Задачи для Летней школы 2014

Contents

Нижний Новгород

Задача 1. Оптимизация генераторов случайных чисел библиотеки Intel®MKL под SIMD и многоядерные архитектуры.

Описание задачи. Реализовать новые подходы к оптимизации на примере 1-2 алгоритмов генераторов случайных чисел.

Требования к претенденту. Профессиональный уровень знаний С, начальный уровень знаний по векторизации и бонусом будет начальный уровень знаний по x86 SSE архитектуре\ассемблеру.

Задача 2. Демонстрация возможностей оптимизации с использованием Intel®MKL

Описание задачи. Подготовить пошаговую лабораторную работу, рассказывающую об оптимизации конкретного приложения с использованием методов Intel®MKL.

Требования к претенденту. С, векторизация, OpenMP. Знание x86 SSE ASM является дополнительным преимуществом.

Задача 3. Ускорение алгоритмов анализа больших данных, дружелюбное к пользователю.

Описание задачи. Задача состоит в демонстрации использования высоко оптимизированной библиотеки анализа больших данных из языков программирования, распространённых среди специалистов по работе с данными. Ключевыми требованиями к реализации являются оптимизация переноса данных между языками программирования и обеспечение потоко-безопасности вычислений.

Требования к претенденту. Базовые знания С, SQL, знание основ курса мат. статистики. Желательно знание: R, Python. Знакомство с алгоритмами анализа данных.

Задача 4. Улучшение открытой библиотеки компьютерного зрения OpenCV с помощью библиотеки повышения производительности Intel®Integrated Performance Primitives

Описание задачи. Открытая библиотека компьютерного зрения OpenCV широко используется многими разработчиками в мире для создания приложений, улучшающих человеко-машинное взаимодействие. Эксперименты показали, что использование функций библиотеки повышения производительности Intel® Integrated Performance Primitives (IPP) внутри OpenCV позволяет существенно повысить её производительность на компьютерах с архитектурой компании Intel. Целью данной задачи и является оптимизация OpenCV для архитектур компании Intel с помощью библиотеки IPP. Для этого необходимо:

- определить оптимальные места в OpenCV для встраивания IPP функций,
- вставить IPP функции таким образом, чтобы они эффективно работали в многопоточном режиме,
- убедиться, что изменения не повлияли на результат работы OpenCV,
- собрать статистику производительности OpenCV до и после интеграции IPP,
  и, наконец (и самое главное),
- подготовить все необходимое для того, чтобы изменения были интегрированы в публично доступную версию OpenCV.

Требования к претенденту.

- Уверенное владение C/C++.
- Умение работать с большими проектами.
- Знакомство с OpenCV и IPP.
- Умение понимать чужой код.
- Умение разрабатывать под Windows*.
- Дополнительный плюс: умение работать в Linux*, OS X*, Android*.

Задача 5. Исследование параллельных алгоритмов хеш-таблиц

Описание задачи. Исследование современных подходов к реализации алгоритмов хеш-таблиц для многопоточных приложений, повторение опубликованных результатов исследований на языке C++, создание набора тестов для оценки корректности и производительности выбранных реализаций (бенчмарков), проведение анализа производительности и участие в прототипировании новых идей к реализации.

Требования к претенденту.

- Знание языка программирования C++.
- Знание любого скриптового языка.
- Представление о многопоточном программировании.
- Представление об устройстве современных процессоров.
- Чтение технической литературы на английском языке.

Задача 6. Разработка модуля криптографической временной метки для Защищенного Сервиса Цифровой Подписи (SCSS)

Описание задачи. Реализовать эффективные алгоритмы создания криптографических временных меток согласно стандартам RFC3161 и Authenticode. На их основе разработать модуль для защищенного сервиса цифровой подписи.

Требования к претенденту. Хорошие навыки в программировании на C#, базовые знания основных классов .Net Framework и веб-сервисов. Уровень владения английским языком, достаточный для чтения технической литературы. Желательно знакомство с основами сетевых протоколов и с такими криптографическими стандартами, как ASN.1, PKCS#n и X.509.

Задача 7. Поддержка аппаратной транзакционной синхронизации в библиотеке Intel®TBB

Описание задачи. Библиотека Intel®Threading Building Blocks (Intel®TBB) предоставляет набор высокоуровневых алгоритмов для параллельных вычислений, позволяя существенно упростить процесс разработки параллельных приложений для многоядерных процессоров. Библиотека Intel®TBB содержит ряд нетривиальных параллельных алгоритмов, активно использующих атомарные операции.

Необходимо разработать новые алгоритмы с поддержкой аппаратных транзакций Intel®Transactional Synchronization Extensions (Intel®TSX) на языке программирования C++ для компонентов библиотеки Intel®TBB с целью улучшения их производительности на процессорах нового поколения 4th Generation Intel® Core™ Processor (code name Haswell). Суть задачи – улучшение компонентов библиотеки Intel®TBB, которые могут выиграть от использования аппаратных транзакций, переработка их фрагментов в виде, дружелюбном к транзакциям, и анализе производительности того, что получилось.

Требования к претенденту.

- Знание C++;
- опыт многопоточного программирования, представление о неблокирующей синхронизации;
- представление об устройстве современных процессоров;
- умение разбираться в существующем коде.

Задача 8. Разработка приложения для Android, позволяющего осуществлять взаимодействие между таблетками (смартфонами) и интерактивной доской

Описание задачи. Разработка прототипа приложения для мобильных устройств на базе Android. Задача приложения: автоматическое сохранение в векторном формате записей на интерактивной доске.

Требования к претенденту. Знание C/C++ и Java. Опыт разработки приложений для мобильных устройств будет плюсом.

Задача 9. Анализ энергопотребления популярных приложений для мобильных платформ Intel (Android, Win8)

Описание задачи. Участие в проекте по поддержке запуска платформ Intel для мобильных устройств. Анализ энергопотребления приложений, валидация приложений. ОС: Android, Win8

Требования к претенденту. Знание Python, bash, Android, Win8 (на уровне разработчика)

Задача 10. Разработка приложения, позволяющего осуществлять общение и обмен образовательной информацией на базе IA устройств

Описание задачи. Разработка мобильного приложения для Android. Задача приложения – организовать удобный процесс общения между студентами и обмен образовательной информацией (лекции). Для работы будет использована передовая технология Intel Common Connectivity Framework, которая позволяет простой и удобный способ кроссплатформенного общения независимо от операционной системы.

Требования к претенденту. Знание C/C++ и Java. Опыт разработки приложений для мобильных устройств будет плюсом.

Задача 11. Внедрение модуля отслеживания объектов в Real Sense SDK

Описание задачи. Необходимо внедрить имеющиеся исследовательские коды отслеживания объектов в Real Sense SDK и написать демонстрационное приложение, иллюстрирующее работу модуля.

Требования к претенденту. Хорошее знание C/C++, OpenCV. Знакомство с базовыми принципами компьютерного зрения.

Задача 12. Разработка демонстрационной программы на основе вычислительного графа

Описание задачи. Библиотека Intel®Threading Building Blocks (Intel®TBB) включает в себя реализацию вычислительного графа - Intel TBB flow graph, позволяющего создавать «неструктурированные» параллельные приложения.

Задача состоит в разработке визуализированной демонстрационной программы, основанной на TBB flow graph, демонстрирующей преимущества этой модели в производительности и программируемости. Это может быть как портированием готового открытого проекта на TBB flow graph, так и разработкой нового приложения.

Требования к претенденту.

- Навыки программирования на C++;
- базовые знания о многопоточном программировании;
- умение разбираться в существующем коде;
- умение читать и писать техническую документацию на английском;
- знакомство с TBB на уровне обучающего руководства (www.threadingbuildingblocks.org).

Задача 13. Разработка демо-приложения для Intel® INDE Media Pack for Android

Описание задачи. Intel®INDE Media Pack for Android это простой и удобный в использовании продукт, который расширяет возможности работы с медиа (видео, аудио, камера) Android. С помощью Media Pack разработчик может создать интересные приложения, внедрив работу с редактированием видео, применением видео эффектов, стримингом видео с мобильного устройства или захвата видео с экрана, распознаванием контента по «аудио отпечаткам». Возможности Media Pack продолжают расширяться.

Необходимо создать мобильное приложение с использованием Media Pack, взяв за основу свою оригинальную идею или наше предложение.

Требования к претенденту.

- Минимальный опыт создания мобильных приложений для Android.
- Умение креативно мыслить.

Задача 14. Разработать fuzzing для видео декодера

Описание задачи. Команда Libraries Product Security (LibPSEC) занимается улучшением безопасности програмных продуктов Интел. Примером такого продукта может быть Intel Media SDK. Эта библиотека предустановлена на каждом устройстве на базе платформы Intel, и она участвует в декодировании видео файлов.

Необходимо реализовать метод поиска уязвимостей безопасности в видео декодере из Intel®Media SDK на основе Peachfuzzer. Это включает в себя описание контейнера видео потока на языке Peachfuzzer (pit файл), а также расширений Peachfuzzer на C#.

Необходимо также разработать инфраструктуру генерации тестовых данных на базе Peachfuzzer и обнаружения уязвимостей при помощи инструментов мониторинга работы приложения (таких как Intel®Inspector, Microsoft Application Verifier).

Требования к претенденту.

- Базовые знания C, XML, CABAC, CAVLC, H.264.
- Уверенное программирование на C#.
- Навыки отладки приложений.
- Чтение документации на английском.

Задача 15. Повышение производительности memcpy для Intel® MPI Library

Описание задачи. Копирование данных – одна из критических операций для производительности Intel®MPI Library (IMPI). Несколько MPI-специфичных оптимизаций этой функциональности были успешно реализованы и применены в различных частях библиотеки и могут быть распространены по всей IMPI.

Задачи:

1. Провести анализ и построить профиль использования различных реализаций memcpy в IMPI.
2. Создать единую для всей IMPI оптимизированную версию memcpy для использования по всей IMPI.

Требования к претенденту.

- Базовое знание особенностей современных Intel архитектур.
- Базовый уровень программирования на C, bash.
- Понимание простого кода на intrinsic/asm.

Задача 16. Улучшение функциональности “replay” для отладки MPI приложений

Описание задачи. Функцинальность “replay” может быть использована для анализа MPI приложений. Суть в том, что MPI вызовы запоминаются и сохраняются в виде файла на языке C. Этот файл может быть использован для эмуляции и анализа приложения. Задание: увеличить количество запоминаемых вызовов.

Требования к претенденту. Умение программировать на C и bash, понимать чужой код, читать и писать по-английски.

Задача 17. Разработать прототип экспертной системы для рекомендаций по оптимизации в Intel®Parallel Studio Advisor.

Описание задачи. Разработать прототип экспертной системы, способной давать советы по векторизации и распараллеливанию кода на основе диагностик компилятора и данных по статическому и динамическому анализу кода. Частью задания будет общение с экспертами для выявления и формализации наиболее часто встречающихся паттернов оптимизации.

Требования к претенденту.

- Представление об экспертных системах (байесовы сети и т.п.).
- Представление об оптимизации (векторизации) кода.
- Технический английский .

Задача 18. Вероятностное моделирование процессорного кеша по картам доступа к памяти

Описание задачи. Реализовать алгоритм, который рассчитывает вероятное число «попаданий» и «промахов» в процессорном кеше по имеющимся картам распределения доступа к памяти реального приложения. Сравнить результаты моделирования с фактическими замерами приложения.

Требования к претенденту. Хороше знание C++, общее представление о том, как работает кеш процессора.

Задача 19. Разработка примеров для Интернет Вещей/Galileo

Описание задачи. Разработать несколько примеров, демонстрирующих возможности IOT dev kit для Galileo, и работающих в разных ОС – Yocto Linux, VxWorks, baremetal, Labview.

Требования к претенденту. Уверенный пользователь Линукс, понимающий среду разработки Линукс – gcc, Makefiles, config scripts.

Задача 20. Оптимизация размера собираемых трасс с данными о производительности программы для продукта Intel®Trace Analyzer and Collector

Описание задачи. Для задач профилирования высокопроизводительных приложений для кластерных систем большого масштаба (из списка Top500 самых высокопроизводительных вычислительных систем) характерны огромные объемы собираемой информации. Однако для вычислений, проходящих в циклах (что характерно для большинства высокопроизводительных вычислительных алгоритмов) при профилировании, имеет смысл выделять только репрезентативные итерации. В результате объем собираемой информации может быть до 100 000 раз меньше исходного. Для профилировщика высокопроизводительных приложений на кластерных системах Intel® Trace Analyzer and Collector необходимо исследовать подход со сбором данных только для значимых итераций цикла, разработать метрики и критерии оценки такого подхода.

Требования к претенденту.

- Операционные системы GNU/Linux, Microsoft* Windows*;
- Языки программирования C/C++, Fortran, awk;
- средства лексического анализа, синтаксического, семантического анализа,
- GNU grep; Технический английский.

Задача 21. Анализ возможностей интеграции кластерных систем управления заданиями и Intel®MPI library

Описание задачи. Большинство современных высокопроизводительных кластерных систем используют системы управления заданиями для эффективного использования ресурсов. Интеграция таких систем и Intel®MPI library важно для удобства использования параллельных приложений.

В рамках данного задания необходимо проанализировать существующие системы управления заданиями, оценить возможности и методы интеграции Intel®MPI Library и данных систем.

Требования к претенденту. Программирование shell, С программирование, Unix-like ОС. Технический английский.

Задача 22. Визуализация отдельных MPI метрик с помощью универсальной системы визуализации данных GrandMesa

Описание задачи. Система визуализации метрик GrandMesa позволяет визуализировать любые метрики, поступающие от различных утилит в режиме реального времени. Задача состоит в исследовании применимости данной системы для визуализации метрик, специфичных для оценки, и оптимизации производительности MPI приложений. Для решения задачи необходимо разработать прототип коллектора MPI вызовов и интегрировать его в систему GrandMesa. Также необходимо оценить накладные расходы подобного решения. IMPI библиотека должна использоваться как реализация стандарта MPI.

Требования к претенденту.

- Хорошее знание C/C++.
- Понимание и умение применять принципы ООП.
- Навыки чтения и понимания технической документации на английском.
- Опыт программирования крос-платформенных решений для ОС Linux и Windows.

Задача 23. Графическая утилита, основанная на HTML5 технологии, для визуализация Intel MPI статистики

Описание задачи. Статистика библиотеки Intel®MPI представляет из себя текстовый файл известного формата. Для больших запусков файл статистики может достигать сотен мегабайт, что делает его трудным для чтения и анализа. Задача состоит в разработке HTML5 кроссплатформенного приложения для визуализации Intel MPI статистики. Для разбора файла статистики должен использоваться существующий парсер, написанный на языке C++.

Требования к претенденту.

- Хорошее знание C++ и стандартной библиотеки C++ (STL).
- Понимание и умение применять принципы ООП.
- Навыки чтения и понимания технической документации на английском.
- Навыки программирования на HTML5.

Задача 24. Разработка крипто провайдера, основанного на Защищенном Сервисе Цифровой Подписи (SCSS)

Описание задачи. Разработать модуль, реализующий Windows CryptoAPI CSP интерфейс для того, чтобы предоставить доступ к SCSS сервису для таких инструментов разработки, как Visual Studio, WHQL HCK, и ряда внутренних утилит.

Требования к претенденту. Хорошие навыки в программировании на C#, базовые знания основных классов, базовые знания в криптографии, SOAP. Уровень владения английским языком, достаточный для чтения технической литературы.

Задача 25. Расширение Google Android Licensing Verification Library для использования с продуктами Intel по защите ПО

Описание задачи. Google предлагает Android-разработчикам сервис лицензирования (Application Licensing Service), который защищает от пиратства и несанкционированного использования приложений. Android SDK включает в себя пример использования этого сервиса на Java в виде License Verification Library (LVL). Задача интеграции LVL в приложение так, чтобы исключить возможность взлома, является нетривиальной. Инструменты защиты кода от компании Intel позволяют упростить эту задачу.

Необходимо разработать защищенный модуль верификации лицензий для Android приложений на Java и C/C++, использующий Application Licensing service. Для защиты разработанного модуля от взлома необходимо использовать средства обфускации нативного кода и white-box cryptography, предложенные Intel.

Требования к претенденту.

- Опыт разработки приложений под Android.
- Хорошее знание основ криптографии (протоколы, алгоритмы шифрования и подписи и т.д.).
- Уверенное программирование на C/C++.
- Высокий уровень навыков письменного и разговорного английского языка.

Новосибирск

Задача 1. Интеграция в базу данных измерений производительности компиляторов

Описание задачи. Наша команда проводит различные измерения производительности для встроенных систем, но не имеет удобного механизма для хранения, сравнения и получения отчетов для этих измерений. Необходимо разработать механизмы для интеграции результатов измерений в базу данных для последующей визуализации и анализа.

Требования к претенденту. Perl, C#, XML, SQL, Linq.

Задача 2. Анализ и проектирование стратегии балансировки нагрузки для процессов автоматизации

Описание задачи. В рамках этого проекта предлагается проанализировать существующее решение, а также предложить и реализовать механизм эффективного распределения задач по вычислительным ресурсам для уменьшения общего времени исполнения задач и улучшения загрузки компьютеров.

Требования к претенденту. Знание скриптовых языков программирования Perl, Windows CMD, bash, базовые знания об алгоритмах обработки больших объемов данных.

Задача 3. Эффективный анализ изменений производительности

Описание задачи. В рамках этого проекта предлагается собрать и проанализировать статистику найденных изменений производительности кода генерируемого компилятором. На основе полученных данных разработать методологию и инструментарий, выдающий оптимальную последовательность действий поиска изменений производительности связанных с изменениями кода компилятора.

Требования к претенденту. Знание C#, SQL, математическое образование.

Задача 4. Поиск оптимальных параметров для развертки циклов

Описание задачи. В рамках этого проекта предлагается изучить возможности цикловых оптимизаций и в особенности развертки циклов (анроллинга, unroll). На некотором наборе приложений исследовать производительность в зависимости от параметров анроллинга и предложить оптимальный набор параметров для каждой функции и/или цикла в программе. Написать инструмент (скрипт) для автоматизации данного исследования.

Требования к претенденту. Знание C/C++, Fortran, shell, Perl, Linux. Опыт ускорения приложений и работы с Intel® VTune Performance Analyzer будет плюсом.

Задача 5. Реорганизация модели использования Intel®Math Kernel Library

Описание задачи. Intel®Math Kernel Library предоставляет оптимизированный набор математических функций для различных архитектур и может быть использована с различными компиляторами. Intel®MKL имеет многоуровневую структуру, чтобы удовлетворить всем пользовательским моделям использования. Необходимо реорганизовать текущую модель использования Intel®MKL, что поможет продемонстрировать новые модели использования Intel®MKL в пользовательских приложениях.

Требования к претенденту. Знания MSVS, GNU-make, C, Fortran.

Задача 6. Интерактивные примеры использования Intel® Math Kernel Library в обработке изображений

Описание задачи. Многие приложения по обработке изображений требуют хорошо оптимизированных алгоритмов, в частности тех, которые предоставляет Intel®MKL. Ключевой функциональностью в обработке изображений является Быстрое Преобразование Фурье (БПФ). В рамках данного проекта необходимо создать визуально привлекательное приложение, демонстрирующее правильное использование БПФ из библиотеки Intel®MKL для получения высокой производительности.

Требования к претенденту. Знание C++ и Qt. Дополнительным плюсом будет знание алгоритмов обработки изображений.

Задача 7. Решатель для сеточного уравнения Пуассона на машинах с распределенной памятью

Описание задачи. Множество прикладных задач требуют решения уравнения Пуассона в простой области с постоянными коэффициентами как предобуславливатель для более сложной системы. Подобный решатель для компьютеров с общей памятью реализован в библиотеке Intel®Math Kernel Library. В рамках данного проекта предлагается адаптировать его для кластера.

Требования к претенденту. Знание C, OpenMP, MPI.

Задача 8. Анализ исторических данных тестовых прогонов с целью выявления возможности улучшения алгоритма по наследованию результатов

Описание задачи. Наша команда анализирует результаты тестирования компиляторов Intel®. В каждом тестовом прогоне мы имеем тысячи тестов. Мы часто сталкиваемся с тем, что одни и те же тесты «падают» на разных операционных системах/платформах по одной и той же причине. На данный момент мы автоматически используем результаты анализа (номера отчетов об ошибках) для тестирований, у которых совпадает операционная система и платформа. В рамках данного проекта предлагается проанализировать имеющиеся исторические данные и понять, можно ли расширить имеющийся алгоритм и использовать результаты анализа для тестирований на платформах IA32 и Intel®64.

Требования к претенденту. Базовое знание Perl, обработчиков текстовых файлов.

Задача 9. Реализация решения задачи по нахождению нестабильных тестов в регулярных прогонах с использованием базы данных

Описание задачи. Наша команда анализирует результаты тестирования компиляторов Intel®, при этом в каждом тестовом прогоне мы имеем тысячи тестов. Некоторые из этих тестов имеют нестабильное поведение. В рамках данного проекта предлагается определить список таких тестов. Кроме того, требуется отследить дальнейшее поведение известных нестабильных тестов, чтобы понять, когда эти тесты перестали быть нестабильными.

Требования к претенденту. Базовое знание Perl, DB, XML.

Задача 10. Вычисление минимальных и максимальных собственных чисел для разреженных матриц

Описание задачи. В рамках данного проекта предлагается реализовать эффективный алгоритм поиска первых собственных значений для задачи $Ax = \lambda x$, где A – разреженная матрица. Реализация планируется на основе существующей функциональности библиотеки Intel®Math Kernel Library для работы с разреженными матрицами. в частности, Extended Eigensolver и Intel®MKL PARDISO.

Требования к претенденту. Базовые знания в области вычислительной математики, C/C++, Fortran, Linux, bash.

Задача 11. Улучшение алгоритма работы парсера Фортран файлов при интеграции Intel®Fortran компилятора в Microsoft Visual Studio*

Описание задачи. Для обеспечения работы IntelliSense в редакторе Фортран файлов применяется парсирование исходных файлов и сохранение результатов парсирования в базу данных. Текущий алгоритм является однопроходным и имеет недостатки, связанные с зависимостью результатов от порядка парсирования. В рамках данного проекта предлагается придумать и реализовать новый многопроходный алгоритм, который решит существующие проблемы.

Требования к претенденту. Знание C# обязательно, знания языка Fortran будут плюсом. Кроме того, необходимо иметь представление о работе генераторов парсеров, лексических анализаторов и компиляторов.

Задача 12. Разработать примеры векторизации для архитектур Intel MIC и GT

Описание задачи. В рамках данного проекта предлагается создать или улучшить существующие программы для демонстрации технологии векторизации на Intel MIC и GT платформах, а также перенести существующие примеры для процессоров Intel® Xeon® на Intel GT.

Требования к претенденту. Знание C/C++.

Задача 13. Разработка и оптимизация NDK программ для Android*

Описание задачи. В рамках данного проекта предлагается оптимизировать и распараллелить существующую программу для операционной системы Андроид (Android*), а также найти или разработать другие программы для демонстрации преимуществ компилятора Intel® Android.

Требования к претенденту. Знание Java, C++.

Задача 14. Сервис Данных Производительности

Описание задачи. В рамках данного проекта предлагается разработать Web сервисы для работы с данными производительности. В нескольких проектах в Intel® используются различные системы для работы с данными производительности приложений. Необходимо для 3-х выбранных проектов разработать общую Web систему, расширить имеющуюся функциональность по генерации отчётов, а также сделать её универсальной для использования в других проектах.

Требования к претенденту. Необходимые знания: Linux shell, Python, SQL, JavaScript; желательно знать: PHP, Java, VBS.

Задача 15. Расширение возможностей инструментов для отладки графического драйвера

Описание задачи. В рамках данного проекта предлагается разработать новые возможности для инструментов, используемых при тестировании и отладке графического драйвера. Эти инструменты работают с последовательностью вызовов функций из интерфейса драйвера, сделанных графическим приложением. Необходимо реализовать возможность усечения последовательностей вызовов для упрощения локализации проблем.

Требования к претенденту. Знание C++, OpenGL.

Задача 16. Транспонирование векторных данных при помощи набора инструкций x86

Описание задачи. В рамках данного проекта предлагается создать утилиту для нахождения всех возможных последовательностей инструкций, осуществляющих перестановку данных в SIMD векторах в соответствии с заданным шаблоном перестановки.

Требования к претенденту. Знание C/C++.

Задача 17. Раскладка визуализированного графа исполнения

Описание задачи. В рамках данного проекта предлагается реализовать алгоритм по раскладке графа исполнения, создающий удобочитаемое визуализированное преставление на базе библиотеки jgraphx. Также предлагается реализовать алгоритм раскладки дуг в визуализированном графе исполнения с фиксированным положением вершин.

Требования к претенденту. Знание Java

Задача 18. Улучшение реализации распространенных алгоритмов для оптимального их выполнения на сопроцессоре Intel® Xeon Phi™

Описание задачи. Производительность кода, порождаемого компилятором Интел для сопроцессора Intel Xeon Phi, измеряется на наборе распространенных алгоритмов. В рамках данного проекта предлагается улучшить реализацию данных алгоритмов, чтобы полностью раскрыть потенциал архитектуры и лучше протестировать векторизационные возможности компилятора.

Требования к претенденту. Знание C/C++, ассемблера x86.