CPSC424

Lab2

Bo Song

Task1 Serial program performance

N TIME (secs)

----- -------------

1000 0.1797

2000 2.2810

4000 19.3756

8000 153.9208

12000 518.4189

Task 2 Parallel program performance

Raw performance table, p = 8

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | 1 node | | 2 nodes | | 4 nodes | |
| #Proc |  | N = 8000 | N=12000 | N = 8000 | N= 12000 | N=8000 | N=12000 |
| 0 | Comm time | 47.728039 | 168.069686 | 51.669500 | 139.064581 | 50.161624 | 133.829175 |
| Comp time | 6.981728 | 23.841349 | 3.781539 | 15.188161 | 2.404897 | 11.083201 |
| 1 | Comm time | 30.928616 | 114.894897 | 36.895063 | 99.661204 | 35.560373 | 96.545476 |
| Comp time | 18.261596 | 55.297461 | 10.245202 | 36.412500 | 9.167001 | 33.146939 |
| 2 | Comm time | 24.660388 | 85.390493 | 30.303248 | 78.393062 | 30.570016 | 61.168537 |
| Comp time | 26.266741 | 89.514275 | 18.113911 | 61.868680 | 15.172678 | 75.852079 |
| 3 | Comm time | 21.454271 | 84.306025 | 26.883242 | 66.318063 | 26.997465 | 44.599300 |
| Comp time | 30.268887 | 96.750693 | 22.730531 | 77.779222 | 19.668534 | 97.079598 |
| 4 | Comm time | 17.583461 | 56.590899 | 21.647295 | 55.586626 | 24.451651 | 62.652676 |
| Comp time | 35.948178 | 126.443870 | 29.806568 | 92.655340 | 23.132160 | 79.088234 |
| 5 | Comm time | 15.946033 | 47.610591 | 19.981328 | 47.045172 | 22.606396 | 53.896293 |
| Comp time | 38.721292 | 143.549835 | 32.386376 | 101.427900 | 25.926298 | 88.140114 |
| 6 | Comm time | 15.024184 | 66.171662 | 8.929359 | 47.225279 | 14.028019 | 50.843508 |
| Comp time | 39.668731 | 125.682471 | 46.499145 | 106.968397 | 38.516393 | 94.009908 |
| 7 | Comm time | 15.380558 | 50.892414 | 17.340718 | 43.223844 | 17.292582 | 48.203713 |
| Comp time | 39.329221 | 141.018594 | 38.110261 | 111.028759 | 35.273926 | 96.708664 |
| Total | Real time | 1m0.112s | 3m17.247s | 0m58.270s | 2m38.940s | 0m56.204s | 2m30.506s |

The table above shows the raw data of the performance under 3 configurations. Discussions are as below,

1. Raw performance – for the raw performance, it seems 1,2,4 nodes performs almost the same.
2. Load balance – Load balance is not good. Proc 0 always do the least computation work, while proc 7 always do the heaviest one. The computation time among different processes vary a lot.
3. Scalability – This algorithm is not scalable because the parallel efficiency decreases when p increases.

Suggestions,

1. Using no-blocking send/recv function– For most processes, a lot of time is used on communication parts. One reason is we are using blocking send/recv functions, which is safe but not efficient. Switching to non-blocking calls will improve the performance.
2. Allocate flexible number of rows – allocate more rows to one process when the number of elements in these rows is relatively small comparing to other rows. Therefore each process can compute approximately the same number of elements and achieve a load balance.

Conclusion from the set of results with p = 8:

Performance can’t be improved by allocating more nodes for the task. We should focus on optimizing algorithm.