

Some Applications of MapReduce (4)

Map task's output is sorted (e.g., quicksort)

Reduce task's input is sorted (e.g., mergesort)

Sort

- Input: Series of (key, value) pairs
- Output: Sorted <value>s

- Map – $\langle \text{key}, \text{value} \rangle \rightarrow \langle \text{value}, _ \rangle$ (*identity*)
- Reducer – $\langle \text{key}, \text{value} \rangle \rightarrow \langle \text{key}, \text{value} \rangle$ (*identity*)
- Partitioning function – partition keys across reducers based on **ranges** (can't use hashing!)
 - Take data distribution into account to balance reducer tasks