

УДК 004.258

ОБЗОР АЛГОРИТМОВ ПРЕДСКАЗАНИЯ ПЕРЕХОДОВ И АНАЛИЗ ИХ ЭФФЕКТИВНОСТИ В КОНВЕЙЕРНЫХ ПРОЦЕССОРНЫХ СИСТЕМАХ

Лазник Н.В., Фельдман Л.П.

*ГВУЗ «Донецкий национальный технический университет», Украина
kolua-17@yandex.ru*

В докладе рассмотрены существующие алгоритмы предсказания переходов в процессорах, а также способы анализа эффективности данных алгоритмов.

Основным фактором, ограничивающим быстродействие конвейерных процессоров, служат наличие в программах команд перехода. Это связано с тем, что до завершения их выполнения остается неизвестным, на какую команду передается управление. Другой важной причиной, снижающей быстродействие, служат наличие межкомандных информационных зависимостей.

Функционирование конвейерного процессора при поступлении на него команд перехода может быть реализовано различным образом.

Наиболее простым является такое решение, при котором поступление команды перехода на процессор запрещает (блокирует) поступление остальных команд до завершения ее выполнения. Однако в этом случае при поступлении команды перехода происходит удаление с процессора всех последующих за ней команд.

При другом решении для ослабления отрицательного влияния команд перехода используют предугадывание направления перехода. Пусть на сегменте уже известно, что переход может произойти либо на команду y' , либо на команду y'' . Тогда на процессор вводятся либо команда y' и команды, следующие за командой y' , либо команда y'' и команды, следующие за командой y'' . Предположим, что через несколько тактов выясняется, что переход должен быть произведен на команду y'' . Если на процессор были введены команда y' и следующие за ней команды, то они с процессора удаляются как ошибочно введенные и на процессор начинают поступать команда y'' и следующие за ней команды. Если же направление перехода было предугадано верно, т.е. на процессор вслед-за командой перехода вводились команда y'' и следующие за ней команды, то непрерывность процесса обработки не нарушается.

Алгоритмы предсказания переходов

Статическое предсказание. Статическое предсказание является простейшей техникой предсказания ветвлений, поскольку она не полагается на информацию о динамической истории выполнения кода. Вместо этого он предсказывает результат основанный исключительно на ветке инструкции [2].

В начале реализации SPARC и MIPS (две из первых коммерческих архитектуры RISC), использовали однонаправленное статическое предсказание переходов: они всегда предсказывали, что условного перехода не будет, поэтому они всегда выбирали следующую последовательную инструкцию. Когда инструкция или переход найдена

указатель перемещается на не последовательный адрес.

Оба процессора вычисляют ветви в стадии декодирования и имеют один цикл выборки. В результате повторения целевой объект ветви двух циклов долго, и машина всегда будет получать инструкцию сразу после любой ветви. Обе архитектуры определяют «слоты задержки инструкций» для использования полученных инструкций.

Более сложная форма статического предсказания предполагает, что обратные ветви будут приняты, и передние ветви не будут приниматься. Обратные ветви – это те которые имеют целевой адрес ниже чем свой собственный адрес. Эта техника может помочь с точность предсказания циклов, которые, как правило, выполняются чаще в обратном направлении.

Некоторые процессоры позволяют встраивать предсказания ветвлений в код, указывая где должно быть вызвано статическое предсказание. Intel Pentium 4 принимает указатели предсказания ветвлений, но не используется в более поздних процессорах. Статическое предсказание используется в качестве запасного метода в некоторых процессорах с динамическим предсказанием ветвлений, когда нет никакой информации для динамических предсказателей.

Предсказание следующей линии. Некоторые из суперскалярных процессоров (MIPS R8000, Alpha 21264 и Альфа +21464 (EV8)) выбирают из каждую линию инструкций с указателем на следующую линию. Следующий предсказатель линии обрабатывает предсказание цели ветви, а также направлении ветви

Когда следующая строка указывает на соответствие группам из 2, 4 или 8 инструкций, целевая ветвь как, не обрабатывается первой и поэтому начальные инструкции обрабатывают вхолостую. Если предположить равномерное распределение инструкций в ветвях, то 0,5, 1,5 и 3,5 инструкции отбрасываются, соответственно.

Поскольку отрасль сама правило, не последняя команда в соответствии группе, инструкции после того, как принято филиал (или ее задержку слота) будут отменены. Снова в предположении равномерного распределения инструкции ветвления размещения, 0,5, 1,5 и 3,5 инструкции будут отброшены

Отброшенные инструкции в целевой ветви дополняются до практически полного цикла выборки, даже для одного цикла предсказателя следующей линии.

Насыщенный счетчик или бимодальный предсказатель – это структурная схема с четырьмя состояниями: сильно не используемая, слабо не используемая, слабо используемая, сильно используемая

Когда ветвь вычисляется, соответствующее состояние обновляется. Ветви что уменьшают состояние переходят в состояние сильно не используемых, а ветви увеличивающие счетчик состояния переходят в состояние сильно используемых. Преимущество такого двойного счетчика в подобной однобитной схеме в том что условный переход должен дважды отклоняться перед изменением. Для примера case выбор будет ошибочным один раз, с большей вероятностью.

Таблицы предсказателей проиндексированы с битами адресов инструкции, так что процессор может получить прогноз для каждой команды перед командой декодирования. Эффективность данного метода достигает 93,5 % в случае соответствия каждой ветви уникального указателя.

Двухуровневый адаптивный предсказатель. Условные переходы, принимаемые

каждую секунду времени или другие регулярно повторяющиеся действия плохо прогнозируются счетчиком насыщения. Двухуровневый адаптивный предсказатель помнит историю последних n вхождений ветви и использует один счетчик насыщения для каждого из возможных 2^n моделей состояний. Рассмотрим на примере $n = 2$. Это означает, что последние два вхождения ветви хранятся в 2-битный регистр сдвига. Этот регистр может иметь 4 различных значения двоичных: 00, 01, 10, 11, где 0 - «не принято», 1 - «принимать». Теперь мы создадим таблицу с состояний из четырех записей, по одному для каждого из $2^n = 4$ возможных состояний. Каждая запись в таблице 2-битный счетчик. Регистр ветви используется для выбора, какой из четырех счетчиков насыщения использовать. Если история 00, то первый счетчик. Если 11, то последний.

Предположим, что условный переход берется каждый третий раз. Последовательность ветви 001001001... В этом случае, запись номер 00 в таблице состояние «определенно используется», указывая что после 00 приходит 1. Последовательность 01 пойдет в состояние «определенно не используется», указывая что после 01 идет 0. То же самое и в случае с вводом 10, в то время как 11 не используется, потому что никогда не бывает двух последовательных переходов.

Общее правило для двухуровневого адаптивного предсказателя с n -разрядной таблицей является то, что он может предсказывать любую повторяющуюся последовательность в любой срок, если все n -разрядные последовательности разные.

Преимущество данного типа предсказателя в возможности предсказывать произвольно повторяющуюся последовательность переходов. Данный метод прогнозирования используются в большинстве современных микропроцессоров.

Локальный предсказатель переходов. Имеет отдельный буфер истории для каждой команды условного перехода. Может использоваться для двухуровневого адаптивного предсказателя. Буфер истории отделен для каждой инструкции условного перехода, в то время как шаблон истории может быть отделен или открыт для всех команд условного перехода. Эффективность данного алгоритма 97,1 %.

Глобальный предсказатель переходов. Не хранит отдельно историю для каждого условного перехода. Вместо этого он хранит историю всех условных переходов. Преимущество данной истории в том что любая корреляция между различными условными переходами является частью процесса предсказания. Недостаток в том что история содержит не актуальную информацию. Данная схема лучше чем насыщенный счетчик только для больших таблиц переходов и редко используется. Эффективность составляет 96,6 %.

Предсказатель согласия. Является двухуровневым адаптивным предсказателем с глобальным буфером истории шаблоном таблицы истории, содержит дополнительный локальный счетчик насыщения. Выходы из локальных и глобальных предсказателей происходят по функции XOR. Целью является уменьшение числа ситуаций когда ветви с противоположными прогнозами одновременно находятся в таблице предсказаний.

Предсказатель циклов. Условный переход который контролируется специальным предсказателем переходов. Условный переход в низ стека команд повторяемого N раз будет осуществлен $N-1$ раз. Условный переход выполняемый несколько раз определяется

как циклический. Подобный условный переход легко реализуется простым счетчиком. Является частью большинства микропроцессоров.

Также можно отметить предсказатели не прямых переходов, предсказатель возвращаемых данных функции, переопределенный предсказатель ветвлений и нейронный предсказатель ветвлений.

Анализируя данные алгоритм статического предсказания можно получить, что среднее время прохождения программы через процессор определяется как

$$t(y_{out}^n) = (N + n + \sum_{i=0}^{N-1} (N-i)l_i)T = n(\frac{N}{n} + 1 + \sum_{i=0}^{N-1} (N-i)\frac{l_i}{n})T, \quad (1)$$

где n – число команд выходного потока. Обозначив $\sum_{i=0}^{N-1} (N-i)\frac{l_i}{n}$ через p' и предположив, что $n \gg N$, быстродействие $perf$ конвейера будет равняться

$$perf = n/t(y_{out}^n) \approx \left\{ 1 / \left[1 + p' \left(N - \frac{\sum_{i=0}^{N-1} ip'_i}{p} \right) \right] \right\} \frac{1}{T} \quad (2)$$

При оценке быстродействия на основе марковских моделей можно получить

$$perf = \frac{1 + \sum_{i=1}^{N-1} ip_i}{1 + N \sum_{i=1}^{N-1} ip_i} \frac{1}{T} \quad (3)$$

Выводы

Больше количество алгоритмов предсказания переходов и вариация их комбинирования открывает широкие возможности для анализа их эффективности на основе математического моделирования и марковских моделей. Что весьма эффективно для поиска наиболее оптимальных комбинаций алгоритмов предугадывания.

Перечень использованных источников

- [1] Авен О.И. Оценка качества и оптимизация вычислительных систем / О.И. Авен, Я.А. Коган / Москва «Наука». – 1982. – № 4. – С. 165-193.
- [2] Фельдман Л.П. Математическое обеспечение САПР/Л.П.Фельдман, В.А.Дедищев/ Киев УМК ВО. – 1992 – С. 23-38.
- [3] P. Shen. Modern processor design: fundamentals of superscalar processors./ P. Shen, John; Lipasti, Mikko / Boston: McGraw-Hill Higher Education. – 2005 – p. 455
- [4] Yeh. “Two-Level Adaptive Training Branch Prediction”. Proceedings of the 24th annual international symposium on Microarchitecture./ Yeh, T.-Y.; Patt, Y. N/ Albuquerque, New Mexico, Puerto Rico: ACM - 1991). pp. 51-61.