**mapreduce**

**Project Mapreduce bao gồm GUI**

**- app.py**

**from flask import Flask, render\_template, request**

**import subprocess**

**app = Flask(\_\_name\_\_)**

**# Các file JSON có thể chọn**

**json\_files = [**

**"input/books\_clean.json",**

**"input/tiki\_book\_data\_clean.json"**

**]**

**jobs = {**

**"avg\_rating": {**

**"mapper": "mapper/avg\_rating\_mapper.py",**

**"reducer": "reducer/avg\_rating\_reducer.py",**

**"input\_type": "single"**

**},**

**"avg\_price": {**

**"mapper": "mapper/avg\_price\_mapper.py",**

**"reducer": "reducer/avg\_price\_reducer.py",**

**"input\_type": "single"**

**},**

**"max\_price": {**

**"mapper": "mapper/max\_price\_mapper.py",**

**"reducer": "reducer/max\_price\_reducer.py",**

**"input\_type": "single"**

**},**

**"min\_price": {**

**"mapper": "mapper/min\_price\_mapper.py",**

**"reducer": "reducer/min\_price\_reducer.py",**

**"input\_type": "single"**

**},**

**"count\_rating": {**

**"mapper": "mapper/count\_rating\_mapper.py",**

**"reducer": "reducer/count\_rating\_reducer.py",**

**"input\_type": "single"**

**},**

**"compare\_avg\_price": {**

**"mapper": "mapper/compare\_avg\_price\_mapper.py",**

**"reducer": "reducer/compare\_avg\_price\_reducer.py",**

**"input\_type": "multiple"**

**}**

**}**

**def run\_mapreduce(mapper, reducer, input\_files):**

**try:**

**if isinstance(input\_files, list):**

**combined\_input = ""**

**for file in input\_files:**

**with open(file, "r", encoding="utf-8") as f:**

**combined\_input += f.read()**

**mapper\_proc = subprocess.Popen(**

**["python3", mapper],**

**stdin=subprocess.PIPE,**

**stdout=subprocess.PIPE,**

**stderr=subprocess.PIPE,**

**text=True**

**)**

**mapper\_output, mapper\_err = mapper\_proc.communicate(input=combined\_input)**

**if mapper\_proc.returncode != 0:**

**return f"Mapper error:\n{mapper\_err}"**

**reducer\_proc = subprocess.Popen(**

**["python3", reducer],**

**stdin=subprocess.PIPE,**

**stdout=subprocess.PIPE,**

**stderr=subprocess.PIPE,**

**text=True**

**)**

**reducer\_output, reducer\_err = reducer\_proc.communicate(input=mapper\_output)**

**if reducer\_proc.returncode != 0:**

**return f"Reducer error:\n{reducer\_err}"**

**return reducer\_output**

**else:**

**with open(input\_files, "r", encoding="utf-8") as f:**

**file\_content = f.read()**

**mapper\_proc = subprocess.Popen(**

**["python3", mapper],**

**stdin=subprocess.PIPE,**

**stdout=subprocess.PIPE,**

**stderr=subprocess.PIPE,**

**text=True**

**)**

**mapper\_output, mapper\_err = mapper\_proc.communicate(input=file\_content)**

**if mapper\_proc.returncode != 0:**

**return f"Mapper error:\n{mapper\_err}"**

**reducer\_proc = subprocess.Popen(**

**["python3", reducer],**

**stdin=subprocess.PIPE,**

**stdout=subprocess.PIPE,**

**stderr=subprocess.PIPE,**

**text=True**

**)**

**reducer\_output, reducer\_err = reducer\_proc.communicate(input=mapper\_output)**

**if reducer\_proc.returncode != 0:**

**return f"Reducer error:\n{reducer\_err}"**

**return reducer\_output**

**except Exception as e:**

**return f"Exception: {str(e)}"**

**@app.route("/", methods=["GET", "POST"])**

**def index():**

**result = None**

**selected\_job = None**

**selected\_file = None**

**disable\_file\_select = False**

**if request.method == "POST":**

**selected\_job = request.form.get("job")**

**selected\_file = request.form.get("input\_file")**

**if selected\_job in jobs:**

**job = jobs[selected\_job]**

**if job["input\_type"] == "single":**

**if selected\_file not in json\_files:**

**result = "Vui lòng chọn file hợp lệ."**

**else:**

**result = run\_mapreduce(job["mapper"], job["reducer"], selected\_file)**

**else:**

**result = run\_mapreduce(job["mapper"], job["reducer"], json\_files)**

**disable\_file\_select = (job["input\_type"] == "multiple")**

**else:**

**result = "Chức năng không hợp lệ."**

**return render\_template(**

**"index.html",**

**jobs=jobs.keys(),**

**files=json\_files,**

**result=result,**

**selected\_job=selected\_job,**

**selected\_file=selected\_file,**

**disable\_file\_select=disable\_file\_select**

**)**

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

**app.run(debug=True)**

* **input**

**+ books\_clean.json**

**+ tiki\_book\_data\_clean.json**

* **mapper**

**+ avg\_price\_mapper.py**

**import sys**

**import json**

**for line in sys.stdin:**

**try:**

**record = json.loads(line)**

**price = float(record.get('price', 0))**

**print(f"price\t{price}\t1")**

**except:**

**continue**

**+ avg\_rating\_mapper.py**

**import sys**

**import json**

**for line in sys.stdin:**

**try:**

**record = json.loads(line)**

**rating = float(record.get("rating", 0))**

**print(f"avg\_rating\t{rating}")**

**except:**

**continue**

**+ compare\_avg\_price\_mapper.py**

**import sys**

**import json**

**import os**

**# map input file name (Hadoop env variable usually)**

**input\_file = os.environ.get('mapreduce\_map\_input\_file', '')**

**for line in sys.stdin:**

**try:**

**record = json.loads(line)**

**price = float(record.get('price', 0))**

**# emit with file tag**

**if 'books\_clean.json' in input\_file:**

**print(f"books\t{price}\t1")**

**elif 'tiki\_book\_data\_clean.json' in input\_file:**

**print(f"tiki\t{price}\t1")**

**except:**

**continue**

**+ count\_rating\_mapper.py**

**import sys**

**import json**

**for line in sys.stdin:**

**try:**

**record = json.loads(line)**

**rating = record.get('rating', None)**

**if rating is not None:**

**print(f"{rating}\t1")**

**except:**

**continue**

**+ max\_price\_mapper.py**

**import sys**

**import json**

**for line in sys.stdin:**

**try:**

**record = json.loads(line)**

**price = float(record.get('price', 0))**

**title = record.get('title', '')**

**print(f"max\_price\t{price}\t{title}")**

**except:**

**continue**

**+ min\_price\_mapper.py**

**import sys**

**import json**

**for line in sys.stdin:**

**try:**

**record = json.loads(line)**

**price = float(record.get('price', 0))**

**title = record.get('title', '')**

**print(f"min\_price\t{price}\t{title}")**

**except:**

**continue**

**reducer**

**+ avg\_price\_reducer.py**

**import sys**

**total\_price = 0.0**

**count = 0**

**for line in sys.stdin:**

**try:**

**key, price, cnt = line.strip().split('\t')**

**price = float(price)**

**cnt = int(cnt)**

**total\_price += price**

**count += cnt**

**except:**

**continue**

**if count > 0:**

**print(f"Average\_Price\t{total\_price/count:.2f}")**

**else:**

**print("Average\_Price\t0")**

**+ avg\_rating\_reducer.py**

**import sys**

**total\_rating = 0**

**count = 0**

**for line in sys.stdin:**

**key, value = line.strip().split('\t')**

**total\_rating += float(value)**

**count += 1**

**if count > 0:**

**print(f"Average Rating: {total\_rating / count:.2f}")**

**else:**

**print("No valid ratings")**

**+ compare\_avg\_price\_reducer.py**

**import sys**

**sum\_books = 0.0**

**count\_books = 0**

**sum\_tiki = 0.0**

**count\_tiki = 0**

**for line in sys.stdin:**

**try:**

**tag, price, cnt = line.strip().split('\t')**

**price = float(price)**

**cnt = int(cnt)**

**if tag == 'books':**

**sum\_books += price**

**count\_books += cnt**

**elif tag == 'tiki':**

**sum\_tiki += price**

**count\_tiki += cnt**

**except:**

**continue**

**avg\_books = sum\_books / count\_books if count\_books > 0 else 0**

**avg\_tiki = sum\_tiki / count\_tiki if count\_tiki > 0 else 0**

**print(f"Books\_Avg\_Price\t{avg\_books:.2f}")**

**print(f"Tiki\_Avg\_Price\t{avg\_tiki:.2f}")**

**+ count\_rating\_reducer.py**

**import sys**

**current\_rating = None**

**count = 0**

**for line in sys.stdin:**

**try:**

**rating, cnt = line.strip().split('\t')**

**cnt = int(cnt)**

**if current\_rating == rating:**

**count += cnt**

**else:**

**if current\_rating is not None:**

**print(f"{current\_rating}\t{count}")**

**current\_rating = rating**

**count = cnt**

**except:**

**continue**

**if current\_rating is not None:**

**print(f"{current\_rating}\t{count}")**

**+ max\_price\_reducer.py**

**import sys**

**max\_price = -1**

**max\_title = ""**

**for line in sys.stdin:**

**try:**

**key, price, title = line.strip().split('\t', 2)**

**price = float(price)**

**if price > max\_price:**

**max\_price = price**

**max\_title = title**

**except:**

**continue**

**print(f"Max\_Price\t{max\_price}\t{max\_title}")**

**+ min\_price\_reducer.py**

**import sys**

**min\_price = float('inf')**

**min\_title = ""**

**for line in sys.stdin:**

**try:**

**key, price, title = line.strip().split('\t', 2)**

**price = float(price)**

**if price < min\_price:**

**min\_price = price**

**min\_title = title**

**except:**

**continue**

**print(f"Min\_Price\t{min\_price}\t{min\_title}")**

* **templates**

**+ index.html**

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<meta charset="UTF-8" />**

**<title>MapReduce Web GUI</title>**

**<style>**

**body { font-family: Arial, sans-serif; margin: 20px; background: #f9f9f9; }**

**select, button, input { padding: 8px 12px; font-size: 16px; }**

**select:disabled, input:disabled {**

**background-color: #ddd;**

**color: #666;**

**}**

**pre { background: #fff; border: 1px solid #ddd; padding: 15px; white-space: pre-wrap; max-height: 400px; overflow-y: auto; }**

**h1 { color: #2c3e50; }**

**form { margin-bottom: 20px; }**

**label { display: inline-block; margin-bottom: 6px; }**

**</style>**

**<script>**

**function toggleFileSelect() {**

**const jobSelect = document.getElementById("job");**

**const fileSelect = document.getElementById("input\_file");**

**const fileInputDisplay = document.getElementById("input\_file\_display");**

**if (jobSelect.value === "compare\_avg\_price") {**

**if (fileSelect) fileSelect.style.display = "none";**

**if (fileInputDisplay) fileInputDisplay.style.display = "block";**

**} else {**

**if (fileSelect) fileSelect.style.display = "inline-block";**

**if (fileInputDisplay) fileInputDisplay.style.display = "none";**

**}**

**}**

**window.onload = function() {**

**toggleFileSelect();**

**document.getElementById("job").addEventListener("change", toggleFileSelect);**

**}**

**</script>**

**</head>**

**<body>**

**<h1>Chạy các chức năng MapReduce</h1>**

**<form method="POST">**

**<label for="job">Chọn chức năng:</label>**

**<select name="job" id="job" required>**

**<option value="">-- Chọn chức năng --</option>**

**{% for job in jobs %}**

**<option value="{{ job }}" {% if job == selected\_job %}selected{% endif %}>{{ job }}</option>**

**{% endfor %}**

**</select>**

**<br /><br />**

**<label for="input\_file">Chọn file đầu vào:</label><br />**

**{% if disable\_file\_select %}**

**<input type="text" id="input\_file\_display" value="{{ selected\_file }}" disabled style="width: 100%;" />**

**<select id="input\_file" name="input\_file" style="display: none;">**

**<option value=""></option>**

**</select>**

**{% else %}**

**<input type="text" id="input\_file\_display" disabled style="display:none;" />**

**<select name="input\_file" id="input\_file" required>**

**<option value="">-- Chọn file --</option>**

**{% for f in files %}**

**<option value="{{ f }}" {% if f == selected\_file %}selected{% endif %}>{{ f }}</option>**

**{% endfor %}**

**</select>**

**{% endif %}**

**<br /><br />**

**<button type="submit">Chạy</button>**

**</form>**

**{% if result is not none %}**

**<h2>Kết quả:</h2>**

**<pre>{{ result }}</pre>**

**{% endif %}**

**</body>**

**</html>**