ADAPTIVE STEADY-STATE SCHEDULING FOR GRID PLATFORMS

By: Jaime Ballesteros, PhD Student

Advisor:

Prof. Jaime Seguel

WALSAIP

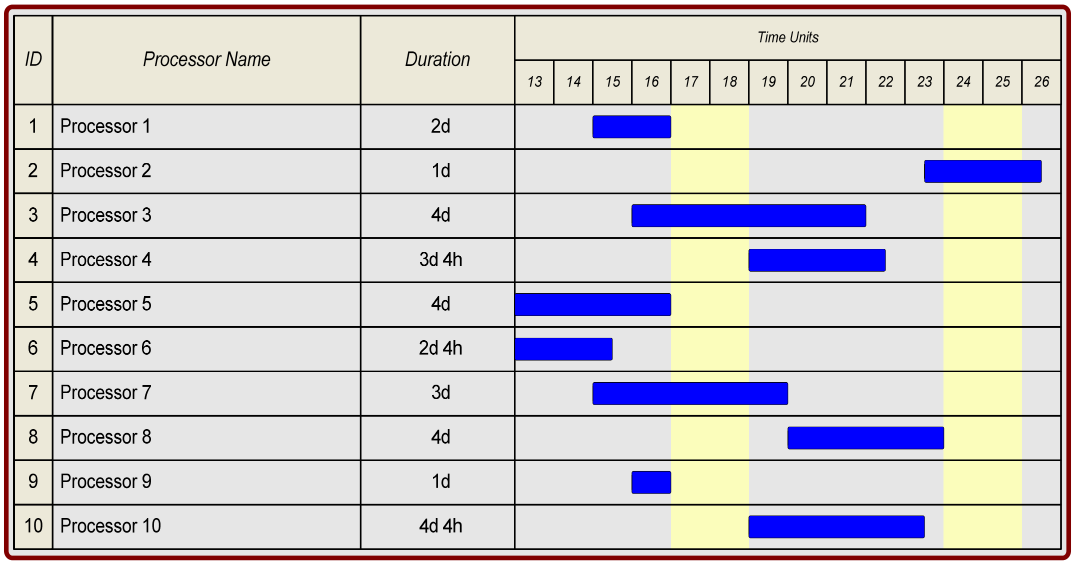
Parallel and Distributed Computing Laboratory

University of Puerto Rico at Mayaguez (UPRM)

March 14, 2007

Problem formulation...

Harnessing the power of wide-area distributed computing platforms is a major challenge nowadays, and **scheduling** is crucial for achieving this goal



ID	Processor Name	Duration	Time Units													
			13	14	15	16	17	18	19	20	21	22	23	24	25	26
1	Processor 1	2d														
2	Processor 2	1d														
3	Processor 3	4d														
4	Processor 4	3d 4h														
5	Processor 5	4d														
6	Processor 6	2d 4h														
7	Processor 7	3d														
8	Processor 8	4d														
9	Processor 9	1d														
10	Processor 10	4d 4h														

Traditional **scheduling** minimizes the **makespan** of the execution of a given set of jobs. In most practical situations, the exact computation of a minimal makespan is **NP-Hard**.

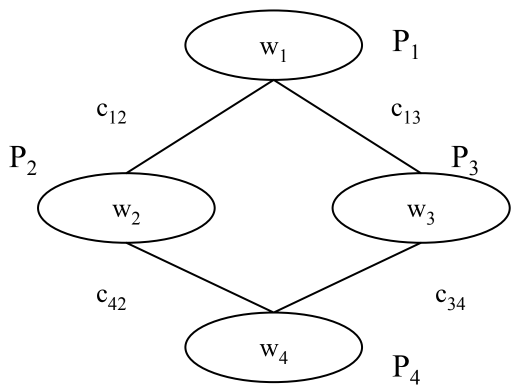
Proposed Solution

A computing platform is represented by a non oriented graph and a linear program maximizes the **throughput** in the system

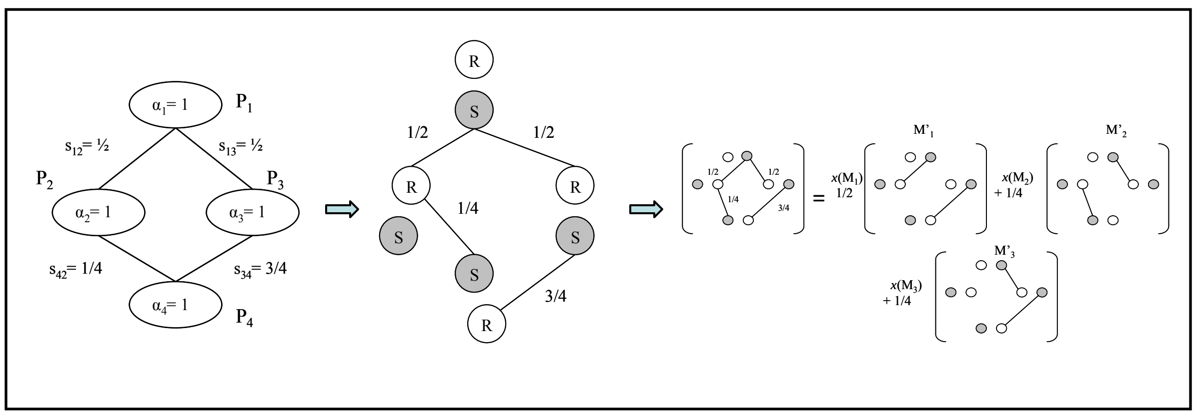
$$\text{Maximize } n_{\text{task}}(\boldsymbol{G}) = \sum_{i=1}^{p} \frac{\alpha_i}{w_i}$$

Subject to

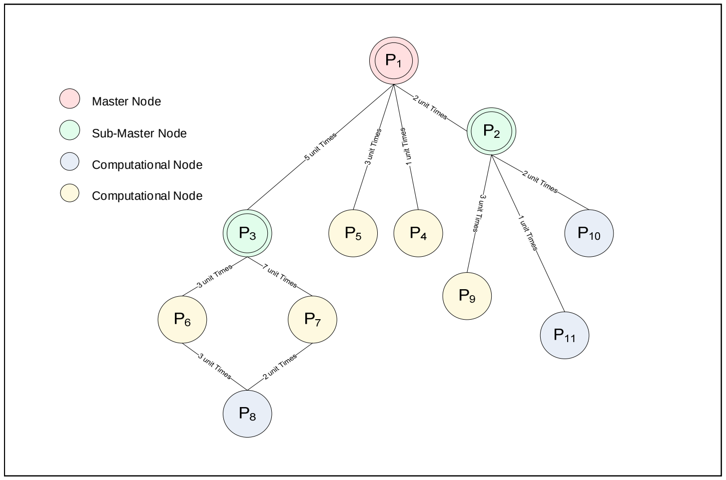
$$
\begin{array}{ll}
\forall i, & 0 \le \alpha_i \le 1 \\
\forall i, \forall j \in n(i), & 0 \le s_{ij} \le 1 \\
\forall i, \forall j \in n(i), & 0 \le r_{ij} \le 1 \\
\forall e_{ij} \in E, & s_{ij} = r_{ij} \\
\forall i, & \sum_{j \in n(i)} s_{ij} \le 1 \\
\forall i, & \sum_{j \in n(i)} r_{ij} \le 1 \\
\forall e_{ij} \in E, & s_{ij} + r_{ij} \le 1 \\
\forall i \ne m, & \sum_{j \in n(i)} \frac{r_{ij}}{c_{ij}} = \frac{\alpha_i}{w_i} + \sum_{j \in n(i)} \frac{s_{ij}}{c_{ij}} \\
\forall i \in n(m), & r_{mj} = 0
\end{array}
$$

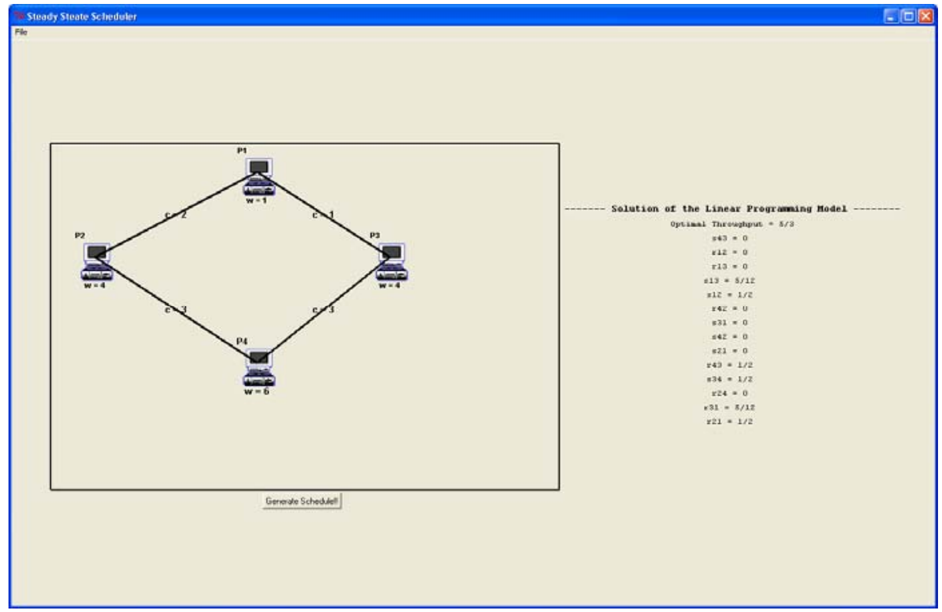


The actual schedule is found in **polynomial** time



Demo construction



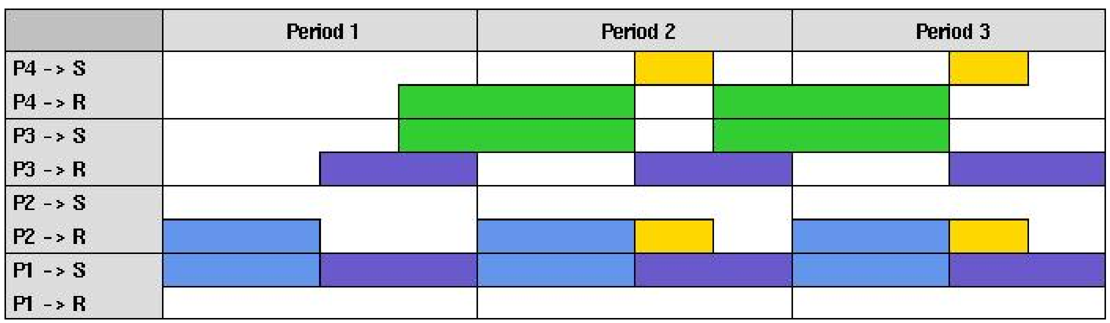


Steady State Scheduler V1.0 :

- Allows to change communication and execution times.
- Uses Glpk® to solve the Master-Slave linear programming problem.
- Constructs the actual schedule.

The Next step

By using the demo, we could identify some observables in the system and the subsequent behavior based on the model. We expect to evaluate a problem with more nodes, simulating a grid based platform, and we will use this results in order to construct a final prototype



Gannt Chart of the new schedule