

**М.В. Руцов**

Эксперт, к.т.н.

Итак, остановились мы на вопросе совместимости. А что сие собственно подразумевает? Ну, самое простое, что приходит в голову – взять ехе-шник, да запустить на новой платформе. Правильно думаете – ничего не заработает! А почему? Да потому, что, например, PXA270 или PXA320 – вроде Intel, а совместимости с архитектурой x86 нет, поскольку внутри этих двух чипов сидит ядро ARM. Тогда на помощь приходит компилятор, благодаря которому наработанный софт ложится на другую платформу, причём не только от Intel. И всё замечательно – с точки зрения оперативной реализации нового продукта и выпуска его на рынок. Однако, как это отражается на быстродействии самих алгоритмов? Статистика показывает – практически большинство решений в области охранного видеонаблюдения (в плане цифровой обработки изображений) реализовано на базе архитектуры x86. Причём с использованием технологий MMX, SSE, SSE2 и SSE3. Самое интересное, именно такими операциями в оптимизированных алгоритмах процессор загружается почти под 99%. Вот она где – совместимость сидит! Так вот, в упомянутых PXA270 и PXA320 как раз такой "аппаратный довесок" и имеет место быть – WireLess MMX называется. Уж и вторая версия вышла. Попробую показать на конкретном примере, что получается, если просто "писать" алгоритмы на языке высокого уровня, с последующей тупой компиляцией.

Решила как-то одна компания наши алгоритмы видеодетекции к себе интегрировать. Начали торговаться – не срослось. Тогда они сказали: "Сами сделаем!". И кое-чего наваяли. А для "возбуждения" своих разработчиков руководство решило подпольное сравнительное тестирование провести. Взяли нашу 4-чиповую плату, да засунули в системник с двухядерным процессором и врубили по четырём каналам все наши детекторы – Детектор Оставленных/Унесённых Предметов и ещё три MotionDetector для разных полос пространственных частот. Всего 16 получилось. Ну и заработало сие хозяйство в реальном времени – по 25 fps! Потом свой Детектор Оставленных/Унесённых Предметов запустили. Уж и не знаю, как он функционально отработал (меня там не было), но тормозил

Народный НаноВидеоСервер II

Статья печатается в авторской редакции

страшно – всего 5 fps и то лишь по одному каналу. Вот и спрашивается, а кому он такой нужен, разве что для PR-мероприятий. Выигрыш нашего детектора по скорости составил – и не проценты, и не разы, а десятки! Естественно процесс "возбуждения" пошёл на ура.

А теперь проанализируем, как сие стало возможным. Лет 15 назад довелось мне самому видеодетекторы сочинять. Под рукой лишь 286-я машина, но есть такое слово – "надо". Получилось, хоть и окошко с живым изображением было урезано до "немогу", система спокойно "ловила" летающие объекты – мои тапки, поскольку они всегда со мной. В компьютерах того времени особых аппаратных наворотов не было, поэтому пришлось максимально оптимизировать сам алгоритм и учиться у матушки-природы. Лет пять для этого нейрофизиологию мозга изучал. И кое-что понял. Мысли по этому поводу в популярном виде изложены в статье "Видеодетекторы – взгляд изнутри. Грани интеллекта" ("Системы безопасности", №№5–6, 2003 г.), в главе "Парад тупых алгоритмов". Вот в частности:

В зрительной системе HomoSapiens работают "тупые" (в хорошем смысле этого слова) алгоритмы видеодетекции, без обилия всевозможных операторов "if-goto" и рекурсий – изображения буквально продавливаются сквозь нейронные слои, как варёный картофель сквозь сито. Фантастическая мощь – и никакого "интеллекта".

Таким образом, удалось реализовать алгоритмы видеодетекции с параллельной обработкой изображений, что в дальнейшем – с появлением технологии MMX, позволило в разы поднять скорость. Но и это ещё было не всё. Как известно из нейрофизиологии, динамический диапазон нейронов узок, ну нет там, в голове никакого Преобразования Фурье с плавающими запятыми. Однако всё прекрасно работает. Пришлось покумекать, как следствие в разработанных алгоритмах промежуточные вычисления не вылезают за пределы одного байта. А затем начался парад SIMD (Single Instruction Multiple Data) технологий, да повышение общей производительности процессоров. Современные системы как минимум на три порядка превосходят машины 15-летней давности.

Вот и спрашивается, как можно было так скорость угробить (в плане проведённого тестирования). Да элементарно – руководство, в гаре конкурентной гонки, вызывает программистов

(не математиков, в виду их полного отсутствия – деньги-то все в PR-мероприятия улетели) и ставит задачу на попа: "Быстро сочинить Детектор Оставленных/Унесённых Предметов – за две недели!". Конечно, в отведённый срок никто не укладывается, но и об оптимизации уж говорить не приходится. Это типа – есть у крестьянина плуг да лошадка. И вдруг бац – "Кировец". В конечном итоге тот же плуг впрягается в многолошадиносильный трактор – результат одинаковый. Тянет ведь! С таким подходом у нас ни НаноВидеоСервер, ни камера "умная" не получится – производительности не хватает, всё куда-то съелось.

А теперь зайдём с другой стороны. Современные процессоры взлетели до космических высот – высокие скорости и, что самое важное, глобально изменилась сама архитектура, предлагающая разработчикам много степеней свободы по своему использованию. Кроме того, и материнские платы от многочисленных производителей с разнообразными чипсетамы буквально кружат головы – что выбрать? Ведь теоретически ничего просчитать невозможно. Взять хотя бы cache-память. Как её размер влияет на производительность? Существует лишь один вариант – тестировать, тестировать и ещё раз тестировать. Сказано – сделано.

Ну, как работает cache объяснять не буду. Главное, что скорость обмена процессора с этой памятью на порядок с лишним выше, чем с системной. Например, у Pentium 4 разрядность cache составляет 32 байта, что в 4 раза шире, да и работает она на частоте процессора. Конечно, если выполняется лишь одна операция типа A=B+C (типа межкадровой разности), то на каждое такое сложение потребуется 2 чтение и 1 запись. Процессор будет фактически простаивать – всё упрётся в системную память. Однако есть такие процедуры, например, как пространственная фильтрация, когда на один пиксел



изображения приходится 1 чтение и 1 запись при куче умножений и сложений результатов от соседних пиксел. Или, скажем, требуется провести целый ряд промежуточных вычислений. Если все это хозяйство помещается в cache целиком, то можно получить многократное увеличение скорости. Поэтому размер cache играет очень большое значение. Важно также, на сколько ассоциативных блоков разбита эта конструкция. В Pentium 4 их число равно 8 (8-way set associative). Много это или мало рассчитать теоретически невозможно – всё очень сильно зависит от структуры данных и алгоритмов вычислений. Каждый "промах" мимо cache приводит к "вышибанию" одного из блоков – обычно того, к которому дольше всего не обращались. Чтобы не быть голословным, приведу конкретные результаты тестирования наших алгоритмов видеодетекции.

Итак, у нас в распоряжении были две машины (так исторически получилось):

1. Двойной Хеон – 2 ГГц, L2 cache – 512 К, FSB – 533 МГц

2. Pentium 4 – 3 ГГц, L2 cache – 1 М, FSB – 800 МГц

Как вы думаете – кто победил и с каким счётом? А победил Pentium 4. Причём, если допустить, что один Хеон вообще не работал (никакого специального софта мы не ставили), то следовало ожидать выигрыш в 1.5 раза. Однако счёт оказался разгромным – 4:1! Это заработала мегабайтная cache, размер ассоциативных блоков которой стал соизмерим с размером обрабатываемых изображений.

Мало того, даже такая вещь, как "гулять" по изображению – по столбцам или по строкам, влияет на производительность. Кстати, насчёт cache – в плане последних "достижений" Intel в области многоядерных процессоров. Вот, в частности, интересная статья – "Исследование эффективности совместного исполь-

зования общего и разделенного L2-кэша современных двухъядерных процессоров". Привожу дословно:

Таким образом, объединенный 4-МБ L2-кэш данных процессоров семейства Intel Core 2 в условиях конкуренции за этот ресурс со стороны двух ядер при любом типе обращения (только чтение, только запись или одновременное чтение и запись) способен эффективно кэшировать данные, объем которых не превышает всего 1,25 МБ, т.е. примерно четверть объема L2-кэша (!). В этих условиях (когда на одно из ядер приходится указанный объем данных, а остальная, большая часть данных запрашивается вторым ядром) наблюдается максимальная эффективность утилизации как шины данных L2-кэша одним из ядер, так и шины данных оперативной памяти вторым ядром, в результате чего в этой области наблюдается максимум суммарной пропускной способности. В случае же примерно равного распределения объема данных, не помещающихся в L2-кэш, между ядрами процессора, пропускная способность, приходящаяся на каждое из ядер, оказывается ниже даже пропускной способности оперативной памяти при однопоточном обращении. Таким образом, эффективность кэширования данных в этой «конфликтной» области оказывается минимальной – можно ска-

зать, что кэширование данных и вовсе отсутствует.

Вот так, мало того, что частоты угробили, так ещё и "коммуналку" в cache организовали. Такие вот дела. Вы, конечно, можете мне задать резонный вопрос: "А куда это ты – "Товарищ Сусанин" нас завёл? Вроде не по теме". Да в том-то и дело – лишь показать хотел насколько надо всё "вылизывать". Какие уж тут две недели – и двух лет не хватит! А насчёт Intel это так – для профилактики. Эх, ещё один пример приведу. Один наш инсталлятор взял, да слегка поэкспериментировал с функцией Hyper-Threading. Сначала выключил её в BIOS – каналы тянут по 25 fps. Включил – всё просело до 23–24 fps. И таких степеней свободы много – периферия, системная память, режимы работы cache и многое другое. Именно для целей такого тестирования мы реализовали в нашей системе честное окно статистики, которое показывает реальные скорости: видеодетекции, записи и работы в сети. Аналогичных решений как-то не попадалось – видимо не хочется кое-кому наступать на горло своей рекламной песни.

Итак, подвожу предварительный итог. Вопрос совместимости алгоритмических решений при их переносе на другие платформы архи важен. Особенно с точки зрения достижения максимального быстродействия. Поэтому выбор совместимой платформы позволяет экономить время, ресурсы, да и вообще – реально осуществить задуманное. Именно с этих позиций чип PXA320 представляется достаточно перспективным. В следующий раз рассмотрим альтернативные варианты других "наладонных" процессоров. ■

Ваше мнение и вопросы по статье направляйте на

ss@groteck.ru

ВЕСТНИК ТЕЛЕКОММУНИКАЦИЙ

ВСЕГДА СВЕЖИЕ НОВОСТИ

Редакционная подписка:
 тел.: (495) 609-3231
monitor@groteck.ru
 или в любом почтовом отделении ФГУП "Почта России"
 – по каталогу агентства "Роспечать" – 79773
 – по каталогу российской прессы "Почта России" – 63455