List-based scheduling paper

Patrick NONKI

May 2022

1 ML-RCS and MR-LCS Algorithms

The goal of the list based scheduling is to attempt to solve the load balancing problem. One idea would be to try a greedy algorithm that consists of:

- Consider n jobs in arbitrary order
- Start with m empty machines
- Place the first job that on the machine that has the smallest amount of load (if 2 machines have the same amount of load place the job on either of them)
- Place the second job on the machine that has now the smallest amount of load ... and so on

But with this greedy algorithm we can see that all the jobs are not equally dispatched over the machine, then it becomes essential to find algorithms that give the same load for all the machines or approximately.

1.1 ML-RCS Algorithm (Minimum latency subject to resource bound)

In order to understand this method the following algorithm is applied :

```
LIST_L( G(V, E), a) {
    l = 1;
    repeat {
        for each resource type k = 1, 2,    , nres {
            Determine ready operations Ul,k;
            Determine unfinished operations Tl,k;
            Select Sk     Ul,k vertices, s.t. |Sk| + |Tl,k|     ak;
            Schedule the Sk operations at step l;
            }
        l = l + 1;
    }
    until (vn is scheduled) ;
```

```
    return (t);
}
```

The example below shows the implementation of the ML-RCS Algorithm:

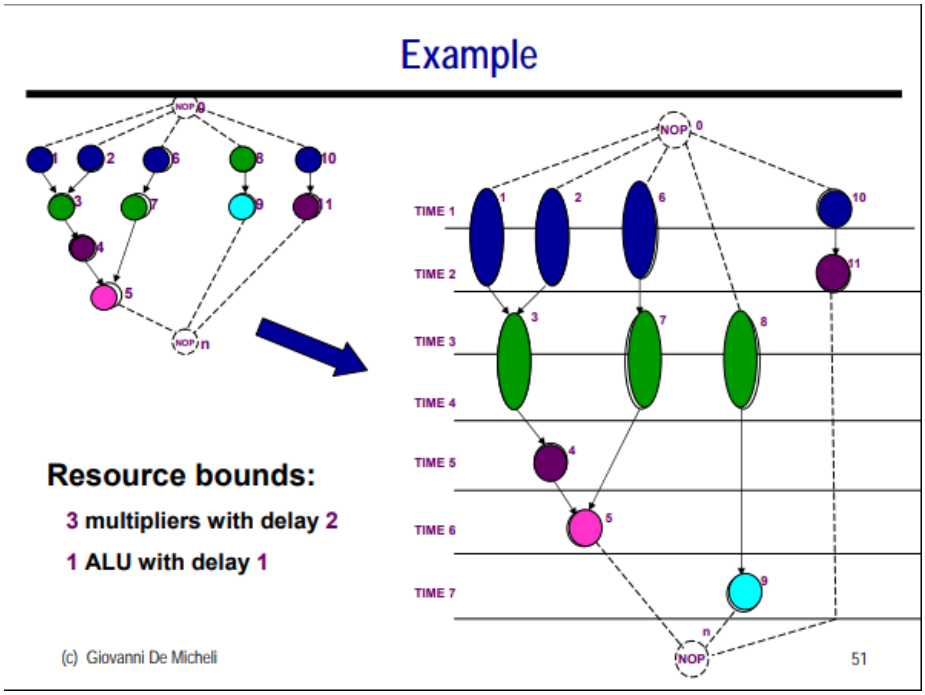

Figure 1: Example of ML-RCS [1]

1.2 MR-LCS Algorithm (Minimum resource subject to latency bound)

In order to understand this method the following algorithm is applied :

```
LIST_R( G(V, E),    ) {
    a = 1;
    Compute the latest possible start times tL by ALAP ( G(V, E),    );
    if (t0 < 0)
        return (   );
    l = 1;
    repeat {
        for each resource type k = 1, 2,     , nres {
            Determine ready operations Ul,k;
            Compute the slacks { si = ti     l for all vi    Ulk};
            Schedule the candidate operations with zero slack and update a;
```

```
                Schedule the candidate operations not needing additionaL resources;
                }
            l = l + 1;
        }
        until (vn is scheduled) ;
        return (t, a);
}
```

The example below shows the implementation of the MR-LCS Algorithm:

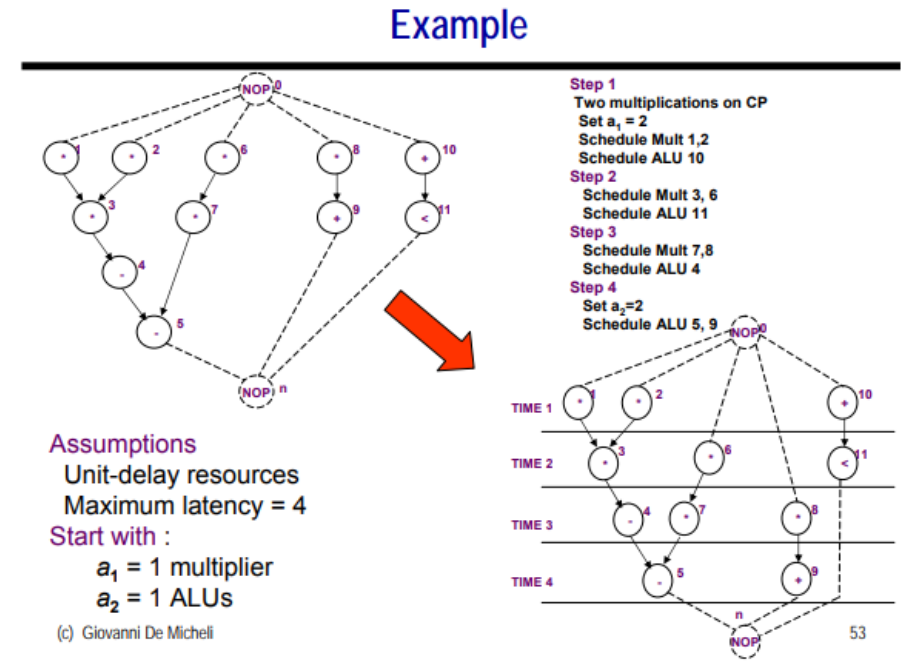

Figure 2: Example of MR-LCS [1]

2 Other similar algorithms

There is a lot of other list based scheduling algorithms that have been developed by the researchers over the years.Some of these methods have a common procedure; they all use the same phases [2]:

- The prioritizing phase: where each task is assigned a priority
- The selection phase: where each task is selected regarding its priority and the ready state

- The processor selection: where a processor is selected for running the chosen task.

2.1 CPOP Algorithm

The CPOP (Critical path on processor) algorithm is based on both upward and downward ranking techniques. It has 2 steps: The task prioritization and the task allocation phases.
On the task prioritization phase, they are prioritized based on their rank values(rank upgrading way + rank downgrading way) that are calculated based on the DAG (Direct acyclic node graph where all the tasks are represented by precedence) [2], and set in a table in a decreasing order.
On the allocation phase, they are selected based on the highest rank value and then assigned to the processor having the less processor cost.

2.2 HCPT Algorithm

This algorithm is divided in three steps:

- The level sorting phase: Where the DAG is traversed from up to down to sort and group tasks on each level that are independent.
- The task prioritization phase: Where tasks are prioritized based on the ranking values as shown in the CPOP.
- The processor selection phase: Where the processor is selected based on the EFT (Earliest finish time) value of each task on a given processor [3].

2.3 HPS Algorithm

The HPS (High Performance task Scheduling) algorithm uses the same steps as the one above. As a major change, we can note that:

- In the task prioritization phase, priorities are assigned based on the Down link cost (DLC) that is the maximum communication cost from a task to his predecessors, the up link cost (ULC) that is the maximum communication cost from a task to his successors, and the link cost (LC) that is the sum of DLC and ULC. The highest priority is then given to the task with the highest LC [2].

References

[1] Janan Paremajalu. G. De Micheli Synthesis And Optimization Of Digital Circuits (Text Recognized Using OCR) [v. 1.03 20 4 2005].

[2] Hamid Arabnejad. List Based Task Scheduling Algorithms on Heterogeneous Systems - An overview. page 10.

[3] C Subramanian, N Rajkumar, and S Karthikeyan. LIST BASED SCHEDULING ALGORITHM FOR HETEROGENEOUS SYSYTEM. *2015*, 10:6, 2015.