Assignment 3

5DV171

Jim Norman
C10jnn

2017-01-13

Code exist at:
https://github.com/jimnor/5dv171a3

Introduction

This program is made to run I/O operation based commands to see how the different I/O schedulers affects the test to see if one I/O scheduler is more effective than the other in the given environment.

Note that the test are done running running Ubuntu 16.04.1 LTS through "VirtualBox-5.1.10-112026-Win" on windows 10 which may affect the outcome quite a bit as it does not function like if you would install it on your computer. However the test data is still valid as it shows how it may be affected in this environment and what best to use when running it at a virtualbox.

The code for the program exist at https://github.com/jimnor/5dv171a3

Design

There are a few interesting design choices made for the program. First off to achieve a more balanced run, so all threads runs about the same time and not with threads that runs alone. A simple mutex lock has been used. This one prevents the threads from running until they main thread has created enough threads. Which allows for a more even run.

The other design choices is the two tasks, since as seen in the "test runs" the lighter task is a lot faster then the heavy task. This is simply because the heavy task is made out of I/O operations, which takes quite some time to perform. The simply open a file, write to it, read from it and then closes it.

The second task is much lighter, it is simple math operations (not math.h just * and / and such). However to not make the thread complete the task to fast they need to run the operation 10000 times before they get a lap (this is to remove a little bit of "noise factor").

Though due to the design of the program it is very sensitive to noise factors which will be seen in the test runs. As they are not consistent even after running for 30 sec. However after a couple of runs they do yield some valid data

The reason why it does not only run the I/O operation is simply to see if regular operations are affected by the different I/O operation schedular when running together with the I/O operations.

Conclusion

Now to the conclusion of the tests the “average throughput” and “the difference between the highest and lowest completed laps” has be compiled into two tables down below. So through studying them we can conclude a few things.

Deadline light	Deadline heavy	noop light	noop heavy	cfq light	cfq heavy
82811	535078	85966	369238	85227	590813
86925	602383	85248	586207	77382	530308
83780	579493	81515	492688	84364	579629
87070	597129	80098	550753	84816	580277

Table 1: average throughput

On average it seems that all three has the capacity to go through with about the same data when considering the light task (non I/O operations) making us not really see any remarkable data there. However when looking at the heavy task (the I/O operations) we can see that “noop” seem to be a bit more unstable than the others. Dropping so far as to almost half that of the others at one time. This could be due to noise but even so on average it is doing worst of the three.

Deadline light	Deadline heavy	noop light	noop heavy	cfq light	cfq heavy
748	8307	1030	9345	1431	16261
1772	6930	1111	9951	437	8014
1166	4997	3252	7865	1758	8912
732	9929	1263	3640	1518	15398

Table 2: the difference between highest and lowest completed laps.

Now when looking through the difference between the highest and the lowest we can see that they are all very uneven though the one that seems to be most unfair is the cfq while the most fair on average is Deadline when it comes to the heavy task(the I/O operation). When it comes to the non I/O operation it seems generally that it is not very affected as it goes up and down on all three. Though it seem to be the most fair on the deadline side and most unfair on the cfq.

So what can we conclude on this? Apart from that the clearly is fairly unstable when running through a virtualbox, it is quite obvious that the deadline seems to be the best one. Which is not surprising given that the deadline algorithm was the default one and thus should function the best with the given operative system

How to improve the I/O schedulers

With the given data we can see that the three I/O schedulers performs quite different but it is clear the cfq and the deadline is better than the noop scheduler. Since throughput is worth more than fairness in most scenarios and even when it comes to fairness it performed worse than deadline. But how do we improve them.

Well the noop scheduler does little to nothing when it comes to sorting. This clearly have affected its performance and why it is so random on the throughput as it is all based on what thread will come through at what time. This makes it fairly obvious that it needs to be using some sort of sorting like cfq or deadline.

Cfq scheduler is a bit more tricky to find a improvement since it does almost have the same throughput as the deadline, so close that it can be just scraped of as noise factor. However when we checked fairness between the threads we see a different thing. Here we see that it can become very unfair. This is probably due to the waiting times it has. As it can give an unfair advantage to some threads. Thus to improve the fairness these waiting times should be removed, however this probably won't improve the overall performance.

When it comes to deadline scheduler there is no direct thing that seems to be able to improve it's performance in this case. It outperform the others when using it here which gives no assistance to it's issues. Thus no obvious improvements can be made.

Test runs

These tests are has been done running Ubuntu 16.04.1 LTS “VirtualBox-5.1.10-112026-Win” at a computer with windows 10.

Deadline

```
*****************************************************************
Test will run with 10 threads during 30 seconds
Initiating test... wait a bit
Making threads
running test...
Test complete, displaying result

Average task laps completion for light task is: 82811
Lowest amount of completed laps for light task: 82401
Highest amount of completed laps for light task: 83149
Gap between highest and lowest completions for light task: 748

Average task laps completion for heavy task is: 535078
Lowest amount of completed laps for heavy task: 530132
Highest amount of completed laps for heavy task: 538439
Gap between highest and lowest completions for heavy task: 8307

*****************************************************************
Test will run with 10 threads during 30 seconds
Initiating test... wait a bit
Making threads
running test...
Test complete, displaying result

Average task laps completion for light task is: 86925
Lowest amount of completed laps for light task: 85950
Highest amount of completed laps for light task: 87722
Gap between highest and lowest completions for light task: 1772

Average task laps completion for heavy task is: 602383
Lowest amount of completed laps for heavy task: 599318
Highest amount of completed laps for heavy task: 606248
Gap between highest and lowest completions for heavy task: 6930

*****************************************************************
```

```
Test will run with 10 threads during 30 seconds
Initiating test... wait a bit
Making threads
running test...
Test complete, displaying result

Average task laps completion for light task is: 83780
Lowest amount of completed laps for light task: 83032
Highest amount of completed laps for light task: 84198
Gap between highest and lowest completions for light task: 1166

Average task laps completion for heavy task is: 579493
Lowest amount of completed laps for heavy task: 576060
Highest amount of completed laps for heavy task: 581057
Gap between highest and lowest completions for heavy task: 4997

******************************************************************
Test will run with 10 threads during 30 seconds
Initiating test... wait a bit
Making threads
running test...
Test complete, displaying result

Average task laps completion for light task is: 87070
Lowest amount of completed laps for light task: 86617
Highest amount of completed laps for light task: 87349
Gap between highest and lowest completions for light task: 732

Average task laps completion for heavy task is: 597129
Lowest amount of completed laps for heavy task: 591496
Highest amount of completed laps for heavy task: 601425
Gap between highest and lowest completions for heavy task: 9929

******************************************************************
```

NOOP

```
******************************************************************
Test will run with 10 threads during 30 seconds
Initiating test... wait a bit
Making threads
running test...
Test complete, displaying result

Average task laps completion for light task is: 85966
Lowest amount of completed laps for light task: 85525
Highest amount of completed laps for light task: 86555
Gap between highest and lowest completions for light task: 1030

Average task laps completion for heavy task is: 369238
Lowest amount of completed laps for heavy task: 364661
Highest amount of completed laps for heavy task: 374006
Gap between highest and lowest completions for heavy task: 9345

******************************************************************
Test will run with 10 threads during 30 seconds
Initiating test... wait a bit
Making threads
running test...
Test complete, displaying result

Average task laps completion for light task is: 85248
Lowest amount of completed laps for light task: 84677
Highest amount of completed laps for light task: 85788
Gap between highest and lowest completions for light task: 1111

Average task laps completion for heavy task is: 586207
Lowest amount of completed laps for heavy task: 581531
Highest amount of completed laps for heavy task: 591482
Gap between highest and lowest completions for heavy task: 9951
```

```
***************************************************************************
Test will run with 10 threads during 30 seconds
Initiating test... wait a bit
Making threads
running test...
Test complete, displaying result

Average task laps completion for light task is: 81515
Lowest amount of completed laps for light task: 79891
Highest amount of completed laps for light task: 83143
Gap between highest and lowest completions for light task: 3252

Average task laps completion for heavy task is: 492688
Lowest amount of completed laps for heavy task: 488128
Highest amount of completed laps for heavy task: 495993
Gap between highest and lowest completions for heavy task: 7865

***************************************************************************
Test will run with 10 threads during 30 seconds
Initiating test... wait a bit
Making threads
running test...
Test complete, displaying result

Average task laps completion for light task is: 80098
Lowest amount of completed laps for light task: 79374
Highest amount of completed laps for light task: 80637
Gap between highest and lowest completions for light task: 1263

Average task laps completion for heavy task is: 550753
Lowest amount of completed laps for heavy task: 548698
Highest amount of completed laps for heavy task: 552338
Gap between highest and lowest completions for heavy task: 3640
***************************************************************************
```

CFQ

```
******************************************************************
Test will run with 10 threads during 30 seconds
Initiating test... wait a bit
Making threads
running test...
Test complete, displaying result

Average task laps completion for light task is: 85227
Lowest amount of completed laps for light task: 84539
Highest amount of completed laps for light task: 85970
Gap between highest and lowest completions for light task: 1431

Average task laps completion for heavy task is: 590813
Lowest amount of completed laps for heavy task: 579126
Highest amount of completed laps for heavy task: 595387
Gap between highest and lowest completions for heavy task: 16261

******************************************************************
Test will run with 10 threads during 30 seconds
Initiating test... wait a bit
Making threads
running test...
Test complete, displaying result

Average task laps completion for light task is: 77382
Lowest amount of completed laps for light task: 77202
Highest amount of completed laps for light task: 77639
Gap between highest and lowest completions for light task: 437

Average task laps completion for heavy task is: 530308
Lowest amount of completed laps for heavy task: 526154
Highest amount of completed laps for heavy task: 534168
Gap between highest and lowest completions for heavy task: 8014
```

```
******************************************************************
Test will run with 10 threads during 30 seconds
Initiating test... wait a bit
Making threads
running test...
Test complete, displaying result

Average task laps completion for light task is: 84364
Lowest amount of completed laps for light task: 83730
Highest amount of completed laps for light task: 85488
Gap between highest and lowest completions for light task: 1758

Average task laps completion for heavy task is: 579629
Lowest amount of completed laps for heavy task: 575118
Highest amount of completed laps for heavy task: 584030
Gap between highest and lowest completions for heavy task: 8912

******************************************************************
Test will run with 10 threads during 30 seconds
Initiating test... wait a bit
Making threads
running test...
Test complete, displaying result

Average task laps completion for light task is: 84816
Lowest amount of completed laps for light task: 84234
Highest amount of completed laps for light task: 85752
Gap between highest and lowest completions for light task: 1518

Average task laps completion for heavy task is: 580277
Lowest amount of completed laps for heavy task: 570666
Highest amount of completed laps for heavy task: 586064
Gap between highest and lowest completions for heavy task: 15398

********************************************************************
```