SIT315 M2.T2C: Complex Threading With OpenMP

Codey Funston | S222250824

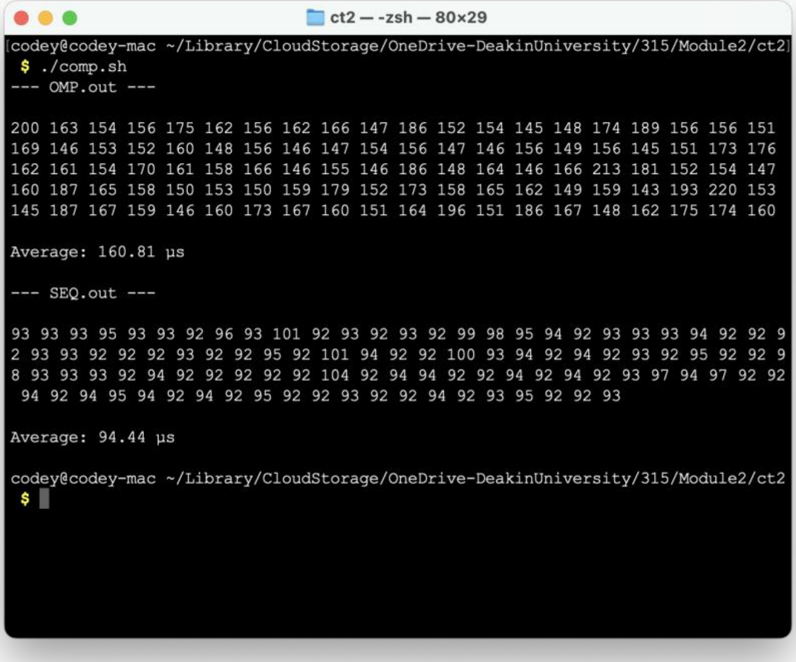
Decomposition and Performance Comparison

I decided to implement a quicksort algorithm that used recursion as that seemed to work better conceptually with parallelising the program. The function runs its body and then has two recursive calls, splitting the data in two based on a random pivot element. I used OpenMP to assign a thread to each call and to not use a barrier since both calls work on separate parts of the data that do not cross over.

The runtime of the sequential program wasn't that bad given that the time complexity for quicksort is $O(nlog(n))$. For very small data input the sequential version runs faster, for example with a vector of 2000 integers, over 100 tests for each program type the average for the sequential version was 96.51 microseconds, compared to the OMP version's average of 166.02 microseconds. This is because the threading overhead is much too high. To help reduce the overhead I added a condition based on testing output to only assign a thread to a recursive call if the section of data being worked on consisted of more than 1000 integers.

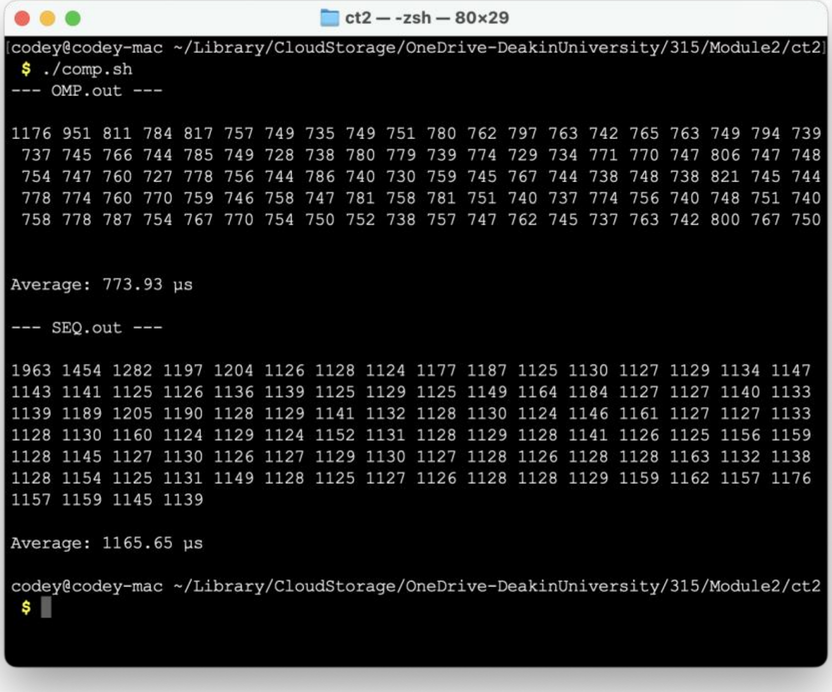
Some Results and Corresponding Input Data Size

- 2,000



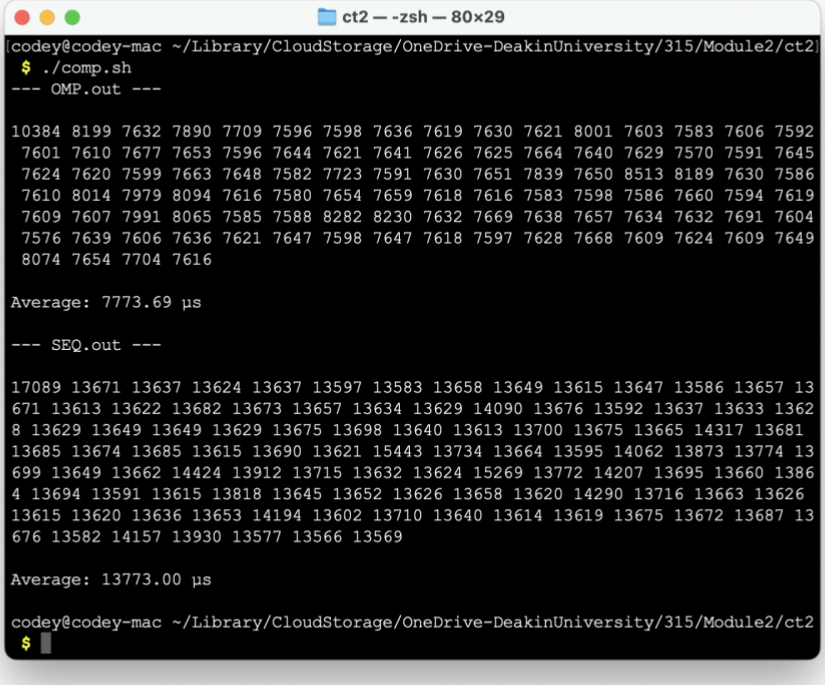
```
[codey@codey-mac ~/Library/CloudStorage/OneDrive-DeakinUniversity/315/Module2/ct2]
 $ ./comp.sh
--- OMP.out ---

200 163 154 156 175 162 156 162 166 147 186 152 154 145 148 174 189 156 156 151
169 146 153 152 160 148 156 146 147 154 156 147 146 156 149 156 145 151 173 176
162 161 154 170 161 158 166 146 155 146 186 148 164 146 166 213 181 152 154 147
160 187 165 158 150 153 150 159 179 152 173 158 165 162 149 159 143 193 220 153
145 187 167 159 146 160 173 167 160 151 164 196 151 186 167 148 162 175 174 160

Average: 160.81 µs

--- SEQ.out ---

93 93 93 95 93 93 92 96 93 101 92 93 92 93 92 99 98 95 94 92 93 93 93 94 92 92 9
2 93 93 92 92 92 93 92 92 95 92 101 94 92 92 100 93 94 92 94 92 93 92 95 92 92 9
8 93 93 93 92 94 92 92 92 92 92 104 92 94 94 92 92 94 92 94 92 93 97 94 97 92 92
 94 92 94 95 94 92 94 92 95 92 92 93 92 92 94 92 93 95 92 92 93

Average: 94.44 µs

codey@codey-mac ~/Library/CloudStorage/OneDrive-DeakinUniversity/315/Module2/ct2
 $ 
```

- 20,000 (OMP is slightly faster now)



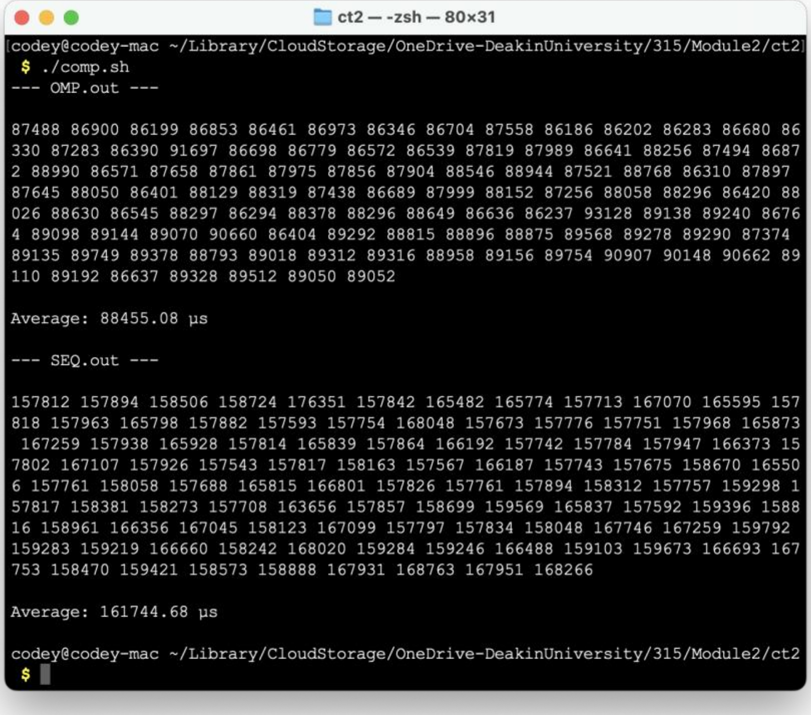
```
[codey@codey-mac ~/Library/CloudStorage/OneDrive-DeakinUniversity/315/Module2/ct2]
 $ ./comp.sh
--- OMP.out ---

1176 951 811 784 817 757 749 735 749 751 780 762 797 763 742 765 763 749 794 739
 737 745 766 744 785 749 728 738 780 779 739 774 729 734 771 770 747 806 747 748
 754 747 760 727 778 756 744 786 740 730 759 745 767 744 738 748 738 821 745 744
 778 774 760 770 759 746 758 747 781 758 781 751 740 737 774 756 740 748 751 740
 758 778 787 754 767 770 754 750 752 738 757 747 762 745 737 763 742 800 767 750


Average: 773.93 µs

--- SEQ.out ---

1963 1454 1282 1197 1204 1126 1128 1124 1177 1187 1125 1130 1127 1129 1134 1147
1143 1141 1125 1126 1136 1139 1125 1129 1125 1149 1164 1184 1127 1127 1140 1133
1139 1189 1205 1190 1128 1129 1141 1132 1128 1130 1124 1146 1161 1127 1127 1133
1128 1130 1160 1124 1129 1124 1152 1131 1128 1129 1128 1141 1126 1125 1156 1159
1128 1145 1127 1130 1126 1127 1129 1130 1127 1128 1126 1128 1128 1163 1132 1138
1128 1154 1125 1131 1149 1128 1125 1127 1126 1128 1128 1129 1159 1162 1157 1176
1157 1159 1145 1139

Average: 1165.65 µs

codey@codey-mac ~/Library/CloudStorage/OneDrive-DeakinUniversity/315/Module2/ct2
 $ 
```

- 200,000 (OMP is now almost twice as fast)



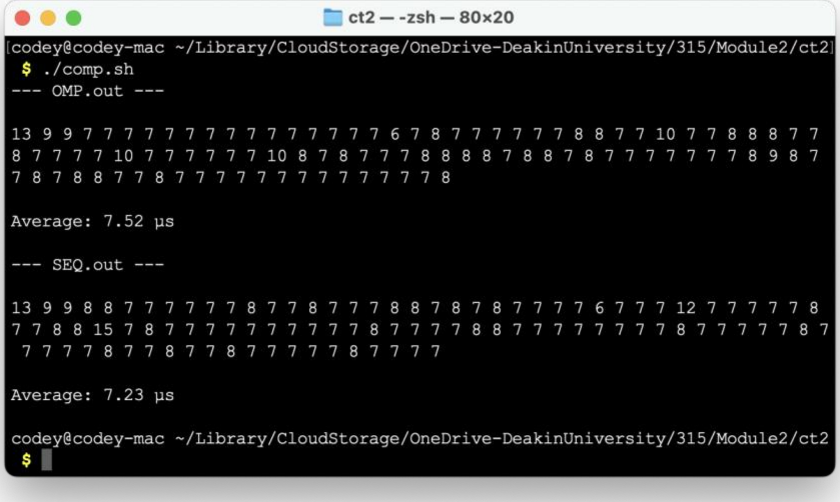
```
[codey@codey-mac ~/Library/CloudStorage/OneDrive-DeakinUniversity/315/Module2/ct2]
 $ ./comp.sh
--- OMP.out ---

10384 8199 7632 7890 7709 7596 7598 7636 7619 7630 7621 8001 7603 7583 7606 7592
 7601 7610 7677 7653 7596 7644 7621 7641 7626 7625 7664 7640 7629 7570 7591 7645
 7624 7620 7599 7663 7648 7582 7723 7591 7630 7651 7839 7650 8513 8189 7630 7586
 7610 8014 7979 8094 7616 7580 7654 7659 7618 7616 7583 7598 7586 7660 7594 7619
 7609 7607 7991 8065 7585 7588 8282 8230 7632 7669 7638 7657 7634 7632 7691 7604
 7576 7639 7606 7636 7621 7647 7598 7647 7618 7597 7628 7668 7609 7624 7609 7649
 8074 7654 7704 7616

Average: 7773.69 µs

--- SEQ.out ---

17089 13671 13637 13624 13637 13597 13583 13658 13649 13615 13647 13586 13657 13
671 13613 13622 13682 13673 13657 13634 13629 14090 13676 13592 13637 13633 1362
8 13629 13649 13649 13629 13675 13698 13640 13613 13700 13675 13665 14317 13681
13685 13674 13685 13615 13690 13621 15443 13734 13664 13595 14062 13873 13774 13
699 13649 13662 14424 13912 13715 13632 13624 15269 13772 14207 13695 13660 1386
4 13694 13591 13615 13818 13645 13652 13626 13658 13620 14290 13716 13663 13626
13615 13620 13636 13653 14194 13602 13710 13640 13614 13619 13675 13672 13687 13
676 13582 14157 13930 13577 13566 13569

Average: 13773.00 µs

codey@codey-mac ~/Library/CloudStorage/OneDrive-DeakinUniversity/315/Module2/ct2
 $ 
```

- 2000000 (no more improvement is seen now from OMP)



```
[codey@codey-mac ~/Library/CloudStorage/OneDrive-DeakinUniversity/315/Module2/ct2]
 $ ./comp.sh
--- OMP.out ---

87488 86900 86199 86853 86461 86973 86346 86704 87558 86186 86202 86283 86680 86
330 87283 86390 91697 86698 86779 86572 86539 87819 87989 86641 88256 87494 8687
2 88990 86571 87658 87861 87975 87856 87904 88546 88944 87521 88768 86310 87897
87645 88050 86401 88129 88319 87438 86689 87999 88152 87256 88058 88296 86420 88
026 88630 86545 88297 86294 88378 88296 88649 86636 86237 93128 89138 89240 8676
4 89098 89144 89070 90660 86404 89292 88815 88896 88875 89568 89278 89290 87374
89135 89749 89378 88793 89018 89312 89316 88958 89156 89754 90907 90148 90662 89
110 89192 86637 89328 89512 89050 89052

Average: 88455.08 µs

--- SEQ.out ---

157812 157894 158506 158724 176351 157842 165482 165774 157713 167070 165595 157
818 157963 165798 157882 157593 157754 168048 157673 157776 157751 157968 165873
 167259 157938 165928 157814 165839 157864 166192 157742 157784 157947 166373 15
7802 167107 157926 157543 157817 158163 157567 166187 157743 157675 158670 16550
6 157761 158058 157688 165815 166801 157826 157761 157894 158312 157757 159298 1
57817 158381 158273 157708 163656 157857 158699 159569 165837 157592 159396 1588
16 158961 166356 167045 158123 167099 157797 157834 158048 167746 167259 159792
159283 159219 166660 158242 168020 159284 159246 166488 159103 159673 166693 167
753 158470 159421 158573 158888 167931 168763 167951 168266

Average: 161744.68 µs

codey@codey-mac ~/Library/CloudStorage/OneDrive-DeakinUniversity/315/Module2/ct2
 $ 
```

- 100 (the OMP version’s safeguard for data with less than 1000 integers means that for this incredibly low data size the program is run sequentially)



```
[codey@codey-mac ~/Library/CloudStorage/OneDrive-DeakinUniversity/315/Module2/ct2]
 $ ./comp.sh
--- OMP.out ---

13 9 9 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 6 7 8 7 7 7 7 7 7 8 8 7 7 10 7 7 8 8 8 7 7
8 7 7 7 7 10 7 7 7 7 7 7 10 8 7 8 7 7 7 8 8 8 8 7 8 8 7 8 7 7 7 7 7 7 7 8 9 8 7
7 8 7 8 8 7 7 8 7 7 7 7 7 7 7 7 7 7 7 7 7 8

Average: 7.52 µs

--- SEQ.out ---

13 9 9 8 8 7 7 7 7 7 7 8 7 7 8 7 7 7 8 8 7 8 7 8 7 7 7 7 6 7 7 7 12 7 7 7 7 7 8
7 7 8 8 15 7 8 7 7 7 7 7 7 7 7 7 7 8 7 7 7 7 8 8 7 7 7 7 7 7 7 7 8 7 7 7 7 7 8 7
 7 7 7 7 8 7 7 8 7 7 8 7 7 7 7 7 8 7 7 7 7

Average: 7.23 µs

codey@codey-mac ~/Library/CloudStorage/OneDrive-DeakinUniversity/315/Module2/ct2
 $ 
```