Schedulablity analysis of conditional parralel task graphs in multicore systems: Mathematical basics and Algorithm description

Patrick NONKI

April 2022

In this research, the main understanding of the topic is based on the schedulability and performance analysis of a system where we have conditional tasks running together in parralel and in different cores of the processor unit. The algorithm [1] below shows the process in general

```
void num_threads(N){
    void master{
        void task{ //t0
            if (condition){
                task {//t1}
            else{
                task{t2}
                task{t3}
                task{t4}
    }}}}
```

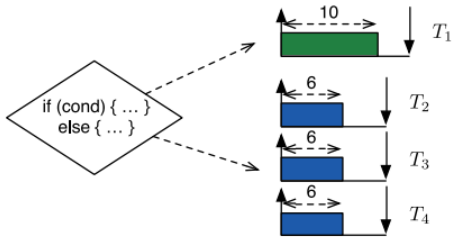

Figure 1: Block diagram

Based on [1], let's assume we have a different set of n conditionnal parralel tasks (cp tasks) running in m identical processors. Let's also have a graph where we map nodes as sub-tasks, arcs to join nodes, and each node having the so called *worst case time execution (WCET).*

A node that has no incoming arc is called a source whereas a node with no ongoing arc is called a sink.
In our representation we can have 2 types of nodes:

- Conditional nodes : represented either by diamond (beginning of the condition) or by circle (end of the condition)
- Regular nodes : represented by rectangles, as normal subtask

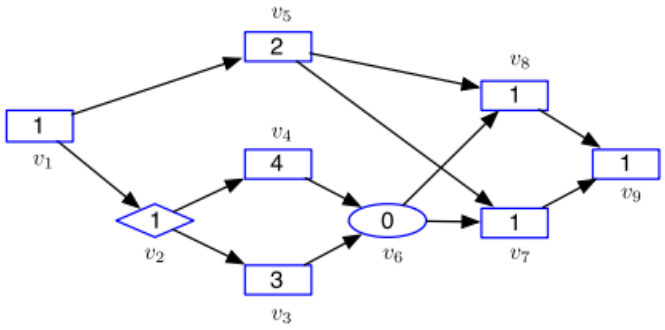

Figure 2: Conditional parralel tasks graph

On the figure we can identify every element listed above.
In the graph we also define the length of the cp task as the length of any longest path in the graph. In the absence of conditional subtasks, it's calculated by having a sum of the WCETs of all the subtasks. In case we have conditional subtasks, the length is now called *Worst case workload* which is calculated by adding WCETs and maximun WCET amongst conditional branches. [1]
In our graph, the length is L=8 and since we have V4¿V3 in our conditional branch, the worst case workload W=11. [1]

References

[1] Alessandra Melani et al. "Schedulability Analysis of Conditional Parallel Task Graphs in Multicore Systems". In: *IEEE Transactions on Computers* 66.2 (Feb. 2017). Conference Name: IEEE Transactions on Computers, pp. 339–353. ISSN: 1557-9956. DOI: 10.1109/TC.2016.2584064.