Q1. [16 x 3pts = 48pts] For each of the following questions, select the correct option; only one is correct.

1. Which of the following model selection approaches/systems can not ensure logical equivalence to sequential SGD for the models trained?
   
   (A) Horovod  
   (B) Asynchronous Parameter Server  
   (C) Model hopper parallelism  
   (D) Task parallelism  
   
   ANSWER: (B)

2. Which of the following consistency models in GraphLab typically offer(s) the maximum amount of parallelism and throughput for large-scale graph analytics?
   
   (A) Full consistency  
   (B) Edge consistency  
   (C) Vertex consistency  
   (D) Eventual consistency  
   
   ANSWER: (C)

3. In a 2-D tradeoff plot to compare model selection approaches with final best accuracy on the x-axis and completion time on the y-axis, toward which corner is the Pareto frontier expected to lie? Note that “left” and “right” here refer to your (the viewer’s) left and right.
   
   (A) Top left  
   (B) Top right  
   (C) Bottom left  
   (D) Bottom right  
   
   ANSWER: (D)

4. What prior ML systems approach does AWS SageMaker adopt to get the independent workers that train on streaming data to reconcile the model with each other?
5. Which of the following ML activities is not part of the “model selection triple”?  
(A) Feature engineering  
(B) Algorithm selection  
(C) Data cleaning  
(D) Hyper-parameter tuning  
**ANSWER: (C)**

6. Which of the following does not have a sequential data access pattern?  
(A) Decision tree  
(B) K-means clustering  
(C) Logistic regression with BGD  
(D) Naive Bayes  
**ANSWER: (A)**

7. What is the term used in the literature for pushing ML computations through joins to avoid denormalization when learning over multi-table data?  
(A) Normalized learning  
(B) Factorized learning  
(C) Push-down learning  
(D) Refactored learning  
**ANSWER: (B)**

8. Which function in the parallel UDA abstraction tends to become a major scalability bottleneck at very large cluster scales (1000s of workers)?  
(A) Initialize  
(B) Transition  
(C) Merge  
(D) Finalize  
**ANSWER: (C)**
9. Which of the following is a popular model exchange format for deep nets?

(A) ODLX  
(B) ONNX  
(C) OMMX  
(D) OMDX

**ANSWER: (B)**

10. Which component of test error is *not* affected by ensembling of decision trees?

(A) Bias  
(B) Variance  
(C) Bayes Noise  
(D) None of the other options

**ANSWER: (C)**

11. Which DL model family will benefit the most from more support for “model parallelism” in DL systems?

(A) CNNs  
(B) RNNs  
(C) MLPs  
(D) Transformers

**ANSWER: (D)**

12. In what way does CPU cache-aware staging of statistics during GBDT computations help reduce runtimes in XGBoost?

(A) Reduces DRAM stalls  
(B) Reduces time complexity  
(C) Reduces random reads to disk  
(D) Raises instruction-level parallelism

**ANSWER: (A)**

13. Which of the following ML systems is the most integrated with the PyData stack?

(A) MADlib  
(B) Mahout  
(C) Spark ML  
(D) GraphLab
14. Which of these layers in a memory hierarchy has the highest capacity to cost ratio?
   (A) CPU caches       (B) GPU memory
   (C) DRAM             (D) Flash SSDs
   **ANSWER: (D)**

15. What is the name of the new programming paradigm ushered in by DL systems?
   (A) Imperative programming       (B) Differentiable programming
   (C) Integrative programming      (D) Functional programming
   **ANSWER: (B)**

16. Which of the following optimizations in TVM does not focus specifically on mitigating the performance impact of memory stalls?
   (A) Operator fusion              (B) Nested parallelism
   (C) Tensorization of operators   (D) Pipelining of operators
   **ANSWER: (C)**

Q2. [2 x 3pts] Early Stopping in Hyperband.

Suppose you run Hyperband for hyperparameter tuning of an ML classifier with the following knobs: $R = 81; \eta = 3$. It yields the following brackets in the same table format from the paper as explained in class:

<table>
<thead>
<tr>
<th>$i$</th>
<th>$s = 4$</th>
<th>$s = 3$</th>
<th>$s = 2$</th>
<th>$s = 1$</th>
<th>$s = 0$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$n_i$</td>
<td>$r_i$</td>
<td>$n_i$</td>
<td>$r_i$</td>
<td>$n_i$</td>
</tr>
<tr>
<td>0</td>
<td>81</td>
<td>1</td>
<td>27</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>1</td>
<td>27</td>
<td>3</td>
<td>9</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>9</td>
<td>3</td>
<td>27</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>27</td>
<td>1</td>
<td>81</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>81</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. How many configs ran for at least 10 epochs?

**ANSWER:** \(3 + 3 + 3 + 6 + 5 = 20\).

2. How many configs got killed in all?

**ANSWER:** \((81 + 27 + 9 + 6) - (1 + 1 + 1 + 2) = 118\).

**Q3. [2 x 3pts] Runtimes of SGD.**

Suppose you use SGD model averaging to train a GLM on 4-worker cluster. Shuffling the datasets takes 10min. An epoch of distributed SGD takes 5min. To accelerate learning you decide to shuffle the dataset only once every 2 epochs.

1. What is the total runtime of training the model for 8 epochs?

**ANSWER:** \(10\text{min} * 4 + 5\text{min} * 8 = 80\text{min}\).

2. You double the number of workers and also decide to train for more epochs. The runtime of shuffling is now only 6min; an SGD epoch, only 3min. How many additional full epochs of training can you afford to run now in the same time as in the last question?

**ANSWER:** Say \(x\) extra epochs. Then we need max \(x\) such that \(6\text{min} * \text{ceil}((8 + x)/2) + 3\text{min} * (8 + x) \leq 80\text{min}\). Works out to \(x = 4\) extra epochs.

**Q4. [2 x 3pts] Communication Costs of Model Selection Systems.**

Suppose you run a DL model selection workload on a 10-worker cluster with 20 training configurations, each trained for 25 epochs. All model sizes are roughly 100 MB each.

The dataset size is 500 GB. It is uniformly randomly sharded on the cluster. The number of data examples is 100 million. The mini-batch size for SGD is fixed to 100.

1. What is the communication cost (in bytes) of Horovod? Round to the nearest multiple of 10 PB.

**ANSWER:** \(2 * 25 * 0.1\ GB * 9 * 20 * 100\ mill / (100 * 10) = 90\ mill\ GB = 90\ PB\).
2. What is the communication cost (in bytes) of MOP? Round to the nearest multiple of 10 GB.

**Answer:**

\[ 25 \times 0.1 \text{ GB} \times 10 \times 20 + 0.1 \text{ GB} \times 20 = 502 \text{ GB} \approx 500 \text{ GB}. \]

**Q5. [9pts]** You are given a large tabular dataset \((Y, X)\) on HDFS with feature vector \(X\) having only a few dozen features, all numeric. You decide to *statistically normalize* the data before training an SVM. This involves subtracting the mean and dividing by the standard deviation for each feature, with the statistics computed over the whole data.

Write succinct pseudocode for (or explain precisely) a single MapReduce job to compute all the means and standard deviations. It should be scalable along the number of examples/rows. Note that you only need to obtain the statistics for this question, not update the features. Write only your final answer in the sections on this page and next.

(Hint: Standard deviation of \(X_i\) can be expressed as \(\sigma(X_i) = (E[X_i^2] - E[X_i]^2)^{\frac{1}{2}}\)

**Answer:** Row-wide sharding; makes it scalable to any number of examples/rows.

Let the number of features in \(X\) be \(d\). The statistics needed for mean and stdev can be collectively decomposed into partial counts, partial sums, and partial sums of squares. The latter two are collected per feature but we can batch all of them onto a single pass over the data. Since they are simple aggregates, they can be easily divided into independent shard computations.

**Map function:**
Iterate through the examples in a shard to compute partial count, partial sum per feature, and partial sum of squares per feature.
Emit them as a single vector of length \(1 + 2d\) with no key.

**Reduce function:**
Iterator has the above vector from all Mappers. Do a vector sum to get the global example count, global sum for each feature, and global sum of squares for each feature.
Divide each feature’s sum by count to get each \(E[X_i]\), that feature’s mean.
Divide each feature’s sum of squares by count to get each \(E[X_i^2]\); plug that and mean into above formula to get that feature’s stdev.

**Extra Credit Question. [5pts]** You are given a matrix \(A\) represented as a relation with one tuple per cell in the following schema: \(A(\text{row}, \text{column}, \text{value})\). Write an SQL query (SELECT ...) to compute the Gramian matrix \(A^T A\). Write each clause in a new line.
**ANSWER:** It is basically a self-join with a group by aggregate.

```sql
SELECT A1.column, A2.column, SUM (A1.row * A2.row) 
FROM A AS A1, A AS A2 
WHERE A1.row = A2.row 
GROUP BY A1.column, A2.column;
```