Al-Mustaqbal University
College of Sciences
Intelligent Medical System Department

جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

كلية العلوم

قسم الانظمة الطبية الذكية

Lecture (9 & 10): Pig in the Hadoop Ecosystem

Subject: Big Data Analysis in Healthcare
Level: Fourth
Lecturer: Asst. Lecturer Qusai AL-Durrah
Duration: Two hours

1. Introduction

Apache Pig is a high-level data processing framework developed at Yahoo! to simplify the analysis of large-scale datasets in Hadoop. Unlike Java-based MapReduce—which can be complex, lengthy, and difficult for non-programmers—Pig provides a powerful scripting language called **Pig Latin**, enabling users to perform data transformations using short, expressive commands.

Pig is especially effective for **data cleaning, preprocessing, transformation, and feature preparation**, making it an essential tool in large-scale healthcare analytics where raw clinical, operational, and sensor data often require extensive preparation before analysis.

2. Learning Objectives

By the end of this lecture, students will be able to:

1. **Describe**: Apache Pig and its purpose in the Hadoop ecosystem.
2. **Explain**: why Pig was created and how it simplifies MapReduce programming.
3. **Understand:** the internal architecture of Apache Pig.
4. **Write and interpret:** Pig Latin commands for loading, filtering, grouping, and transforming data.
5. **Execute**: Pig Latin operations in a healthcare data context.
6. **Compare** Pig with Hive and identify when each tool is most effective.

3. What Is Apache Pig?

Apache Pig is a platform for analyzing large datasets stored in Hadoop. It provides:

- **A high-level language (Pig Latin)** for expressing data flows

- **An execution engine** that converts Pig Latin scripts into MapReduce (or Tez) jobs
- **A flexible model** capable of handling structured, semi-structured, and unstructured data

Pig was designed to reduce development complexity, increase productivity, and provide an intuitive scripting environment for large-scale data manipulation.

4. Why Apache Pig?

4.1 Difficulty of Writing Java MapReduce

Before Pig, analysts and developers had to write lengthy Java code to perform even simple data operations. Pig reduces this burden dramatically.

4.2 Pig Latin: A High-Level Interface

Pig Latin allows developers to write expressive scripts that describe data flows without specifying low-level execution details.

4.3 Productivity Benefits

- Up to **16× fewer lines of code** compared to Java MapReduce
- Easy to learn for users familiar with SQL or scripting languages
- Supports **multi-query optimization** for efficient execution

4.4 Ideal for Healthcare Data Processing

Pig is excellent for preparing messy, raw, or semi-structured datasets such as:

- EHR exports
- Laboratory output logs
- Device-generated data streams
- Medical imaging metadata

- Clinical text files

5. Features of Apache Pig

5.1 In-Built Operators

Pig Latin includes operators for:

- Filtering (FILTER)
- Grouping (GROUP)
- Sorting (ORDER)
- Joins (JOIN)
- Projection (FOREACH … GENERATE)

5.2 Ease of Programming

Pig Latin is concise, readable, and similar to SQL in spirit.

5.3 Automatic Optimization

Pig includes an optimizer that determines efficient execution strategies.

5.4 Handles All Data Types

Pig supports structured, semi-structured, and unstructured formats, making it highly suitable for healthcare datasets.

6. Apache Pig Architecture

Apache Pig follows a structured data processing pipeline:

6.1 Parser

- Validates syntax
- Creates a **Logical Plan** represented as a DAG

6.2 Logical Optimizer

- Performs rule-based optimizations
- Pushes filters early
- Removes redundant expressions

6.3 Compiler

- Converts the optimized logical plan into a **physical execution plan**
- Translates Pig Latin operators into MapReduce or Tez tasks

6.4 Execution Engine

- Sends generated jobs to YARN
- Monitors execution and returns results to the user

Pipeline Summary:
Pig Latin Script → Parser → Optimizer → Compiler → MR/Tez Jobs → Output

7. Pig Latin: The Language of Apache Pig

Pig Latin is the scripting language used to express data processing pipelines in Pig. It is simple, expressive, and designed to make data transformation tasks intuitive.

7.1 Characteristics of Pig Latin

7.1.1 High-Level

Pig Latin describes transformations without requiring knowledge of MapReduce internals.

7.1.2 Procedural

Scripts are written as step-by-step pipelines:

Load → Filter → Transform → Group → Aggregate → Store

7.1.3 Flexible with Data

Works with structured, semi-structured, and unstructured datasets.

7.1.4 Extensible

Supports **User Defined Functions (UDFs)** written in Java, Python, or JavaScript.

7.2 Basic Structure of Pig Latin

Pig Latin assigns each operation to a relation (dataset):

relation_name = OPERATION data_source;

Example:

```
A = LOAD 'visits.txt' USING PigStorage(',')
   AS (id:int, diag:chararray, cost:double);
```

7.3 Core Pig Latin Commands

7.3.1 LOAD

```
visits = LOAD 'visits.txt'
        USING PigStorage(',')
        AS (id:int, diag:chararray, cost:double);
```

7.3.2 DUMP

```
DUMP visits;
```

7.3.3 FILTER

```
high_cost = FILTER visits BY cost > 50;
```

7.3.4 FOREACH … GENERATE

```
only_diag_cost = FOREACH visits GENERATE diag, cost;
```

With transformation:

new_cost = FOREACH visits GENERATE id, diag, cost * 1.1;

7.3.5 GROUP

grouped = GROUP visits BY diag;

7.3.6 Aggregation

avg_cost = FOREACH grouped GENERATE

group AS diagnosis,

AVG(visits.cost) AS avg_cost;

7.3.7 ORDER

sorted = ORDER visits BY cost DESC;

7.3.8 DISTINCT

unique_diags = DISTINCT visits.diag;

7.3.9 JOIN

joined = JOIN visits BY diag, diag_ref BY code;

7.3.10 STORE

STORE avg_cost INTO 'output/avg_cost' USING PigStorage(',');

7.4 Example Pig Latin Script (Healthcare Dataset)

Assume **visits.txt** contains:

101,J10.1,45.5

102,E11.9,70

103,I10,33

104,E11.9,110

101,J10.1,55

Script:

```
visits = LOAD 'visits.txt' USING PigStorage(',')
        AS (id:int, diagnosis:chararray, cost:double);


high_cost = FILTER visits BY cost > 50;


grouped = GROUP visits BY diagnosis;


avg_cost = FOREACH grouped GENERATE
            group AS diagnosis,
            AVG(visits.cost) AS mean_cost;


DUMP avg_cost;
```

This computes the average cost per diagnosis code—a common hospital analytics task.

8. Pig in Healthcare Analytics

Pig is particularly valuable in medical data engineering because healthcare data is often messy and inconsistent. Pig excels at:

- Cleaning laboratory results
- Transforming EHR exports

- Preprocessing imaging metadata
- Filtering abnormal sensor readings
- Constructing feature datasets for machine learning
- Standardizing diagnosis codes prior to Hive storage

Pig typically appears early in the data pipeline before Hive, Spark, or ML systems perform higher-level analysis.

9. Pig vs. Hive

Apache Pig and Apache Hive are both high-level tools built to simplify data processing on Hadoop, but they serve different purposes.

9.1 Purpose Comparison

Hive

- SQL-like (HiveQL)
- Declarative
- Ideal for **analytics**, reporting, dashboards
- Works best with **structured** data

Pig

- Procedural scripting (Pig Latin)
- Ideal for **ETL**, cleaning, transformation
- Excellent with **semi-structured and unstructured** data

9.2 Processing Model

Aspect	Hive	Pig
Primary Use	Analytics & Reporting	ETL & Data Processing
Language Style	SQL-like	Script-based procedural
Ideal For	Structured data	Semi/unstructured data
Users	Analysts / SQL developers	Data engineers
Strength	Summaries & aggregations	Transformations & cleaning
Weakness	Complex transformations	Complex analytics

9.3 Use Case Comparison

Hive is better for:

- Structured datasets
- Reporting and summaries
- Hospital dashboards
- Aggregations on massive fact tables

Pig is better for:

- Raw healthcare data cleaning
- Complex transformations
- Preprocessing datasets for ML

- Parsing logs or semi-structured device output

9.4 Which Is Better?

There is **no universal winner**.
Pig and Hive complement each other.

Use Hive for ANALYTICS
Use Pig for PREPROCESSING

In real healthcare pipelines:

Raw Clinical Data → Pig (cleaning) → Hive (analytics) → BI Dashboards

10. Summary

Apache Pig provides a flexible, expressive way to perform large-scale data transformations in Hadoop.
Pig Latin is simple yet powerful, enabling efficient preprocessing pipelines for complex healthcare datasets.
When combined with Hive, Pig becomes an essential component in end-to-end big data workflows—supporting data preparation, storage, analysis, and advanced medical decision-making.