

Отчет
по лабораторной работе
«Инструменты параллельного
программирования»

Выполнил:

Студент группы 0836-2

Смертин Д. С.

Проверил:

Ассистент кафедры МОСТ

Козинов Е. А.

Оглавление

Постановка задачи.....	3
Метод решения.....	4
Схема распараллеливания.....	5
Описание программной реализации.....	6
Подтверждение корректности	8
Результаты экспериментов по оценке масштабируемости.....	9

Постановка задачи

В рамках данной лабораторной работы необходимо достичь следующих целей:

1. Реализовать алгоритм глобального поиска (Стронгина) для одномерных задач оптимизации в трех версиях (solver):
 - a. Последовательная версия;
 - b. Параллельная версия с использованием средств OpenMP;
 - c. Параллельная версия с использованием средств TBB;
2. Реализовать генератор тестов;
3. Реализовать программу, проверяющую корректность работы алгоритма;
4. Провести эксперимент по оценке масштабируемости.

Метод решения

Алгоритм глобального поиска (АГП) позволяет находить абсолютный минимум функции на отрезке и основан на вероятностном подходе.

На основе набора известных значений функции в точках отрезка ищется интервал между соседними точками, на котором абсолютный минимум наиболее вероятен (характеристика интервала между этими точками будет максимальной). На этом интервале берется точка, соответствующая математическому ожиданию положения минимума, вычисляется значение функции в ней. Точка добавляется в список известных значений, и происходит переход к следующей итерации. Алгоритм останавливается, когда расстояние между точками отрезка последовательных итераций становится меньше заданного критерия.

Единственным требованием, которому должна удовлетворять целевая функция $g(x)$, — это выполнение обобщенного условия Липшица на всем интервале поиска:

$$|g(x_1) - g(x_2)| \leq K \rho(x_1, x_2),$$

С полным описанием алгоритма можно ознакомиться в [1].

Схема распараллеливания

Данный алгоритм распараллеливается по характеристикам: на каждом шаге цикла работы метода вычисляется столько максимальных характеристик, сколько доступно потоков для вычисления. Благодаря этому на каждой итерации данного цикла можем вычислить столько новых точек, сколько вычислили максимальных характеристик.

Таким образом, можно было бы получить ускорение работы пропорциональное количеству потоков. Но, как покажет эксперимент по оценке масштабируемости, такое ускорение не достигается, так как создается приоритетная очередь для хранения максимальных характеристик. В связи с этим и не получается желаемое ускорение.

Описание программных реализаций

Последовательная версия

Последовательная версия состоит из пяти основных частей:

1. Solver – включает в себя реализацию алгоритма;
2. Testsgenerator – генерирует тесты, проверяющие корректность работы алгоритма;
3. Answersmaker – программа, генерирующая правильные ответы на тесты;
4. Taskmaker – программа, создающая задачи.
5. Checker – программа, которая сверяет результат полученный в результате работы алгоритма и правильного ответа, который генерирует answersmaker;

Рассмотрим эти части подробнее и начнем с самой важной части программного комплекса – Solver. Она включает в себя программу, которой на вход подается путь к задаче, которую сгенерировал Taskmaker, и статическую библиотеку Tofunction, которая позволяет строчный вид функции переводить в исполняемый. По завершению работы Solver создает файл с ответом на поставленную задачу.

Testsgenerator по сути своей является скриптом, который случайным образом выбирает задачу и передает её в Solver.

Checker – сценарий, который последовательно запускает Answersmaker, генерирующий правильные ответы на задачи, Testsgenerator, и саму программу Checker, на вход которой подается решаемая задача, правильный ответ на нее и ответ полученный в результате работы алгоритма. Далее Checker выводит на экран данные по задаче, её решений и результат их сравнения.

```
deathboydmi@deathboydmi-K55A: ~/Documents/git/parprog-2018-1/groups/1506-2
deathboydmi@deathboydmi-K55A:~/Documents/git/parprog-2018-1/groups/1506-2
/smertin_ds/1-test_version/checker$ python3 script_checker.py
answersmaker is doing...
answersmaker is done.
testsgenerator is doing...
testsmaker is doing...
testsmaker is done.
gsa is doing with ../tasks/task_06.task
gsa is done.
testsgenerator is done.
checker is doing...

TASK:
  function: (x*sin((x^(0.5)))+cos(x)*x*10)/100
  left border: 0
  right border 1000
  number of iterations: 10000000
  accuracy: 0.0001

RESULT OF THE METHOD:
  (939.336, -100.455)

CORRECT ANSWER:
  (939.336, -100.455)

  OK

checker is done.
deathboydmi@deathboydmi-K55A:~/Documents/git/parprog-2018-1/groups/1506-2
/smertin_ds/1-test_version/checker$
```

Параллельные версии

Структура параллельных версий схожа с последовательной версией, за исключением того, что сам проект был немного реструктурирован: был добавлен stake-файл, собирающий все части комплекса вместе, и скрипт *main-script.py*, автоматизирующий работу всех частей программного комплекса. Он запускает сборку проекта и ставит в очередь работу тестов (одного случайного, если параметров запуска скрипта нет, и всех, если входным параметром будет "0").

```
deathboydmi@deathboydmi-K55A: ~/Documents/git/parprog-2018-1/groups/1506-2/
smertin_ds/2-openmp_version$ ls
bin      CMakeLists.txt  log-files      pyscripts
build    lib             main-script.py sources
deathboydmi@deathboydmi-K55A:~/Documents/git/parprog-2018-1/groups/1506-2/
smertin_ds/2-openmp_version$ python3 main-script.py
building is doing...
-- Configuring done
-- Generating done
-- Build files have been written to: /home/deathboydmi/Documents/git/parprog-2018-1/groups/1506-2/smertin_ds/2-openmp_version/build
[ 20%] Built target tofunction
[ 40%] Built target gsa
[ 60%] Built target checker
[ 80%] Built target tasksmaker
[100%] Built target answersmaker
building is done.
answersmaker is doing...
answersmaker is done.
testsgenerator is doing...
testsmaker is doing...
testsmaker is done.
gsa is doing with ../bin/tasks/task_02
gsa is done.
testsgenerator is done.
checker is doing...

TASK:
    function: (x/4)*cos(x)
    left border: 0
    right border 30
    number of iterations: 1000000
    accuracy: 0.0001

RESULT OF THE METHOD:
    (28.3096, -7.073)

CORRECT ANSWER:
    (28.3096, -7.073)

    OK

checker is done.
```

Подтверждение корректности

Для подтверждения корректности была создана программа Answersmaker. Для каждой из 10 задач, были построены графики исходных функций, визуальным образом были определены области глобального минимума, и на этих областях с шагом на порядок меньше чем необходимая точность методом полного перебора вычисляется необходимая точка. А если была возможность, то глобальный минимум определялся аналитическим способом.

Если разность значений функций в точках, найденных АГП и алгоритмом полного перебора, меньше заданной точности, то работа АГП признается корректной, а тест пройденным.

Результаты экспериментов по оценке масштабируемости

	0	1	2	3	4	5	6	7	8	9	average
serial_time	2,35362	8,69485	0,0938587	0,204975	0,392872	0,122132	1,98525	1,3355	29,8236	0,0060009	
serial_iters	3004	5589	683	964	1350	802	3220	2325	9998	145	
omp_time (4 threads)	1,19539	4,92152	0,0330061	0,210318	0,144977	0,101642	1,30956	0,674288	9,29869	1,09E-05	
omp_iters (4 threads)	884	1806	177	369	358	300	1043	667	2498	1	
serial_time / omp_time (4 threads)	1,968913911	1,766700125	2,843677381	0,974595612	2,709891914	1,201589894	1,515967195	1,980607693	3,207290485		2,018803802
serial_iters / omp_iters (4 threads)	3	3	4	3	4	3	3	3	4		3,331532004
tbb_time (4 threads)	0,68044	2,92733	0,0472582	0,0867358	0,156732	0,127844	0,833746	0,570152	4,67213	1,30E-05	
tbb_iters (4 threads)	880	1905	184	249	368	303	1080	795	2498	1	
serial_time / tbb_time (4 threads)	3,458967727	2,970232259	1,986082835	2,363211039	2,506648291	0,955320546	2,381120869	2,342357827	6,38329841		2,816359979
serial_iters / tbb_iters (4 threads)	3	3	4	4	4	3	3	3	4		3,350521305
omp_time (2 threads)	5,39917	19,0607	0,140137	0,581142	0,758529	0,33804	3,61311	2,89249	15,2917	9,95E-06	
omp_iters (2 threads)	3050	5540	565	1068	1254	862	2920	2284	4998	1	
serial_time / omp_time (2 threads)	0,4359225585	0,4561663528	0,669763873	0,352710697	0,517939327	0,361294521	0,549457392	0,461712918	1,950312915		0,639475617
serial_iters / omp_iters (2 threads)	1	1	1	1	1	1	1	1	2		1,137030487
tbb_time (2 threads)	2,6953	9,07031	0,124727	0,333936	0,546229	0,225809	2,42413	1,23855	7,2981	1,13E-05	
tbb_iters (2 threads)	3034	5593	557	924	1285	774	3142	1961	4998	1	
serial_time / tbb_time (2 threads)	0,8732311802	0,9586056044	0,752513088	0,613815222	0,719244126	0,540864181	0,818953604	1,078277017	4,086488264		1,160221365
serial_iters / tbb_iters (2 threads)	1	1	1	1	1	1	1	1	2		1,17294476
omp_time (8 threads)	0,00616606	0,00437834	0,00430923	0,0173333	0,0103631	0,134229	0,0120468	0,0350927	0,195989	1,35E-05	
omp_iters (8 threads)	36	30	26	64	49	223	54	102	260	1	
serial_time / omp_time (8 threads)	381,7056597	1985,87821	21,7808518	11,82550351	37,9106638	0,909877895	164,7948003	38,0563479	152,1697646		310,5590755
serial_iters / omp_iters (8 threads)	83	186	26	15	28	4	60	23	38		51,45568907
tbb_time (8 threads)	0,41331	0,849936	0,0255233	0,0596325	0,0541535	0,0328277	0,282845	0,128987	2,10039	1,04E-05	
tbb_iters (8 threads)	521	765	93	156	148	103	437	254	1248	1	
serial_time / tbb_time (8 threads)	5,694563403	10,23000555	3,677373224	3,437303484	7,254785005	3,720394667	7,018861921	10,35375658	14,19907731		7,287346795
serial_iters / tbb_iters (8 threads)	6	7	7	6	9	8	7	9	8		7,559611354

<https://docs.google.com/spreadsheets/d/1vGxoP7MKxvILlyQlb5Yp4MXZ3hEQoV3TVZR-Mx0ib-yo/edit?usp=sharing>

Литература

[1] Стронгин Р. Г. Численные методы в многоэкстремальных задачах;

Приложение

https://github.com/deathboydmi/parprog-2018-1/tree/master/groups/1506-2/smertin_ds