

# 1 Варианты организации обновления таблиц

Можно рассмотреть несколько вариантов организации обновления таблиц: использование теневой памяти, использование специальных (управляющих) пакетов, инкрементальное обновление.

## 1.1 Теневая память

Для обновления таблиц классификации можно использовать теневую память, содержащую копию основной таблицы на конкретной стадии конвейера (см. рис. 1). В данном варианте плоскость управления (control plane), при необходимости провести операцию обновления таблицы на конкретной стадии, выполняет обновление не в основной памяти таблицы, а в памяти копии. После чего, вычислительный элемент на данной стадии все еще продолжает использовать таблицу из основной памяти пока не закончит обработку текущего заголовка пакета. После обработки текущего заголовка, данная стадия будет использовать обновленную таблицу из теневой памяти. Таким образом основная и теневая память поменяются местами.

Переключение между основной и теневой памятью можно выполнить, например, с помощью некоторого регистра, который будет указывать вычислительному элементу какую память использовать в данный момент.

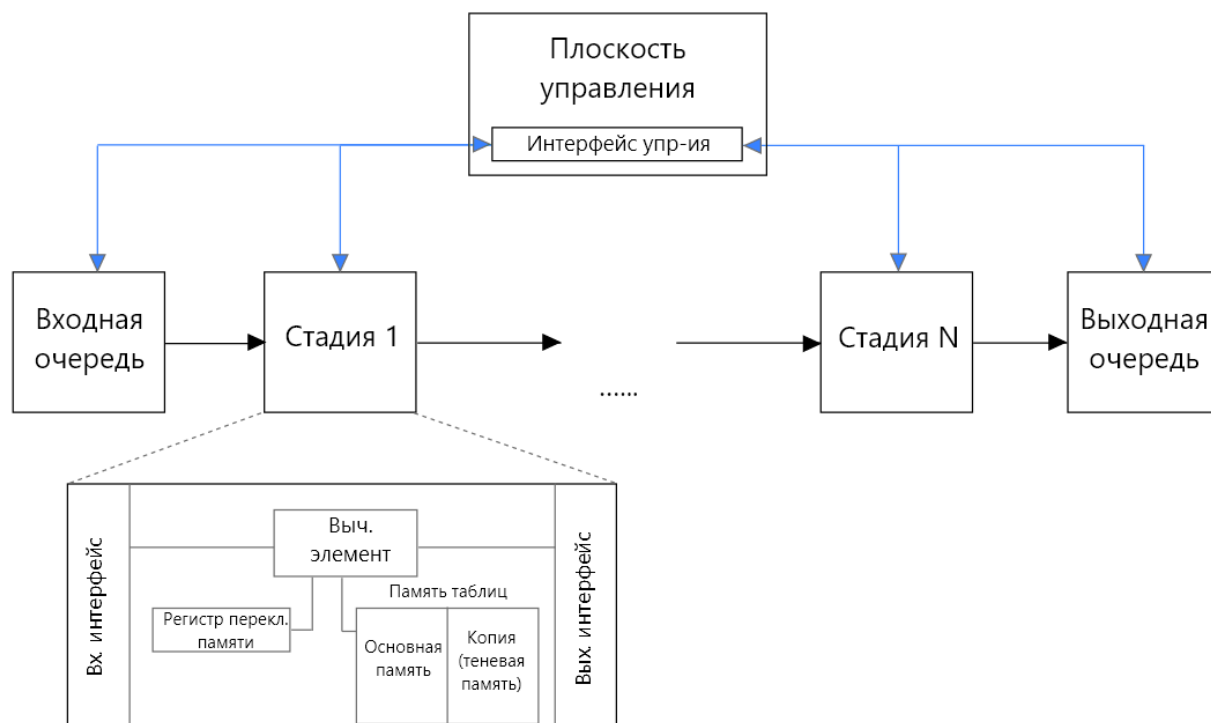


Рисунок 1 – Использование теневой памяти для обновления таблиц

Преимущества:

- 1) Нет задержек, связанных с остановкой конвейера для обновления таблиц.

Недостатки:

- 1) Данный вариант не актуален для TCAM.
- 2) Требуется в два раза больше памяти для хранения копии таблицы.

## **1.2 Использование управляющих пакетов**

При данном варианте [1], control plane посылает на конвейер пакеты специального типа, которые будут обновлять необходимые таблицы, последовательно проходя по всем стадиям конвейера. Имеется два варианта использования управляющих пакетов:

- 1) Данные пакеты несут в себе информацию о том, в каких таблицах и на каких стадиях необходимо выполнить обновление, и какие записи нужно добавить, удалить либо изменить. Например, управляющий пакет может включать четверку: <стадия, ключ, значение, операция (добавить/удалить/изменить)>. При таком варианте стадия сама обновляет таблицу. Возможна отправка нескольких специальных пакетов, если в одном пакете не помещается вся необходимая информация.
- 2) Пакеты специального типа содержат только информацию о том, на каких стадиях необходимо сообщить плоскости управления о готовности стадии к обновлению. Данные пакеты проходят по каждой стадии, и при достижении стадии, в которой необходимо выполнить обновление таблицы, сигнализируют об этом плоскости управления. После чего стадия приостанавливает свое выполнение и происходит обновление таблицы.

Преимущества:

- 1) Нет лишних затрат по памяти.
- 2) Нет существенных задержек, связанных с остановкой выполнения конвейера (при небольшом количестве обновлений таблиц).
- 3) Данный вариант может быть использован для SRAM и TCAM.

Недостатки:

- 1) При слишком большом количестве обновлений таблиц необходимо отправить достаточно много управляющих пакетов на конвейер, вследствие чего снижается скорость обработки.

- 2) Для поддержки данного варианта необходимо внести соответствующие дополнения в программный код стадий конвейера.

## 1.2 Инкрементальное обновление

Данный вариант организации обновления таблиц предназначен для TCAM (Ternary Content-Addressable Memory).

В TCAM хранятся записи (например, IP префиксы), а возвращаемые значения хранятся в отдельной (например, SRAM) памяти. Записи в TCAM располагаются в порядке убывания приоритетов. Например, если это таблица маршрутизации, то в порядке убывания длин префиксов. Результатом поиска является самая верхняя (с наивысшим приоритетом) найденная запись.

Алгоритм инкрементального обновления таблиц [2, 3, 4] выполняется без остановки конвейера (поиска по таблицам), тем самым не снижая скорость обработки пакетов.

Основная идея инкрементального обновления состоит в том, что каждая запись имеет так называемый бит валидности. Если бит валидности записи равен 0, то данная запись не участвует в процессе поиска. Таким образом при поиске учитываются только записи, у которых бит валидности равен 1. При выполнении операции обновления, у обновляемых записей бит валидности устанавливается равным 0, после чего происходит обновление записи и бит валидности устанавливается равным 1. Во время обновления может происходить перемещение некоторых записей в таблице, чтобы вставить новую запись (префикс) в надлежащее место. Для решения данной проблемы имеются некоторые способы распределения свободных записей по таблице TCAM [2]. Свободные записи распределяются таким образом, чтобы минимизировать кол-во перемещений при обновлении. Например, для каждого блока префиксов (префиксов одинаковой длины) можно выделить некоторое кол-во свободных записей (пропорционально кол-ву префиксов в блоке) и располагать свободные записи по краям блока (сверху и снизу блока).

Преимущества:

- 1) Нет задержек, связанных с остановкой конвейера для обновления таблиц.

Недостатки:

- 1) Применим только для TCAM.
- 2) TCAM должна быть двухпортовой – одновременный доступ к записям со стороны Control plane и Data plane.

## **Литература**

1. Basu A., Narlikar G. Fast incremental updates for pipelined forwarding engines //IEEE/ACM Transactions on Networking. 2005. T. 13. N. 3. P. 690-703.
2. Mishra T., Sahni S. Duos-simple dual tcam architecture for routing tables with incremental update //The IEEE symposium on Computers and Communications. IEEE, 2010. P. 503-508.
3. Banerjee T., Sahni S., Seetharaman G. PC-DUOS+: A TCAM architecture for packet classifiers //IEEE Transactions on Computers. – 2012. – T. 63. – №. 6. – C. 1527-1540.
4. Banerjee T., Sahni S., Seetharaman G. PC-TRIO: A power efficient TCAM architecture for packet classifiers //IEEE Transactions on Computers. 2014. T. 64. N. 4. P. 1104-1118.