Single-objective Optimization

Notations:

- 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^{Ram} [€], $1 \times N$ 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^{Ram} $[Byte]$, $N \times 1$ 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^{Ram} $[Byte]$, $M \times 1$ 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

- t^{Boot}, constant representing the average virtual machine boot time
- b, constant representing the percentage of link bandwidth allocated for sending tuples
- ϵ small positive number

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:

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

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

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

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

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

Problem formulation:

Operational Cost (C)

$$C = P^{Mips} \times P \times m^{Mips} + P^{Ram} \times P \times m^{Ram} + P^{Strg} \times P \times m^{Strg} + \left(P^{Bw} \sum_{z \in Z} m^B_{l_z^S, l_z^D} \times R_z \right) 1N + C_{Pw} \ ,$$

$$where \ :$$

$$C_{Pw} = \sum_{i \in N} P_i^{Pw} \left[f_i^{iPw} + (f_i^{bPw} - f_i^{iPw}) \times \frac{P_i \times m^{Mips}}{f_i^{Mips}} + f_i^{Tx} \sum_{z \in Z} m^B_{l_z^S, l_z^D} \sum_{j \in N} \frac{R_{z,i,j}}{f_{i,j}^B \times b + \epsilon} \right]$$

Final problem:

$$
\begin{aligned}
\underset{P,R}{\text{minimize}} \quad & C \\
\text{subject to} \quad & P \times m^{Mips} \leq f^{Mips}, \\
& P \times m^{Ram} \leq f^{Ram}, \\
& P \times m^{Strg} \leq f^{Strg}, \\
& P \leq D \\
& P_{i,j} \in \{0,1\},\ \forall i \in [0,N],\ \forall j \in [0,M] \\
& R_{z,i,j} \in \{0,1\},\ \forall z \in [0,Z],\ \forall i \in [0,N],\ \forall j \in [0,N] \\
& V_{z,i,j} \in \{0,1\},\ \forall z \in [0,M],\ \forall i \in [0,N],\ \forall j \in [0,N] \\
& \sum_{i \in N} P_{i,j} = 1,\ \forall j \in [0,M] \\
& \sum_{j \in N} R_{z,i,j} - \sum_{j \in N} R'_{z,i,j} = P_{i,l_z^S} - P_{i,l_z^D} \quad ,\forall z \in [0,Z],\ \forall i \in [0,N] \\
& \sum_{z \in Z} m^B_{l_z^S,l_z^D} \times R_{z,i,j} \leq f^B_{i,j} \times b \quad ,\ \forall i \in [0,N],\ \forall j \in [0,N] \\
& \sum_{j \in N} V_{z,i,j} - \sum_{j \in N} V'_{z,i,j} = C_{j,i} - P_{j,i} \quad ,\ \forall z \in [0,M],\ \forall i \in [0,N] \\
& L_q^P + L_q^T + L_q^M < A_q^D,\ \forall q \in [0,Q]
\end{aligned}
$$

$where:$

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

$$
L_q^T = \sum_{i \in M} \sum_{j \in M} A^L_{q,i,j} \sum_{l \in N} \sum_{k \in N} R_{z',l,k} \left(\frac{\sum_{z \in Z} R_{z,l,k} \times m^{NW}_{l_z^S,l_z^D}}{f^B_{l,k} \times b + \epsilon} + f^L_{l,k} \right),
$$

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

$$
L_q^M = \sum_{i \in M} A_i \left(\sum_{l \in N} \sum_{k \in N} V_{i,l,k} \left(\frac{m_i^{Strg} + m_i^{Ram}}{f^B_{l,k} \times (1-b) + \epsilon} + f^L_{l,k} \right) + t^{Boot} \right),
$$

$$
A_i = \begin{cases} 1, & \sum_{j \in M} A^L_{q,i,j} + A^L_{q,j,i} \geq 1 \\ 0, & Otherwise \end{cases}
$$

$where:$

$$L_q^P = \sum_{i\in M}\sum_{j\in M} A_{q,i,j}^L \times m_{i,j}^{CPU} \times \sum_{n\in N} P_{n,j} \frac{\sum_{l\in M} P_{n,l}}{f_n^{Mips}}$$

$$L_q^L = \sum_{j\in N}\sum_{j\in N} A_{i,j} \times f_{i,j}^L,$$

$$A_{i,j} = \begin{cases} 1, & \sum_{z\in Z} R_{z,i,j} + \sum_{m\in M} V_{m,i,j} \geq 1 \\ 0, & Otherwise \end{cases}$$

$$L_q^T = \sum_{i\in M}\sum_{j\in M} A_{q,i,j}^L \sum_{l\in N}\sum_{k\in N} R_{z',l,k} \left(\frac{\sum_{z\in Z} R_{z,l,k} \times m_{l_z^S,l_z^D}^{NW}}{f_{l,k}^B + \epsilon}\right),$$

$$l_{z'}^S = i,$$

$$l_{z'}^D = j$$

$$L_q^M = \sum_{i\in M} A_i \left(\sum_{l\in N}\sum_{k\in N} V_{i,l,k} \left(\frac{\sum_{j\in M} V_{j,l,k} \times \left(m_j^{Strg} + m_j^{Ram}\right)}{f_{l,k}^B + \epsilon}\right) + t^{Boot}\right),$$

$$A_i = \begin{cases} 1, & \sum_{j\in M} A_{q,i,j}^L + A_{q,j,i}^L \geq 1 \\ 0, & Otherwise \end{cases}$$