# A detour into the world of Linux schedulers

**Abstract:**

I explored 3 different schedulers implemented in Linux and how they perform on different processes. We tested Round Robin, First-In-First-Out and the CFS scheduler with I/O bound processes, CPU-bound processes and a mixture of those processes and charted their performance. I found that CFS faltered when it came to IO-intensive processes whilst outperforming RR and FIFO on CPU-intensive processes. RR and FIFO performed relatively similarly.

**Introduction:**

In this report, I explored various schedulers available within the Linux OS; round robin (SCHED_RR), FIFO (SCHED_FIFO) and CFS (SCHED_OTHER)**.** A scheduler is crucial to the running of a modern OS. Schedulers are often implemented so they keep all compute resources busy (as in load balancing), allow multiple users to share system resources effectively, or to achieve a target quality of service. Scheduling is fundamental to computation itself, and an intrinsic part of the execution model of a computer system; the concept of scheduling makes it possible to have computer multitasking with a single central processing unit (CPU).[1]

The tests were performed on a 64-bit Ubuntu 12.04 VM allocated 5728 MB with 4 cores of a 2.7 GHz Intel Core i5.

**Method:**

To thoroughly test the schedulers, we needed to establish certain processes. I created a CPU-intensive process that calculates pi and an I/O-bound process that reads an input file and writes data to a number of outputs. I then combined these into a testing C file that requires several parameters to perform its task. We pass in the scheduler policy (S, the number of processes we will fork and the type of processes we'd like to test. Given these parameters, our program runs the required task.

e.g.

```
$ ./test-sched SCHED_FIFO SMALL IO
$ ./test-sched SCHED_OTHER MEDIUM CPU
```

<SCHEDULER>: SCHED_FIFO, SCHED_RR, SCHED_OTHER (DEFAULT)
<# OF ITERATIONS>: LOW (DEFAULT) (10), MEDIUM(100), HIGH(1000)
<PROCESSTYPE>: CPU, IO, MIXED

To test our programs and their combinations systematically, we wrote a bash script that iterates through all of the combinations presented. We had 3 schedulers, 3 amounts of processes and 3 types of process which results in $3^3$ combinations, or 27 unique combinations. We ran the tests 5 times each to account for discrepancies and average them out. We run 500,000 iterations of every process. The bash script tests them in order and returns data in a

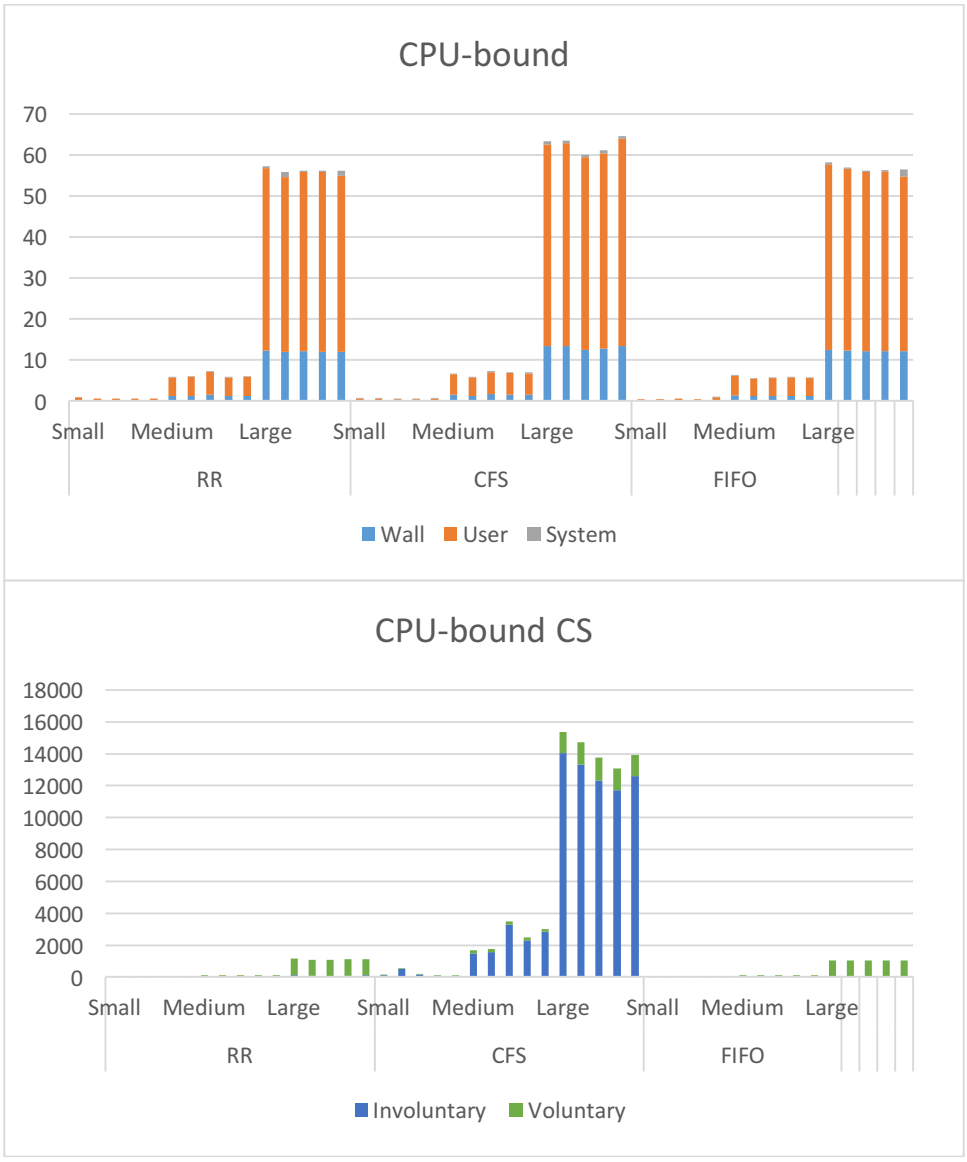
[1] https://en.wikipedia.org/wiki/Scheduling_(computing)

text file for easy visualization using spreadsheet software. Without further ado, let's move onto the results section!

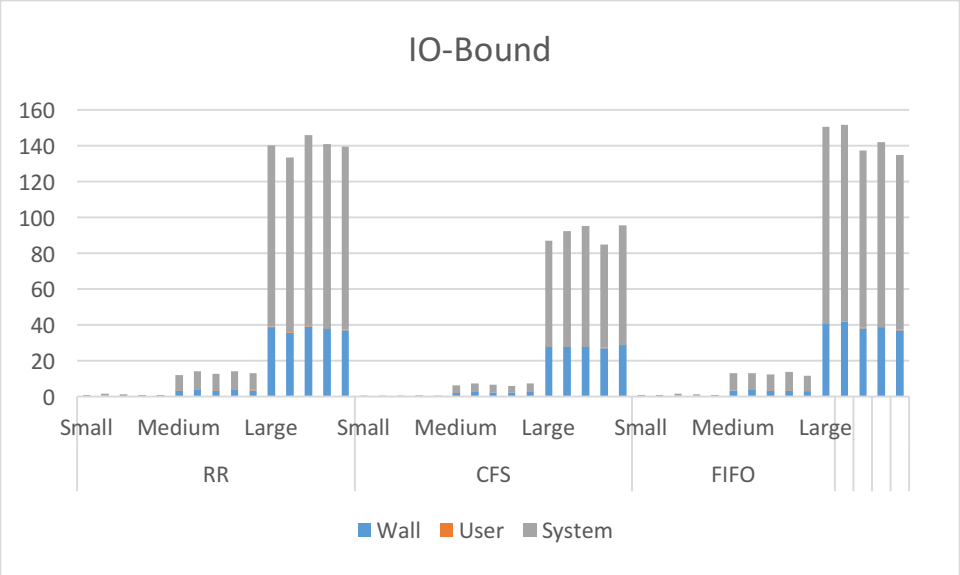
**Results:**

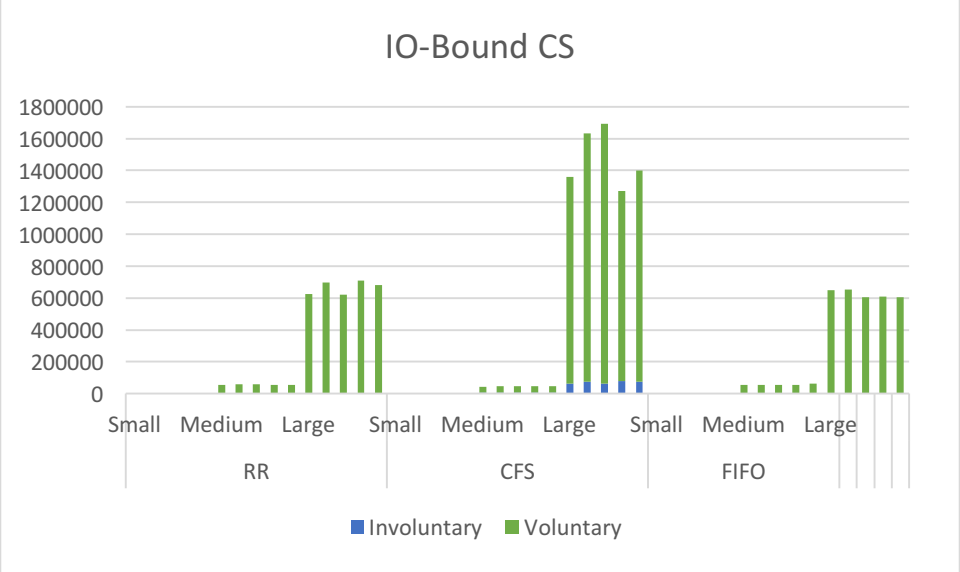
The graphs are vertical GANTT charts. We show the run times of the processes in a chart and the amounts of voluntary/involuntary context switches in the following chart.

CPU-intensive:

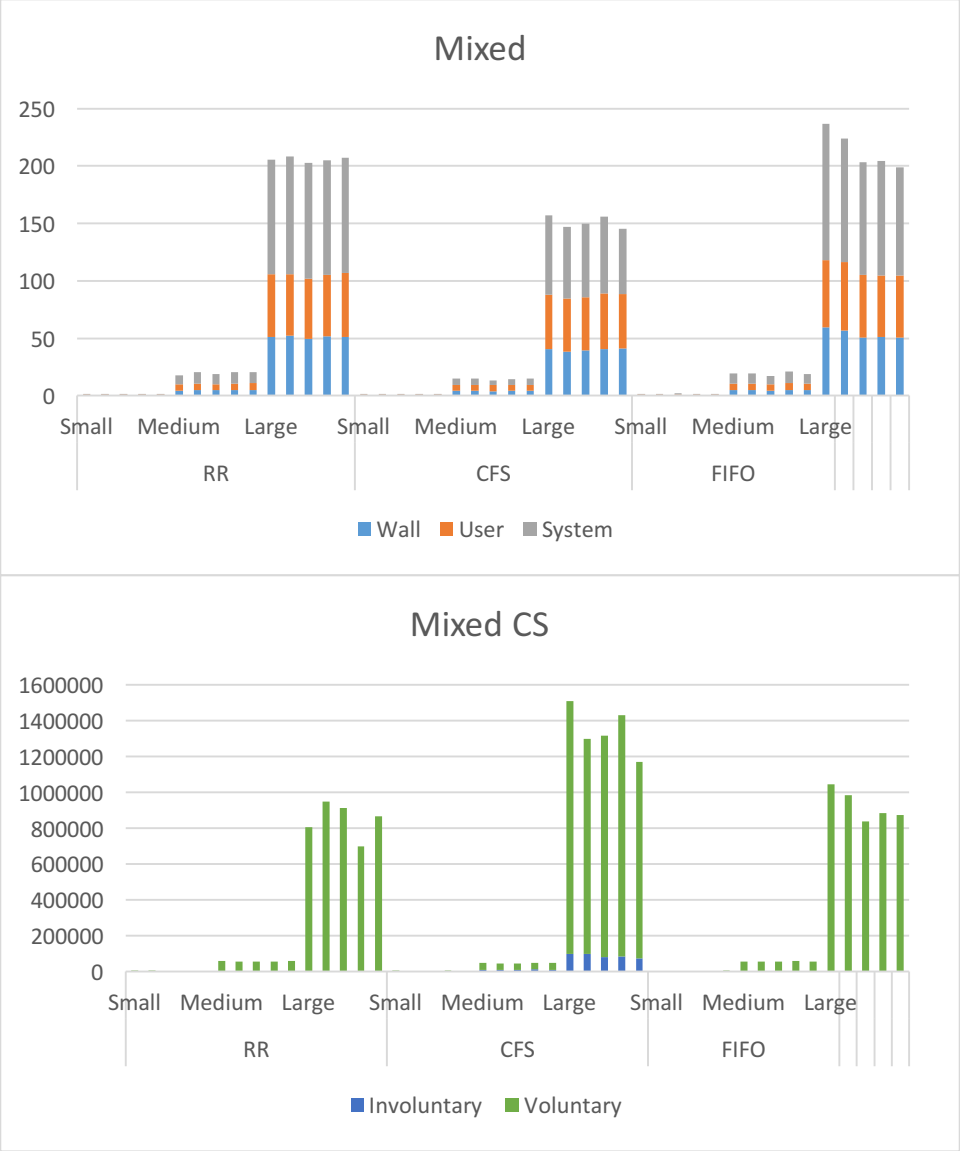


IO-Bound:


IO-Bound
160
140
120
100
80
60
40
20
0
Small
Medium
Large
Small
Medium
Large
Small
Medium
Large
RR
CFS
FIFO
Wall
User
System


IO-Bound CS
1800000
1600000
1400000
1200000
1000000
800000
600000
400000
200000
0
Small
Medium
Large
Small
Medium
Large
Small
Medium
Large
RR
CFS
FIFO
Involuntary
Voluntary

Mixed:




**Analysis:**

To begin, we'll focus on the CPU-bound processes. We notice that all three of the schedulers performed almost equally well. However, looking at the context switch chart for CPU-bound processes shows a radically different picture. The CFS spends relatively massive amounts doing involuntary context switches. They're understandably involuntary, as the CPU's won't yield to the scheduler whilst there's still a task to be completed. Despite those large wait times, it still performed on almost equal footing as RR and FIFO (fractionally slower). We conclude that RR and FIFO are marginally better at CPU-intensive tasks.

On the other hand, in the I/O bound processes we notice that despite having more than double the amount of context switches, the CFS is about 40% quicker than both RR and FIFO which again are almost equal. The majority of the context switches are voluntary, as expected

when dealing with I/O devices. In this case, CFS outperforms both RR and FIFO scheduling and is our clear winner.

Lastly, our mixed process results are predictable given our previous results. Since CFS is almost on par with RR and FIFO in CPU-Intensive processes and 40% quicker in I/O bound processes, then our mixed process results show that CFS is about 25% quicker than both RR and FIFO despite have 25-30% more context switches.

**Conclusion:**

In conclusion, we see that CFS outperforms both RR and FIFO schedulers in I/O bound processes and mixed processes which most accurately reflect personal real-world applications. Both RR and FIFO are on par with each other and outperform CFS in CPU-intensive processes, though suffer serious drawbacks with I/O bound processes.

Building off our results, CFS would be most efficient for modern-day personal computing whilst RR/FIFO would work best in scientific computing that utilize heavy CPU usage with little I/O sequences. RR would falter heavily in interactive applications, which CFS can handle. CFS would increase wait times substantially given a really CPU intensive scientific computation that spans weeks/years.

We notice that CFS doesn't scale very well in the CPU-intensive tasks, with the wait times increasing near exponentially while RR/FIFO understandably remain linear in their growth.

**References:**

Andy Sayler
Class Lectures
Operating Systems Concepts – Silverschatz

**Appendix A:**

| | Scheduler | Processes | Wall | User | System | CPU | Involuntary | Voluntary |
|---|---|---|---|---|---|---|---|---|
| CPU | RR | Small | 0.21 | 0.65 | 0 | 309% | 3 | 18 |
| | | | 0.14 | 0.46 | 0 | 321% | 4 | 18 |
| | | | 0.15 | 0.48 | 0 | 320% | 4 | 18 |
| | | | 0.14 | 0.46 | 0 | 331% | 3 | 18 |
| | | | 0.15 | 0.49 | 0 | 324% | 5 | 18 |
| | | Medium | 1.22 | 4.5 | 0.01 | 368% | 6 | 107 |
| | | | 1.26 | 4.63 | 0.02 | 368% | 9 | 108 |
| | | | 1.54 | 5.56 | 0.04 | 363% | 12 | 108 |
| | | | 1.25 | 4.57 | 0.02 | 367% | 9 | 108 |
| | | | 1.26 | 4.68 | 0.02 | 371% | 8 | 109 |
| | | Large | 12.29 | 44.54 | 0.38 | 365% | 127 | 1041 |
| | | | 12.04 | 42.64 | 1.19 | 364% | 62 | 1009 |
| | | | 12.13 | 43.79 | 0.32 | 363% | 58 | 1011 |
| | | | 12.09 | 43.74 | 0.35 | 364% | 88 | 1025 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | 12.08 | 42.93 | 1.13 | 364% | 134 | 1008 |
| | CFS | Small | 0.18 | 0.5 | 0.01 | 277% | 135 | 24 |
| | | | 0.18 | 0.39 | 0.01 | 218% | 520 | 23 |
| | | | 0.14 | 0.36 | 0.01 | 251% | 164 | 24 |
| | | | 0.12 | 0.38 | 0.01 | 322% | 64 | 23 |
| | | | 0.13 | 0.43 | 0.01 | 342% | 74 | 24 |
| | | Medium | 1.53 | 5.06 | 0.12 | 337% | 1493 | 193 |
| | | | 1.3 | 4.53 | 0.08 | 354% | 1566 | 197 |
| | | | 1.77 | 5.32 | 0.23 | 313% | 3303 | 201 |
| | | | 1.58 | 5.29 | 0.18 | 347% | 2305 | 199 |
| | | | 1.58 | 5.19 | 0.24 | 342% | 2834 | 187 |
| | | Large | 13.46 | 49.11 | 0.84 | 371% | 14032 | 1330 |
| | | | 13.38 | 49.49 | 0.64 | 374% | 13337 | 1382 |
| | | | 12.56 | 46.9 | 0.56 | 377% | 12334 | 1450 |
| | | | 12.77 | 47.59 | 0.77 | 378% | 11712 | 1389 |
| | | | 13.49 | 50.43 | 0.72 | 379% | 12598 | 1314 |
| | FIFO | Small | 0.12 | 0.39 | 0 | 330% | 4 | 18 |
| | | | 0.12 | 0.39 | 0 | 316% | 6 | 18 |
| | | | 0.14 | 0.46 | 0 | 316% | 5 | 18 |
| | | | 0.12 | 0.4 | 0 | 323% | 6 | 16 |
| | | | 0.22 | 0.72 | 0.02 | 331% | 10 | 17 |
| | | Medium | 1.34 | 4.88 | 0.04 | 366% | 6 | 108 |
| | | | 1.18 | 4.29 | 0.02 | 365% | 8 | 108 |
| | | | 1.2 | 4.4 | 0.04 | 368% | 9 | 108 |
| | | | 1.22 | 4.49 | 0.03 | 369% | 8 | 108 |
| | | | 1.23 | 4.41 | 0.02 | 358% | 10 | 107 |
| | | Large | 12.47 | 45.26 | 0.41 | 366% | 48 | 1008 |
| | | | 12.28 | 44.4 | 0.3 | 364% | 53 | 1008 |
| | | | 12.1 | 43.8 | 0.36 | 364% | 48 | 1009 |
| | | | 12.12 | 43.85 | 0.34 | 364% | 52 | 1008 |
| | | | 12.13 | 42.62 | 1.67 | 364% | 50 | 1008 |
| IO | Scheduler | Processes | Wall | User | System | CPU | Involuntary | Voluntary |
| | RR | Small | 0.41 | 0.01 | 0.61 | 150% | 6 | 4015 |
| | | | 0.51 | 0 | 1.07 | 208% | 6 | 3401 |
| | | | 0.44 | 0 | 0.8 | 183% | 5 | 3772 |
| | | | 0.34 | 0 | 0.72 | 209% | 4 | 3811 |
| | | | 0.36 | 0 | 0.65 | 179% | 5 | 3622 |
| | | Medium | 3.35 | 0.06 | 8.76 | 262% | 15 | 55247 |
| | | | 3.8 | 0.01 | 10.38 | 273% | 19 | 60543 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | 3.47 | 0.05 | 9.36 | 271% | 20 | 59868 |
| | | | 3.79 | 0.02 | 10.3 | 272% | 29 | 55824 |
| | | | 3.66 | 0.02 | 9.29 | 254% | 19 | 56583 |
| | | Large | 38.65 | 0.34 | 101.24 | 262% | 502 | 623868 |
| | | | 35.68 | 0.26 | 97.56 | 274% | 469 | 697109 |
| | | | 39.31 | 0.32 | 106.16 | 270% | 429 | 620903 |
| | | | 37.56 | 0.28 | 102.97 | 274% | 415 | 708244 |
| | | | 37.04 | 0.3 | 102.08 | 276% | 416 | 679669 |
| | CFS | Small | 0.32 | 0 | 0.31 | 99% | 561 | 2360 |
| | | | 0.28 | 0 | 0.3 | 105% | 529 | 3092 |
| | | | 0.2 | 0 | 0.18 | 90% | 291 | 2991 |
| | | | 0.37 | 0.01 | 0.38 | 107% | 397 | 2791 |
| | | | 0.19 | 0 | 0.1 | 55% | 258 | 2566 |
| | | Medium | 2.52 | 0.02 | 3.64 | 145% | 7636 | 37344 |
| | | | 2.88 | 0.02 | 4.5 | 157% | 8363 | 36996 |
| | | | 2.58 | 0 | 3.97 | 154% | 7599 | 40435 |
| | | | 2.47 | 0 | 3.65 | 147% | 7712 | 38795 |
| | | | 2.84 | 0.02 | 4.64 | 164% | 8252 | 39979 |
| | | Large | 28.05 | 0.1 | 58.98 | 210% | 63009 | 1297221 |
| | | | 27.59 | 0.12 | 64.68 | 234% | 73939 | 1557439 |
| | | | 27.99 | 0.12 | 67.13 | 240% | 63745 | 1628215 |
| | | | 27.2 | 0.18 | 57.28 | 211% | 79247 | 1191711 |
| | | | 28.75 | 0.2 | 66.61 | 232% | 76711 | 1323476 |
| | FIFO | Small | 0.37 | 0 | 0.66 | 177% | 5 | 3639 |
| | | | 0.37 | 0 | 0.62 | 165% | 6 | 3640 |
| | | | 0.59 | 0 | 1.26 | 213% | 3 | 3498 |
| | | | 0.48 | 0 | 1 | 209% | 5 | 3723 |
| | | | 0.34 | 0 | 0.64 | 190% | 7 | 4207 |
| | | Medium | 3.52 | 0 | 9.42 | 267% | 18 | 55853 |
| | | | 3.72 | 0.05 | 9.41 | 253% | 11 | 56884 |
| | | | 3.38 | 0.05 | 8.9 | 265% | 11 | 56899 |
| | | | 3.54 | 0.06 | 10.1 | 287% | 13 | 55935 |
| | | | 3.17 | 0.05 | 8.34 | 264% | 11 | 64608 |
| | | Large | 40.84 | 0.27 | 109.29 | 268% | 94 | 648636 |
| | | | 41.58 | 0.31 | 109.6 | 264% | 130 | 651701 |
| | | | 38.08 | 0.32 | 99.02 | 260% | 109 | 603797 |
| | | | 38.64 | 0.23 | 102.92 | 266% | 102 | 609987 |
| | | | 36.96 | 0.28 | 97.68 | 265% | 116 | 605897 |
| Mixed | Scheduler | Processes | Wall | User | System | CPU | Involuntary | Voluntary |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Small | 0.43 | 0.56 | 0.3 | 198% | 3 | 4837 |
| | | | 0.37 | 0.56 | 0.22 | 208% | 8 | 4477 |
| | | | 0.51 | 0.54 | 0.5 | 205% | 4 | 3578 |
| | | | 0.39 | 0.56 | 0.31 | 219% | 5 | 3927 |
| | | | 0.37 | 0.49 | 0.26 | 203% | 6 | 4172 |
| | | Medium | 4.5 | 5.58 | 7.61 | 292% | 65 | 60114 |
| | | | 5.03 | 5.38 | 10.08 | 307% | 64 | 56209 |
| | RR | | 4.6 | 5.37 | 8.98 | 311% | 49 | 55996 |
| | | | 4.91 | 5.75 | 9.67 | 313% | 58 | 57532 |
| | | | 4.85 | 5.91 | 9.51 | 318% | 73 | 57920 |
| | | Large | 51.33 | 54.66 | 99.82 | 300% | 651 | 803791 |
| | | | 52.24 | 53.74 | 102.53 | 299% | 684 | 946067 |
| | | | 49.71 | 52.32 | 100.6 | 307% | 687 | 911065 |
| | | | 51.69 | 53.78 | 99.79 | 297% | 630 | 697404 |
| | | | 51.15 | 55.65 | 100.46 | 305% | 684 | 866909 |
| | | Small | 0.48 | 0.51 | 0.33 | 176% | 852 | 2972 |
| | | | 0.49 | 0.5 | 0.46 | 192% | 684 | 2930 |
| | | | 0.4 | 0.46 | 0.31 | 194% | 556 | 3113 |
| | | | 0.4 | 0.46 | 0.46 | 228% | 1036 | 3108 |
| | | | 0.54 | 0.56 | 0.69 | 232% | 715 | 2923 |
| | | Medium | 4.16 | 5.04 | 5.47 | 252% | 9792 | 39081 |
| | | | 4.31 | 5.11 | 5.36 | 242% | 11540 | 33295 |
| | CFS | | 3.71 | 5.35 | 4.15 | 256% | 9051 | 37001 |
| | | | 4.12 | 5.18 | 5.31 | 254% | 11599 | 37835 |
| | | | 4.27 | 4.94 | 5.76 | 250% | 10717 | 37622 |
| | | Large | 40.49 | 47.38 | 69.45 | 288% | 100158 | 1409292 |
| | | | 38.43 | 46.15 | 62.58 | 282% | 99389 | 1198817 |
| | | | 39.24 | 46.38 | 64.13 | 281% | 81040 | 1235369 |
| | | | 40.77 | 48.28 | 67.09 | 282% | 83025 | 1348386 |
| | | | 40.86 | 47.36 | 57.02 | 255% | 73862 | 1095514 |
| | | Small | 0.48 | 0.54 | 0.64 | 247% | 6 | 3926 |
| | | | 0.46 | 0.56 | 0.6 | 247% | 6 | 3853 |
| | | | 0.53 | 0.52 | 0.88 | 265% | 4 | 4032 |
| | | | 0.38 | 0.47 | 0.5 | 255% | 5 | 3866 |
| | FIFO | | 0.36 | 0.52 | 0.28 | 217% | 5 | 4656 |
| | | Medium | 4.83 | 5.78 | 8.82 | 302% | 11 | 55200 |
| | | | 4.84 | 5.54 | 9.04 | 301% | 16 | 56786 |
| | | | 4.36 | 5.34 | 7.59 | 296% | 11 | 55481 |
| | | | 5.1 | 5.68 | 10.15 | 310% | 15 | 60247 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | 4.61 | 5.66 | 8.34 | 303% | 10 | 54628 |
| | | Large | 59.74 | 58.1 | 118.67 | 295% | 139 | 1042543 |
| | | | 56.47 | 59.81 | 107.4 | 296% | 149 | 984459 |
| | | | 50.66 | 54.39 | 98.38 | 301% | 156 | 837963 |
| | | | 51.13 | 53.29 | 100.15 | 300% | 145 | 881949 |
| | | | 50.4 | 54.02 | 94.31 | 294% | 161 | 873904 |

**Appendix B:**

- test-sched.c :- runs the various processes required for testing
- testscript :- runs a systematic analysis of all 27 cases