

УДК 681.3 + 519.68

**ПАРАЛЛЕЛЬНАЯ ЛИНЕАРИЗАЦИЯ ДЕРЕВА ЛОГИЧЕСКОГО ВЫВОДА В  
РАСПРЕДЕЛЁННЫХ СИСТЕМАХ УПРАВЛЕНИЯ ДЕДУКТИВНЫМИ БАЗАМИ  
ДАнных**

Пушкаренко С.А., Дацун Н.Н.  
Донецкий национальный технический университет

Рассматриваются проблемы реализации распределённых систем управления дедуктивными базами данных. Описаны предложения по повышению эффективности процесса линеаризации сложной структуры данных - дерева произвольной арности. Приведено описание моделирования процесса параллельной линеаризации и результаты исследований.

**1 Проблемы реализации распределённых систем управления дедуктивными базами данных**

Процесс параллельного логического вывода в исследуемых дедуктивных базах данных (ДБД) требует необходимости линеаризации дерева логического вывода, для

передачи его по коммуникационной среде на удалённые узлы многопроцессорной вычислительной системы (МВС) [1, 2].

Дерево логического вывода представляет собой сильноветвящееся дерево (дерево произвольной арности).

При большом количестве узлов в дереве процесс линеаризации является достаточно трудоёмкой операцией, требующей значительных временных и вычислительных затрат. В связи с этим возникает необходимость в программно-аппаратных средствах повышения эффективности процесса линеаризации.

## **2 Предложения по повышению эффективности процесса линеаризации**

С целью повышения эффективности процесса линеаризации предложено осуществлять процесс линеаризации параллельно.

Для осуществления процесса параллельной линеаризации управляющий узел МВС (узел 0), осуществляющий автоматическое распараллеливание логической программы и распределяющий задания по узлам МВС, должен являться SMP-системой либо иметь многоядерный процессор (см. рис. 1). Программа, выполняющая распараллеливание, в данном случае будет являться многопоточной. Основным потоком (планировщик) работающий на ядре 0, осуществляет распараллеливание логической программы в зависимости от особенностей данной программы и имеющегося в наличии количества вычислительных узлов. Основным потоком создаёт дополнительные потоки. Количество дополнительных потоков равно количеству вычислительных узлов, на которых будет выполняться выбранный планировщиком параллельный алгоритм логического вывода. Для каждого из вычислительных узлов отдельный поток выполняет линеаризацию выделенного планировщиком пространства в дереве логического вывода.

## **3 Моделирование процесса параллельной линеаризации**

Для моделирования процесса параллельной линеаризации использовалась среда Visual C++ 6.0. Была разработана многопоточная программа, моделирующая процесс линеаризации и делинеаризации дерева. Использовался алгоритм линеаризации, описанный в [1, 2]. Для работы с потоками были использованы следующие функции, определённые в заголовочных файлах `process.h` и `windows.h`: `_beginthreadex`, `_endthreadex`, `CloseHandle`, `WaitForMultipleObjects`, `SetThreadPriority`.

Структура данных “дерево” (TREE) включает в себя следующие поля:

- `id` - идентификатор узла дерева (нумеруется в порядке добавления);
- `predicate` - номер предиката;
- `arg` - список номеров аргументов предиката (переменных и констант);
- `brotherPtr` - ссылка на следующий узел текущего уровня;
- `childPtr` - ссылка на узел нижнего уровня;
- `parentPtr` - ссылка на родительский узел.

Нумерация узлов (поле `id`) производится во время добавления узлов. Предусмотрена возможность переиндексации (перенумерации) узлов дерева. Функция `Reindex` заново производит нумерацию узлов в порядке “сверху вниз / слева направо”. При делинеаризации выполняется поиск максимального значения поля `id`. Это позволяет после делинеаризации (“восстановления”) дерева производить добавление новых узлов, продолжая нумерацию узлов. Предусмотрена также возможность линеаризации в файл и делинеаризации из файла.

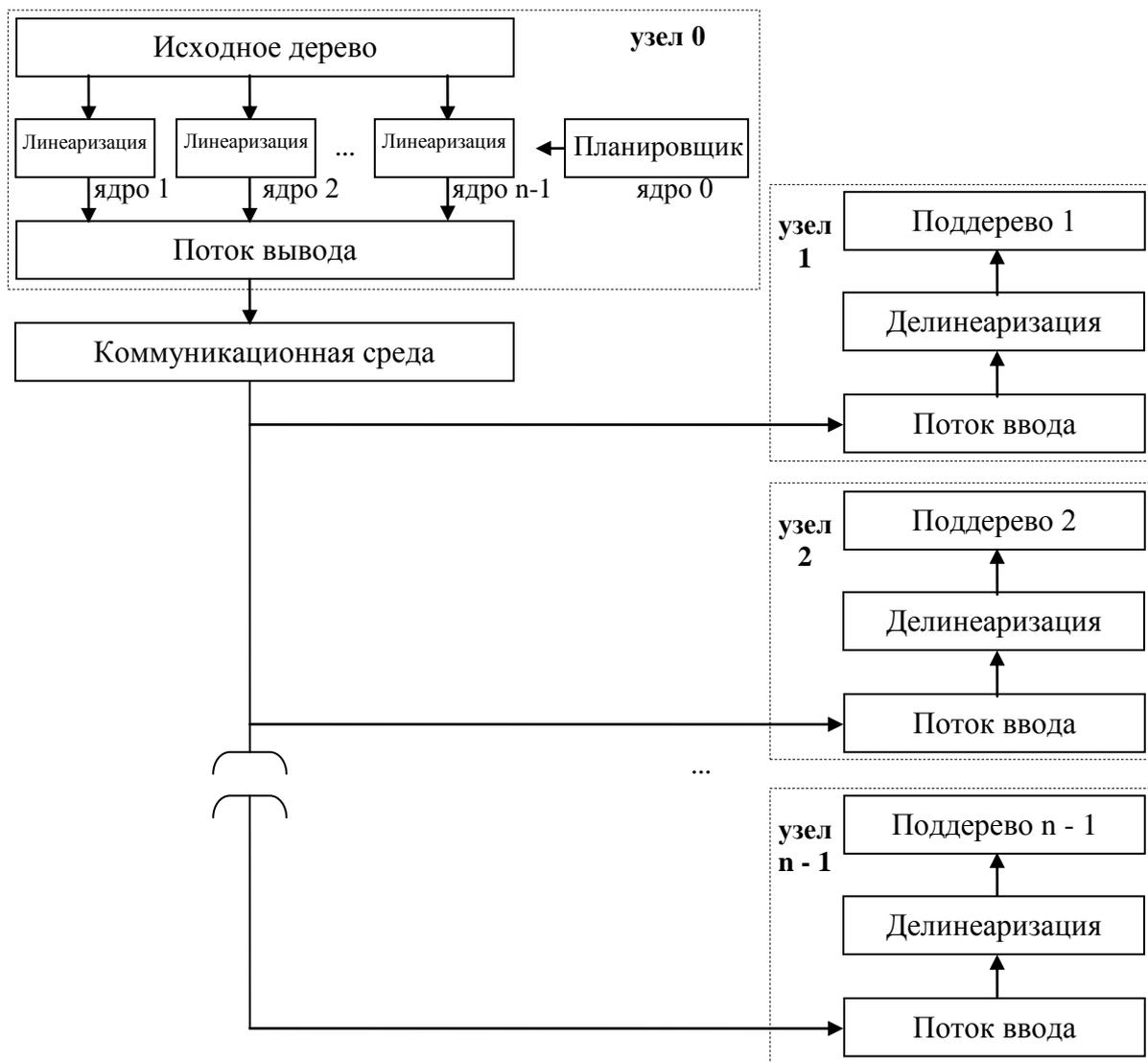


Рис. 1. Функциональная схема процесса параллельной линейаризации

Во время исследований приоритет потоков программно выставлялся в максимальный - TIME\_CRITICAL. Все посторонние приложения были предварительно выгружены из памяти.

При проведении экспериментов по определению времени выполнения операций, использовалась функция clock, определённая в заголовочном файле time.h.

Исследования проводились на ЭВМ с двухядерным процессором Intel Core 2 Duo.

Основные характеристики ЭВМ:

- тактовая частота процессора – 2,2 ГГц;
- объём оперативной памяти – 1 Гб.

Исходными данными для исследования являлось дерево с количеством узлов равным 174762. Проведено 10 экспериментов и определено среднее время выполнения операции линейаризации. Результаты исследования представлены в таблице 1.

Таблица 1

Результаты экспериментов по определению времени выполнения линеаризации

№	Время работы в параллельном режиме (два потока), с	Время работы в последовательном режиме, с
1.	0.297	0.500
2.	0.282	0.484
3.	0.282	0.500
4.	0.297	0.500
5.	0.281	0.484
6.	0.281	0.500
7.	0.282	0.500
8.	0.297	0.485
9.	0.297	0.485
10.	0.297	0.500
	$2,893/10 = 0,2893$	$4,938/10 = 0,4938$

Прирост производительности при использовании двух потоков по сравнению с одним потоком составил:

$$\frac{0.2893 * 100}{0.4938} = 58.59\%$$

### Литература

1. Пушкаренко С.А., Дацун Н.Н. *Сравнительная оценка способов реализации распределенного дерева вывода в дедуктивных базах данных/ Научные труды Донецкого национального технического университета. Серия "Информатика, кибернетика и вычислительная техника" (ИКВТ-2008). Выпуск 9. - Донецк: ДонНТУ. - 2008. – 316 с.*
2. Пушкаренко С.А., Дацун Н.Н. *Проблемы аппаратно-программной реализации дедуктивных баз данных в многопроцессорных системах с распределённой памятью/ "Информатика и компьютерные технологии 2007". Материалы третьей научно-технической конференции молодых учёных и студентов. - Донецк: ДонНТУ. - 2007. – 580 с.*