Today

Internet scale Computing (Lesson 9)

✓ * giant scale services

⇒ * Map-reduce programming model

Friday

* CDN (Coral)

* Please watch videos before class *

Map-Reduce

Input & output to each of Map & Reduce

- <Key, Value> Pairs

Map-Reduce

Input & output to each of Map & Reduce

- <Key, Value> Pairs

Example

- Emit # of occurrence of names in DOCs

- W = "Kishore"
- W = "Arun"
- W = "Drew"
- ⋮

Map-Reduce

Input & output to each of Map & Reduce
- <Key, Value> Pairs

Example
- Emit # of occurrence of names in DOCs

Key: filename Value: contents

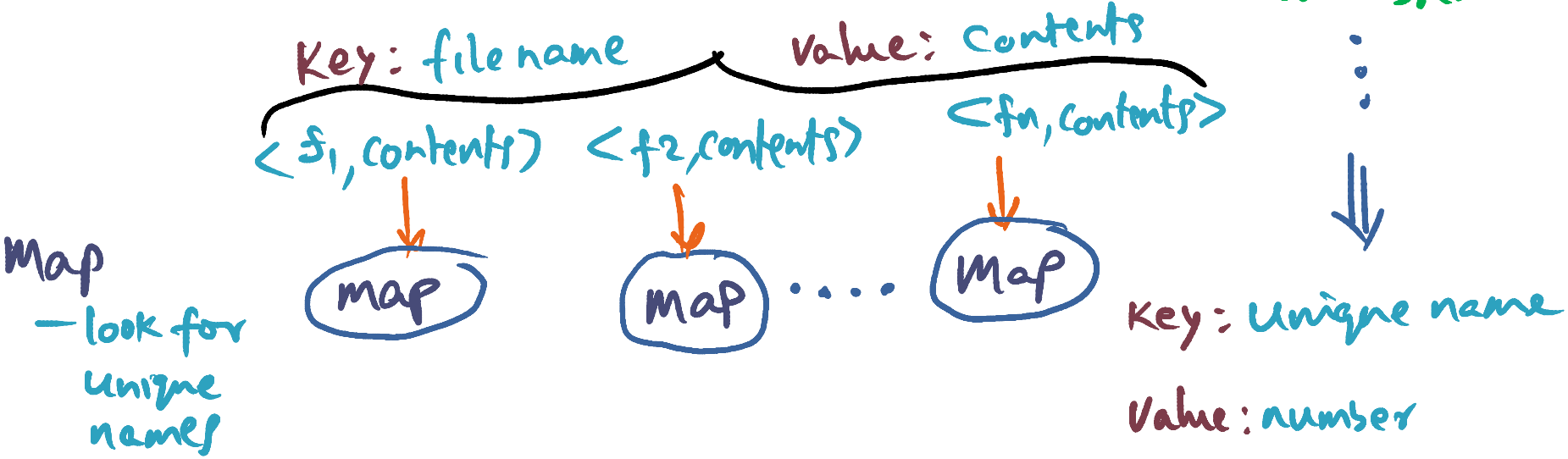
<f1, contents> <f2, contents> <fn, contents>

- W = "Kishore"
- W = "Arun"
- W = "Drew"
- ⋮

Map-Reduce

Input & output to each of map & reduce
- <Key, Value> pairs

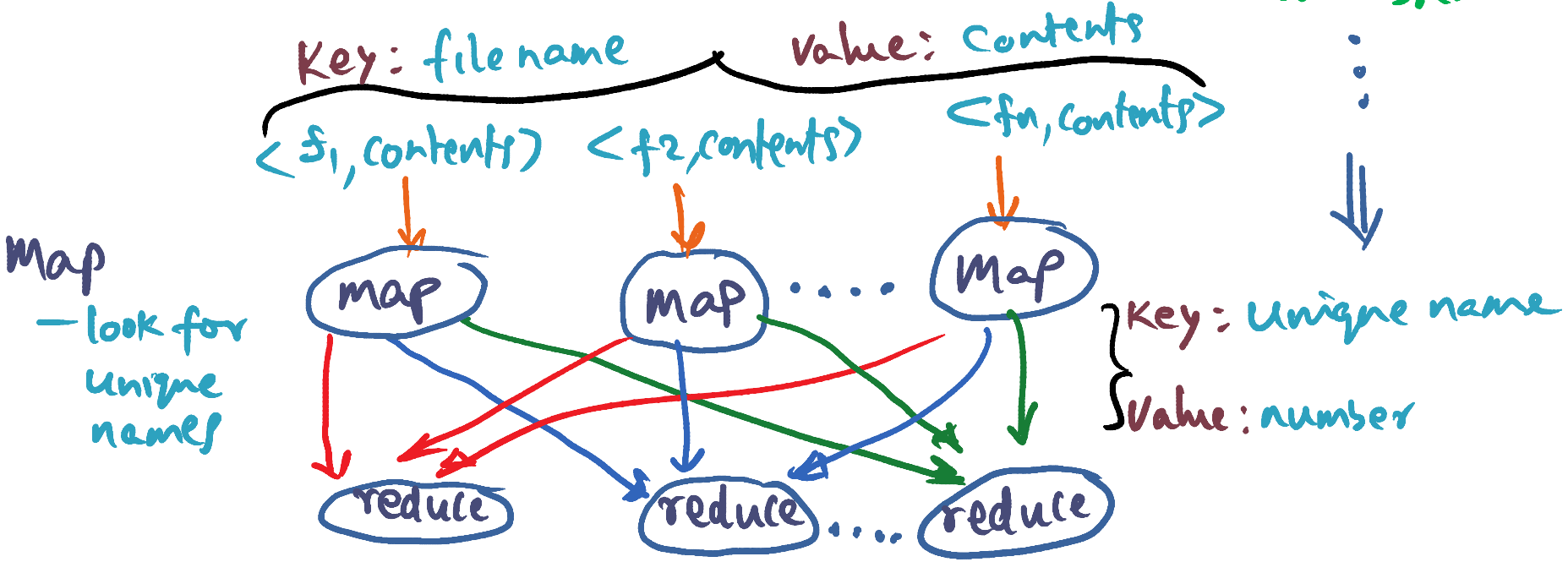
Example
- Emit # of occurrence of names in DOCs



Map-Reduce

Input & output to each of Map & Reduce
- <Key, Value> Pairs

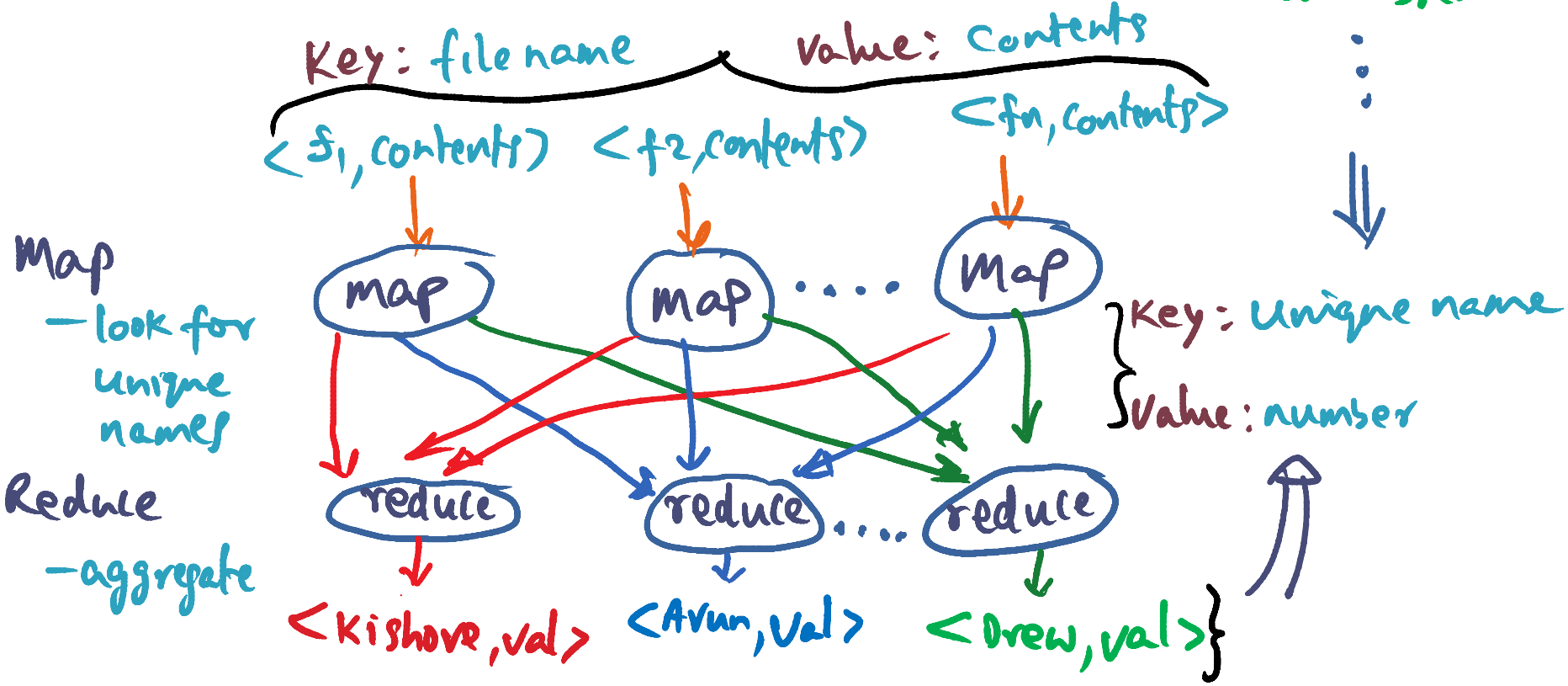
Example
- Emit # of occurrence of names in DOCs



Map-Reduce

Input & output to each of map & reduce
- <Key, Value> pairs

Example
- Emit # of occurrence of names in DOCs



Why Map-Reduce?

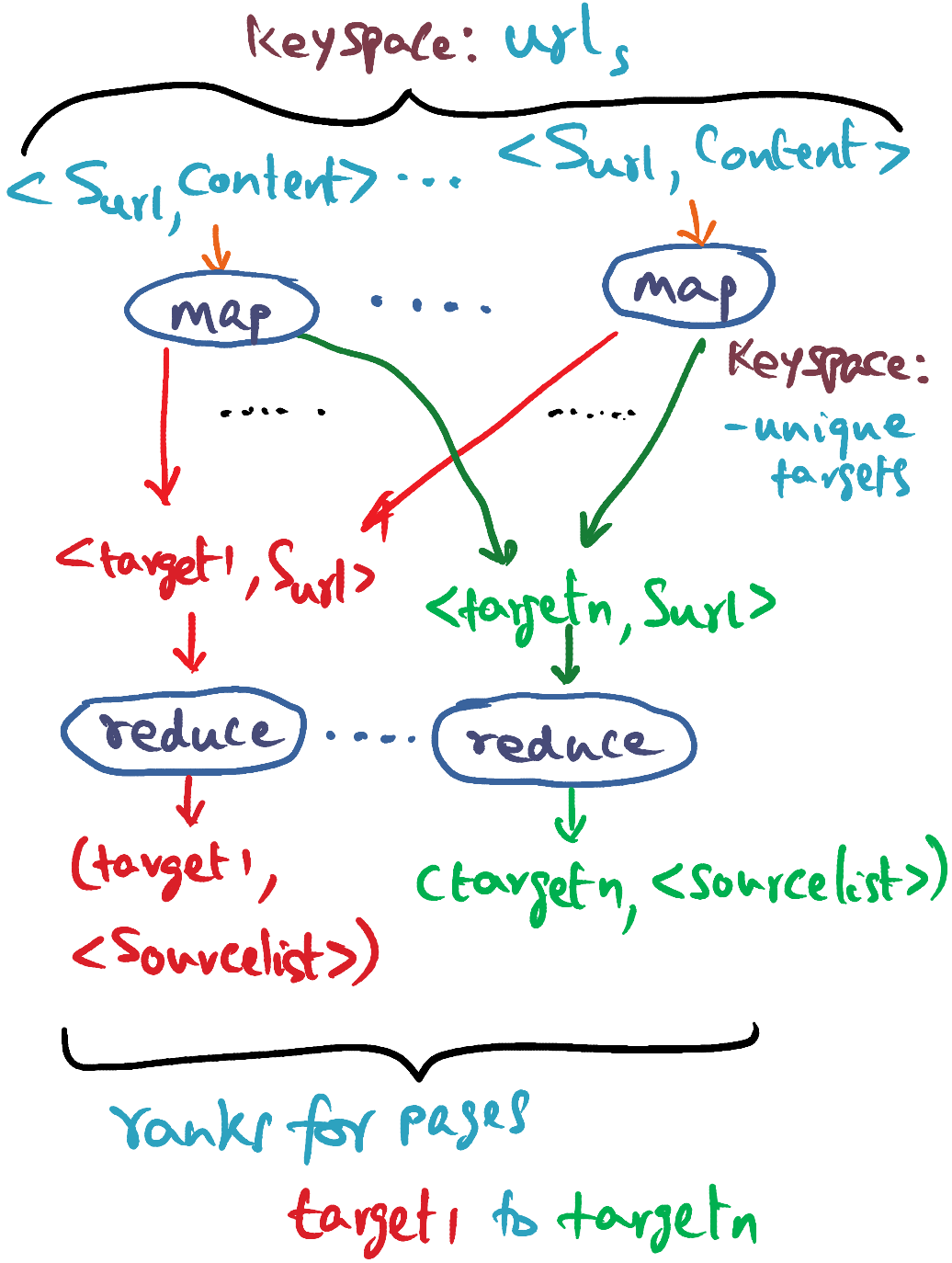
- Several processing steps in giant-scale services expressible

Why Map-Reduce?

- Several processing steps in giant-scale services expressible
- Domain expert writes
  * Map
  * reduce

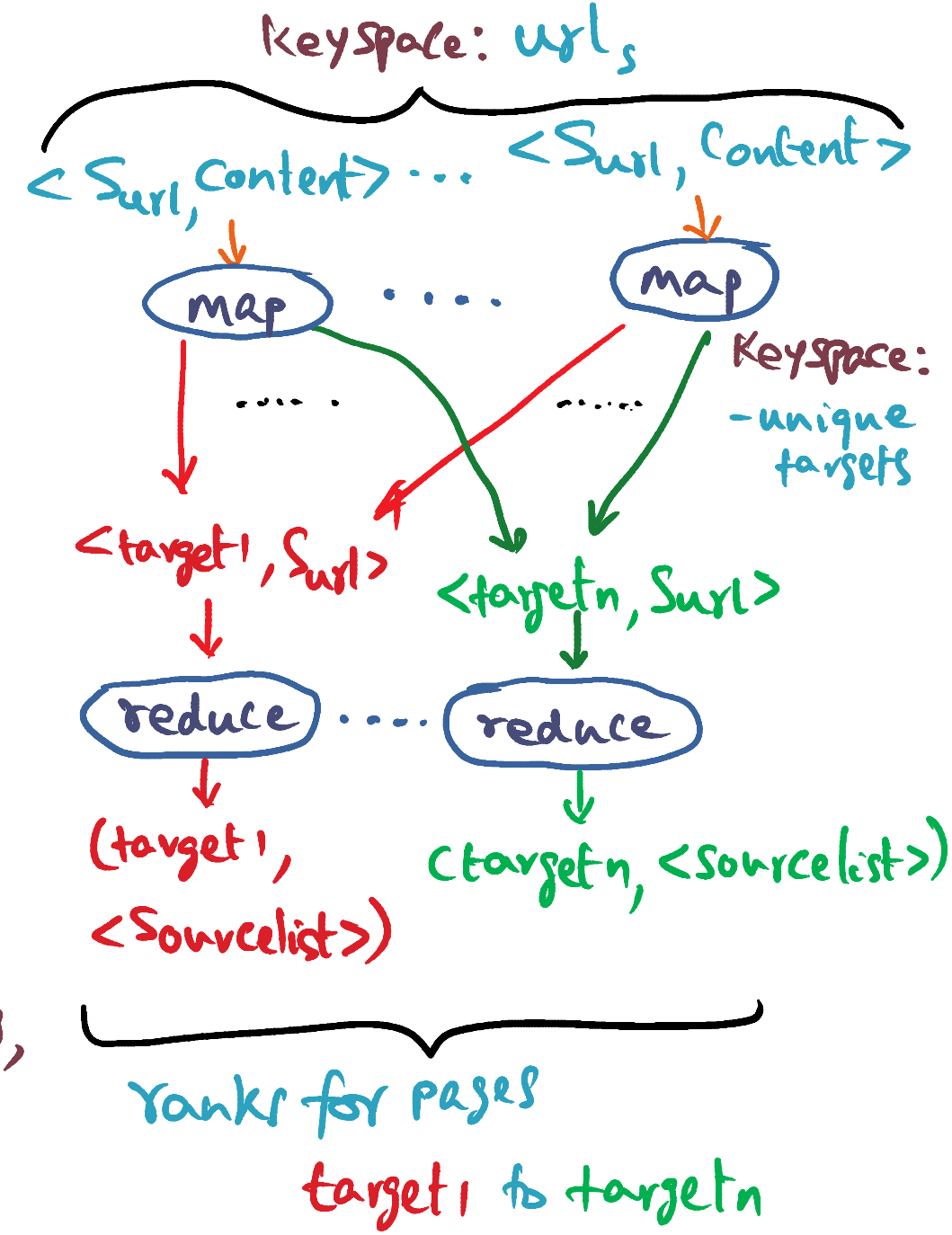
Why Map-Reduce?

- Several processing steps in giant-scale services expressible
- Domain expert writes
  * Map
  * reduce

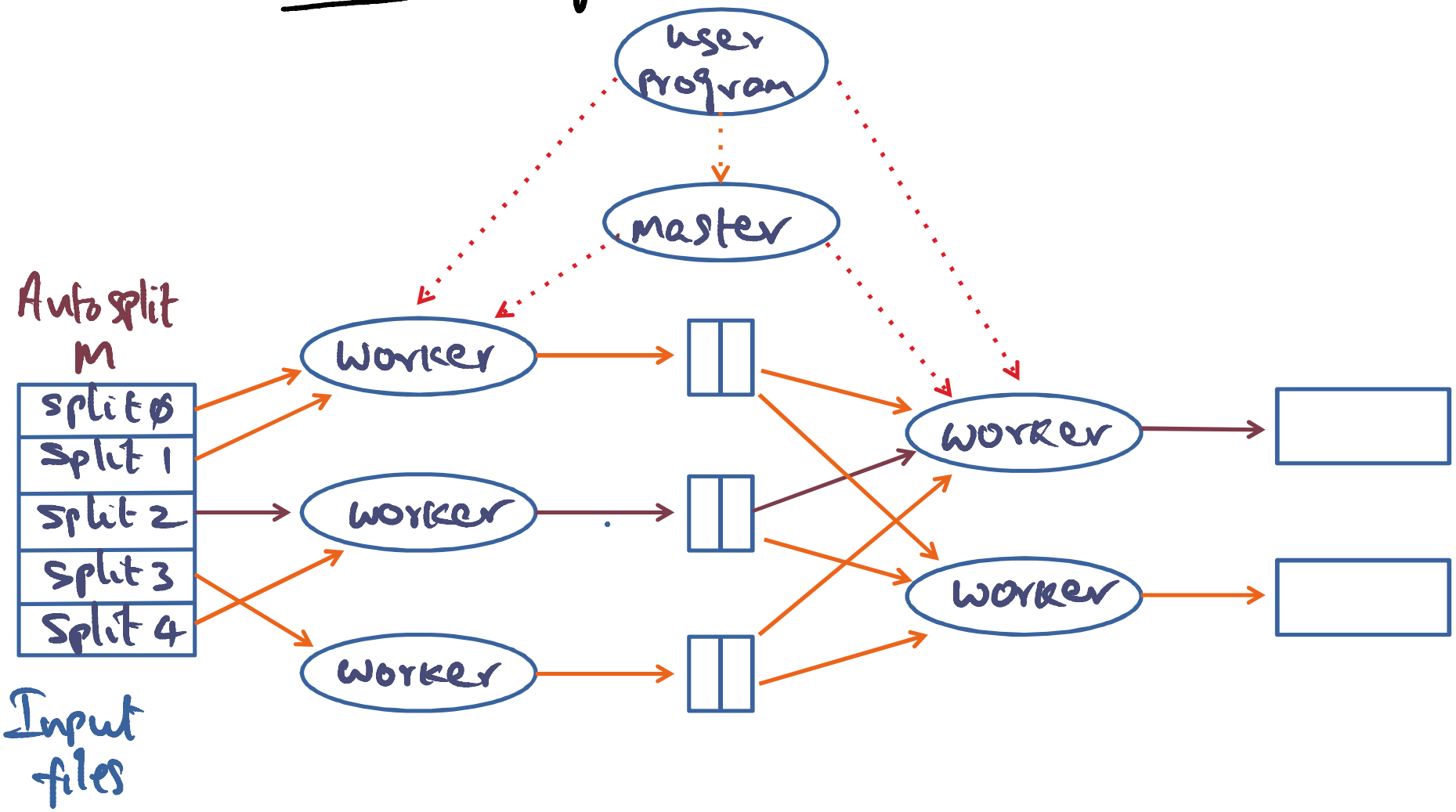


Why Map-Reduce?

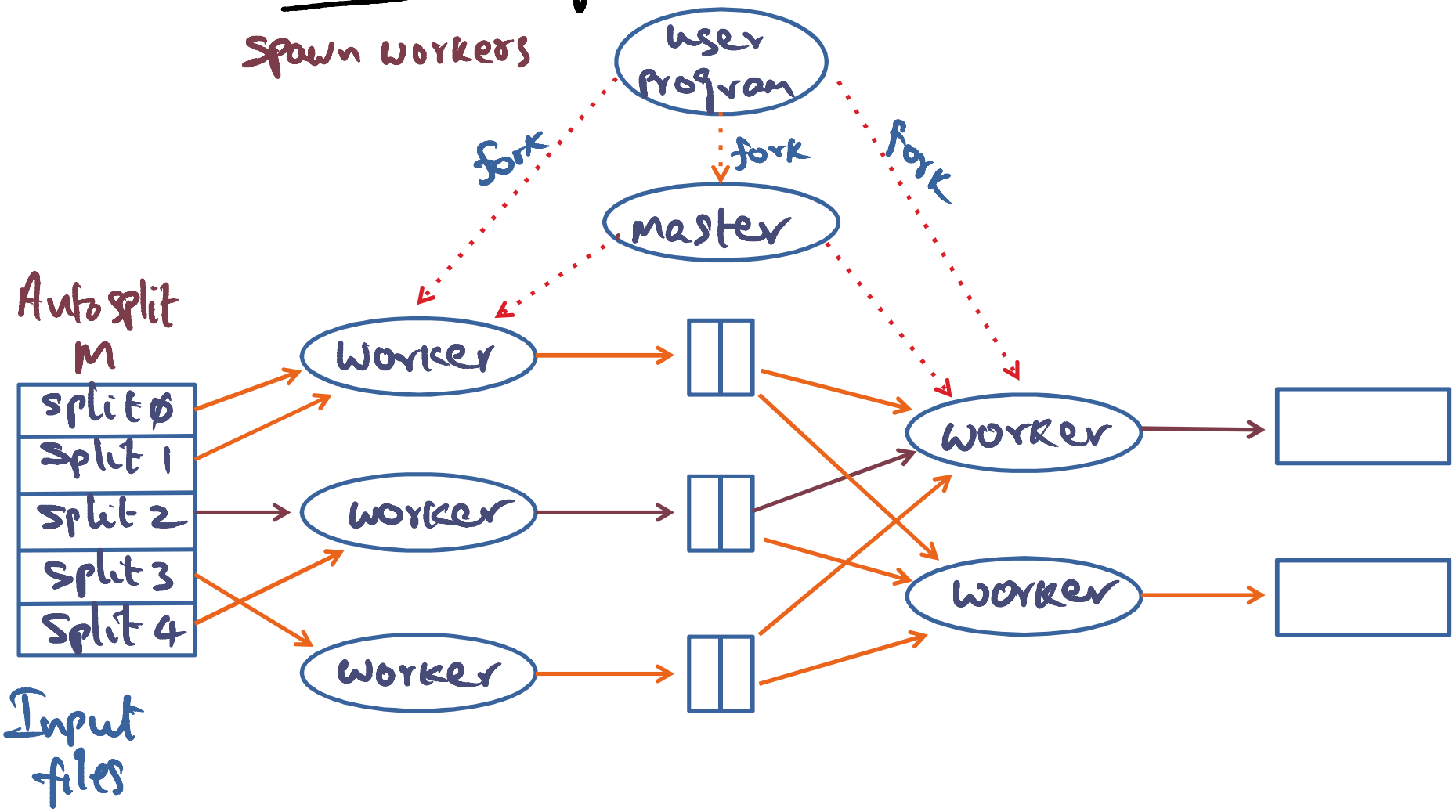
- Several processing steps in giant-scale services expressible
- Domain expert writes
  - * Map
  - * reduce
- runtime does the rest
  - * instantiating number of mappers, reducers
  - * data movement



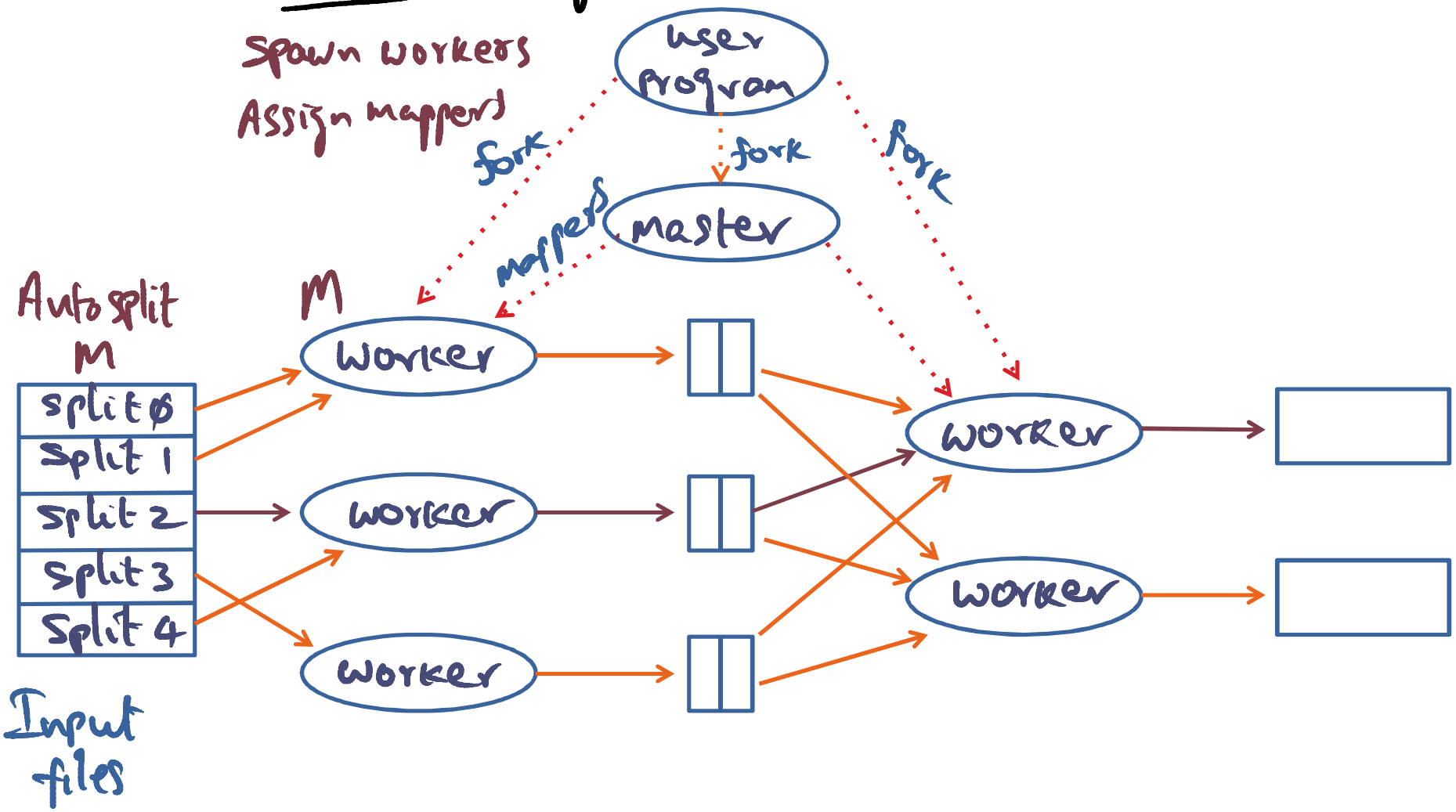
Heavy Lifting Done by the runtime



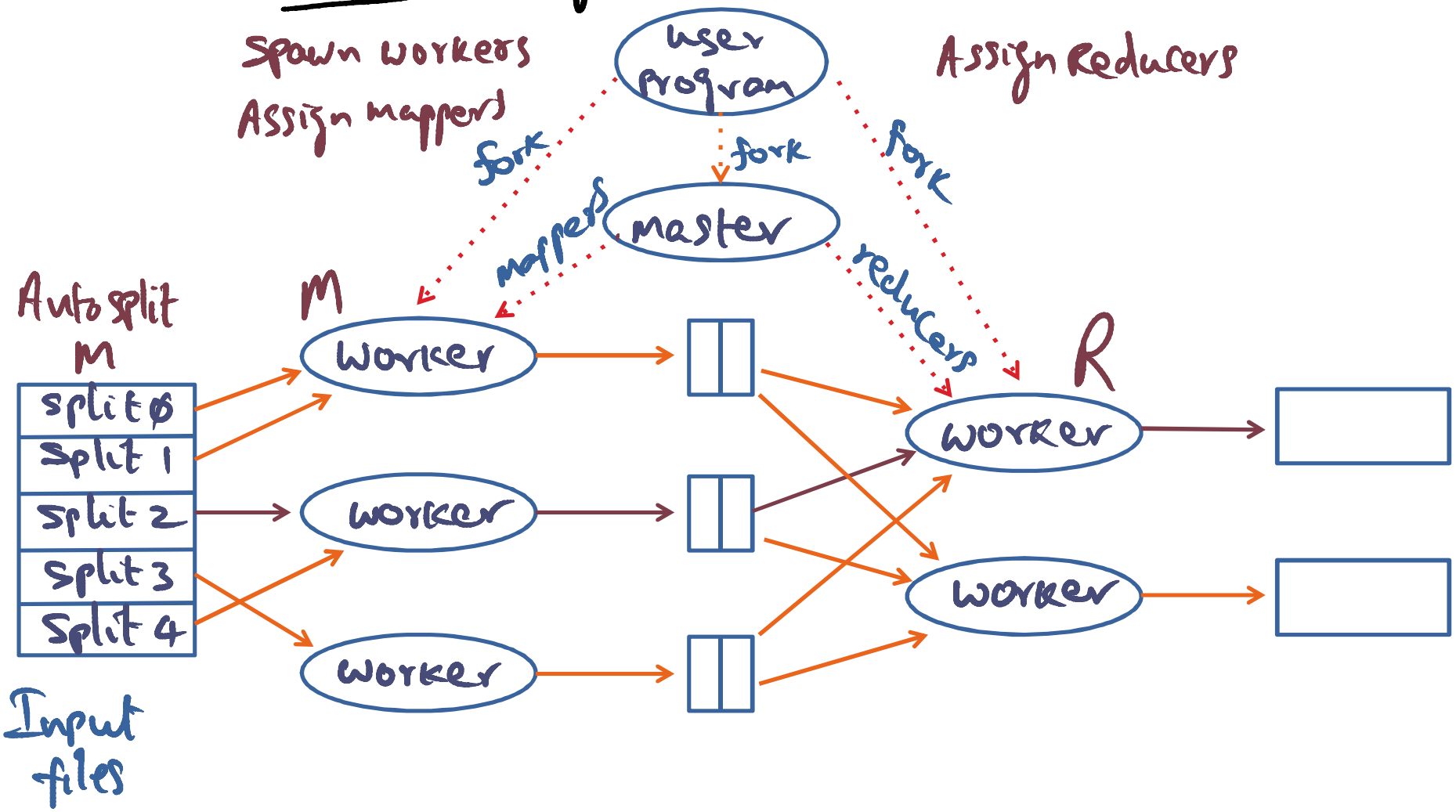
Heavy Lifting Done by the runtime



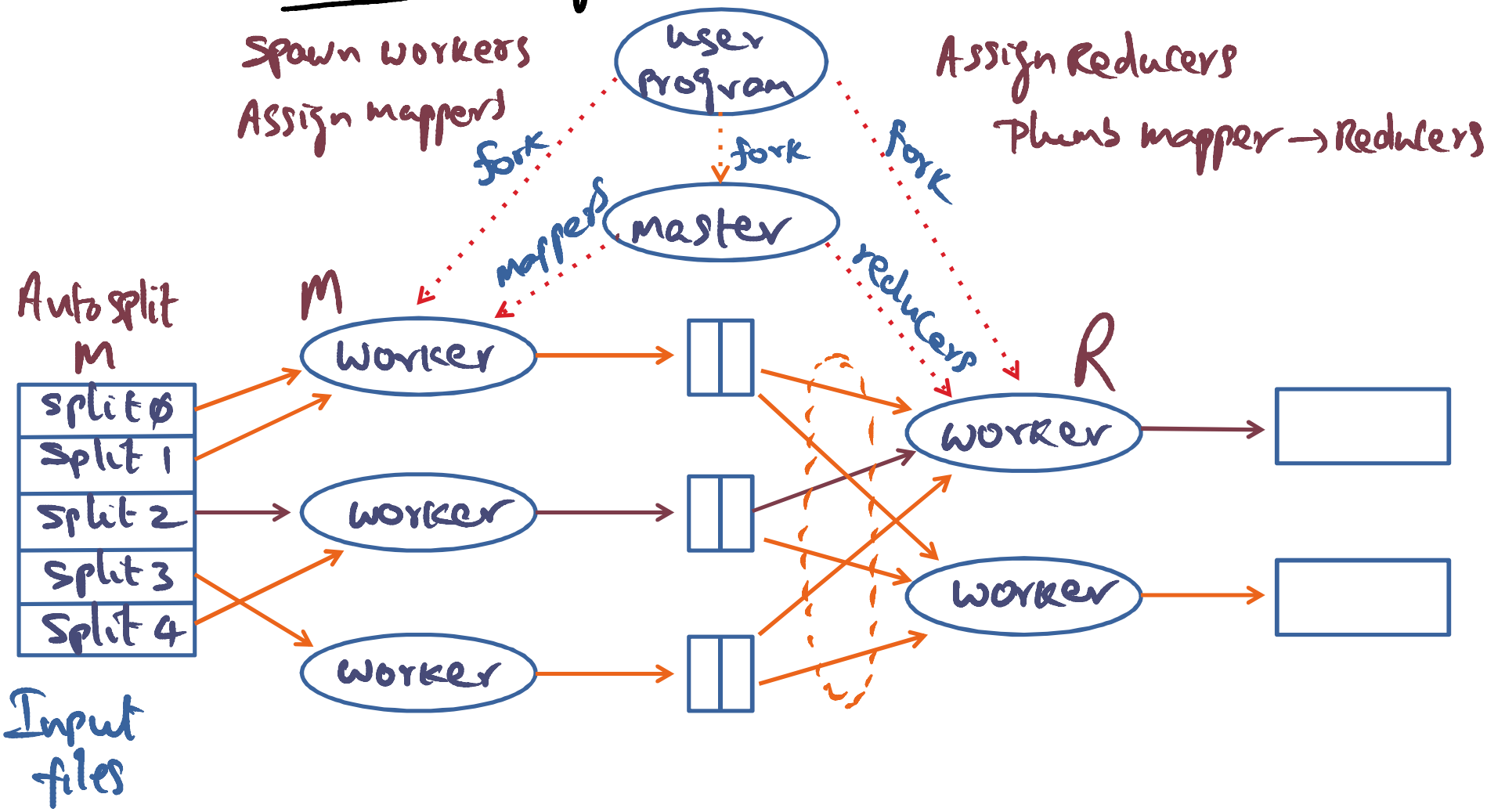
Heavy Lifting Done by the runtime



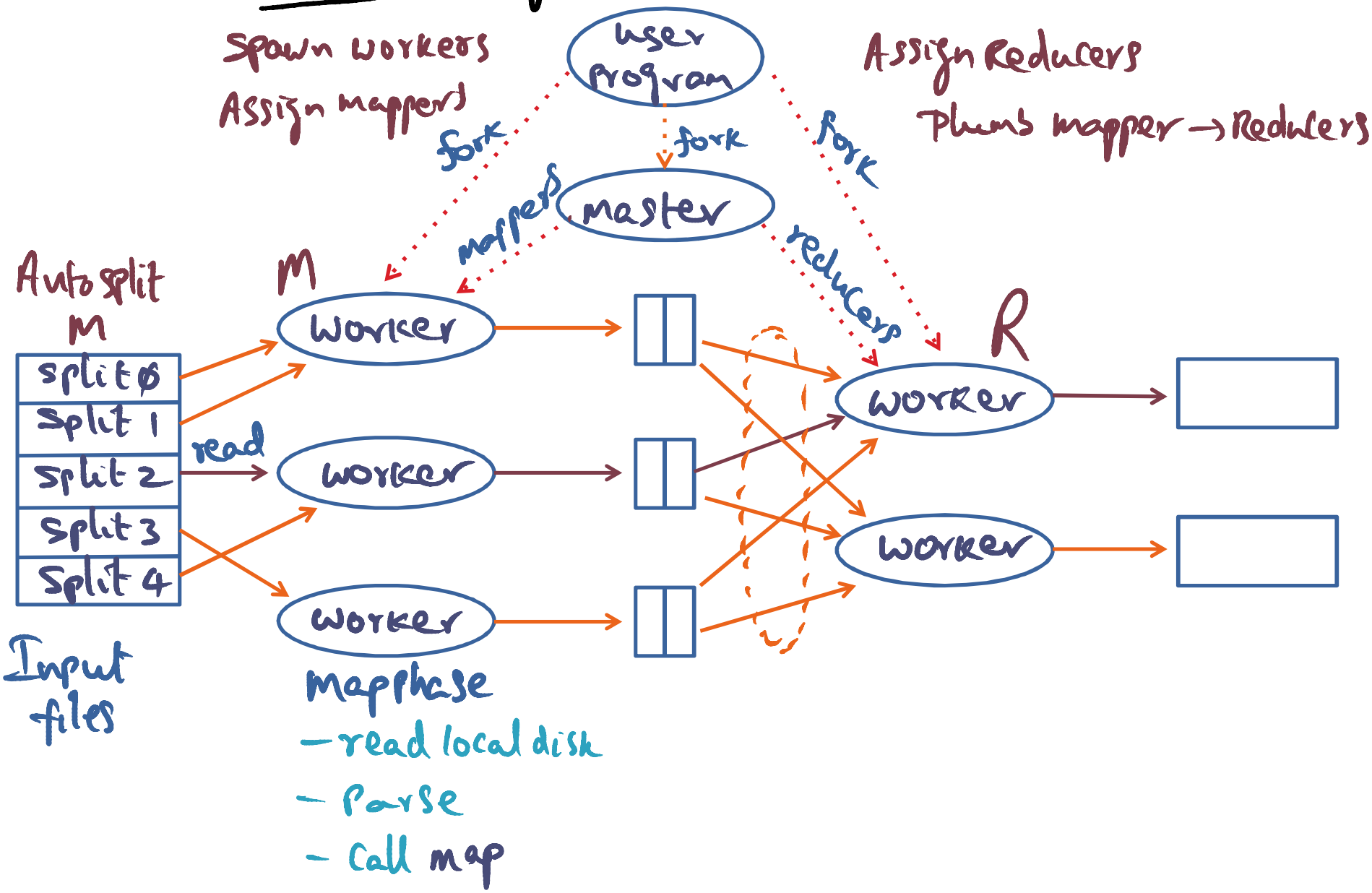
Heavy Lifting Done by the runtime



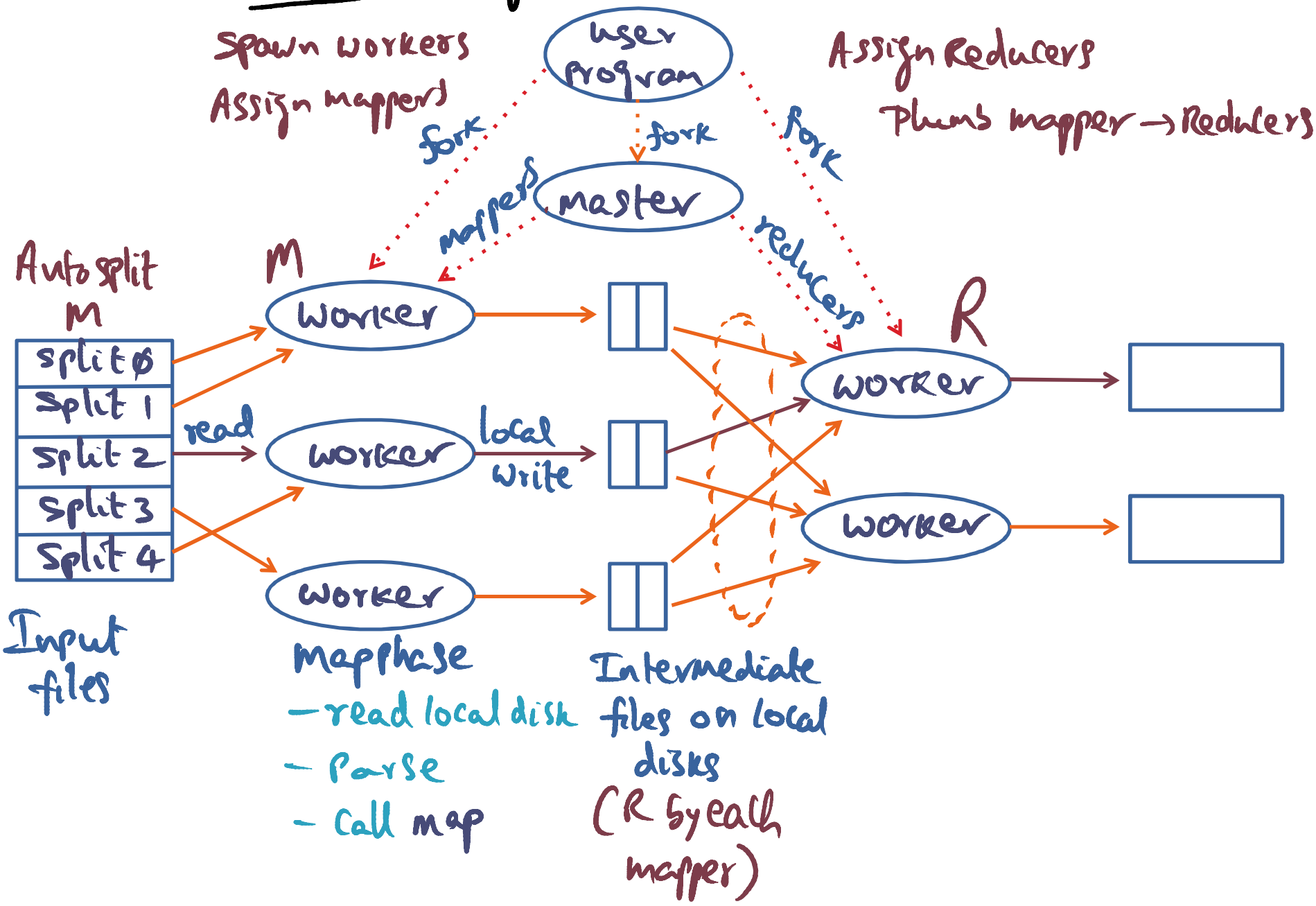
Heavy Lifting Done by the runtime



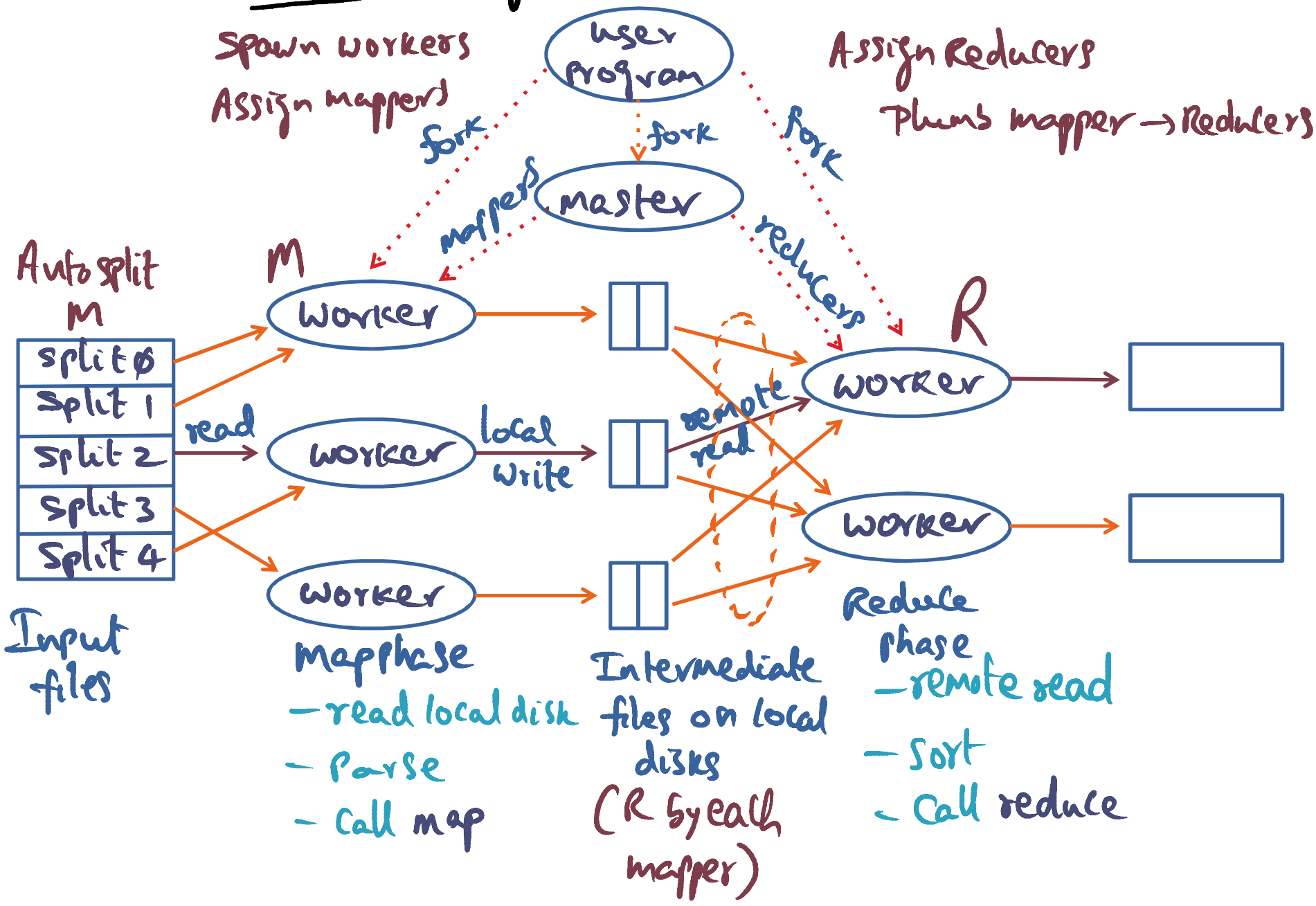
Heavy Lifting Done by the runtime



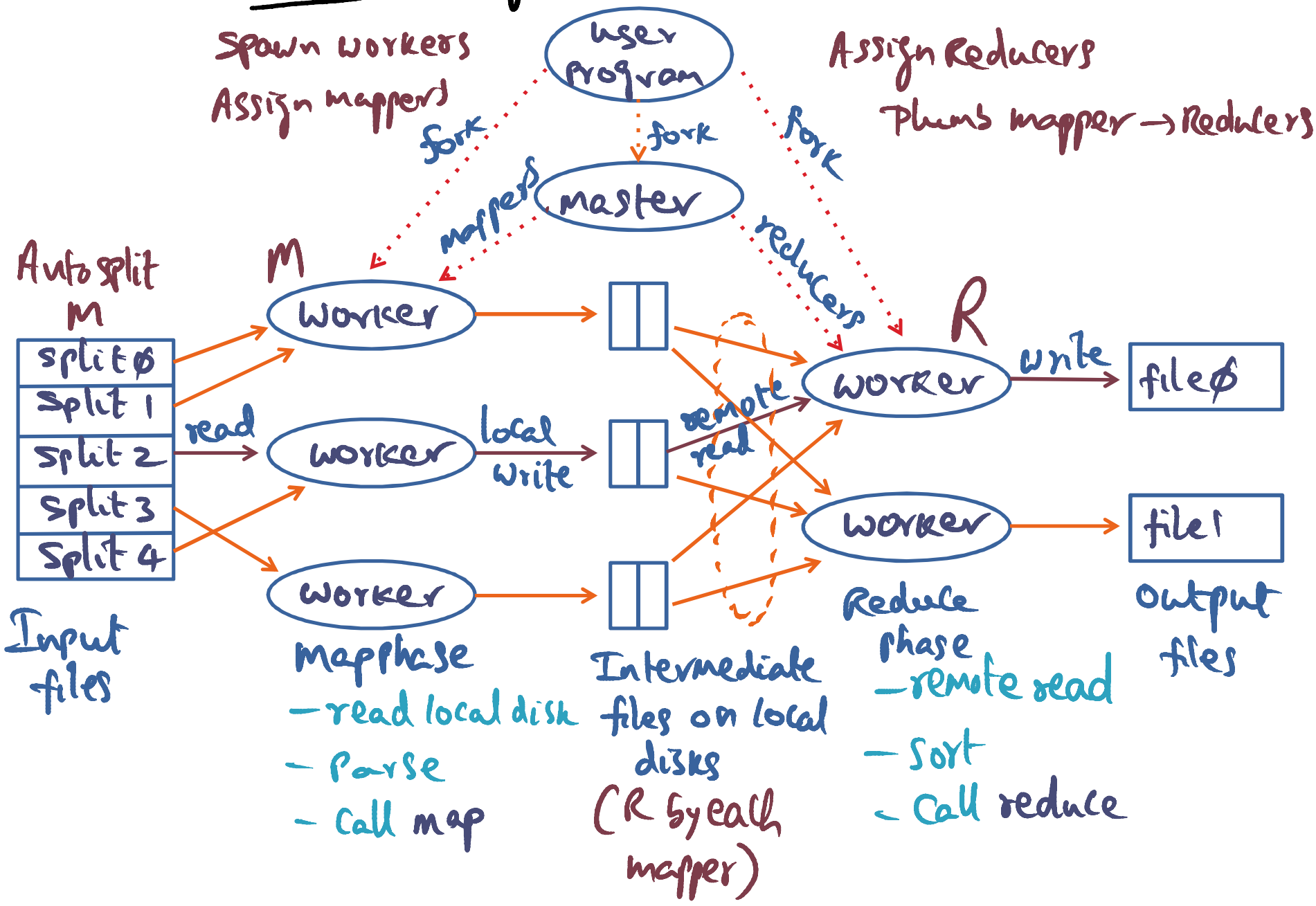
Heavy Lifting Done by the runtime



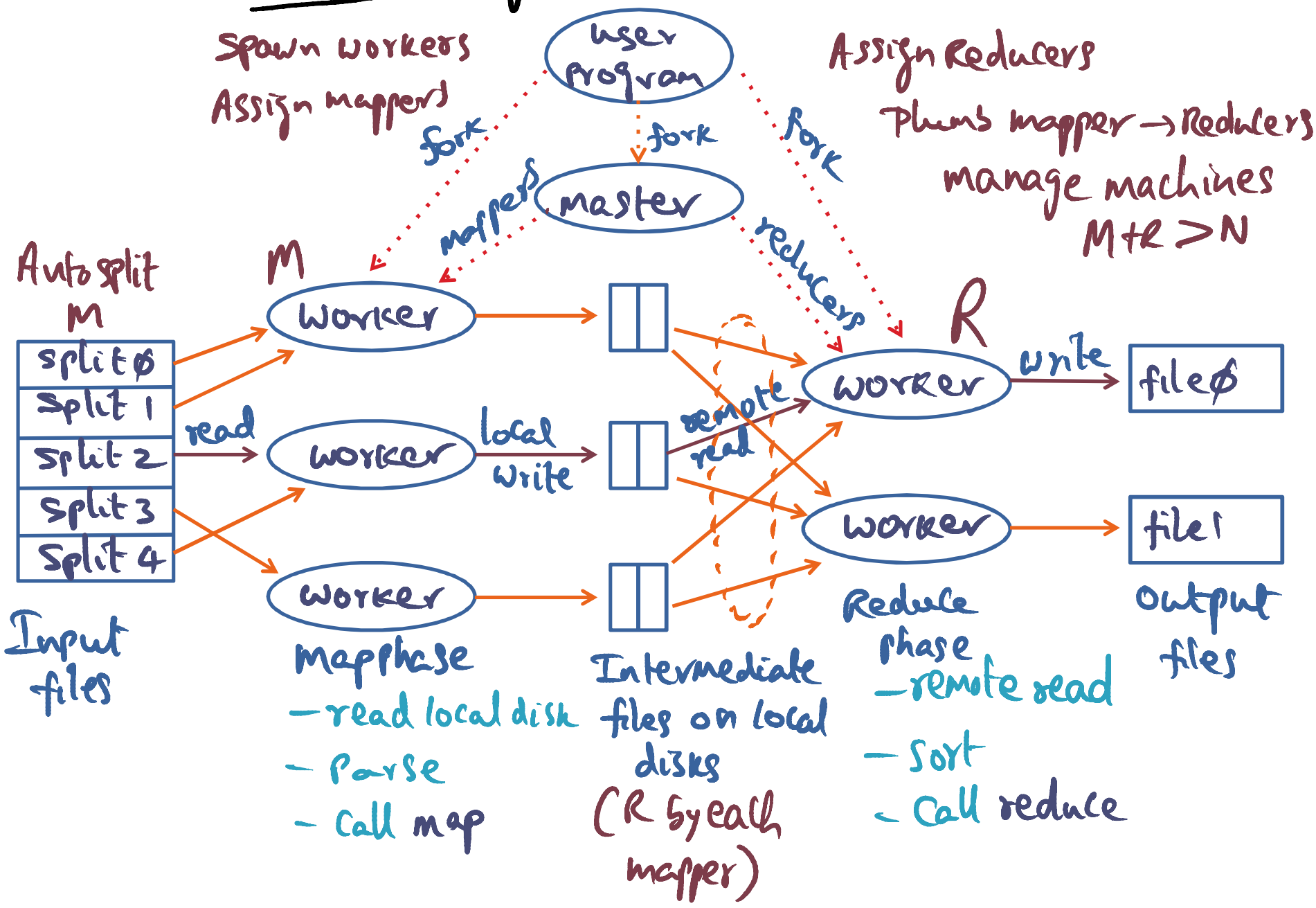
Heavy Lifting Done by the runtime



Heavy Lifting Done by the runtime



Heavy Lifting Done by the runtime



Issues to be handled by the runtime

Master data structures

- location of files created by completed mappers
- scoreboard of mapper/reducer assignment
- Fault tolerance
  - * start new instances if no timely response
  - * completion message from redundant stragglers
- Locality management
- Task granularity
- Backup tasks

Key takeaways

- map-reduce => simplicity
- domain expert's programming burden?
  - write "map" and "reduce" functions
- heavy lifting under the cover by the runtime system