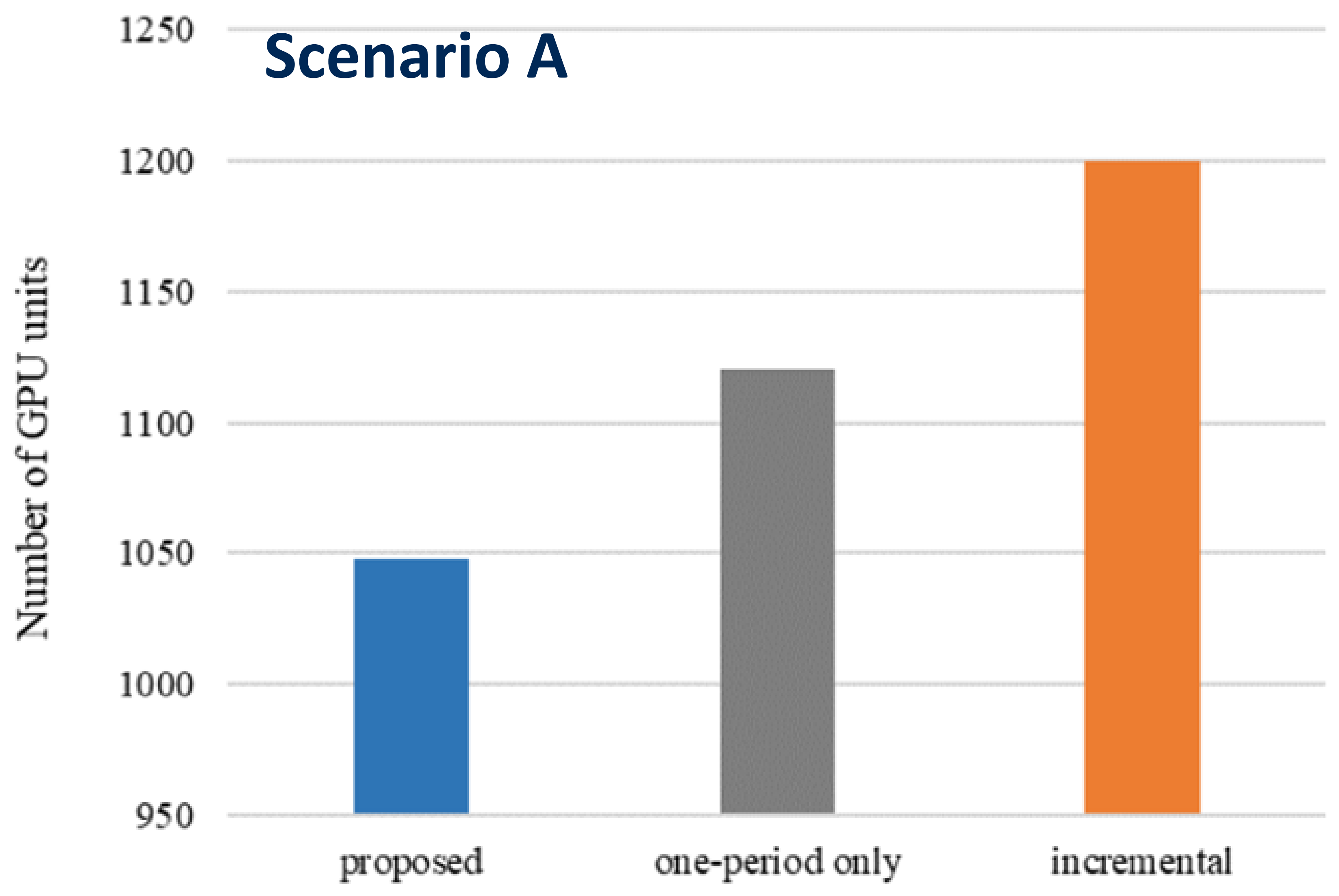
Results (Accuracy vs. monetary cost)

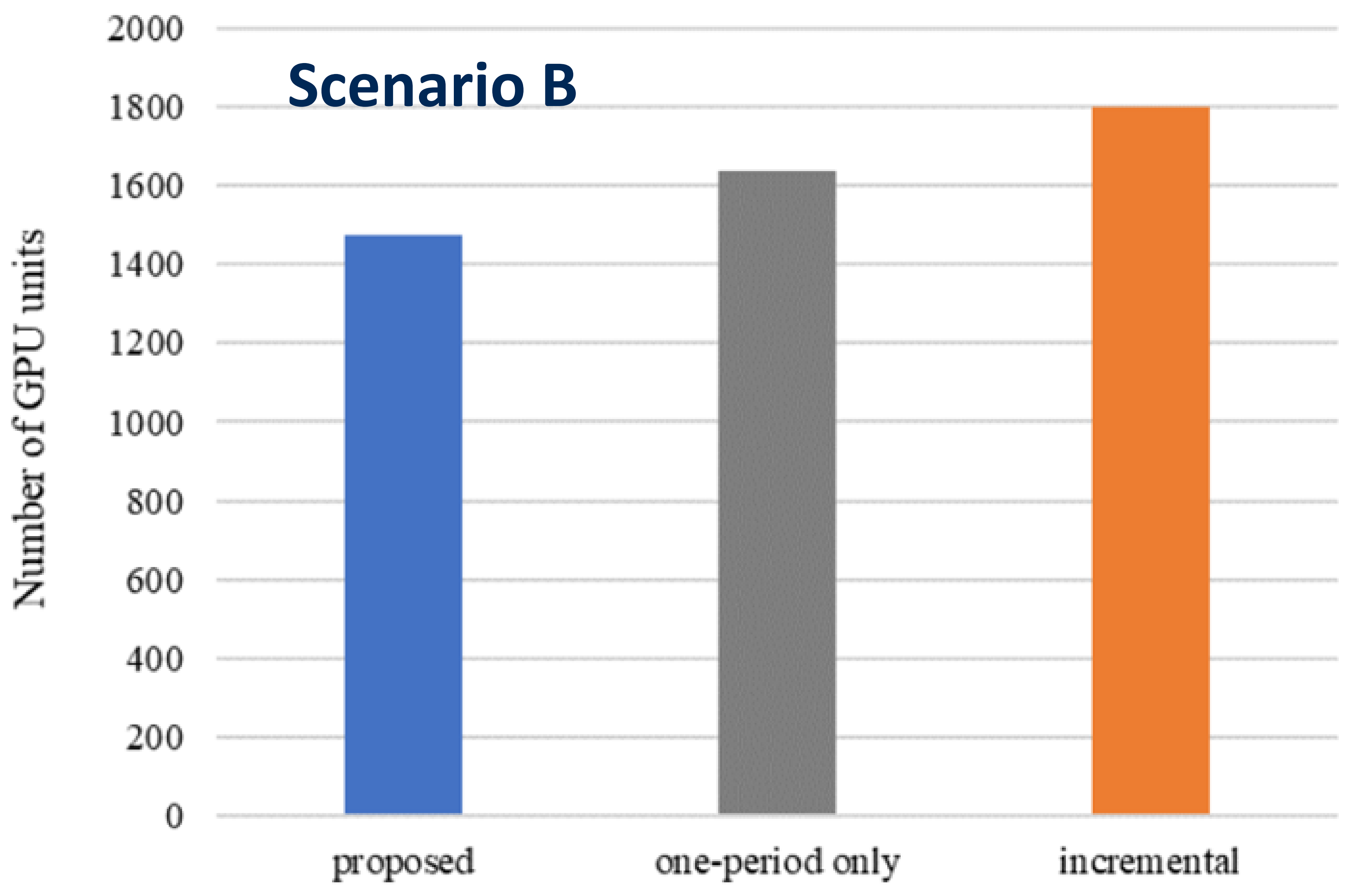




- Proposed algorithm achieves the **best accuracy** coupled with the **lowest monetary cost** in both scenarios
- Planning algos have complete view of all demands and make **optimal placement decisions** based on the overall objective
- Larger accuracy targets require expensive allocation decisions
  - Little room for improvement by placement optimization
  - Negligible differences between the algos
- Scenario B results in better savings for our proposal
  - Additional jobs create more opportunities for better job placement

Results (processing utilization)





- Common target accuracy for all algos – examine GPU utilization
- Scenario A: 12.6% and 6.4% less GPU units of our algo compared to the two SotA algos (incremental and one-period solutions)
- Scenario B: Slightly higher savings of our solution
- The results translate to **less energy** and **fewer resources** to achieve the same output

Conclusion & Future Work

IEEE ICC®

IEEE ComSoc®

❑Summary

- We considered the resource allocation problem for distributed computations at edge/cloud in the context of (non) periodic demands.
- We presented a **planning algorithm** that serves the periodic semi-static demands. We also proposed a **traffic predictor** and a **reconfiguration algorithm** that serves the unexpected demands.
- We performed a number of **realistic** simulation experiments.
- Against 2 SotA solutions under 2 scenarios:
  - best accuracy with the lowest monetary cost for medium accuracy targets
  - less GPU utilization to achieve the same output

❑Next steps

- Generalize results on several scenarios/configs
- Cross-validation measurements on a real 5G-testbed

Questions?

Thank you!

Drainakis Georgios, Software Research Engineer

giorgos.drainakis@iccs.gr

Institute of Communication & Computer Systems (ICCS)

Iroon Politechniou str. 9, NTUA Polytechnic Campus

15773 Zografou, Athens, GR

www.iccs.gr