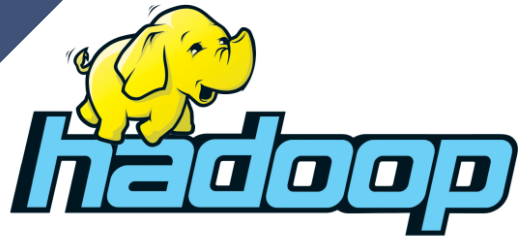
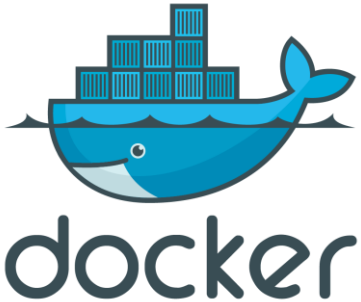
Analisi del dataset MovieLens con Hadoop

Sistemi e architetture per Big Data

Outline



- Scopo del progetto.
- Presentazione e configurazione dell’ambiente.
- Descrizione delle query.



Scopo e obiettivi del progetto

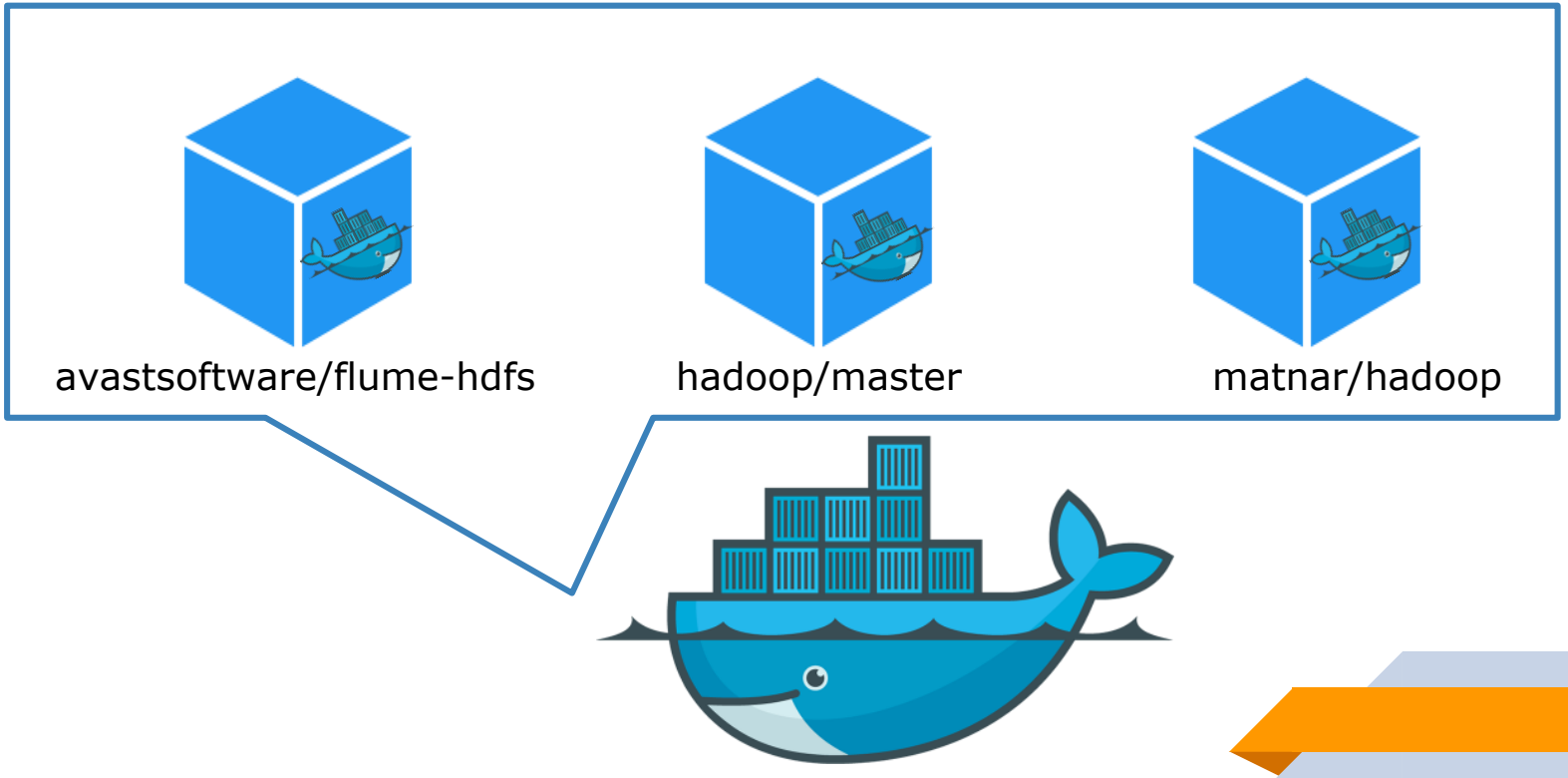
Lo scopo del progetto e di rispondere, utilizzando il framework Apache Hadoop, ad alcune query riguardanti il dataset MovieLens.

MovieLens è il nome del sistema di raccomandazione gratuito realizzato da GroupLens (Università del Minnesota) per effettuare una ricerca nel campo dei sistemi di raccomandazione

Scelta dell’ambiente

Abbiamo deciso di utilizzare **Docker** come tecnologia per standardizzare e semplificare il processo di deployment ed esecuzione dell’intero progetto.

Architettura...



Data ingestion

Rappresentazione di dati:

Nella nostra soluzione non abbiamo trasformato la rappresentazione dei dati rendendo più semplice ed intuitivo il parser dei dati.

Anche se consapevoli che Avro e Parquet siano più efficienti e che abbiano prestazione migliori, ma che vengono utilizzati soprattutto per schema che cambia dinamicamente, pertanto non strettamente necessario al funzionamento.

Data ingestion

Flume:

Flume viene utilizzato per spostare i dati generati in HDFS a una velocità superiore.

Tra le sue caratteristiche abbiamo l’essere: affidabile, fault tolerant, scalabile, gestibile e personalizzabile.

Data ingestion

HBase:

Abbiamo deciso di esportare i dati di output al sistema di storage HBase.

Apache HBase è un datastore per Big Data distribuito dotato di elevata scalabilità che fa parte dell'ecosistema di Apache Hadoop.

Apache HBase è progettato per mantenere elevate le prestazioni anche quando sono allocati centinaia di nodi, per supportare miliardi di righe e milioni di colonne.

Eseguire il progetto

L’intero progetto è stato realizzato per essere eseguito da chiunque e non richiede particolari configurazioni manuali.

Pertanto un cliente interessato all’applicazione può facilmente avviare il tutto e valutare il suo sistema di raccomandazione.

Eseguire il progetto

1. Scaricare la cartella contenente le istruzioni ed i file necessari

1. Installare Docker

1. Scaricare le immagini:
    a. docker pull matnar/hadoop
    b. docker pull avastsoftware/flume-hdfs
    c. docker load master-hbase.tar

Eseguire il progetto

Eseguita questa prima parte bisognerà inserire i file o eventuali aggiornamenti nella cartella flume e successivamente lanciare lo script “create.sh” che provvederà a creare la rete e lanciare i vari container.

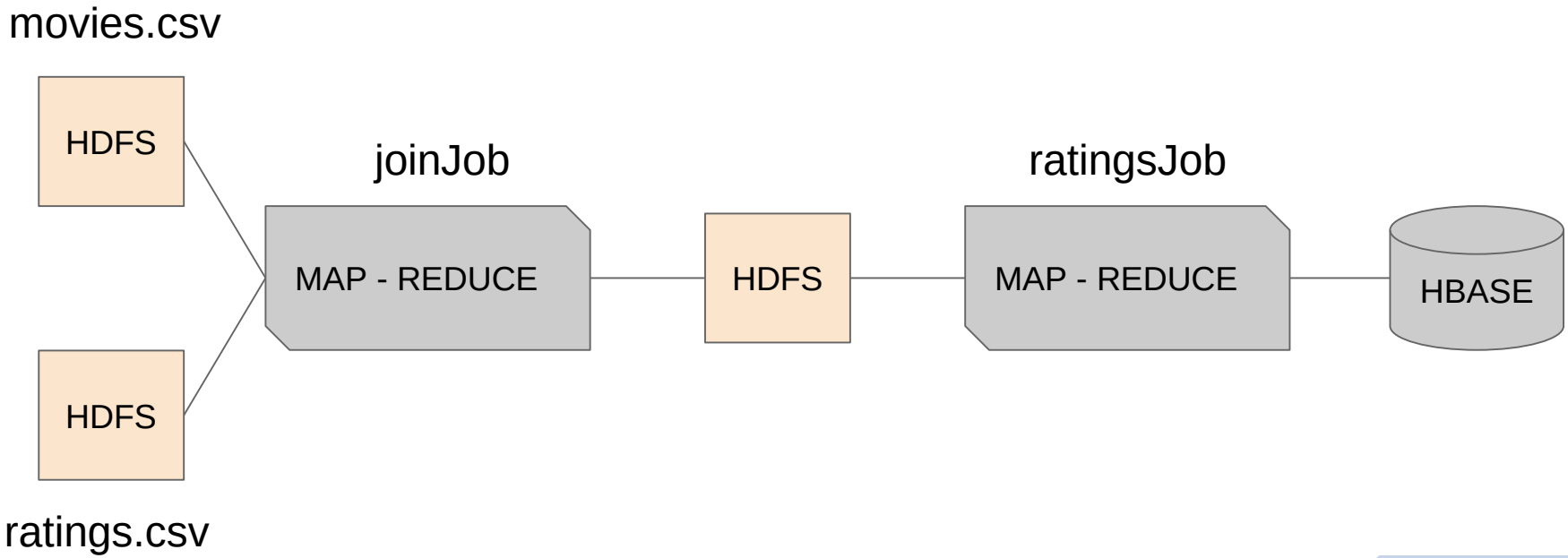
Al termine dello script potremmo eseguire le query e visualizzare i risultati.

Query del progetto

Le query sono:

1. Individuare i film con una valutazione maggiore o uguale a 4.0 e valutati a partire dal 1 Gennaio 2000.
2. Calcolare la valutazione media e la sua deviazione standard per ciascun genere di film.
3. Trovare i 10 film che hanno ottenuto la più alta valutazione nell'ultimo anno del dataset e confrontare la posizione rispetto a quella conseguita l'anno precedente.

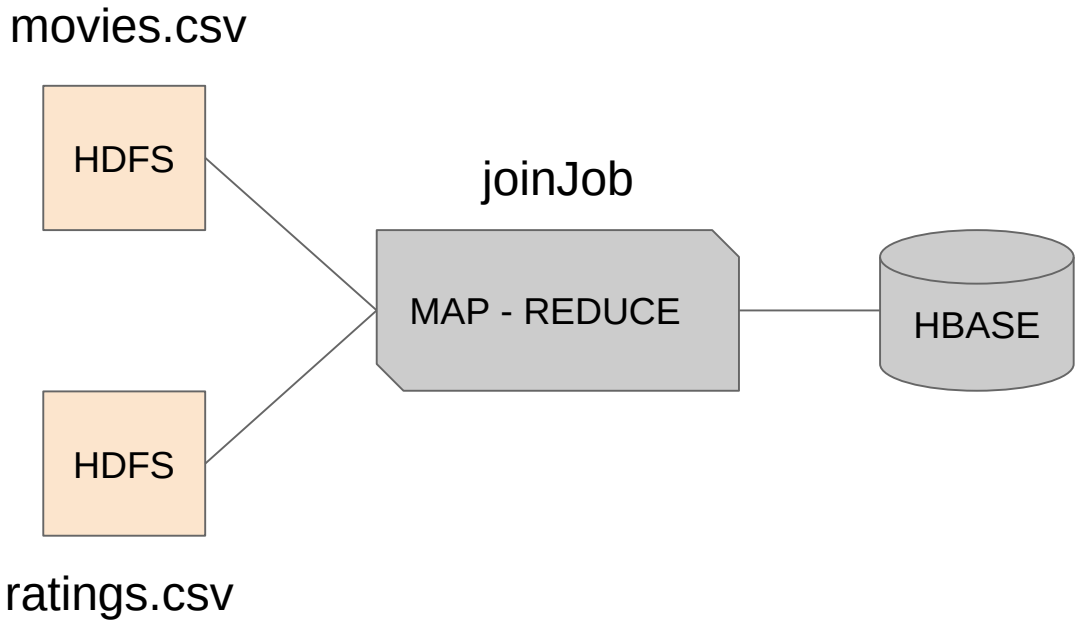
Query Uno: Architettura prima versione



Query Uno: Architettura prima versione

- **Mapper del JoinJob**
  - filtra tutte le valutazioni precedenti al 1 Gennaio 2000
  - Emette le seguenti coppie chiave valore:
    - key = movieId value = “M” + titolo
    - key = movieId value = “R” + rating
- **Reducer del JoinJob**
  - Emette la seguente coppia chiave valore:
    - key = movieId value = rating + titolo
- **Reducer HighRatings** computa la media dei valori relativi ad un film

Query uno: Architettura seconda versione

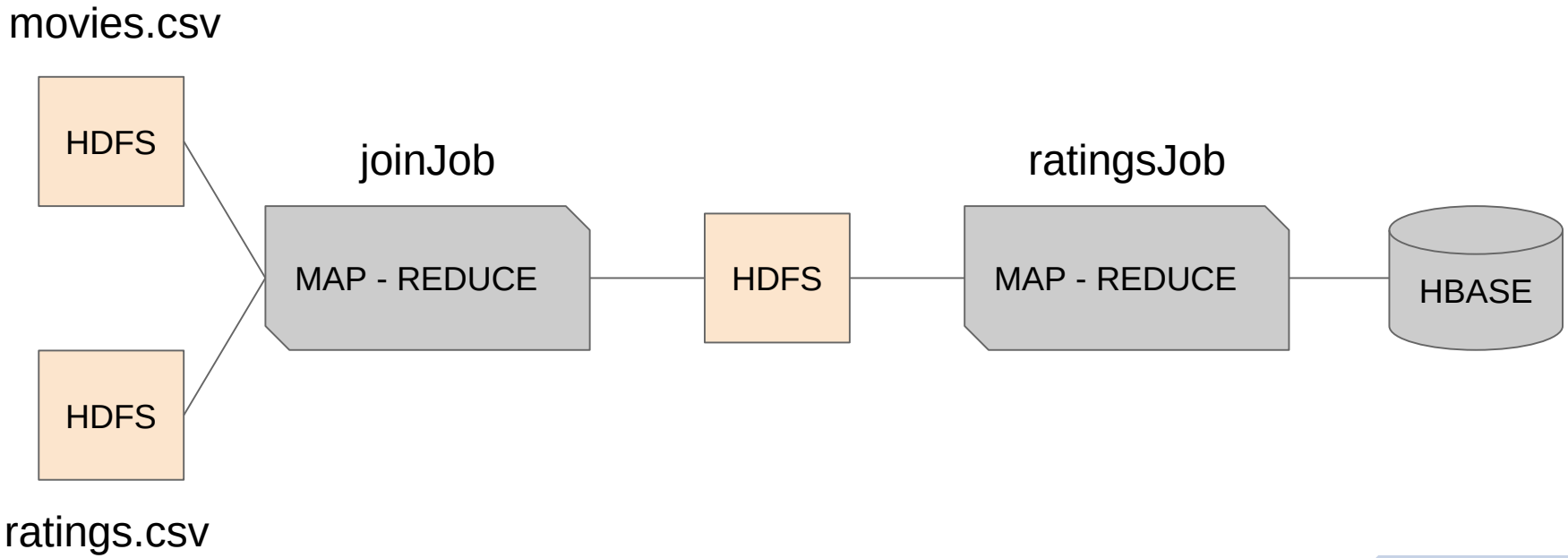


Con questa versione ottimizzata si è ottenuto lo stesso risultato in output ma con tempi più brevi.

Query uno: Architettura seconda versione

- **Mapper del JoinJob**
  - filtra tutte le valutazioni precedenti al 1 Gennaio 2000
  - Emette le seguenti coppie chiave valore:
    - key = movieId value = “M” + titolo
    - key = movieId value = “R” + rating
- **Reducer del JoinJob** computa la media dei valori relativi ad un film e la memorizza su HBase.

Query due: Architettura



Query Due: Architettura

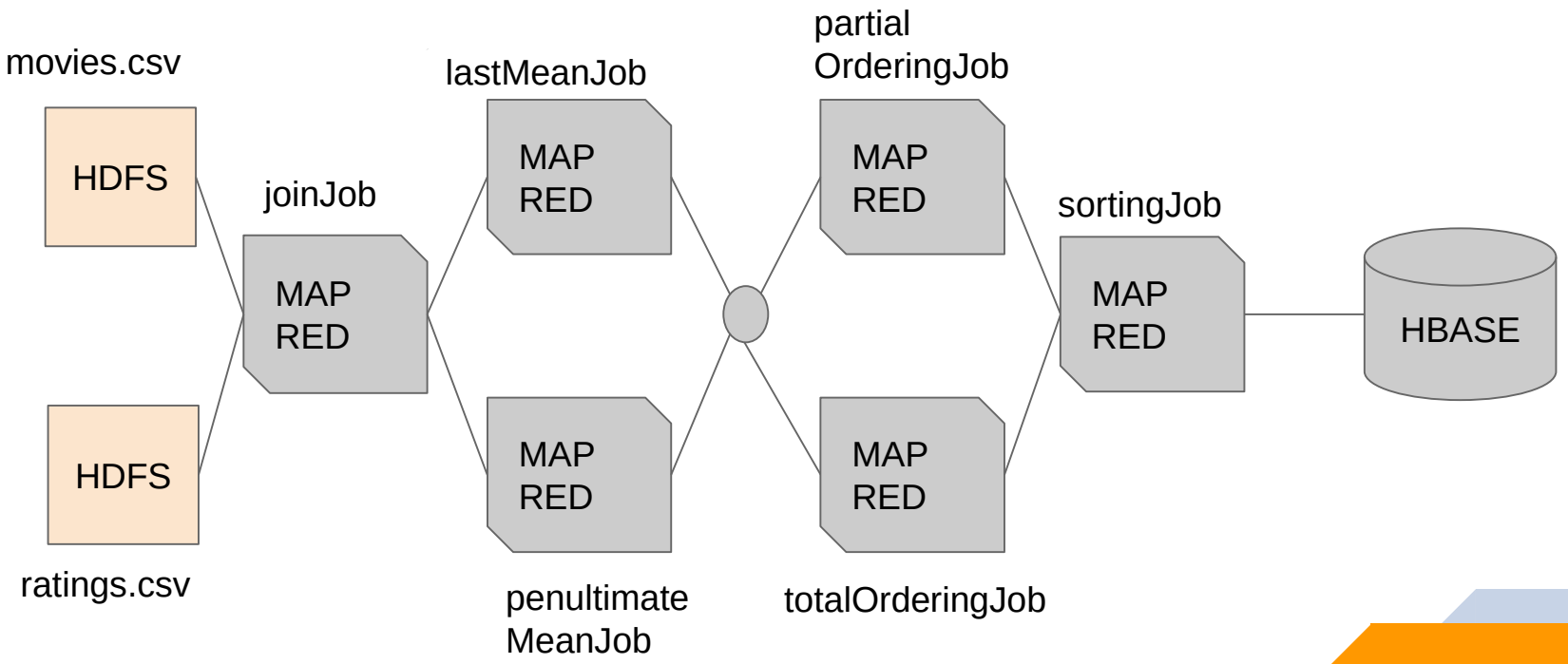
- **Mapper del JoinJob**
  - Emette le seguenti coppie chiave valore:
    - key = movieId value = “M” + genere
    - key = movieId value = “R” + rating

- **Reducer del JoinJob**
  - Emette la seguente coppia chiave valore:
    - key = genere value = voto

Query Due: Architettura

- **Reducer del ratingsJob**
  - computa la media e la deviazione standard dei valori relativi ad un genere e li memorizza su HBase.

Query Tre: Architettura prima versione



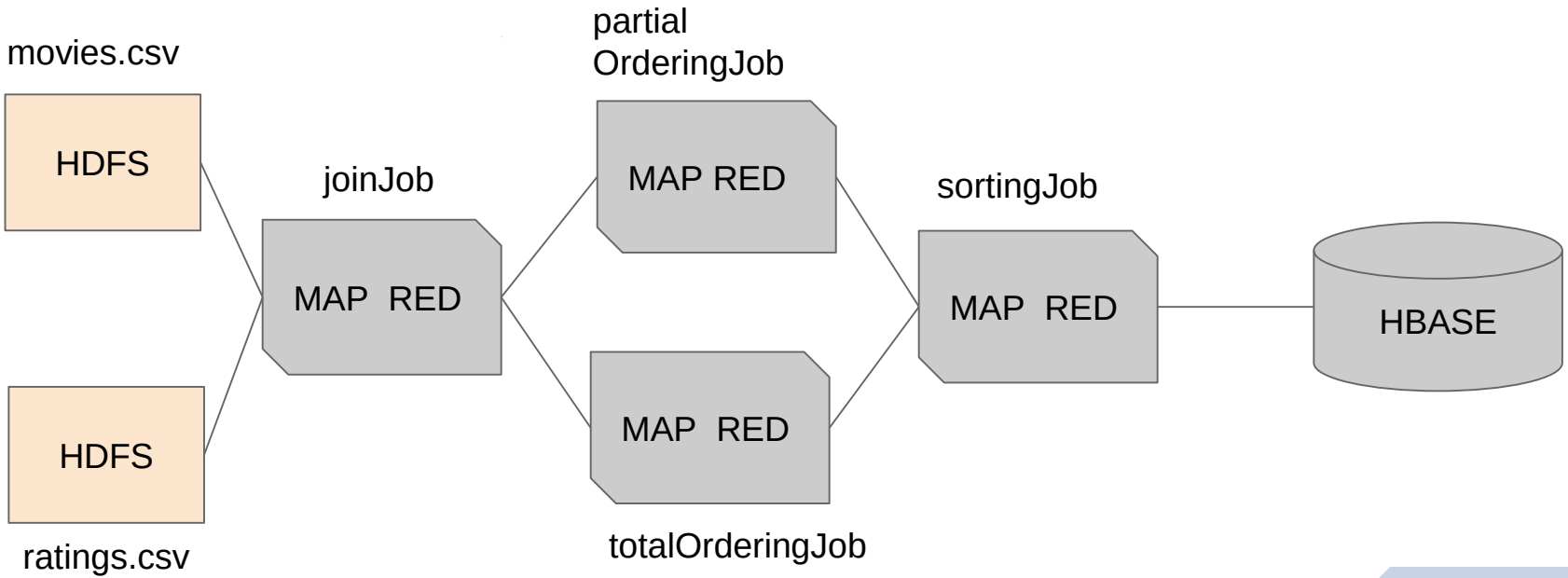
Query Tre

- **Job Control** per la gestione dei job in parallelo.
- Il **joinJob** della query3 effettua il salvataggio dell'output in due cartelle distinte:
  - una per i rating dell'ultimo anno, che verranno analizzati da lastMeanJob
  - una per i rating del penultimo anno, che verranno analizzati da penultimateMeanJob
- **lastMeanJob** calcola il rating medio dei film dell'ultimo anno e memorizza tutto in un file che sarà input del job PartialOrderingJob

Query Tre

- **penultimateMeanJob** calcola il rating medio dei film del penultimo anno e memorizza tutto in un file che sarà input
  del job totalOrderingJob
- **partialOrderingJob** e **totalOrderingJob** effettuano un ordinamento totale dei film basato sul rating.
  - Tale job sfrutta il *TotalOrderingPartitioner* ed effettua la partizione tramite *InputSampler.RandomSampler*
- **storingJob** effettua il conteggio della posizione dei singoli film, e filtra i film, memorizzando su HBase

Query Tre: Architettura seconda versione



Tempi sperimentali

- Tempi sperimentali medi ottenuti eseguendo le versioni finali delle query sull'ambiente descritto in precedenza:
  - tempi nell'ordine del minuto

	Tempi medi di esecuzione medio
Query uno	130004 ms
Query due	321021 ms
Query tre	117007 ms

Tempi sperimentali: architetture a confronto

Confrontiamo i tempi impiegati dalle query 1 e 3 prima e dopo le modifiche architetturali di cui abbiamo discusso.

	Tempo medio prima versione	Tempo medio seconda versione
Query uno	395010 ms	130004 ms
Query tre	368407 ms	117007 ms

Gabriele Belli

GRAZIE PER L’ATTENZIONE.

Giuseppe Chiapparo