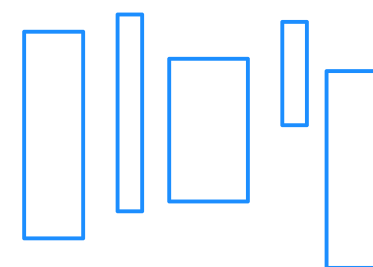


Cascading

www.cascading.org

Chris K Wensel
chris@concurrentinc.com



Concurrent, Inc.
www.concurrentinc.com

In a nutshell

- An explicit way of describing ‘how’ data should be processed, independently from ‘what’ data will be processed.
- Process definitions are a pipeline of operations applied to a stream of tuples moving through it.
- At runtime, the process definition is combined with the data sources.
- The result is a set of interdependent MapReduce jobs, scheduled by dependency.
- A set of these resulting processes can be organized and scheduled by dependencies.

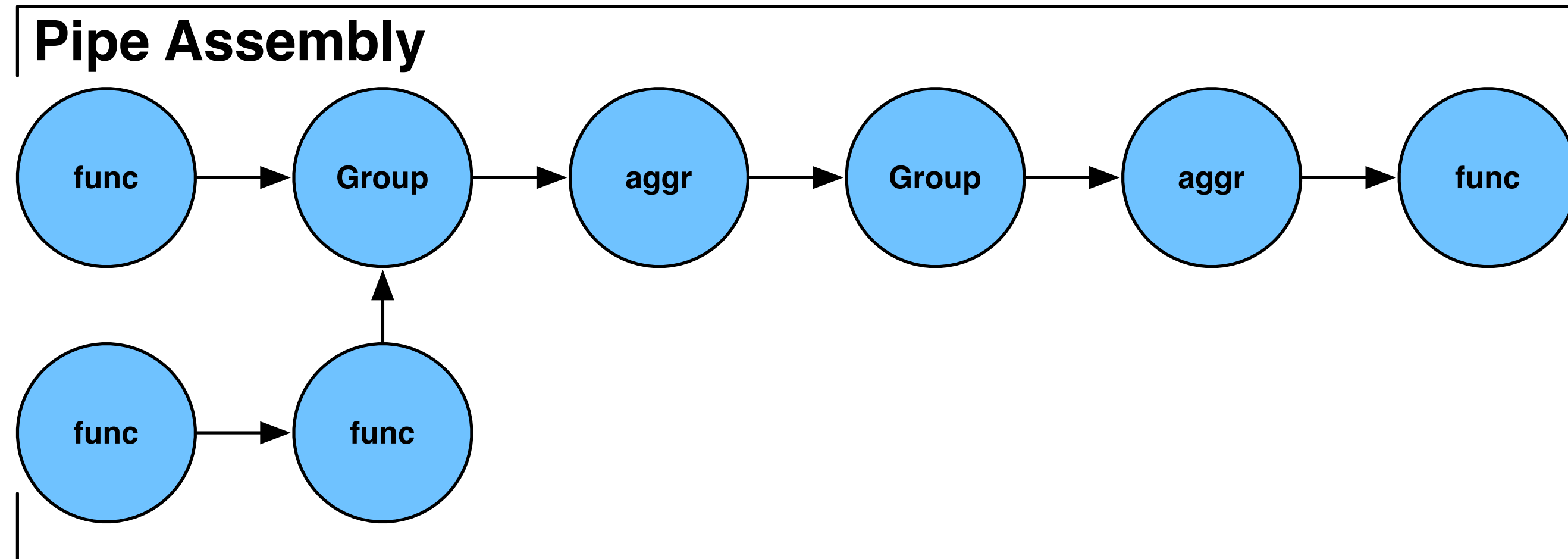
Why this is good

- No need to think in MapReduce
- Processing definitions are composable and reusable
- Result data-sets defined by process definitions can be lazily (re)evaluated, or 'cached' for other processes

Features

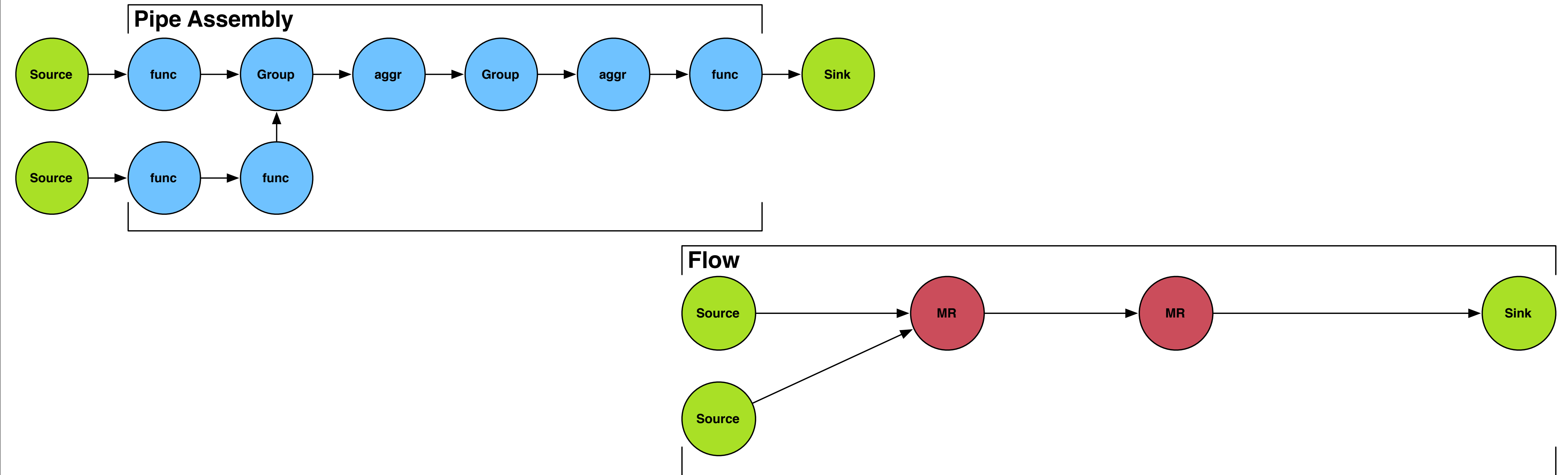
- Process assembly API
- Job planner
- Topological Scheduler
- Joins and Merges
- Stream assertions
- Failure Traps
- Job event notification
- Custom MapReduce jobs
- Scriptable

Processing API



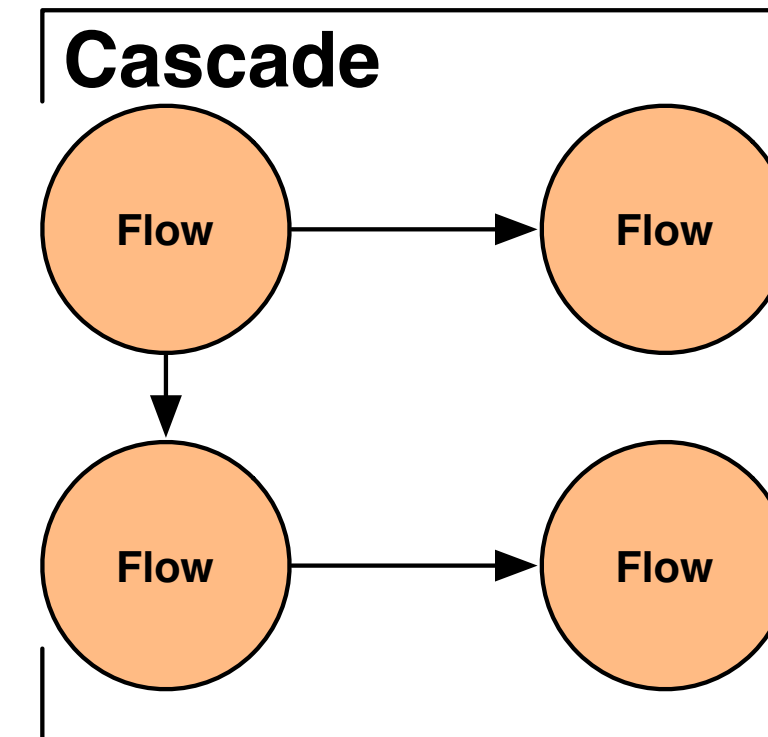
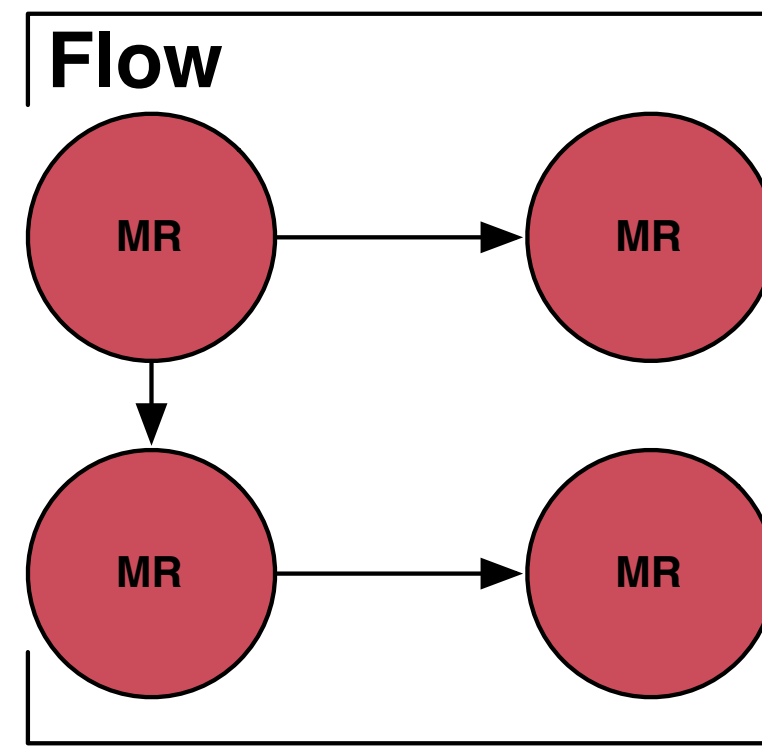
- Operations are chained together to define a Pipe assembly or a reusable sub-assembly

Job Planner



- Pipe Assemblies become Flows
- Translates a DAG of operations to a DAG of MapReduce jobs

Topological Scheduler

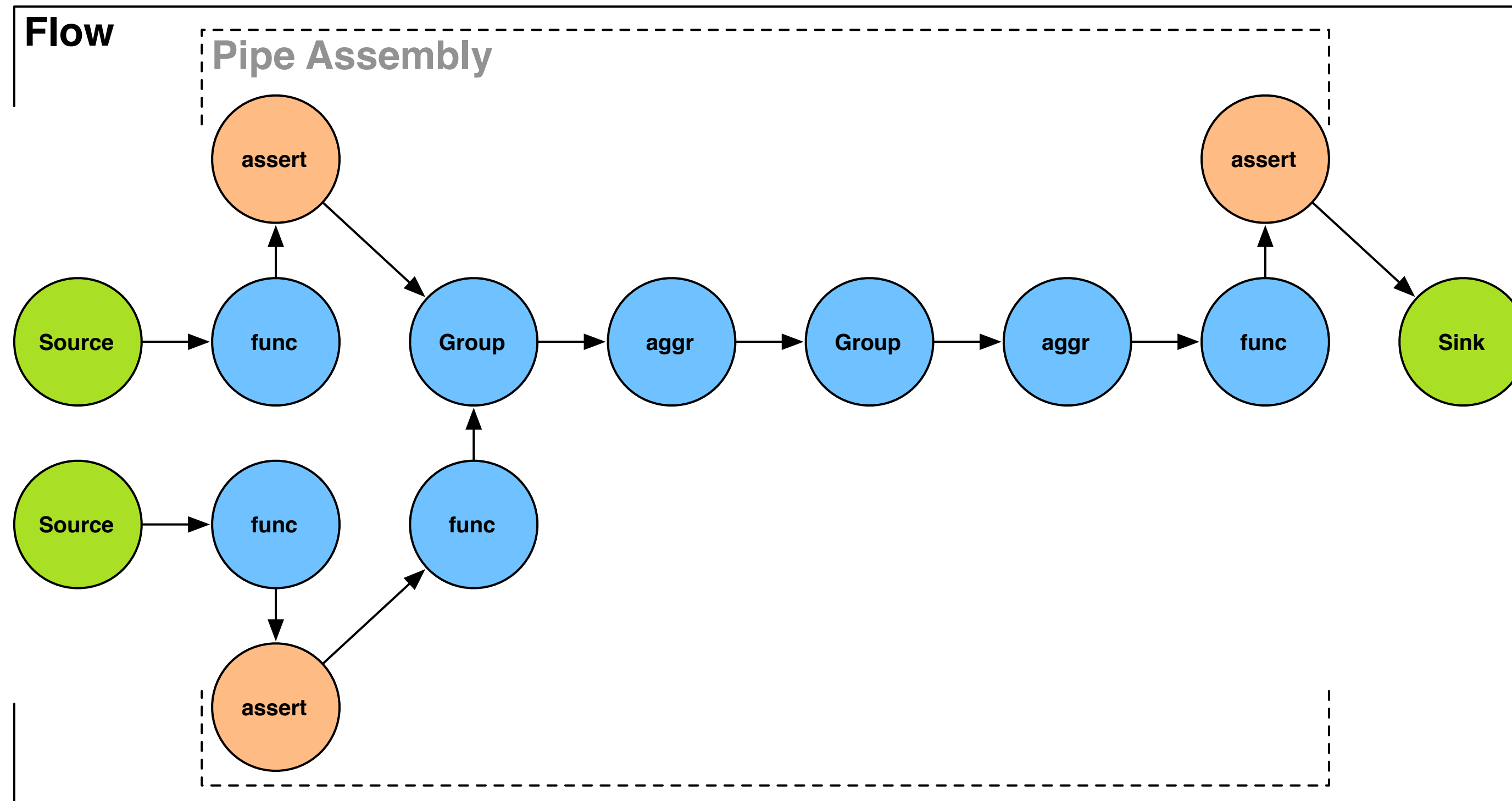


- All MapReduce jobs in Flow scheduled in dependency order
- Flows can be combined into single process, a Cascade
- Flows scheduled by Cascade in dependency order
- Cascade can rerun only 'stale' Flows
- Custom MapReduce jobs can participate in Cascade

Joins and Merges

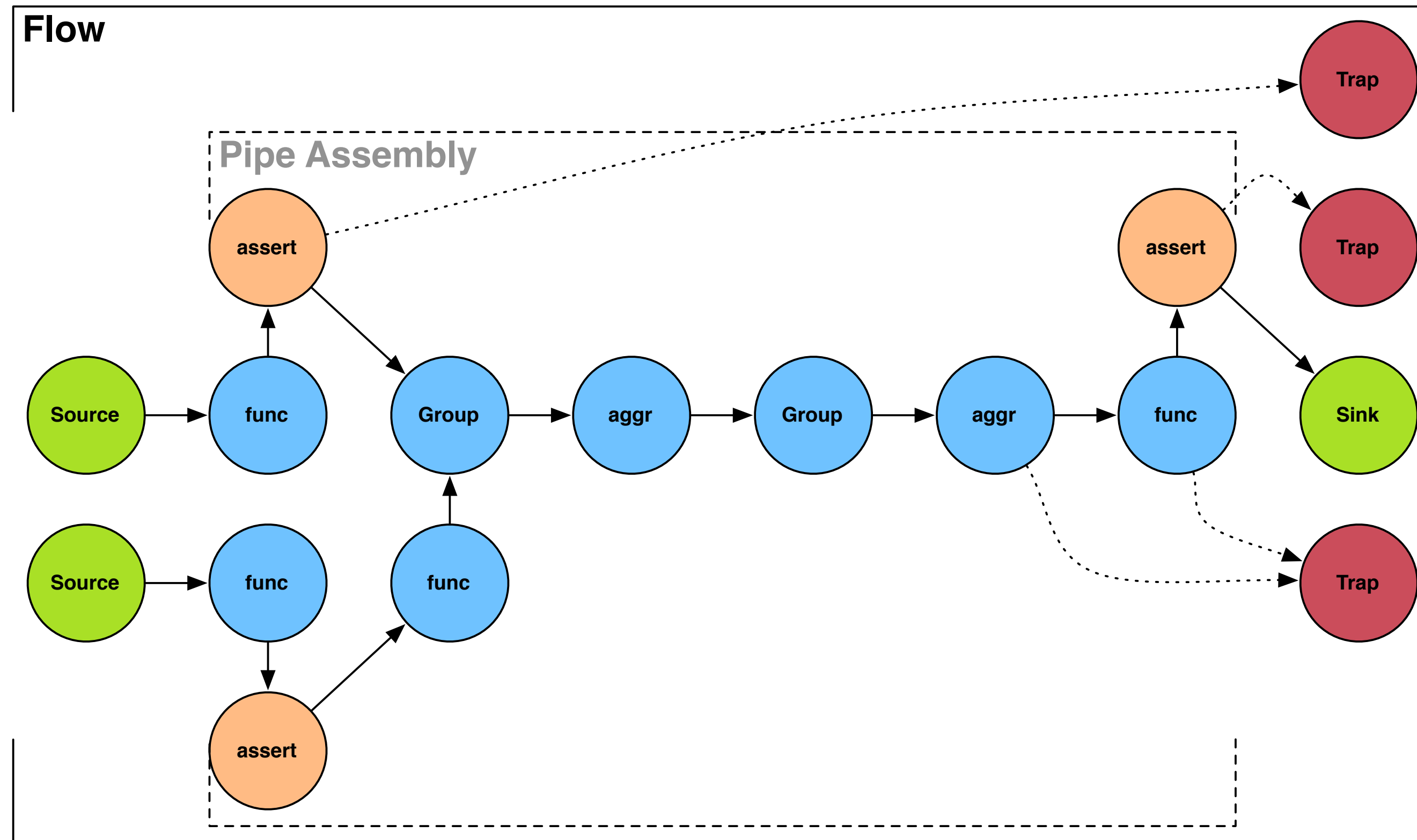
- Inner, Outer, Left, Right, and Mixed (N streams) Joins
- Merge of N number of heterogeneous streams into one normalized stream

Stream Assertions



- Unit and Regression tests for Flows
- Planner can remove 'strict', 'validating', or all assertions

Failure Traps



- Catch data causing Operations or Assertions to fail
- Allows processes to continue without data loss

Event Notification

- Listeners on Flows allow for notification for common events (start, complete, failed, and stopped)
- Useful for notifying external applications a process has completed

Custom MapReduce

- Anything that can be stuffed into a JobConf can be managed by a Cascade

Scriptable

- Scripts are used for assembly and/or implementing operations
- Groovy DSL available now
- Groovy does not run on cluster

Samples

