REPORT

MUKKU VAMSI KRISHNA REDDY-2019CS10374

NALLA KHYATEESWAR NAIDU-2019CS10376

March 2021

1 METRICS

.

I have used different definitions for utility.And used some common terms for all the methods which we can discuss here.

queoriginal——-which contains queue density values of baseline.

quemodified——-which contains queue density values of the obtained frames in the methods.

error————- average of absolute sum of the differnce between the rate of queue densities in queoriginal and quemodified.

utility———1/error.(for method 1 and 2)

utility———10-error.(for method 3 and 4)

2 METHODS

.

we will discuss the methods we have done in this .

2.1 Method1 (sub-sampling frames)

.

x—- skip number.

We have taken the values of the queoriginal by reading the text "baseline.txt"(which contains all quedensity values of baseline).

We are reading the trafficvideo.mp4 and then taking the quedensity values of the frames b skipping x.(if we take Nth frame then we have taken N+x and not for between frames).

After this we have to give utility.for that we have taken rates of quedensity of quemodified frames between N and N+x have same rate as of N+x.

for runtime we have considered from the starting of video reading to the end of giving the utility.

2.2 Method2 (reduce resolution for each frame)

.

X,Y——image width and height respectively.

We have taken the values of the queoriginal by reading the text "baseline.txt"(which contains all quedensity values of baseline).

We are reading the traffic video and made the size of each frame to the x and y.

And then calucalated utility with the given values of queoriginal and obtained values of the quemodified.

For runtime we have considered from the starting of video reading to the end of giving the utility.

2.3 Method3 (split work spatially across threads)

.

x—- number of threads.

We have taken the values of the queoriginal by reading the text "baseline.txt"(which contains all quedensity values of baseline).

we are reading the traffic video and whenever we obtain a frame we have splitted the work of calucalating queue density for "x" different threads (ex: 1st thread calucalates 0th row ,x th row,2x th row.....)

Here as we are using the same frames and using same queue density rate formula for the queoriginal and quemodified.so error we will get zero.

for runtime we have considered from the starting of video reading to the end of giving the utility.

2.4 Method4 (split work temporally across threads)

.

x—- number of threads.

We have taken the values of the queoriginal by reading the text "baseline.txt"(which contains all quedensity values of baseline).

We are reading the traffic video and we are taking the "x" framesat a time for the calucalation of the quedensity at the same time in those "x" threads.

Here as we are using the same frames and using same queue density rate formula for the queoriginal and quemodified.so error we will get zero.

for runtime we have considered from the starting of video reading to the end of giving the utility.

3 Trade-off Analysis

.

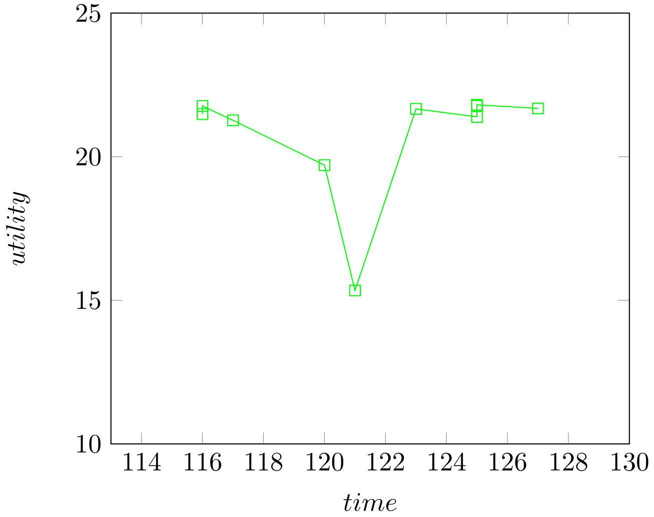
Now lets get into the tablular data and graphical data.

3.1 method1

3.1.1 TABLE

s.no	x	utility	time
1	2	15.3438	121s
2	3	19.7083	120s
3	5	21.4851	116s
4	7	21.27	117s
5	10	21.3912	125s
6	20	21.6677	123s
7	30	21.7695	125s
8	40	21.684	127s
9	50	21.7691	116s
10	100	21.8113	125s

3.1.2 GRAPH



3.1.3 OBSERVATIONS

.

we can observe from the table that as x increases the runtime first decreases and then for some values it is increasing it is may be due to variation in the cpu utilization. And the utility is going to a saturation value.As utilization may or may not depend on the skip number we cannot comment on those values.

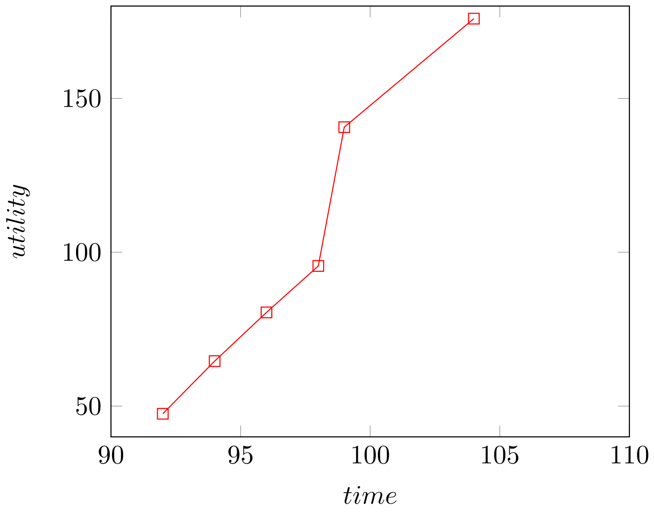
3.2 method2

.

TABLE

s.no	X.	Y.	utility	time
1	960	540	175.907	104s
2	640	360	140.636	99s
3	480	270	95.5367	98s
4	384	216	80.4402	96s
5	320	180	64.6069	94s
6	240	135	47.4998	92s

3.2.1 GRAPH



3.2.2 OBSERVATIONS

.

As we can observe from the table that decrease in the resolution of video (i.e. decreasing the size of each frame) then the runtime decreases because the time taken for computing the differed pixels from the image takes less time as the size of frame(Matrix) decreases.

And we can observe that utility decreases with decrease in resolution because with compression of image some data will be lost which will result in higher error,so decrease in utility.

.

3.3 method3

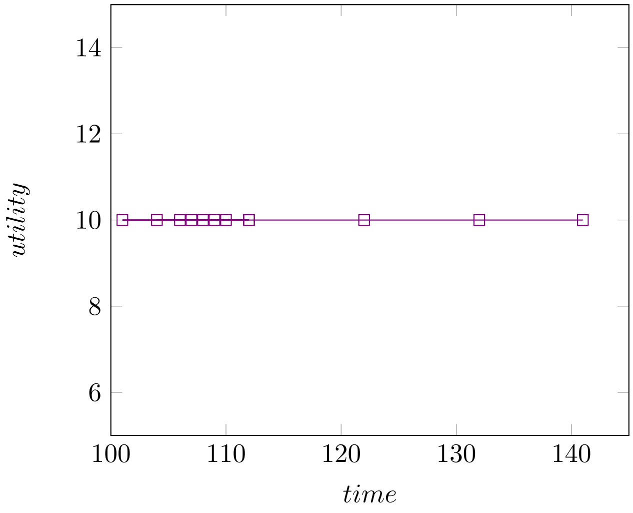
.

3.3.1 TABLE

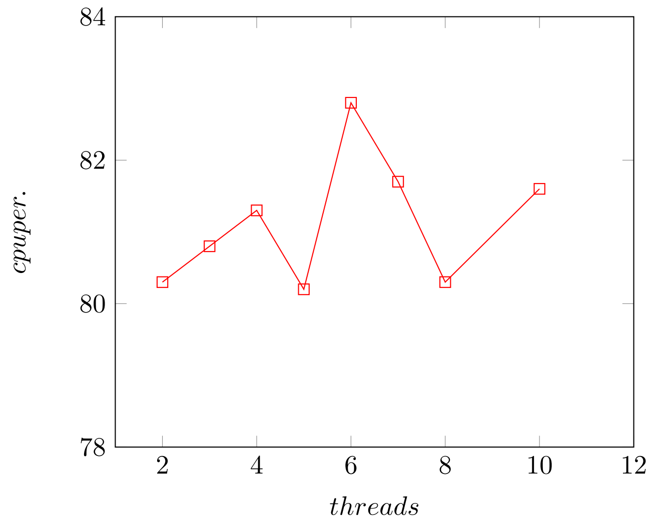
thds	time	utility	cpu per.	mem per.
2	112	10	80.3	3.64
3	101	10	80.8	3.64
4	106	10	81.3	3.65
5	108	10	80.2	3.65
6	107	10	82.8	3.65
7	110	10	81.7	3.65
8	112	10	80.3	3.65
16	104	10	81.6	3.65
25	109	10	80.0	3.67
40	122	10	75.3	3.68
60	132	10	71.6	3.70
90	141	10	70.0	3.70

.

3.3.2 GRAPHS



.



.

3.3.3 OBSERVATIONS

.

Utility is constant because there will be no error as we are taking the same frames. We can observe that cpu utilization is proportional to the no.of threads upto some threads.As no.of threads increases joining of threads takes time therefore cpu utilization decreases after a certain no.of threads.From that reason time decreases and then increases.

3.4 method4

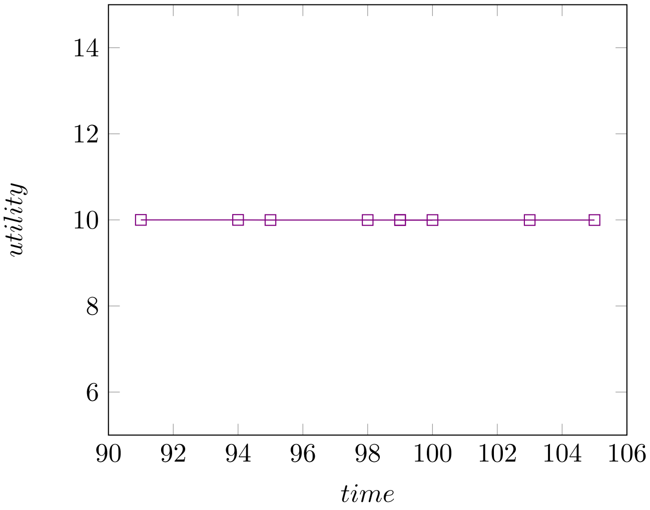
.

3.4.1 TABLE

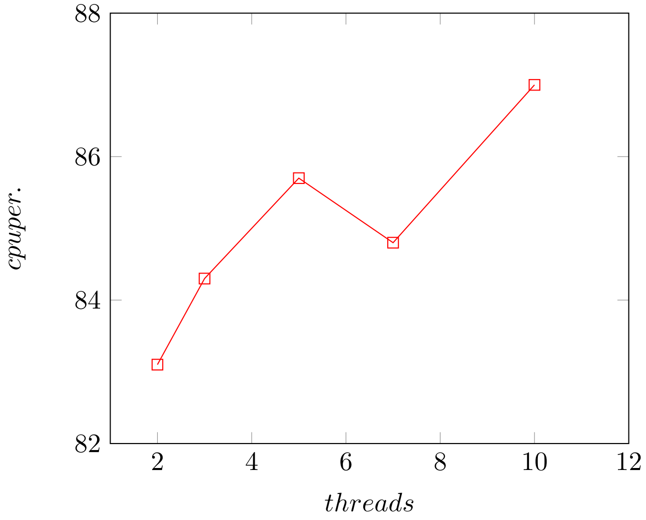
thds	time	utility	cpu per.	mem per.
2	91	10	83.1	5.66
3	94	10	84.3	5.57
5	95	9.997	85.7	5.68
7	98	9.997	84.8	5.77
10	99	9.997	87.0	6.05
20	100	9.997	85.4	6.70
40	99	9.997	86.3	7.91
60	103	9.997	84.8	9.17
100	105	9.997	709.4	11.76

.

3.4.2 GRAPHS



.



.

3.4.3 OBSERVATIONS

.

Utility is constant because there will be no error as we are taking the same frames. we can observe that cpu utilization increases with the no.of threads.As no.of threads increases joining takes time but x threads will run simultaneously so that time decreases.From that time is almost similar for different types of threads .