

# Средства разработки коммуникационных и встраиваемых приложений на основе процессоров Intel® с многоядерной архитектурой

Артем ФОКИН  
artem.fokin@intel.com

**В наши дни для удовлетворения разнообразных требований заказчика зачастую недостаточно улучшить лишь один из параметров системы, например тактовую частоту или форм-фактор. Наоборот, заказчику требуются эффективное использование энергии, управляемость системы, более высокая пропускная способность подсистемы ввода-вывода, возможность адресации большего объема виртуальной памяти, гибкость и масштабируемость — и часто на одной платформе.**

Именно поэтому Intel предлагает платформенные технологии, которые помогут представить заказчикам самые мощные и экономичные решения. Платформенный подход Intel позволяет разработчикам добиться множества целей благодаря дополнению многоядерной архитектуры специальными технологиями, расширяющими возможности системы, обеспечивающими масштабируемые вычисления с эффективным энергопотреблением для широкого круга встраиваемых и коммуникационных приложений. Используя интегрированное аппаратное ускорение критически важных платформенных сервисов, Intel освобождает вычислительные циклы для новых функциональных возможностей в приложениях, повышающих их готовность и управляемость и снижающих суммарные эксплуатационные расходы. С помощью этих инновационных интегрированных платформенных технологий и элементов системы, базирующихся на стандартах, можно сконцентрировать ресурсы на создании более широкого спектра функциональных возможностей и поставке заказчикам решений нового поколения при сокращении затрат времени и средств на разработку. Благодаря поддержке широкого круга поставщиков оборудования и ПО, в том числе участников Intel® Communications Alliance, корпорация Intel делает возможным создание решений с высокой степенью оптимизации на всех уровнях интеграции.

## Многоядерные процессоры Intel®

Тактовая частота больше не дает полной картины производительности. Двухъядерные



Рис. 1. Схема вызовов функций — графическое представление исполнения программного кода — помогает быстро установить критически важные функции и последовательности вызовов функций

процессоры Intel® сочетают преимущества двух высокопроизводительных исполнительных ядер, размещенных на одном кристалле кремния (рис. 1), и предоставляют более высокую производительность без повышения количества рассеиваемой тепловой энергии. Многоядерная архитектура способна обеспечить намного более высокие значения производительности и отношения производительность/ватт, чем одноядерные процессоры, даже при более низких тактовых частотах (результаты тестового измерения производительности приведены далее). Такое повышение производительности в расчете на ватт потребляемой энергии позволяет добиться особых преимуществ для приложений, которые должны быть высокоэффективными при ограниченных размерах, и помогает удовлетворить сложные требования разработки коммуникационных блейд-серверов, прило-

жений для управления производственными процессами, терминалов, устанавливаемых в точках обслуживания, и ноутбуках повышенной надежности.

В отличие от традиционных одноядерных многозадачных окружений многоядерные конструкции позволяют назначать выполнение конкретных приложений различным ядрам, повышая тем самым производительность и безопасность. В средах встраиваемых приложений можно выполнять задачи реального времени на специализированном исполнительном ядре, освобожденном от выполнения задач, которые могли бы конкурировать за ресурсы процессора. Например, корпорация TenAsys обнаружила, что, запустив две операционные системы на двух отдельных ядрах, можно избавиться от конкуренции за ресурсы процессора. Это помогло минимизировать дрожание изображения и сделало возможной работу циклических приложений со строгим контролем, в частности используемых в автоматизированном обороте, на максимальном уровне точности (см. историю успеха по адресу: [www.intel.com/design/embedded/casestudies/311275.pdf](http://www.intel.com/design/embedded/casestudies/311275.pdf)).

Даже в среде с одним приложением многоядерные технологии могут повысить производительность, позволив разработчику разделить приложение на потоки параллельно исполняемых инструкций и/или потоки данных. В этом случае выигрыш в производительности зависит от количества вычислений, которые приложение может выполнять параллельно.

Двухъядерные процессоры Intel не только поддерживают новейшие процессорные технологии, но и обеспечивают защиту капиталов-

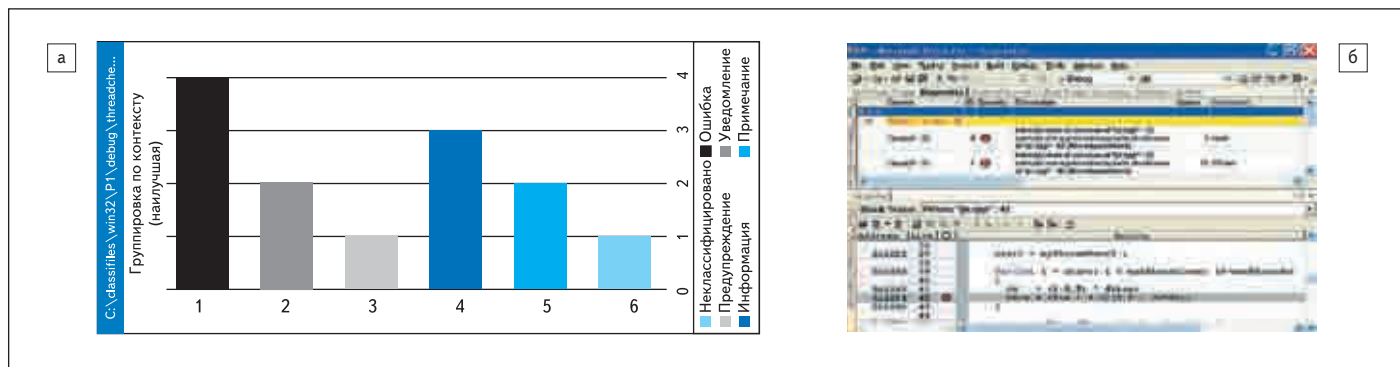


Рис. 2. ПО Intel Thread Checker наблюдает за выполнением приложения и локализует конкурентный доступ к данным, взаимоблокировку потоков и другие ошибки, возникающие при многопоточной обработке информации: а) графический режим; б) режим исходного кода

вложений, оставаясь программно совместимыми с предыдущими 32-разрядными процессорами архитектуры Intel®.

### Использование преимуществ параллелизма

Корпорация Intel создала мощный набор интегрированных инструментальных средств разработки ПО, позволяющий быстро формировать и оптимизировать многопоточные приложения. В состав этих инструментальных средств входят:

- анализатор производительности Intel® VTune, который позволяет оценить характеристики приложения и определить фрагменты кода, оказывающие наибольшее влияние на производительность;
- ПО Intel® Thread Checker (рис. 2) и утилита Intel® Thread Profiler для определения специфических возможностей повышения эффективности программ многокода и производительности приложения;
- компилятор Intel® C++ и библиотеки программ для автоматической организации параллельного выполнения вычислительных потоков приложения.

### Анализатор производительности Intel VTune

Чтобы помочь разработчику определить возможности разделения потоков, Intel предлагает инструментальное средство, которое использует аппаратные прерывания, предоставляя программисту реальную картину работы приложения. Выпущенный в версиях как для ОС Windows, так и для ОС Linux, анализатор производительности Intel VTune позволяет разработчикам сконцентрировать основное внимание на интенсивно работающих фрагментах приложения<sup>1</sup>. Для идентификации возможностей многопоточных вычислений это инструментальное средство предлагает две технологии: метод измерений и схему вызовов функций.

Хотя анализатор производительности Intel VTune предоставляет возможность измерять практически любой рабочий параметр про-

цессора, акцентирование внимания на тактовых интервалах помогает разработчику понять, как много времени тратится на обработку каждой функции в программе. Идентифицировав функции, потребляющие больше всего времени процессора, разработчик может углубленно изучить их исходный код, чтобы определить возможность эффективной реализации их разделения на потоки.

Некоторые функции, интенсивно использующие ресурсы процессора, могут не подходить для параллельного исполнения. Если не удастся выделить в отдельный поток обнаруженную «горячую точку», возможно, удастся улучшить производительность, организовав отдельный поток вычислений для функции более высокого уровня. Благодаря технологии схем вызовов функций анализатора производительности Intel VTune отображает дерево вызовов функций по всему приложению, что позволяет разработчику выделить отдельный поток для вызовов более высокого уровня и тем самым создать несколько потоков для одновременного обращения к самым важным с точки зрения производительности функциям.

Анализатор производительности Intel VTune не только помогает разработчику оценить возможность разделения исходного кода на потоки, но и помогает обнаружить проблемы работы многопоточного кода, вызванные ложным разделением (false sharing). Ложное разделение происходит в том случае, если два потока манипулируют данными, находящимися на одной и той же кэш-линии (cache line). Когда один из потоков изменяет данные на этой кэш-линии, содержимое кэш-памяти становится недостоверным. Из-за этого второй поток должен ждать обновления кэш-памяти, при этом производительность может снижаться. С помощью технологии измерений анализатора производительности Intel VTune разработчик может определить процент промахов кэш-памяти второго уровня (L2 cache misses). Частые промахи кэш-памяти указывают на вероятность возникновения ложного разделения.

### ПО Intel® Thread Checker

ПО Intel® Thread Checker, поставляемое в качестве модуля, который подключается к анализатору производительности Intel VTune, обнаруживает ошибки в многопоточных приложениях, возникающие во время работы программы. Такие ошибки отображаются на экране, при этом выделяются те линии исходного кода, в которых они возникли. ПО Intel® Thread Checker указывает возможные причины появления ошибок при многопоточных вычислениях и подсказывает решение. Указывая на конфликты между общими и частными переменными, ПО Intel® Thread Checker ускоряет разработку многопоточных приложений.

При разделении приложения на потоки возникает возможность конкурентного доступа к данным, когда два или более потоков пытаются одновременно получить доступ к одним и тем же ресурсам. Если потоки взаимодействуют неэффективно, во время выполнения программы это может привести к получению невоспроизводимых результатов.

Например, при конкурентном доступе к данным на чтение/запись один поток может попытаться получить доступ к переменной на запись, в то время как другой — пытается прочитать значение переменной. Результат выполнения операции чтения может быть разным в зависимости от того, была ли произведена запись к этому моменту. В таком случае очень трудно провести отладку, поскольку конкуренция за доступ к данным недетерминирована. Программа может многократно выполнять тесты верно, а при запуске у пользователя — выйти из строя. Чтобы избежать ошибок конкурентного доступа к данным, необходимо обеспечить синхронизацию доступа. Однако синхронизация способна замедлить работу приложения, вот почему следует проводить ее только в случае необходимости. ПО Intel® Thread Checker помогает выяснить, нужна ли синхронизация с помощью поиска ошибок, причем не только имеющихся, но и тех, которые могут возникнуть.

<sup>1</sup> Анализатор производительности Intel® VTune™, [ht tp:// www .intel. com/cd/software/products/asm-na-eng/vtune/index.htm](http://www.intel.com/cd/software/products/asm-na-eng/vtune/index.htm)

Для приложений, разделенных на потоки с использованием технологий OpenMP, Windows API или POSIX, Intel предлагает утилиту Intel® Thread Profiler. Утилита Intel® Thread Profiler (рис. 3) усиливает преимущества инструментальных технологий анализатора производительности Intel VTune, чтобы помочь в настройке многопоточных приложений. Утилита для настройки потоков:

- позволяет проводить визуальный контроль приложения;
- делает возможным более эффективное использование потоков благодаря демонстрации критического пути (critical path) и его продвижения вдоль вычислительных потоков;
- обнаруживает проблемы синхронизации и чрезмерного блокирования, которые приводят к снижению производительности;
- привлекает внимание к несбалансированности рабочей нагрузки;
- позволяет максимально использовать параллельные вычисления для повышения производительности приложения.

Утилита Intel Thread Profiler позволяет разработчикам контролировать оптимизацию кода, чтобы добиться точной сбалансированности рабочей нагрузки и синхронизации вычислений (более подробную информацию может предоставить руководство Getting Started with the Intel® Thread Profiler<sup>2</sup>).

## Компилятор Intel® C++ 9.0

Компиляторы, автоматически анализирующие циклы и определяющие возможность параллелизации, представляют собой альтернативный вариант, упрощающий труд разработчиков в условиях ограниченных ресурсов. Автоматическая параллелизация — рекомендуемый первый шаг для разработчиков, желающих повысить производительность приложения на многоядерных системах.

Компилятор Intel C++ — это оптимизирующий компилятор для процессоров Intel®, построенных на базе архитектуры IA-32. Предлагаемый в версиях для операционных систем Microsoft Windows и Linux, компилятор Intel® соответствует требованиям языков C и C++, а также обеспечивает совместимость на уровне двоичных кодов с пакетом GCC.

Этот компилятор поддерживает технологию OpenMP и предоставляет средства автоматической параллелизации. OpenMP<sup>3</sup> — это программный интерфейс приложения, основанный на совместном использовании оперативной памяти многопроцессорными платформами. OpenMP имеет простой

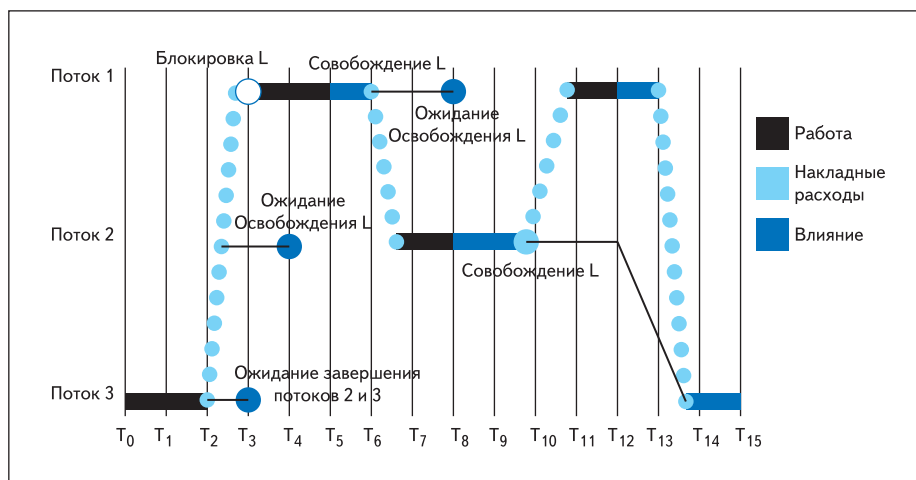


Рис. 3. Утилита Intel® Thread Profiler — посмотрите, как накладные расходы на организацию многопоточных вычислений и синхронизацию влияют на производительность приложения

интерфейс, скрывающий детали управления потоками и их взаимодействия. Разработчики могут указать специальные параллельные регионы кода, добавив соответствующие директивы #pragma в исходный код. Кроме того, директивы #pragma сообщают различную информацию, например о свойствах переменных, а также о необходимости простой синхронизации. Код, использующий технологию OpenMP, остается очень похожим на исходную версию последовательного кода.

В режиме автоматической параллелизации компилятор анализирует циклы и создает для них многопоточный код, что позволяет лучше использовать возможности параллельных вычислений. Компилятор оценивает объем вычислительной работы, выполняемой в цикле. Если, по его мнению, этот объем слишком мал, параллелизация не производится. Эти действия могут быть переопределены путем указания порогового значения. Компилятор разбивает на параллельные вычислительные потоки только те циклы, чья параллелизация, с его точки зрения, безопасна, и сообщает о причинах, по которым автоматическая параллелизация не может быть выполнена.

Библиотеки функций для многопоточных вычислений (доступные также и для интерфейсов POSIX и Windows API) — это решение для разработчиков, которым необходим явный контроль потоков для достижения оптимальной производительности. Эти библиотеки позволяют очень точно управлять вычислительными потоками. Компилятор Intel использует тот же набор базовых библиотек, что и реализация интерфейса OpenMP. Таким образом, код, применяющий технологию OpenMP, тоже может исполняться в режиме автоматической параллелизации.

## Техническая подготовка и ресурсы

Корпорация Intel предлагает широкий спектр обучающих ресурсов для разработчиков. Ниже приведены ссылки на некоторые интерактивные обучающие ресурсы.

- Подразделение Intel® Software College предлагает разносторонние курсы для интерактивного обучения и для аудиторных занятий, которые помогут разработчикам создавать самые передовые приложения и позволят им добиться максимальной производительности с помощью новейших процессоров и технологий Intel.

[ht tps://or1cedar.cps.intel.com/softwarecollege/](https://or1cedar.cps.intel.com/softwarecollege/)

- Сеть Intel® Software Network предоставляет бесплатное обучение и статьи, которые помогут разработчикам ПО в кратчайшие сроки добиться максимальной производительности программ с минимальными усилиями.

[ww w.intel.com/cd/ids/developer/asm-na/eng/index.htm](http://www.intel.com/cd/ids/developer/asm-na/eng/index.htm)

- Дополнительная информация по продукции Intel® для разработчиков ПО, в том числе по компиляторам Intel, анализаторам производительности Intel VTune, библиотекам Intel® Performance Libraries и инструментальным средствам Intel® Threading Tools, доступна по ссылке:

[ww w.intel.com/cd/software/products/asm-na/eng/index.htm](http://www.intel.com/cd/software/products/asm-na/eng/index.htm)

Чтобы в полной мере использовать возможности повышения производительности с помощью многоядерных процессоров, разработчики могут модифицировать свои приложения для применения преимуществ «собственного» параллелизма процессора. Программные инструментальные средства Intel® облегчают переход от последовательной обработки к параллельной, помогая разработчику оценить возможность разделения приложения на потоки и настроить многопоточную обработку для достижения максимальной производительности.

<sup>2</sup> Getting Started with the Intel® Thread Profiler (Начало работы с утилитой Intel® Thread Profiler),

[f tps://download.intel.com/support/performance/tools/threadprofiler/gettingstartedtp.pdf](https://download.intel.com/support/performance/tools/threadprofiler/gettingstartedtp.pdf)

<sup>3</sup> OpenMP standard (другие наименования и товарные знаки являются собственностью своих законных владельцев),

[ht tps://www.openmp.org](http://www.openmp.org)