УДК 510.53

УВЕЛИЧЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ ТРАНСЛЯТОРА ЯЗЫКА ПРОЛОГ С ИСПОЛЬЗОВАНИЕМ МЕХАНИЗМА ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ

Шуликов А.В., Дацун Н.Н.
Донецкий национальный технический университет

В 70-е годы логическое программирование и базы данных развивались одновременно. Пролог - самый популярный язык Программирования в ЛОГике – появился как результат упрощения более общих методов доказательства теорем для достижения возможности эффективного программирования. Аналогично реляционная модель данных возникла в результате упрощения сложных иерархический и сетевой моделей для обеспечения возможности непроцедурного манипулирования множественными данными. Па протяжении 70-х и в начале 80-х годов Пролог и реляционные базы данных получили широкое распространение не только и академической или научной среде, но и в деловом мире [1].

Одним из важнейших вопросов при построении вычислительных систем является увеличение их быстродействия. Но достигнуть большего быстродействия возможно лишь двумя методами: экстенсивным (лучшие архитектуры, процессоры, большая тактовая частота) или интенсивным (оптимизация, разгрузка кода, распараллеливание). Постоянно обновлять и покупать новое дорогое оборудование менее рентабельно, чем приобрести программное обеспечение, которое работает эффективнее и быстрее на любом имеющемся оборудовании. Распараллеливание в процессе трансляции даёт преимущество в работе программы, в быстродействии выполняемого кода.

На данный момент развитие аппаратной части ЭВМ достигло трудно преодолеваемого барьера. Как показывает практика - получение значительно более мощных систем, относительно существующих, в ближайшей перспективе маловероятно. Да и всегда существуют задачи, для которых недостаточно производительности одной системы. Поэтому актуально увеличение быстродействия и продуктивности за счёт организации и расширения кластеров, сложных многопроцессорных систем, поддержка которых программными продуктами достаточно ограничена.

Программа на языке Пролог состоит из набора фактов, определенных отношений между объектами данных (фактами) и набором правил (образцами отношений между объектами базы данных). Для работы программы пользователь должен ввести запрос - набор термов, которые все должны быть истинны. Факты и правила из базы данных используются для определения того, какие подстановки для переменных в запросе (называемые унификацией) согласуются с информацией в базе данных [2].

Исходя из структуры программы на языке Пролог, вытекает проблема интеграции различных парадигм программирования. Логической парадигмы с одной стороны, и, например, функциональной - с другой. Решение задачи такой интеграции, а значит и сам процесс распараллеливания Пролог-программы, может быть построен по одному из трех следующих принципов, выделенных исходя из степени прозрачности этого процесса в отношении текста программы [1]:

1. принцип **полной прозрачности** - в данном случае процесс распараллеливания происходит внутри транслятора за счет особого устройства внутреннего представления данных и процесса получения результата и происходит без участия пользователя (программиста);
2. принцип **промежуточной прозрачности** - пользователь должен каким-то образом декларировать предикаты распараллеливания, после чего можно работать с ними, как и с любыми другими предикатами. В этом случае остается скрытым факт, что во время сопоставления с предикатом используется механизм распараллеливания;
3. принцип **отсутствии прозрачности** - программист в программе на Прологе должен явно формулировать вставки на другом языке программирования (например, на языке Си с использованием библиотеки MPI).

Использование любого из данных принципов требует модификации транслятора языка. В случае полной прозрачности необходима модификация внутренних структур данных и алгоритмов трансляции. Промежуточная прозрачность требует расширения диалекта языка и введения дополнительных предикатов, управляющих параллельными вычислениями. При отсутствии прозрачности требуется доработка интерфейса для интеграции транслятора с модулями на других языках программирования, которые

будут явно реализовать параллелизм (например, вставки кода на языке с использованием MPI).

Наиболее оптимальной с точки зрения пользователя системы является реализация механизма полной прозрачности. Это позволяет пользователю писать код программы, не заботясь об изучении дополнительных расширений или сторонних языков. Особенно актуальна эта проблема в связи с появлением большого количества пользователей, не обладающих обширными профессиональными навыками программировании.

В итоге при написании транслятора языка логического программирования по принципу полной прозрачности можно выделить несколько возможных типов параллелизма:

- «И»-параллелизм имеет место, когда подцели, принадлежащие данному запросу, выполняются параллельно различными вычислительными модулями;

- «ИЛИ»-параллелизм имеет место, когда одна подцель может быть унифицирована с заглавиями нескольких клауз, и тела этих клауз вычисляются различными вычислительными модулями;

- Мультитрединг - получение и обработка одновременно «многих» потоков команд и данных: при этом устройство управления распределяет потоки для параллельного выполнения на внутренних вычислительных устройствах процессора.

Среди существующих реализаций, параллелизм типа «И» реализован в таких трансляторах, как Parlog (разработчик Parallel Logic Programming Ltd. [3]), параллелизм типа «ИЛИ» в трансляторах «Акторный пролог» (разработчик Морозов А.А. [4]) и в YAP Prolog (разработчик LIACC/Universidade do Porto [5]). Кроме того, существуют и другие реализации Пролога, как с поддержкой распараллеливания, так и без нее.

Литература

1.Чери С, Готлоб Г., Танка Л., Логическое программирование и базы данных – М.: Мир, 1992.

2.URL: http://schools.keldysh.ru/sch444/MUSEUM/LANR/Prolog.htm – ресурс "Школьные страницы"

3.URL: http://www.parlog.com - официальный сайт "Parallel Logic Programming Ltd."

4.URL: http://www.cplire.ru/Lab144/125l/09010000.html – официальный сайт Института радиотехники и электроники РАН.

5.URL: http://www.ncc.up.pt/~vsc/Yap – официальный сайт "Computer Science Group / LIACC"