Отчет о проделанной работе
по первому заданию
студента 418 группы
Гордеева Михаила Михайловича

Москва 2014

1 Введение

Требовалось написать программу, которая по матрице смежности ориентированного графа находит матрицу состоящую из длин кратчайших путей. Была написана соответствующая программа `main.cpp` и ее параллельный вариант, с использованием `OpenMp` `main_p.cpp`. Этим программам, в качестве аргументов командной строки передаются входной файл(откуда берется число вершин графа и его матрица смежности) и выходной файл(куда записывается полученная матрица кратчайших путей). В стандартный поток вывода программы пишут время их работы.

Так же были написаны две программы непосредственно для замера скорости работы алгоритма. Они называются `test.cpp` и `test_p.cpp` последовательная и параллельная версии соответственно. Для их запуска и автоматического построения тфблиц и графиков была написана программа `run.pl`, подробнее описанная в следующих разделах.

Была написана программа `fw_mpi.c`, которая реализует алгоритм с использованием `MPI`, для запуска этой программы нужно запустить `mpi_fw_run.pl`, передав этомо скрипту входной и выходной файлы(аргументами командной строки). Для его работы нужны `mpicc` и `mpirun`.

2 Описание реализованных алгоритмов

В программах реализован алгоритм Флойда — Уоршелла поиска всех кратчайших путей в графе. Код, реализующий этот алгоритм:

```
1   for (uint k = 0; k < n; ++k) {
2     for (uint i = 0; i < n; ++i) {
3       for (uint j = 0; j < n; ++j) {
4         if (graph[i][k] >= 0 && graph[k][j] >= 0) {
5           if (graph[i][j] == -1 || graph[i][k] + graph[k][j] < graph[i][j])
6             graph[i][j] = graph[i][k] + graph[k][j];
7         }
8       }
9     }
10  }
```

Распараллеливание ведется только по внешнему циклу `for`, исползуя следующую прагму `#pragma omp parallel for schedule(dynamic, 10)`

3 Результаты тестировани и замеры производительности

Программы тестировались на корректность решения сравнением праллельной и последовательной версий программ. Примеры, на которых проводилось тестирование содержатся в папке `tests`, вместе со скриптом `test.pl`, производившим тестирование.

Тестирование времени работы программ `test` и `test_p` производилось с использованием скрипта `run.pl`. Для построения таблиц этому скриту необходим программа `pdflatex` или другая программа, делающая pdf файл по LaTeXдокументу. Графики строятся с использованием `Rscript`. По умолчанию настроики запуска считываются из файла `runrc`, изначально содержащего все настройки с их описанием, но можно передать программе любой конфигурационный файл агрументом командной строки. Примеры таблиц, граффиков и конфигурационных файлов, с помощью которых они были получены, можно найти в каталоге `data`.

По получнным результатам можно сделать вывод, что при увиличении числа нитей(в пределах числа физических ядер) время работы программы снижаетя пропорционально числу нитей, если же увеличивать число нитей дальше(до числа виртуальных ядер), то возможно как небольшое увеличение производительности, так и уменьшение, что наглядно видно на первом примере(файлы `myplot1.pdf` и `table1.pdf` в папке `data`).