

## ПРИМЕНЕНИЕ ТЕХНОЛОГИЙ ПАРАЛЛЕЛЬНЫХ И РАСПРЕДЕЛЁННЫХ ВЫЧИСЛЕНИЙ ДЛЯ ПОИСКА ОБРАЗУЮЩИХ ПОЛИНОМОВ

Мышко Е.А., Мальчуков А.Н.

Научный руководитель: Мальчуков А.Н., к.т.н., доцент

Томский политехнический университет, 634050, Россия, г. Томск, пр. Ленина, 30

E-mail: evgenrus70@mail.ru

В данной работе рассматривается применение технологий параллельных и распределённых вычислений OpenMP и MPI для поиска образующих полиномов, что является достаточно трудоёмкой задачей. В статье рассмотрен алгоритм поиска образующих полиномов и проанализирован с точки зрения возможности применения технологий параллельных и распределённых вычислений. Описана программная реализация алгоритма поиска образующих полиномов с применением технологии параллельных и распределённых вычислений OpenMP и MPI. Для сравнения быстродействия программных реализаций с применением и без применения приведённых технологий поставлен компьютерный эксперимент.

### Введение

При построении блочных помехоустойчивых кодов (Боуза-Чоудхори-Хоквингема (БЧХ-кодов), Рида-Соломона) используются полиномы Жегалкина. Полиномы, участвующие в построении кода, называются образующие полиномы. Отдельные коды, такие как коды БЧХ-коды [1], имеют чёткие рекомендации выбора образующего полинома, но такие коды ограничены фиксированной длиной кодового слова. Остальные длины кодовых слов получаются путём укорачивания более длинных кодовых слов за счёт уменьшения разрядности информационного блока при сохранении количества контрольных разрядов. Укорачивание БЧХ-кодов вводит лишнюю избыточность, которая не несёт в себе дополнительных корректирующих возможностей, а лишь уменьшает эффективность кода. Для поиска образующих полиномов, более эффективных, чем укороченные БЧХ-коды, необходимы большие вычислительные затраты. Это связано с тем, что мы знаем только необходимые начальные условия для поиска, такие как минимальные вес образующего полинома, минимальная длина кодового слова, а значит и минимальная длина образующего полинома. Для того, чтобы найти подходящий полином необходимо перебрать все полиномы, подходящие под выше озвученные необходимые условия и если среди них не найдётся полином, который пройдет проверки на достаточные условия для образующего полинома, то поиск продолжится среди полиномов, длина которых на единицу больше, чем предыдущих [2]. Среди достаточных условий образующего

полинома для заданных значений  $m$  и  $t$  – являются два основных: 1) строящийся на основе этого образующего полинома код должен иметь расстояние Хэмминга не меньше, чем  $d=2t+1$ ; 2) остатки от деления всех комбинаций ошибок на образующий полином в рамках заданной  $t$  должны быть уникальными.

### Применений технологий MPI + OpenMP для задачи поиска образующего полинома

Для решения поставленной задачи были применены технологии параллельных и распределённых вычислений, такие как MPI и OpenMP. При использовании технологии MPI [3] осуществляется межпроцессорный обмен полиномами в распределённой системе (суперкомпьютерный кластер). Главный процессор с помощью команды MPI\_Send пересылает наборы полиномов в определённом интервале другим процессорам, которые принимают данные с помощью команды MPI\_Recv и осуществляют поиск образующего полинома среди принятого набора. На рис. 1 приведена схема применения технологий MPI + OpenMP для задачи поиска образующего полинома.

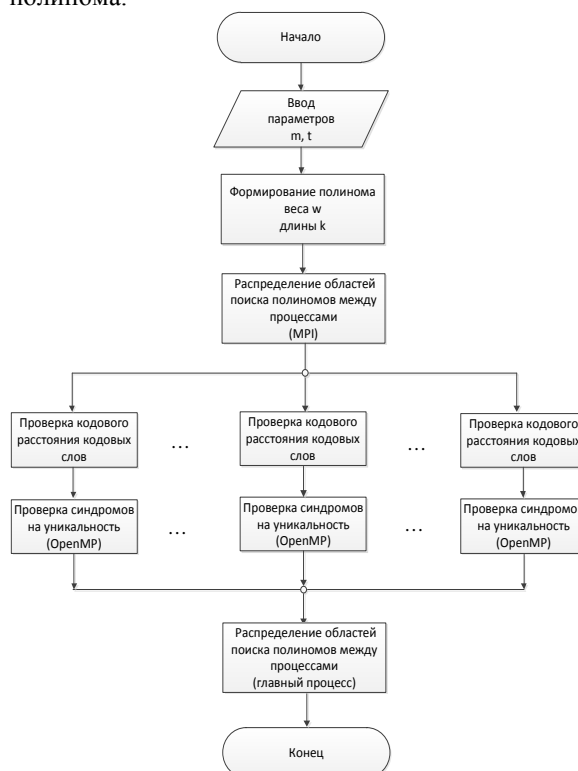


Рис.1. Схема применения MPI + OpenMP для поиска образующего полинома

Таким образом, разделяя диапазоны поиска полиномов между вычислительными узлами, сокращается общее время поиска. Также, для достижения наилучшего результата по быстродействию программы поиска образующего полинома для каждого процессора применяется технология OpenMP [4], позволяющая распараллелить самые трудоёмкие вычисления, необходимые для проверки полиномов на выполнение достаточных условий. В данном случае, наиболее трудоёмким является этап проверки синдромов ошибок на уникальность, где и была применена технология OpenMP.

#### Результаты применения технологий параллельных и распределённых вычислений

Для получения статистики по длительности поиска полиномов было произведено по 10 запусков программы поиска с входными параметрами  $m$  (8-24) и  $t = 4$  на 10 процессорах Intel XEON 5150 2.66Ghz на суперкомпьютерном кластере «СКИФ-политех» [5]. Средние значения полученных результатов представлены в таблице.

**Таблица.** Средние значения времени поиска образующего полинома

$m, t$	Последовательный, с	MPI + OpenMP, с	Ускорение, %
8,4	1,656	0,278	495
9,4	3,741	0,405	823
10,4	5,661	0,424	1234
11,4	10,832	0,518	1990
12,4	35,808	1,307	2634
13,4	52,878	1,42	3622
14,4	132,757	5,968	2124
15,4	166,861	7,669	2076
16,4	301,419	18,498	1529
17,4	440,362	37,722	1067
18,4	493,577	47,791	933
19,4	569,603	64,781	780
20,4	659,76	97,582	576
21,4	739,98	166,606	344
22,4	12172,68	2176,21	459
23,4	37390,24	7367,014	407
24,4	38618,16	9907,568	290

Для наглядного представления результатов на рис.2 приведён график столбца «Ускорение» из таблицы.



**Рис.2.** Ускорение алгоритма с применением MPI + OpenMP относительно последовательного

#### Заключение

Алгоритм поиска образующих полиномов программно реализован с применением технологий параллельных вычислений MPI и OpenMP. Результаты поставленного компьютерного эксперимента по исследованию быстродействия последовательной и распараллеленной реализаций алгоритма поиска полиномов показали, что применение технологии MPI в совокупности с OpenMP значительно ускоряет поиск образующих полиномов. Так, на 10 процессорах при определенных входных параметрах ( $m=13, t=4$ ) ускорение достигает 3622 %, а в среднем ускорение для исследуемых входных параметров составляет 1258 %, т.е. применение MPI + OpenMP позволяет ускорить алгоритм поиска в среднем более, чем в 10 раз.

#### Литература

1. Боуз Р.К., Рой-Чоудхури Д.К. Об одном классе двоичных групповых кодов с исправлением ошибок. – В кн.: Кибернетика. М., 1964. – С.112-118.
2. Мальчуков А.Н. Алгоритм поиска образующих полиномов для системы проектирования кодеров помехоустойчивых кодов //Современные техника и технологии: Труды XIV Международной научно-практической конференции студентов, аспирантов и молодых ученых - Томск, 24-28 марта 2008. - Томск: ТПУ, 2008. - с. 340-341 (12738229)
3. Г.И. Шпаковский, Н.В. Серикова Программирование для многопроцессорных систем в стандарте MPI // Минск. БГУ, 2002. – 323 С.
4. Аксёнов С. В. , Мальчуков А. Н. , Мыцко Е. А. Применение технологии параллельных вычислений OpenMP для поиска образующих полиномов [Электронный ресурс] // Интернет журнал Науковедение. - 2013 - №. 6 (19). - С. 1-12. - Режим доступа: <http://naukovedenie.ru/PDF/86TVN613.pdf>
5. СКИФ -1. Center of supercomputing. 2013. URL: <http://cluster.tpu.ru/?q=node/26> (дата обращения: 15.01.2013)