Building Complex Data Workflows with Cascading on Hadoop

Gagan Agrawal, Sr. Principal Engineer, Snapdeal

Agenda

- What is Cascading
- Why use Cascading
- Cascading Building Blocks
- Example Workflows
- Testing
- Advantages / Disadvantages
- Cascading @ Snapdeal

What is Cascading ?

What is Cascading ?

- Abstraction over Map Reduce
- API for data-processing workflows
- JVM framework and SDK for creating abstracted data flows
- Translates data flows into actual Hadoop/RDBMS/Local jobs

Why use Cascading ?

Why use Cascading ?

- Options in Hadoop
  - Map Reduce API
  - Pig
  - Hive

Why use Cascading ?

- Map Reduce is
  - Low Level API
  - Requires lot of boilerplate code
  - Can be difficult to chain jobs

Why use Cascading ?

- Pig
  - Good for scripting
  - Can be difficult to manage multi-module workflows
  - Can be difficult to Unit Test
  - UDF written in different language (Java, Python (via streaming) etc.)

Why use Cascading ?

- Hive
  - Good for simple queries
  - Complex workflows can be very difficult to implement

Why use Cascading ?

- Hive
  - Good for simple queries
  - Complex workflows can be very difficult to implement

Why use Cascading ?

- Cascading
  - Helps create higher level data processing abstractions
  - Sophisticated data pipelines
  - Re-usable components

Cascading Building Blocks

Cascading Building Blocks

- Tap
  - Source
  - Sink
- Pipes
  - Each
  - Every
  - Merge
  - GroupBy
  - CoGroup
  - HashJoin

Cascading Building Blocks

- Tap
  - Source
  - Sink
- Pipes
  - Each
  - Every
  - Merge
  - GroupBy
  - CoGroup
  - HashJoin

Cascading Building Blocks

- Tap
  - Source
  - Sink
- Pipes
  - Each
  - Every
  - Merge
  - GroupBy
  - CoGroup
  - HashJoin

Cascading Building Blocks

- Functions
- Filter
- Aggregator
- Buffer
- Sub Assemblies
- Flows
- Cascades
-

Cascading Terminologies

- Flow
  - A path for data with some number of inputs, some operations, and some outputs
- Cascade
  - A series of connected flows
- Operation
  - A function applied to data, yielding new data
-

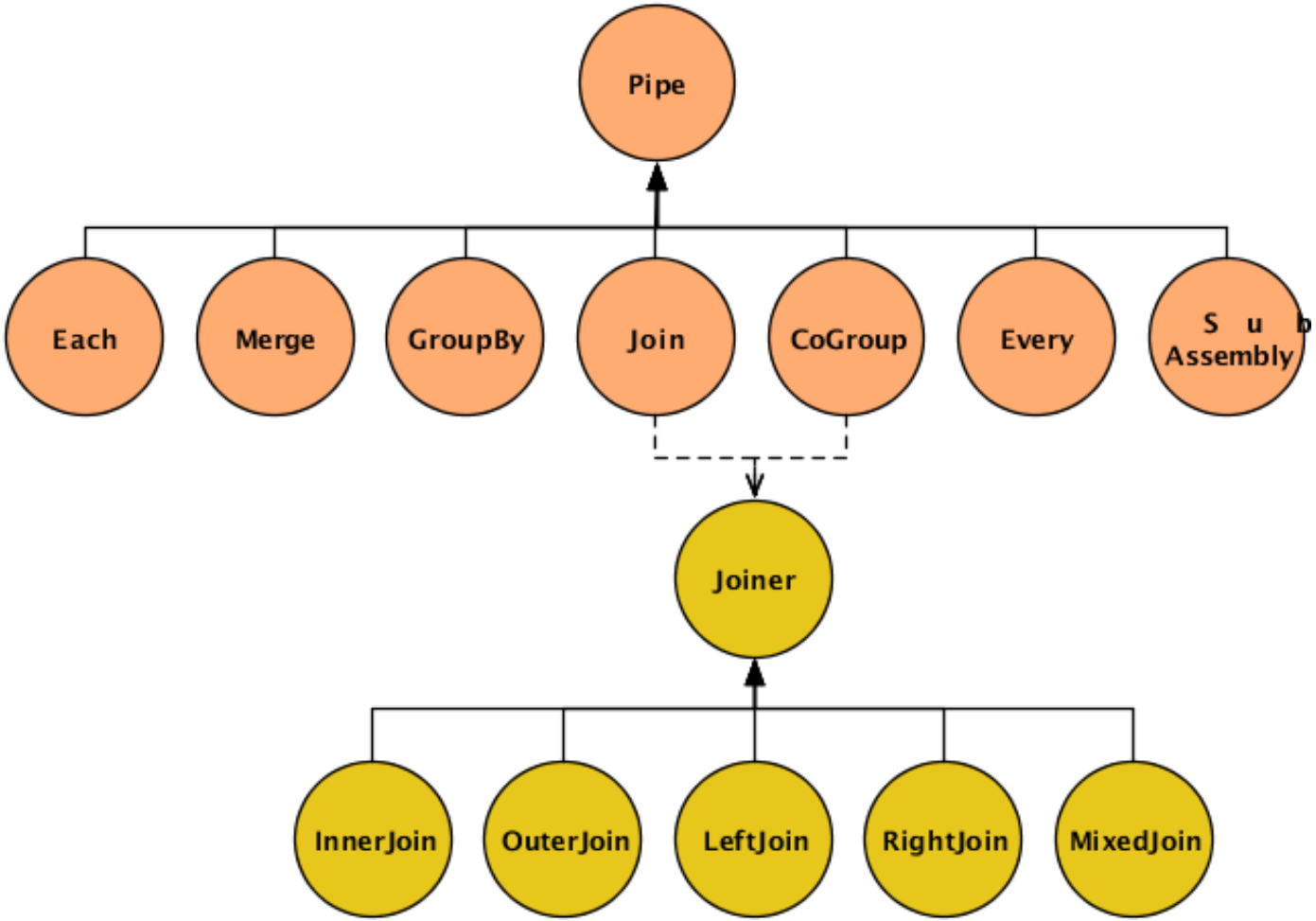
Cascading Terminologies

- Pipe
  - Moves data from some place to some other place
- Tap
  - Feeds data from outside the flow into it and writes data from inside the flow out of it.

Pipe Assemblies

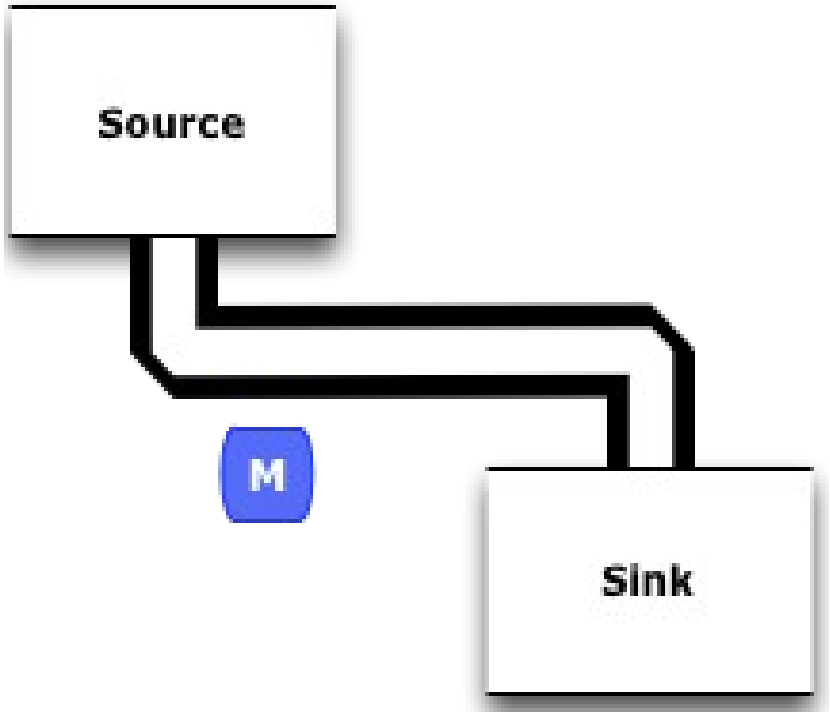
- Define work against a Tuple Stream
- May have multiple sources and sinks
  - Splits
  - Merges
  - Joins

Pipe

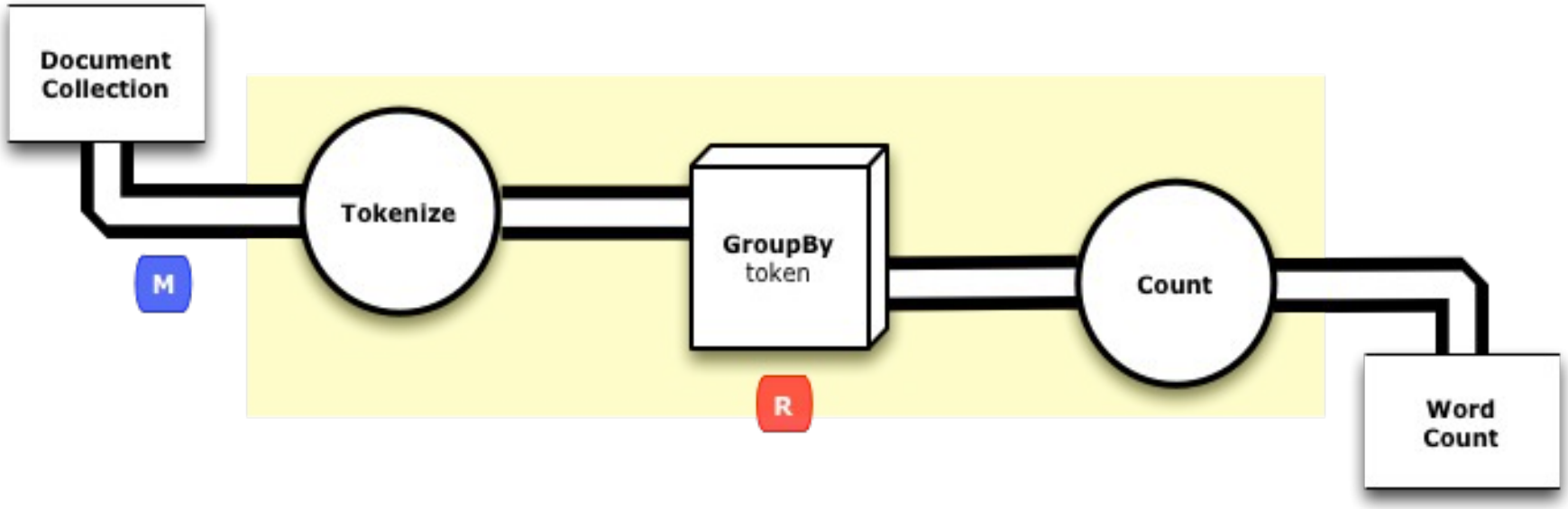

Pipe
Each
Merge
GroupBy
Join
CoGroup
Every
Sub Assembly
Joiner
InnerJoin
OuterJoin
LeftJoin
RightJoin
MixedJoin

Example Workflows

Distributed Copy



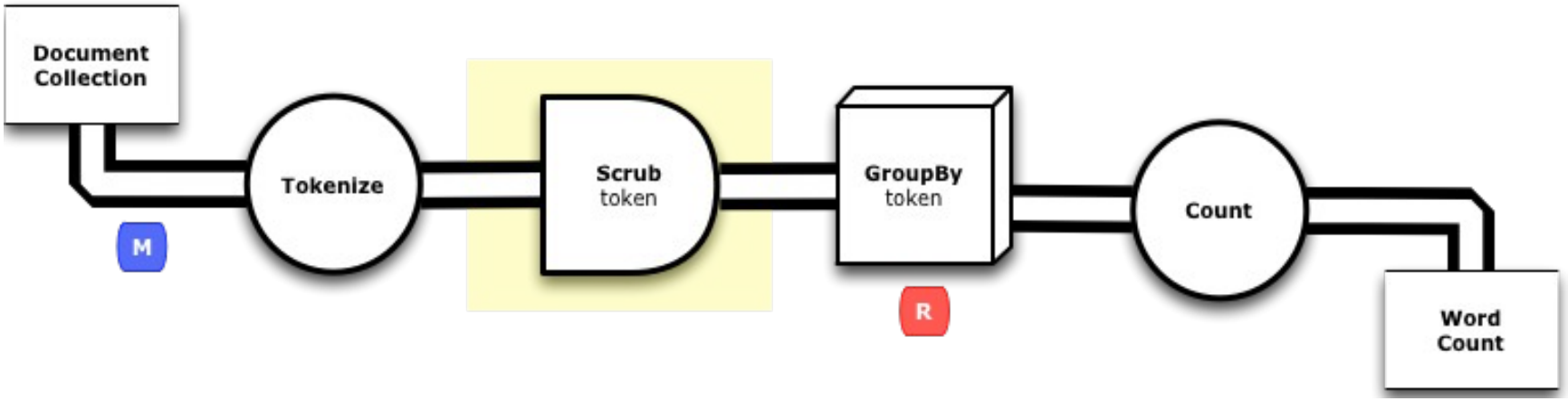
Word Count


Document Collection
Tokenize
M
GroupBy
token
R
Count
Word Count

Custom Function

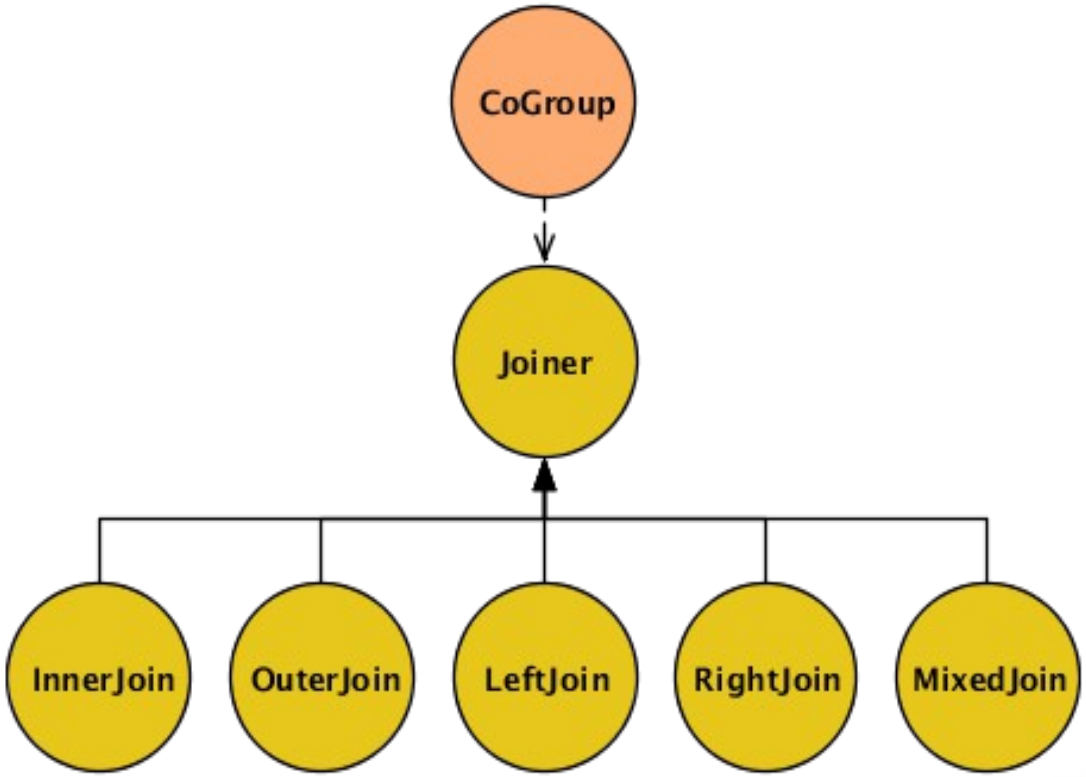
- A function expects a stream of individual tuples
- Returns zero or more tuples
- Used with Each pipe
- To create custom function
  - Sublcass *cascading.operation.BaseOperation*
  - Implement *cascading.operation.Function*

Custom Function

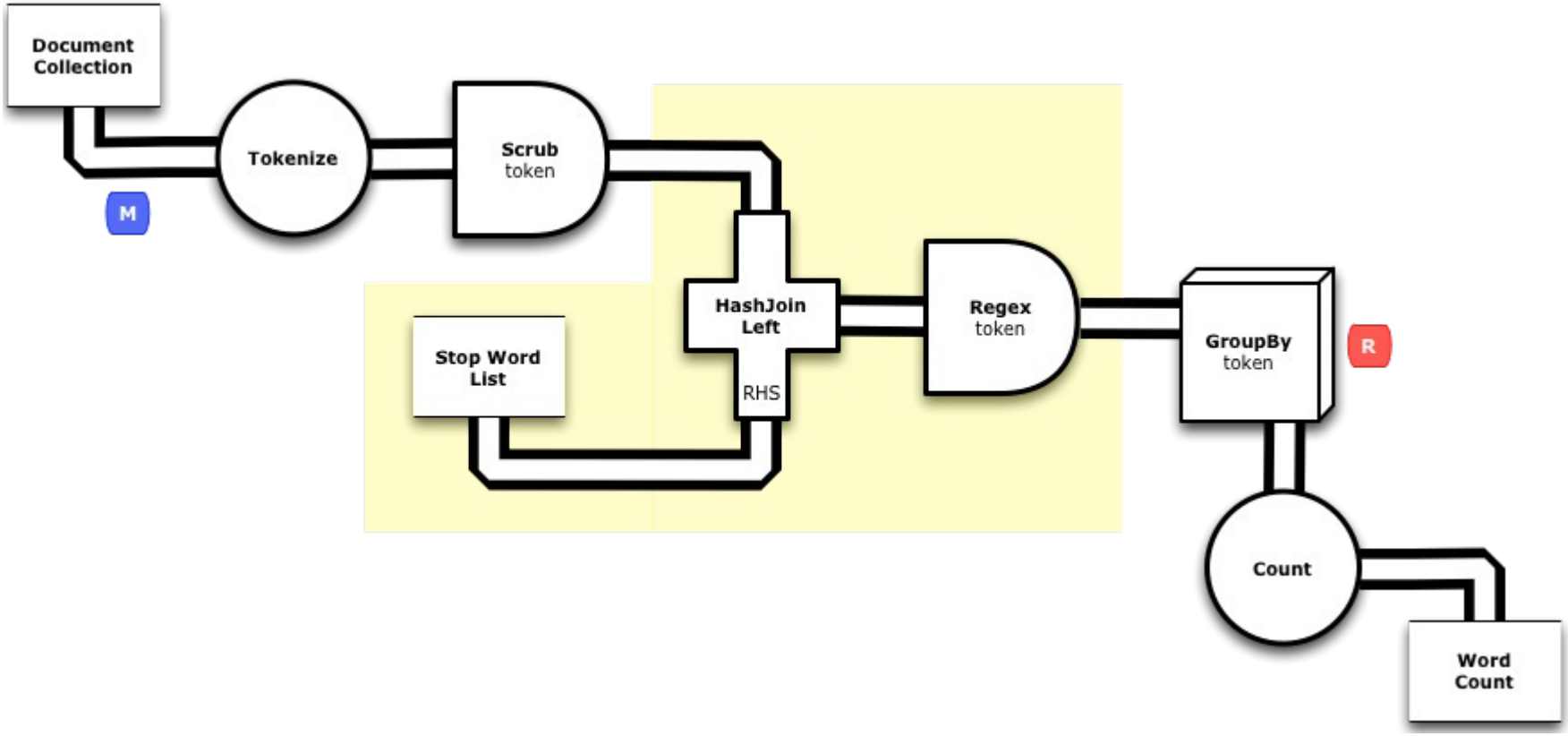


Joins

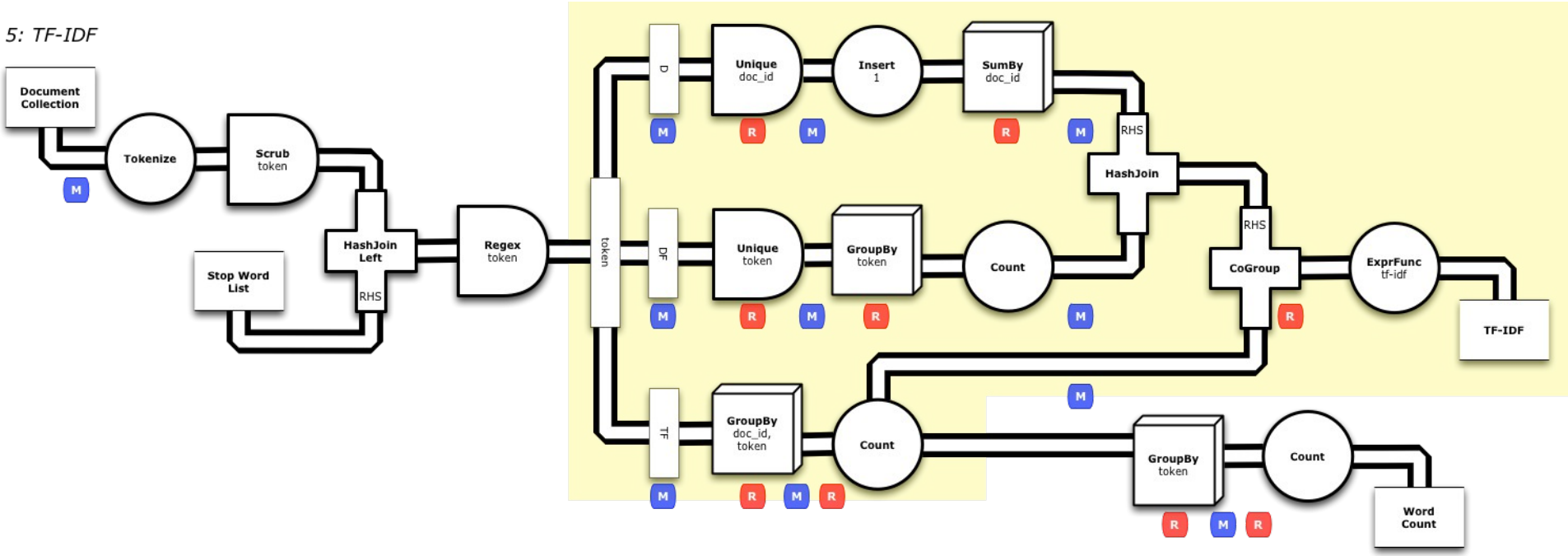
- CoGroup
- HashJoin



Joins


Document Collection
Tokenize
M
Scrub
token
Stop Word List
HashJoin Left
RHS
Regex
token
GroupBy
token
R
Count
Word Count

TF-IDF Example



Testing

Testing

- Create flows entirely in code on a local machine
- Write tests for controlled sample data sets
- Run tests as a regular old Java without needing access to actual Hadoop or databases
- Local machine and CI testing are easy

Advantages / Disadvantages

Advantages

- Re-usability
  - Pipe assemblies are designed for reuse
  - Once created and tested, use them in other flows
  - Write logic to do something only once
- Simple Stack
  - Cascading creates DAG of dependent jobs for us
  - Removes most of the need for oozie

Advantages

- Simpler Stack
  - Keeps track of where a flow fails and can rerun from that point on failure
- Common Code Base
  - Since everything is written in java, everybody can use same terms and same tech

Disadvantages

- JVM Based
  - Java / Scala / Clojure
- Doesn't have job scheduler
  - Can figure out dependency graph for jobs, but nothing to run them on regular interval
  - We still need scheduler like quartz

Disadvantages

- No real built-in monitoring
- Easy to have a flow report what it has done; hard to watch it in progress

Cascading @ Snapdeal

Thank you!