

## Assignment 1

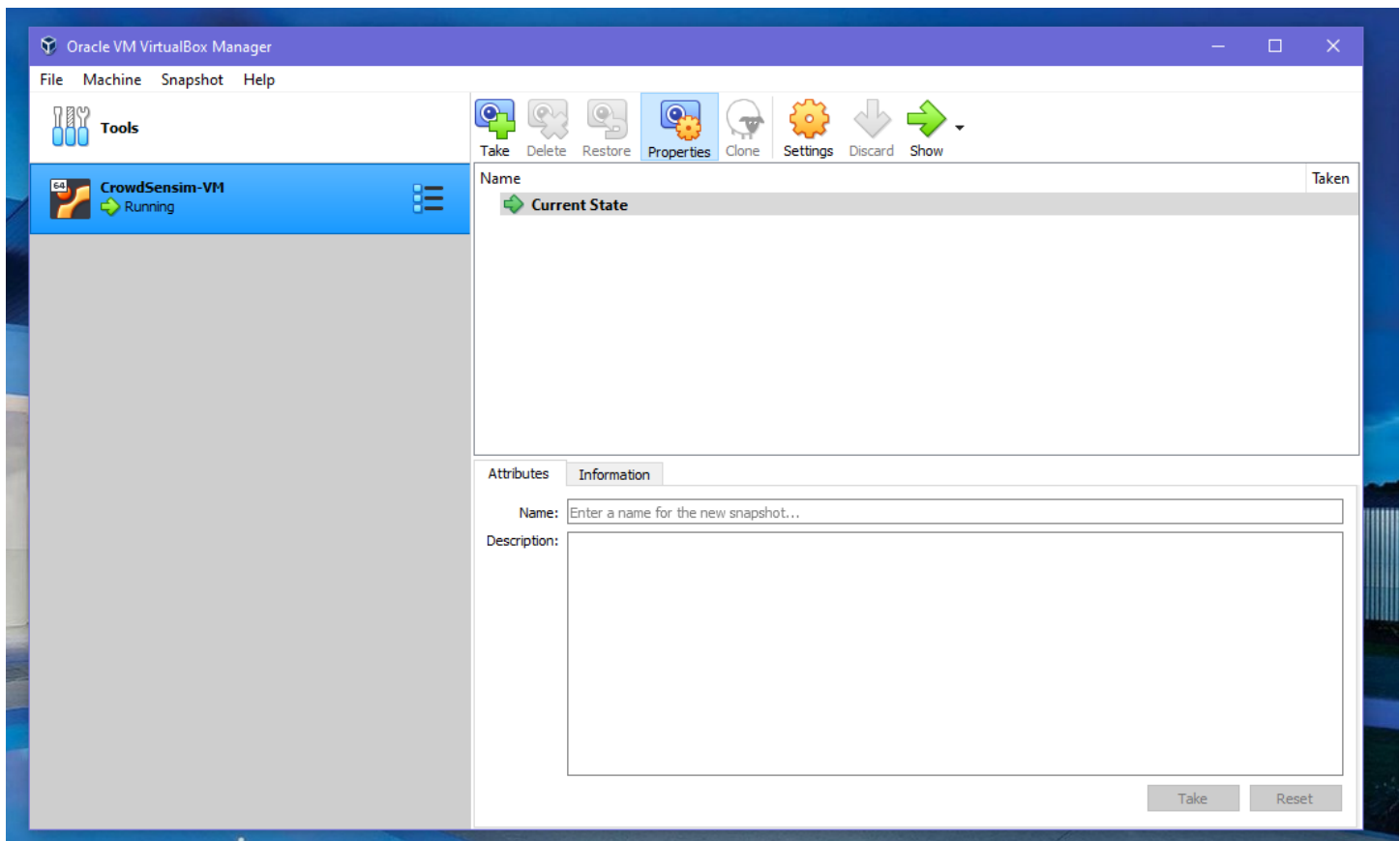
### Group 23: Assignment 1

**Assignment Steps that we are followed:**

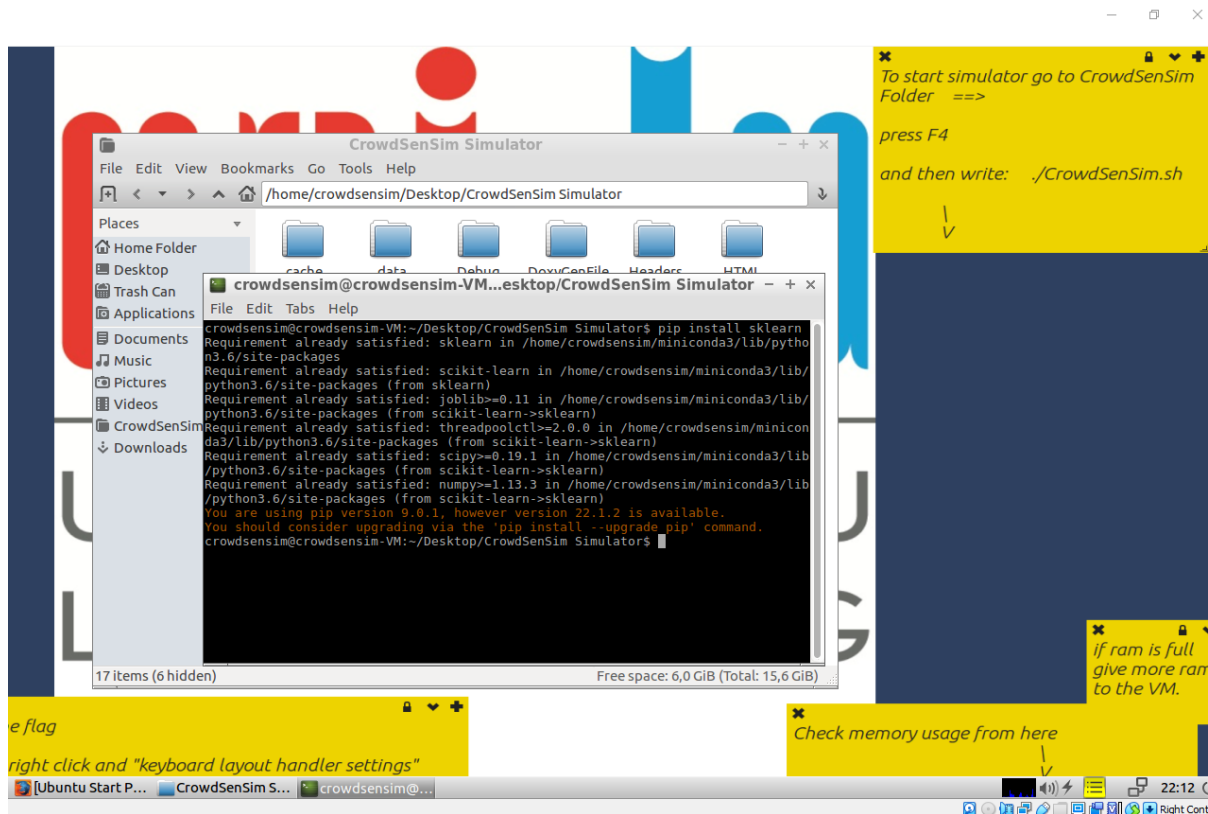
- 1. Setup test environment by installing python and related libraries, install virtual machine and load CrowdSenSim image.**

## 1) Generating Tasks.

Frist, we downloaded VirtualBox and import CrowdSenSim image.

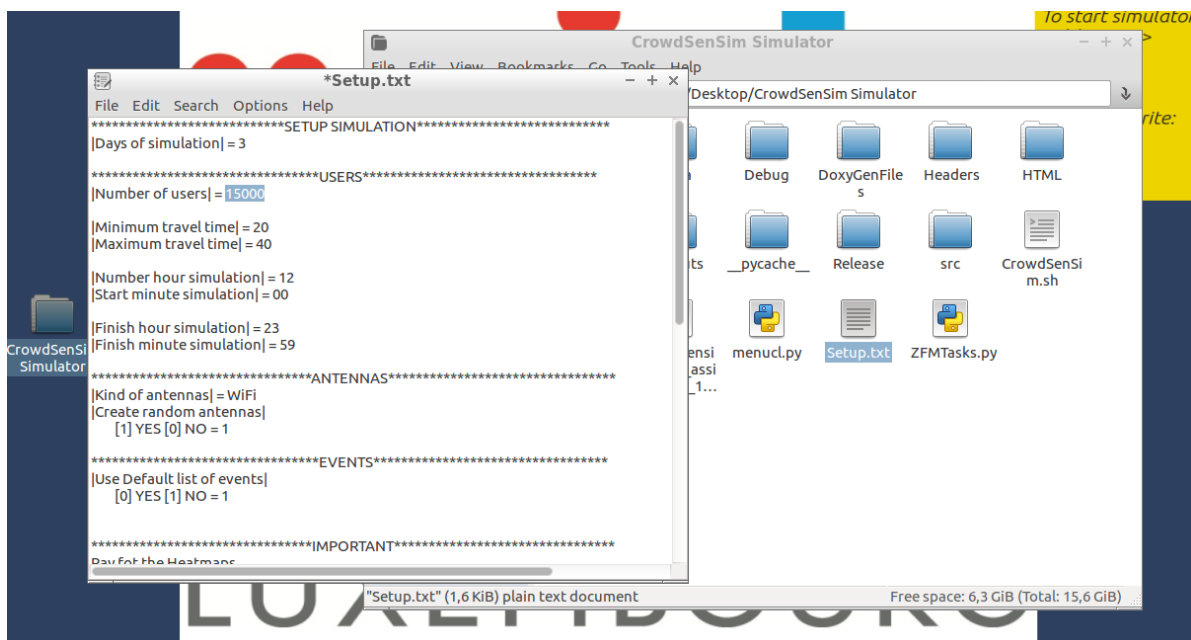


After that we downloaded some python libraries like sklearn.



## 2. Configure Setup.txt file accordingly. The number of days and the number of users is configured in Setup.txt file.

After that we have configured the setup file based on the assignment requirements.



### 3. Refer to ZFMTasks.py to generate tasks and save in mytask.txt

And then we have changed the task\_generator function that in ZFMTasks.py based on the assignment requirement.

**Table 1 Task generation requirements**

Task Feature	Requirement
Day	Distribution consistent in [1, 2, 3]
Hour	50%: 9:00 AM-11:00 AM 25%: 12:00 PM-5:00 PM 25%: 6:00 PM-8:00 AM
Duration (minutes)	50% in {20, 40, 60} 30% in {30, 50, 70} 20% in {10, 80, 100}
Task Value	Uniformly distributed in [1,10]

task\_generator function after the modifications that we have made on it.

```
95
96 def task_generator(big_graph, num_tasks, days, df, ligi, on_peak, attackLocations, num_att):
97     # attack_radius = 50
98     tl = []
99     l = []
100     l = list(big_graph.nodes())
101     # lengthl = len(l)
102     for i in range(num_tasks):
103         day = random.randint(0,days-1)
104         # if(day == 0):
105         #     h = random.randint(12,23)
106         # else:
107         #     h = random.randint(0,23)
108
109         Task_value = np.random.uniform(1,10,(num_tasks)).astype(int) # Task Values (Uniformly distributed)
110         Minute = random.randint(0,59) # Minute Range
111         Hour = random.randint(1,100) # Hours Range
112
113         # Hours
114         if(Hour>=50): # 50%
115             h = random.randint(9, 11) # hours from 9am to 11am
116         elif(Hour<=25): # 25%
117             h = random.randint(12, 17) # hours from 12Pm to 5pm
118         elif(Hour<50 and Hour>25): # 25%
119             h = random.randint(18, 20) # hours from 6Pm to 8pm
```

```

ZFMTasks.py x
home > crowdsensim > Desktop > CrowdSenSim Simulator > ZFMTasks.py
116 elif(Hour<=25): # 25%
117     h = random.randint(12, 17) # hours from 12Pm to 5pm
118 elif(Hour<50 and Hour>25): # 25%
119     h = random.randint(18, 20) # hours from 6Pm to 8pm
120
121 # Duration
122 dur = random.randint(1,10)
123 if(dur<6): # 50%
124     duration = random.choice([20, 40, 60]) # {20, 40, 60}
125 elif(dur<9 and dur<=6): # 30%
126     duration = random.choice([30, 50, 70]) # {30, 50, 70}
127 else: # 20%
128     duration = random.choice([10, 80, 100]) # {10, 80, 100}
129
130
131 remaining_t = duration
132 resources = random.randint(7,10)
133 index = random.choice(range(len(l)))
134 y=big_graph.node[l[index]]['y']
135 x=big_graph.node[l[index]]['x']
136 grid_num=convert_location(big_graph,y,x)
137
138 tl.append([i+1, float(y), float(x), day, h, Minute, duration, remaining_t, resources, df,ligi,on_peak,grid_num,Task_value[i]])
139
140 return tl

```

Mytask.txt and task\_value is uniformly distributed from 1 to 10

mytask.txt - Notepad														
File	Edit	Format	View	Help										
/ID-Task/	/Lat/	/Long/	/Day/	/Hour/	/Minute/	/Duration/	/Remaining time/	/Resources/	/Coverage/	/Legitimacy/	/on peak hour/	/grid_number/	-/task_value	
1	42.12226496514614	12.534513085571489	2	14	51	60	8	100	True	True	19	3		
2	42.15576890373553	12.495746059653097	2	11	40	80	80	100	True	True	65	4		
3	42.14397482701999	12.506639458908973	1	10	3	80	80	8	100	True	True	46		
4	42.14299031339726	12.49604406184683	2	10	39	60	60	9	100	True	True	45		
5	42.14777241495238	12.521271239709776	2	10	43	100	100	8	100	True	True	58		
6	42.145508959947676	12.466538268302731	0	9	45	60	60	9	100	True	True	51		
7	42.121752603039944	12.534938453003988	0	11	29	10	10	9	100	True	True	19		
8	42.14396601555058	12.464758061491924	0	11	57	60	60	7	100	True	True	41		
9	42.16056135650676	12.525319716537046	2	10	42	40	40	9	100	True	True	78		
10	42.1521923370804	12.477919367932504	0	18	25	60	60	10	100	True	True	62		
11	42.13007936365291	12.478966118155453	1	16	25	20	20	10	100	True	True	22		
12	42.154980432236144	12.50515678581365	1	10	12	10	10	9	100	True	True	66		
13	42.154644161678824	12.527088115633667	2	11	26	80	80	8	100	True	True	68		
14	42.14911953906333	12.506205614861857	0	11	37	100	100	8	100	True	True	56		
15	42.15069187675908	12.492931807132631	0	20	31	30	30	7	100	True	True	64		
16	42.156454917090514	12.500553781397018	2	10	56	80	80	7	100	True	True	75		
17	42.14412738934361	12.47867288243402	0	13	40	80	80	9	100	True	True	42		
18	42.13629943227805	12.4930267810776	0	11	27	10	10	7	100	True	True	34		
19	42.14841517291863	12.477475298289447	0	9	31	20	20	9	100	True	True	52		
20	42.14670219474318	12.509569228818969	2	9	20	20	20	7	100	True	True	56		
21	42.15669032445653	12.492258093757234	0	10	31	80	80	10	100	True	True	74		
22	42.14930809262161	12.516585221285435	0	11	9	80	80	7	100	True	True	57		
23	42.14058504259458	12.527607108447398	2	19	8	40	40	10	100	True	True	49		
24	42.14718200796527	12.492278901129529	0	10	59	40	40	8	100	True	True	54		
25	42.14487185910861	12.478374718445254	1	9	47	20	20	9	100	True	True	52		
26	42.135803698566576	12.48560290421483	0	13	14	40	40	9	100	True	True	33		
27	42.15654320206294	12.485188079369696	0	9	11	20	20	7	100	True	True	73		
28	42.13050834495915	12.521232399797368	1	19	25	80	80	10	100	True	True	28		
29	42.13237601612404	12.516513976138306	0	11	41	80	80	9	100	True	True	27		

Mytask.csv and task\_value is uniformly distributed from 1 to 10

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	ID	Latitude	Longitude	Day	Hour	Minute	Duration	RemainingTime	Resources	Coverage	Ligitimacy	OnPeakHours	GridNumber	task_value
2	1	42.12226497	12.53451	2	14	51	60	60	8	100	TRUE	TRUE	19	3
3	2	42.1557689	12.49575	2	11	40	80	80	10	100	TRUE	TRUE	65	4
4	3	42.14397483	12.50664	1	10	3	80	80	8	100	TRUE	TRUE	46	6
5	4	42.14299031	12.49604	2	10	39	60	60	9	100	TRUE	TRUE	45	1
6	5	42.14777241	12.52127	2	10	43	100	100	8	100	TRUE	TRUE	58	8
7	6	42.14550896	12.46654	0	9	45	60	60	9	100	TRUE	TRUE	51	2
8	7	42.1217526	12.53494	0	11	29	10	10	9	100	TRUE	TRUE	19	1
9	8	42.14396602	12.46476	0	11	57	60	60	7	100	TRUE	TRUE	41	5
10	9	42.16056136	12.52532	2	10	42	40	40	9	100	TRUE	TRUE	78	3
11	10	42.15219234	12.47792	0	18	25	60	60	10	100	TRUE	TRUE	62	3
12	11	42.13087936	12.47897	1	16	25	20	20	10	100	TRUE	TRUE	22	3
13	12	42.15498043	12.50516	1	10	12	10	10	9	100	TRUE	TRUE	66	4
14	13	42.15464416	12.52709	2	11	26	80	80	8	100	TRUE	TRUE	68	3
15	14	42.14911954	12.50621	0	11	37	100	100	8	100	TRUE	TRUE	56	1
16	15	42.15069188	12.49293	0	20	31	30	30	7	100	TRUE	TRUE	64	3
17	16	42.15645492	12.50055	2	10	56	80	80	7	100	TRUE	TRUE	75	5
18	17	42.14412739	12.47867	0	13	40	80	80	9	100	TRUE	TRUE	42	7
19	18	42.13629943	12.49303	0	11	27	10	10	7	100	TRUE	TRUE	34	4
20	19	42.14841517	12.47748	0	9	31	20	20	9	100	TRUE	TRUE	52	7
21	20	42.14670219	12.50957	2	9	20	20	20	7	100	TRUE	TRUE	56	7
22	21	42.15669033	12.49225	0	10	31	80	80	10	100	TRUE	TRUE	74	2
23	22	42.14930089	12.51659	0	11	9	80	80	7	100	TRUE	TRUE	57	7
24	23	42.14058504	12.52761	2	19	8	40	40	10	100	TRUE	TRUE	49	7
25	24	42.14718201	12.49228	0	10	59	40	40	8	100	TRUE	TRUE	54	1
26	25	42.14487186	12.47837	1	9	47	20	20	9	100	TRUE	TRUE	52	4
27	26	42.1358037	12.4856	0	13	14	40	40	9	100	TRUE	TRUE	33	7
28	27	42.1565432	12.48519	0	9	11	20	20	7	100	TRUE	TRUE	73	3
29	28	42.13050834	12.52123	1	19	25	80	80	10	100	TRUE	TRUE	28	2

## Task\_value

109

Task\_value = np.random.uniform(1,10,(num\_tasks)).astype(int) # Task Values (Uniformly distributed)

## 2) Obtaining user movement event.

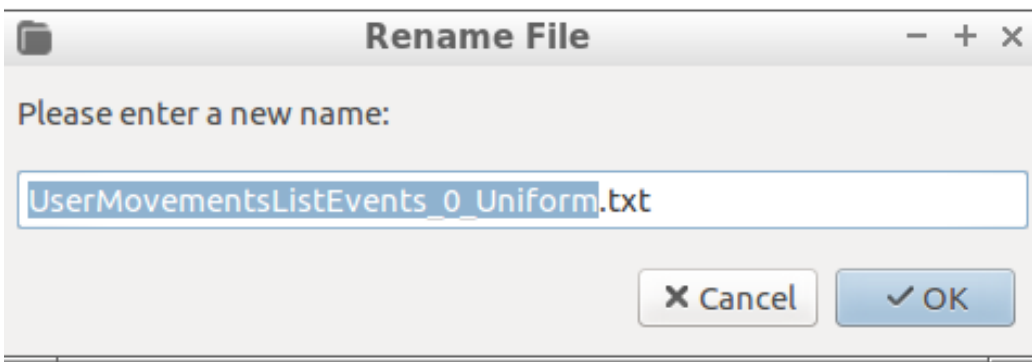
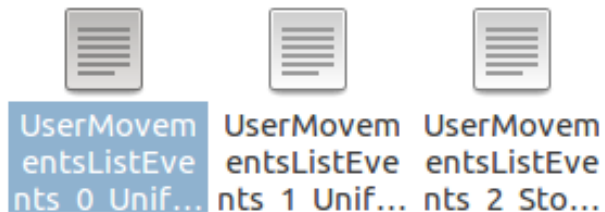
4. Refer to crowdsensim2.py to generate user movements event and save in to usermovements\_0.txt, usermovements\_1.txt, and usermovements\_2.txt (representing three-day simulation)

- **Generate user mobility under uniform algorithm**

- Uniform algorithm

```
625
626 speed = random.uniform(min_speed,max_speed)
627 mincamm=random.randint(20,40)
628 cut=mincamm*60*speed
629 cutadded=cut+maxlen
630
631 (length, path)= nx.single_source_dijkstra(G_old, origin_node,target=None, cutoff=cutadded, weight='length')
632
633 idr=0
634 for l in length:
635     if length[l]>cut:
636         idr=l
637
638 if idr == 0:
639     idr = max(length, key=length.get)
640
641 route=path[idr]
642
643 #get the last node so we can continue our path!!
644
645 # If userused==1, do the HTML pathway generation for it
646 # most important part of this refactor, we can now do this for any path
647 if userused==1 :|
648     htmlRouteGen(route, G_old, save=toSave)
649
```

The generated files using uniform algorithm

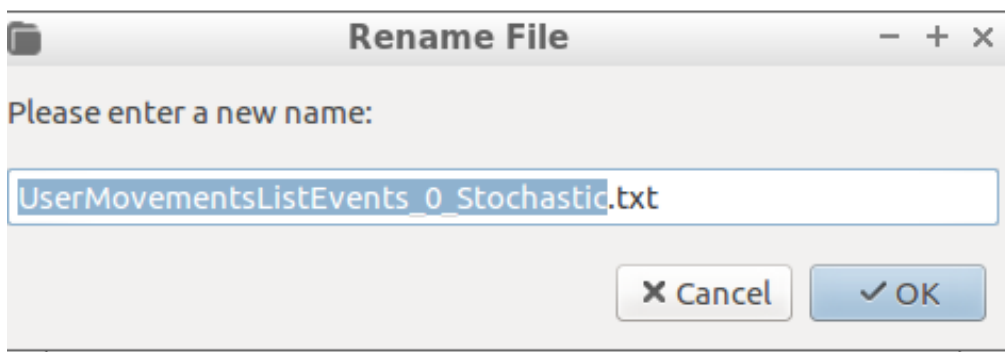
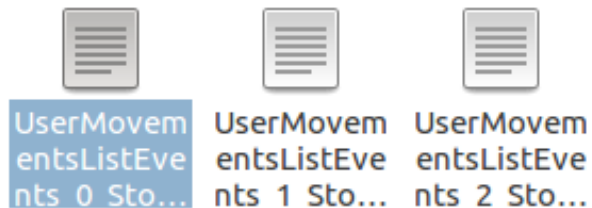


- **Generate user mobility under stochastic algorithm**

- Stochastic algorithm

```
625
626     speed = random.uniform(min_speed,max_speed)
627     mincamm=random.randint(20,40)
628     cut=mincamm*60*speed
629     cutadded=cut+maxlen
630
631     route = None
632     if (not social_model_routes):
633         try:
634             (length, path)= nx.single_source_dijkstra(G_old, origin_node,target=None, cutoff=cutadded, weight='length')
635         except:
636             continue
637
638     idr = 0
639     if stochastic_model_routes:
640         Lengths_Array = np.array(list(length.values()))
641         Lengths_Array = np.where(Lengths_Array <= 0, 0.1, Lengths_Array)
642         Weights = np.random.dirichlet(Lengths_Array , 1)
643         idr = random.choices(tuple(length), weights = Weights[0])[0]
644
645     else:
646         for l in length:
647             if length[l]>cut:
648                 idr=l
649             if idr==0:
650                 idr=max(length, key=length.get)
651
652     route=path[idr]
653 else:
654     route = socialModel.chooseRoute(userused-1, cut, G_old)
655
```

The generated files using Stochastic algorithm





## Results of Stochastic and uniform algorithms.

For example, this is the first file (**different file for each algorithm**) and it has 3000 users.

CrowdSensim-VM [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

UserMovementsListEvents\_0\_Stochastic.txt

1 /ID-User-/Lat-/Long-/Alt-/Day-/Hour-/Minute-/

2 1 42.149424892440244 12.516403702979078 0 0 12 28

3 1 42.1500032 12.5156932 0 0 12 29 1 319.7333653852286 25.560384420918076

4 1 42.1503124 12.5147451 0 0 12 30 2 285.79156058748265 25.8642868918928

5 1 42.149906 12.5141119 0 0 12 31 3 217.63242399480944 39.01915553668857

6 1 42.1497453 12.5132157 0 0 12 32 4 268.2056625132144 34.08882716384403

7 1 42.1492758 12.5125293 0 0 12 33 5 227.35431684173165 29.79036949082997

8 1 42.1489486 12.5117146 0 0 12 34 6 238.661210801781 39.08181693047841

9 1 42.1485044 12.5110505 0 0 12 35 7 226.52098623562742 33.09558638415172

10 1 42.1482354 12.5102347 0 0 12 36 8 247.66809202386165 33.03903074488508

11 1 42.1487621 12.5098878 0 0 12 37 9 330.42808032298404 16.454257727317373

12 1 42.1494325 12.5089736 0 0 12 38 10 314.6862955852897 106.005389781834

13 1 42.1496138 12.5085206 0 0 12 39 11 285.13003385358417 18.57437870209047

14 1 42.1498355 12.5076146 0 0 12 40 12 288.26612492639316 78.65318903834725

15 1 42.1494938 12.5068642 0 0 12 41 13 221.5257947719291 30.78924529237164

16 1 42.1491201 12.5062782 0 0 12 42 14 239.7144360310289 12.965205129553457

17 1 42.1493216 12.5050823 0 0 12 43 15 291.77202377193504 38.40276110230279

18 1 42.149419 12.5045296 0 0 12 44 16 261.03493320551644 21.549088391553152

19 1 42.1493413 12.5034456 0 0 12 45 17 269.5988329022038 36.51350503576786

20 1 42.1489115 12.5032718 0 0 12 46 18 208.78210105219478 10.872904927797483

21 1 42.1490966 12.5021422 0 0 12 47 19 289.6105053159331 66.3287578500791

22 1 42.1491059 12.5014334 0 0 12 48 20 259.99165900305286 16.12309995160703

23 1 42.1486936 12.5006292 0 0 12 49 21 217.66254013129162 22.92371262650768

24 1 42.148042 12.499936 0 0 12 50 22 218.26460726863846 92.28002353776873

25 1 42.147743 12.4994649 0 0 12 51 23 235.8850367874445 22.026656998726164

26 1 42.147471 12.4981195 0 0 12 52 24 257.84015603125994 56.16672534641496

27 1 42.1473731 12.4978503 0 0 12 53 25 243.8719350874431 24.71951670103445

28 1 42.1466969 12.4972405 0 0 12 54 26 205.56729838784042 42.42774815994957

29 1 42.1462539 12.4963046 0 0 12 55 27 264.4308664952423 56.485066289531055

30 1 42.1459727 12.4960253 0 0 12 56 28 209.02256572506056 30.061271212946846

31 1 42.1454032 12.4951362 0 0 12 57 29 238.3384280648961 48.024005803702

32 1 42.1452191 12.4945494 0 0 12 58 30 247.06501140813663 52.53164442128977

33 1 42.1451374 12.4935303 0 0 12 59 31 277.80703267797514 29.63354425484127

34 1 42.1452738 12.4928832 0 0 13 0 32 284.87660318425816 22.434911842357323

35 2 42.149623107595616 12.518532234935687 0 0 12 35

36 2 42.1495797 12.5176318 0 0 12 36 1 266.5949970825227 29.392138558990965

37 2 42.1491651 12.5169403 0 0 12 37 2 260.6502625028479 22.107117329660497

38 2 42.1498278 12.5158936 0 0 12 38 3 316.8283644508876 70.43985152229264

39 2 42.1500032 12.5156932 0 0 12 39 4 319.7333653852286 25.560384420918076

UserMovementsListEvents\_0\_Uniform.txt

1 /ID-User-/Lat-/Long-/Alt-/Day-/Hour-/Minute-/

2 1 42.13257418302367 12.516324855920264 0 0 12 25

3 1 42.1333996 12.5156545 0 0 12 26 1 329.1525498455697 103.8730293718048

4 1 42.1340957 12.5148735 0 0 12 27 2 316.7627209143561 42.49248854850592

5 1 42.1347473 12.5136193 0 0 12 28 3 305.0149035718331 126.27529514340397

6 1 42.1351929 12.5130861 0 0 12 29 4 314.4379443123036 44.23115034379217

7 1 42.13578198272946 12.51212554081099 0 0 12 30 6 311.7843218089833 156.13181717260733

8 1 42.1361786 12.511524 0 0 12 31 6 311.7843218089833 156.13181717260733

9 1 42.1368525 12.5110809 0 0 12 32 7 345.65423279523395 27.48852116066554

10 1 42.1375319 12.510788 0 0 12 33 8 328.5032536758164 22.521517964830956

11 1 42.1383839 12.5102773 0 0 12 34 9 340.4376432217406 49.940368739181885

12 1 42.1393571 1267254 12.509030109025371 0 0 12 35 11 306.6738597283788 108.85778677469425

13 1 42.1394587 12.5088429 0 0 12 36 11 306.6738597283788 108.85778677469425

14 1 42.139804 12.5078666 0 0 12 37 12 295.5002785118867 89.1863689976912

15 1 42.140264496815035 12.506076450170905 0 0 12 38 14 288.3385572825905 114.15534705792551

16 1 42.140333 12.5057939 0 0 12 39 14 288.3385572825905 114.15534705792551

17 1 42.140656 12.5048326 0 0 12 40 15 283.538596969229 34.38956850606637

18 1 42.1409177 12.5038487 0 0 12 41 16 317.4483642468839 10.082840128296528

19 1 42.1420649 12.5033456 0 0 12 42 17 354.28773203698614 49.784651100748036

20 1 42.1425542 12.5030535 0 0 12 43 18 335.66042582045077 43.41051745086098

21 1 42.1419974 12.5037481 0 0 12 44 19 143.430922050731 29.877524382456066

22 1 42.1413431 12.504471 0 0 12 45 20 89.99990552817297 23.217876332594095

23 1 42.1420809 12.5045555 0 0 12 46 21 336.1680567345428 36.38294381303815

24 1 42.1429023 12.5049779 0 0 12 47 22 27.233828925779505 59.20265190697298

25 1 42.143306 12.5053065 0 0 12 48 23 31.11238022370236 52.43143906460275

26 1 42.1431389 12.5062171 0 0 12 49 24 169.86234910143446 22.015606527672592

27 1 42.1425386 12.506888 0 0 12 50 25 137.14340962292545 16.89797355607287

28 1 42.1436244257666 12.50722837915263 0 0 12 51 27 1.9369947474600053 112.2043634845748

29 1 42.1438245 12.5072321 0 0 12 52 27 1.9369947474600053 112.2043634845748

30 1 42.1451984 12.5071902 0 0 12 53 28 358.7046489382124 152.8099541282007

31 1 42.1458004 12.5074915 0 0 12 54 29 22.053926452692167 53.13620178599096

32 1 42.1466426 12.5069188 0 0 12 55 30 328.5647692015395 71.62537040073825

33 1 42.1473804 12.506487 0 0 12 56 31 336.5430700008155 89.43038637713838

34 1 42.1476292 12.506258 0 0 12 57 32 325.68982131193326 33.49324196391354

35 1 42.1480643 12.5052124 0 0 12 58 33 282.8773468937476 30.385419456959333

36 1 42.1486541 12.5046001 0 0 12 59 34 318.49967104952475 16.895306535813504

37 1 42.1487417 12.5035794 0 0 13 0 35 289.11627652372135 24.41299292664264

38 1 42.1490966 12.5021422 0 0 13 1 36 289.6105053159331 66.3287578500791

39 1 42.1491059 12.5014334 0 0 13 2 37 259.99165900305286 16.12309995160703

for more layout right click and "keyboard layout handler settings"

Check memory usage from here

21:52