CHAPTER IV

DISCUSSION

In this section, the time complexity for the EPC and the ESL algorithms will be explained in more detail. The time complexity, T_{alg}, of the EPC algorithm is $O\left(\left(v^3 \times p\right)/l\right)$, where v is the number of tasks in a graph, p is the number of processing units, and l is the number of graph level. If it is assumed that the number of tasks in all level is uniform, the average number of tasks in each level is v/l. In identifying preliminary candidate processing units step, the amount of time require to identify preliminary candidate processing units of one task depends on the number of its ancestor tasks in preceding level and the candidate processing unit itself, that is, the number of elements considered is $(v/l)\times p$. Thus, the number of elements to be considered to identify preliminary candidate processing units of all task on a given dependent task graph is equal to $\left(v^2/l\right)\times p$. Therefore, the overall time complexity for identifying preliminary candidate processing unit step is equal to $O\left(\left(v^2/l\right)\times p\right)$.

In identifying actual candidate processing units step, the amount of time require to clustering one task in to group depends on the number of ancestors, descendants, and candidate processing units, that is, the number of elements considered is $(v/l)\times p$. Thus, the number of elements to be considered to clustering all tasks into groups is equal to $\left(v^2 \times p\right)/l$. By the same token, identifying actual candidate processing units of each task depends on the number of candidate processing units. The number of elements considered for each task is equal to $v \times p$. Therefore, the overall time complexity for identifying actual candidate processing unit step is equal to $O((v^2 \times p)/l)$.

In task scheduling of processing units step, the amount of time required to scheduling a task on primary candidate processing unit depends on the number of its ancestor tasks in the preceding level, that is, the number of elements considered is

v/l. Thus, the number of elements to be considered to scheduling all tasks on primary candidate processing unit is equal to v^2/l. Therefore, the overall time complexity for task scheduling in processing unit step is equal to $O(v^2/l)$.

In minimizing idle energy of task scheduling step, the amount of time required to minimize idle energy of the task list on each processing unit depends on the number of empty slots on each processing unit. If it is assumed that the average number of assigned tasks on all processing units is uniform, the average number of empty slots in each processing unit is not over v/p. The number of ancestor tasks to be inserted is v/l, and the number of tasks which are affected by the insertion of one of these tasks into the idle slot cannot be greater than the number of all tasks in the task graph. Therefore, the time to reschedule of inserted tasks is equal to v^2/l. The number of elements to minimize idle energy of the task list on one processing unit is equal to $v^4/(l \times p)$. The overall time complexity for minimizing idle energy in task scheduling step is equal to $O(v^4/l)$. The combined time complexity in all steps becomes

$$\begin{aligned} T_{alg} &= O\left(\left(v^2 \times p\right)/l\right) + O\left(\left(v^2 \times p\right)/l\right) + O\left(v^2/l\right) + O\left(v^4/l\right) \\ &= O\left(v^4/l\right). \end{aligned}$$

The time complexity, T_{alg}, of the ESL algorithm is $O((v^2 \times p)/l)$, where v is the number of tasks in a graph, p is the number of processing units, and l is the number of graph levels. In the level-based task scheduling step, the algorithm considers a task graph one level at a time. If it is assumed that the number of tasks in all level is uniform, the average number of tasks in each level is v/l. The amount of time required to compute an EST of one task depends on the number of its ancestor tasks in the preceding level and the number of candidate processing unit itself, that is, the number of elements considered is $(v/l) \times p$. Thus, the number of elements to be considered to compute the EST for all tasks on a given level is equal to $(v/l)^2 \times p$. The procedure for building the EFT candidate task list is shown from line 9 to 18 of the ESL Algorithm 1. The EFT of all tasks on the same level for all candidate processing units must be updated every time one task is assigned and taken out of consideration. This means that the number of elements to be considered in the update process is equal to $(v/l) \times p$. For the battery check function, the algorithm requires some time to calculate the number of candidate processing units and to insert the considered tasks into idle slots of each processing unit. Since the number of tasks to be inserted cannot be greater than the number of tasks on a given level, the time complexity is $(v/l)\times p$. The overall time complexity of the level-based task scheduling step is equal to $l\left(\frac{v^2 p}{l^2}+\frac{v}{l}\left(\frac{vp}{l}+\frac{vp}{l}\right)\right)=O\left(\left(v^2\times p\right)/l\right)$.

In idle slot reduction step, the number of tasks to be considered, which are the ancestor tasks of the task next to the idle slot marked in the previous step as the latest task to send the result to the task next to the idle slot, cannot be greater than the number of levels. The number of tasks which are affected by the insertion of one of these tasks into the idle slot cannot be greater than the number of all tasks in the task graph. The complexity for rescheduling of inserted tasks is equal to v^2/l. Thus, the overall time complexity for the idle slot reduction step is equal to $O\left(v^2\right)$. The combined time complexity in all steps becomes

$$T_{alg}=\max\left(O\left(\left(v^2\times p\right)/l\right)+O\left(v^2\right)\right)$$

From the above result, it can be seen that by identifying the actual candidate processing unit before assignment and reducing the idle energy, the EPC algorithm can achieve the lowest total energy consumption. However, the algorithm does not take into account system finish time, limitation of energy supply, or cost of algorithm overhead. These issues were addressed in the ESL algorithm.

Since level based scheduling method incurs lower algorithm overhead and yields the lowest total energy consumption, the use of EFT candidate task lists and idle slot reduction scheme helps provide minimum finish time in four experimental cases and second in the remaining two cases. Despite the advantages of ESL algorithm have been shown through the experimental results, its performance depends primarily on the estimation of various parametric information pertaining to each processing unit along with the structure and costs of the underlying dependent task graph.