

УДК 004.318

Недбайло Ю.А.

Институт электронных управляющих машин

ЗАО «МЦСТ»

Особенности реализации поддержки когерентности кэшей в системах на кристалле «Эльбрус-S» и «Эльбрус-2С»

В системе «Эльбрус-S», предназначенной для использования в многопроцессорных NUMA-системах, реализован механизм поддержки когерентности кэшей по протоколу MOESI, используемому также в системах AMD [1]. У этого механизма есть несколько особенностей.

Первой особенностью является проблема перехода в состояние Exclusive в результате операции считывания. В процессорах серии «Эльбрус» может возникать ситуация, при которой в память выходят два запроса на считывание по одному адресу, после которых также была запись по этому адресу, оставшаяся в буфере кэша, и в этом случае получение первым запросом данных со статусом Exclusive может привести к нарушению консистентности. Основных решений этой проблемы известно два: более строгое упорядочивание обращений в память и отказ от перехода в состояние Exclusive в результате запроса на считывание. Каждое из этих решений влечёт за собой потерю производительности на эталонных тестах порядка 1%. Автором доклада предложен механизм, позволяющий при небольших затратах оборудования обнаруживать проблемную ситуацию и отменять установку статуса Exclusive только в ней, что должно решить проблему без заметной потери производительности.

В устройстве, называемом MAU, реализован механизм подклеивания заявок [2], используя который можно обнаруживать проблемную ситуацию как положительный результат сравнения адреса поступившей заявки с адресами остальных и отсутствие подклеивания. В этом случае MAU должен снабдить ответ в кэш на первый запрос признаком Shared, запрещающим установку состояния Exclusive. Помимо этого, кэш должен сам выработать такой же

признак, если адрес выдаваемого из MAU в кэш ответа с данными совпадает (не обязательно полностью а, например, тэг) с адресом запроса, уже промахнувшегося в кэш но ещё не выданного в MAU.

Вторая особенность – разрешение ситуации одновременных записей в кэш-строку, разделяемую разными процессорами в состоянии Shared, известной как конфликт Invalidate-Invalidate. Обычные решения этой проблемы подразумевают передачу данных в ответ на запрос Invalidate даже в случае, когда эти данные уже есть у запросчика, при Shared и Owned состояниях строки у запросчика и отвечающего. В системе «Эльбрус-S» применён оригинальный подход, позволяющий этого избежать и сэкономить таким образом до 50% передач данных на последовательностях чтений и модификаций одной кэш-строки разными процессорами.

Проблема заключается в том, что в интервале между выдачей запроса Invalidate и началом его выполнения, кэш запросчика может получить Invalidate от другого процессора. Таким образом, при отсутствии полного справочника и специальных средств, за пределами кэша запросчика неизвестно, есть ли у него актуальная копия данных, вследствие чего обладатель состояния Owned вынужден на Invalidate всегда отвечать данными.

В системе «Эльбрус-S» введён дополнительный тип ответа на запрос когерентности, названный «merge» и вырабатываемый в случае, если отвечающий кэш уже выдал запрос Invalidate. Получение этого ответа сохраняется как дополнительный признак строки кэша, благодаря чему кэшу всегда известно, есть ли в системе процессоры, которым на Invalidate следует ответить данными.

В системе «Эльбрус-2С» используется похожий протокол когерентности и вспомогательный фильтр запросов когерентности на основе неполного справочника. Благодаря этому, с одной стороны, состояние Exclusive в системе не нужно, с другой стороны, конфликт Invalidate-Invalidate требует разрешения другим способом. Автором разработан механизм, аналогичный используемому в «Эльбрус-S» но приспособленный к наличию фильтра на основе неполного справочника.

Неприменимость механизма, используемого в «Эльбрус-S», обусловлена тем, что, благодаря фильтру запросов на основе справочника, процессор не

получает запросы когерентности, если в результате выполнения предыдущих соотв. строка в нём уже имеет состояние Invalid. В то же время, поскольку справочник неполный, хранящий общее состояние строки в процессоре но не в отдельных его ядрах, по нему в общем случае нельзя определить состояние в ядре-запросчике. Таким образом теряется возможность отслеживать наличие кэшей, имеющих состояние Invalid но уже выдавших Invalidate.

Предложенный автором вариант решения проблемы заключается в использовании ответа «merge», аналогичного используемому в «Эльбрус-S» но хранении его не кэшем а справочником. Наличие этого признака должно запретить перевод состояний процессоров в Invalid, для того чтобы проблемные кэши продолжали получать запросы когерентности до тех пор пока их запросы Invalidate не выполнятся, не получив также «merge». Экономии трафика при сохранении корректности работы можно получить, если превращать запросы Invalidate в запросы когерентности Read-Invalidate т.и т.т., когда последний Invalidate или Read-Invalidate этой кэш-строки вернул признак «merge»; на запросы когерентности Invalidate кэш отвечать данными не должен.

Отдельным вопросом остаётся выбор: хранить признак «merge» в самом справочнике, что увеличит его объём (на 1 бит для каждой кэш-строки), либо хранить только в кэше справочника и каким-либо образом избежать взаимных блокировок и ошибок в случае переполнения кэша строками с таким признаком. Простым, надёжным и не требующим увеличения объёма справочника вариантом выглядит переключение схемы в такой режим работы, когда каждая загружаемая в кэш справочника строка получает признак «merge», в том редком случае, когда пришлось вытеснить из кэша справочника строку с таким признаком. Даже в этом режиме лишние передачи данных должны происходить сравнительно редко.

Литература

1. *Advanced Micro Devices, Inc. AMD64 Architecture Programmer's Manual Vol 2 'System Programming'*. — AMD, 2010. — P. 167-169.
2. *Недбайло Ю.А., Шерстнёв А.Е.* Оптимизация доступа к памяти в вычислительном комплексе «Эльбрус-3S» // Информационные технологии. — 2008. — № 11.