Analysis of Federated and Global Scheduling For Parallel Real-Time Tasks

Jing Li[*], Jian-Jia Chen[!], **Kunal Agrawal**[*], Chenyang Lu[*], Chris Gill[*] and Abusayeed Saifullah[*]

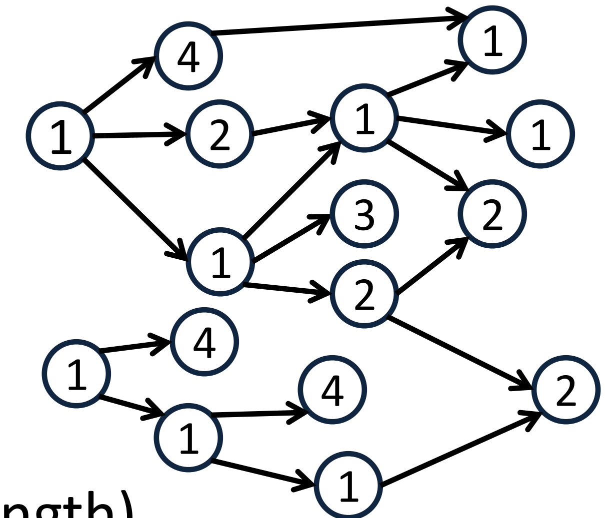
[*]Washington University in St. Louis
[!]TU Dortmund University

Problem: Scheduling Hard-Real Time DAG Tasks

Schedule task set $\tau = \{\tau_1, \tau_2, \ldots, \tau_k\}$ on m identical cores.

Each task τ_i is a DAG

- Nodes: sequential subtasks
- Edges: dependences.



C_i: Execution time on 1 core (total work)

L_i: Execution time on ∞ cores (critical-path length)

D_i: Deadline/minimum inter-arrival time

$C_i = 31$

$L_i = 6$

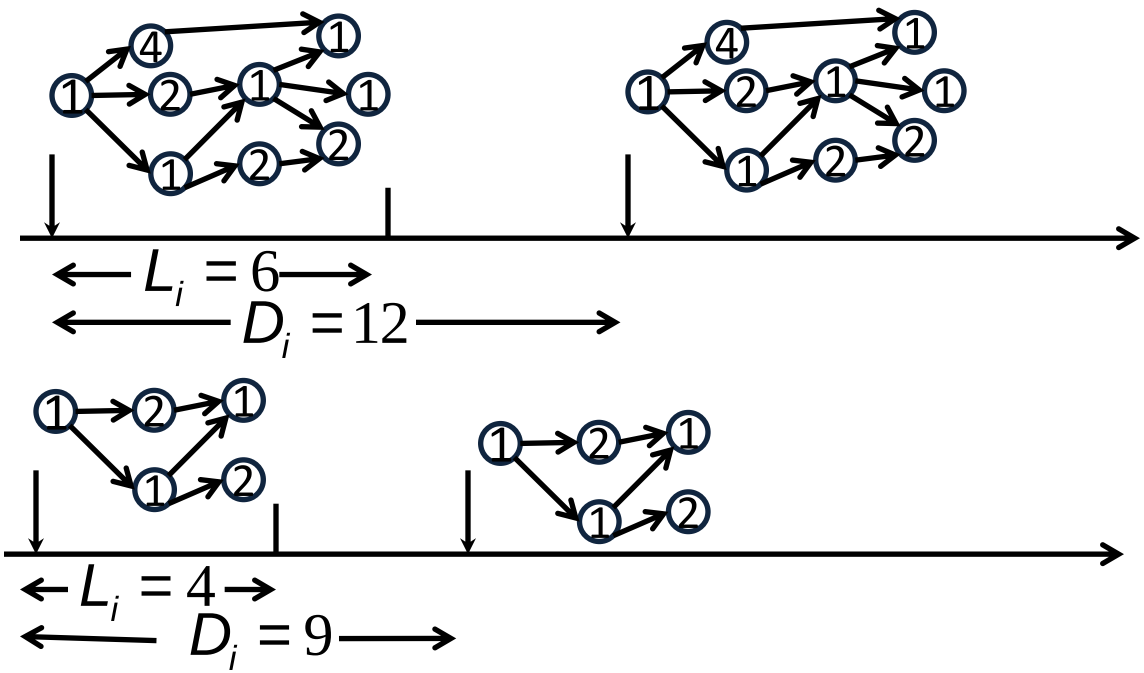
Utilization of τ_i: $u_i = C_i / D_i$

Total utilization of task set: $U_\Sigma = \sum_i u_i$

Performance Criterion: Capacity Augmentation Bound

A scheduler S provides a *capacity augmentation bound* of α if it can always schedule a task set τ on m processors if:

(a) For each task $L_i \le D_i / \alpha$



(b) $U_S \le m / \alpha$

Notes: No scheduler can provide $\alpha < 1$.

The conditions do not depend on the structure of the DAG.

CONTRIBUTIONS

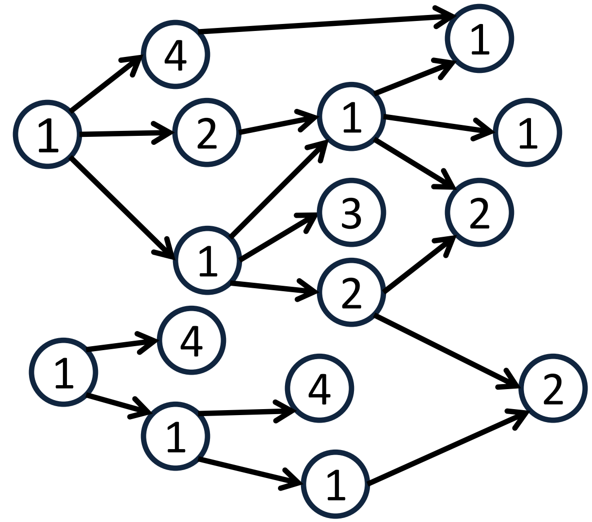
Scheduler	Prior Work	This Paper
Federated		Upper bound: $\alpha \le 2$ Lower bound: $\alpha > 2 - 1/m$
Global EDF	Resource augmentation (speedup) bound ≤ 2 Schedulability test [BMSW13] Upper bound: $\alpha \le 4$ Lower bound: $\alpha \ge \left(3+\sqrt{5}\right)/2 \approx 2.618$ for large m [LALG13]	Upper bound: $\alpha \le \left(3+\sqrt{5}\right)/2$ for large m Improved lower bound for small m
Global RM	Resource augmentation ≤ 3 Schedulability test [BMSW13] For synchronous tasks (a subset of DAG tasks), $\alpha \le 2+\sqrt{3}$ for large m (using decomposition and DM)	Upper bound: $\alpha \le 2+\sqrt{3} = 3.73$ for large m

OUTLINE

- **Canonical form of a DAG task.**
- Federated Scheduling
- Upper Bound on GEDF

High vs. Low-Utilization Tasks

- Classify task as
  - Low-utilization if $u_i \leq 1$
  - High-utilization if $u_i > 1$
- Low utilization tasks can execute sequentially and still meet their deadlines.
- High utilization tasks need parallelism to complete within their deadline.

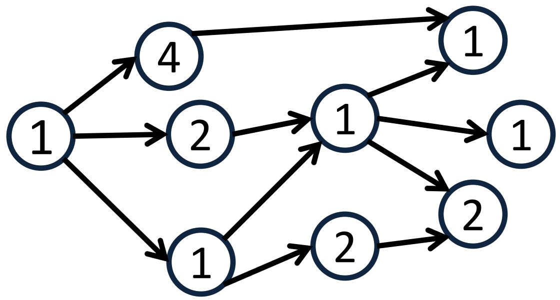


$C_i = 31$

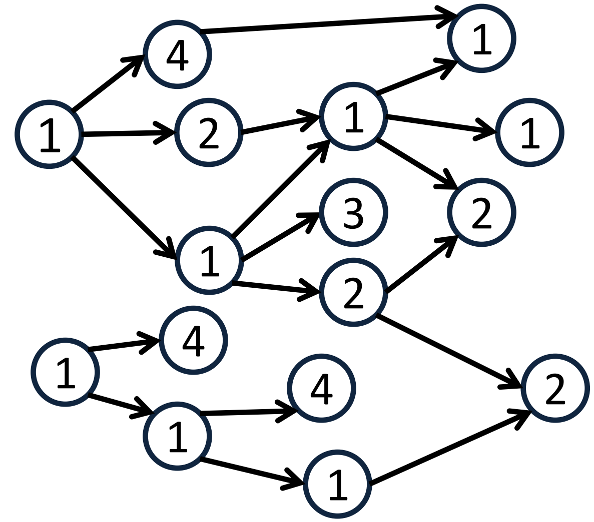
$L_i = 6$

- Case 1: $D_i = 32; u_i = 0.96$
  - Low utilization.
- Case 2: $D_i = 18; u_i = 1.72$
  - High utilization.

Key Intuition: Canonical Form Of a DAG Task

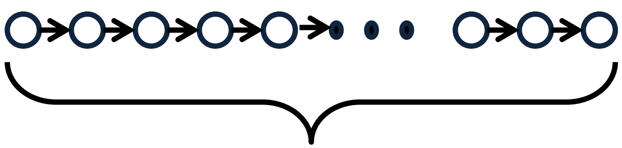


$C_i = 15$
$L_i = 6$
$D_i = 18$
$u_i = 0.83$

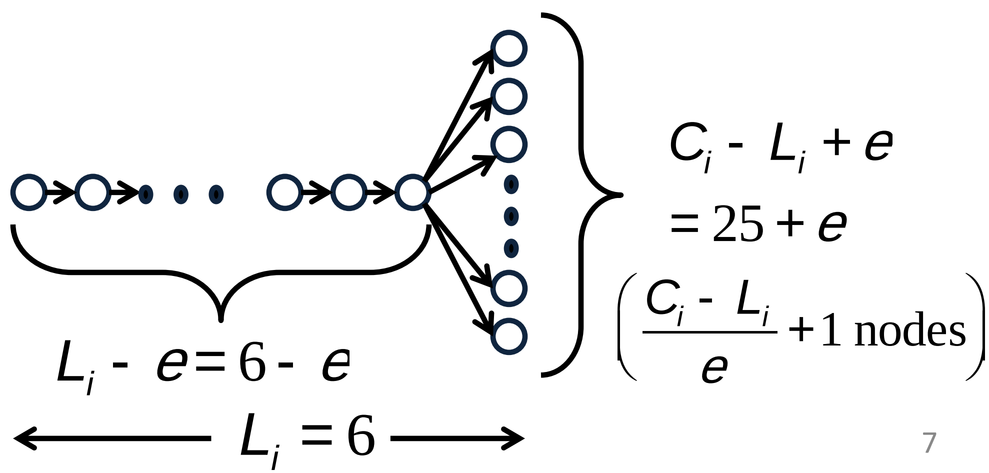


$C_i = 31$
$L_i = 6$
$D_i = 18$
$u_i = 1.72$

LU task: A sequential task with work C_i



HU task: A chain of $(L_i / e) - 1$ nodes of size e. All remaining work is maximally parallel with e-sized nodes.



CANONICAL DAG IS THE "WORST-CASE" DAG

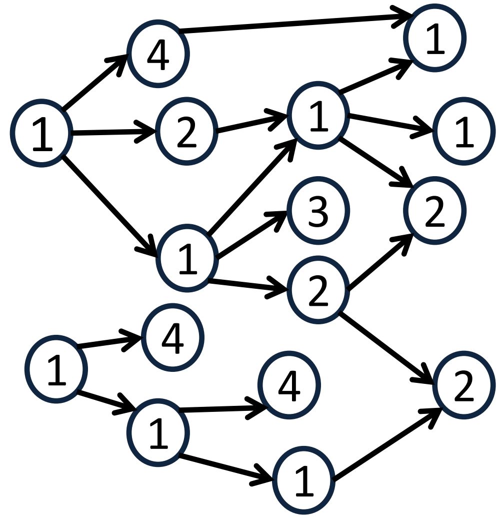
Pretend we give the job ∞ processors.

DEFINE

- $work_i(t)$: maximum work that task τ_i has to finish in any interval of time t.
- $work_i^*(t)$: maximum work that task τ_i's canonical form has to finish in any interval of time t.

We can prove that for all t

$work_i^*(t) \geq work_i(t)$



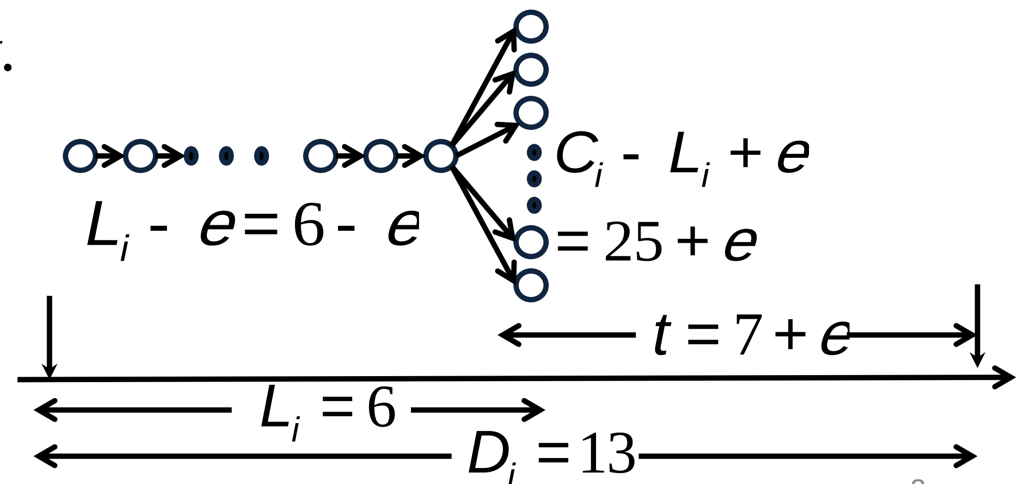
$C_i = 31; L_i = 6$

Consider

$t = D_i - L_i + e$:

$work_i(t) = 4e$

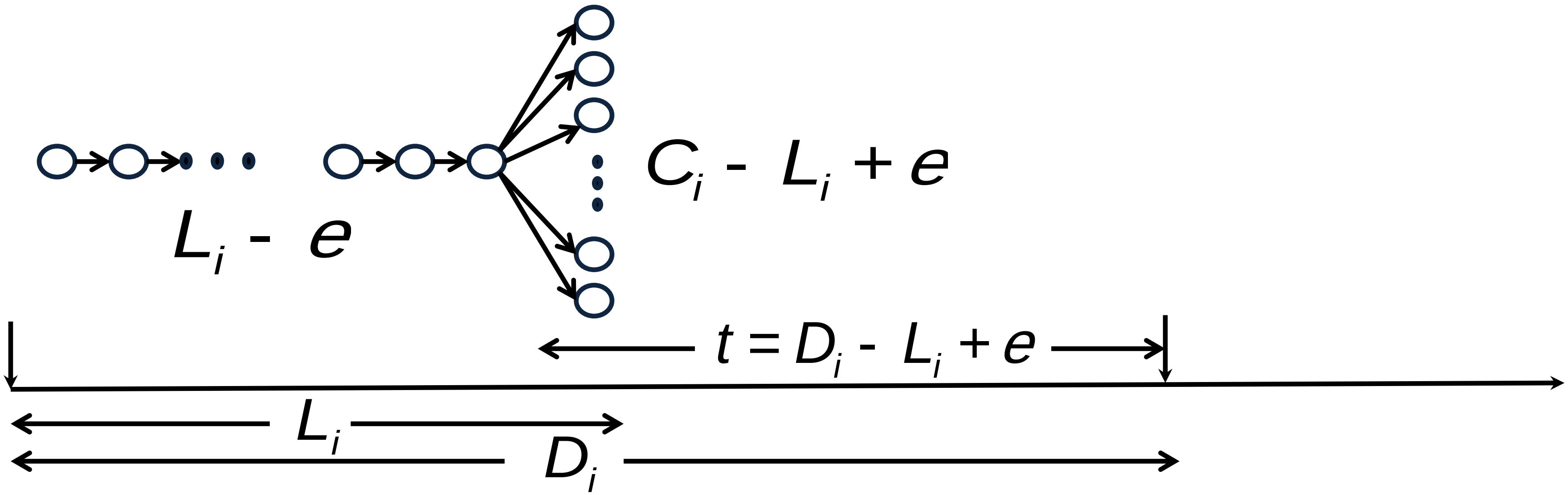
$work_i^*(t) = 25 + e$



OUTLINE

- Canonical form of a DAG task.
- **Federated Scheduling**
- Upper Bound on GEDF

How Many Cores Does A High-Utilization Task Need If It Is The Only Task In The System?



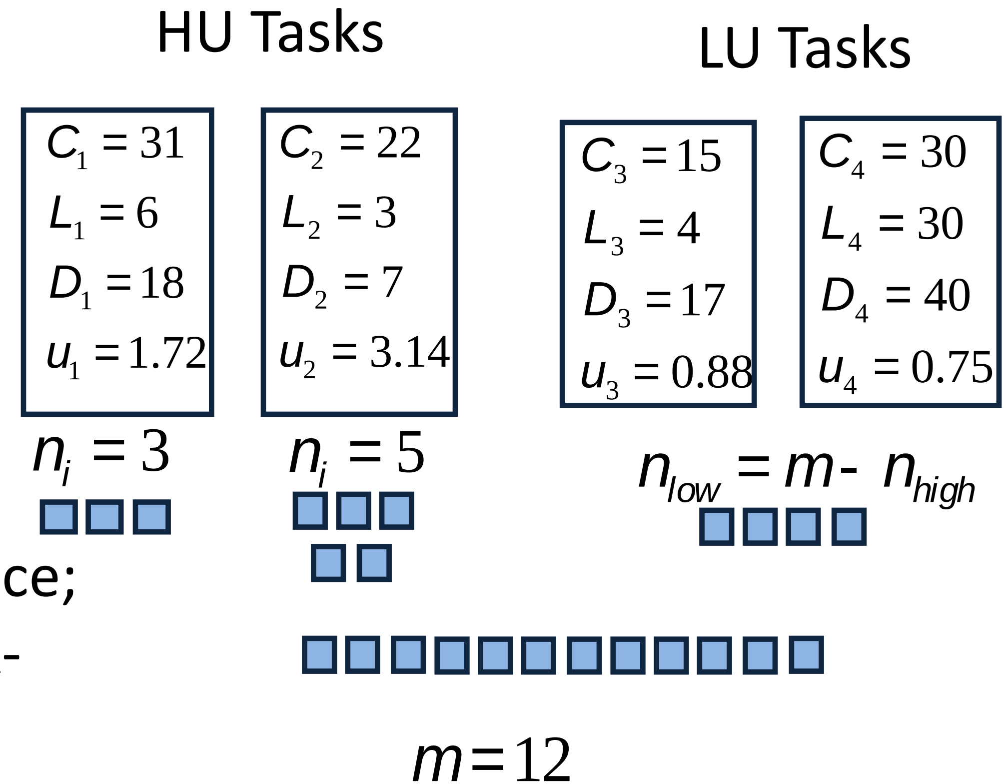
For $t = D_i - L_i + e$, $work_i^*(t) = C_i - L_i + e$.

It needs $n_i = \dfrac{C_i - L_i + e}{D_i - L_i + e} \le \left\lceil \dfrac{C_i - L_i}{D_i - L_i} \right\rceil$ (for small enough ε)

We can prove that on n_i dedicated cores and using a work-conserving scheduler, an HU task never misses a deadline.

FEDERATED SCHEDULER

- Assign $n_i = \left\lceil \frac{C_i - L_i}{D_i - L_i} \right\rceil$ dedicated cores to each high-utilization task τ_i.
- All remaining processors are assigned to low-utilization tasks collectively.



No interference; Use any work-conserving scheduler.

Treat as sequential tasks and use multiprocessor scheduler such as P-EDF.

Capacity Augmentation Bound of $\alpha \leq 2$

1. For HU tasks, show that, if $L_i \leq D_i/2$ (using algebra):

$$n_i = \left\lceil \frac{C_i - L_i}{D_i - L_i} \right\rceil < 2u_i$$

2. Therefore, $n_{low} = m - \sum_{\tau_i:high} n_i \geq m - 2\sum_{\tau_i:high} u_i = m - 2u_{high}$

3. If $m \geq \alpha U_\Sigma = 2U_\Sigma = 2u_{high} + 2u_{low}$, we have $n_{low} \geq 2u_{low}$

4. There are many schedulers, such as partitioned EDF [LDG04] and various fixed priority schedulers [ASJ01, AJ03] that guarantee schedulability to sequential tasks if utilization is at most 50%. Any of these can be used to schedule the low-utilization tasks with a total utilization of $n_{low}/2$ on n_{low} cores.

5. Checking schedulability for federated scheduler is fast and easy. It often admits task sets with utilization $> m/2$.

OUTLINE

- Canonical form of a DAG task.
- Federated Scheduling
- **Upper Bound on GEDF**

Bound The Total Load Of Canonical Tasks

For all tasks, we bound $\frac{work_i^*(t)}{t}$

For LU tasks, $\frac{work_i^*(t)}{t} \le C_i / D_i = u_i$

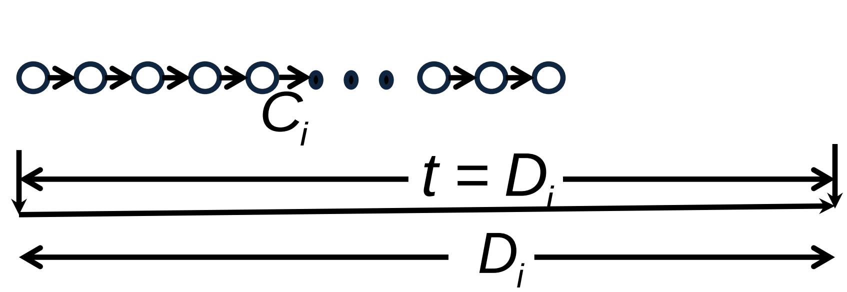
For HU tasks,

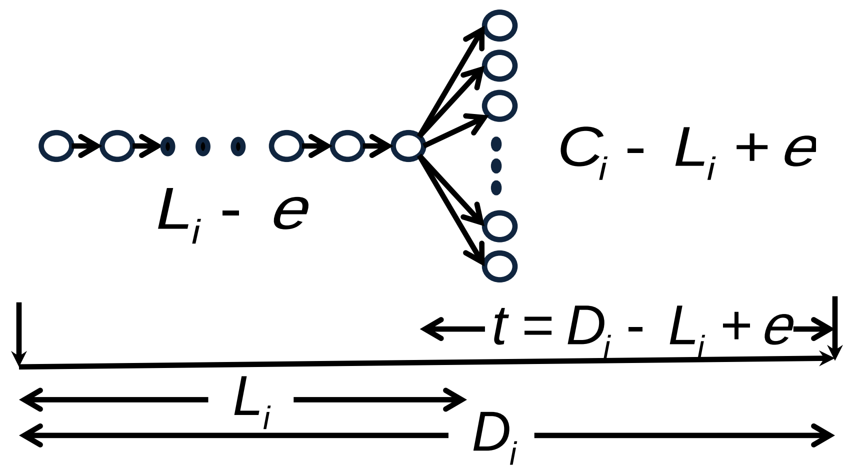
$$\frac{work_i^*(t)}{t} \le \frac{C_i - L_i + \varepsilon}{D_i - L_i + \varepsilon} \approx \frac{C_i - L_i}{D_i - L_i}$$

If $D_i \ge \alpha L_i$,

$$\frac{work_i^*(t)}{t} \le \frac{C_i}{D_i - D_i/\alpha} = \frac{u_i}{1 - 1/\alpha} = \frac{\alpha u_i}{\alpha - 1}$$

Over all tasks, $\sum_t work_i^*(t) \le \frac{\alpha U_S}{\alpha - 1} t$





GEDF Has Capacity Augmentation Bound $\alpha \le 2.618$

1. Bonifaci et. al [BMSW13] proved that τ is schedulable by GEDF on m processors if
   - $\alpha L_i \le D_i$, and
   - $\sum_t work_i(t) \le \frac{\alpha m - m + 1}{\alpha} t$

1. We know that $\sum_t work_i(t) \le \sum_t work_i^*(t) \le \frac{\alpha U_\Sigma}{\alpha - 1} t$

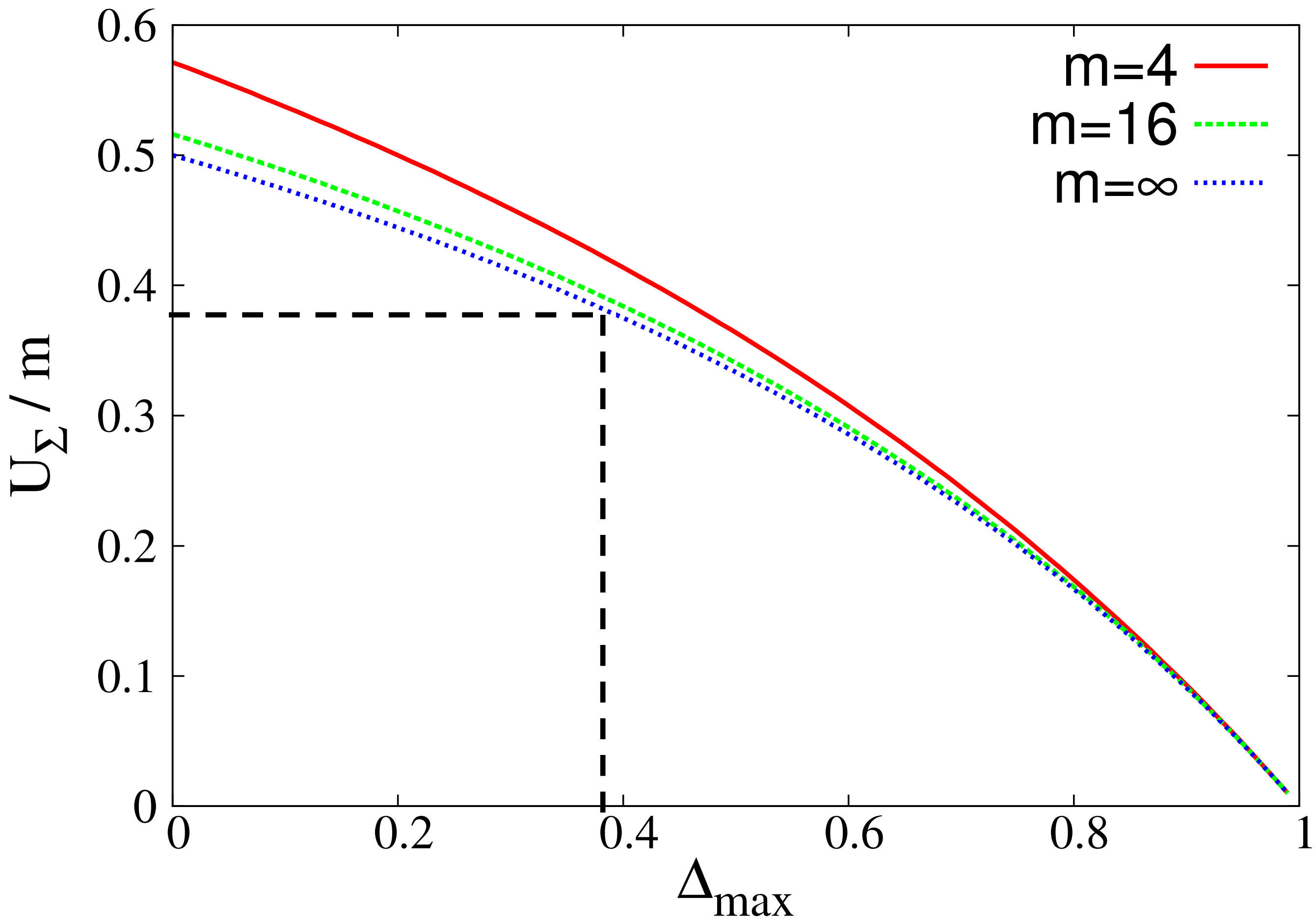
1. Therefore, the task set is schedulable if $\frac{\alpha U_\Sigma}{\alpha - 1} \le \frac{\alpha m - m + 1}{\alpha}$

1. We substitute $U_\Sigma \le m/\alpha$ and solve for α to get

$$\alpha \le \frac{3 - 1/m + \sqrt{5 - 2/m + 1/m^2}}{2} \approx \frac{3 + \sqrt{5}}{2}$$

EXTENSION TO GEDF ANALYSIS

- With simple extensions, we can show that if $D_{max} = \max_i \{L_i / D_i\}$ is "small", then EDF also provides utilization close to *m/2*.



Conclusions and Future Work

- The canonical DAG allows us to ignore the DAG structure --- we need only know the upper bounds on execution time C_i and critical path length L_i.
- Federated scheduler has close-to-optimal capacity augmentation bound for large m. What about small m?
- For global RM for parallel tasks, the best lower bound is 2.668 (inherited from sequential tasks) [L02], while the upper bound is 3.73. Can we improve either?
- We have speedup bounds for constrained and arbitrary deadline parallel tasks. Can we prove utilization/capacity augmentation bounds for these tasks?