Notation:

- N – number of nodes
- A – number of applications
- M – number of modules
- K – number of dependencies
- Z – number of module pairs with dependencies
- Q – number of loop deadlines
- E – number of network links

- P^{Mips} [€], $1 \times N$ matrix representing the price of using processing resources in each node
- P^{Mem} [€], $1 \times N$ matrix representing the price of using memory resources in each node
- P^{Strg} [€], $1 \times N$ matrix representing the price of using storage resources in each node
- P^{Bw} [€], $1 \times N$ matrix representing the price of using network resources in each node
- P^{Pw} [€], $1 \times N$ matrix representing the power price in each node

- f^{Fog}, $N \times 1$ binary matrix representing whether each node is a client or not (f_n^{Fog} is 0 if n is a client)
- f^{Mips} $[MIPS]$, $N \times 1$ matrix representing the processing capacity of each node
- f^{Mem} $[Byte]$, $N \times 1$ matrix representing the memory capacity of each node
- f^{Strg} $[Byte]$, $N \times 1$ matrix representing the storage capacity of each node
- f^{bPw} $[W]$, $N \times 1$ matrix representing the busy power consumption of each node
- f^{iPw} $[W]$, $N \times 1$ matrix representing the idle power consumption of each node
- f^{Tx} $[W]$, $N \times 1$ matrix representing the transmitter power (f_n^{Tx} is non 0 if n is a mobile node)

- m^{Mips} $[MIPS]$, $M \times 1$ matrix representing the processing resources needed for each module
- m^{Mem} $[Byte]$, $M \times 1$ matrix representing the memory resources needed for each module
- m^{Strg} $[Byte]$, $M \times 1$ matrix representing the storage resources needed for each module
- m^{MigD} $[s]$, $M \times 1$ matrix representing the migration deadline for each module

- e^{Cpu} $[MI]$, $K \times 1$ matrix representing the tuple CPU size needed to be processed for each dependency
- e^{Nw} $[Byte]$, $K \times 1$ matrix representing the tuple network size needed to be sent for each dependency
- e^{Pe} $[s]$, $K \times 1$ matrix representing the periodicity of sending the tuple for each dependency
- e^{Prob}, $K \times 1$ matrix representing the probability of sending the tuple for each dependency
- e^{S}, $K \times 1$ matrix representing the source module for each dependency
- e^{D}, $K \times 1$ matrix representing the destination module for each dependency

- l^{S}, $Z \times 1$ matrix representing the source module for each pair of modules with dependencies
- l^{D}, $Z \times 1$ matrix representing the destination module for each pair of modules with dependencies

- m^{Bw} $[Byte/s]$, $M \times M$ matrix representing the bandwidth needed between modules
- m^{CPU} $[MI]$, $M \times M$ matrix representing the CPU size of dependencies between modules
- m^{NW} $[Byte]$, $M \times M$ matrix representing the network size of dependencies between modules

- E^{S}, $E \times 1$ matrix representing the source node for each network link
- E^{D}, $E \times 1$ matrix representing the destination node for each network link
- E^{L} $[s]$, $E \times 1$ matrix representing the link latency between each two nodes
- E^{Bw} $[Byte/s]$, $E \times 1$ matrix representing the link bandwidth between each two nodes

- D, $N \times M$ binary matrix representing the nodes where each module can be deployed
- C, $N \times M$ binary matrix representing the current module placement

- A^L, $Q \times M \times M$ binary matrix representing the loop module list
- A^D $[s]$, $Q \times 1$ matrix representing the loop deadline list
- A^A, $Q \times 1$ matrix representing the application index of each loop
- A^P [€], $A \times 1$ matrix representing the price for not accomplishing the application loops deadline

- α^p, the percentage of processing capacity which is not used for control operations
- α^m, the percentage of memory capacity which is not used for control operations
- α^s, the percentage of storage capacity which is not used for control operations
- α^b, the percentage of bandwidth capacity which is not used for control operations
- t^{Boot} $[s]$, constant representing the average virtual machine boot time

Variables:

- P, $N \times M$ binary matrix representing the module placement
- R, $Z \times E$ binary matrix representing the tuple routing map
- V, $M \times E$ binary matrix representing the module migration routing map

Preliminary computations:

$$m_i^{Mips} = \sum_{k \in K} \frac{e_k^{Prob} e_k^{Cpu}}{e_k^{Pe}}, \; e_k^D = i$$

$$m_{i,j}^{Bw} = \sum_{k \in K} \frac{e_k^{Prob} e_k^{Nw}}{e_k^{Pe}}, \; e_k^S = i, \; e_k^D = j$$

$$m_{i,j}^{CPU} = \sum_{k \in K} e_k^{Cpu}, \; e_k^S = i, \; e_k^D = j$$

$$m_{i,j}^{NW} = \sum_{k \in K} e_k^{Nw}, \; e_k^S = i, \; e_k^D = j$$

Objectives:

Operational Cost:

$$\begin{aligned} C_O = & P^{Mips} \times P \times m^{Mips} + P^{Mem} \times P \times m^{Mem} + P^{Strg} \times P \times m^{Strg} + \\ & \left(P^{Bw} \sum_{z \in Z} m^{Bw}_{l^S_z, l^D_z} \times R_z \right) 1N + C_{Pw}(P^{Pw'}) + C_Q(A^P) \end{aligned}$$

QoS Cost:

$$\begin{aligned} & C_Q(x) = \sum_{a \in A} e_a \; x_a, \\ & \qquad e_a = min \left(\sum_{q \in Q} e_q, \; 1 \right), A^A_q = a, \\ & \qquad e_q = \begin{cases} 1, & if \; L^P_q + L^T_q > A^D_q; \\ 0, & otherwise. \end{cases} \quad , \; \forall q \in [0, Q] \end{aligned}$$

Power Cost:

$$C_{Pw}(x) = C_P(x. \times (f^{bPw} - f^{iPw})) + C_B(x. \times f^{Tx})$$

Processing Cost:

$$C_P(x) = \sum_{n \in N} f^{Fog}_n x_n \frac{P_n \times m^{Mips}}{\alpha^p \times f^{Mips}_n}$$

Bandwidth Cost:

$$\begin{aligned} & C_B(x) = \sum_{z \in Z} m^{Bw}_{l^S_z, l^D_z} \sum_{e \in E} f^{Fog}_i x_i \frac{R_{z,e}}{\alpha^b \times E^{Bw}_e} \\ & \qquad i = E^S_e \end{aligned}$$

Migration Cost:

$$\begin{aligned} & C_M(x) = \sum_{m \in M} s_m \sum_{e \in E} f^{Fog}_i x_i \frac{V_{m,e}}{(1 - \alpha^b) \times E^{Bw}_e} \\ & \qquad i = E^S_e, \; s_m = m^{Strg}_m + m^{Mem}_m \end{aligned}$$

Multiple-objective (Fog service provider):

$$F = [\ C_Q(1N),\ C_{Pw}(1N),\ C_P(1N),\ C_B(1N),\ C_M(1N)\ ]^T$$

Single-objective (Broker):

$$F = C_O$$

Final problem:

$$\begin{aligned}
\underset{P,R,V}{\text{minimize}} \quad & F \\
\text{subject to} \quad & P_{n,m} \in \{0,1\},\ \forall n \in [0,N],\ \forall m \in [0,M] \\
& R_{z,e} \in \{0,1\},\ \forall z \in [0,Z],\ \forall e \in [0,E] \\
& V_{m,e} \in \{0,1\},\ \forall m \in [0,M],\ \forall e \in [0,E] \\
& f_n^{Mips} > 0,\ \forall n \in [0,N] \\
& E_e^{Bw} > 0,\ \forall e \in [0,E] \\
& P \times m^{Mips} \leq \alpha^p \times f^{Mips} \\
& P \times m^{Mem} \leq \alpha^m \times f^{Mem} \\
& P \times m^{Strg} \leq \alpha^s \times \leq f^{Strg} \\
& \sum_{z \in Z} m^{Bw}_{l_z^S, l_z^D} \times R_{z,e} \leq \alpha^b \times E_e^{Bw},\ \forall e \in [0,E] \\
& \sum_{n \in N} P_{n,m} = 1,\ \forall m \in [0,M] \\
& P \leq D \\
& \sum_{i \in E} R_{z,i} - \sum_{j \in E} R_{z,j} = P_{n,l_z^S} - P_{n,l_z^D}\ ,\forall z \in [0,Z],\ \forall n \in [0,N],\ E_i^S = n,\ E_j^D = n \\
& \sum_{i \in E} V_{m,i} - \sum_{j \in E} V_{m,j} = C_{n,m} - P_{n,m}\ ,\forall m \in [0,M],\ \forall n \in [0,N],\ E_i^S = n,\ E_j^D = n \\
& L_m^M \leq m_m^{MigD},\ \forall m \in [0,M]
\end{aligned}$$

$where:$

$$L_q^P = \sum_{i \in M} \sum_{j \in M} A^L_{q,i,j} \sum_{n \in N} P_{n,j}\ \frac{\sum_{l \in M} \sum_{k \in M} m^{CPU}_{l,k} \times P_{n,k}}{\alpha^p \times f_n^{Mips}}$$

$$L_q^T = \sum_{i \in M} \sum_{j \in M} A^L_{q,i,j} \sum_{e \in E} R_{z',e} \left(\frac{\sum_{z \in Z} R_{z,e} \times m^{NW}_{l_z^S, l_z^D}}{\alpha^b \times E_e^{Bw}} + E_e^L \right)$$

$$l_{z'}^S = i,\ l_{z'}^D = j$$

$$L_m^M = b \times t^{Boot} + \sum_{e \in E} V_{m,e} \left(\frac{m_m^{Strg} + m_m^{Mem}}{(1-\alpha^b) \times E_e^{Bw}} + E_e^L \right)$$

$$b = C_{n,m} - P_{n,m},\ C_{n,m} = 1,\ \sum_{n' \in N} C_{n',m} = 1$$