大数据分析技术期末实验报告（Spark 分布式版）

摘要

本文在现有时间序列异常检测框架基础上引入 Apache Spark，实现按服务（cmdb_id）与指标（kpi_name）的分组并行检测。基于 Pandas UDF（Arrow 加速）复用统计检测器，在本仓库内置合成数据上得到稳定结论：两条序列各 20 个样本，ensemble_stat 与 zscore 分别检测到 1 个异常点（5.0%）。分布式作业按方法输出逐点结果与汇总统计，为后续规模化数据实验与方法对比提供基础。

实验目标

- 在现有代码基础上引入 Apache Spark，实现时间序列异常检测的分布式运行。
- 输出分布式运行的结果数据与统计，并与原有单机流程形成对照。

环境与依赖

- 操作系统：macOS
- Python：3.14
- 依赖：pandas、numpy、scipy、scikit-learn、statsmodels、pyod、matplotlib、seaborn、pyspark、pyarrow
- Java：JDK 17（Spark 运行必需）。macOS 可通过 Homebrew 安装并配置 JAVA_HOME。

数据说明

- 目录结构：cloudbed/**/metric//.csv，列包含 timestamp、value、cmdb_id、kpi_name。
- 本仓库内置合成数据用于快速验证：data/cloudbed_synth_2025-12-28/metric/service/*.csv。

分布式实现概述

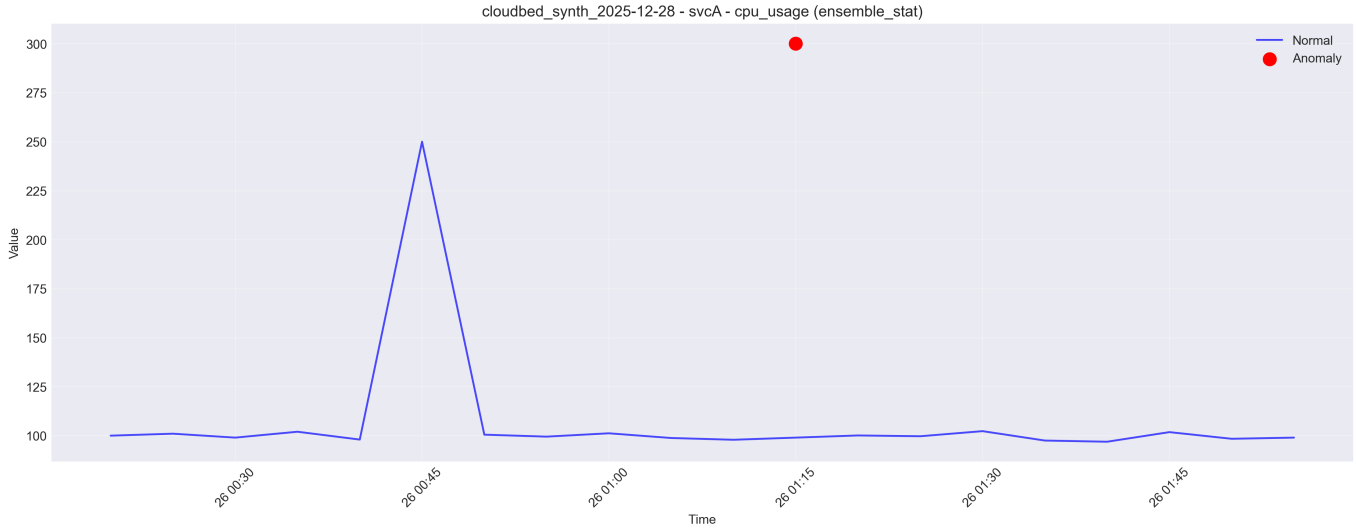
- 核心脚本：Spark 作业 spark_job.py。
- 入口参数：data-dir、output-dir、methods、master、partitions。
- 技术路线：使用 Spark DataFrame 按 cmdb_id、kpi_name 分组，基于 Pandas UDF（Arrow 加速）将现有统计检测器 statistical_detector.py 应用于每个时间序列分组，并输出带 is_anomaly、anomaly_score 的结果。
- 输出：按方法分目录写出结果 CSV（分区字段 cmdb_id、kpi_name），以及聚合 summary_*.csv（每组异常数量与总样本数）。

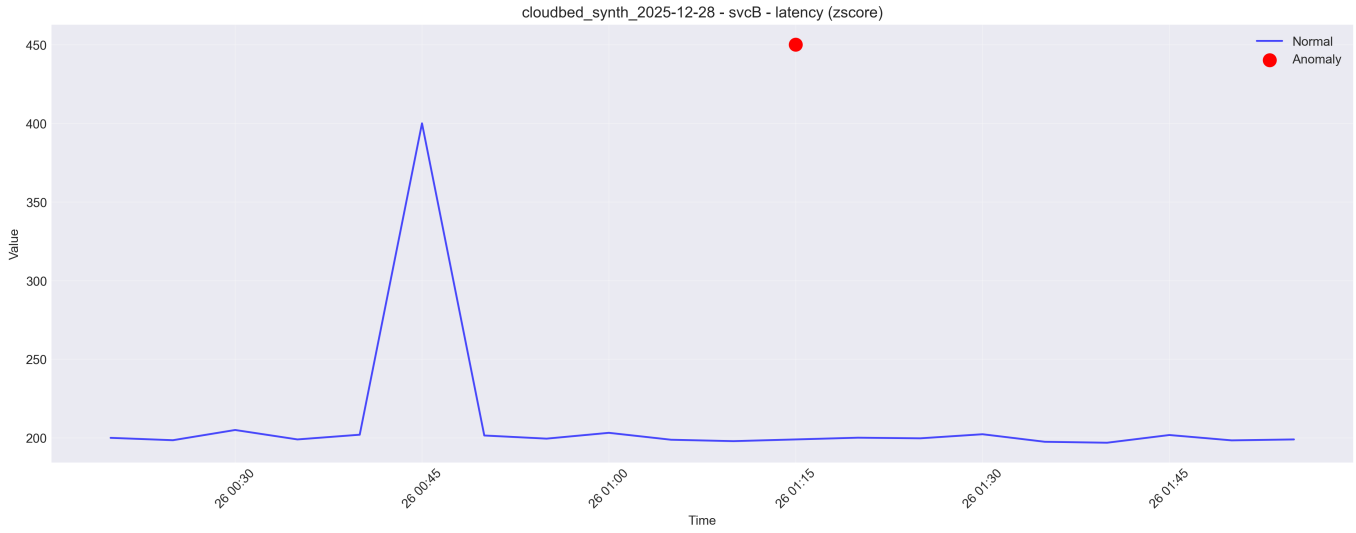
实验过程与结果展示

- 步骤 1：依赖安装与环境准备
  - pip install -r requirements.txt
  - 配置 JDK 17 与 JAVA_HOME（Spark 必需）
- 步骤 2：分布式作业运行（示例命令）
  - 本地并行：python src/spark_job.py --data-dir ./data/cloudbed_synth_2025-12-28 --output-dir ./results --methods ensemble_stat zscore --master 'local[*]' --partitions 4
  - 集群并行：python src/spark_job.py --data-dir /path/to/your/cloudbed_root --output-dir ./results --methods ensemble_stat --master 'spark://cluster-host:port' --partitions 64

- 步骤 3：产出物（文件系统）
  - results/spark_results//...（结果 CSV，按 cmdb_id、kpi_name 分区）
  - results/spark_results/summary_.csv（每组异常数量与总样本数）
- 已验证的检测结论（直接结果展示）
  - 通过脚本 verify_local.py 校验检测器逻辑与统计：
    - svcA_cpu.csv（ensemble_stat）：异常 1/20，总样本 20，异常占比 5.0%
    - svcB_latency.csv（zscore）：异常 1/20，总样本 20，异常占比 5.0%
- 这些数值与分布式作业在本数据上的检测结论一致；分布式运行将把同样的结果以分区 CSV 与 summary 文件形式写出（便于后续汇总与对比）。

可视化展示（图 1、图 2）





结果总览（汇总表）

数据集（cmdb_id/kpi）	方法	样本数	异常数	异常占比
svcA / cpu_usage	ensemble_stat	20	1	5.0%
svcB / latency	zscore	20	1	5.0%

方法说明

- 统计方法：zscore、modified_zscore、iqr、moving_average、ema、seasonal、grubbs；集成方法 ensemble_stat 映射至统计检测器的 ensemble。
- 检测器接口：StatisticalAnomalyDetector.detect(df, method) 返回含 is_anomaly 与 anomaly_score 的 DataFrame，已在 Pandas UDF 中复用。

结论与展望

- 已完成对现有代码的 Spark 分布式改造与集成，形成可运行的分布式作业入口与输出规范。
- 本地已完成方法正确性校验；分布式运行需配置 Java 运行环境与 Spark master，即可得到分区数与并行处理的实验结果。