Q1. Write an in-mapper combiner **algorithm** for the “average problem”.

Q2. Assume that there are three datanodes and three reducers; one reducer in each datanode.

Also assume that there are six input splits:

 input split1 : [{cat eat rat cat} {rat sat bat fat}]

 input split2 : [{fat rat bat fat} {bat fat sat tat}]

 input split3 : [{cat bat rat mat} {sat mat pat mat}]

 input split4 : [{cat pat bat fat} {eat pat cat tat}]

 input split5 : [{tat eat pat sat} {bat sat cat mat}]

 input split6 : [{bat cat bat sat} {fat pat mat bat}]

and  input splits1 and 2 are in datanode 1,  input splits 3 and 4 are in datanode 2 and  input splits 5 and 6 are in datanode 3.

Further, let the partitioner  assign all words less than letter ‘k’ to Reducer 1,  all words greater than ‘r’ to Reducer 3 and  everything else to Reducer 2.

1. Illustrate word count algorithm (with **no** combiner, **no** in-mapper combining).
2. Illustrate word count algorithm (**no** combiner, with in-mapper combining).
3. How many key-value pairs are transported across the network in (a) and (b).

Top of Form

Bottom of Form

a)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Data Node 1 | | Data Node 2 | | Data Node 3 | |  |
| {cat eat rat cat}  {rat sat bat fat} | [{fat rat bat fat}  {bat fat sat tat} | {cat bat rat mat}  {sat mat pat mat} | {cat pat bat fat}  {eat pat cat tat}] | {tat eat pat sat}  {bat sat cat mat} | {bat cat bat sat}  {fat pat mat bat} | Data |
| (cat,1) (eat,1), (rat,1), (cat,1) | (fat,1), (rat,1), (bat,1), (fat,1) | (cat,1), (bat,1), (rat,1), (mat,1) | (cat,1), (pat,1), (bat,1), (fat,1) | (tat,1), (eat,1), (pat,1), (sat,1 | (bat,1), (cat,1), (bat,1), (sat,1 | 1st recordmappers output |
| (rat,1), (sat,1), (bat,1), (fat,1) | (bat,1), (fat,1), (sat,1), (tat,1) | (sat,1), (mat,1), (pat,1), (mat,1) | (eat,1), (pat,1),(cat,1), (tat,1) | (bat,1), (sat,1), (cat,1), (mat,1) | (fat,1), (pat,1),(mat,1), (bat,1) | 2nd record  mappers output |
| (bat, [1,1,1,1,1,1,1,1,1]), (cat, [1,1,1,1,1,1,1]), (eat, [1,1,1]), (fat, [1,1,1,1,1,1]) | | (mat, [1,1,1,1,1]), (pat, [1,1,1,1,1]), (rat, [1,1,1,1]) | | (sat, [1,1,1,1,1,1])  (tat, [1,1,1]) | | reducers Input |
| (bat,9), (cat,7), (eat,3), (fat,6 | | (mat,5), (pat,5), (rat,4 | | (sat,6), (tat,3 | | reducers output |

b)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Data Node 1 | | Data Node 2 | | Data Node 3 | |  |
| {cat eat rat cat}  {rat sat bat fat} | [{fat rat bat fat}  {bat fat sat tat} | {cat bat rat mat}  {sat mat pat mat} | {cat pat bat fat}  {eat pat cat tat}] | {tat eat pat sat}  {bat sat cat mat} | {bat cat bat sat}  {fat pat mat bat} | Data |
| (cat,2) (eat,1), (rat,2) (sat,1), (bat,1), (fat,1) | (fat,3), (rat,1), (bat,2) (sat,1), (tat,1) | (cat,1), (bat,1), (rat,1), (mat,3) (sat,1), (pat,1) | (cat,2), (pat,2), (bat,1), (fat,1) (eat,1), (tat,1) | (tat,1), (eat,1), (pat,1), (sat,2), (bat,1),(cat,1), (mat,1 | (bat,3), (cat,1), (sat,1 (fat,1), (pat,1),(mat,1) | mappers output |
|
| (bat, [1,2,1,1,1,3]), (cat, [2,1,2,1,1]), (eat, [1,1,1]), (fat, [1,3,1,1] | | (mat, [3,1,1]), (pat, [1,2,1,1]), (rat, [2,1,1] | | (sat, [1,1,1,2,1])  (tat, [1,1,1]) | | reducers Input |
| (bat,9), (cat,7), (eat,3), (fat,6 | | (mat,5), (pat,5), (rat,4 | | (sat,6), (tat,3 | | reducers output |

c)

- in a: **27**

- in b: **23**