Notation:

- N − number of nodes
- M − number of modules
- K − number of dependencies
- Z − number of module pairs with dependencies
- Q − number of loop deadlines

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

- f^{Mips} $[MIPS]$, $N \times 1$ matrix representing the processing capacity of each fog node
- f^{Mem} $[Byte]$, $N \times 1$ matrix representing the memory capacity of each fog node
- f^{Strg} $[Byte]$, $N \times 1$ matrix representing the storage capacity of each fog node
- f^{bPw} $[W]$, $N \times 1$ matrix representing the busy power consumption of each fog node
- f^{iPw} $[W]$, $N \times 1$ matrix representing the idle power consumption of each fog 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

- 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^{D}, $M \times M$ matrix representing the dependencies between modules
- m^{B}, $M \times M$ matrix representing the bandwidth needed between modules
- m^{CPU}, $M \times M$ matrix representing the CPU size of dependencies between modules
- m^{NW}, $M \times M$ matrix representing the network size of dependencies between modules

- f^{L}, $N \times N$ matrix representing the link latency between each two nodes
- f^{B}, $N \times N$ 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, $Q \times 1$ matrix representing the loop deadline list

- 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 both for control operations and migrations
- t^{Boot}, constant representing the average virtual machine boot time
- ϵ small positive constant

Variables:

- P, $N \times M$ binary matrix representing the placement mapping between modules and nodes
- R, $Z \times N \times N$ binary matrix representing the tuple routing map between modules
- V, $M \times N \times N$ binary matrix representing the module migration routing map

Preliminary computations:

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

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

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

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

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

Objectives:

Operational Cost:

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

Power Cost:

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

Processing Cost:

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

Bandwidth Cost:

$$C_B(x) = \sum_{z \in Z} m^B_{l^S_z, l^D_z} \sum_{i \in N} x_i \sum_{j \in N} \frac{R_{z,i,j}}{\alpha^b \times f^B_{i,j} + \epsilon}$$

Migration Cost:

$$C_M(x) = \sum_{m \in M} s_m \sum_{i \in N} x_i \sum_{j \in N} \frac{V_{m,i,j} \times \sum_{l \in M} m^D_{l,m}}{(1 - \alpha^b) \times f^B_{i,j} + \epsilon},$$

$$s_m = m^{Strg}_m + m^{Mem}_m$$

Multiple-objective:

$$F = \{C_{Pw}(1N),\ C_P(1N),\ C_B(1N),\ C_M(1N)\}$$

Single-objective:

$$F = C_O$$

Final problem:

$$
\begin{aligned}
\underset{P,R,V}{\text{minimize}} \quad & F \\
\text{subject to} \quad & P_{i,j} \in \{0,1\},\ \forall i \in [0,N],\ \forall j \in [0,M] \\
& R_{z,i,j} \in \{0,1\},\ \forall z \in [0,Z],\ \forall i \in [0,N],\ \forall j \in [0,N] \\
& V_{z,i,j} \in \{0,1\},\ \forall z \in [0,M],\ \forall i \in [0,N],\ \forall j \in [0,N] \\
& P \times m^{Mips} \leq \alpha^p \times f^{Mips} \\
& P \times m^{Mem} \leq \alpha^m \times f^{Mem} \\
& P \times m^{Strg} \leq \alpha^s \times \leq f^{Strg} \\
& \sum_{z \in Z} m^B_{l^S_z, l^D_z} \times R_{z,i,j} \leq \alpha^b \times f^B_{i,j} \quad ,\ \forall i \in [0,N],\ \forall j \in [0,N] \\
& \sum_{i \in N} P_{i,j} = 1,\ \forall j \in [0,M] \\
& P \leq D \\
& \sum_{j \in N} R_{z,i,j} - \sum_{j \in N} R'_{z,i,j} = P_{i,l^S_z} - P_{i,l^D_z} \quad ,\forall z \in [0,Z],\ \forall i \in [0,N] \\
& \sum_{j \in N} V_{z,i,j} - \sum_{j \in N} V'_{z,i,j} = C_{j,i} - P_{j,i} \quad ,\ \forall z \in [0,M],\ \forall i \in [0,N] \\
& L^P_q + L^T_q \leq A^D_q,\ \forall q \in [0,Q] \\
& L^M_m \leq 1,\ \forall m \in [0,M]
\end{aligned}
$$

$where:$

$$L^P_q = \sum_{i \in M} \sum_{j \in M} A^L_{q,i,j} \sum_{n \in N} P_{n,j} \ \frac{\sum_{l \in M} \sum_{k \in M} m^{CPU}_{l,k} \times P_{n,k}}{f^{Mips}_n}$$

$$L^T_q = \sum_{i \in M} \sum_{j \in M} A^L_{q,i,j} \sum_{l \in N} \sum_{k \in N} R_{z',l,k} \left(\frac{\sum_{z \in Z} R_{z,l,k} \times m^{NW}_{l^S_z, l^D_z}}{\alpha^b \times f^B_{l,k} + \epsilon} + f^L_{l,k} \right),$$

$$l^S_{z'} = i,\ l^D_{z'} = j$$

$$L^M_m = b \times t^{Boot} + \sum_{i \in N} \sum_{j \in N} V_{m,i,j} \times \left(\frac{m^{Strg}_m + m^{Mem}_m}{(1-\alpha^b) \times f^B_{i,j} + \epsilon} + f^L_{i,j} \right)$$

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

Abandoned ideas

Transmission Cost:

$$C_T = \sum_{i \in N} \sum_{j \in N} f^L_{i,j} \sum_{z \in Z} \left(m^D_{l^S_z, l^D_z} \times R_{z,i,j} \right)$$

Constraint:

$$L^P_q = \sum_{i \in M} \sum_{j \in M} A^L_{q,i,j} \times \frac{m^{CPU}_{i,j}}{m^{Mips}_j},\ m^{Mips}_j \neq 0$$

$$L^T_q = \sum_{i \in M} \sum_{j \in M} A^L_{q,i,j} \sum_{l \in N} \sum_{k \in N} R_{z,l,k} \left(\frac{m^{NW}_{i,j}}{m^B_{i,j}} + f^L_{l,k} \right),\ l^S_z = i,\ l^D_z = j,\ m^B_{i,j} \neq 0$$

$$\begin{aligned} L^M_q &= \sum_{i \in M} A_i \times \left(\sum_{l \in N} \sum_{k \in N} V_{i,l,k} \left(\frac{m^{Strg}_i + m^{Mem}_i}{f^B_{l,k} + \epsilon} + f^L_{l,k} \right) + B_i \times t^{Boot} \right), \\ &A_i = \min\Big\{ \sum_{j \in M} A^L_{q,i,j} + A^L_{q,j,i},\ 1 \Big\}, \\ &B_i = C_{n,i} - P_{n,i},\ C_{n,i} = 1, \sum_{n' \in N} C_{n',i} = 1 \end{aligned}$$