

РЕЗУЛЬТАТЫ РАБОТЫ ГИБРИДНОГО МЕТОДА ПРИ ПАРАЛЛЕЛЬНОЙ РЕАЛИЗАЦИИ

В данной статье приведены результаты исследования гибридного алгоритма. Выявлено, что наиболее затратным является этап генерации решений. Данный этап был распараллелен на двух процессорах, что увеличило скорость распределения заданий в распределенной компьютерной системе почти в два раза. Также приведены результаты диспетчеризации многопроцессорных заданий.

The paper describes research results of hybrid algorithm. Revealed that the most expensive step of algorithm is the generation of solutions. This stage was parallelized on two processors, which increased the speed of distribution of tasks in a distributed computer system nearly doubled. Results of scheduling multiprocessor tasks are also presented.

Введение

В современном мире, когда увеличение прибыли стоит на первом месте практически у всех предприятий, сокращение времени выполнения тех или иных работ способствует этому. Распараллеливание операций, а особенно вычислительных, значительно сокращает временные, а как следствие, и материальные затраты.

Как описывалось ранее в [1], для диспетчеризации заданий в распределенных компьютерных системах был разработан гибридный алгоритм, позволяющий сократить максимальное время выполнения заданий в системе в среднем на 28.4%, а суммарное – на 6.5%, по сравнению с алгоритмами FCFS и Backfill.

Однако, достигнутые результаты могут быть улучшены. С целью сокращения времени работы гибридного алгоритма, реализующего гибридный метод, было применено распараллеливание генетического алгоритма, входящего в состав гибридного метода.

1. Постановка задачи

За счет применения параллельного генетического алгоритма появляется возможность увеличить скорость нахождения решения оптимизационной задачи не только путем использования новой технологии эволюционного моделирования для задач оптимизации, но и путем распараллеливания разработанного генетического алгоритма при конкретной его реализации.

Проведенные экспериментальные исследо-

вания (табл.1) показали, что наиболее затратным по времени при использовании гибридного алгоритма является этап генерации решений. Поэтому было проведено распараллеливание данного этапа для выполнения на двух процессорах.

2. Результаты работы

Сравнение результатов работы гибридного метода при последовательной и параллельной реализациях представлены в табл. 2.

Ускорение при использовании параллельного алгоритма рассчитываем по следующей формуле:

$$S = \frac{T_s}{T_p}, \quad (1)$$

где T_s , T_p – время работы алгоритма при последовательной и параллельной реализации алгоритма соответственно.

Эффективность при использовании параллельного алгоритма на p процессорах рассчитываем как:

$$F_p = \frac{S}{p}. \quad (2)$$

В среднем ускорение при использовании параллельного алгоритма на двух процессорах равно 1.83, а эффективность – 0.92, что говорит об эффективности использования параллельного генетического алгоритма при реализации гибридного метода для диспетчеризации заданий в распределенных компьютерных системах.

Таблица 1. Результаты экспериментальных исследований

n	m	k	Начало генерации решений	Окончание генерации решений	Общее время решения (t)
5	15	5	0,019	0,047	0,049
5	30	3	0,006	0,183	0,189
5	40	3	0,003	0,431	0,437
5	50	2	0,003	0,605	0,609
5	60	1	0,003	0,693	0,701
5	70	3	0,004	2,228	2,234
5	80	1	0,006	1,639	1,643
5	90	2	0,005	3,532	3,539
5	100	1	0,005	3,201	3,207
5	110	1	0,007	04,293	04,297
5	120	1	0,008	05,649	05,656
5	130	1	0,007	07,186	07,196
5	140	1	0,006	08,991	08,998
5	150	2	0,006	16,636	16,644
5	160	2	0,009	20,330	20,338
5	170	2	0,008	24,492	24,500
5	180	1	0,010	19,449	19,457
5	190	2	0,009	34,451	34,461
5	200	2	0,008	40,468	40,478
10	100	4	0,008	14,523	14,528
20	100	5	0,016	32,292	32,303
30	100	8	0,02	66,652	66,667
40	100	6	0,026	66,122	66,14
50	100	7	0,038	89,572	89,640
10	200	3	0,014	98,692	98,701
20	200	4	0,026	133,919	133,936
30	200	6	0,040	471,118	471,146
40	200	7	0,051	697,683	697,712
50	200	8	0,095	930,613	930,654
60	200	9	0,085	1215,162	1215,202
70	200	7	0,026	1101,116	1101,171
80	200	7	0,026	1170,774	1170,867
90	200	7	0,151	1360,800	1360,872
100	200	8	0,145	1552,777	1552,924

В таблице приведены следующие обозначения:

n – число компьютеров системы;

m – число поступающих для распределения заданий;

k – количество шагов корректировки суммы.

3. Распределение многопроцессорных заданий

Если отобразить графически эффективность работы разработанного гибридного алгоритма диспетчеризации заданий по сравнению с наиболее распространенными алгоритмами FCFS и

Backfill для многопроцессорных заданий, то получим результаты, представленные на рис. 1.

Для сравнения на рис. 2 и рис. 3 приведены результаты работы алгоритмов FCFS, и алгоритма Backfill. Подробная сравнительная информация об этих двух алгоритмах описана в [2].

На рис.1-3 представлены результаты для обобщенного случая диспетчеризации многопроцессорных заданий.

Как видно из рис. 1-3, предложенный гибридный алгоритм диспетчеризации заданий выполняет задания за меньшее время.

Таблица 2. Сравнение последовательной и параллельной реализации гибридного метода

номер экспе- римен- та	Число компьюте- ров, n	Число зада- ний, m	Время работы алгоритма, t (с)	Время работы ал- горитма на 2-х процессорах, tp (с)	Ускорение при ис- пользовании парал- лельного алгоритма на 2 процессорах, t/tp	Эффективность при использовании парал- лельного алгоритма на 2 процессорах, t/2*tp
1	5	15	0.049	0.026	1.77	0.88
2	5	30	0.189	0.096	1.87	0.93
3	5	40	0.437	0.228	1.82	0.91
4	5	50	0.609	0.309	1.87	0.94
5	5	60	0.701	0.355	1.87	0.94
6	5	70	2.234	1.130	1.88	0.94
7	5	80	1.643	0.844	1.85	0.92
8	5	90	3.539	1.803	1.86	0.93
9	5	100	3.207	1.617	1.88	0.94
10	5	110	4.297	2.164	1.89	0.94
11	5	120	5.656	2.884	1.86	0.93
12	5	130	7.196	3.614	1.89	0.95
13	5	140	8.998	4.556	1.87	0.94
14	5	150	16.644	8.441	1.87	0.94
15	5	160	20.338	10.328	1.87	0.93
16	5	170	24.500	12.459	1.87	0.93
17	5	180	19.457	9.818	1.88	0.94
18	5	190	34.461	17.456	1.87	0.94
19	5	200	40.478	20.524	1.87	0.94
20	5	100	3.207	1.614	1.89	0.94
21	10	100	14.528	7.358	1.87	0.94
22	20	100	32.303	16.402	1.87	0.93
23	30	100	66.667	33.758	1.87	0.94
24	40	100	66.140	33.272	1.89	0.94
25	50	100	89.640	49.880	1.70	0.85
26	10	200	98.701	52.781	1.77	0.89
27	20	200	133.936	73.189	1.73	0.87
28	30	200	471.146	237.953	1.88	0.94
29	40	200	697.712	365.294	1.81	0.91
30	50	200	930.654	479.719	1.84	0.92
31	60	200	1215.202	616.854	1.87	0.94
32	70	200	1101.171	564.703	1.85	0.93
33	80	200	1170.867	643.334	1.72	0.86
34	90	200	1360.872	705.115	1.83	0.92
35	100	200	1552.924	800.476	1.84	0.92
36	50	400	6660.976	3659.877	1.72	0.86
37	50	100	0.806	0.431	1.77	0.89
38	50	200	5.000	2.646	1.79	0.90
39	50	400	19.100	10.269	1.76	0.88
40	50	1000	167.000	86.082	1.84	0.92
41	50	2000	932.000	482.902	1.83	0.92
42	57	567	34.780	17.477	1.89	0.95
43	100	2000	2296.372	1165.671	1.87	0.94
44	234	2345	115205.511	64360.621	1.69	0.85
45	500	6000	167470.597	92016.812	1.72	0.86

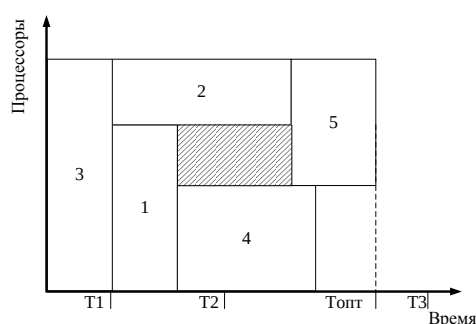


Рис. 1. Работа гибридного алгоритма

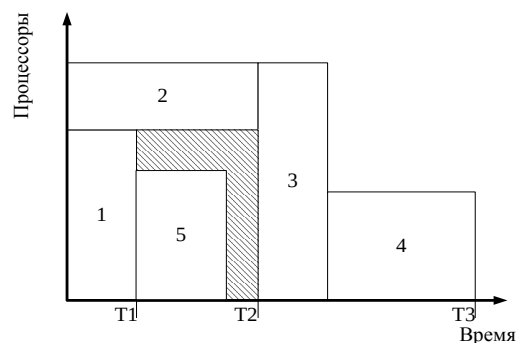


Рис. 3. Работа алгоритма Backfill

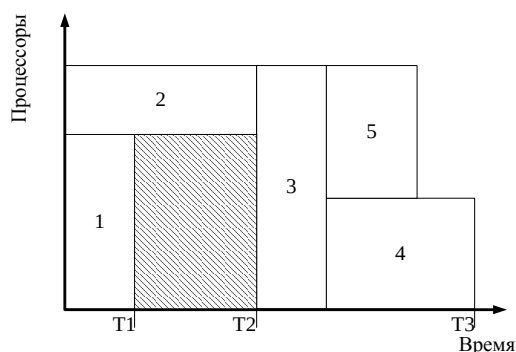


Рис. 2. Работа алгоритма FCFS

На рис. 1-3 области, в которые не распределено ни одно задание, называемые окнами, заштрихованы. При этом, если для представленного обобщенного случая диспетчеризации заданий подсчитать, насколько эффективнее является использование гибридного метода, то получим следующие результаты, выраженные в условных единицах (табл.3).

Как видно из табл. 3, и по занимаемой площади окон, и по максимальному времени выполнения заданий гибридный метод диспетчеризации многопроцессорных заданий работает эффективнее, чем FCFS и Backfill.

Таблица 3. Результаты работы алгоритмов для многопроцессорных заданий

	FCFS	Backfill	Гибридный метод	Эффективность гибридного метода по сравнению с FCFS	Эффективность гибридного метода по сравнению с Backfill
Площадь окна	17,9	8,1	7,4	141%	9,1%
Максимальное время выполнения	121	121	105	15,2%	15,2%

Висновки

Таким образом, в данной работе разработана параллельная реализация гибридного метода. Полученные результаты показали, что при распараллеливании этапа генерации на двух процессорах ускорение в среднем равняется 1,83, а эффективность составляет 0,92.

Также приведены результаты распределения многопроцессорных заданий. Эффективность гибридного метода по сравнению алгоритмом с FCFS – 141%, а по сравнению с Backfill – 9,1%.

СПИСОК ЛІТЕРАТУРИ

1. 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. – С. 66-72.

2. Y. Zhang, H. Franke, J. E. Moreira, A. Sivasubramaniam. A Comparative Analysis of Space- and Time-Sharing Techniques for Parallel Job Scheduling in Large Scale Parallel Systems [Электронный ресурс] // Режим доступа: http://csl.cse.psu.edu/publications/tpds_gang02.pdf

М.Ю. Тягунова, Р.К. Кудерметов, Д.В. Тягунов, Т.А. Паромова. **The results of the hybrid method for parallel realizations.**

М.Ю. Тягунова, Р.К. Кудерметов, Д.В. Тягунов, Т.О. Паромова. **Результати роботи гібридного методу при паралельній реалізації**

У даній статті наведено результати дослідження гібридного алгоритму. Виявлено, що найбільш витратним є етап генерації рішень. Даний етап було розпаралелено на двох процесорах, що збільшило швидкість розподілу завдань у розподіленій комп'ютерній системі майже у два рази. Також наведено результати диспетчеризації багатопроцесорних завдань.

