Makeflow Evaluation for Efficient Pre-Allocation of Resources

Haiyan Meng, Olivia Choudhury

Motivation

Two steps of executing a makeflow:

1. Start your makeflow

   % makeflow -T wq example.makeflow
2. Submit worker processes to execution engine

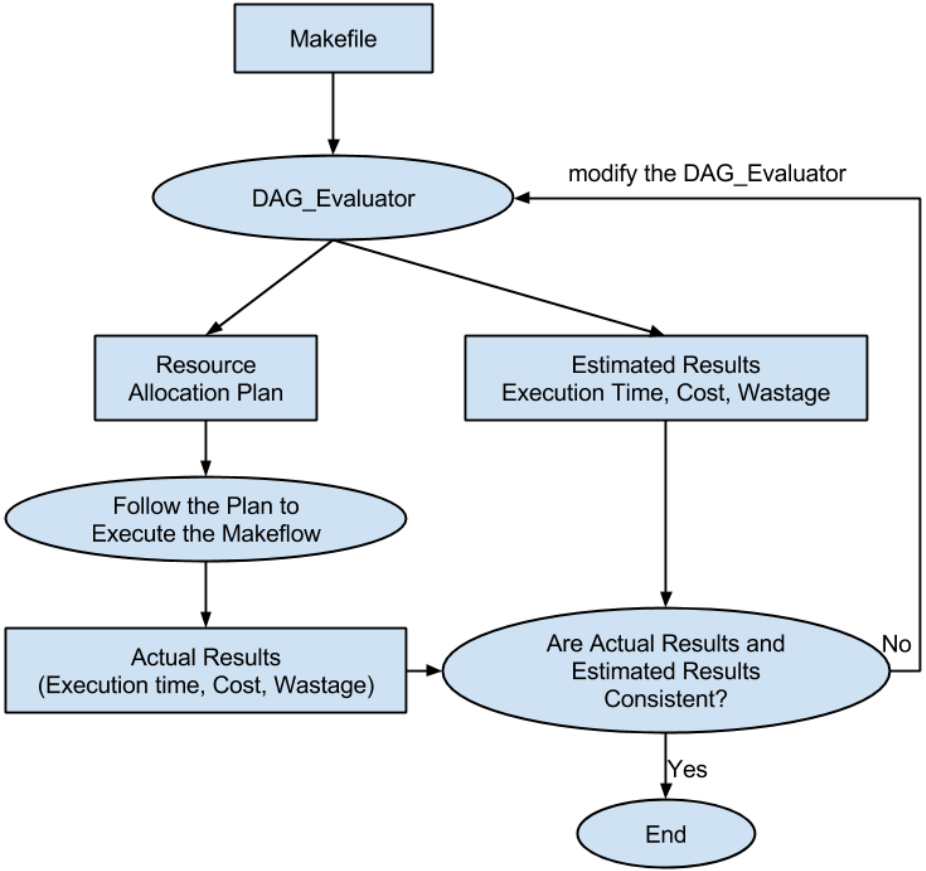
   % condor_submit_workers barney.nd.edu 9123 **10**

Problem: Why do we allocate 10 worker machines? Why not 1 or 100?

Aim:

Given a makefile, decide optimal number of machines to be allocated in **reasonable** time.

Methods



DAG_Evaluator Factors:
Total task number
DAG height
Average DAG width = Total task number / DAG height
Width of each level
DAG Width = max {width of each level}
Distribution of level widths
Task Durations
Task Dependency Relationship

Measurements:
Time of DAG_Evaluator
Real cost = sum {each task duration}
Execution time = time to finish the whole DAG
Total cost = Execution time * machine number
Idle cost = Total cost - Real cost
Wastage = Idle cost / Total cost

Core Algorithm:
Topological sort

DAG Categories

	Task durations **known** in advance	Task durations **not known** in advance
Tasks in the same level require **similar** time	Case 1	Case 3
Tasks in the same level require **different** times	Case 2	Case 4

For all cases:

Given a makefile, decide a near-optimal number of machines (DAG_Evaluator).

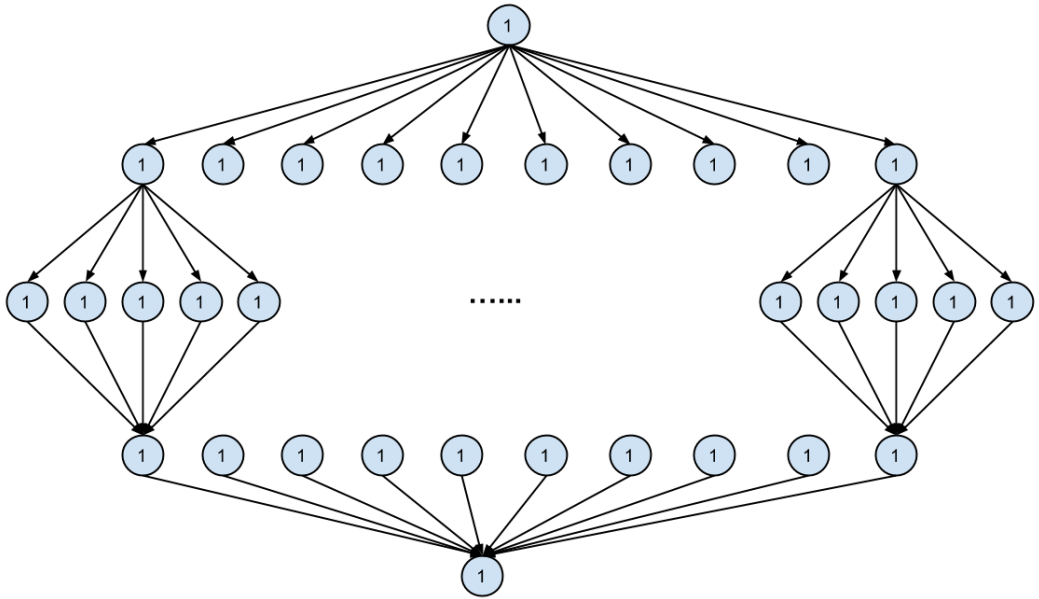
For cases 1 and 2:

Given a makefile and number of machines, estimate the execution time, total cost, and wastage (Task_Scheduler).

Case 1

Factors:

1. Width of each level
2. Distribution of level widths
3. Task duration at each level
4. Task Dependency Relationship

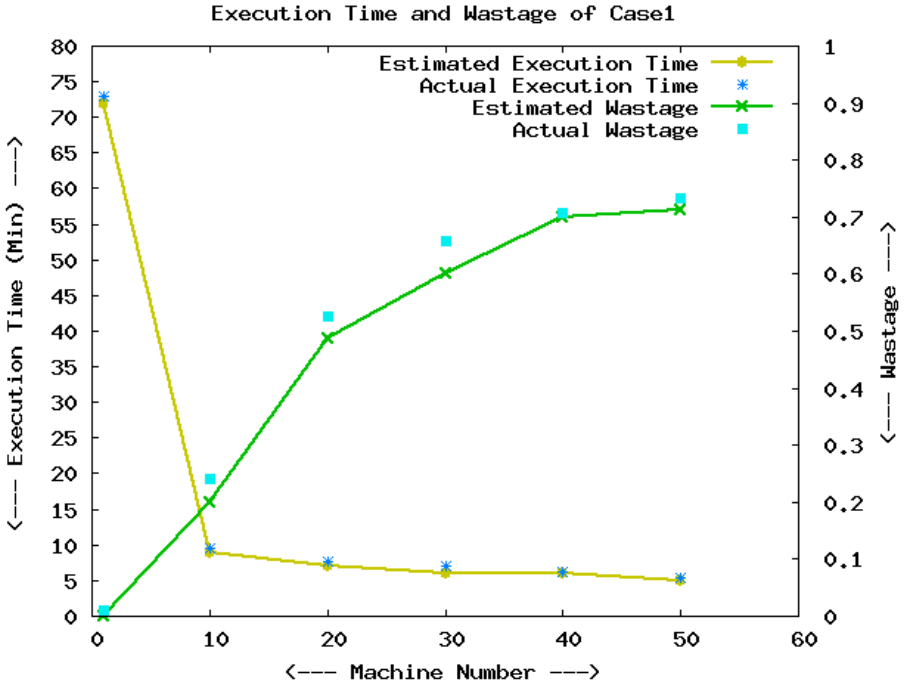


$$Machine\ number = \sum_{1 \le i \le n} width\ of\ level\ i \times \frac{total\ time\ taken\ at\ level\ i}{real\ cost}$$

$$total\ time\ taken\ at\ level\ i = width\ of\ level\ i \times each\ task\ duration\ of\ level\ i$$

Case 1

Output of DAG_Evaluator:	
Real cost:	72 mins
Maximum machines used:	50
Execution time:	5 mins
Total cost:	250 mins
Wastage (idle cost / total cost):	71.2%
Optimal number of machines:	**38**
Execution time:	6 mins
Total cost:	228 mins
Wastage (idle cost / total cost):	68.4%



Estimations based on Task_Scheduler

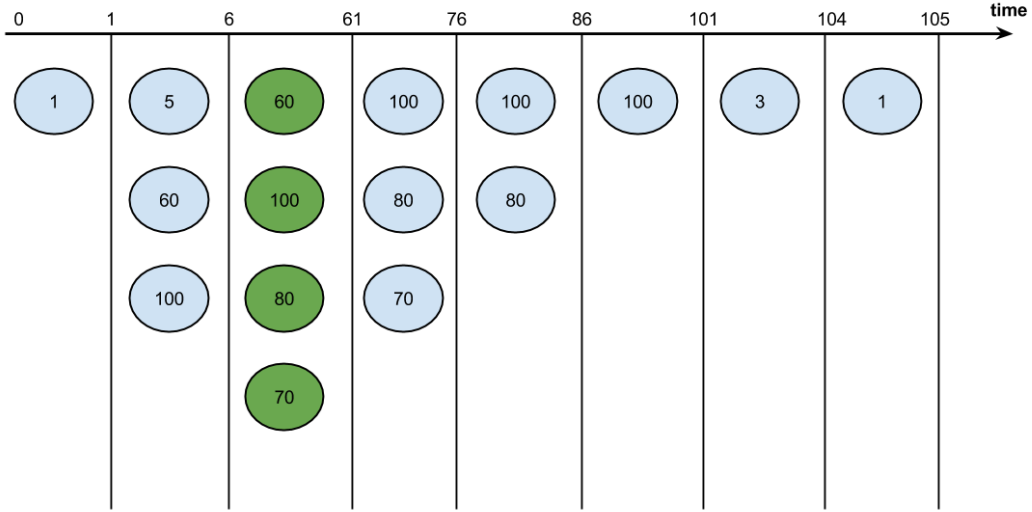
Case 2

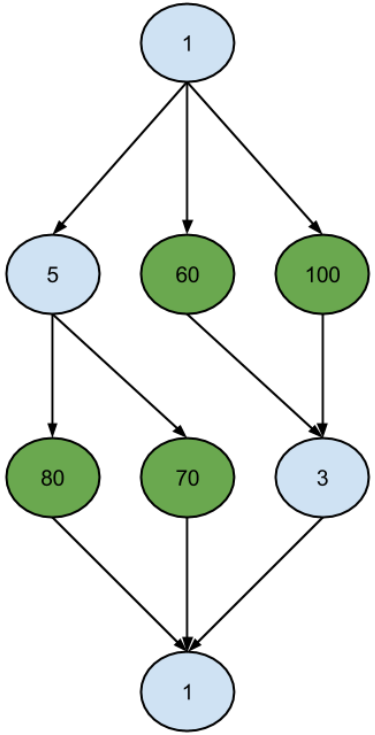
Factors:

1. Task duration
2. Task Dependency Relationship

$$Machine\ number = \sum_{1 \le i \le n} width\ of\ time\ interval\ i \times \frac{total\ task\ duration\ of\ time\ interval\ i}{real\ cost}$$

Total task duration of time interval i = length of time interval i x width of time interval i





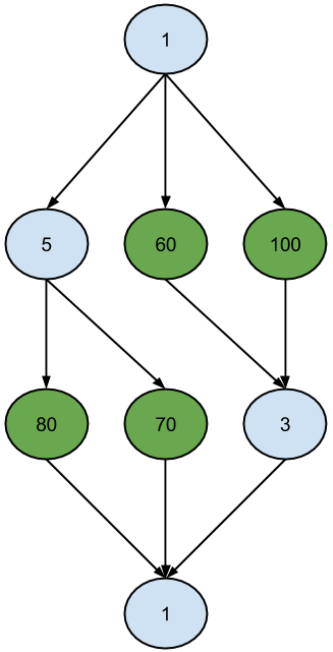
Case 2

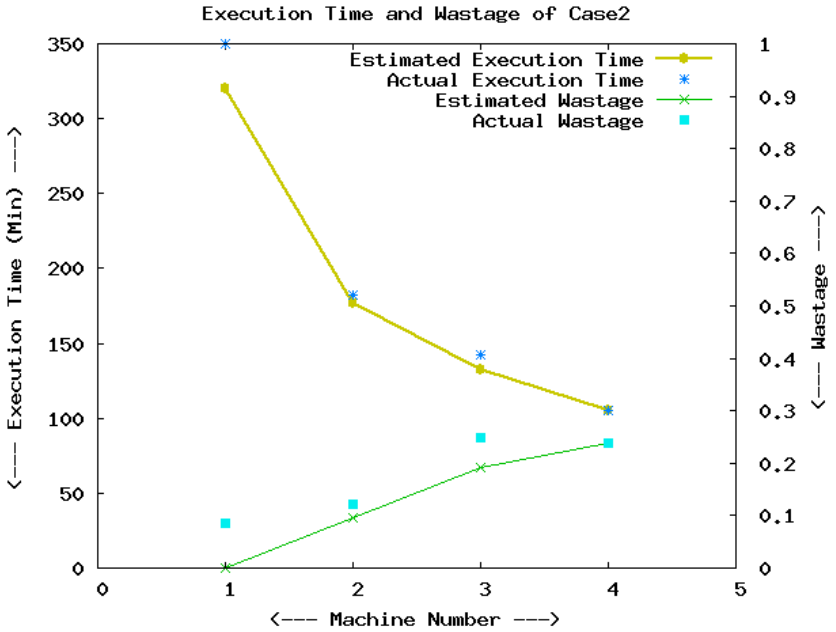
Output of DAG_Evaluator:

Real cost:	320 mins
Max Used Machines:	4
Execution time:	105 mins
Total cost:	420 mins
Wastage:	23.8%

Optimal number of machines: 4

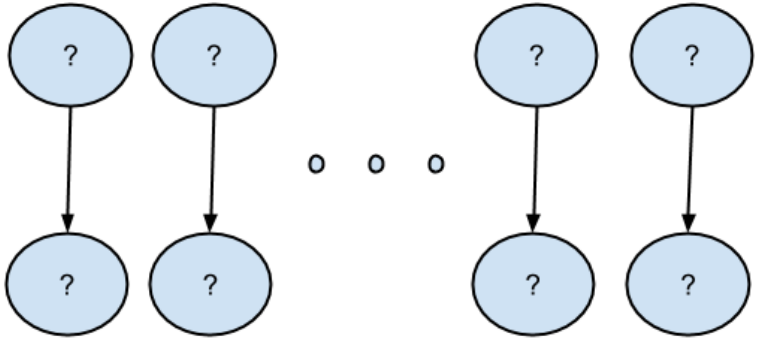
Execution time:	105 mins
Total cost:	420 mins
Wastage:	23.8%



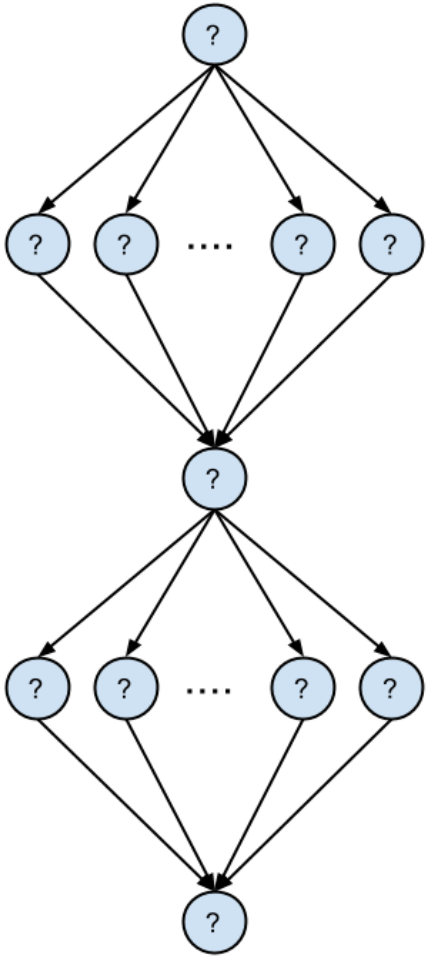


Estimations based on Task_Scheduler

Pilot tasks



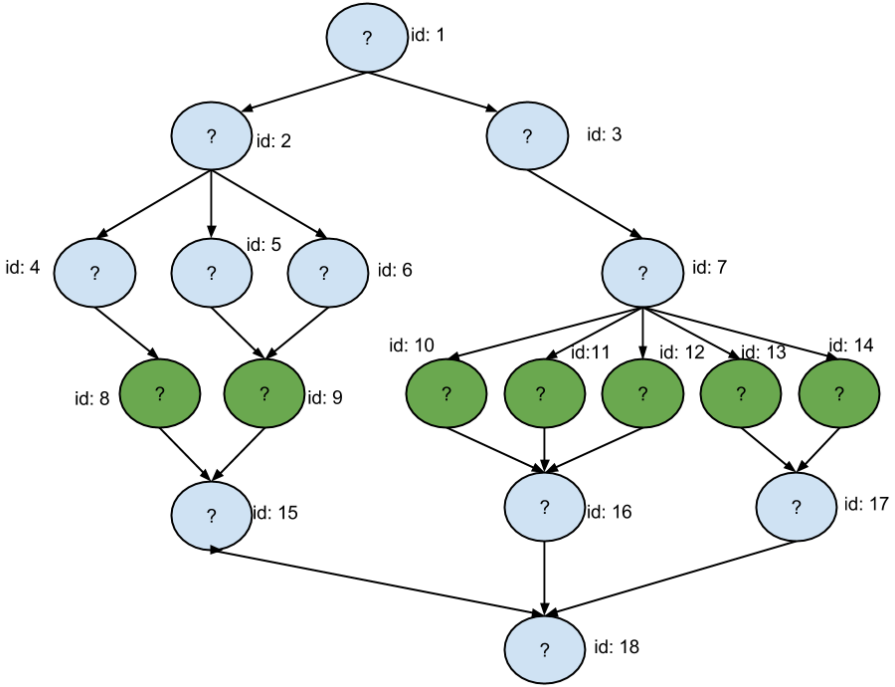
	Avg. Time (Mins.)	S.D.
Level 1	53.4	34.7
Level 2	29.7	14.1



Case 3

Factors:

1. Tasks in the same level require similar time
2. Task Dependency Relationship



Output of DAG_Evaluator:
Number of machines: 7

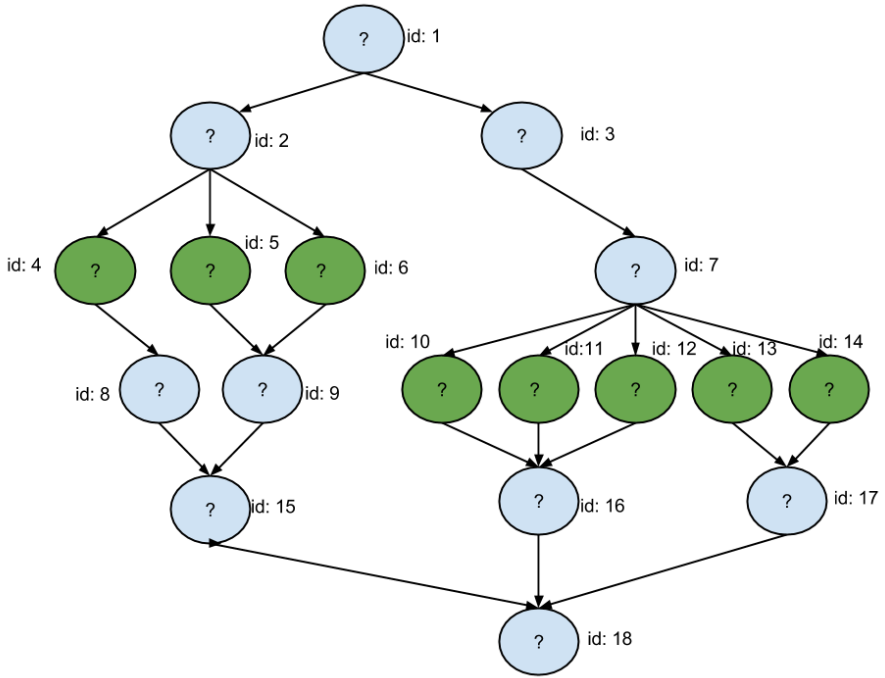
Topological sort (Task queue):

1
2, 3
3, 4, 5, 6
4, 5, 6, 7
5, 6, 7, 8
6, 7, 8
7, 8, 9
8, 9, 10, 11, 12, 13, 14
9, 10, 11, 12, 13, 14
10, 11, 12, 13, 14, 15
11, 12, 13, 14, 15
12, 13, 14, 15
13, 14, 15, 16
14, 15, 16
15, 16, 17
16, 17
17
18
NULL

Case 4

Factors:

1. Tasks in the same level require different times (times not known in advance)
2. Task Dependency Relationship



Output of DAG_Evaluator:

(1)	Total task number:	18
	DAG height:	6
	Average DAG width:	3
(2)	Distribution of level width:	1, 2, 4, 7, 3, 1
	DAG width:	7

Number of machines= max {Average DAG width, DAG width} = 7

Summary

DAG Categories	Approach
Similar time, known	DAG_Evaluator, Task_Scheduler
Different times, known	DAG_Evaluator, Task_Scheduler
Similar time, not known	Topological Sort
Different times, not known	DAG width, DAG height, Average DAG width, Total task number

DAG_Evaluator: Evaluates optimal number of machines.
Task_Scheduler: Estimates execution time, cost, wastage of a DAG for a given number of machines.

Future Work

- For Case 1 and Case 2, integrate DAG_Evaluator and Task_Scheduler
  - Optimize execution time, cost, and wastage
- For Case 4, develop an algorithm to find maximum possible concurrent tasks in a DAG
  - Maximum number of machines
  - Estimate execution time, cost, wastage
  - Compare with actual execution time, cost, wastage
- For Case 3 and Case 4, design test cases