Ayalew Getachew

ID 985644

Lab5

Questions (For reference )

**Assignment Instructions**

Q1. Write an in-mapper combiner **algorithm** modifying algorithm 3.8 (That is, pairs approach)

Q2. Write an in-mapper combiner **algorithm** modifying algorithm 3.9 (That is, stripes approach)

Q3. Assume that there are two input spits and two reducers. Note that Mapper 1 and Reducer 1 run on the same machine. Mapper 2 and Reducer 2 run on the same machine.

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

Input Split 1 : [ {cat mat rat, cat}, {cat  bat cat pat},{cat bat rat bat}]    (Note : 3 records)

Input Split 2 : [{cat rat bat rat}, {bat mat pat bat}, {pat cat bat mat}]    (Note: 3 records)

**Let the neighborhood of X, N(X) be set of all term after X and before the next X.**

Example: Let Data block be [a b c a d e]

N(a) = {b, c}, N(b) = {c, a, d, e}, N(c) = {a, d, e}, N(a) ={d, e}, N(d) = {e}, N(e) = {}.

1. Illustrate algorithm 3.8.
2. Illustrate algorithm 3.8 (with in-mapper combining. That is, apply your algorithm Q1).
3. Illustrate algorithm 3.9
4. Illustrate algorithm 3.9 (with in-mapper combining. That is, apply your algorithm Q2).

Solution

Question 1) Write an in-mapper combiner algorithm modifying algorithm 3.8 (That is, pairs approach)

class Mapper

Method Initialize

H = new AssociativeArray()

Method Map(docid a; doc d)

for all term w in doc d do

for all term u in Neighbors(w) do

H{pair(w;u)} ++

Method close

Emit(pair (w; u); H{pair(w;u)})

class Reducer

method Reduce(pair p; counts [c1; c2; …])

s = 0

for all count c in counts [c1; c2; …] do

s = s + c .

Emit(pair p; count s)

Question 2) Write an in-mapper combiner algorithm modifying algorithm 3.9 (That is, stripes approach)

class Mapper

Method Initialize

H = new AssociativeArray()

Method Map(docid a; doc d)

for all term w in doc d do

for all term u in Neighbors(w) do

H{w}{u} = H{w}{u} + 1 .

Method close

Emit(Term w; Stripe H)

class Reducer

method Reduce(term w; stripes [H1;H2;H3; : : :])

Hf = new AssociativeArray

for all stripe H in stripes [H1;H2;H3; …] do

Sum(Hf; H)

Emit(term w; stripe Hf )

Illustrate algorithm 3.8.

|  |  |  |
| --- | --- | --- |
| Mapper 1 | Mapper 2 |  |
| {cat mat rat, cat},  {cat bat cat pat},  {cat bat rat bat}, | {cat rat bat rat},  {bat mat pat bat},  {pat cat bat mat}, | Mapper Input |
| ((cat, mat), 1)  ((cat, rat), 1)  ((mat, rat), 1)  ((mat, cat), 1)  ((rat, cat), 1)  ((cat, bat), 1)  ((bat, cat), 1)  ((bat, pat), 1)  ((cat, pat), 1)  ((cat, bat), 1)  ((cat, rat), 1)  ((cat, bat), 1)  ((bat, rat), 1)  ((rat, bat), 1) | ((cat, rat), 1)  ((cat, bat), 1)  ((cat, rat), 1)  ((rat, bat), 1)  ((bat, rat), 1)  ((bat, mat), 1)  ((bat, pat), 1)  ((mat, pat), 1)  ((mat, bat), 1)  ((pat, bat), 1)  ((pat, cat), 1)  ((pat, bat), 1)  ((pat, mat), 1)  ((cat, bat), 1)  ((cat, mat), 1)  ((bat, mat), 1) | Mapper Output |
| Reducer 1 | Reducer 2 |  |
| ((cat, bat), [1, 1, 1, 1, 1])  ((cat, mat), [1, 1])  ((cat, pat), [1])  ((cat, rat), [1, 1, 1, 1])  ((bat, cat), [1])  ((bat, mat), [1,1])  ((bat, pat), [1, 1])  ((bat, rat), [1, 1]) | ((mat, bat), [1]) ((mat, cat), [1]) ((mat, pat), [1]) ((mat, rat), [1])  ((pat, bat), [1])  ((pat, cat), [1])  ((pat, mat, [1]))  ((rat, bat), [1, 1]) ((rat, cat), [1]) | Reducer Input |
|  |  |  |
| ((cat, bat), 5)  ((cat, mat), 2)  ((cat, pat), 1)  ((cat, rat), 4)  ((bat, cat), 1)  ((bat, mat), 2)  ((bat, pat), 2)  ((bat, rat), 2) | ((mat, bat), 1)  ((mat, cat), 1)  ((mat, pat), 1)  ((mat, rat), 1)  ((pat, bat), 1)  ((pat, cat), 1)  ((pat, mat, 1))  ((rat, bat), 2)  ((rat, cat), 1) | Reducer Output |

Illustrate algorithm 3.8 (with in-mapper combining. That is, apply your algorithm Q1).

|  |  |  |
| --- | --- | --- |
| Mapper 1 | Mapper 2 |  |
| {cat mat rat, cat},  {cat bat cat pat},  {cat bat rat bat}, | {cat rat bat rat},  {bat mat pat bat},  {pat cat bat mat}, | Mapper Input |
| ((cat, mat), 1)  ((cat, rat), 2)  ((mat, rat), 1)  ((mat, cat), 1)  ((rat, cat), 1)  ((cat, bat), 3)  ((bat, cat), 1)  ((bat, pat), 1)  ((cat, pat), 1)  ((bat, rat), 1)  ((rat, bat), 1) | ((cat, rat),2)  ((cat, bat), 2)    ((rat, bat), 1)  ((bat, rat), 1)  ((bat, mat), 2)  ((bat, pat), 1)  ((mat, pat), 1)  ((mat, bat), 1)    ((pat, cat), 1)  ((pat, bat), 2)  ((pat, mat), 1)    ((cat, mat), 1) | Mapper Output |
| Reducer 1 | Reducer 2 |  |
| ((cat, bat), [3,2])  ((cat, mat), [1, 1])  ((cat, pat), [1])  ((cat, rat), [2,2])  ((bat, cat), [1])  ((bat, mat), [2])  ((bat, pat), [1, 1])  ((bat, rat), [1, 1]) | ((mat, bat), [1]) ((mat, cat), [1]) ((mat, pat), [1]) ((mat, rat), [1])  ((pat, bat), [1])  ((pat, cat), [1])  ((pat, mat, [1]))  ((rat, bat), [1, 1]) ((rat, cat), [1]) | Reducer Input |
|  |  |  |
| ((cat, bat), 5)  ((cat, mat), 2)  ((cat, pat), 1)  ((cat, rat), 4)  ((bat, cat), 1)  ((bat, mat), 2)  ((bat, pat), 2)  ((bat, rat), 2) | ((mat, bat), 1)  ((mat, cat), 1)  ((mat, pat), 1)  ((mat, rat), 1)  ((pat, bat), 1)  ((pat, cat), 1)  ((pat, mat, 1))  ((rat, bat), 2)  ((rat, cat), 1) | Reducer Output |

Illustrate Algorithm 3.9

|  |  |  |
| --- | --- | --- |
| Mapper 1 | Mapper 2 |  |
| {cat mat rat, cat},  {cat bat cat pat},  {cat bat rat bat}, | {cat rat bat rat},  {bat mat pat bat},  {pat cat bat mat}, | Mapper Input |
| (cat, [(mat, 1), (rat, 1)])  (mat,[(rat, 1), (cat, 1)])  (rat, [(cat, 1)]) (cat, [(bat, 1)])  (bat, [(cat, 1), (pat, 1)])  (cat, [(pat,1)]) (cat, [(bat, 2),  (rat, 1)]) (bat, [(rat, 1)]) (rat, [(bat, 1)]) | (cat, [(rat,2), (bat,1)]) (rat, [(bat,1)]) (bat, [(rat,1)])  (bat, [(mat,1), (pat, 1)])  (mat, [(pat,1), (bat, 1)])  (pat, [(bat, 1)])  (pat, [(cat,1), (bat,1), (mat,1)]) (cat, [(bat,1), (mat,1)])  (bat, [(mat,1)]) | Mapper Output |
| Reducer 1 | Reducer 2 |  |
| (bat,[ [(cat, 1), (pat, 1)], [(rat, 1)], [(rat,1)], [(mat,1), (pat, 1)], [(mat,1)] ])  (cat,[ [(mat, 1), (rat, 1)], [(bat, 1)], [(pat,1)], [(bat, 2), (rat, 1)], [(rat,2), (bat,1)], [(bat,1), (mat,1)] ]) | (mat,[ [(rat, 1), (cat, 1)], [(pat,1), (bat, 1)] ])  (pat, [ [(bat, 1)], [(cat,1), (bat,1), (mat,1)] ])  (rat,[ [(cat, 1)], [(bat, 1)], [(bat, | Reducer Input |
|  |  |  |
| (bat,[ (cat,1), (pat,2), (rat,2),(mat,2) ])  (cat,[ (mat,2),(rat,4),(bat,5),(pat,1) ]) | (mat,[ (rat,1), (cat,1), (pat,1),(bat,1) ]) (pat,[ (bat,2),(cat,1),(mat,1) ]) (rat,[ (cat,1),(bat,2) ]) | Reducer Output |

Illustrate Algorithm 3.9 with In Mapper

|  |  |  |
| --- | --- | --- |
| Mapper 1 | Mapper 2 |  |
| {cat mat rat, cat},  {cat bat cat pat},  {cat bat rat bat}, | {cat rat bat rat},  {bat mat pat bat},  {pat cat bat mat}, | Mapper Input |
| (cat, [(mat, 1), (rat, 2), (bat, 3),(pat,1)]) (mat,[(rat, 1), (cat, 1)]) (rat, [(cat, 1),  (bat, 1)]) (bat, [(cat, 1), (pat, 1), (rat, 1)]) | (cat, [(rat,2), (bat,2), (mat,1)]) (rat, [(bat,1)]) (bat, [(rat,1),(mat,2), (pat, 1)]) (mat, [(pat,1), (bat, 1)]) (pat, [(bat, 2),(cat,1), (mat,1)]) | Mapper Output |
| Reducer 1 | Reducer 2 |  |
| (bat,[ [(cat, 1), (pat, 1), (rat, 1)], [(rat,1),(mat,2), (pat, 1)] ])  (cat,[ [(mat, 1), (rat, 2), (bat, 3), (pat,1)], [(rat,2), (bat,2), (mat,1)] ]) | (mat,[ [(rat, 1), (cat, 1)], [(pat,1), (bat, 1)] ])  (pat, [ [(bat, 2),(cat,1), (mat,1)] ])  (rat,[ [(cat, 1),(bat, 1)], [(bat,1)] ]) | Reducer Input |
|  |  |  |
| (bat,[ (cat,1), (pat,2), (rat,2),(mat,2) ]) (cat,[ (mat,2),(rat,4),(bat,5),(pat,1) ]) | (mat,[ (rat,1), (cat,1), (pat,1),(bat,1) ]) (pat,[ (bat,2),(cat,1),(mat,1) ]) (rat,[ (cat,1),(bat,2) ]) | Reducer Output |