Сравнение алгоритмов обращения элементов $GF(2^N)$

Докладчик: Щербаков А. В. - студент кафедры системного программирования СПбГУ

Научный руководитель: Платонов С. М. – руководитель исследовательской лаборатории RAIDIX

Актуальность задачи

В современных СХД на RAID для восстановления данных и нужно деление в $GF(2^N)$.
Можно хранить, но в общем случае придётся хранить

$$2 * 2^N \text{ бит.}$$

Задача

- Провести реализацию и сравнение алгоритмов обращения элементов $GF(2^N)$
- Провести замеры производительности каждого реализованного алгоритма

Обращение элементов $GF(2^N)$

$$GF(2^N) = Z_2[x]_{/f},$$

f -неприводимый в $Z_2[x]$ мн-лен вида:

$$f(x) = x^N + p(x), \deg p(x) < N.$$

Тогда, $\forall a \in GF(2^N)\backslash\{0\}$

$\exists a^{-1} \in GF(2^N)\backslash\{0\}$:

$$a * a^{-1} \equiv 1 \ (mod\ f).$$

1. Обращение элементов $GF(2^N)$ возведением в степень

$$a^{-1} \equiv a^{2^N - 2} (mod\ f)$$

2. Обращение элементов в $GF(2^N)$ подсчётом континуанты

Континуанту можно определить рекуррентно:

$$K_{-1} = 0, K_0 = 1,$$

$$K_{m+1}(x_1, \dots, x_m, x_{m+1}) = x_{m+1} * K_m(x_1, \dots, x_m) + K_{m-1}(x_1, \dots, x_{m-1})$$

Идея алгоритма

- $\forall a \in GF(2^N) \backslash \{0\}$ НОД(a, f) = 1
- $1 = f * q + a * w \ (mod\ f)$
- $a * w \equiv 1 (mod\ f) \Rightarrow w = a^{-1}$
- Если во время счёта НОД мы получили $q_1, \dots, q_s$ - частные при делении с остатком, то $w = K_s(q_1, \dots, q_s)$

2. Обращение элементов в $GF(2^N)$ с использованием ганкелевых матриц

Данный метод, основанный на результате Кронекера по представлению результанта полиномов посредством ганкелевой матрицы, предложен профессором А.Ю.Утешевым.

Алгоритм

- Из f $\{d_i\}$ – коэффициенты разложения $\frac{1}{f}$ в ряд Лорана
- Из $\{d_i\}$ и a получается последовательность $\{c_i\}_{i=1}^{2*(N-1)}$
- $\{c_i\}_{i=1}^{2*(N-1)}$ - на вход алгоритму Берлекэмпа-Месси, на выходе - набор коэффициентов обратного элемента

Результаты

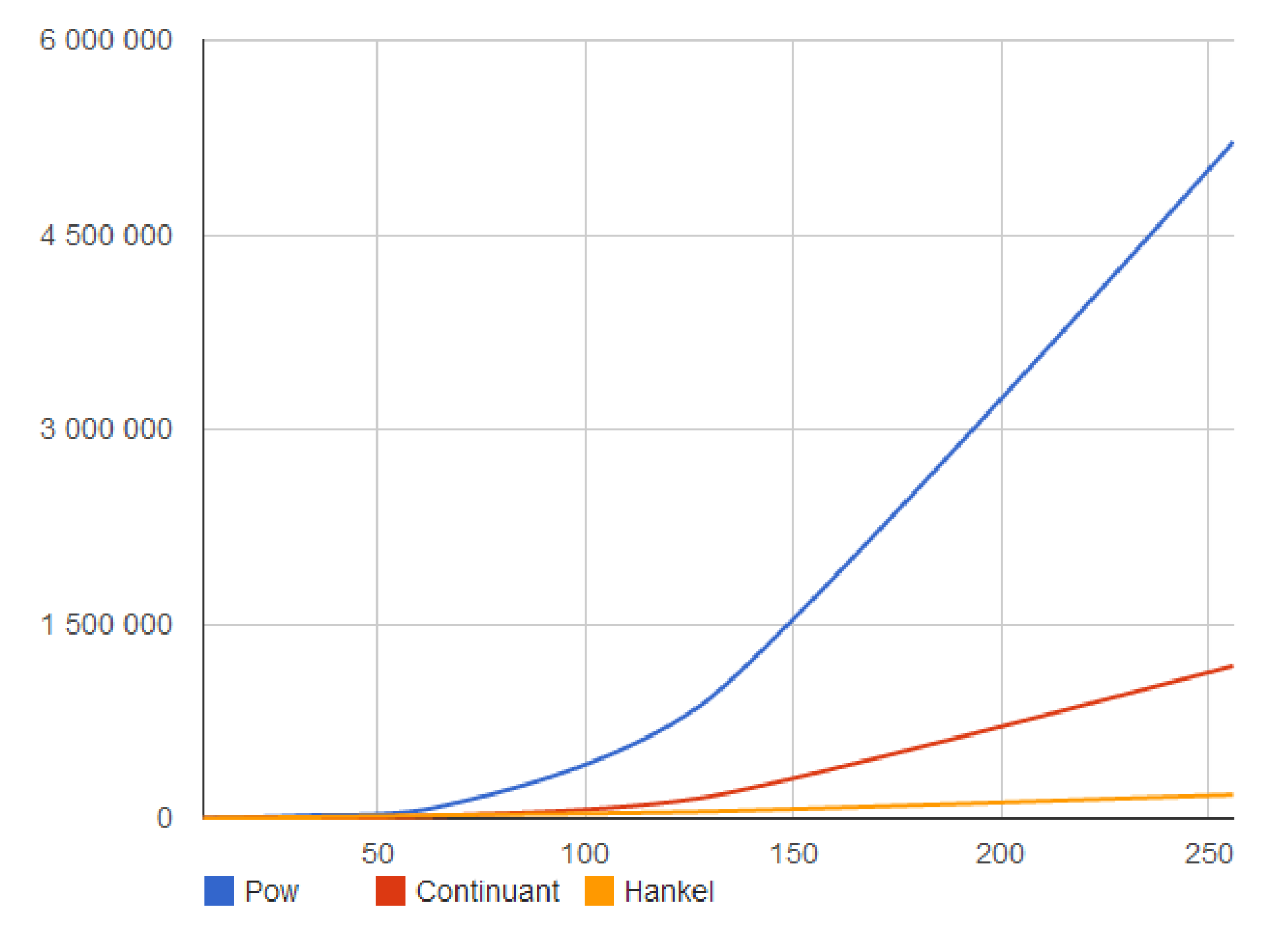
Замеры скорости работы производились на компьютере со следующими характеристиками:

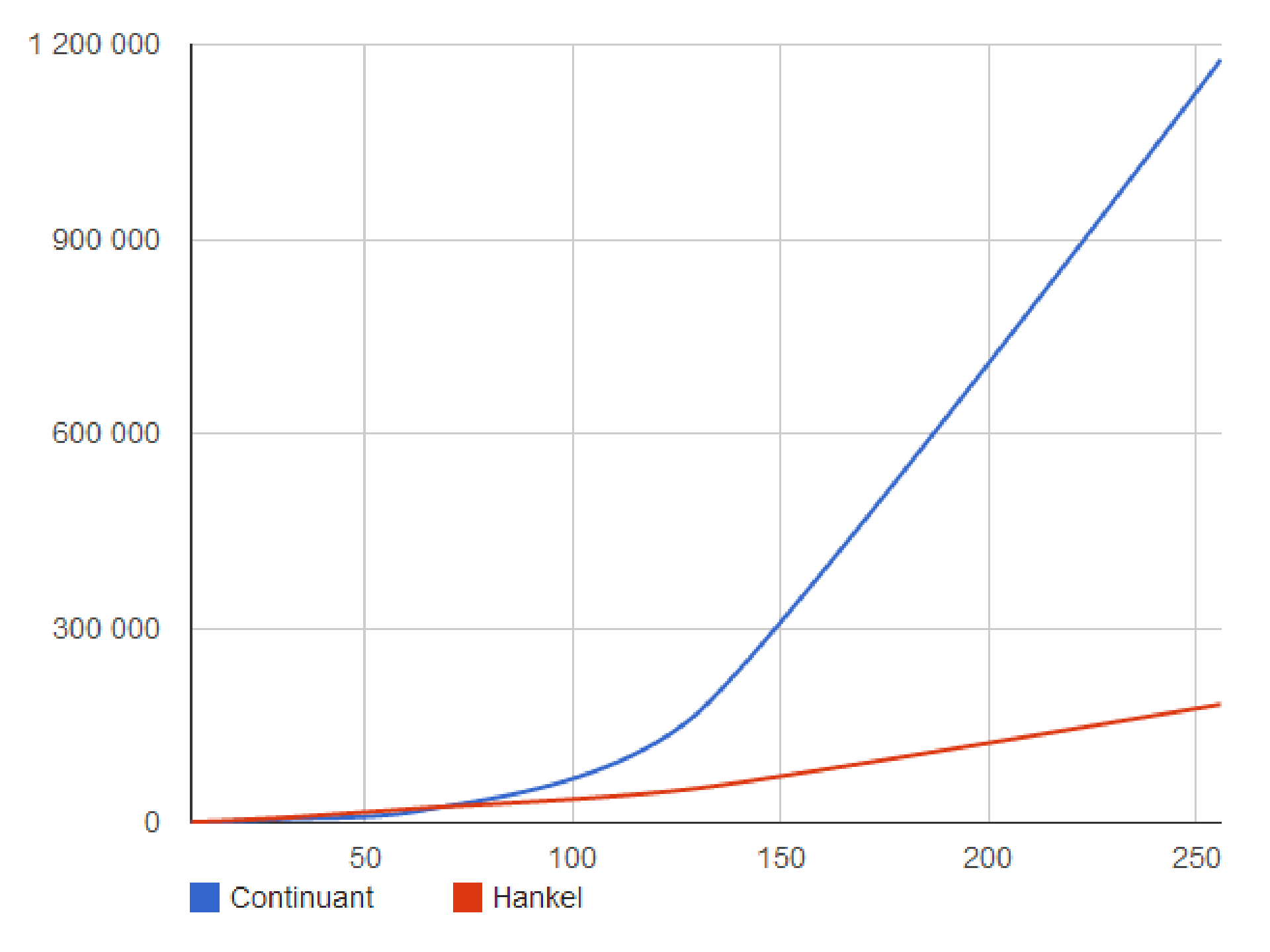
- Процессор - Intel® Xeon(R) CPU E5-2620 0 @2.00GHz × 18
- ОЗУ - 39,4 ГиБ
- ОС – Debian 7.0 x64

Замеры

N	Power	Continuant	Hankel
8	741	480	1181
16	2395	1481	2991
32	8929	5675	8191
64	35774	19169	22671
128	878762	158103	51344
256	5216242	1176736	182407

Время работы каждого алгоритма для обращения одного элемента (ts)


6 000 000
4 500 000
3 000 000
1 500 000
0
50
100
150
200
250
Pow
Continuant
Hankel


1 200 000
900 000
600 000
300 000
0
50
100
150
200
250
Continuant
Hankel

Заключение

- Реализованы и сравнены 3 алгоритма обращения элементов $GF(2^N)$
- Сделан доклад на конференции СПИСОК
- Результат работы будет применён компанией RAIDIX