1. Suppose you magically have a base-3 computer. Consider this custom float6 representation using base-3 digits:

\[
(-1)^{sign} \times 3^{exponent-4} \times (1 + \sum_{i=1}^{3} d_{3-i}3^{-i})
\]

What is the decimal real number value (round to 3 decimal places) of these float6 bit sequences interpreted as above?

(a) 011012
(b) 120210

Solution:

(a) Given: Sign = 0, Exponent = 11, Fraction = 012 all in base 3
Converting each of them to base 10 we get:
Sign = 0, Exponent = 1 * 3^1 + 1 * 3^0 = 4, Fraction = 012 (leaving as is since individual bit used in calculation later. This is not in base 10 but base 3 still)

Float Number = \(-1^{sign} \times 3^{exponent-4} \times (1 + \sum_{i=1}^{3} d_{3-i}3^{-i})\)

\[
= -1^{0} \times 3^{11-4} \times (1 + d_{2}3^{-1} + d_{1}3^{-2} + d_{0}3^{-3})
\]

\[
= -1 \times 3^{7} \times (1 + (0 \times 3^{-1}) + (1 \times 3^{-2}) + (2 \times 3^{-3}))
\]

\[
= 1 \times 1 \times (1 + 0 + \frac{1}{9} + \frac{2}{27})
\]

\[
= 1 + 0 + \frac{1}{9} + \frac{2}{27}
\]

\[
= 1 + \frac{1}{9} + \frac{2}{27}
\]

\[
= \frac{32}{27}
\]

\[
= 1.185
\]
(b) Given: Sign = 1, Exponent = 20, Fraction = 210 all in base 3

Converting each of them to base 10 we get:

Sign = 1, Exponent = 20\cdot3^1 + 0\cdot3^0 = 6, Fraction = 210 (leaving as is since individual bit used in calculation later. This is not in base 10 but base 3 still)

\[
\text{Float Number} = -1^{\text{sign}} \cdot 3^{\text{exponent}-4} \cdot (1 + \sum_{i=1}^{3} d_{3-i} \cdot 3^{-i})
\]

\[
= -1^{\text{sign}} \cdot 3^{\text{exponent}-4} \cdot (1 + d_{2}3^{-1} + d_{1}3^{-2} + d_{0}3^{-3})
\]

\[
= -1 \cdot 9^{-4} \cdot (1 + (2 \cdot 3^{-1}) + (1 \cdot 3^{-2}) + (0 \cdot 3^{-3}))
\]

\[
= -1 \cdot 9^{-2} \cdot (1 + \frac{2}{3} + \frac{1}{9} + 0)
\]

\[
= -1 \cdot 9^{-1} \cdot \left( \frac{16}{9} \right)
\]

\[
= -9 \cdot \frac{16}{9}
\]

\[
= -16
\]
2. Consider this task graph with known task runtimes but all edges between tasks being hidden/unknown.

(a) Suppose the degree of parallelism is 6. Propose a plausible set of edges that conforms to this.

(b) Suppose the degree of parallelism is only 4. Propose a new plausible set of edges for this.

(c) For your task graph in B, what is the highest possible speedup on a cluster with task parallelism vs using just 1 worker with no idle times

Solution:

(a) 

(b) An interesting subquestion I thought off while answering this was what was the
lowest completion time task graph you can make with a degree of parallelism as 4. Above was my solution (basically keep longest path as short as possible. So nothing over 40 and 50). If you have an alternate answer feel free to discuss

(c) 1 worker = T1 + T2 + T3 + T4 + T5 + T6 = 5 + 10 + 15 + 20 + 40 + 50 = 140
4 workers = Longest Path = D - > T6 = 50

\[
\text{Speedup} = \frac{\text{Completion on 1 worker}}{\text{Completion on 4 workers}} = \frac{140}{50} = 2.8
\]
3. Suppose you need to run following classification workflow. Try two alternative feature engineering approaches; on each feature set, try both a Naive Bayes model and a Random Forest model. For the latter model, you try 4 different hyperparameter combinations each

(a) Draw the task graph for this whole workflow. Make sure to identify/label every task clearly.

(b) How many workers are necessary to achieve lowest-possible completion time with task parallelism no matter the tasks’ runtimes?

(c) Suppose the task runtimes are known: 20 and 30 units for the feat. eng. approaches; 5 units to build Naive Bayes on either feature set; 15 to 25 units to build Random Forest depending on the feature set and hyperparameter combination. What is the upper bound on the lowest-possible completion time for this workflow?

(d) What is the speedup when hitting the upper bound vs using just 1 worker with no idle times for your whole workflow?

Solution:

(a)

(b) The degree of parallelism in this graph is 10. Therefore 10 workers are necessary to achieve lowest possible completion time with task parallelism no matter the tasks’ runtimes

(c) The longest path will help find the lowest completion time. In the question it isn’t stated which ones are which. But it does give highest values so that would be 30 + 25 = 55 where this path would likely be one of the FE -> RF paths. Therefore the shortest completion time’s upper bound is 55 (can be lower if none of the RFs on the FE with 30 units take 25 units)

(d) As computed above the lowest completion time is 55. The single worker case will be (20 + 30)(For each feature engineering) + (5 + 5) (For each Naive Bayes model)
+ (25 * 8)(All of the RFs will be the worst case hyperparameter combination for the given feature sets) = 50 + 10 + 200 = 260.

\[
\text{Speedup} = \frac{\text{Completion on 1 worker}}{\text{Completion on 10 workers}} = \frac{260}{55} = 4.727
\]

4. Suppose you have tasks T1-T5 arriving at times 0, 0, 10, 10, and 20, respectively. Suppose Dask gave you the following Gantt chart for running this workload on a cluster of 3 workers.

Sorry for the quality. The one on Slides had a typo. May update later

(a) What is the average turnaround time?

(b) What is the speedup obtained on this cluster vs using just 1 worker with no idle times?

(c) Is it possible to reduce completion time further for this workload by giving Dask additional workers?

Solution:

<table>
<thead>
<tr>
<th>Task</th>
<th>Arrived</th>
<th>Start</th>
<th>Completion</th>
<th>Response</th>
<th>Turnaround</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
</tr>
<tr>
<td>T2</td>
<td>0</td>
<td>0</td>
<td>15</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>T3</td>
<td>10</td>
<td>10</td>
<td>15</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>T4</td>
<td>10</td>
<td>15</td>
<td>30</td>
<td>5</td>
<td>20</td>
</tr>
<tr>
<td>T5</td>
<td>20</td>
<td>20</td>
<td>25</td>
<td>0</td>
<td>5</td>
</tr>
</tbody>
</table>

Total turnaround time = 20 + 15 + 5 + 20 + 5 = 65
Average turnaround time \( \frac{65}{5} = 13 \)
(b) 3 worker case = We can see that on 3 workers the tasks complete at 30
1 worker case = 20 + 15 + 5 + 15 + 5 = 60

\[
\text{Speedup} = \frac{\text{Completion time on 1 worker}}{\text{Completion time on 3 workers}} = \frac{60}{30} = 2
\]

(c) Yes. As seen in part a, T4 has a response time of 5. If we add an additional worker
that delay in response time would get cleared thus leaving us with a completion
time of 25 (T4 arrives at 10, starts and finishes in 15 along with T5)

(d) There was an interesting subsection in class about lowering runtime with 3 workers.
I won’t talk about it here since it was a lot. Watch the recording