Processing Unstructured Data and Dask Bag

Course Summary

Description

This class module focuses on Dask Bag, a functional-programming pattern for distributed computation over unstructured or heterogeneous data.

Dask Bag is useful for initial processing of unstructured text, large collections of heterogeneous business records which require special processing, images or diagrams, etc. The class focuses on functional style, the Bag API, and best practices.

Objectives

At the end of this course, students will understand:

- How Dask Bag applies your Python code to large data collections
- Transforming, filtering, combining, aggregating, and matching objects
- Addressing performance concerns

Topics

- Introduction
- Core Bag APIs and Operations
- Best Practices

Audience

This course is intended for engineers or data scientists who typically work with large data collections.

Prerequisites

Students should have experience in Python at a basic to intermediate level. Additionally, some knowledge of functional programming is helpful but not required.

Duration

One day

Course Outline

Processing Unstructured Data and Dask Bag

Course Outline

I. Introduction

- A. Python functional constructs in the standard library
- B. Why use a functional model for “big data”?
- C. Dask Bag vs. local Python collections

II. Core Bag APIs and Operations

- A. Ingesting data and creating Bags
- B. Understanding partitions
- C. Transform and project data with map
- D. Understanding execution: Compute, Persist, and Visualize
- E. Filter data
- F. Builtin aggregations: math/stats, conditionals, counting, and sorting
- G. Aggregate data with group and fold
- H. Combine data with zip and join
- I. Writing or retrieving output

III. Best Practices

- A. General query improvement patterns
- B. Minimizing expensive data transfer
- C. Integration with from_delayed, to_delayed, to_dataframe
- D. Partition sizing