Задание практикума: построение расписаний с помощью параллельного алгоритма имиитации отжига

Сальников А.Н.

Содержание

1 Общее описание

Требуется написать программу, реализующую построение расписания для выполнения работ с зависимостями по данным без прерывания. Программа должна быть написана на языке программирования C++ и должна быть многопоточной. Многопоточность допускается реализовывать либо средствами библиотеки pthread, либо средствами организации потоков в стандартной библиотеке C++11. Программа должна создавать файл с расписанием и, через определённое в параметрах число итераций, выдавать характеристики полученного промежуточного расписания.

Необходимо провести исследование производительности реализованного алгоритма, построить графики:

- времени работы в зависимости от числа потоков и количества работ в расписании (на графике представить относительную погрешность для времени работы при 20 запусках. Число работ варьировать в некоторой линейной шкале, например от 10 до 1000 с шагом 100. Экспериментально определите, при каких количествах работ и доступных им процессорах последовательный алгоритм имитации отжига работает больше 1 минуты.

- ускорения в зависимости от числа потоков и количества работ.

- Времени последовательной реализации от числа работ и доступных им процессоров.

Необходимо написать генератор ориентированных взвешенных лесов (ориентированных графов без циклов с несколькими компонентами связанности с весами заданными на вершинах). Собственно вершины в графе – работы, рёбра в графе – зависимости между работами. Параметрами генератора должны быть:

- число вершин в графе.
- диапазон весов приписываемых вершине и дисперсия для весов вершины. Вес вершины должен быть натуральным числом.
- среднее число связанных компонент и дисперсия числа вершин в компоненте связанности.
- средняя степень вершины в графе и дисперсия для степени вершины.

Необходимо создать некоторое количество небольших графов, на которых будет проверяться корректность работы алгоритма.

Необходимо написать конвертер графов в .dot формат, чтобы потом его можно было отрисовать средствами graphviz.

Необходимо написать программу и функцию, которым на вход подаётся множество работ (граф) и расписание, а на выходе мы получаем вычисленные характеристики расписания: длительность, степень разбалансированности, объём простоев[1].

2 Задача построения расписания

Дана некоторая многопроцессорная система P состоящая из M одинаковых процессоров $P = \{p_1, ..., p_M\}$. Дано множество работ $J = \{j_1, ..., j_N\}$, здесь каждая работа обладает длительностью и множеством k зависимостей от других работ. Зависимости не дают запустить работу на процессоре, пока не закончатся все те работы, от которых данная работа непосредственно зависит. Здесь $j_i :< \tau_i, \{j_{dep_1}, ..., j_{dep_k}\} >$.

Расписание в данном случае будет списком работ для каждого процессора. Список задаёт последовательность выполнения работ на процессоре. Время окончания всех работ для конкретного процессора определяется временами исполнения каждой работы и вынужденым простоем для обеспечения зависимости по данным.

Требуется так разместить работы на многопроцессорной системе, чтобы выполнялись следующие критерии[2]:

[1] Описание характеристик указано в тексте ниже

[2] Конкретный критерий является вариантом задания, студенту нужно реализовывать только один из указанных критериев

1. Минимизация времени исполнения всех работ. Определяется по времени окончания последней работы. (длительность расписания).

2. Минимизация разбалансированности расписания.
   Минимизация $T_{max} - T_{min}$, где T_{max} – последний момент времени освобождения процессора от работ в расписании, T_{min} – первый момент времени освобождения процессора от работ.

3. Минимизация простоев. Минимизируем общую длительность дырок в расписании. Если есть свободные процессоры, то это будет дырка продолжительностью в длину расписания помноженную на число свободных процессоров.

3 Параллельный алгоритм имитации отжига

Требуется реализовать алгоритм, где каждый поток обрабатывает своё начальное приближёние расписания отличающееся от расписаний на других потоках, а затем происходит синхронизация решений между выполняющимися потоками. Предполагается, на определённых шагах алгоритма будет произведён обмен решениями, выбор среди них лучшего и продолжение независимого варьирования пространства поиска каждым потоком. См. лекцию В.Костенко: https://asvk.cs.msu.su/sites/all/themes/professional_theme/files/Kostenko/K_2021_os/%D0%9B4_5_%D0%98%D0%9E.ppt.

Для алгоритма имитации отжига использовать понижение температуры по закону Больцмана, но можно предложить и свой вариант, если на задаче построения расписания эксперементально будет видно, что ваш закон работает лучше.

4 Форматы входных и выходных файлов

Файл со списком работ задаётся в следующем формате. Это текстовый файл. На первой строчке в нём указано число работ. Далее на каждой строчке длительность работы, а потом, через пробельные символы, до конца строчки номера работ, от которых зависит данная работа.

Файл с расписанием задаётся в следующем формате. Это тоже текстовый файл, первой строчкой идёт число процессоров, далее на каждой строчке: сперва идёт номер процессора, а затем двоеточие и через пробельные символы номера работ в порядке их назначения на процессор.

В файлах могут встречаться коментарии кодируемые символом решётка. От решётки до конца строки – всё коментарий, а так же могут встречаться пустые строки и строки, целиком состоящие из пробельных символов.

Пример графа работ:

```
#
# File with jobs
#
5

# Job number 0 has duration 100 and
# no dependencies.
100             # job 0
50        0 2   # job 1
12        0     # job 2

8888      4     # job 3
70000           # job 4
```

Пример расписания:

```
#
# file with schedule.
# We have three processors and five jobs
#
3

# proc number 0
0: 0 2 1

# There are no jobs for processor 1

# proc number 2
2: 3 4
```