УДК 004.421.2:004.032.24:512.714+

Исследование масштабируемости параллельных вычислений инволютивных базисов и базисов Грёбнера на многоядерном SMP компьютере

Д. А. Янович

Лаборатория информационных технологий
Объединённый институт ядерных исследований
ул. Жолио-Кюри, д. 6, Дубна, Московская область, Россия, 141980

В прошлых работах автором была представлена программная реализация двух подходов к параллелизации вычислений базисов Грёбнера и инволютивных базисов полиномиальных систем: на уровне редукций полиномов с вычислениями, проводимыми в кольце $\mathbb{Z}$ и на уровне вычисления базисов целиком по модулю простого числа с последующим подъёмом результатов. Их масштабируемость была исследована только на восьмиядерном компьютере.

В этой работе приводятся результаты тестирования улучшенной реализации данных алгоритмов на компьютере с 32 ядрами, производится анализ масштабируемости и факторов, на неё влияющих.

Ключевые слова: базис Грёбнера, базис Жане, параллельные вычисления, масштабируемость.

1. Введение

Мы будем использовать следующие обозначения:

$\mathbb{X} = \{x_1, \ldots, x_n\}$ — множество полиномиальных переменных.

$\mathbb{R} = \mathbb{Z}[\mathbb{X}]$ — кольцо многочленов с целочисленными коэффициентами.

$\mathbb{M}$ — множество мономов, т.е. произведение степеней переменных из $\mathbb{X}$ с целыми неотрицательными показателями.

$\succ$ — допустимый порядок на мономах, такой что $x_1 \succ x_2 \succ \cdots \succ x_n$.

$u \mid v$ — обычное отношение делимости монома v мономом u. Если $u \mid v$ и $\deg(u) < \deg(v)$, т.е. если u является *собственным делителем* v, мы будем записывать это условие как $v \sqsupset u$.

$lm(f)$ и $lt(f)$ — старший моном и старший одночлен многочлена $f \in \mathbb{R} \setminus \{0\}$, соответственно.

$lm(F)$ — набор старших мономов полиномиального множества $F \subset \mathbb{R} \setminus \{0\}$.

Неким образом разделим переменные лидирующего монома полинома $f \in G$: $M(f, F) \subset \mathbb{X}$ и $NM(f, F) = \mathbb{X} \setminus M(f, F)$ — множество мультипликативных и немультипликативных переменных соответственно.

Разделение переменных на немультипликативные и мультипликативные переменные порождает инволютивной деление мономов [1, 2]. Это деление определяется по заданному конечному набору многочленов F и порядку на мономах $\succ$ следующим образом: если мономы $u \in lm(F)$, v и w связаны соотношением $w = u \cdot v$ и при этом моном v содержит только мультипликативные переменные для u, то u является *инволютивным делителем* монома w. В этом случае мы будем записывать отношение инволютивной делимости как $u \mid_L w$.

Конечное множество F ненулевых многочленов является инволютивно авторедуцированным, если каждый моном, входящий в $f \in F$, не имеет инволютивных делителей среди $lm(F) \setminus \{lm(f)\}$. *инволютивная нормальная форма* $NF_L(p, F)$ многочлена $p \notin F$ определяется так:

$$NF_L(p, F) = \tilde{p} = p - \sum_{ij} \alpha_{ij} m_{ij} g_j,$$

Статья поступила в редакцию 31 мая 2013 г.

Работа частично поддержана грантами РФФИ 12-07-00294, 13-01-00668 и грантом 3802.2012.2 Министерства образования и науки РФ.

где $\alpha_{ij} \in \mathbb{K},\ g_j \in F,\ m_{ij} \in L(lm(g_j), lm(F)),\ lm(m_{ij}g_j) \preceq lm(p)$ и $\tilde{p}$ не содержит мономов, имеющих инволютивного делителя среди $lm(F)$.

Для заданного идеала $I \subset \mathbb{R}$ и порядка на мономах $\succ$, конечное, инволютивно авторедуцированное множество ненулевых многочленов $G \subset \mathbb{R}$, порождающее I, является его *инволютивным базисом*, если выполнено следующее [1]:

$$(\forall f \in G) (\forall x_i \in NM_L(f, G)) \ [NF_L(x_i \cdot f, G) = 0].$$

Произведение $x_i \cdot f$ многочлена $f \in F$ и $x_i \in NM_L(f, F)$ называется *немультипликативным продолжением* f. Тем самым для инволютивного базиса любое его немультипликативное продолжение приводится мультипликативно к нулю.

Базис Грёбнера получается из инволютивного путём авторедукции по обычному делению [1].

Кратко, алгоритм получения инволютивного базиса можно записать так:
1: **INPUT** F, L, $\prec$
2: **OUTPUT** T – инволютивный базис F
3: $T := \emptyset\ \ Q := F$
4: **WHILE** $Q \neq \emptyset$
5: $\quad T := T \cup \{p \mid lm(p) = \min(lm(Q)),\ NF_L(p, T) \neq 0\}$
6: $\quad Q := Q \setminus \{p\},\ Q := Q \cup p \cdot NM_L(p, T)$

Одним из достоинств инволютивного алгоритма является достаточно слабая связанность по данным между операциями, проводимыми в ходе вычислений, что позволяет построить несколько его параллельных вариантов [3–7].

2. Исследование масштабируемости

В ходе вычислений базисов почти все время последовательный алгоритм проводит в вычислениях отдельных инволютивных нормальных форм (см. шаг 5). Распределив их вычисление по разным ядрам, получим первый алгоритм для исследования ($\mathbb{Z}$ в легенде). Он интересен тем, что более «выгодные» (с более простой арифметикой) элементы базиса вычисляются раньше «невыгодных». В некоторых примерах приводит к интересному эффекту суперлинейного ускорения вычислений [6].

Второй вариант параллельного алгоритма вычисляет множество модулярных образов базиса, восстанавливая прообраз с помощью Китайской теоремы об остатках [7]($\mathbb{Z}_p$ в легенде). При реализации этого алгоритма необходимо помнить о существовании «неудачных» модулей, которые несут неполную информацию о прообразе, поэтому модулярные образы и восстановленный базис надо подвергать серии тестов для отсеивания неправильных результатов.

Реализации этих алгоритмов были протестированы на 32-х ядерных серверах, предоставляемых сервисом *Elastic Cloud Computing*, использовался шаблон `cc2.8xlarge`, соответствующий 32х-ядерному серверу с процессорами Xeon E5-2670@2.60GHz с 64Гб RAM. Коэффициенты масштабирования двух характерных примеров (с большим временем счета в последовательном варианте) отображены на рис. 1.

`Jcf26` представляет класс примеров с большой промежуточной арифметикой. Как видно, примеры такого класса демонстрируют отличную масштабируемость в случае использования модулярного алгоритма и постоянное, но не сильное, снижение эффективности на процессор в случае вычислений в «полной» арифметике.

`Redcyc7` является представителем примеров с малой промежуточной арифметикой. Как видно из диаграммы, примеры данного типа оправдано считать на небольших машинах с 8–12 ядрами, не более.

В обоих примерах наблюдаются места локального роста производительности в случае модулярного алгоритма (самый яркий пример — 8 ядер в примере `redcyc7`). Эти выбросы можно объяснить тем, что иногда наибольший коэффициент получаемого базиса удаётся восстановить, посчитав количество модулярных

образов, кратное количеству процессоров, т.е. не считая лишних для восстановления базисов.

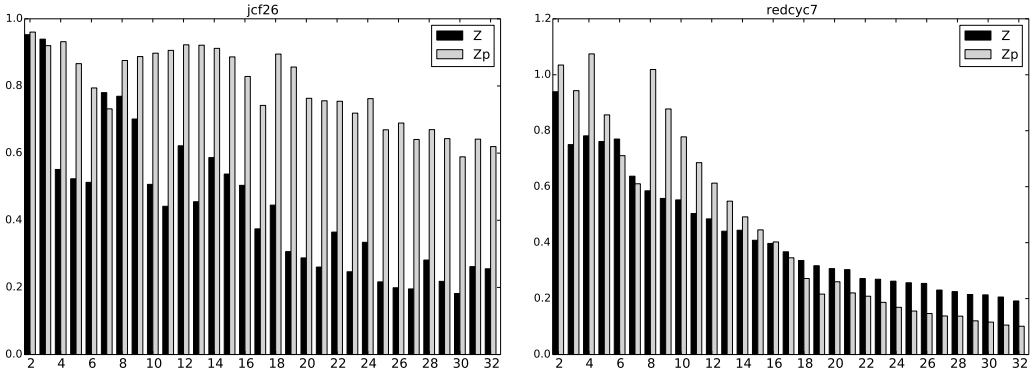

Рис. 1. Масштабируемость характерных примеров

3. Заключение

Как показывают диаграммы, масштабируемость тем лучше, чем длиннее промежуточные коэффициенты примера. В этом случае наиболее оправдано применять модулярный алгоритм, который позволяет избежать их вычисления вовсе. Если промежуточные коэффициенты невелики, можно применять любой вариант алгоритма, они обеспечат достаточно заметный прирост скорости. Также видно, что практически любой пример получит хорошее ускорение на легко доступных 8-ядерных машинах.

Литература

1. *Gerdt V. P., Blinkov Y. A.* Involutive Bases of Polynomial Ideals // Math. Comp. Sim. — 1998. — Vol. 45. — Pp. 519–542.
2. *Gerdt V. P., Blinkov Y. A.* Minimal Involutive Bases // Math. Comp. Sim. — 1998. — Vol. 45. — Pp. 543–560.
3. *Yanovich D. A.* Parallelization of an Algorithm for Computation of Involutive Janet Bases // Programming and Computer Software. — 2002. — Vol. 28, No 2. — Pp. 66–69.
4. *Gerdt V. P., Yanovich D. A.* Parallel Computation of Involutive and Gröbner Bases // "Computer Algebra in Scientific Computing / CASC 2004", V. G. Ganzha, E. W. Mayr, E. V. Vorozhtsov (Eds.). Institute of Informatics, Technical University of Munich, Garching. — 2004. — Pp. 185–194.
5. *Yanovich D. A.* Efficiency Estimate for Distributed Computation of Gröbner Bases and Involutive Bases // Programming and Computer Software. — 2008. — Vol. 34, No 4. — Pp. 210–215.
6. *Yanovich D. A.* Reduction-Level Parallel Computations of Gröbner and Janet Bases // Bulletin of Peoples' Friendship University of Russia. — 2010. — Vol. Mathematics. Information Sciences. Physics, No 2. — Pp. 19–24.
7. *Yanovich D. A.* Parallel Modular Computation of Gröbner and Involutive Bases // Programming and Computer Software. — 2013. — Vol. 39, No 2. — Pp. 110–113.

UDC 004.421.2:004.032.24:512.714+

Evaluation of Parallel Computations of Gröbner and Involutive Bases on the Massive SMP Computer

D. A. Yanovich

Laboratory of Information Technologies
Joint Institute for Nuclear Research
6, Joliot-Curie str., Dubna, Moscow region, Russia, 141980

In previous papers author presented realizations of two different approaches to parallelization of computation of Gröbner and involutive bases of polynomial systems with benchmarking on the 8-cores SMP computer: reduction-level parallelism with coefficients of polynomials in $\mathbb{Z}$-ring and basis-level parallelism using modular basis computation and lifting.

In this work further development of this algorithms described, benchmarking results and maximal speedup achieved on the massive 32-cores computer presented, scalability differences of algorithms investigated.

Key words and phrases: Gröbner bases, Janet bases, parallel computations, scalability.