Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment

Presenters: Forrest Iandola and Ilge Akkaya

(thanks to Marco Di Natale and Tarek Abdelzaher)

C.L. Liu[1], James W. Layland[2]

[1] MIT and UIUC

[2] NASA JPL

Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment

(The Liu-Layland Bound)

Presenters: Forrest Iandola and Ilge Akkaya

(thanks to Marco Di Nitale and Tarek Abdelzaher)

C.L. Liu[1], James W. Layland[2]

[1] **MIT and UIUC**

[2] **NASA JPL**

Outline

Goals

- Synthesize Marco Di Nitale's discussion of the Liu-Layland bound
- Review the key takeaways on this topic

Roadmap

- State the Liu-Layland bound
- Do some short examples, build up intuition
- …and move on!

Is a task set schedulable?

Schedulability analysis strategies

- Utilization bounds (easy but pessimistic)
  - Lower utilization -> easier to meet deadlines
  - Higher utilization -> harder to meet deadlines
- Exact analysis (optimal, but NP-hard)
- Heuristics

Is a task set schedulable?

Schedulability analysis strategies

- Utilization bounds (easy but pessimistic)
  - Lower utilization -> easier to meet deadlines
  - Higher utilization -> harder to meet deadlines
- Exact analysis (optimal, but NP-hard)
- Heuristics

The Liu-Layland Bound

- Utilization bound for *n* periodic tasks:

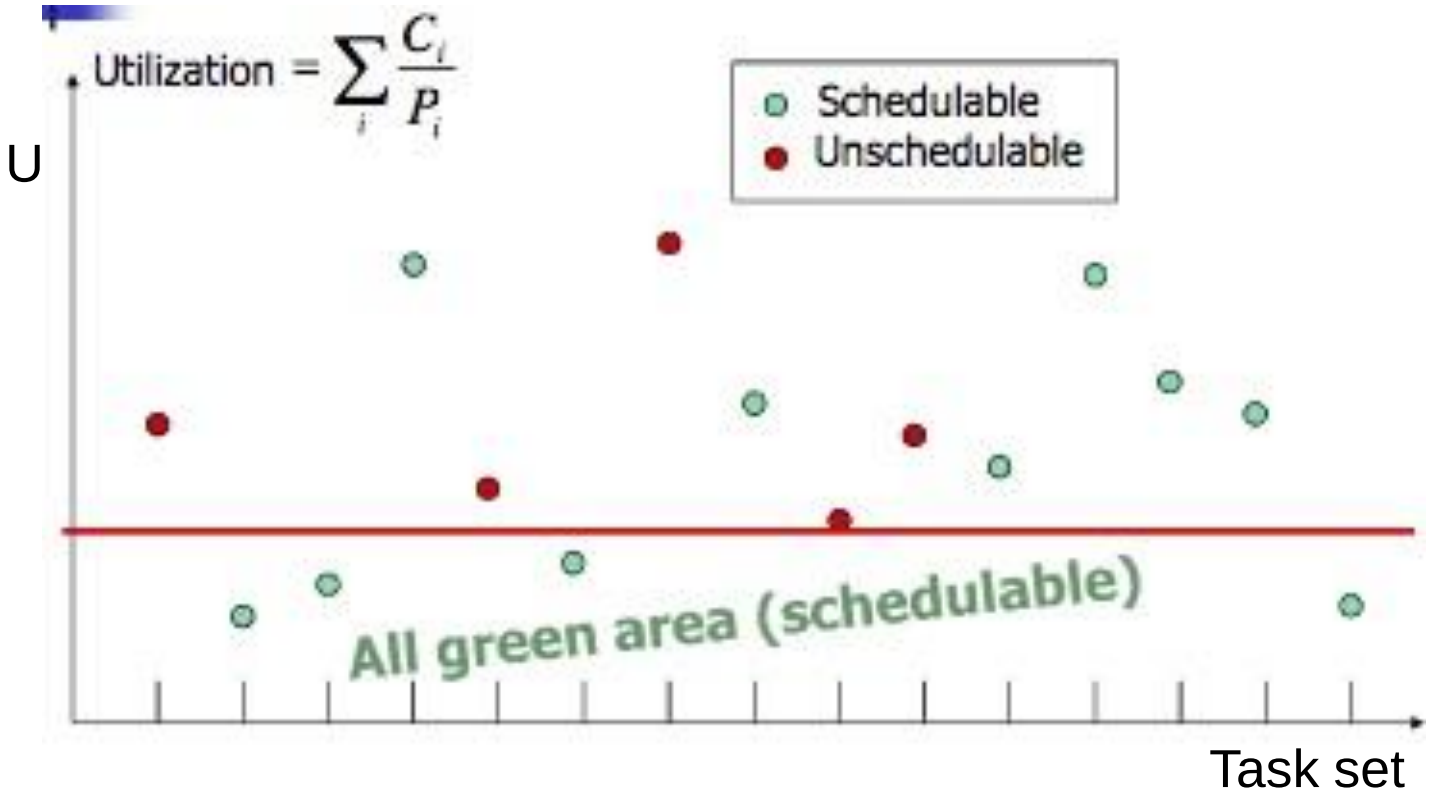
$$U = n\left(2^{1/n} - 1\right)$$

- For *n*=2 tasks, U = 0.83
- As *n* goes to infinity, U = 0.69

- Assumption: fixed-priority scheduling

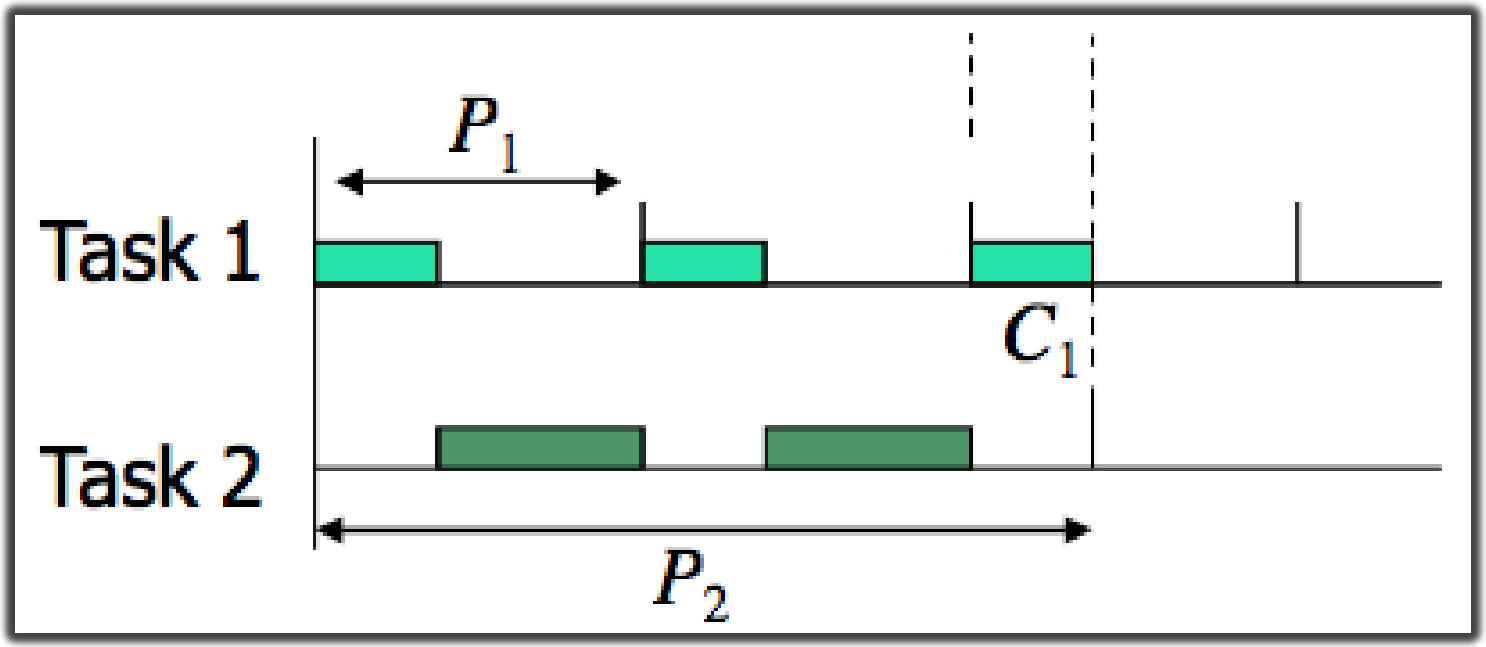
The bound is NOT “if and only if.”

- If system utilization is within the bound, it is *guaranteed* to be schedulable
- If system utilization exceeds the bound, it *may* be schedulable



Key insight: Blocking time

- 2 tasks, both arrive at the same time
- Notice that the lower-priority task waits for a long time



Conclusion

- Liu-Layland bound:

$$U = n\left(2^{1/n} - 1\right)$$

- Assuming worst-case arrival and blocking times leads to less-than-100% utilization bound
- If a system exceeds the bound, it may still be schedulable