Scheduling independent tasks of a meta tasks with significant task dispatch times

Sanjeev Baskiyar and Prashanth SaiRanga
Dept. of CSSE, Auburn University
Auburn, AL 36849
{baskiyar@auburn.edu, sairang@auburn.edu}

In this short report we introduce a novel heuristic called Earliest Finish Times with Deadlines (EFT-DT) to schedule independent tasks of a meta-task onto a network of heterogeneous processors to minimize the makespan of the meta-task.

In grids a centralized scheduler may make all scheduling decisions with respect to independent tasks. The mapping of tasks onto processors requires time to dispatch the task from the scheduler onto a processor. Previous works related to scheduling independent tasks of a meta-task onto a network of heterogeneous processors, the dispatch times of the tasks have not been considered in making scheduling decisions. The Sufferage, Min-Min and the Min-Max algorithms assume a zero dispatch time in their scheduling model. In this paper, we introduce a novel heuristic to schedule independent tasks of a meta-task onto a network of heterogeneous processors considering the dispatch times of tasks.

In the EFT-DT algorithm, the priority of a task is defined as the sum of its mean execution time over all the processors and the standard deviation of its execution time over all the processors. At each scheduling step, EFT-DT picks the task with the highest priority and schedules it onto a processor that provides its earliest completion time. The completion time of task on a processor is defined as $CT(t_k, p_j) \leftarrow Max\{T\_available(p_j), D_{kj}\} + W_{k,j}$ to account for the dispatch times.

Table 1: Definition of Terms

Term	**Definition**
T	Meta-task set of size s
M	Set of processors available for scheduling
m	Number of processors
$mean_k$	Mean execution time of task t_k over all the processors
std_k	Standard deviation of the execution times of task t_k over all the processors
$T\_available(p_j)$	Time at which processor p_j can start execution of a new task.
$W_{k,j}$	Running time of task t_k on processor $p_{j.}$
D_{kj}	Time required to dispatch task t_k from the scheduler to processor p_j
$CT(t_k, p_j)$	$= T\_available(p_j) + D_{kj} + W_{k,j}$ // Execution completion time of task t_k on processor $p_{j.}$
$ECT(t_k)$	$= \text{Min}_{k \in T\ \&\ j \in M} \{ CT(t_k, p_j) \}$//Earliest Completion time of task t_k
$Proc(t_k)$	The processor on which $ECT(t_k)$ is obtained

EFT-DT Algorithm
For all $t_k \in T$
$priority(t_k) \leftarrow mean_k + std_k$
End For
While $T \neq \Phi$ **do**
Pick a task $t_k \in T$ with the highest priority
  For all $j \in M$
    $CT(t_k, p_j) \leftarrow Max\{T\_available(p_j), D_{kj}\} + W_{k,j}$
  End For
  $ECT(t_k) \leftarrow \text{Min}_{k \in T \,\&\, j \in M}\{ CT(t_k, p_j) \}$
  Compute $Proc(t_k)$
  Assign t_k to $Proc(t_k)$
  $T\_available(Proc(t_k)) \leftarrow ECT(t_k)$
  $T = T - \{t_k\}$
End While
End EFT-DT

Figure 1: The EFT-DT Algorithm

EFT-DT takes $O(s)$ to compute the priorities of all the tasks and $O(s \times m)$ to calculate the earliest completion times of the tasks. Thus the overall complexity is $O(s \times m)$.