

# Параллельное вычисление дискретного преобразования Фурье полинома в простом поле

А.О. Лапаев

Тамбовский государственный университет  
alapaev@gmail.com

## Аннотация

Предлагается алгоритм параллельного вычисления дискретного преобразования Фурье полинома многих переменных в простом поле, и докладываются результаты численных экспериментов на кластере МСЦ.

## Введение

Пусть  $p$  – простое число,  $\mathbb{Z}_p$  – конечное поле,

$$f = \sum_{t_1=0}^{n_1-1} \sum_{t_2=0}^{n_2-1} \dots \sum_{t_d=0}^{n_d-1} f_{t_1 t_2 \dots t_d} x_1^{t_1} x_2^{t_2} \dots x_d^{t_d}$$

– полином из кольца  $\mathbb{Z}_p[x_1, x_2, \dots, x_d]$ . Пусть в  $\mathbb{Z}_p$  существуют  $w_i$  – корни из 1 степени  $n_i$ .

**Определение.** Дискретным преобразованием Фурье (ДПФ) полинома  $f$  называется  $d$ -мерный вектор размерности  $n_1 \times n_2 \times \dots \times n_d$ , элементами которого являются следующие числа:

$$f_{j_1 j_2 \dots j_d} = \sum_{t_1=0}^{n_1-1} w_1^{t_1 j_1} \sum_{t_2=0}^{n_2-1} w_2^{t_2 j_2} \dots \sum_{t_d=0}^{n_d-1} w_d^{t_d j_d} f_{t_1 t_2 \dots t_d}. \quad (1)$$

Формула (1) содержит  $n = n_1 * n_2 * \dots * n_d$  слагаемых. Массив  $f$  также содержит  $n$  элементов. Таким образом, оценка сложности алгоритма вычислений по этой формуле –  $O(dn^2)$  при предварительно вычисленных степенях  $w_1^{t_1 j_1}, \dots, w_d^{t_d j_d}$ .

Далее мы будем рассматривать случай, когда каждое из  $n_i = 2^{N_i}$ . При фиксированных значениях  $j_1 j_2 \dots j_{d-1}$  сумма

$$\sum_{t_d=0}^{n_d-1} f_{t_1 t_2 \dots t_d} w_d^{t_d j_d}$$

является одномерным преобразованием Фурье и может быть вычислена при помощи алгоритма быстрого преобразования Фурье. Таким образом, на первом шаге требуется вычислить  $n_1 * n_2 * \dots * n_{d-1}$  одномерных преобразований Фурье.

Аналогично: на шаге  $k$  при фиксированных значениях  $j_1 \dots j_{d-k-1} j_{d-k+1} \dots j_d$  вычисляется  $n_1 \dots n_{d-k-1} n_{d-k+1} \dots n_d$  одномерных преобразований Фурье.

Сложность такого способа вычислений составляет

$$O(n d * \log_2 n),$$

где  $n = n_1 * n_2 * \dots * n_d$ .

Схема распараллеливания будет состоять из  $d$  последовательных шагов, каждый из которых выполняется параллельно. Схему вычисления каждого шага можно представить в виде бинарного дерева, у которого данные распределяются от корневой вершины к листовым, а результат вычислений собирается снова в корневой вершине. На каждом шаге параллельно вычисляются одномерные ДПФ, при этом все вычисления происходят в листовых вершинах. При переходе к следующему шагу меняется порядок переменных, а после шага  $d$  будет получен искомым  $d$ -мерный вектор дискретного преобразования Фурье. Приведем алгоритм.

### Алгоритм

Будем проводить вычисления в следующем порядке: на шаге  $k$  параллельно вычисляются  $n_1 \dots n_{d-k-1} n_{d-k+1} \dots n_d$  дискретных преобразований Фурье при фиксированных значениях индексов  $j_1 \dots j_{d-k-1} j_{d-k+1} \dots j_d$ .

Пусть  $F_k$  – результат вычислений  $k$ -го шага, и пусть  $F_0 = f$ .

Приведем схему вычислений на шаге  $k$ . В корневой вершине  $d$ -мерный массив  $F_{k-1}$ , полученный на предыдущем шаге делится на две равные части по индексу  $j_{d-k+1}$ . И каждая часть отправляется в соответствующую дочернюю вершину. В дочерних вершинах повторяется разделение этого массива по тому же индексу на две половины. Такой процесс выполняется рекурсивно до тех пор, пока имеются свободные процессоры, либо пока возможно деление по данному индексу, т.е. число процессоров, вовлеченных в вычисления меньше  $\frac{n}{n_k}$ . На листовом уровне вычисляются одномерные ДПФ и результат возвращается в обратном порядке. После чего, при переходе к следующему шагу вычислений, порядок индексов массива  $F_k$  меняется с  $[j_{d-k+2} \dots j_1 \dots j_{d-k+1}]$  на  $[j_{d-k+1} \dots j_1 \dots j_{d-k}]$ . Отметим, что на шаге  $k$  требуется не больше, чем  $\frac{n}{n_k}$ .

Пусть имеется  $m$  компьютерных модулей (КМ) с номерами  $1 \dots m$ . Пусть  $k = 1$ .

1) КМ с номером 1 получает  $d$ -мерный массив  $F_{k-1}$  и список свободных КМ.

Вычисляются степени  $w_{d-k+1}^{i_{d-k+1} j_{d-k+1}}$ . Массив коэффициентов и список свободных процессоров делятся пополам. Одна половина массива коэффициентов и списка свободных процессоров вместе с массивом степеней  $w_{d-k+1}^{i_{d-k+1} j_{d-k+1}}$  отсылается КМ с номером  $(m/2 + 1)$ , а вторые половины остаются в данном КМ.

2) Каждый КМ, получивший свою часть задачи, продолжает такое деление дальше. Процесс продолжается до достижения листового уровня, либо исчерпания всех свободных процессоров.

3) Выполняется параллельное вычисление одномерных преобразований Фурье.

4) Каждый КМ пересылает результат обратно по дереву тому процессору, от которого получил данные.

5) КМ с номером 1 собирает результат и меняет порядок индексации результирующего массива с  $[j_{d-k+2} \dots j_1 \dots j_{d-k+1}]$  на  $[j_{d-k+1} \dots j_1 \dots j_{d-k}]$ .

6) Увеличивается номер шага  $k$ . Если  $k=d+1$ , то конец всех вычислений, иначе переходим на п.1.

По этому алгоритму разработан программный комплекс на кластере МСЦ РАН. Результаты проведенных экспериментов будут представлены в докладе.

### Литература

1. Ноден П., Китте К. Алгебраическая алгоритмика. М.: Мир. 1999
2. Кормен, Томмас Х., Лейзерсон. Алгоритмы: построение и анализ, 2-е издание. : Пер. с англ. – М.: Издательский дом «Вильямс», 2005. – 1296 с.: ил. – Парал. Тит. англ.