Map-Reduce Framework

Tuesday, February 14, 2023 2:21 PM

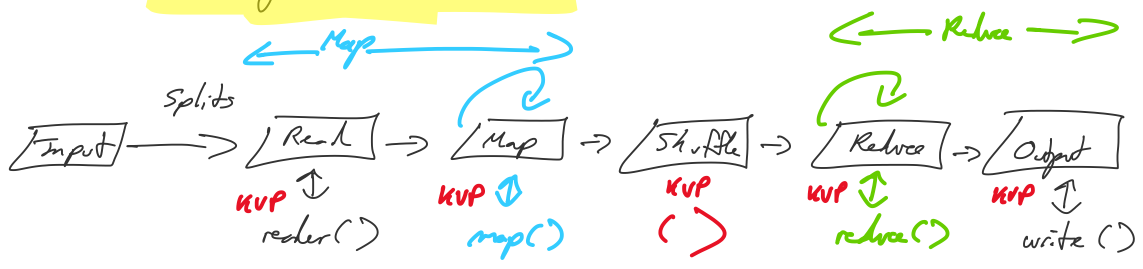
- Map-Reduce Framework
  - Framework
    - Services: (Zookeeper), HDFS / YARN
    - Interfaces: (Common), API

      Wrapper Types

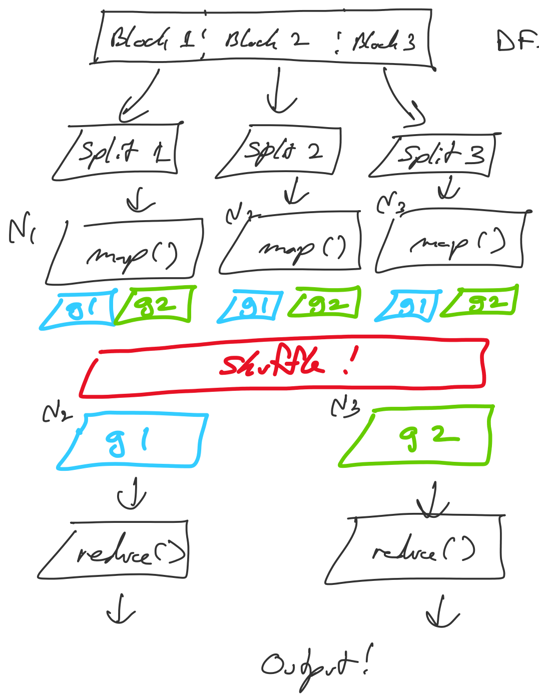
      Driver Hooks / Callbacks

* Data Exchange : Framework ⟷ User Functions

↳ Key-Value Pairs (KVPs)



Ex.



DFS / HDFS

* For this example, Blocks & Splits Align!

Batch Example:

60616, 3

60616, 2

60617, 5

60617, 2

* KVPs : Can be any POD (Plain-Old-Datatype) OR any Complex Datatype

↳ struct

↳ Array / List

↳ Dictionaries ...

⋮

Ex. Name = "Jay"

User = {

Name = "Jay" ?

Id = 401 } ← be a key!

}

✱ Languages : { Java → C/C++ ...
{ Scala

→ Python, R, . . . (SQL)

- Customizations
  - Map Hooks

Input → Map → Combine → Partition :

Record Reader Mapper Combine Partition
- Init
- Final

  - Reduce Hooks

Shuffle → Reduce → Output

Combiner Reducer Output Formatter
- Init
- Final

✱ Combiners
- Localized Reducer → to minimize shuffle I/O cost
- Not guaranteed to execute!
- Mapper has a memory buffer
  ↳ on overflow → spills to disk : Spill Files
  ↳ Run Combiner to merge spill files!

✱ Partitioner
- Intermediate KVP → split / Sharded to have one shard per reducer
- Hash object (md5, sha1 ..)

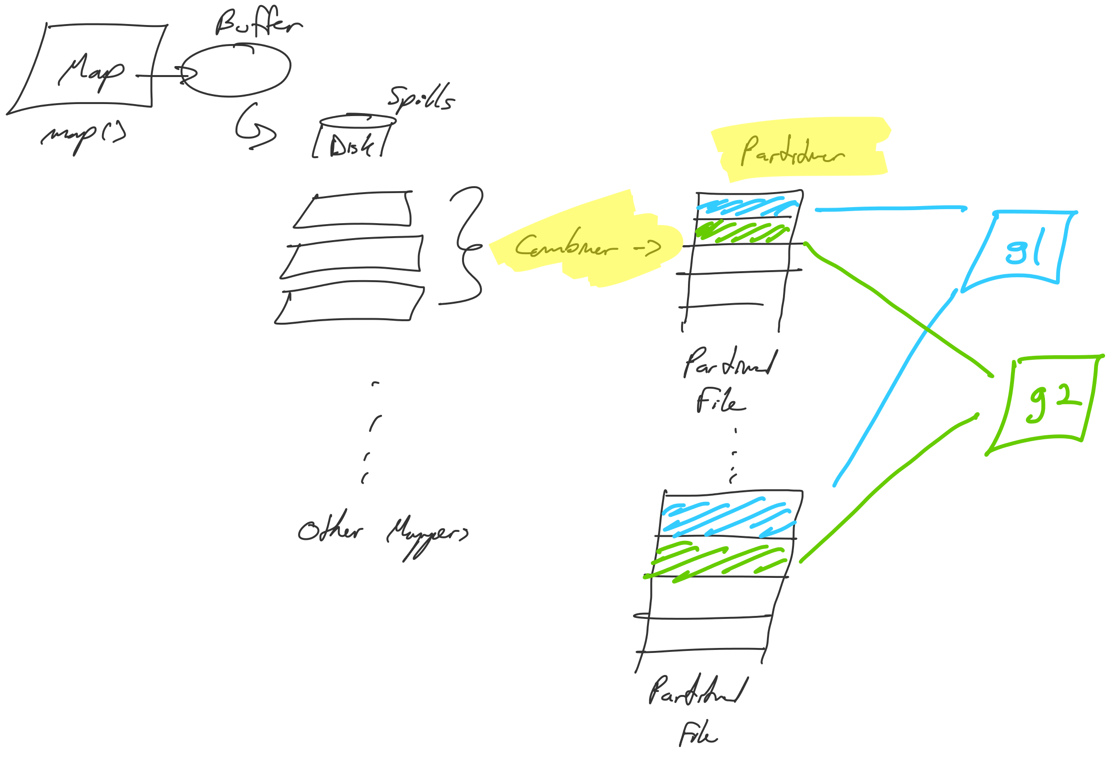
  ↳ hash (key) % # of reducers

- Keyspace of Intermediate KVP → Distribute evenly across Reducer nodes, need to guarantee mapping to identical Reducer nodes

· Shuffle & Sort
  - Map
    · Output to buffer, spill to disk
    · Run Combiner if needed

- Single partitioned file for Reduce step
  ↖ sorted keys within each partition

- Reduce
  - Copy map files by partition to Reduce nodes
  - Run Combiner if needed
  - Multi-Pass Merge
    Sort all Intermediate KVP for Reducer
  - Reducer is run



Map
map()
Buffer
Spills
Disk
Combiner ->
Partitioner
Partitioned File
Other Mappers
Partitioned File
g1
g2