Congratulations! You passed!
TO PASS 80% or higher

Keep Learning

GRADE
100%

Module 2 Quiz

LATEST SUBMISSION GRADE
100%

1. What are the different units of parallelism? (Select all that apply.) — 1 / 1 point

- [x] Executor

Correct
An executor is one worker node in a cluster.

- [x] Partition

Correct
A partition is a subset of data.

- [x] Task

Correct
A job can be divided into many tasks.

- [x] Core

Correct
A processor has many cores.

2. What is a partition? — 1 / 1 point

- ( ) The result of data filtered by a WHERE clause
- (•) A portion of a large distributed set of data
- ( ) A division of computation that executes a query
- ( ) A synonym with "task"

Correct
Data distributed across the cluster is divided into different partitions.

3. What is the difference between in-memory computing and other technologies? (Select all that apply.) — 1 / 1 point

- [x] In-memory operates from RAM while other technologies operate from disk

Correct
In-memory operation works using RAM.

- [x] In-memory operations were not realistic in older technologies when memory was more expensive

Correct
The price of memory has come down drastically enabling Spark to rely on in-memory calculations.

- [x] Computation not done in-memory (such as Hadoop) reads and writes from disk in between each step

Correct
Hadoop (the precursor to Spark) was much slower because it had to read from and write to disk between every step.

- [ ] In-memory computing is slower than other types of computing

4. Why is caching important? — 1 / 1 point

- ( ) It always stores data in-memory to improve performance

- ○ It reformats data already stored in RAM for faster access
- ◉ It stores data on the cluster to improve query performance
- ○ It improves queries against data read one or more times

✓ **Correct**
By storing data we know we'll see again, caching improves query performance.

5. Which of the following is a wide transformation? (Select all that apply.) **1 / 1 point**

- ☐ WHERE
- ☐ SELECT
- ☑ GROUP BY

✓ **Correct**
A GROUP BY transfers data across the network and is therefore a wide transformation.

- ☑ ORDER BY

✓ **Correct**
An ORDER BY transfers data across the network and is therefore a wide transformation.

6. Broadcast joins... **1 / 1 point**

- ○ Shuffle both of the tables, minimizing computational resources
- ○ Shuffle both of the tables, minimizing data transfer by transferring data in parallel
- ○ Transfer the smaller of two tables to the larger, increasing data transfer requirements
- ◉ Transfer the smaller of two tables to the larger, minimizing data transfer

✓ **Correct**

7. When is it appropriate to use a shuffle join? **1 / 1 point**

- ◉ When both tables are moderately sized or large
- ○ When the smaller table is significantly smaller than the larger table
- ○ Never. Broadcast joins always out-perform shuffle joins.
- ○ When both tables are very small

✓ **Correct**
Shuffle joins are more efficient when both tables are of similar, larger sizes.

8. Which of the following are bottlenecks you can detect with the Spark UI? (Select all that apply.) **1 / 1 point**

- ☑ Data Skew

✓ **Correct**
Data skew is when partitions are not of similar sizes and can be detected by the Spark UI.

- ☑ Shuffle reads

✓ **Correct**
The Spark UI can show shuffles triggered by Spark actions.

- ☑ Shuffle writes

✓ **Correct**
The Spark UI can show shuffles triggered by Spark actions.

- ☐ Incompatible data formats

9. What is a stage boundary? **1 / 1 point**

- ○ A narrow transformation

- Any transition between Spark tasks
- When all of the slots or available units of processing have to sync with one another
- An action caused by a SQL query is predicate

Correct
A stage boundary is when all Spark tasks must come together to exchange a result.

10. What happens when Spark code is executed in local mode? **1 / 1 point**

- The executor and driver are on the same machine
- The code is executed in the cloud
- A cluster of virtual machines is used rather than physical machines
- The code is executed against a local cluster

Correct
Local mode refers to when the executor and driver are the same machine, such as when prototyping Spark code on your laptop.