УДК 004.75, 51-74

© 2011 р. М.Ю. Тягунова

Запорожский национальный технический университет, Запорожье

ВЫЧИСЛИТЕЛЬНАЯ СЛОЖНОСТЬ ГИБРИДНОГО АЛГОРИТМА

Определена вычислительная сложность гибридного алгоритма, реализующего метод диспетчеризации заданий в распределенных компьютерных системах. Выведена точная формула времени работы гибридного алгоритма, основываясь на количестве элементарных операций, приходящихся на каждый пункт алгоритма. А также определен порядок вычислительной сложности предложенного гибридного алгоритма. Это позволяет спрогнозировать время выполнения заданий в распределенной компьютерной системе.

The computational complexity of the hybrid algorithm, which implements the method of scheduling tasks in distributed computer systems, is determined. An exact formula for the execution time of the hybrid algorithm based on the number of elementary operations per each point of the algorithm is evaluated. Also degree of computational complexity of the proposed hybrid algorithm is defined. This allows to predict the execution time of tasks in a distributed computer system.

Введение

Принятие решения о скоординированном разделении ресурсов и распределении множества заданий в условиях большого количества виртуальных организаций формирует основную проблему грид-вычислений [1]. При этом, конечно, важным является выбранный метод диспетчеризации заданий, от которого зависит эффективность использования грид-ресурсов в целом. Вычислительная сложность алгоритмов является существенным показателем эффективности разработанного подхода по сравнению с существующими. Следовательно, оценив вычислительную сложность используемого алгоритма, можно делать выводы об его эффективности.

Рассмотрим модель диспетчеризации заданий, подробно описанную в [2]. Поскольку рассматриваемая задача диспетчеризации заданий представлена модифицированной транспортной задачей и конкретно в таком виде (представленном в [2]) не решалась до этого ни кем, то, следовательно, оценить на уровне вычислительной сложности предложенный для ее решения метод и реализующий его алгоритм по сравнению с другими является невозможным. Поэтому для определения вычислительной сложности предложенного гибридного алгоритма проанализируем полученные результаты времени работы алгоритма при различных комбинациях числа компьютеров и числа заданий и рассчитаем теоретически вычислительную сложность, основываясь на количестве элементарных операций, приходящихся на каждый пункт алгоритма.

1. Теоретическая оценка вычислительной сложности

Определим теоретическую оценку вычислительной сложности гибридного алгоритма следующим образом. Определим количество элементарных операций, приходящихся на каждый пункт алгоритма, а затем, в зависимости от хода алгоритма сложим или перемножим их между собой. Умножение будем применять к тем пунктам, которые повторяются в цикле, на количество повторяемых циклов. Во всех остальных случаях применим сложение. При расчете будем учитывать наихудшие ситуации, определяющие вычислительную сложность пункта. Отобразим в табл. 1 номер пункта гибридного алгоритма, подробно представленного в [3], и его вычислительную сложность. Таким образом, вычислительная сложность алгоритма равна:

$$\begin{aligned} &n\cdot\log n+m\cdot\log m+3n+2+m/3+3+(2(m-1)+\\ &+m/3+1+n+3n-1+1+2+n+2+1)\cdot h+2+2+\\ &+((n+2m-2)\cdot m^d+\left(n+2m+2m\cdot n\right)\cdot m^d+4n-1+\\ &+2+1)\cdot(k+1)+(2n+2+m-2+1+n+2m+2m\cdot n+\\ &+1)\cdot m^d+n+p+n+2m+2m\cdot n+1. \qquad (1)\end{aligned}$$

Таблица 1. Вычислительная сложность пунктов гибридного алгоритма

Номер пункта	Вычислительная сложность
1	$n\cdot\log n + m\cdot\log m$
2	$(n+n-1)+3+n=3n+2$
3	$m/3+3$
4	$2m$
5	$m/3$
6	1
7	n
8	$(n-1)+n+n=3n-1$
9	1
10	2
11	$(n-1)+1=n$
12	2
13	1
14	h
15	2
16	2
17-21	$n\cdot(1+2\cdot(m-1)/n)\cdot m^d = (n+2m-2)\cdot m^d$
22	$[n+(m+(m+1))]+[n\cdot m+(n\cdot m-1)]=n+2m+2m\cdot n$
23	$(n-1)+n+n+n=4n-1$
24	2
25	1
26	$k+1$
27	$2n$
28	2
29	$m-2$
30	$1+n+2m+2m\cdot n$
31	1
32	$n+2m+2m\cdot n$
33	m^d
34,35	$n+p$
36	$n+2m+2m\cdot n$
37	1

После несложных математических преобразований полученного выражения теоретической вычислительной сложности, получим следующее выражение:

$$n\cdot\log n + m\cdot\log m + 3n + m/3 + 9 + (2\cdot m + m/3 + 5n + 4)\cdot h + ((2n+4m+2m\cdot n-2)\cdot m^d + 4n+2)\cdot (k+1) + (3n+3m+2m\cdot n+2)\cdot m^d + 2n + 2m + 2m\cdot n + p + 1. \quad (2)$$

Теоретически рассчитанное время выполнения алгоритма $O\left(m^{d+1}\cdot n\right)$. Параметр d может быть установлен по желанию, но рекомендуемое значения один или два.

Учитывая, что при использовании гибридного метода максимальное значение параметра d на этапе генерации решений было взято равным двум, то получим, что вычислительная сложность разработанного метода для рассмотренных примеров $O(m^3\cdot n)$.

Однако, проведенные исследования показали, что если изменить вычислительную сложность алгоритма с $O(m^3\cdot n)$ на $O(m^2\cdot n)$, точность полученных результатов уменьшится в среднем на 10 %. Такое изменение параметра d (с двух до одного) приемлемо в случаях, когда суммарная длительность заданий, рассчитанная по наиболее мощному компьютеру и разделенная на число используемых компьютеров, меньше рассчитанного теоретическим путем времени.

2. Результаты экспериментальных исследований

Выше было приведено теоретическое обоснование зависимости времени работы разработанного гибридного алгоритма, реализующего гибридный метод, от числа заданий и числа компьютеров, участвующих при диспетчеризации. Для подтверждения правильности полученной вычислительной сложности, сравним теоретически рассчитанное время выполнения алгоритма с результатами времени работы гибридного алгоритма, полученными экспериментальным путем.

Время работы гибридного алгоритма при различных комбинациях числа компьютеров и числа заданий представлены в табл. 2.

Объясним некоторые приведенные результаты. Рассмотрим эксперимент под номером 16. Количество компьютеров, среди которых распределяем 170 заданий, равняется 5. При этом задача решена за количество шагов корректировки результирующей суммы, равном двум. Параметр точности и в то же время сложности метода при этом равняется двум. Время нахождения решения (время работы алгоритма) равняется 24,5 секунды. Рассмотрим эксперимент под номером 17. При этом изменено количество заданий на 180, количество шагов корректировки равняется одному, а время нахождения решения равняется 19,457 секунды. Уменьшение времени работы алгоритма при увеличении количества заданий объясняется уменьшением количества шагов корректировки с двух до одного, которая имеет непосредственное влияние на время. Рассмотрим эксперименты под номерами 30 и 38. Уменьшение времени нахождения решения гибридным методом в 186 раз обусловлено изменением значения параметра d с двух на единицу, который непосредственно влияет оценку сложности метода и, конечно, на время работы алгоритма.

В таблице 2 использованы следующие обозначения:

m – количество заданий в системе;

n – количество компьютеров;

k – число шагов корректировки результирующей суммы. Из экспериментальных исследований $k = \overline{1,20}$;

d – параметр точности метода. Оптимальное значение параметра, найденное экспериментальным путем, равно двум. Однако, его значение может варьироваться от единицы до n ($d = \overline{1,n}$);

h – число шагов процесса распределения всех заданий. Из экспериментальных исследований $h = \overline{1,5}$;

p – число шагов при проведении оператора мутации, всегда $p < n$.

3. Анализ результатов

Сравним время работы алгоритма, замеренное экспериментальным путем, с теоретически вычисленным временем работы алгоритма. При этом, значение, полученное при помощи выражения (1) делим на 8 300 000. Результаты сравнения экспериментальной и теоретической оценки времени выполнения алгоритма представлены в виде графиков на рис. 1 – 2.

При этом на рис. 1 построена зависимость времени выполнения алгоритма от количества заданий при константном числе компьютеров, равном пяти, а на рис. 2 – от количества компьютеров при постоянном числе заданий, равном 200.

На графиках (рис. 1, рис. 2), пунктирной линией с маркерами отображено значение вычислительной сложности алгоритма, определенное экспериментальным путем, сплошной – отображено теоретически рассчитанное время работы алгоритма, в основу которого положен гибридный алгоритм.

Как видно из графиков, теоретически рассчитанное и полученное практическим путем время почти совпали, что указывает на достоверность теоретически полученной оценки вычислительной сложности алгоритма.

Полученная оценка вычислительной сложности лучше, чем вычислительная сложность симплекс-метода (в худшем случае – экспоненциальная) и, конечно же, полного перебора. К примеру, время решения рассмотренного примера (5 компьютеров и 15 заданий) полным перебором составило 9 ч 30 мин, в то время как с использованием гибридного алгоритма – 0,05 сек. При этом найденный результат полностью совпал с результатом, полученным при полном переборе, что указывает на точность полученных результатов.

Таблица 2. Результаты экспериментальных исследований

Номер эксперимента	Число компьютеров, *n*	Число заданий, *m*	Число шагов корректиров-ки суммы, *k*	Параметр точности, *d*	Время работы алгоритма, *t* (c)
1	5	15	5	2	0.049
2	5	30	3	2	0.189
3	5	40	3	2	0.437
4	5	50	2	2	0.609
5	5	60	1	2	0.701
6	5	70	3	2	2.234
7	5	80	1	2	1.643
8	5	90	2	2	3.539
9	5	100	1	2	3.207
10	5	110	1	2	4.297
11	5	120	1	2	5.656
12	5	130	1	2	7.196
13	5	140	1	2	8.998
14	5	150	2	2	16.644
15	5	160	2	2	20.338
16	5	170	2	2	24.500
17	5	180	1	2	19.457
18	5	190	2	2	34.461
19	5	200	2	2	40.478
20	5	100	1	2	3.207
21	10	100	4	2	14.528
22	20	100	5	2	32.303
23	30	100	8	2	66.667
24	40	100	6	2	66.140
25	50	100	7	2	89.640
26	10	200	3	2	98.701
27	20	200	4	2	133.936
28	30	200	6	2	471.146
29	40	200	7	2	697.712
30	50	200	8	2	930.654
31	60	200	9	2	1215.202
32	70	200	7	2	1101.171
33	80	200	7	2	1170.867
34	90	200	7	2	1360.872
35	100	200	8	2	1552.924
36	50	400	6	2	6660.976
37	50	100	6	1	0.806
38	50	200	8	1	5.000
39	50	400	7	1	19.100
40	50	1000	7	1	167.000
41	50	2000	7	1	932.000
42	57	567	5	1	34.780
43	100	2000	9	1	2296.372
44	234	2345	15	1	115205.511
45	500	6000	8	1	167470.597

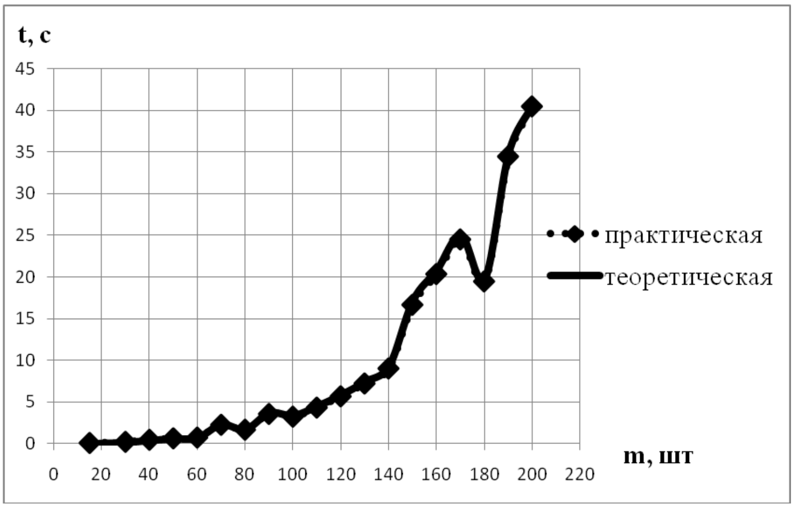

Рис. 1. Зависимость времени работы алгоритма от количества заданий (число компьютеров равно 5)

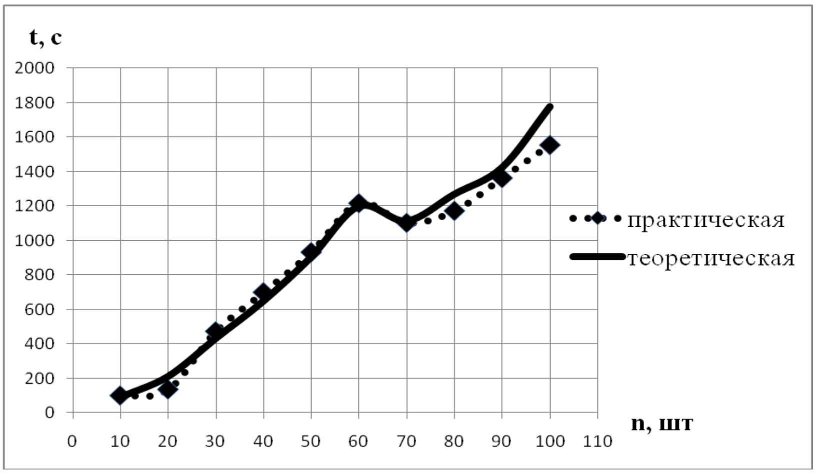

Рис. 2. Зависимость времени работы алгоритма от количества компьютеров (число заданий равно 200)

Выводы

Таким образом, разработанный гибридный метод диспетчеризации заданий в компьютерной системе с использованием основных операторов классического генетического алгоритма эффективно решает задачу диспетчеризации заданий в распределенных компьютерных системах. Полученная точная формула и, следовательно, порядок вычислительной сложности предложенного гибридного алгоритма позволят спрогнозировать время выполнения заданий в распределенной компьютерной системе.

СПИСОК ЛІТЕРАТУРИ

1. *Schopt J.* Ten Actions When SuperScheduling [Электронный ресурс] / *J. Schopt* // Document of Scheduling Working Group. Global Grid Forum. – July 2001. Режим доступа: http://www.ggf.org/documents/GFD.4.pdf
2. *Юрич М.Ю.* Математическая модель системы диспетчеризации задач /М.Ю. Юрич // Науковий вісник Чернівецького національного університету. Серія: Комп'ютерні системи та компоненти. – 2009. – Випуск 479. – С. 135–139
3. *Юрич М.Ю.* Гибридный алгоритм распределения задач в вычислительной системе / М.Ю. Юрич // Сборник научных трудов Донецкого национального технического университета. Серия: Информатика, кибернетика и вычислительная техника. – 2010. – № 12 (165) – С. 66–71

М.Ю. Тягунова. **Обчислювальна складність гібридного алгоритму.**

Визначено обчислювальну складність гібридного алгоритму, який реалізує метод диспетчеризації завдань у розподілених комп'ютерних системах. Виведено точну формулу часу роботи гібридного алгоритму, що грунтується на кількості елементарних операцій, що припадають на кожен пункт алгоритму. А також визначено порядок обчислювальної складності запропонованого гібридного алгоритму. Це дозволяє спрогнозувати час виконання завдань у розподіленій комп'ютерній системі.