Численное решение уравнения переноса на больших кластерах. Параллельный алгоритм. Теоретическая оценка ускорения.

Рассмотрим уравнение переноса.

Ut + 2Ux = x + t;

При границчных условиях:

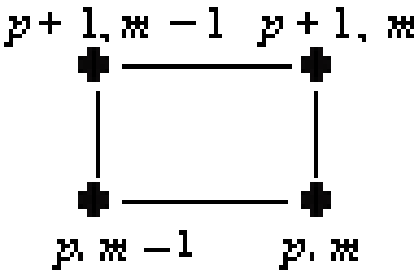
0 < x < 1, 0 < t < 1, U(x, 0) = cos(pi * x), U(0, t) = exp(-t);

1) Проверим граничные условия в точке (0, 0)

Для решения этого уравнения переноса будем использовать схему прямоугольник, так как можем решить за один проход по двумерной табличной сетки. Так же эта схема хорошо подходит для параллелизации алгоритма.

Схема прямоугольник.

Сеточный шаблон:



Разностная схема:

$$\frac{u_{m-1}^{p+1}-u_{m-1}^{p}+u_{m}^{p+1}-u_{m}^{p}}{2\tau}+\frac{u_{m}^{p+1}-u_{m-1}^{p+1}+u_{m}^{p}-u_{m-1}^{p}}{2h}=f_{m-1/2}^{p+1/2},$$

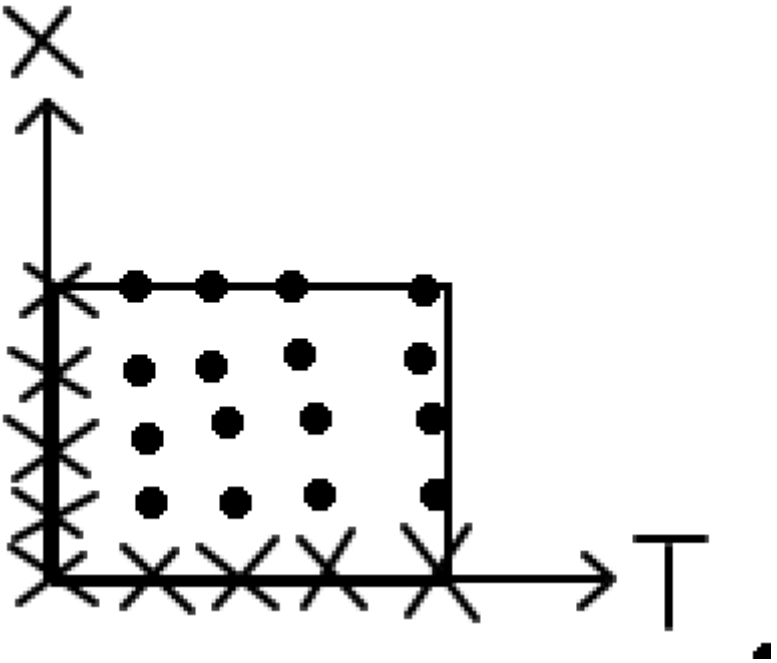
Также схема сходится при любых соотношениях между шагами по осям t, x.

Порядок аппроксимации.

O(t * t + h * h)

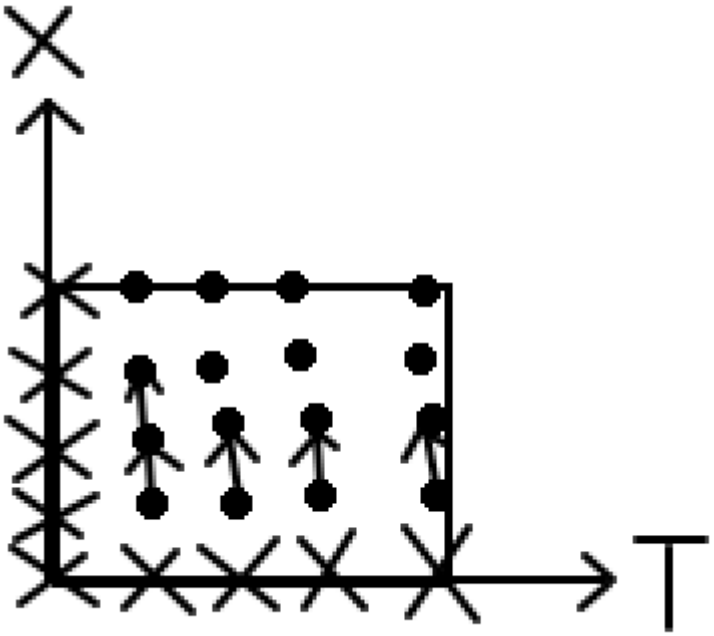
Решение для одного процесса.

Начальные условия.



X – начальные условия. Точками обозначены значения, которые мы должны посчитать. Чтобы подсчитать значение, нам необходимо знать значения 3 точек. Можем вычислить их из начальных условий. Дальше выражаем правый верхний угол прямоугольника через разностную схему. Далее мы можем выразить следующее значение и т.д.

Разрешим нашу задачу для большого числа процессов. Воспользуемся идей пайпланов из микроархитектуры процессоров. Пусть у нас есть N процессов и $2N$ строк матрицы, которую нам нужно вычислить. Пусть 0 процесс считает 1 строчку и $N + 1$, 2 процесс 2 и $N + 2$. и т. д. Как только 1 процесс подсчитал первое значение, он передает это значение 2 процессу. Далее, они параллельно вычисляют 1 и 2 значения. Передача данных от одного процесса к другому осуществляется за $O(1)$, поэтому в теоретических оценках мы не будем ее учитывать. Тогда для квадратной матрицы N строчек можем подсчитать за $3N - 1$ операций, а на одном процессе это бы заняло $N * N$ операций.



Этот алгоритм нельзя математически улучшить, так как мы не можем начать считать значения 4 точки, пока не получим значения 3 соседних точек, образующих «прямоугольник». Но так как передача осуществляется гораздо больше времени, чем идет вычисление в точке, то мы можем архитектурно улучшить наши результаты, уменьшив количество передач.
Результаты: