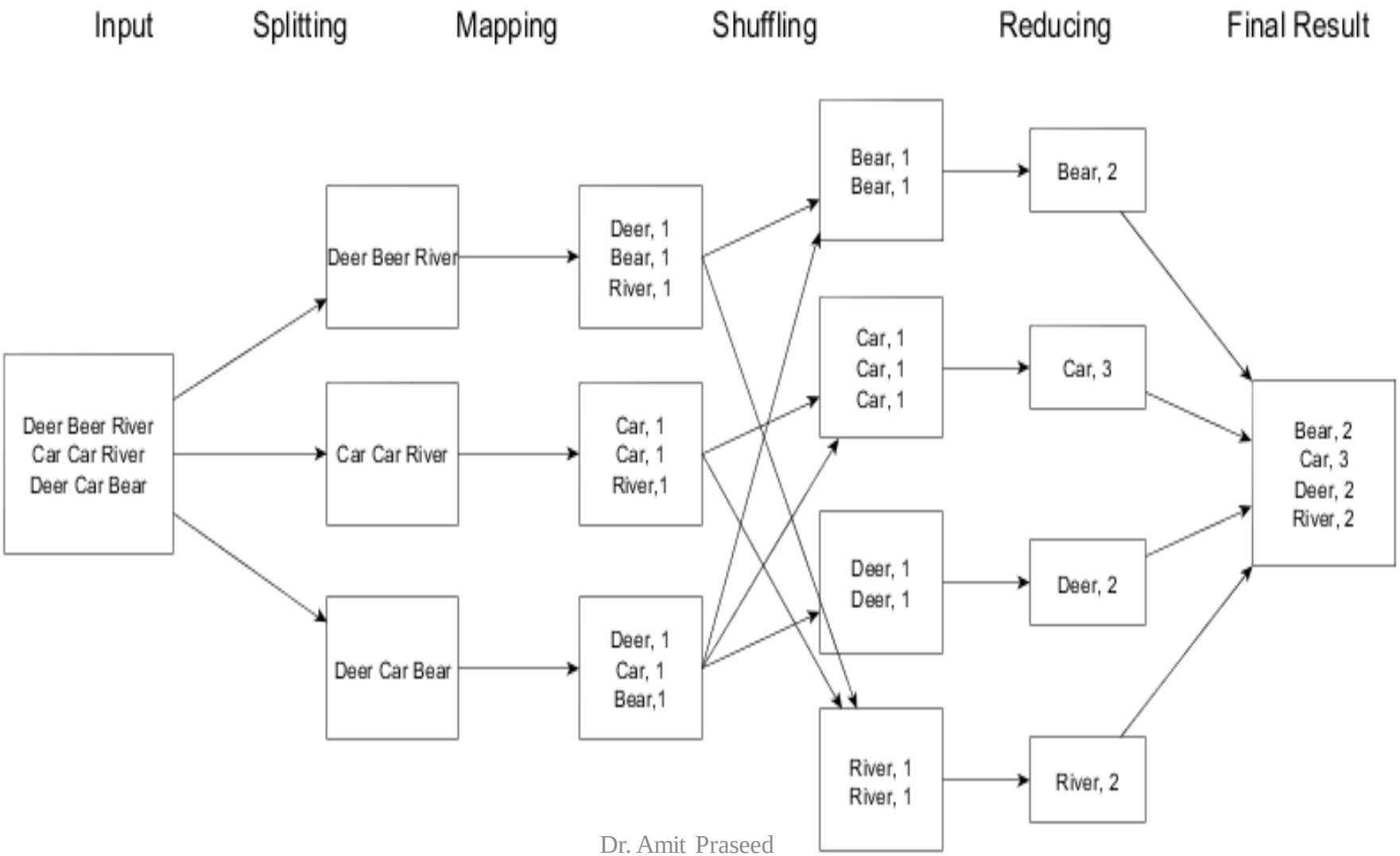
Introduction to MapReduce

Dr. Amit Praseed

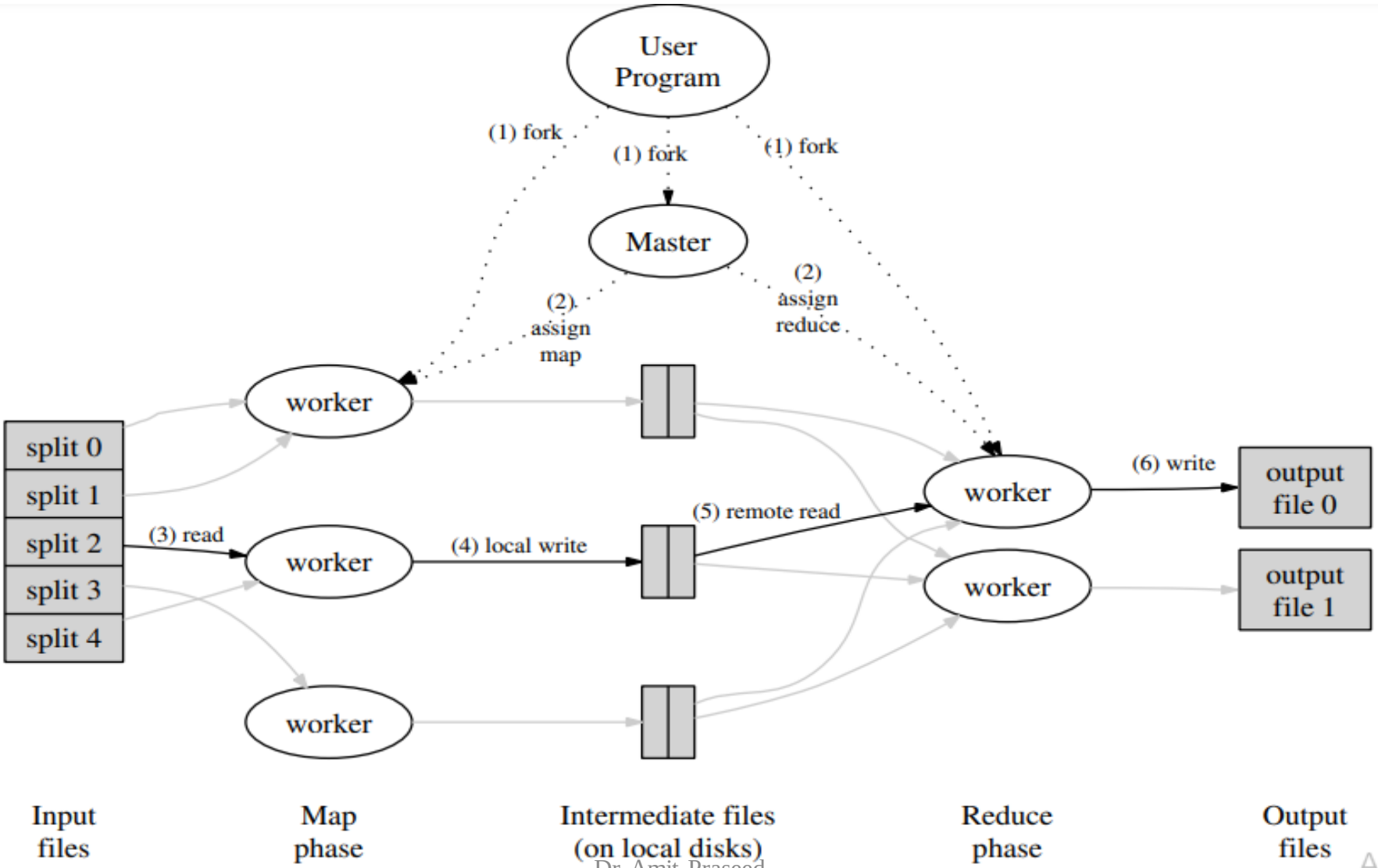
What is MapReduce?

- MapReduce is a software framework for processing large datasets in a distributed fashion
  - Map data into multiple <key, value> pairs
  - Reduce over all keys having the same data
- Consider a simple example of finding the count of every word present in a document

Word Count using MapReduce



MapReduce Workflow



Handling Node Failures

- Worker Node Failure
  - Any map task or reduce task in progress on a failed worker is set to idle and rescheduled
  - Any completed map task is also restarted, as output is stored on local disk
  - Completed reduce tasks don’t need to be restarted as output is written to a global file
- Master Node Failure
  - Periodic check pointing
  - Restart on a different master

Handling Stragglers

- Straggler : a machine that takes an unusually long time to complete one of the last few map or reduce tasks in the computation
  - a machine with a bad disk may experience slow read performance
  - Other scheduled other tasks on the machine
- When a MapReduce operation is close to completion
  - The master schedules backup executions of the remaining in-progress tasks
  - The task is marked as completed whenever either the primary or the backup execution completes

PageRank using MapReduce

- PageRank is the magic algorithm behind Google search results
- Every web site is awarded a rank based on certain features, most notable of which is the number of links from other sites to itself
- This algorithm needs to be executed on a large amount of data – hence MapReduce is a good option

Random Surfer Model

- Intuition for PageRank
- Imagine a surfer who starts on a randomly chosen page and then follows outgoing links at random
  - Markov Process
- PageRank is probability that user will arrive at a given page during this random walk

A little more complex!

- Model assumes that surfer doesn't always follow a link, but sometimes e.g. bookmarks instead.
- Before each move, surfer flips a coin
  - With probability α, follows an out-link
  - With probability 1- α, moves to a (uniformly chosen) random page

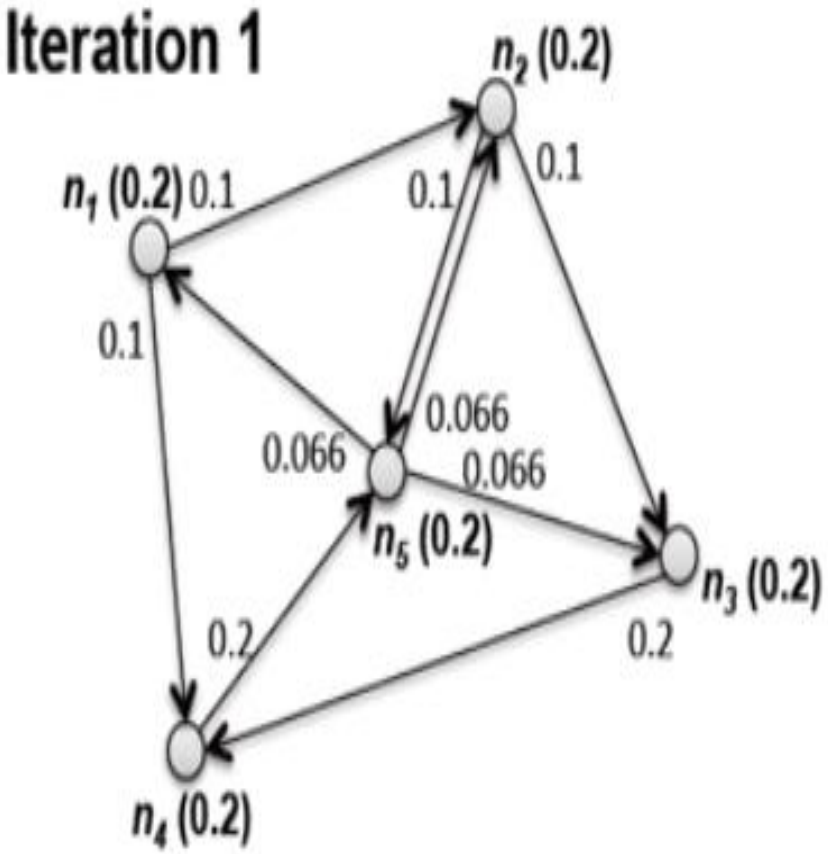
Simplified Page Rank

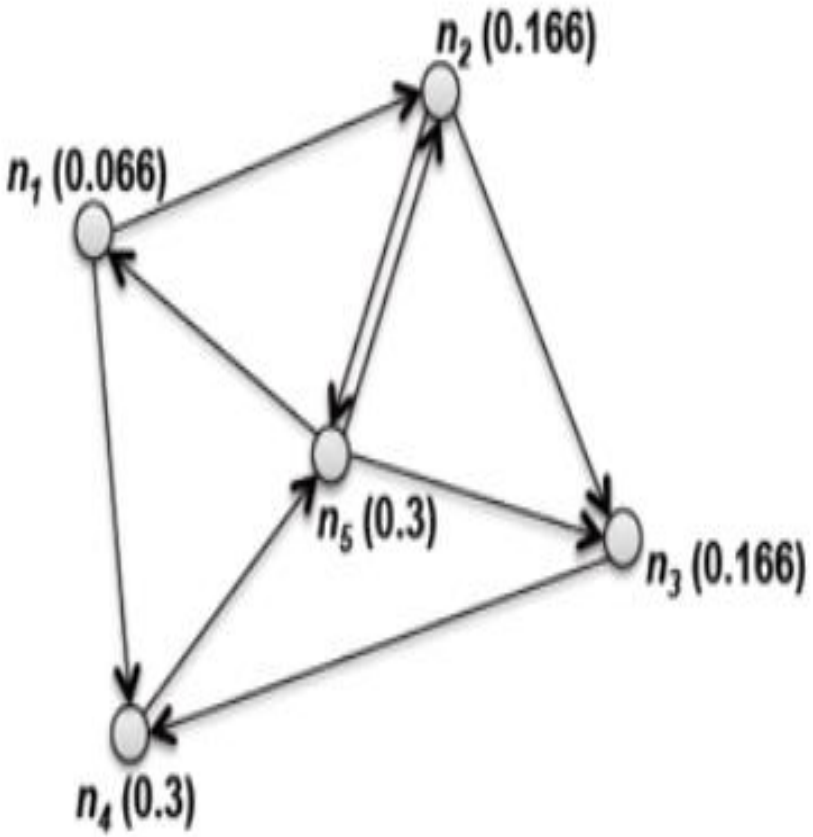
- PageRank of a page is based on the number of pages which link to it
  - But what if a page owner creates a swarm of dummy pages pointing to his website?
- PageRank of a page is based on the PageRank of the pages which link to it
  - A page divides its PageRank equally among all its outgoing links

$$P(n) = \sum_{m \in L(n)} \frac{P(m)}{C(m)}$$

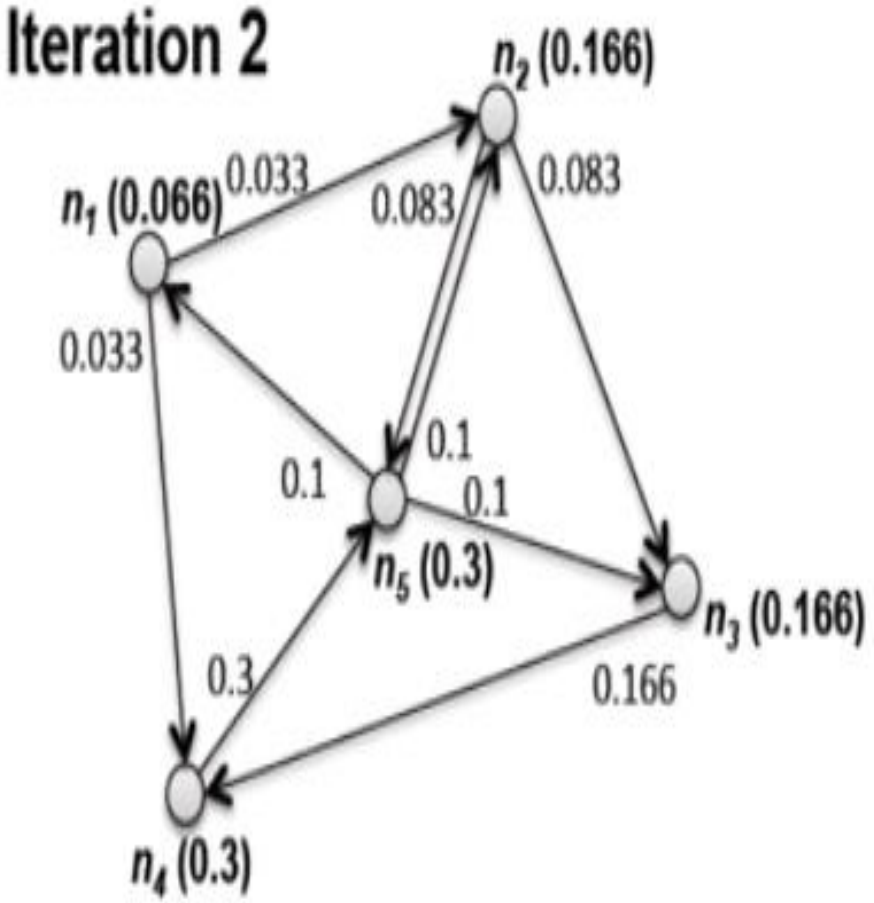
L(n) is the pages connected to n and C(m) is the number of pages that m is connected to

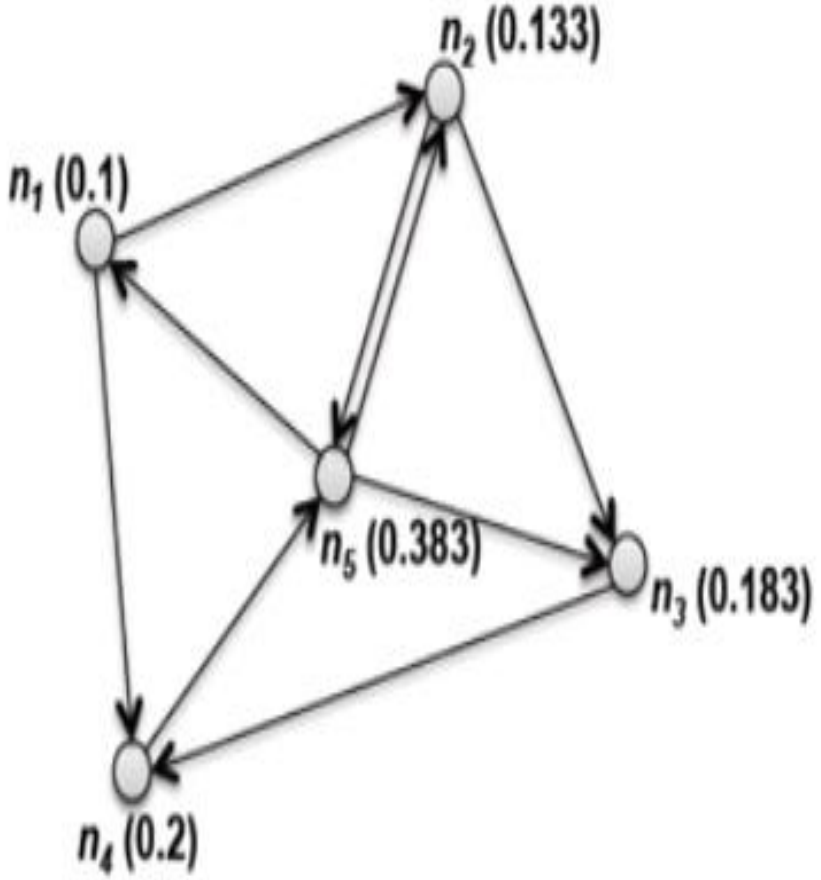
Example


Iteration 1
n_2 (0.2)
n_1 (0.2)
0.1
0.1
0.1
0.1
0.066
0.066
0.066
n_5 (0.2)
n_3 (0.2)
0.2
0.2
n_4 (0.2)


n_2 (0.166)
n_1 (0.066)
n_5 (0.3)
n_3 (0.166)
n_4 (0.3)

Example


Iteration 2
n_2 (0.166)
n_1 (0.066)
0.033
0.083
0.083
0.033
0.1
0.1
0.1
n_5 (0.3)
n_3 (0.166)
0.3
0.166
n_4 (0.3)


n_2 (0.133)
n_1 (0.1)
n_5 (0.383)
n_3 (0.183)
n_4 (0.2)

PageRank using MapReduce

- Full algorithm is iterative
- Initialize the nodes to uniform distribution
- Run the two MR jobs described iteratively
  - Map job emits page rank distributions continuously
  - Reduce job constantly collects these emitted page ranks and combines them for every node
- Until convergence (no change)
- At the end there is a clean up/normalization phase to correct the distributions

$$p' = \alpha \frac{1}{|G|} + (1 - \alpha)\left(\frac{m}{|G|} + p\right)$$