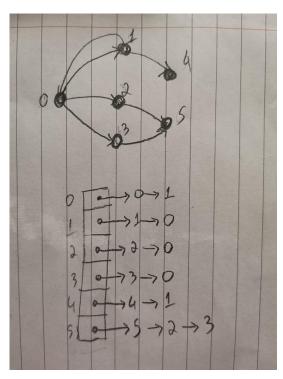
HY342 - PARALLEL COMPUTING

Irodotos Apostolou csd4437

<u>How my algorithm works</u>: I store the graph in Adjacency List that every node in the graph store in the same position as the node id and it have a list with all the incoming nodes. For example



Every time I run the pagerank I check my list with neighbors and I divide neighbors value with neighbors outgoing edges and I add it in my value.

<u>How I get speedup with threads:</u> I split the table in the number of threads so I give less job in each thread an run parallel

	Α	В	С	D	E	F	G	Н	1	J	K	L	М	
1		1 thread			2 threads			3 threads			4 threads			
2		Gnutella	Facebook	Email	Gnutella	Facebook	Email	Gnutella	Facebook	Email	Gnutella	Facebook	Email	
3	1st run	0.925	0.631	5.079	0.666	0.451	4.606	0.538	0.482	4.123	0.438	0.314	3.895	
4	2 nd run	0.916	0.633	5.025	0.68	0.386	4.502	0.543	0.496	4.185	0.4	0.303	3.922	
5	3 rd run	0.884	0.633	5.039	0.644	0.422	4.553	0.501	0.515	4.19	0.457	0.298	3.975	
6	4th run	0.926	0.653	5.11	0.629	0.439	4.528	0.513	0.484	4.208	0.444	0.315	3.704	
7	5th run	0.917	0.655	5.15	0.655	0.404	4.608	0.553	0.485	4.18	0.442	0.291	4.12	
8	Stand. Deviation	0.01715517	0.011916375	0.051247439	0.019639246	0.026159128	0.047051036	0.021743965	0.013758634	0.032088939	0.021452273	0.010329569	0.150215512	
9	Average Deviation	0.01184	0.0104	0.03952	0.01464	0.02032	0.03808	0.01808	0.01048	0.02168	0.01448	0.00824	0.09944	
10	Average value	0.9136	0.641	5.0806	0.6548	0.4204	4.5594	0.5296	0.4924	4.1772	0.4362	0.3042	3.9232	
11	speed up	1	1	1	1.395235186	1.524738344	1.114313287	1.725075529	1.301787165	1.216269271	2.094452086	2.107166338	1.295014274	
12							I							
13							i							