В.Н. Вагин, А.В. Деревянко, В.П. Кутепов

Алгоритмы параллельного логического вывода и исследование их эффективности на компьютерных системах

Аннотация. В статье приведены результаты исследования на кластерных системах с многоядерными узлами, созданных метода и алгоритма, реализующих параллельный вывод при доказательстве формул в логике первого порядка. Алгоритм основан на резолютивном методе вывода Дж. Робинсона и для его распараллеливания предложена процедура, позволяющая динамически управлять равномерным разделением множества резольвируемых дизъюнктов, распределяемых для вывода на узлы компьютерной системы. Также введен ряд эвристик, позволяющих существенно уменьшить время на реализацию управляющих решений. Экспериментами показано, что достигаемый эффект по ускорению выше, чем у ранее исследованных алгоритмов параллельного вывода.

Ключевые слова: параллельный логический вывод, принцип резолюции, параллельные вычисления, логика.

Введение

Автоматизация процедур доказательства в логических теориях, в частности в логике первого порядка [1, 2], существенно расширила их применение на практике. Для упрощения описания предметной области и решаемой задачи логическими средствами, а также для реализации на компьютере алгоритмов логического вывода были разработаны языки логического программирования Prolog, Planner и другие. Вместе с тем, известно, что проблема вывода в логике первого порядка является полуразрешимой, а сложность алгоритмов растет экспоненциально в зависимости от сложности задачи и длительное время выполнения алгоритма может поставить пользователя перед неразрешимой дилеммой: либо решить, что вывод отсутствует, либо ожидать неопределенное время, пока он не будет получен. Поэтому разработка эффективных параллельных алгоритмов логического вывода с возможностью их реализации на современных компьютерных системах является актуальной задачей и расширяет круг задач, которые могут быть решены с помощью логических средств.

Разработанный в статье алгоритм параллельного вывода в логике первого порядка основан на принципе резолюции, предложенном Дж. Робинсоном [2] и широко применяемом на практике в автоматическом доказательстве теорем.

Общая идея резолютивного вывода заключается в представлении отрицания доказываемой формулы логики первого порядка в конъюнктивной нормальной форме [2] и последующем доказательстве ее невыполнимости путем процедур попарного резольвирования порождаемых дизъюнктов. Этот процесс продолжается до тех пор, пока не будет выведен «пустой» дизъюнкт (значение ложь), или процесс резольвирования не имеет продолжения.

Алгоритмы логического вывода, основанные на принципе резолюции, хорошо распараллеливаются [3, 4] и поэтому открываются большие возможности для существенного уменьшения времени вывода за счет применения современных компьютерных систем.

Тем не менее, результатов по созданию и исследованию эффективности алгоритмов параллельного логического вывода на удивление мало.

В статьях [5, 6] описаны результаты экспериментального исследования на многоядерных

компьютерах, предложенных авторами алгоритмов параллельного вывода в логике первого порядка, а в статье [4] эти исследования расширены на случай применения кластерных систем.

Предложенные в статье [4] алгоритмы параллельного логического вывода имеют существенный недостаток, который заключается в централизации управления выводом, которое реализуется на одном выделенном узле кластера. В созданном алгоритме этот недостаток преодолевается путём децентрализации выполнения пересылок дизъюнктов между узлами кластера на основе применения специальных эвристик с целью достижения равномерной загруженности узлов. Кроме того, при реализации алгоритма параллельного вывода подмножество резольвируемых дизъюнктов упорядочивается по сложности и хранится в виде ассоциативного массива. Как показывают эксперименты, это существенно уменьшает время, затрачиваемое на выполнение необходимых операций резольвирования и, главное, позволяет увеличить интенсивность порождения новых дизъюнктов при их резольвировании в порядке уменьшения сложности. Принятые решения позволяют почти на порядок увеличить ускорение резольвирования при реализации на кластерах с многоядерными узлами по сравнению с алгоритмами, предложенными и исследованными в [4].

В разделе 1 описаны основополагающие решения, примененные при создании алгоритма параллельного вывода в логике первого порядка. В разделе 2 приведены результаты экспериментального исследования на кластере с многоядерными узлами эффективности алгоритма по критериям времени ускорения резольвирования на примере типовой логической задачи «Ханойские башни». В Заключении перечислены основные результаты и проблемы, требующие дополнительных исследований.

1. Описание алгоритма параллельного вывода

Вначале выскажем несколько общих соображений, касающихся критериев и параметров, по которым оценивается эффективность реализации параллелизма на КС, в том числе реализации параллельных алгоритмов логического вывода.

Для эффективной реализации параллельных алгоритмов на КС приходится учитывать множество факторов, связанных не только с применением соответствующего метода распараллеливания для конкретной задачи, но также касающихся особенностей использования языка для описания параллельной программы, организации компьютерной системы и ее технических характеристик операционных средств, предназначенных для управления параллельными программами.

Распараллеливание процедуры логического вывода предполагает разбиение множества резольвируемых дизъюнктов на более или менее равные по мощности подмножества, резольвирование которых выполняется одновременно несколькими процессами на соответствующих компонентах КС (ее узлах и процессорах).

Как показывают экспериментальные исследования, для заданной сложности задачи существует оптимальная степень распараллеливания, при которой достигается максимальное ускорение и, следовательно, минимальное время резольвирования. В нашем случае степень распараллеливания характеризуется количеством подмножеств резольвент, на которые разбивается общее множество резольвент, порожденных на определенном интервале выполнения вывода. Ясно, что до выполнения программы практически невозможно определить оптимальную степень распараллеливания и только путем многократных «прогонок» программы на КС можно это сделать.

Существенное влияние на ускорение вывода оказывают также особенности организации используемой компьютерной системы и ее технические характеристики. Для кластеров важна минимизация интенсивности обменов данными между узлами. Многопроцессорные системы с общей памятью (многоядерные компьютеры, используемые в кластерах) частично упрощают решение этой проблемы, однако имеют большие ограничения по масштабированию.

Наконец, большое влияние на ускорение вывода оказывают собственно программные решения по реализации функций вывода. Параллельно работающие процессы на узле оперируют со случайно изменяемыми очередями резольвируемых дизъюнктов, причем если несколько процессов обращаются к общей очереди, то приходиться использовать семафорную технику, что до 50-ти тактов увеличивает обращение к очереди. В разработанном для кластерных систем и экспериментально исследо-

ванном алгоритме параллельного логического вывода все вышеперечисленные проблемы, оказывающие большое влияние на ускорение, разрешаются путем применения специальных стратегий и эвристик.

Перейдем к описанию основных решений, которые были реализованы при создании алгоритма и программы параллельного логического вывода.

1. При инициализации алгоритма параллельного вывода на кластере создается ровно столько параллельных MPI процессов, сколько процессоров предполагается для использования.

2. Каждый процесс (каждая нить каждого узла) при резольвировании использует копию общей очереди (списка) резольвируемых дизъюнктов, которая пополняется в процессе вывода новыми дизъюнктами, выводимыми конкретными процессами. Это решение увеличивает требования к оперативной памяти, используемой при выводе, однако обеспечивает большую автономность процессов, выполняемых на многоядерном узле кластера.

3. Для обеспечения корректности алгоритма вывода, как известно, необходимо, чтобы любая пара дизъюнктов была прорезольвирована за конечное время. В созданном алгоритме это достигается путем следующего правила: *i*-й параллельный процесс выбирает дизъюнкты с номерами $i + N * k$, $k = 0, 1, 2, ..$, где N — общее количество параллельных процессов (для кластера с многоядерными узлами — ядер).

Полученные в результате новые резольвенты через определенный квант времени пересылаются в очереди дизъюнктов всех процессов, участвующих в резольвировании. Если процесс завершает резольвирование определенной для него порции дизъюнктов в своей очереди, то он сразу передает вновь полученные дизъюнкты всем остальным процессам. Легко показать, что данная стратегия гарантирует корректность вывода.

4. В созданном алгоритме, резольвирование дизъюнктов выполняется в порядке их сложности (количества литер в дизъюнкте). С этой целью дизъюнкты в очереди упорядочиваются по сложности. Это ускоряет процесс вывода, увеличивая количество вновь порождаемых дизъюнктов.

Разработанный на основе данных решений алгоритм параллельного вывода запрограммирован с использованием средств MPI для описания параллельных процессов на языке C++. Поскольку для выполнения MPI-программы применялся кластер с многоядерными узлами, управление параллельными процессами на узле осуществляется операционными средствами MULTITHREADING.

2.Экспериментальное исследование эффективности алгоритма на кластерных системах

Исследования проводились на кластере Московского энергетического института, включающего 16 узлов, каждый из которых состоит из двух двухъядерных процессоров с быстродействием ядра $4x10^9$ опер/с и пропускной способностью каналов передачи данных между узлами равной 10 Гбит/с; объем оперативной памяти узла – 4 Гб.

В качестве показателей эффективности алгоритмов параллельного логического вывода используется реальное время, затрачиваемое на вывод, и ускорение, значения которых в экспериментах измеряются в зависимости от варьируемого количества ядер и узлов кластера. При этом программа вывода устроена так, что количество ядер кластера выбирается равным количеству параллельных процессов MPI.

Чтобы оценить степень влияния на эффективность параллельного вывода принятых в п. 1 решений, созданный алгоритм (далее он обозначается А3) сравнивается с другими алгоритмами. В качестве последних рассматриваются:

- алгоритм А1 [4], в котором управление выводом осуществляется централизованно, одним из узлов кластера, на котором хранится общая очередь резольвируемых дизъюнктов, и дизъюнкты порциями рассылаются на узлы для вывода;
- алгоритм А2 [4], где применялась общая очередь дизъюнктов, копия которой находится на каждом узле и не применялось нитевое распараллеливание на узле. В алгоритме А2 каждая копия очереди дизъюнктов также упорядочивалась по сложности;
- алгоритм А3м, который представляет собой алгоритм А3 с отключенной сортировкой по сложности копий очередей дизъюнктов, что позволяет оценить, как сортировка влияет на ускорение с увеличением сложности вывода.

Основной особенностью алгоритма А3 является наличие отсортированной копии очереди дизъюнктов в каждой из нитей, работающих на узле.

Табл. 1. Сравнение алгоритмов логического вывода по времени выполнения (время указано в секундах)

Число нитей	5 колец			6 колец			7 колец	
	A1	A2	A3	A1	A2	A3	A1	A3
1	6,027	1,572	1,321	625,143	52,868	43,12	28146,162	22652,72
2	3,997	0,853	0,767	357,025	28,614	19,894	17651,863	14556,23
3	2,824	0,558	0,487	276,26	22,296	15,702	12665,371	9241,63
4	2,014	0,429	0,392	197,729	17,969	12,136	9192,652	6213,31

Табл. 2. Сравнение алгоритмов логического вывода по ускорению

Число нитей	5 колец			6 колец			7 колец	
	A1	A2	A3	A1	A2	A3	A1	A3
1	1	1	1	1	1	1	1	1
2	1,51	1,84	1,722	1,75	1,85	2,26	1,59	1,56
3	2,13	2,82	2,71	2,26	2,37	2,74	2,23	2,45
4	2,99	3,66	3,37	3,16	2,94	3,55	3,06	3,64

Так как показатели эффективности зависят от того, выполняется вывод на узле, который представляет собой многоядерный кластер с общей памятью, или на нескольких узлах, где требуется обмен данными между узлами, соответствующие результаты экспериментов разделены на две части.

В выполненных экспериментах исследования эффективности алгоритмов параллельного вывода варьировалось не только количество ядер или узлов кластера, но также и сложность задачи, в качестве которой была выбрана тестовая логическая задача «Ханойские башни» [2], сложность которой достаточно просто изменяется путем увеличения количества в ней колец.

2.1. Эффективность параллельного вывода на узле

В Табл. 1 и Табл. 2 приведены результаты эффективности работы на узле алгоритмов A1, A2 и A3 в зависимости от количества ядер и количества колец в задаче «Ханойские башни». Как следует из этих результатов, ускорение почти пропорционально увеличивается с увеличением количества ядер. При этом для алгоритма A3 время вывода уменьшается почти на порядок по сравнению с алгоритмом A1 для пяти и шести колец. Для семи колец это отношение меньше, что можно объяснить увеличением времени на упорядочение по сложности дизъюнктов в очередях, так как для семи колец эти очереди в среднем значительно больше, чем для пяти и шести колец.

2.2. Эффективность параллельного вывода при варьировании количества узлов

В Табл. 3 и Табл. 4 приведены данные, показывающие изменение времени выполнения вывода и ускорения вывода, при увеличении количества узлов для алгоритмов A1 и A3. Напомним, что алгоритм A1 [4] отличается от A3 тем, что в нем не используется стратегия упорядочения дизъюнктов в очередях по сложности. Также в A1 вывод осуществляется централизованно, одним из узлов кластера, на котором хранится общая очередь резольвируемых дизъюнктов.

Как следует из полученных результатов, алгоритм A3 почти на порядок уменьшает время логического вывода и увеличивает ускорение при решении задачи «Ханойские башни» с пятью и шестью кольцами по сравнению с алгоритмом A1. Для всех алгоритмов существует оптимальное количество узлов, после увеличения которого ускорение начинает изменяться незначительно. Это связано с тем, что с увеличением количества узлов увеличивается отрицательный эффект обменных взаимодействий между узлами. В Табл. 5 приведено сравнение времени выполнения алгоритмов A1, A3 и A3м.

Табл.3. Сравнение алгоритмов логического вывода по времени выполнения в зависимости от количества узлов (время указано в секундах)

Число узлов	**5 колец**		**6 колец**		**7 колец**	
	A1	A3	A1	A3	A1	A3
1	2,014	0,340094	197,729	12,136	9192,652	6213,31
2	1,87	0,229228	115,7718	8,0172	5382,367	2260,594
3	1,62	0,16345	99,66443	5,7134	4633,516	1228,808
4	1,51	0,168456	84,22819	5,12	3915,867	997,7629
5	1,52	0,1691	71,47651	4,3522	3323,026	858,2684
6	1,55	0,15721	66,10738	3,8311	3073,409	743,4577
7	1,61	0,159	58,38926	3,3088	2714,585	638,4704
8	1,68	0,202	54,36242	3,3088	2527,372	806,2317
9	1,67	0,19521	49,32886	2,7258	2293,356	691,6762
10	1,71	0,24552	47,31544	3,0176	2199,75	448,749
11	1,73	0,24969	42,61745	3,2277	1981,335	570,6245
12	1,75	0,23788	39,93289	3,6566	1856,526	463,946
13	1,82	0,3486	41,94631	3,3372	1950,133	491,4335
14	1,86	0,41552	39,26174	3,3117	1825,324	449,0298
15	1,96	0,30975	39,93289	4,028	1856,526	502,0048
16	1,94	0,332	36,57718	3,8092	1700,516	359,8292

Табл. 4. Сравнение алгоритмов логического вывода по ускорению в зависимости от количества узлов

Число узлов	**5 колец**		**6 колец**		**7 колец**	
	A1	A3	A1	A3	A1	A3
1	1	1	1	1	1	1
2	1,077005	1,820784	1,70792	1,354404	1,70792	1,832353
3	1,24321	2,356007	1,983948	1,721663	1,983948	3,370914
4	1,333775	2,502671	2,34754	2,576427	2,34754	4,151494
5	1,325	2,395712	2,766349	2,647643	2,766349	7,41974
6	1,299355	2,048705	2,991028	3,099635	2,991028	5,571543
7	1,250932	2,108434	3,386393	3,888746	3,386393	6,487704
8	1,19881	2,160745	3,637237	4,104437	3,637237	5,137737
9	1,205988	1,539419	4,008384	4,081248	4,008384	5,98865
10	1,177778	1,778865	4,178953	3,584594	4,178953	9,230565
11	1,164162	1,538908	4,639625	4,088949	4,639625	7,259076
12	1,150857	1,221962	4,951533	3,670346	4,951533	8,928209
13	1,106593	0,920139	4,713859	3,005746	4,713859	8,428825
14	1,082796	1,121795	5,036175	3,755067	5,036175	9,224793
15	1,027551	0,928117	4,951533	3,490566	4,951533	8,251329
16	1,034655	0,996465	5,405802	3,682486	5,405802	11,51159

Табл. 5 отражает ситуацию, когда отключение упорядочения по сложности помогло заметно уменьшить время выполнения алгоритма для сложной задачи вывода на семи кольцах. Напомним, что алгоритмы A3 и A3м отличаются от алгоритма A1 тем, что в них каждая нить каждого узла имеет собственную копию очереди дизъюнктов. Таким образом, можно сделать вывод, что на больших очередях дизъюнктов их сортировка существенно увеличивается по времени, поэтому от ее применения стоит отказаться. Тем не менее, на малых очередях (6 колец) применение эвристики имеет смысл.

Сравнение алгоритмов A1, A3 и A3м по времени вывода и ускорению при изменении количества узлов показывает, что для каждой сложности задачи существует некоторое оптимальное значение, после которого эти показатели уменьшаются. Это объясняется тем, что с увеличением степени распараллеливания (количества параллельных процессов) и соответственно количества узлов начинает оказывать заметное влияние интенсивность обменов между узлами кластера.

Табл. 5. Сравнение алгоритмов логического вывода А1, А3 и А3м по времени выполнения в зависимости от количества узлов (время указано в секундах)

Число узлов	**6 колец**			**7 колец**		
	А1	А3	А3м	А1	А3	А3м
1	197,729	12,136	15,612	9192,652	6213,31	1088,342
2	115,7718	8,0172	10,161	5382,367	2260,594	647,324
3	99,66443	5,7134	8,486	4633,516	1228,808	589,152
4	84,22819	5,12	7,53	3915,867	997,7629	511,236
5	71,47651	4,3522	7,12	3323,026	858,2684	502,974
6	66,10738	3,8311	6,73	3073,409	743,4577	497,246
7	58,38926	3,3088	5,98	2714,585	638,4704	432,658
8	54,36242	3,3088	5,77	2527,372	806,2317	413,491
9	49,32886	2,7258	5,61	2293,356	691,6762	387,395
10	47,31544	3,0176	5,43	2199,75	448,749	377,15
11	42,61745	3,2277	5,53	1981,335	570,6245	325,912
12	39,93289	3,6566	5,48	1856,526	463,946	317,423
13	41,94631	3,3372	5,87	1950,133	491,4335	325,632
14	39,26174	3,3117	6,23	1825,324	449,0298	291,24
15	39,93289	4,028	6,01	1856,526	502,0048	270,89
16	36,57718	3,8092	5,97	1700,516	359,8292	251,236

Заключение

Выполненные исследования показывают, что реализация параллельных алгоритмов на кластерных системах с многопроцессорными узлами позволяет существенно уменьшить время вывода. Однако для этого необходимо учитывать целый ряд факторов, касающихся не только выбора оптимальной степени распараллеливания, но также учета ограничений самой компьютерной системы: быстродействия процессоров, пропускной способности ее памяти и др.

Также алгоритмы параллельного вывода относятся к сильно связанным задачам в том смысле, что с увеличением степени распараллеливания пропорционально увеличивается количество связей между частями параллельного алгоритма, а, следовательно, интенсивность обменов данными между ними при выполнении алгоритма.

Как показывает качественный анализ других известных методов вывода в логике первого порядка, отличных от резолютивного метода Робинсона [1], они имеют еще большую связность и для их реализации необходимо использовать синхронизацию параллельно выполняемых процессов вывода, что является еще одним фактором, препятствующим увеличению достигаемого ускорения.

Литература

1. Вагин В.Н., Головина Е.Ю., Загорянская А.А., Фомина М.В. Достоверный и правдоподобный вывод в интеллектуальных системах / Под ред. В.Н. Вагина, Д.А. Поспелова. 2-е издание доп. и испр. М.: Физматлит, 2008. с. 712.
2. Чень Ч., Ли Р. Математическая логика и автоматическое доказательство теорем. М.: Наука. 1983.
3. Боначина М. П. Таксономия параллельных стратегий дедукции . AMAI. 2000. с. 1-36.
4. Кутепов В. П., Кумачев М. М. Параллельный логический вывод на компьютерных системах // Искусственный интеллект и принятие решений.2012, №2.
5. Вагин В.Н., Кутепов В.П., Хотимчук К.Ю. Параллельный логический вывод на многоядерном компьютере // Вестник Московского энергетического института. 2011. № 3. с. 82-87.
6. Зарецкий Д. С. Параллельная реализация принципа резолюции с использованием управляющих эвристик / Сборник трудов международной суперкомпьютерной конференции «Научный сервис в сети Интернет: суперкомпьютерные центры и задачи», Новороссийск, 2010. с. 611-618.

Вагин Вадим Николаевич. Профессор кафедры прикладной математики Московского энергетического института, действительный член Российской академии естественных наук и международной академии информатизации. Доктор технических наук. Область научных интересов: искусственный интеллект, принятие решений, методы и модели представления знаний, дедукция, абдукция и индуктивное формирование понятий. E-mail: vagin@appmat.ru

Деревянко Андрей Владимирович. Аспирант Московского энергетического института. Окончил Московский энергетический институт в 2016 году. Область научных интересов: искусственный интеллект, параллельные вычисления. E-mail: derevyankoanv@gmail.com

Кутепов Виталий Павлович. Профессор кафедры прикладной математики Московского энергетического института. Окончил Московский энергетический институт в 1961 году. Доктор технических наук. Область научных интересов: компьютерные системы, параллельные вычисления. E-mail: kutepov@appmat.ru

Parallel algorithms for inference and researching of their effectiveness on computer systems

V.N. Vagin, A.V. Derevyanko, V.P. Kutepov

Abstract. This article presents the results of research on cluster systems with multicore nodes to create a method and algorithm implementing parallel output in the proof of formulas in first order logic. The algorithm is based on the operative method of J. Robinson and the proposed procedure for its parallelization, which allows to dynamically control the uniform division of the set of resolvents distributed to the components of a computer system for infering. A number of heuristics are also introduced to significantly reduce the time to implement the decisions of the infering control. Experiments have shown that the effect on acceleration is higher than the previously studied parallel output algorithms.

Keywords: parallel inference, resolution principle, parallel computing, logic.

References

1. Vagin V.N., Golovina E.Iu., Zagorianskaia A.A., Fomina M.V. Dostovernyi i pravdopodobnyi vyvod v intellektual'nykh sistemakh Moscow, Fizmatlit, 2008. 712 p.
2. Chen, C., Li, R. Mathematical logic and automatic theorem proving Moscow, Nauka. 1983
3. Bonacina Maria Paola. A taxonomy of parallel strategies for deduction. AMAI. 2000. P. 1-36.
4. Kutepov V.P., Kumachev M. M. Parallel'nyy logicheskiy vyvod na komp'yuternykh sistemakh [Iskusstvennyy intellekt i prinyatiye resheniy] 2012, N2.
5. Vagin V.N., Kutepov V.P., Xotimcuk K. Iu. Parallel'nyy logicheskiy vyvod na mnogoyadernom komp'yutere [Vestnik Moskovskogo energeticheskogo instituta.] 2011. № 3. 82-87 p.
6. Zaretskiy D.S., Parallel'naya realizatsiya printsipa rezolyutsii s ispol'zovaniyem upravlyayushchikh evristik [Sbornik trudov mezhdunarodnoy superkomp'yuternoy konferentsii «Nauchnyy servis v seti Internet: superkomp'yuternyye tsentry i zadachi»], Novorossiysk, 2010. 611-618 p.

Vagin V.N. Doctor of Technical Sciences, Professor, Department of Applied Mathematics of the Moscow Power Engineering Institute, member of the Russian Academy of Natural Sciences and International Academy of Informatization. His research interests include artificial intelligence, decision-making methods and knowledge representation model, deduction, abduction and inductive formation of concepts. E-mail: vagin@appmat.ru

Derevyanko A.V. Graduate student of the Moscow Power Engineering Institute. He graduated from the Moscow Energy Institute in 2016. His research interests include artificial intelligence, parallel computing. E-mail: derevyankoanv@gmail.com

Kutepov V.P. Doctor of Technical Sciences, Professor, Department of Applied Mathematics of the Moscow Energy Institute. He graduated from the Moscow Energy Institute in 1961. Research interests: computer systems, parallel computing. E-mail: kutepov@appmat.ru