Research on real-time scheduling algorithm of federated learning tasks based on energy optimization on NoC platform

Lin Chen[1], *Hong Zhou*[1], *Tao Song*[1], *Ziyi Zhou*

1. Information Center of Guizhou Province Company, China National Tobacco Corporation, Guiyang 550081, China
2. School of Automation, Southeast University, Nanjing 210096, China

Abstract: Federal learning technology can realize global data sharing and reduce the risk of privacy disclosure under the premise of ensuring data security. Aiming at the problem of task assignment scheduling in federated learning process, this paper studies the problem of federated learning task scheduling on NoC multi-core platform under cloud computing architecture. Considering the limited computing resources of physical nodes, this paper describes the problem of optimal assignment and execution of tasks on network nodes as a mixed integer nonlinear programming problem. In order to improve the computational efficiency, the original problem can be equitably converted into a mixed integer linear programming problem. Finally, the scheduling method is verified by real application of task set, and the influence of parameter selection on scheduling scheme is studied.

Key words: Task mapping; Network on chip; Cloud computing; Federated learning

I. Introduction

In recent years, the Internet of Things, cloud computing and artificial intelligence have been rapidly developed, and are widely used in smart cities, industrial Internet of Things, smart homes, car networking and other fields. However, uploading raw data containing sensitive personal information to the central node for training will increase the risk of privacy disclosure. In order to ensure privacy security, federated learning technology has become a popular distributed computing method. Compared with traditional centralized learning architecture, federated learning technology does not need to share and transmit raw data containing sensitive information, but only needs to upload locally trained models to the cloud for aggregation and update. However, the overall performance of federated learning is affected by heterogeneous data, time-delay sensitive tasks, device computing resources and energy capacity limitations. How to allocate device resources, optimize scheduling, and improve model prediction accuracy have become a major challenge affecting the application and development of federated learning technology.

Task optimization mapping mainly studies how to assign tasks on the processor and determine the execution time and order. The optimization goal is often to reduce the execution time or energy consumption of tasks. In the literature, task scheduling on heterogeneous multi-core platforms based on Dynamic voltage and frequency scaling (DVFS) is studied with the goal of minimizing total energy consumption. This paper proposes an energy optimization method for multi-core chips based on task scheduling, which combines DVFS and DPM (Dynamic Power Management) through the modeling of processor idle time. In addition, the offloading of federated learning tasks to the cloud needs to be considered. Cloud computing offers advantages such as high flexibility, high scalability and on-demand deployment. The literature proposes a cloud-end workload awareness algorithm based on task allocation, which maintains task QoS through calculation and communication frequency optimization. The literature studies the joint task scheduling problem of local processor, wireless channel and cloud offloading of mobile devices, and adjusts the working mode of nodes through task migration and DVFS to reduce the overall energy consumption.

The traditional bus structure has the problems of delay and poor performance. In order to improve communication efficiency, Network-on-Chip (NoC) came into being. With the characteristics of low latency, high bandwidth, low power consumption and high integration, NoC has been widely used in large-scale integrated circuit design. In this paper, we study the optimal mapping of federated learning tasks on NoC platform and cloud.

II. System modeling

Network nodes independently train local models based on the local data they collect, and upload parameters via wireless links to the cloud, which aggregates the collected parameters and conducts global modeling. In this paper, we consider that the federated learning task set contains correlation task $\{t_1...t_M\}$. Each task can be expressed as a set $\{C_i, D_i, p_{ij}, s_{ij}, H\}$. The tasks are released at time 0 and share a common scheduling horizon $H.C_i$. is the Worst Case Execution Cycle (WCEC) of t_i, and D_i is the relative deadline of t_i. The binary matrix $p=[p_{ij}]M\times M$ represents the dependency between tasks. If t_i is the predecessor of t_i, p_{ij}, otherwise, $p_{ij}=0$.

We consider a multi-core platform based on NoC in this paper. The structure of NoC includes N processors $\{\theta_1,...\theta_N\}$ and N routers $\{R_1,...R_N\}$. The processor supports DVFS and has L various Voltage/Frequency (V/F) levels $\{(v_l, f_l),...,(v_L, f_L)\}$. When the working voltage/frequency is (v_l, f_l), the calculated power is $P_l=P^s_l+P^d_i$, where $P^s_l=L_g(v_l K_1 e^{K_2 v_l} e^{K_3 v_b}+|v_b|I_b)$ is the static power, and $P^d_i=C_e v^2_l f_i$ is the dynamic power, L_g、K_1、K_2、K_3、v_b、I_b、C_e are the related parameters of the processor model. When the working voltage/frequency is (v_l, f_l) , the computing energy consumption is $e_i^{comp}=P_l t_i^{comp}$ and the computing time is $t_i^{copm}=C_i/f_i$. According to XY routing algorithm, the communication energy matrix $e=\{e_{\beta\gamma k}\}_{N\times N\times N}$ and communication time matrix $t=\{t_{\beta\gamma}\}_{N\times N}$ can be obtained, where $e_{\beta\gamma k}$ represents the energy consumed by θ_k when θ_β transmit unit of data to θ_γ, and $t_{\beta\gamma}$ represents the time required for θ_β to transmit unit of data to θ_γ.

For related tasks t_i、t_j and t_k, that is $p_{ij}=p_{jk}=1$, if t_i and t_k are executed locally but t_j offloaded to the cloud, the sending time from t_i to t_j is $t^{se}_j=S_{ij}/R^s$, and the sending energy is $e^c_j=P^s t^{se}_j$, where R^s is the sending rate and R^s is the sending power. The time for t_k to receive data from t_j is $t^r_k=s_{jk}/R^r$, where is the receiving rate. The NoC platform energy consumption is negligible when receiving data, since the computing time and energy consumption of tasks executed in the cloud is much smaller than that of tasks executed on local processors, the computing time and energy consumption of tasks executed in the cloud can be represented by constants t^c and e^c.

III. Description of task optimization mapping problem

Since each task can be executed on the NoC platform locally or offloaded to the cloud, and tasks can only be assigned to one processor when executed locally, that is:

$(\lambda_i \sum_{l\in\mathcal{N}} x_{il}) + (1-\lambda_i) = 1, \forall i \in M \quad \mathcal{M}$ (1)

where represents is executed on the NoC platform, otherwise; represents that is assigned to , .

When the task is executed locally on one processor, that processor can only select one V/F level, that is:

$(\lambda_i \sum_{l\in\mathcal{L}} y_{il}) + (1-\lambda_i) = 1, \forall i \in \mathcal{M}$ (2)

where y_{il} denotes t_i select (v_i, f_i), otherwise $y_{il}=0$.

t_i will not start executing until it receives all the data from its predecessor tasks, and once it finishes executing, the data can be immediately sent to its successor tasks s. Considering the task offloading and assignment, the time when t_i receives the data from its predecessor task t_j, that is, the communication time is $t_i^{comm}=\sum_j\sum_\beta\sum_\gamma p_{ji}[\lambda_i\lambda_j x_{i\beta} x_{j\gamma} t_{\beta\gamma}+(1-\lambda_i)t^{se}_t+(1-\lambda_j)t^r_i]$. Therefore, the computing time of t_i is $t_i^{comm}=\lambda_i\sum_l y_{il}\dfrac{c_i}{f_i}+(1-\lambda_i)t^c$. For correlation tasks and , the order of execution is:

$t^s_j+(1-p_{ij})H\geq t^s_i+p_{ij}t_i^{comp}+t_j^{comm}, \ \forall\ i\neq j\ \in \mathcal{M}$ (3)

where t^s_i and $t^e_i+t_i^{comp}$ are the start and end time of t_i.

When independent tasks t_i and t_j are assigned to the same processor, since only one task can be executed at a time, that is:

$t^e_i\leq t^s_j+(2-\lambda_i x_{ik}-\lambda_i x_{jk})H+(1-u_{ij})H, \ \forall\ i\neq j\ \in \mathcal{M}, \ \forall\ k\ \in \mathcal{N}$ (4)

where $u_{ij}=1$ represents the execution of t_i before t_j, otherwise $u_{ij}=0$.

The real-time task execution time should be less than the relative deadline D_i, and all tasks must be executed within the scheduled period , that is:

$t_i^{comp}\leq D_i,\ t^e_i\leq H, \ \forall\ i\ \in \mathcal{M}$ (5)

If t_i and t_j are allocated to θ_β and θ_γ respectively, the communication energy consumption is $e^{locomm}_{ij\beta\gamma k} = s_{ij}x_{i\beta}p_{ij}x_{j\gamma}e_{\beta\gamma k}$; If t_j is executed in the cloud, the task communication energy consumption is $e^{clcomm}_{ij\beta\gamma k} = s_{ij}t^{se}_j P^s$. In consideration of task offloading, the communication energy consumption of t_i and t_j is $e^{comm}_{ij\beta\gamma k} = \lambda_i\lambda_j e^{locomm}_{ij\beta\gamma k} = s_{ij}[\lambda_i\lambda_j x_{i\beta}p_{ij}x_{j\gamma}e_{\beta\gamma k} + (1-\lambda_j)t^{se}_j P^s]$. Therefore, the communication energy consumption of θ_k is $e^{locomp}_i = \sum_i \sum_j \sum_\beta \sum_\gamma e^{comm}_{ij\beta\gamma k}$. When t_i is executed locally, the computing energy consumption is $e^{clcomp}_i = \sum_l y_{il}\dfrac{c_i}{f_l}P_l$. When t_i is offloaded to the cloud, the computing energy consumption is $e_i^{clcomp}=e^c$. Considering task offloading, the calculated energy consumption of t_i is. $e_i^{comp}=\lambda_i e_i^{cocomp}+(1-\lambda_i)e_i^{clcomp}=\lambda_i\sum_l y_{il}\frac{c_i}{f_l}P_l+(1-\lambda_i)e^c$. Therefore, the computing energy consumption of θ_k is $E_k^{comp}=\sum_i x_{ik}e_i^{comp}$.

The objective of this problem is to balance the energy consumption of processors, so the task scheduling problem can be modeled as:

P1= $\min_{x,y,u,ls,\lambda}(\max_{\forall k}\{E_k^{comm}+E_k^{comp}\})$ (7)

s.t.(1)-(5)

P1 is a MINLP problem because of the nonlinear term of multiplying variables in the constraint and objective function. For ease of solution, based on Lemma 1, P1 can be converted to a MILP problem.

Lemma 1: Assume that *x*, *y*, *z* are binary variables and dz=*xy*, the nonlinear term can be replaced by constraints *z*-*x*≤0, *z*-*y*≤0, *x*+*y*-*z*≤1.

IV. Simulation results

The actual task set and NoC parameters in this paper are derived from literature, and MATLAB is used for simulation. Figure 1 compares the system energy consumption when the objective function is to balance energy consumption (BE) and minimize energy consumption (ME). Compared with ME, when the objective function is BE, the total energy consumption on the processor may be larger, but the maximum energy consumption of all processors is smaller, that is, the processor energy consumption is more uniform, and the data processing and data transmission tasks are evenly distributed as far as possible, which avoids routing holes caused by excessive node energy consumption and prolongs the network life.

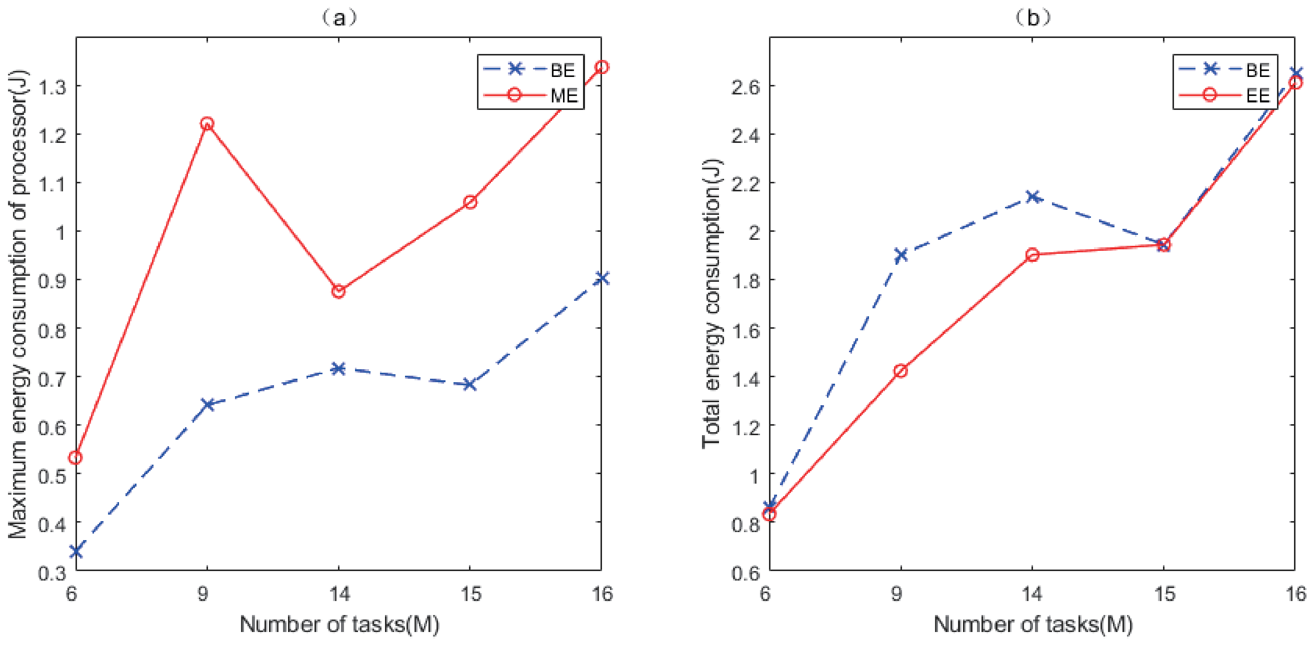

Figure 1: BE compared with EE

Figure 2 shows the effect of scheduling horizon on the value of the objective function, where $\phi=H/H_0$, H_0 is a given constant. The problem is infeasible when ϕ is small. With the increase of ϕ, the value of the objective function gradually decreases and the running time of the algorithm shows a downward trend. When *H* is small, the real-time constraint is more difficult to meet, and a higher V/F level needs to be selected, which leads to the increase of communication and calculation energy consumption. With the increase of *H*, the real-time constraints of the task are more easily satisfied, the running time of the algorithm becomes smaller, and the objective function value becomes smaller.

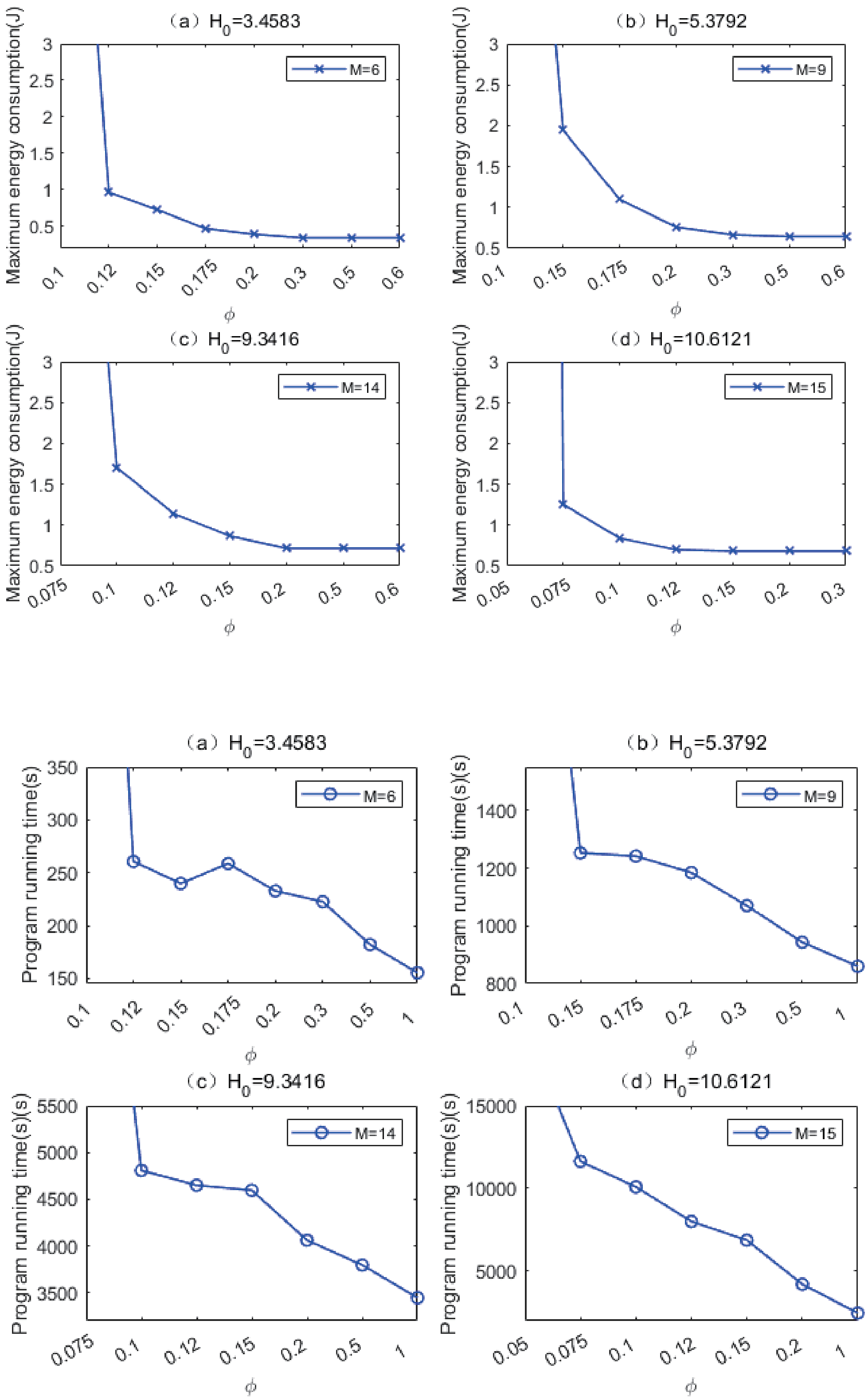

Figure 2: effects on objective function and running time

Figure 3 analyzes the influence of communication energy consumption on task allocation on the processor, where $\mu = \mu' \ \ e_{avr}^{comm} / e_{avr}^{comp}$, and e_{avr}^{comm}, e_{avr}^{comp}, are the average communication and computing energy consumption of the platform respectively; ω=($e_{avr}^{comm} + e_{avr}^{comp}$)/(e_{clo}^{comp} + e_{clo}^{comm}) represents the energy consumption ratio, where $e_{clo}^{comm} = P^s t^{se}$ and $e_{clo}^{comp} = P^s t^{se}$ are respectively the cloud communication and computing energy consumption. When μ is small, tasks are evenly distributed across the processors to reduce the power consumption of a single processor. As the increases of μ, tasks tend to be distributed centrally, as correlated tasks run on different processors increase the number of times they communicate. As we increase, the number of tasks sent to the cloud for execution as a percentage of the total number of tasks increases. Tasks tend to be sent to the cloud because they consume less total energy when executed in the cloud compared to on-premises execution.

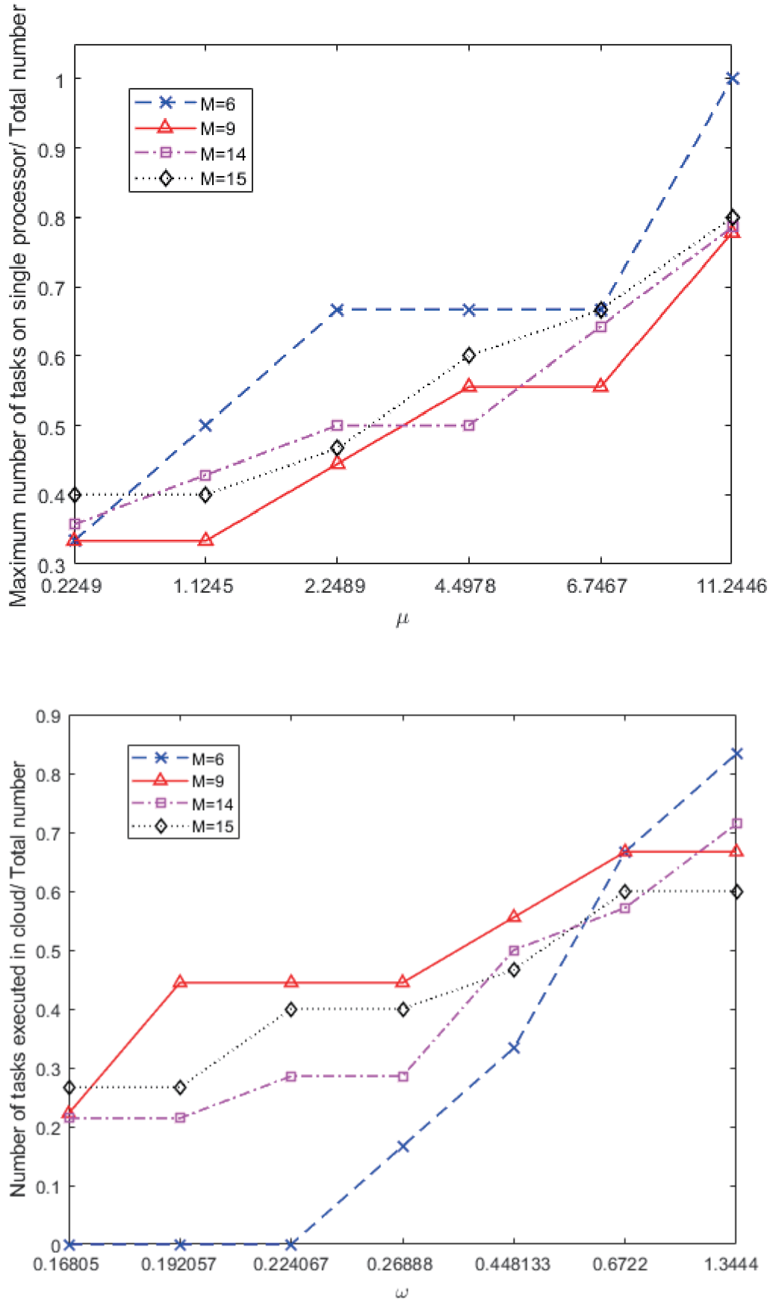

Figure 3: The effect of communication parameters on task distribution

V. Summary

This paper studies the optimized deployment scheme of federated learning task on NoC platform based on cloud computing, and achieves energy balance under real-time constraints. In order to reduce the computational complexity, the original MINLP is relaxed into MILP by linearization without affecting the result optimality. Finally, the performance of the scheduling algorithm is verified by using the real application task set, and the influence of related parameters on the scheduling scheme is analyzed.

References:

[1] D. Li and J. Wu. Minimizing energy consumption for frame-based tasks on heterogeneous multiprocessor platforms [J]. IEEE Transactions on Parallel Distributed Systems, 2015, 26(3): 810-823.

[2] G. Chen, K. Huang and A. Knoll. Energy optimization for real-time multiprocessor system-on-chip with optimal DVFS and DPM combination [J]. ACM Transactions on Embedded Computing Systems, 2014, 13(3): 111-130.

[3] P. Arroba, J. M. Moya, J. L. Ayala and R. Buyya. DVFS-aware consolidation for energy-efficient clouds [C] International Conference on Parallel Architecture and Compilation, 2015: 494-495.

[4] X. Lin, Y. Wang, Q. Xie and M. Pedram. Task scheduling with dynamic voltage and frequency scaling for energy minimization in the mobile cloud computing environment [J]. IEEE Transactions on Services Computing, 2015, 8(2): 175-186.

[5] S. Abd Ishak, H. Wu and U. U. Tariq, Energy-aware task scheduling on heterogeneous NoC-based MPSoCs [C] International Conference on Computer Design, 2017: 165-168.

[6] J. Singh, B. Mangipudi, S. Betha and N. Auluck, Restricted duplication-based MILP formulation for scheduling task graphs on unrelated parallel machines [C] International Symposium on Parallel Architectures, Algorithms and Programming, 2012: 202-209.