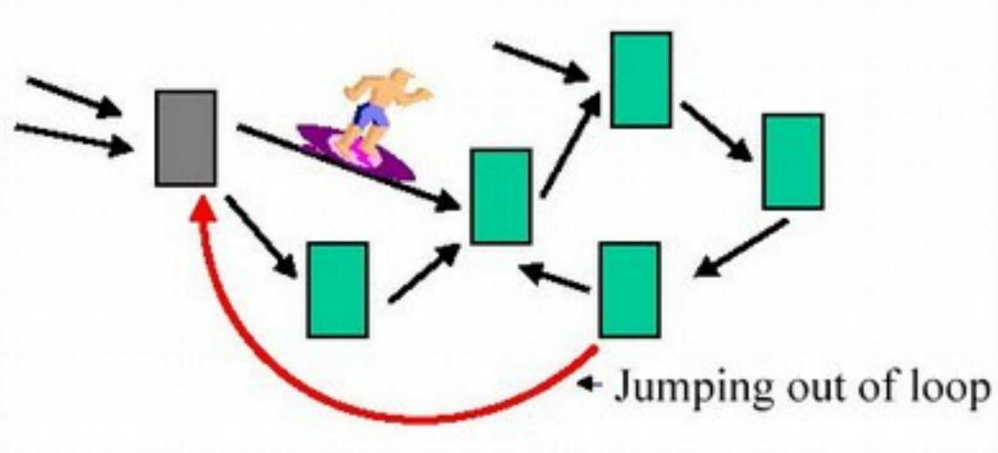
Example 2: PageRank

- Input: pages with output links
- Output: ranking of page
- Based on the random surfer model
  - The page rank is the probability to reach a page by:
    - Randomly clicking on links
    - Sometimes the user input an URL directly: put a small probability on this

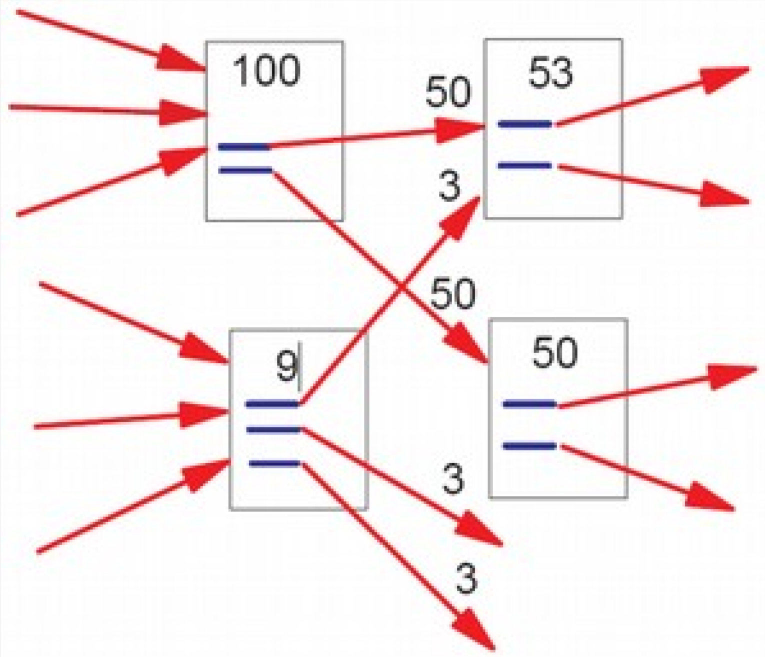


PageRank Formula

- $Rk(P) = (1\text{-}d)/n + d\,[\,Rk(T_1)\,/\,out(T_1) + \cdots + Rk(T_n)\,/\,out(T_n)\,]$

  with :

  - n: number of pages
  - P: ranked page
  - d: damping factor (usually 0.85)
  - $T_1,\ldots,T_n$: pages pointing to P
  - $out(T_i)$: number of output links in T_i



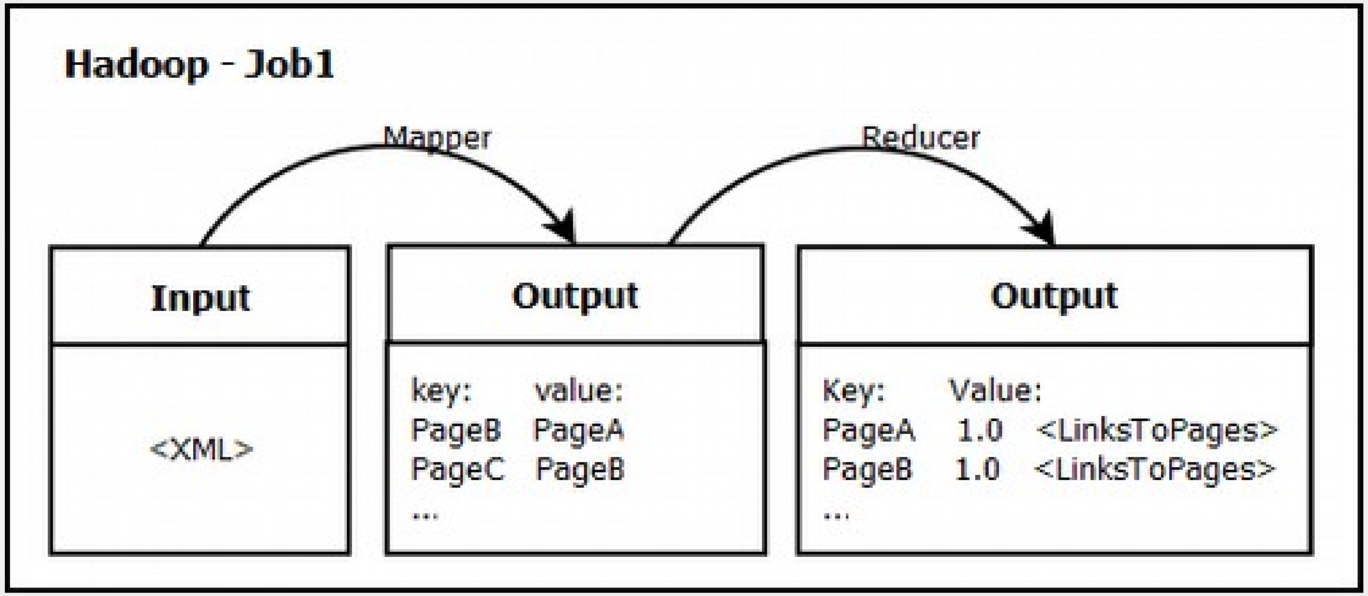
- Iterate: $Rk^{i+1}(P)$ computed from $Rk^i(T_n)$
  - $Rk^0(P) =1$
- Stops when all the pageranks are stable during two iterations

PageRank stage 1

- Extract links from XML pages

  map: (pageid, body) → (pageid, (Rk^{0}, list(pageid)))

  reduce: identity



Input

<XML>

Output

key:	value:
PageB	PageA
PageC	PageB
...	

Output

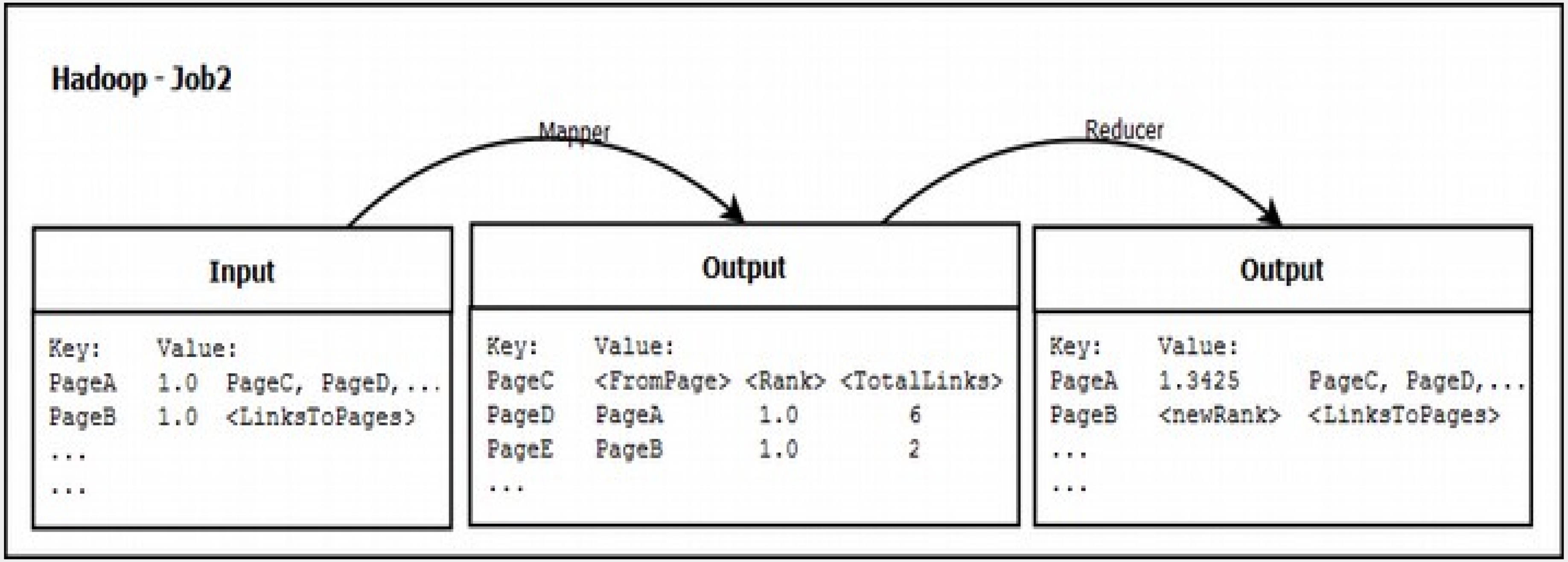
Key:	Value:	
PageA	1.0	<LinksToPages>
PageB	1.0	<LinksToPages>
...		

PageRank stage 2

- Iterates on page ranks computation

map^i: $(\text{pid}, (Rk^i,\text{list(pid)})) \rightarrow (\text{pid}, (Rk^i,\text{list(pid)}))$ and
listof $(\text{pid}, (\text{out}^i,\text{list()}))$

reduce^i: $(\text{pid}, (Rk^i ,\text{list(pid)}))$ and listof $(\text{pid}, (\text{out}^i,\text{list()}))$
$\rightarrow (\text{pid}, (Rk^{i+1},\text{list(pid)}))$



Hadoop - Job2

Input

Key:	Value:	
PageA	1.0	PageC, PageD,...
PageB	1.0	<LinksToPages>
...		
...		

Output

Key:	Value:		
PageC	<FromPage>	<Rank>	<TotalLinks>
PageD	PageA	1.0	6
PageE	PageB	1.0	2
...			

Output

Key:	Value:	
PageA	1.3425	PageC, PageD,...
PageB	<newRank>	<LinksToPages>
...		
...		