Future Data-Aware Decision Making for Edge Computing

Dr Konstantinos (Kostas) Kolomvatsos

Monday, September 26th, 2022

School of Computing Science, University of Glasgow

SICSA DVF funding call

University of Glasgow

sicsa* The Scottish Informatics & Computer Science Alliance

At a Glance

2013
PhD in Computer Science
National and Kapodistrian
University of Athens

2016 Call
Marie Skłodowska-Curie
Action (MSCA)
Individual Fellowship
School of Computing Science
University of Glasgow

June 2020
Assistant Professor
Department of Informatics
and Telecommunications
University of Thessaly
http://kostasks.users.uth.gr

July 2020
Founder of the
Intelligent Pervasive Systems
(iPRISM) Research Group
http://www.iprism.eu

Oct. 2020
Co-Founder of the Intelligent
Systems for Orchestrating
Pervasive Computing Applications
(METIS) Research Lab
http://metis.cs.uth.gr

Dec. 2020
Director of the METIS Lab

Current Activities:
- Applied Artificial Intelligence and Machine Learning
- Distributed Intelligence
- Pervasive Data Science

At a Glance

Intelligent Pervasive Systems
(iPRISM)
http://www.iprism.eu

Lead: Dr Konstantinos (Kostas) Kolomvatsos

Research axes:

- Artificial Intelligence
- Applied (Deep) Machine Learning
- Computational Intelligence
- Distributed Intelligence
- Pervasive Computing
- Pervasive Data Science
- Proactive Decision Making
- Applications for Distributed Systems, Internet of Things, Edge Computing
- Predictive Intelligence
- Large Scale Data management

Research Projects

Subjects

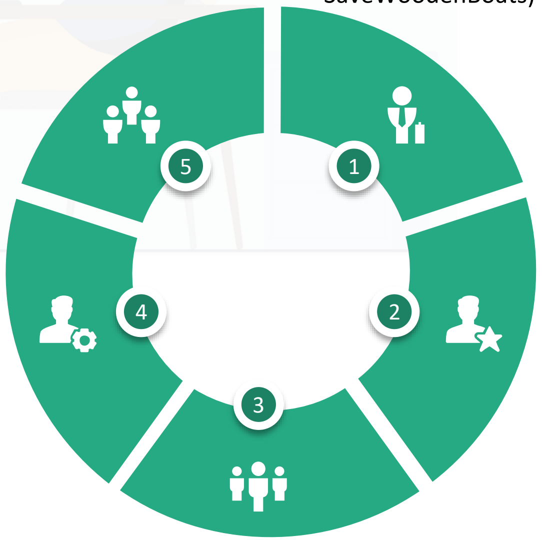
- Sensors Data Management
- Situational Awareness for Emergency Management
- Intelligent Decision Making
- Deep Learning for Image Processing
- Analytics
- Digitization
- Augmented Reality

iPRISM

- SILVANUS H2020
- ELLIOT H2020 Open Call
- PLATON National Initative
- SaveWoodenBoats National Initiative
- Digitization of Cultural Material Network of Museums
- Intelligent Waste Management Lamia Municipality

Outcomes

1. Heatlh Impact (SILVANUS, ELLIOT)
2. Evacuation Planning (SILVANUS)
3. Recommender (ELLIOT)
4. Chatbot, Drug impact assesment (PLATON)
5. Virtual Wolrd for presenting information and training (PLATON, SaveWoodenBoats)



Recent Research

01 Intelligent Systems in Pervasive, Edge Computing and Internet of Things

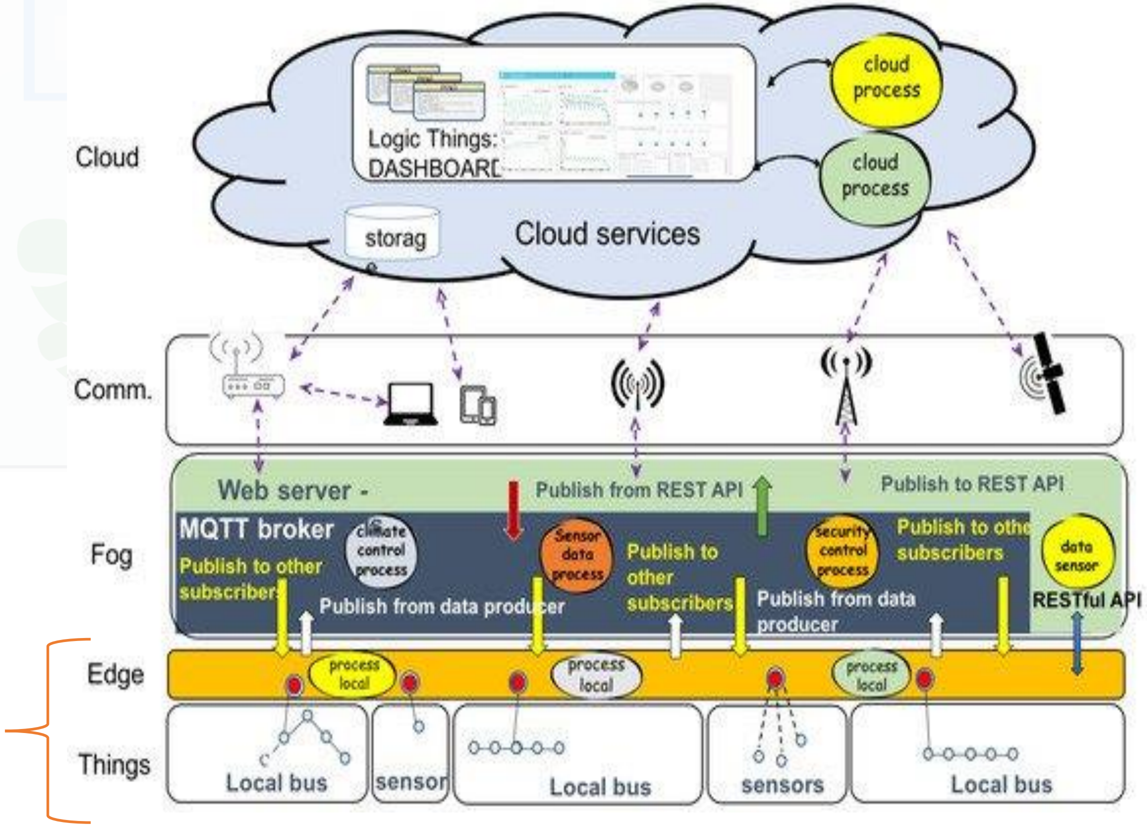
02 Contextual and Fuzzy Logic Reasoning for Pervasive Computing

03 Proactive Reasoning for Autonomous Behaviour and Decision Making

04 Pervasive Data Science Applications

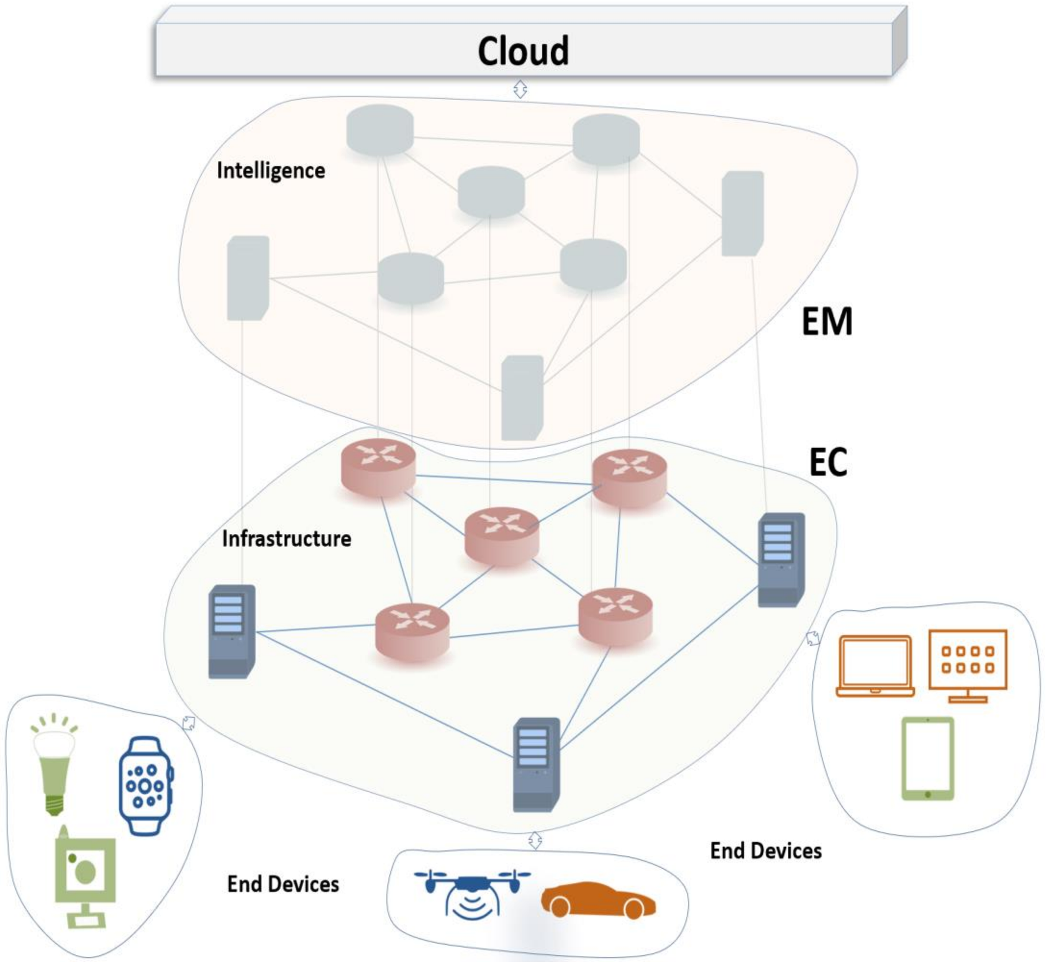
Edge Computing

- Edge Computing (EC) deals with an additional infrastructure above the Internet of Things (IoT)
- EC 'imposes' an ecosystem of processing nodes that can execute tasks upon the collected data
- Gartner shared a report on ten (10) strategic trends affecting the Internet of Things (IoT) from 2019 to 2023 and beyond where the following are identified as the most impactful:
  - Artificial intelligence (AI)
  - The shift from intelligent edge to **intelligent mesh**
  - New IoT user experiences



Edge Computing

- We are at the early stages of the EC revolution to prepare the infrastructure for the new, modern, **Edge Mesh** (EM)
- EM provides a 'virtual' layer (a computational/processing overlay) that enables the cooperation between heterogeneous EC nodes to conclude a cooperative infrastructure close to end users
- Operators can/should/will open the ecosystem to third-parties, allowing them to rapidly deploy innovative applications and content



State of the Art

Cloud

- ✓ Tasks are processed at the central server that will send the response back to the device
- ✓ The whole process typically takes less than a second, however, there might be delays, e.g., due to a network glitch, weak internet connection, or long distance with the datacentre

EC

- ✓ Every EC node is responsible for processing and can respond to any request

EC Nodes

- ✓ EC nodes are:
  - resource constrained
  - heterogeneous
  - subjects to dynamic workloads change

EM

- ✓ EM is a computation overlay network over the EC nodes
- ✓ EM overcomes the problem of constrained resources through a cooperative model
- ✓ A mesh network of nodes enables distributed decision-making, sharing data and computation

Intelligent EM

- ✓ AI/ML are required to support intelligence
- ✓ Distributed intelligence is the key enabler for cooperation and optimal decision making

Research Questions

How to define the network and computing model?

How to distribute data processing?

How to jointly optimize computation?

How to be stateful, i.e., exhibit different behaviour even for the same data according to the conditions met at a time instance?

Challenges

Modelling Challenges

Support the creation/migration/replication of virtual resources at different levels of granularity

---

The demand for edge resources (even they are from peer nodes) have to be modelled (based on the traffic generated, this challenge is complex)

Mobility could result in the migration of services/data from one node onto another (Follow me Edge) or their replication (we define the Proactive Edge)

Coordination Challenges

Enable the required coordination that facilitates the management of geographically distributed devices

---

Facilitate fast migration/replication of abstract entities (such as functions or programs) or data

---

Evolution of a new model of multi-locational hybrid (edge & Cloud) data architectures

---

Exhibit the necessary intelligence for the proactive response to potential problems

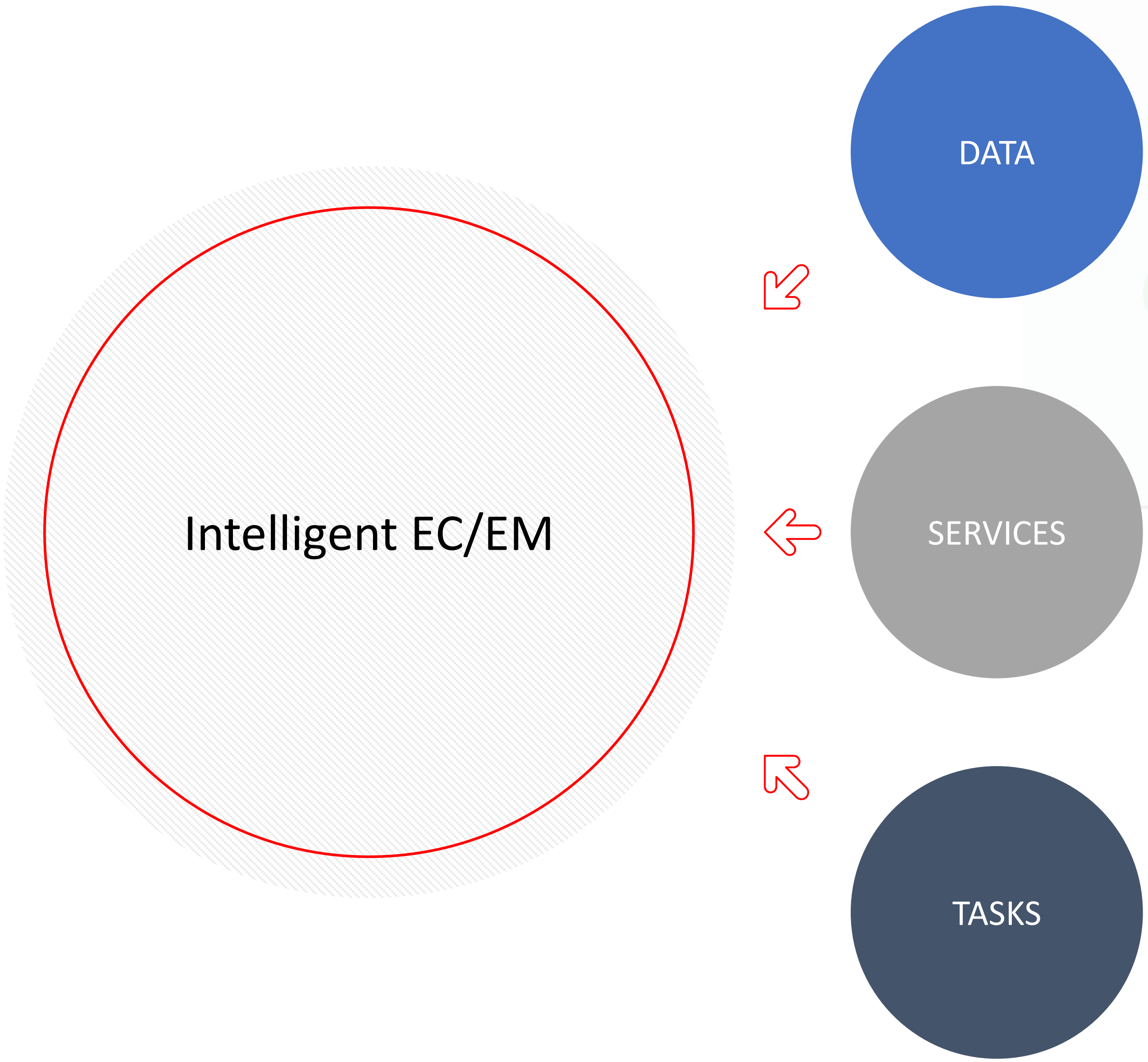
Recent Publications

Kolomvatsos, K., 'A Proactive Inference Scheme for Data-Aware Decision Making in Support of Pervasive Applications', Future Generation Computer Systems (FGCS), Elsevier, 136, 2022, pp. 193-204

Kolomvatsos, K., Anagnostopoulos, C., 'A Proactive Statistical Model Supporting Services and Tasks Management in Pervasive Applications', IEEE Transactions on Network and Service Management, 2022, doi: 10.1109/TNSM.2022.3161663

Kolomvatsos, K., 'Data-driven Type-2 Fuzzy Sets for Tasks Management at the Edge', IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 6, no. 2, pp. 377-386, April 2022

Research Axes



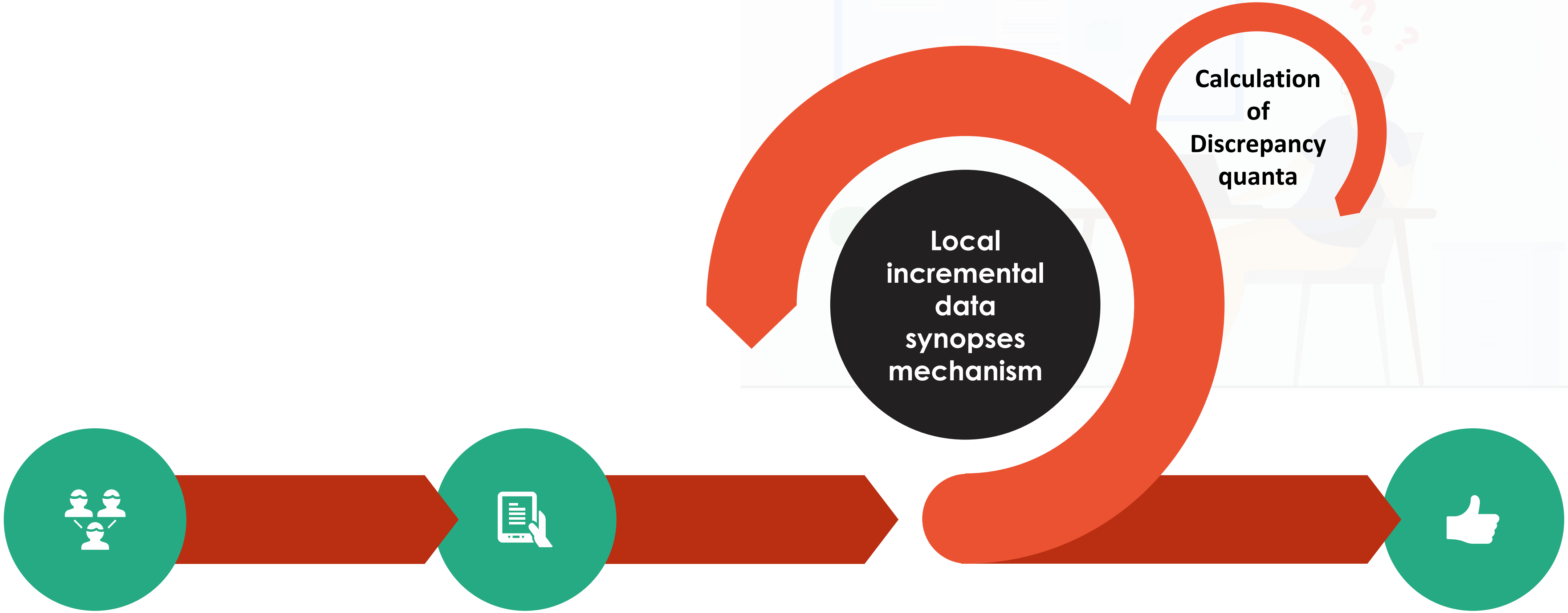
- ✓ A data map of the ecosystem in support of data-aware decision making
- ✓ Proactive inference of nodes ‘matching’ based on their data

- ✓ Statistical inference upon the demand for services and the contextual performance data of nodes
- ✓ Utility aware decision making model

- ✓ Tasks offloading based on uncertainty-driven decision making model
- ✓ Data-driven automated inference mechanism

Research Axis A
DATA

Data-aware Matching Inference



IoT devices collect and send data in an upwards mode

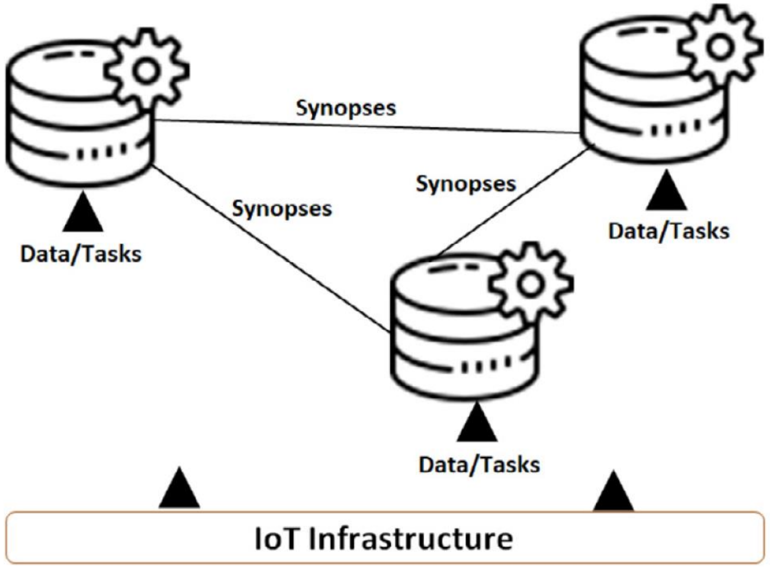
EC nodes receive data and process them

To reduce network overhead, EC nodes share data synoipses

Statistical and correlation inference for peers

Data-aware Matching Inference

- EC nodes, at regular intervals, exchange the calculated synopses
- Without loss of generality, we consider that at t, a node n_i receives N – 1 synopses $\{s^t_j\}$ $\forall j, j \neq i$.
- n_i continuously monitors the discrepancy quanta with peers
- The discrepancy quantum d^t_{ij} is calculated as the absolute value of the difference between the j^{th} synopsis s^t_j and the local synopsis s^t_i
- We generate the time series $d^1_{ij}, d^2_{ij}, \ldots, d^W_{ij}$ (sliding window) upon which the proposed 'inference process' is applied



Data-aware Matching Inference

Random Variables

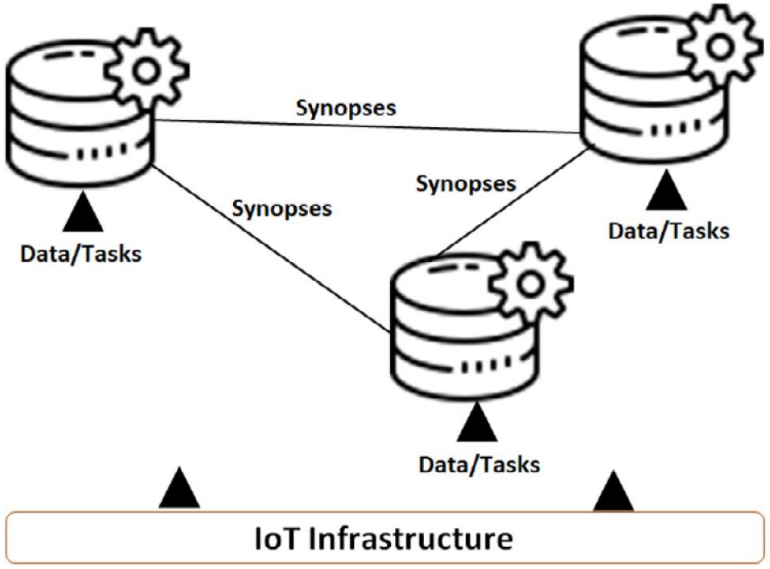
S depicts the realization of the synopsis of a specific data dimension k

$$D_{ij} = \sum_{k=1}^{M} Z_k$$

$$Z_k = |S_{ik} - S_{jk}|$$

Expected Synopses Difference

Lemma. *The expected difference between the synopses calculated by the ith node and the jth peer is given by* $\mathbb{E}\left(D_{ij}\right) = \sum_{k=1}^{M} 2 \cdot A_k - \mu_{ik} - \mu_{jk}$ *with* $\mu_{ik} \& \mu_{jk}$ *being the mean of the kth dimension in the ith and the jth synopses and* $A_k = \int_{-\infty}^{+\infty} s \cdot \left[f_{S_{ik}}(s) F_{S_{jk}}(s) + f_{S_{jk}}(s) F_{S_{ik}}(s)\right] ds.$



Expected Difference of the Discrepancy Quanta

Proposition. *The expected discrepancy quantum when data follow an Exponential distribution with the same rate* λ *is given by* $\mathbb{E}\left(D_{ij}\right) = \frac{M}{\lambda}.$

Proposition. *The expected discrepancy quantum when data follow an Exponential distribution with different rates is given by* $\mathbb{E}\left(D_{ij}\right) = \sum_{k=1}^{M} \left(\frac{\lambda_{ik}+\lambda_{jk}}{\lambda_{ik}\lambda_{jk}} - \frac{2}{\lambda_{ik}+\lambda_{jk}}\right).$

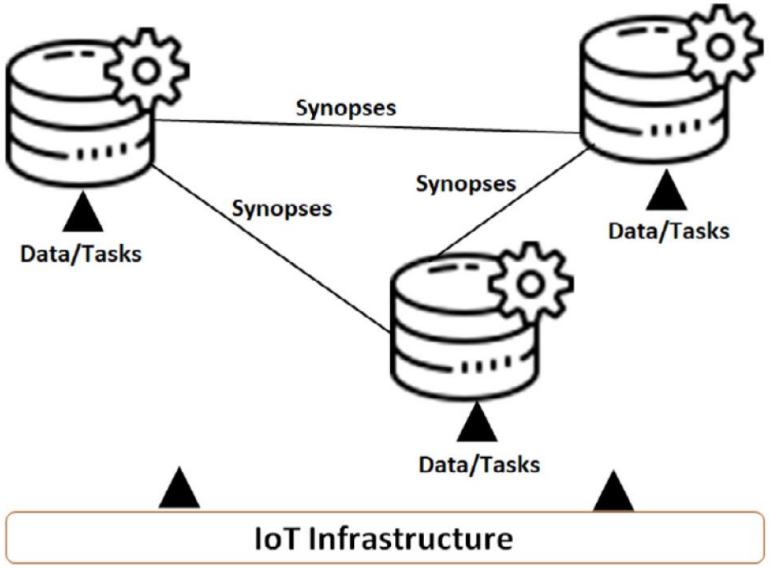
Data-aware Matching Inference

- ✓ The expected discrepancy quantum depicts the anticipated value of the difference between synopses at some point in the future
- ✓ We combine such knowledge with the historical correlation of synopses to depict the trend of the discrepancy quanta
- ✓ We adopt the known Pearson Correlation Coefficient (PCC) r_k

$$r_k = \frac{\sum_{t=1}^{W} \left(s_{ik}^t - \mu_{ik}\right)\left(s_{jk}^t - \mu_{jk}\right)}{\sqrt{\sum_{t=1}^{W} \left(s_{ik}^t - \mu_{ik}\right)^2} \sqrt{\sum_{t=1}^{W} \left(s_{jk}^t - \mu_{jk}\right)^2}}$$

- ✓ **Ideal scenario**: Observe a positive correlation for the M dimensions
- ✓ **Real cases**: 'Mix' of positive or negative correlations
- ✓ We assume a threshold θ_k over which we consider that the observed correlation is 'acceptable'
- ✓ We record the cardinality of the set $|\{r_k \geq \theta_k\} \forall k|$
- ✓ The set of M indicators that should be aggregated into a single value
- ✓ We rely on a sparsity metric and define the logarithmic sparsity indicator $\rho \in R^+$ which depicts the population of unity values in $\{r_k\}, \forall k$

$$\rho = \sum_{k=1}^{M} log \left(1 + \mathbb{1}_{r_k \geq \theta_k}\right)$$



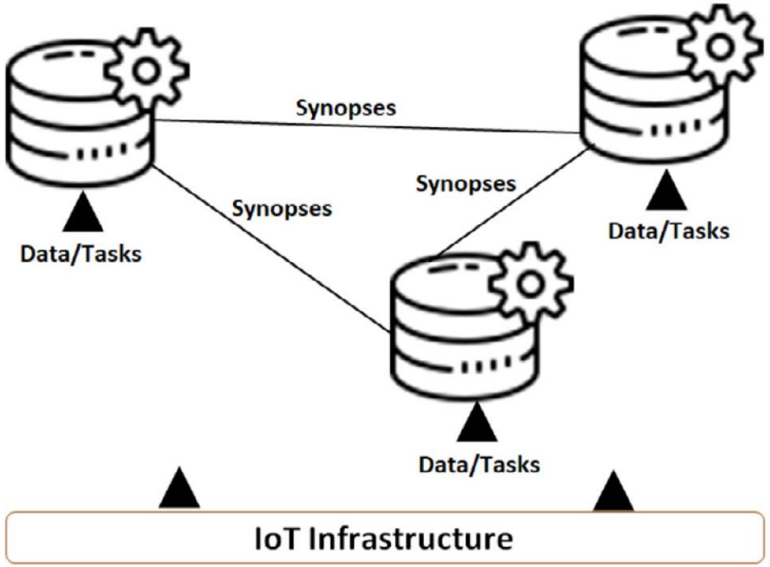
Data-aware Matching Inference

- ✓ We create a temporal matching map and the selection of the appropriate peers for collaborative activities
- ✓ We define the ***Matching Synopses Indicator*** (MSI) **R** which aggregates
  - ✓ The expected discrepancy quantum $E(D_{ij})$
  - ✓ The correlation indicator ρ_{ij}
  - ✓ The communication cost c_{ij}

$$R_{ij} = \frac{e^{-\alpha \mathbb{E}(D_{ij})}}{1 + e^{-\beta \rho_{ij} + \gamma}} \frac{1}{1 + e^{-\delta c_{ij} + \epsilon}}$$

$\alpha, \beta, \gamma, \delta, \varepsilon \in \mathbb{R}^{+}$ are smoothing parameters

- ✓ A sorted list $\{R_{ij}\}$ $\forall j$ is provided in a descending order
- ✓ When required (e.g., to offload a task or 'borrow'/'lend' data from/to peers), every node can interact with peers exhibiting the highest R (a sub-set can be adopted)



Research Axis B
SERVICES

Services Management Scenario

A — EC nodes host a set of services

B — Services are adopted for tasks execution

C — Tasks are reported/requested by end users or applications

D — The mobility and the number of users affect the demand for services

E — A tasks may ‘request’ a service that is not locally present

F — Two choices:
- ✓ Offload the task to peers that host the service
- ✓ Migrate/request the service

G — Proactive mechanism: Migrate services by analyzing the demand in the ecosystem

Services Management Scenario

We propose a model that deals with the decision of

where to migrate

a service

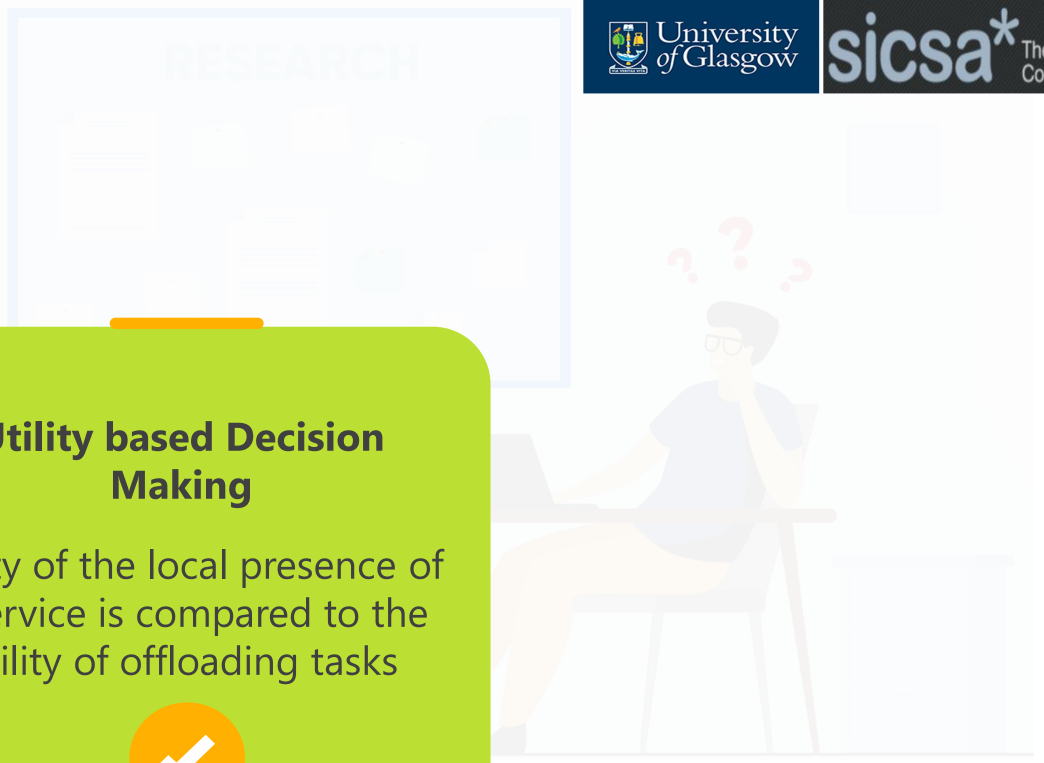
The optimal migration strategy is

intractable due to the dynamics of the EC ecosystem

Tasks offloading can be affected by an additional level of decision and delays in the response

Services migration should be carefully decided due to the resource constraints

Utility Based Model



Statistical Inference

Order statistics for analyzing the demand of a service

Utility based Decision Making

Utility of the local presence of a service is compared to the utility of offloading tasks

Target

Aggregate a statistical inference technique with utility based decisions

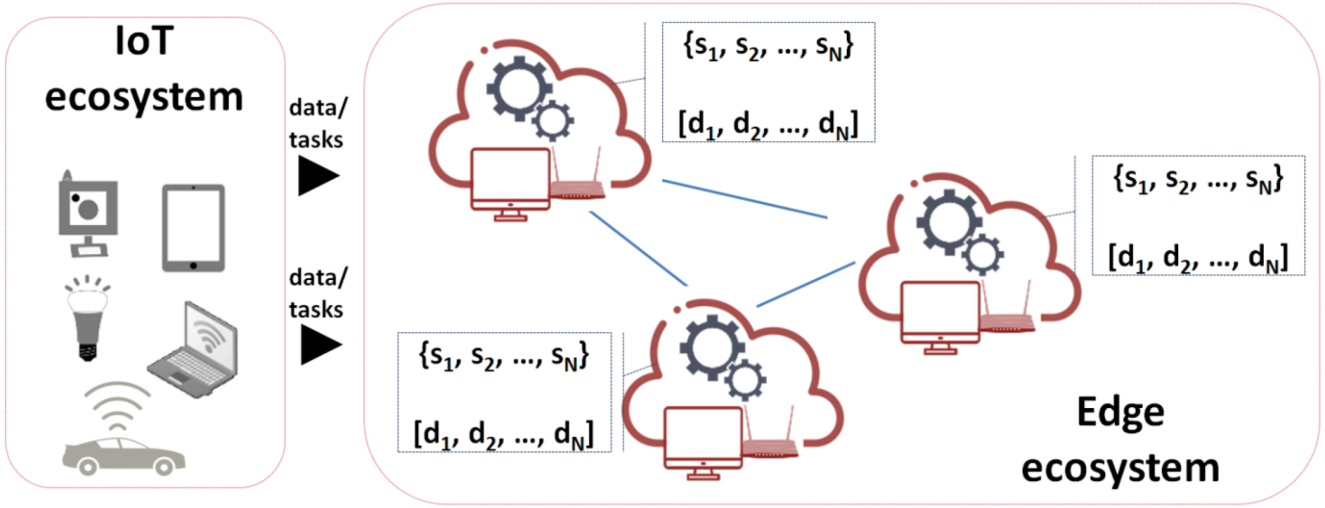
Utility Based Model

- ✓ $\{s_1, s_2, \ldots, s_N\}$: Set of services
- ✓ **Services Demand Vector** (SDV) for the ith node, i.e., $\mathbf{d}_i = \{d_{i1}, d_{i2}, \ldots, d_{iN}\}$
- ✓ After the reception of a task T_{it} , we detect the required services and update the demand
- ✓ **Our approach**: keep the execution of popular tasks locally if the current load is at ‘acceptable’ levels (the node is not overloaded)

Decisions

Decision 1. Keep locally the execution of the task and, if needed, request the necessary services;

Decision 2. Offload the task to the appropriate peer(s).



Utility Based Model

- ✓ d_i & l_i (load) are updated after the arrival of tasks
- ✓ We define the following variables:

$$g = \begin{cases} \frac{\varepsilon}{1+e^{-\gamma d+\delta}} & \text{if } d \text{ is in top-k} \\ 0 & \text{otherwise} \end{cases}$$

$$\hat{g} = \begin{cases} \frac{\hat{\varepsilon}}{1+e^{-\hat{\gamma} d+\hat{\delta}}} & \text{if } l > \hat{\theta} \;\&\; d \text{ is not in top-k} \\ 0 & \text{otherwise} \end{cases}$$

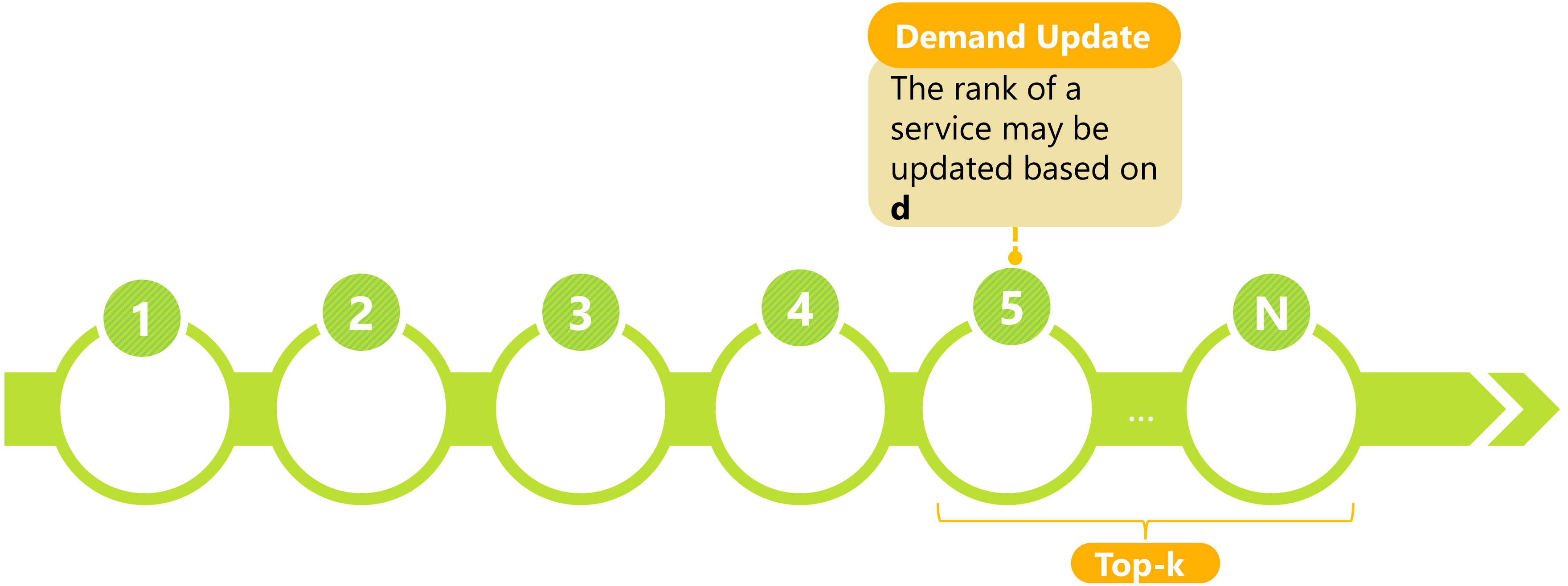
- ✓ g depicts the utility of the local execution of T_i and $\hat{g}$ represents the utility of the offloading action
- ✓ For both decisions, we define the expected utilities U & $\hat{U}$ upon g and $\hat{g}$

Utility Based Model

- ✓ We assume that $D_{(r)}$ is defined upon the random variable D with a random sample of size N and realizations **d** = $[d_1, d_2, \ldots, d_N]$
- ✓ For instance, $D_{(1)} = \min(d_1, d_2, \ldots, d_N)$, $D_{(2)}$ =2nd $\min(d_1, d_2, \ldots, d_N)$ and so on and so forth

Proposition. *The expected utility for the local execution of a task that requires a service with demand d is given by* $\mathbb{E}(G) = \frac{\varepsilon}{1+e^{-\gamma d+\delta}} F_{D_{(N-k)}}(d)$ *where* $F_{D_{(N-k)}}(d)$ *is the cumulative distribution function (cdf) of the variable* $D_{(N-k)}$.

Proposition. *The expected utility for the offloading action of a task that requires a service with demand d is given by* $\mathbb{E}(\hat{G}) = \frac{\hat{\varepsilon}}{1+e^{-\hat{\gamma} d+\hat{\delta}}} \left(1 - F_L(\hat{\theta})\right) \left(1 - F_{D_{(N-k)}}(d)\right)$.



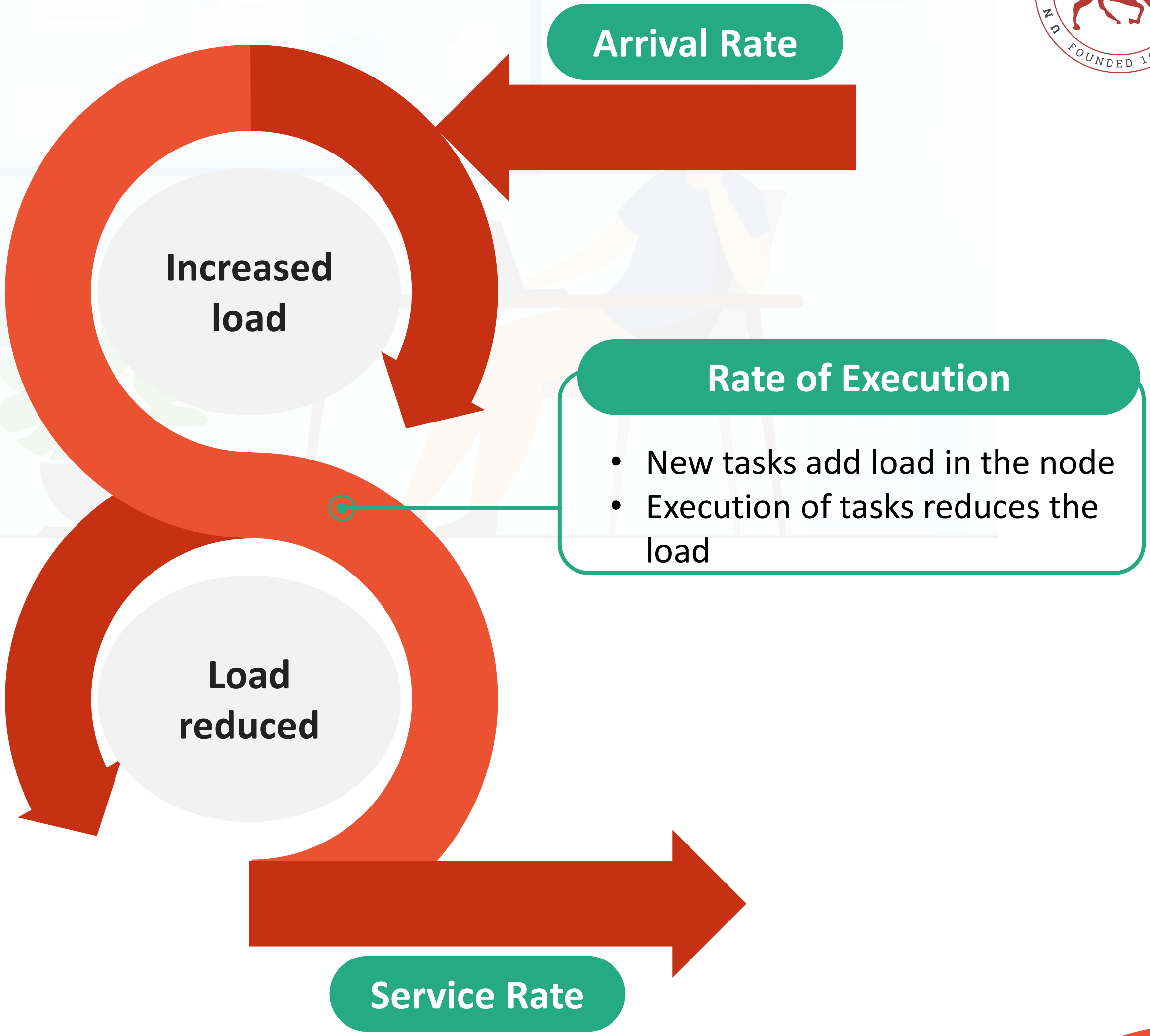
Utility Based Model

Estimation of Load

- ✓ L is the random variable with realizations depicted by l
- ✓ Assume that nodes monitor L over the discrete time while storing the W recent values
- ✓ We adopt the widely known nonparametric Kernel Density Estimation (KDE) method for estimating the cdf and pdf of L

$$\hat{f}_L(l;W) = \frac{1}{W \cdot h} \sum_{j=1}^{W} K\left(\frac{|l - l_{t-W+j}|}{h}\right)$$

$$\hat{F}_L(l;W) = \frac{1}{W} \sum_{j=1}^{W} \frac{1}{2}\left(1 + \mathrm{erf}\left(\frac{l - l_{t-W+j}}{\sqrt{2}}\right)\right)$$



Utility Based Model

Theorem. *The joint density function of* $D_{(1)}, D_{(2)}, \ldots, D_{(N)}$ *is given by* $f_{1,2,\ldots,N}(d_1, d_2, \ldots, d_N) = N! f(d_1) f(d_2) \ldots f(d_N) \mathbb{I}_{d_1 < d_2 < \ldots < d_N}$.

Theorem. *The probability of success for* $D_{(r)}$ *when the cdf of* D_i *is* $F_D()$ *is given by* $F_{D_{(r)}}(x) = \sum_{j=r}^{N} \binom{N}{j} (F_D(x))^j (1 - F_D(x))^{N-j}$.

Proposition. *The cdf of the* $N-k$ *order statistic upon services demand values is given by* $F_{D_{(N-k)}}(d) = \sum_{j=N-k}^{N} \binom{N}{j} (F_D(d))^j (1 - F_D(d))^{N-j}$.

We can get that the pdf of the $D_{(r)}$

$$f_{D_{(r)}}(x) = \frac{N!}{(r-1)!(N-r)!} (F_D(x))^{r-1} (1 - F_D(x))^{N-r} f_D(x), x \in \mathbb{R}$$

Scenario A. Uniform distribution

$$f_{D_{N-k+1}}(x) = \frac{N!}{(N-k)!(k-1)!} x^{N-k} (1-x)^{k-1}$$

Scenario B. Exponential distribution

$$f_{D_{N-k+1}}(x) = \lambda \frac{N!}{(N-k)!(k-1)!} \left(1 - e^{-\lambda x}\right)^{N-k} e^{-\lambda k x}$$

Order Statistics

Keep locally top-k services (based on demand)

Two scenarios

Utility Based Model

Algorithm Local Decision Making

for $t = 1, 2, \dots$ **do**
$\langle t, T_t, \mathscr{C}_t \rangle = getTask(\mathscr{T});$
Update($\mathbf{d}$);
Calculate($g, \hat{g}$);
getExpectedDemandRankings($\mathbf{d}$);
getExpectedUtilities($\mathbb{E}(G), \mathbb{E}(\hat{G})$);
Calculate($U, \hat{U}$);
Decision = max($U, \hat{U}$);
end for

$$U = \mathbb{E}(G) \cdot e^{-\eta \frac{\hat{\alpha}}{\zeta}}$$

$$\hat{U} = \mathbb{E}(\hat{G}) \cdot e^{-\eta \frac{\hat{\beta}}{\zeta}}$$

Receive
Get tasks, parameters and constraints

Update
Update the demand and load

Estimate
Get the expected ranking and utilities for Decisions 1 & 2

Decide
Get the appropriate decision

Research Axis C
TASKS

Task Offloading

Data and Tasks

- ✓ IoT devices collect data
- ✓ Users and applications request tasks execution

Local Repository

Tasks Manager

Tasks Offloading

Decision Making

- ✓ Select the appropriate peer
- ✓ Manage the uncertainty about the appropriateness of peers

Challenges

Tasks Characteristics
- ✓ Load, constraints, processing
- ✓ Define the contextual vector of tasks

Peers Characteristics
- ✓ Load, data, processing capabilities
- ✓ Define the Peer Contextual Vector (PCV): load, data relevance, speed of processing, communication cost

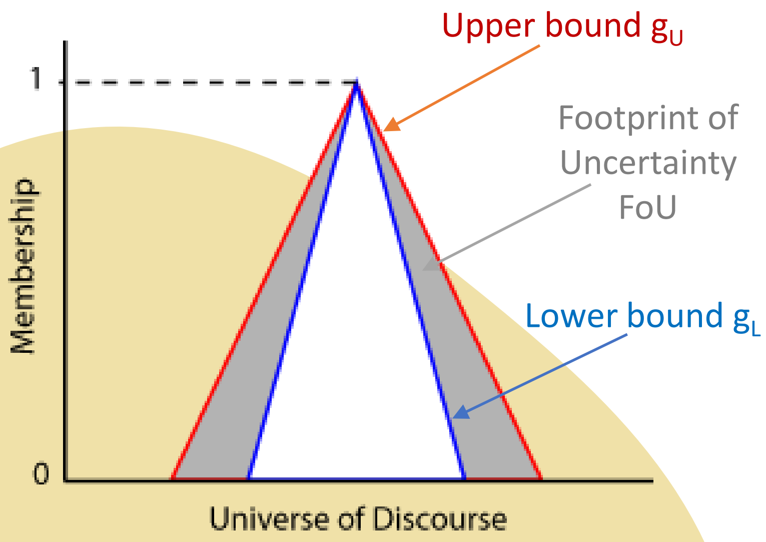
Efficiency of Allocation
- ✓ How can we match tasks contextual vector with PCVs

Uncertainty Management
- ✓ Select the appropriate technology
- ✓ Fuzzy Logic (FL) seems the solution

Contribution

- ✓ We define a function $h(t_j, PCV_k)$ (k is the index of a peer node) that delivers a 'judgement' of the efficiency for the allocation
- ✓ We realize h() with a Type-2 FL System (T2FLS)
- ✓ We handle the uncertainty in two axes: **(i)** in the definition of fuzzy sets; **(ii)** In the definition of membership functions
- ✓ We define the new concept of **Type2D Sets**, i.e., membership functions are automativcally defined by ML

Uncertainty Management



Inputs
- ✓ Load of the peer
- ✓ Speed of processing
- ✓ Estimate of the required processing steps

Output
Potential of Allocation (PoA): depicts the belief that a task will be efficiently in a peer

How to define membership functions?
Answer: from data
> Type-2D sets

Type-2 Fuzzy Sets

Type-2 fuzzy sets and systems generalize standard Type-1 fuzzy sets and systems so that more uncertainty can be handled

Uncertainty Management

- ✓ We adopt a dataset D fed with PCV tuples
- ✓ For each dimension, we generate the appropriate fuzzy sets and their membership functions
- ✓ How? Clustering for each dimension (univariate scenario) upon the latest W tuples
- ✓ The number of clusters is the number of fuzzy sets
- ✓ For values present in a cluster, we pursue the centroid, the variance and the radius to define the two membership functions, i.e., g_L & g_U

The compactness of clusters affects the FoU

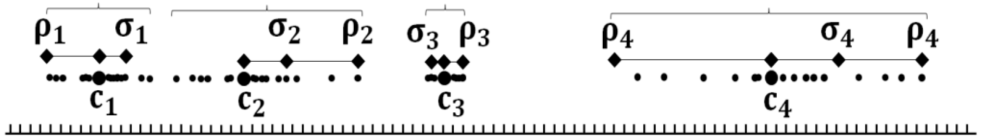
Lower Bound: deviation from the centroid

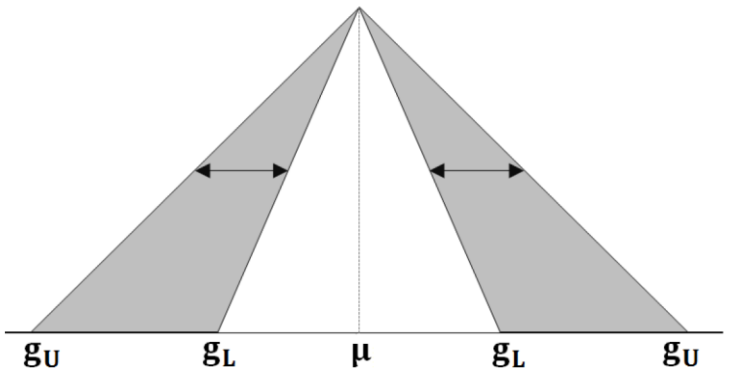
$$g_L^j = \{c_j - \hat{\sigma}_j, c_j + \hat{\sigma}_j\}$$

Upper bound: based on the lower bound

$$g_U = \{c_j - \gamma\hat{\sigma}_j, c_j + \gamma\hat{\sigma}_j\},\ \gamma > 1$$

$$\gamma = \frac{1}{\epsilon_1 + e^{-\epsilon_2 q + \epsilon_3}}$$





|C$_j$|: cardinality of the jth cluster

$$B_i = \begin{cases} 1, & |x_i - c_j| > \theta \\ 0, & \text{Otherwise} \end{cases}$$

Threshold beyond which, values in a cluster are considered as 'outliers'

$$q = \sum_{i=\upsilon}^{|C_j|} P(Y=i) = \sum_{i=\upsilon}^{|C_j|} \binom{|C_j|}{i} P(|x_i - c_j| > \theta)^i (1 - P(|x_i - c_j| > \theta))^{|C_j|-i}$$

Probability of having at least υ points out of |C$_j$| being away from θ

Reward and Decision Making

Calculate the total reward and select the winner!

Smoothing

We consider the room for execution in peers

Reward A

If the PoA is over/below a threshold, a reward/penalty is applied

Reward B

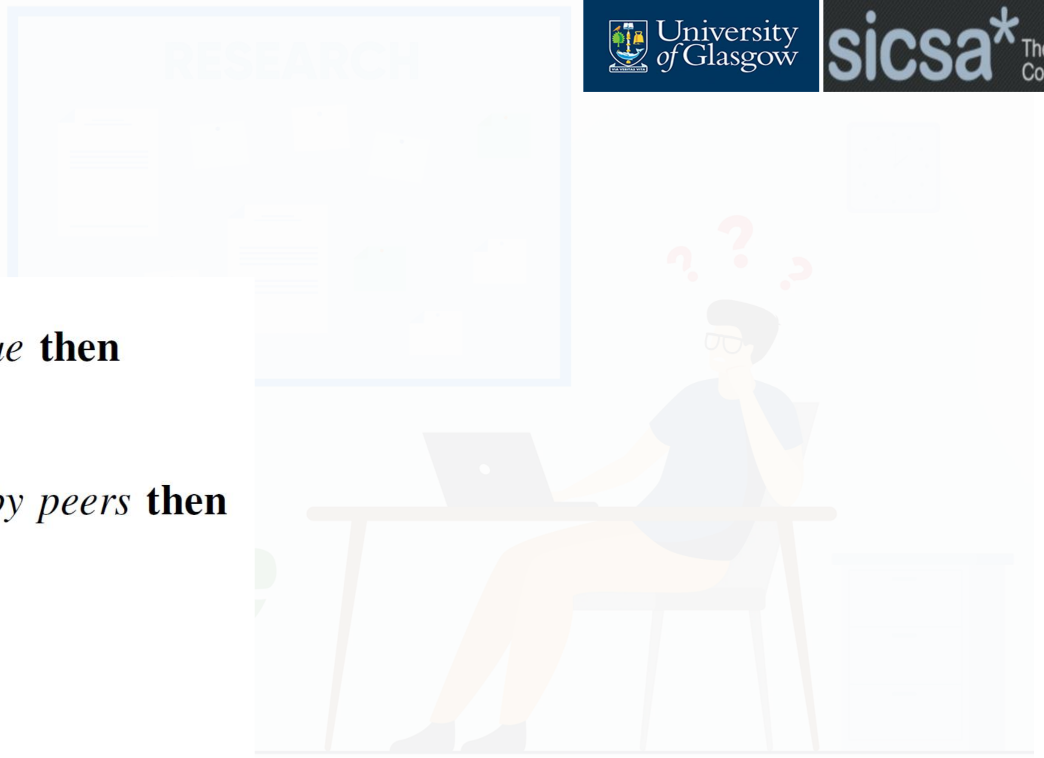
If data in the peer are similar to task requirements a reward/penalty is applied

Reward C

If the communication cost is below a threshold a reward/penalty is applied

$$PoA_k = \begin{cases} \frac{1}{1+e^{-\xi(1-l_k-\lambda_j)}} & \lambda_j < 1 - l_k \\ \frac{1}{2(1+e^{-\xi s_k})} & \lambda_j \geq 1 - l_k \end{cases}$$

Algorithm



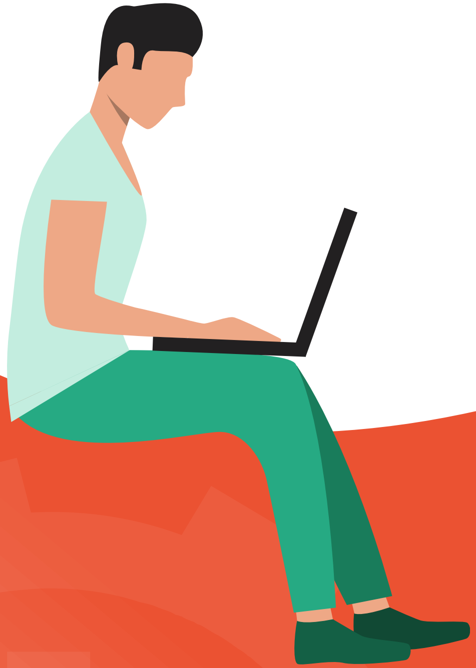
while *true* **do**
    if *training interval expiration is true* **then**
        trainT2DFLS();
    end
    if *contextual vectors are reported by peers* **then**
        updatePCVs();
    end
    if t_j *is received* **then**
        $\mathbf{t}_j$ = defineTaskVector();
        for $k \leftarrow 1$ **to** N **do**
            PoA_k = getT2DFLSResult(l_k, s_k, λ_j);
            Z_k = getReward(PoA_k, r_k, κ_k);
            Z.add(Z_k);
        end
        sort(Z);
        select the best node and allocate t_j;
    end
end

Future Research Directions

- Swarm based collaborative management
- Uncertainty driven self healing models
- Intelligent and collaborative data caching at the EC
- Research on the adoption of auto ML
- Automated distributed decision making

Intelligent EC/EM

Intelligent Algorithms /Models



THANK YOU

More Publications, Datasets, Presentations can be found at:

http://kostasks.users.uth.gr

http://www.iprism.eu

Email: kostasks@uth.gr