Parallel and distributed processing of big data

Elioz Geller & Yanir Avitan

Functional programming in concurrent and distributed systems course

Lecturer's name -Dr. Yehuda Ben-Shimol

Practitioner's name -Mr. David Leon

Ben GurionUniversity, Beer Sheva, ISRAEL

About

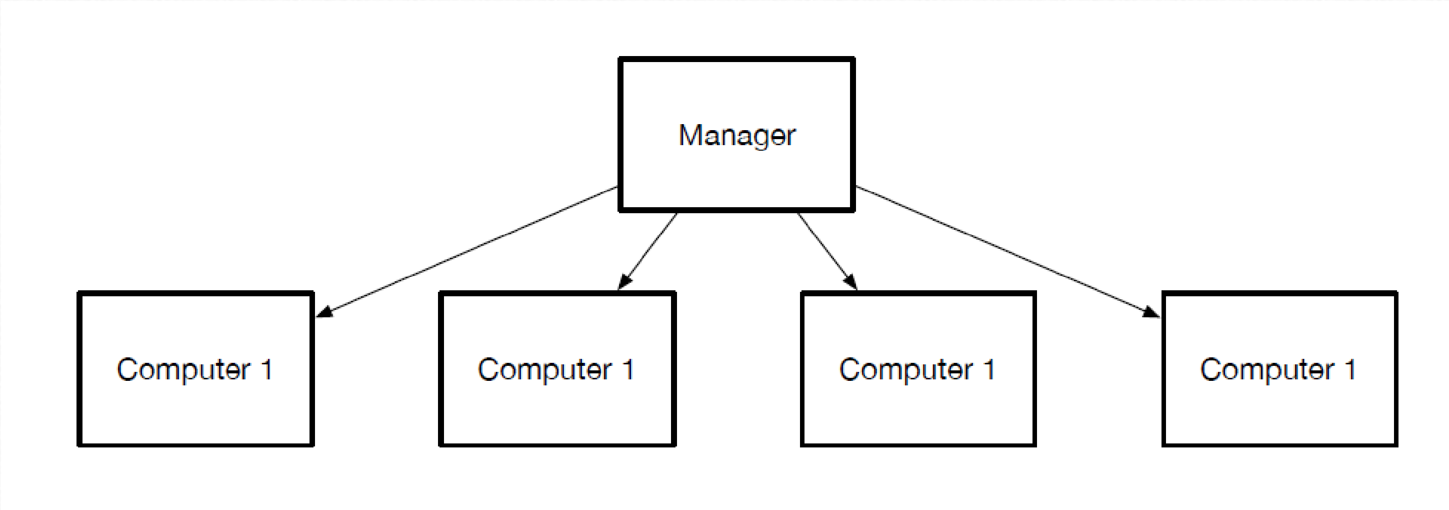
- In this project we will use map reduce to analyze big data in parallel and distributed systems with Erlang programming language.
- This program analyze big data from www.dblp.org and create tree of the partners for the author that the user chooses. In addition this program displays a table of the number of surnames beginning with the same letter at each level in the tree.

How it works

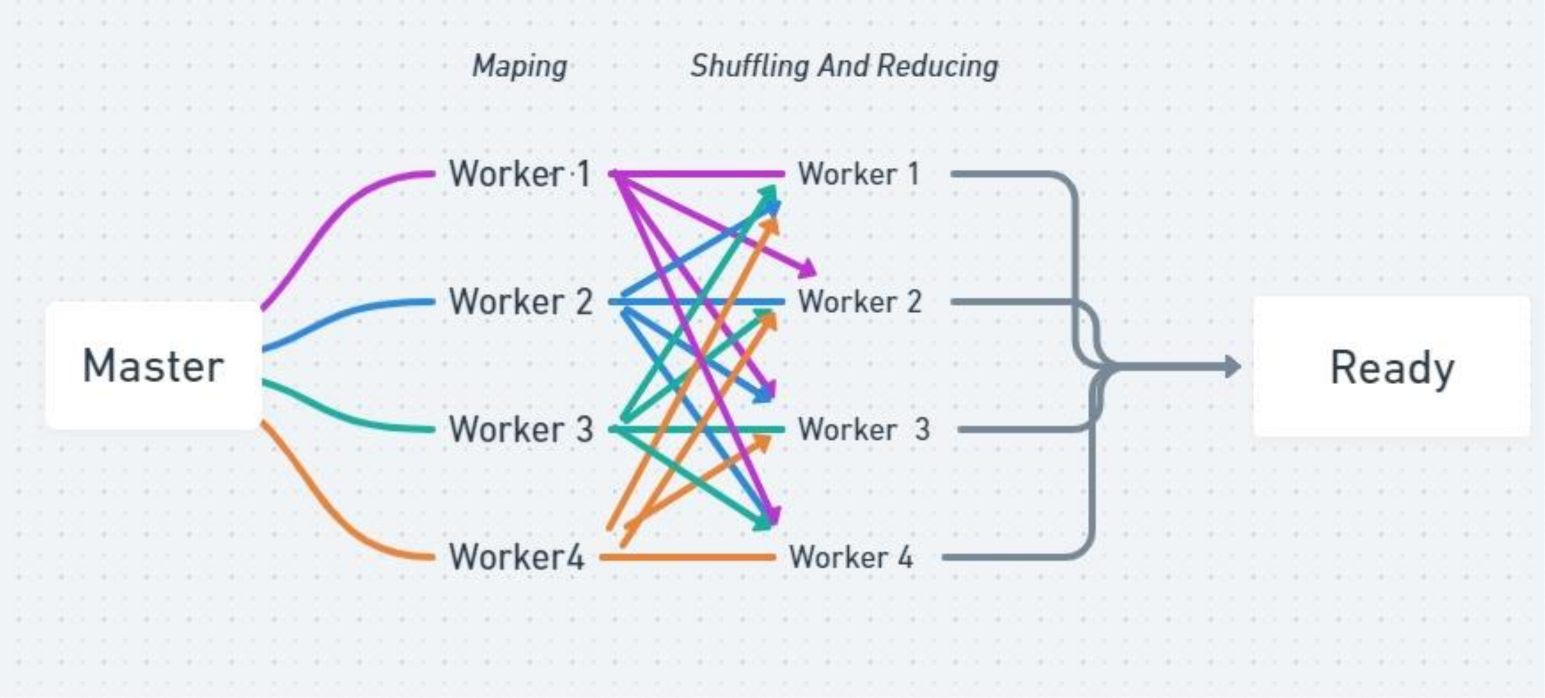
- The master assigns responsibility to workers for certain letters.
- The master divides the files between the computers.
- Each computer processes its part and transmits the data to the computer responsible on this author.
- After the processing done the program wait for input from the user.
- After the input, the data from the computer responsible on each researcher is requested in a parallel and recursive way.

Multiple computers

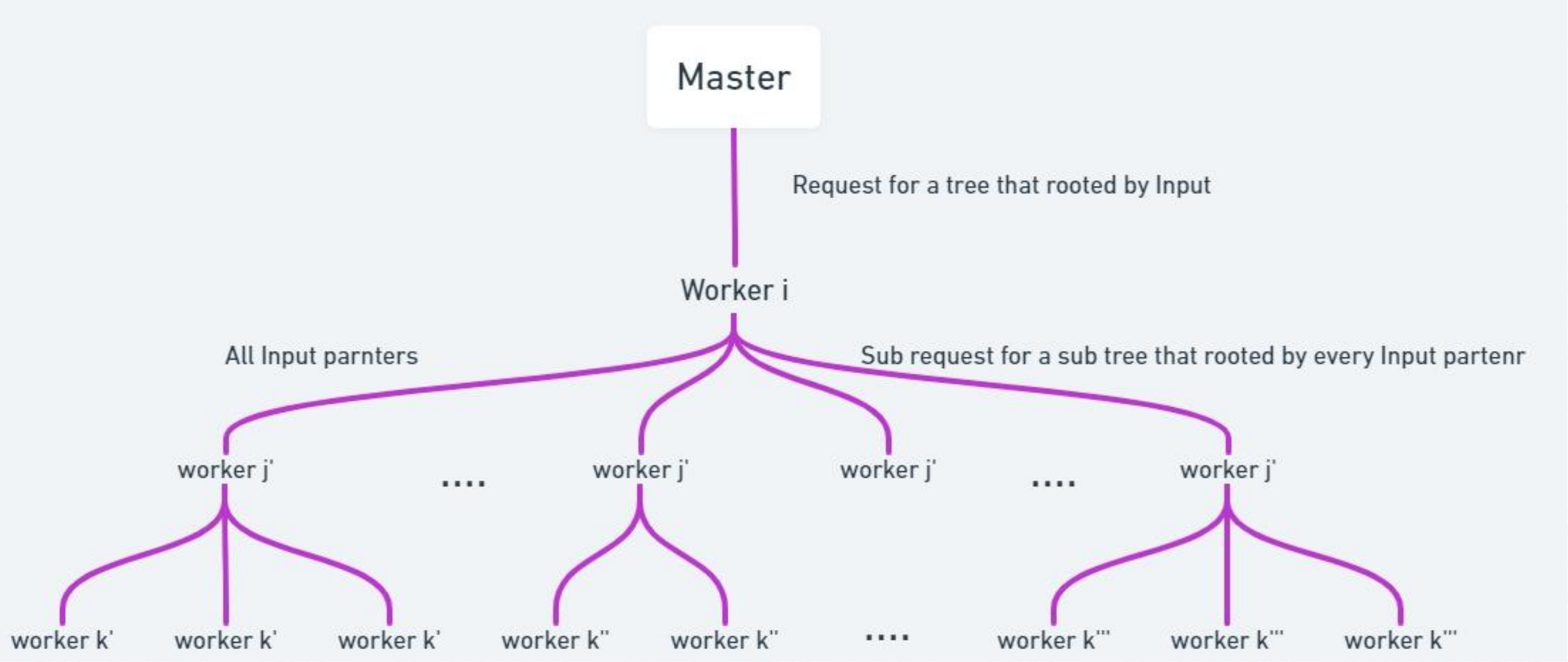
- Our software is unlimited in the number of computers that will participate in the process.



MapReduce - Processing stage



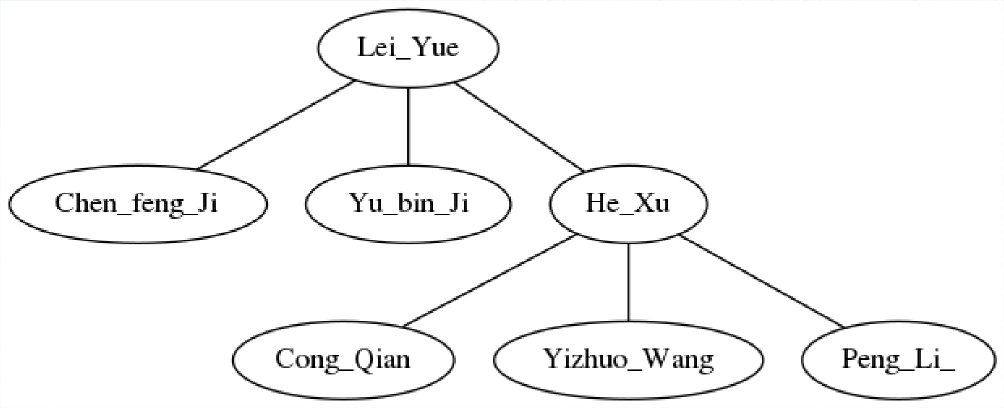
Tree construct stage



Tree construct stage - example

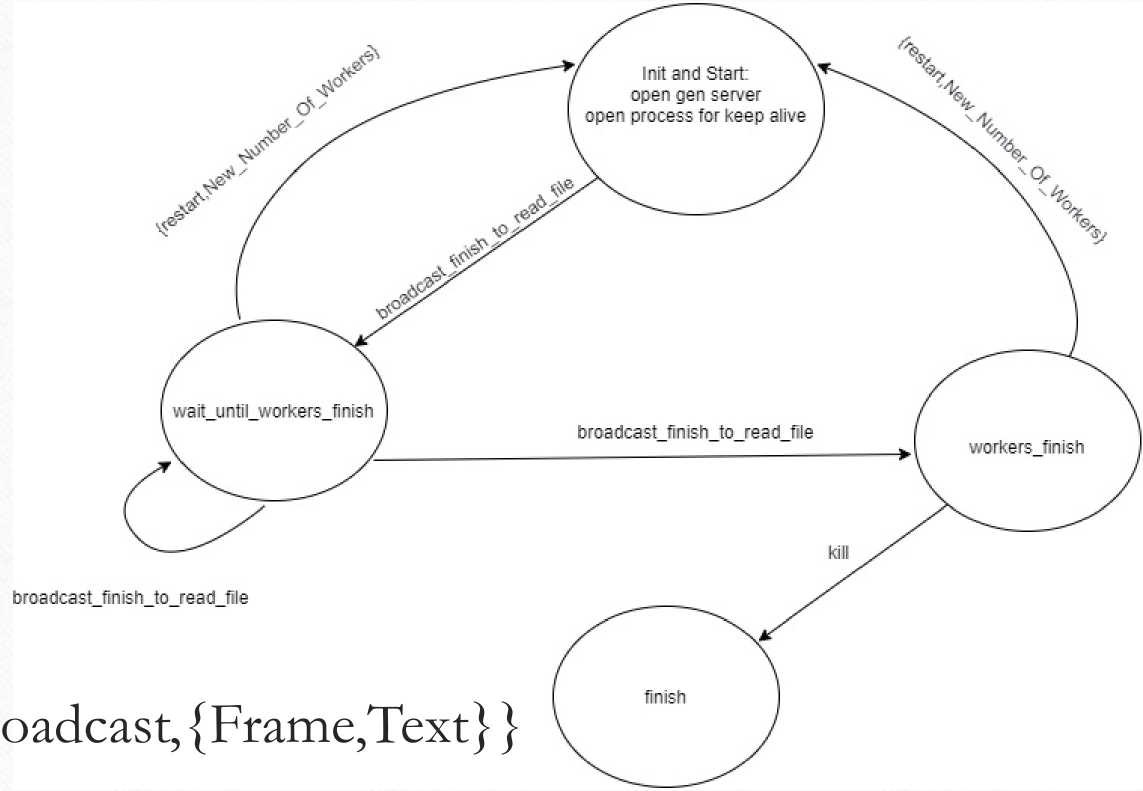
Worker 1	Worker 2	Worker 3	Worker 4
Letters a - f	Letters g - l	Letters m - s	Letters t - z

Structure = #{Chen feng Ji =>[]...}	Structure = #{Lei Yue =>
	[Yu Bin Ji,Chen feng Ji, He Xu],
	He Xu => [Yizhhuo Wang,
	Cong Qian,Peng Li]...}
			Structure = #{Yu Bin =>[]...}



Master - state machine

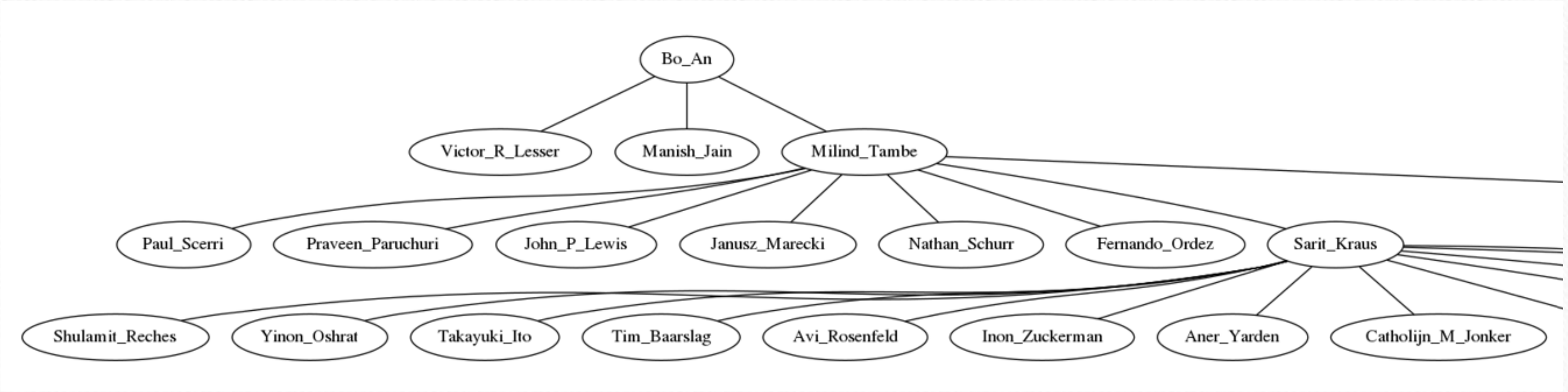
- Init and start
- Wait to workers
- Workers finish to process
- Finish


Init and Start:
open gen server
open process for keep alive
{restart,New_Number_Of_Workers}
{restart,New_Number_Of_Workers}
broadcast_finish_to_read_file
wait_until_workers_finish
broadcast_finish_to_read_file
workers_finish
kill
broadcast_finish_to_read_file
finish

Data structure = {Number_Of_Workers,Count_Start,Count_Broadcast,{Frame,Text}}

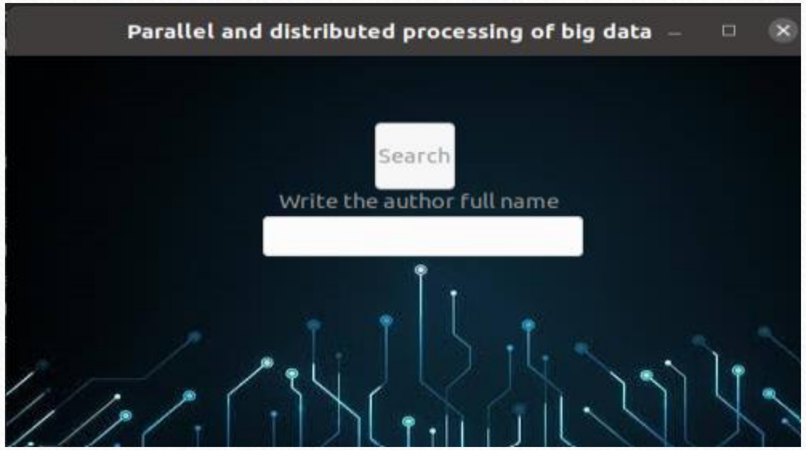
Dynamic code

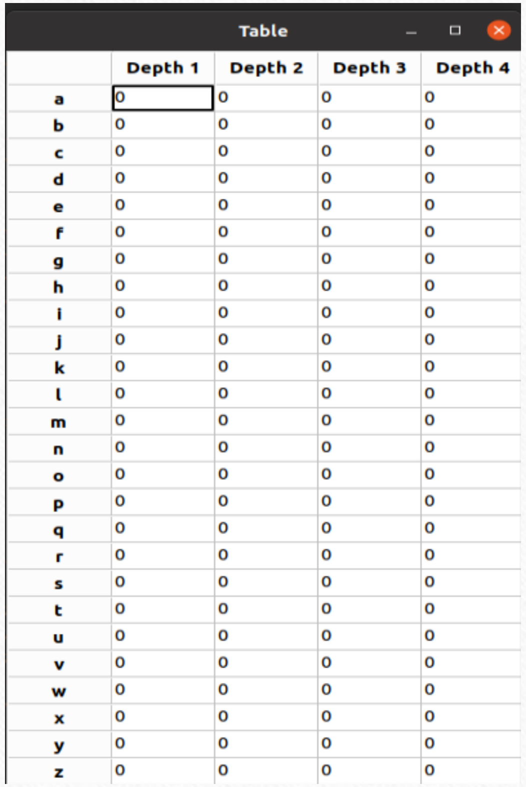
- There is no limit to the amount of computers that can participate in processing.
- There is a DEFINE to control the depth of the tree we get.



GUI

- Simple interface for users.

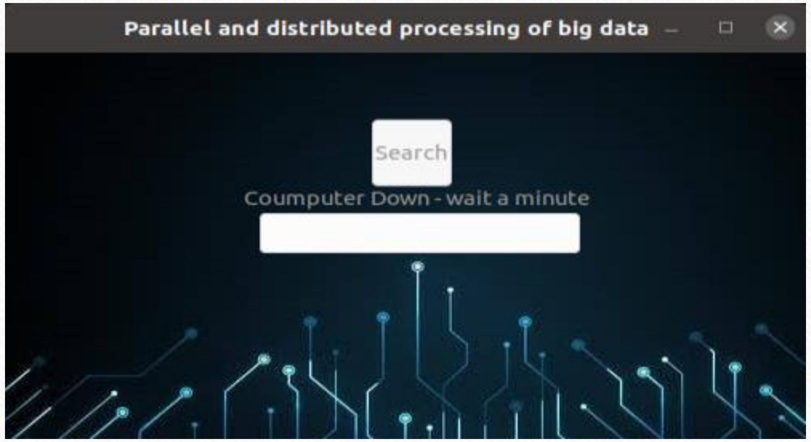

Parallel and distributed processing of big data
Search
Write the author full name



	Depth 1	Depth 2	Depth 3	Depth 4
a	0	0	0	0
b	0	0	0	0
c	0	0	0	0
d	0	0	0	0
e	0	0	0	0
f	0	0	0	0
g	0	0	0	0
h	0	0	0	0
i	0	0	0	0
j	0	0	0	0
k	0	0	0	0
l	0	0	0	0
m	0	0	0	0
n	0	0	0	0
o	0	0	0	0
p	0	0	0	0
q	0	0	0	0
r	0	0	0	0
s	0	0	0	0
t	0	0	0	0
u	0	0	0	0
v	0	0	0	0
w	0	0	0	0
x	0	0	0	0
y	0	0	0	0
z	0	0	0	0

Keep alive

- The master constantly checks how many computers are running with keep-alive message.
- If a computer crashes, the master change the responsibility of the computer that crashed to another computer.



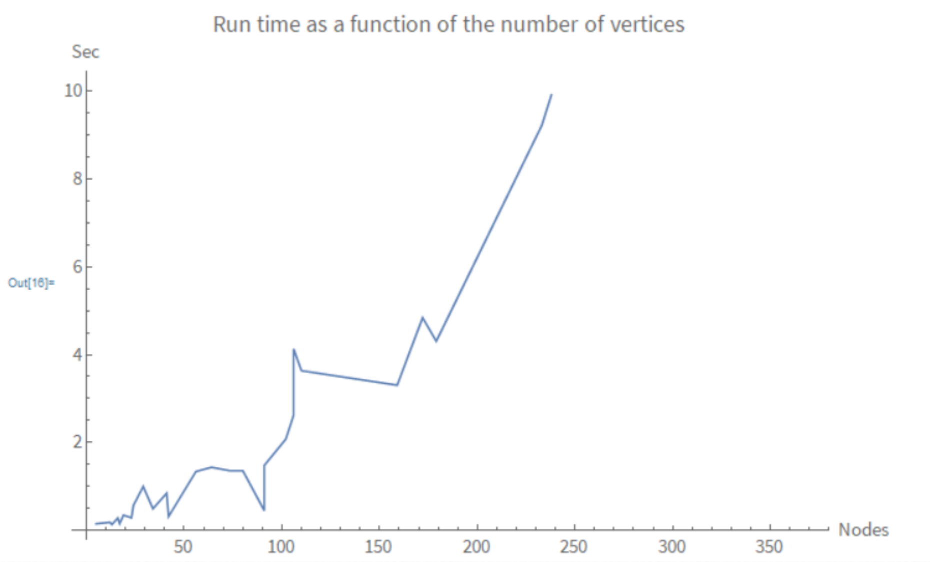
Coumputer Down - wait a minute

Statistics - Processing stage

Average time after 20 test	
4.7475 sec	1 computer 1 worker
4.8825 sec	1 computer 4 workers
4.79 sec	5 computers 4 workers

- As the size of the big data increases, it will be better to process with a large number of computers.
- As the size of the big data decreases, network messaging time will be more significant.

Statistics - Tree construct stage



Average time [sec]	Number of nodes in result	Number of nodes - pc4	Number of nodes - pc3	Number of nodes - pc2	Number of nodes - pc1
0.347658	19	5	5	5	4
1.437036	64	14	24	16	10
0.152049	5	0	0	5	0
2.62173	106	27	35	39	5
0.183969	12	4	4	4	0
0.494069	34	12	9	8	5
0.845886	41	7	15	14	5
0.287291	23	6	4	6	7
0.28002	16	7	4	5	0
0.56946	24	7	11	6	0
4.306796	179	31	64	45	39
85.098412	772	143	241	182	206
0.449196	91	9	5	73	4
4.842396	172	73	50	25	24
2.07836	102	17	33	15	37
0.998763	29	16	8	0	5
0.134736	13	5	4	0	4
0.155572	17	5	4	0	8
0.315998	42	4	22	7	9
1.361206	73	20	22	17	14
1.340326	56	20	19	17	0
1.47988	91	29	45	17	0
3.305386	159	21	93	45	0
9.222754	233	53	18	51	111
1.356621	80	15	38	13	14
3.635366	110	15	42	23	30
4.133644	106	28	4	26	48

Youtube & GitHub

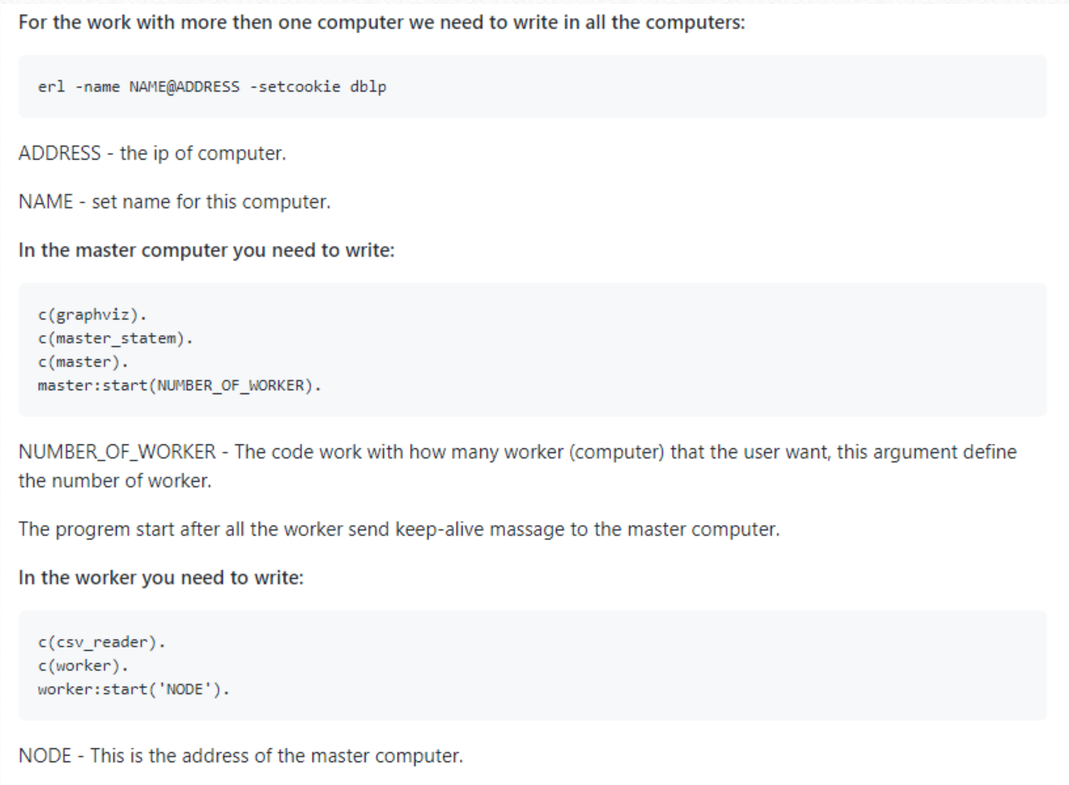
- Link for Youtube video:

https://www.youtube.com/watch?v=7HWlEaO4jUk

- Link for Github project:

https://github.com/yanir26/Parallel-and-distributed-processing-of-big-data

How to run the program



For the work with more then one computer we need to write in all the computers:

```
erl -name NAME@ADDRESS -setcookie dblp
```

ADDRESS - the ip of computer.

NAME - set name for this computer.

In the master computer you need to write:

```
c(graphviz).
c(master_statem).
c(master).
master:start(NUMBER_OF_WORKER).
```

NUMBER_OF_WORKER - The code work with how many worker (computer) that the user want, this argument define the number of worker.

The progrem start after all the worker send keep-alive massage to the master computer.

In the worker you need to write:

```
c(csv_reader).
c(worker).
worker:start('NODE').
```

NODE - This is the address of the master computer.

thanks for listening
any questions ?