7.12.2013

Существует алгоритм Холта-Деккера – который позволяет организовать атомарность в задаче обедающих философов без семафоров дейкстры.

Семафоры Дейкстры лучше, потому что

Разрешимые операции над семафорами: считать значение, записать, поднять и опустить (в общем случае это увеличение и уменьшение на некоторое значение), а так же создать и уничтожить.

С ходу нужно уметь решать задачу – «поставщик и потребитель» (другое название «читатели и писатели»).

Алгоритм Банкира:

|  |  |  |  |
| --- | --- | --- | --- |
| Разные процессы | Max количество ресурсов | Выделенное количество | Сводбодное количество |
| A | 4 | 2 | 2 |
| B | 6 | 3 | 3 |
| C | 8 | 2 (потом добавим сюда 2) | 6 |

|  |  |  |  |
| --- | --- | --- | --- |
| Разные процессы | Max количество ресурсов | Выделенное количество | Сводбодное количество |
| A | 4 | 2 | 2 |
| B | 6 | 3 | 3 |
| C | 8 | 4 | 4 |

Тут есть deadlock, т.к. если кому-то потребуется слишком большое количество ресурсов (всем нужно большое количество ресурсов, чтобы закончить), а кто-то другой, уже немного занял и ресурсов не хватает, чтобы кто-то один закончил, и первому – не хватит, то всё приехали, все будут висеть.

call setseq (seq, 0) – создание семафора

do i = 1, N C#DIR DO PARALLEL(ORDER) - обозначает, что цикл будет выполняться параллельно, ORDER - значит, что распространять операции будем по-единично (т.е. i =1 будет выполнять первое ядро, i = 2 – второе ядро, … по кругу)

 Ai= E\*D(B Ci)

 Ci=Di + Ei; CALL WAIT(SEQ, I-1) -- тут синхронизация по семафору

 Xi=S + Xi-1 + Ci CALL POSTSEQ (SEQ, 1)

 Zi = Xi + Yi

Существует система спецкомментариев OpenMP, которая позволяет распараллеливать

CP DO

STATIC m – приказ распределить блоками по m на каждой из параллелей, т.е. dynamic – даёт последний необработанный, вроде, а static – просто из очереди, в общем не ясно.

DIN m

GUIDED

R

Фисун не одобряет OpenMP, это конечно круто, однако подходит только для простых программ, а сложные и реальные задачи нужно делать вручную.

Другая система спец комментариев (другой язык) – HPF (High Performance Fortran) – она чситается очень успешной и перспективной.

POSIX Pthreads – библиотека для межпроцессорного взаимодействия.

MPI – Message passing Interface – библиотека функций, которая позволяет слать сообщения, между процессорами.

Задача на суперЭВМ программируется следующим способом – пишется программа, которая стартует на всех ядрах, а потом начинает выяснять – «сколько нас», «кто мы», и потом начинает распараллеливаться задача.

Другой подход распараллеливания (кроме спецкомментариев) – встроенные редства параллельного программирования – например Fortran (полагаю, он рассказывает, что можно запустить параллельный поток)

Task program

…

End

Newtask (s, b)

В любой системе программирования есть send и recv – там есть – кому посылаем (от кого принимаем), что шлём, способ передачи – синхронное (тот, кто отослал – ждёт, пока прочтут), асинхронное (тот кто отослал, не ждёт пока прочтут) и no\_wait (тот, что читает – не ждёт, пока пришлют).

MPP, MPI, последовательная, графический процессор – позволяет делать универсальный код.