**Advantages**

* **Ease of Profiling: Automatically measures and logs the execution time of each pipeline stage.**
* **Modular: The decorator can be applied to any function without modifying its code.**
* **Scalable: Works for any pipeline function, from file reading to distributed data processing.**

**Input Data (big\_data\_input.json)**

{"id": 1, "value": 50}

{"id": 2, "value": 200}

{"id": 3, "value": 150}

{"id": 4, "value": 80}

measure the execution time of functions.

This is useful for profiling and identifying bottlenecks in your pipeline.

import time

import json

# Decorator to calculate execution time

def measure\_time(func):

def wrapper(\*args, \*\*kwargs):

start\_time = time.time()

result = func(\*args, \*\*kwargs)

end\_time = time.time()

print(f"Function '{func.\_\_name\_\_}' executed in {end\_time - start\_time:.2f} seconds")

return result

return wrapper

# Simulated Big Data Pipeline Components

# Step 1: Read data (e.g., from a large file)

@measure\_time

def read\_data(file\_path):

with open(file\_path, "r") as f:

data = [json.loads(line) for line in f]

return data

# Step 2: Filter data

@measure\_time

def filter\_data(data):

return [record for record in data if record.get("value") > 100]

# Step 3: Transform data (e.g., enrich records)

@measure\_time

def transform\_data(data):

for record in data:

record["processed\_value"] = record["value"] \* 2

record["status"] = "processed"

return data

# Step 4: Save transformed data to output

@measure\_time

def save\_data(data, output\_path):

with open(output\_path, "w") as f:

for record in data:

f.write(json.dumps(record) + "\n")

# Main Pipeline Execution

if \_\_name\_\_ == "\_\_main\_\_":

input\_file = "big\_data\_input.json"

output\_file = "big\_data\_output.json"

# Simulate data pipeline steps

raw\_data = read\_data(input\_file)

filtered\_data = filter\_data(raw\_data)

transformed\_data = transform\_data(filtered\_data)

save\_data(transformed\_data, output\_file)

print("Pipeline execution completed!")