Потокобезопасная односвязная очередь

список, заполненный случайными элементами, *n* - количество операций (вставки случайных элементов в начало или в конец и удаления из начала или с конца). Операции выполняются параллельно в *p* потоках

*Выход*: список после выполнения *n* операций

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **10^3** | | | **10^4** | | | **10^5** | | | **10^6** | |
|  | **time** | **speed** | **time** | | **speed** | **time** | | **speed** | **time** | | **speed** |
| **1** | 0,000368 | 2718279 | 0,003512 | | 2861248 | 1,699198 | | 58851 | 16,85838 | | 59318 |
| **4** | 0,000453 | 551882 | 0,005026 | | 497403 | 1,485669 | | 16827 | 14,4808 | | 17264 |
| **8** | 0,001006 | 124268 | 0,006302 | | 198354 | 1,422253 | | 8789 | 14,08221 | | 8876 |

Работаем в 1, 4, 8 потоков

**1) пропускная способность**

Пропуснкая способность b – скорость выполнения операций со структурой.

**2) латентность** l – время выполнения отдельной операции.

**3) коэффициент ускорения**

p - число процессоров, n - размер задачи.

T 1 (n) - время, необходимое одному процессору для решения задачи размера n

T p (n) - время для решения p процессорами задачи размера n

χ = T 1 (n) / T p (n)

- ускорение параллельных вычислений, определяет “выгоду” от распараллеливания задачи данного размера n

**4) эффективность распараллеливания**

Эффективность параллельного алгоритма - соотношение лучшего возможного времени параллельного алгоритма к реальному значению времени выполнения параллельного алгоритма.

**E p = (T 1 (n) / p) / T p (n) = χ / p**

▪ E p ≤ 1, поскольку s ≤ p

▪ E p = Const ⇒ линейное ускорение

**5) закон Амдала**

Пусть программа содержит часть **r**, которая может быть распараллелена идеально (ускорение равно **p**) и полностью последовательную часть

s = 1 – r, которая вообще не может быть распараллелена.

Тогда ускорение, которое может быть получено на ВС из p процессоров:

χ = 1 / (s + r / p) = 1 / (s + (1 – s) / p)

**6) масштабирование**

Параллельный алгоритм сильно масштабируется, если χ = Θ(p) (ускорение линейное, не зависит от n).

Параллельный алгоритм слабо масштабируется, если существует

медленно растущая функция f(p), такая что для n = Ω(f(p)) функция E p (n) остаётся постоянной.