MapReduce: Page Rank Performance Analysis

1. Iteration time Analysis-
    - Below are the times taken per epoch and prior initialization for a graph with 10,000 nodes and 100,000 edges-

      Global Initialization: 0.2238s
      Epoch: 1
      Read time: 0.005483s
      Write time: 0.4192s
      Init time: 0.2238s
      Total time: 2.654s

      Epoch: 2
      Read time: 0.005674s
      Write time: 0.426s
      Init time: 0.2238s
      Total time: 2.656s

      Epoch: 3
      Read time: 0.005665s
      Write time: 0.4259s
      Init time: 0.2238s
      Total time: 2.763s

      Epoch: 4
      Read time: 0.005756s
      Write time: 0.4231s
      Init time: 0.2238s
      Total time: 2.639s

      Epoch: 5
      Read time: 0.008487s
      Write time: 0.4248s
      Init time: 0.2238s
      Total time: 2.676s

      Epoch: 6
      Read time: 0.005923s
      Write time: 0.4206s
      Init time: 0.2238s
      Total time: 2.657s

Epoch: 7
Read time: 0.005777s
Write time: 0.4248s
Init time: 0.2238s
Total time: 2.669s

Epoch: 8
Read time: 0.005887s
Write time: 0.4237s
Init time: 0.2238s
Total time: 2.676s

Epoch: 9
Read time: 0.005627s
Write time: 0.4234s
Init time: 0.2238s
Total time: 2.657s

Epoch: 10
Read time: 0.008176s
Write time: 0.4194s
Init time: 0.2238s
Total time: 2.593s

Total time: 26.63838505744934
Average Time: 2.663838505744934

- As seen in the next section, the time taken per epoch scales linearly with V but sub linearly with E. So calculating on the basis of V->
    - V = (600/26.6)*10,000 = approx 222,222 nodes for 10 epochs

2. Scalability Analysis-
    - Expectation: Page rank is O(epoch * (V+E))
    - Scale with number of edges (V = 10,000) -
        - E = 100,000 ->
            - Total time: 26.992541551589966
            - Average Time: 2.6992541551589966
        - E = 10,000 ->
            - Total time: 21.533185243606567
            - Average Time: 2.1533185243606567
        - E - 1,000,000->
            - Total time: 29.375189781188965

        - Average Time: 2.9375189781188964
    - We can see that in our implementation the change in the number of edges does not make as big of a difference as expected. This is because the addition of extra edges is only a factor in the reducer that handles them privately and only passes the final value to the answer. Because of this the increase is not linear as expected.
- Scale with number of vertices(E = 100,000)-
    - V = 1,000 ->
        - Total time: 3.2159156799316406
        - Average Time: 0.3215915679931641
    - V = 5,000 ->
        - Total time: 13.439592838287354
        - Average Time: 1.3439592838287353
    - V = 10,000 ->
        - Total time: 26.992541551589966
        - Average Time: 2.6992541551589966
    - As expected, the time increase is linear(1,000->5,000 = approx 5x time increase, 5,000->10,000 = approx 2x increase)

3. I/O Performance-
    - In the case shown above(V = 10,000, E=100,000)
        - Avg read time: 0.006245
        - Avg write time: 0.42309
        - We can see read time is inconsequential but write time accounts for 15-16% of the total epoch time.
    - Total written data = 1.5 Mb