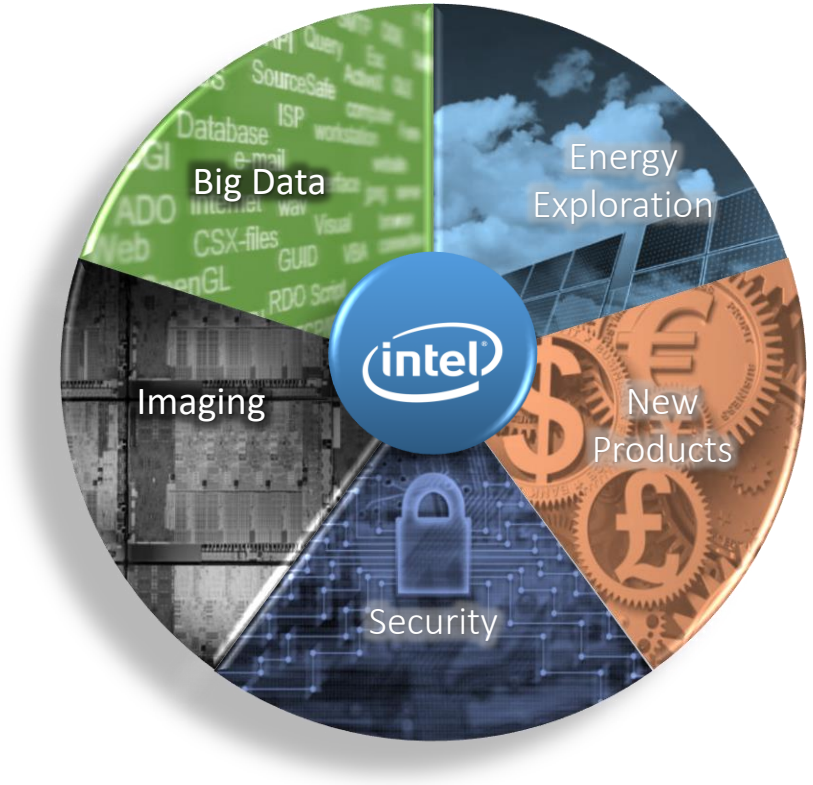
# Intel Advisor и Roofline Анализ

Ануфриенко Андрей

"Intel® Software 2017: HPC Code Modernization and Artificial Intelligence“

15 сентября 2017

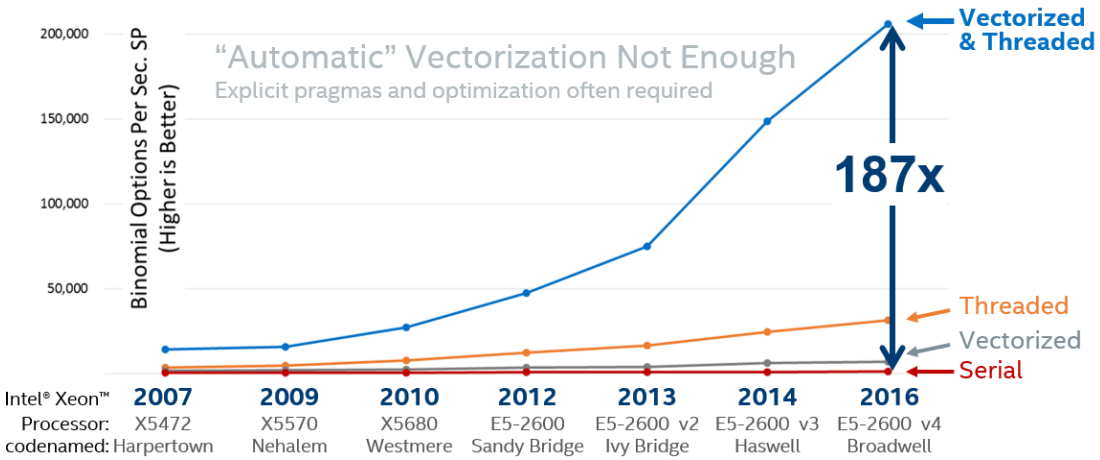
# HPC :
# Для того чтобы конкурировать нужно вычислять



# Последовательные приложения

# Intel® Advisor – Модернизируйте ваш код

## Векторизация и прототипирование многопоточности

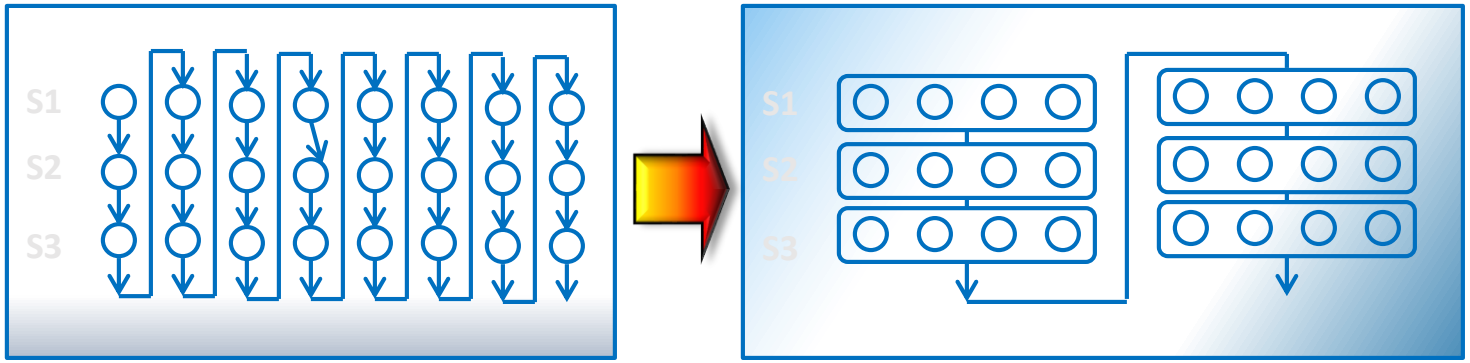



**Производительность увеличивается с каждым новым поколением железа**

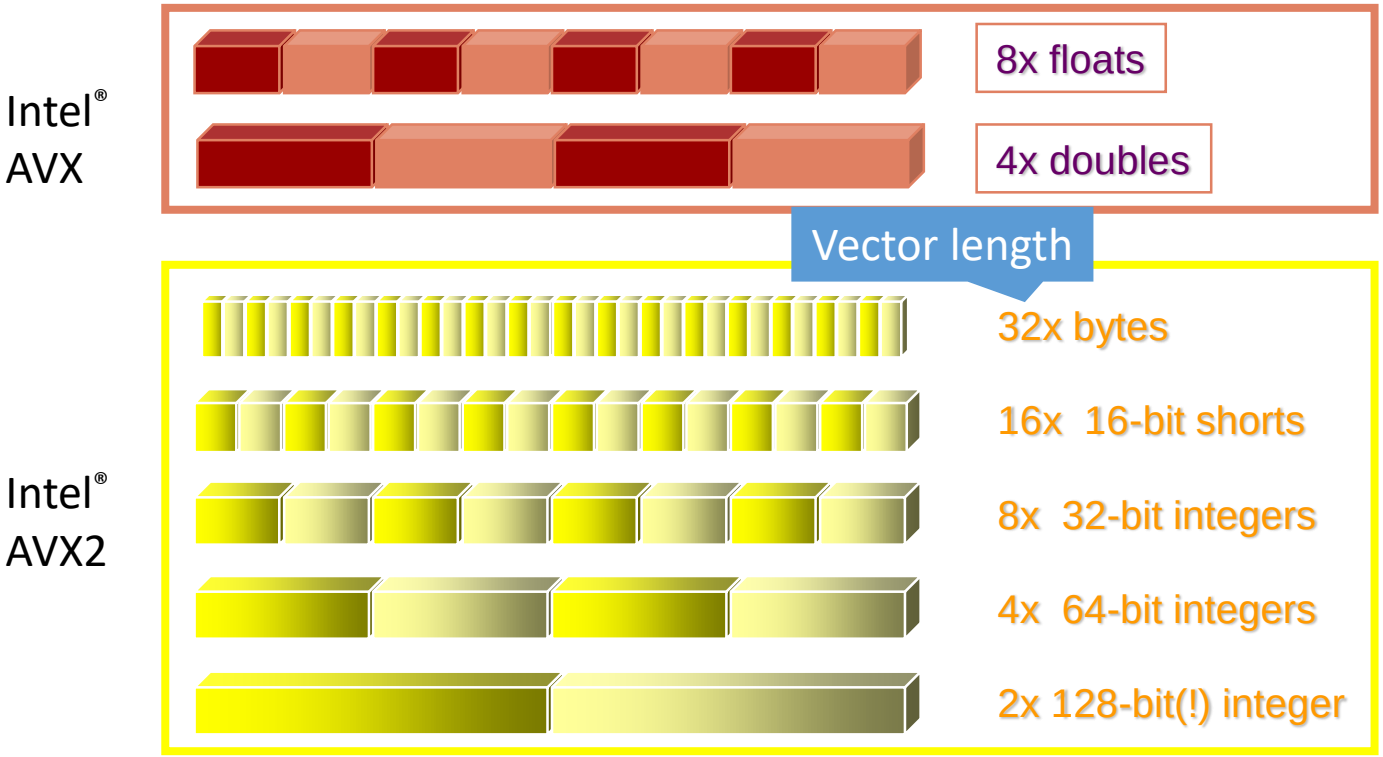
- Векторизуйте и распараллеливайте для максимальной производительности приложений на современных процессорах
- Получайте информацию о числе итераций, зависимостях по данным, особенностях доступа к памяти
- Следуйте простому рабочему процессу оптимизации с подсказками для получения более быстрого кода

# Векторные операции (SIMD)

```
for(i = 0; i <= MAX; i++)
    c[i] = a[i] + b[i];
```

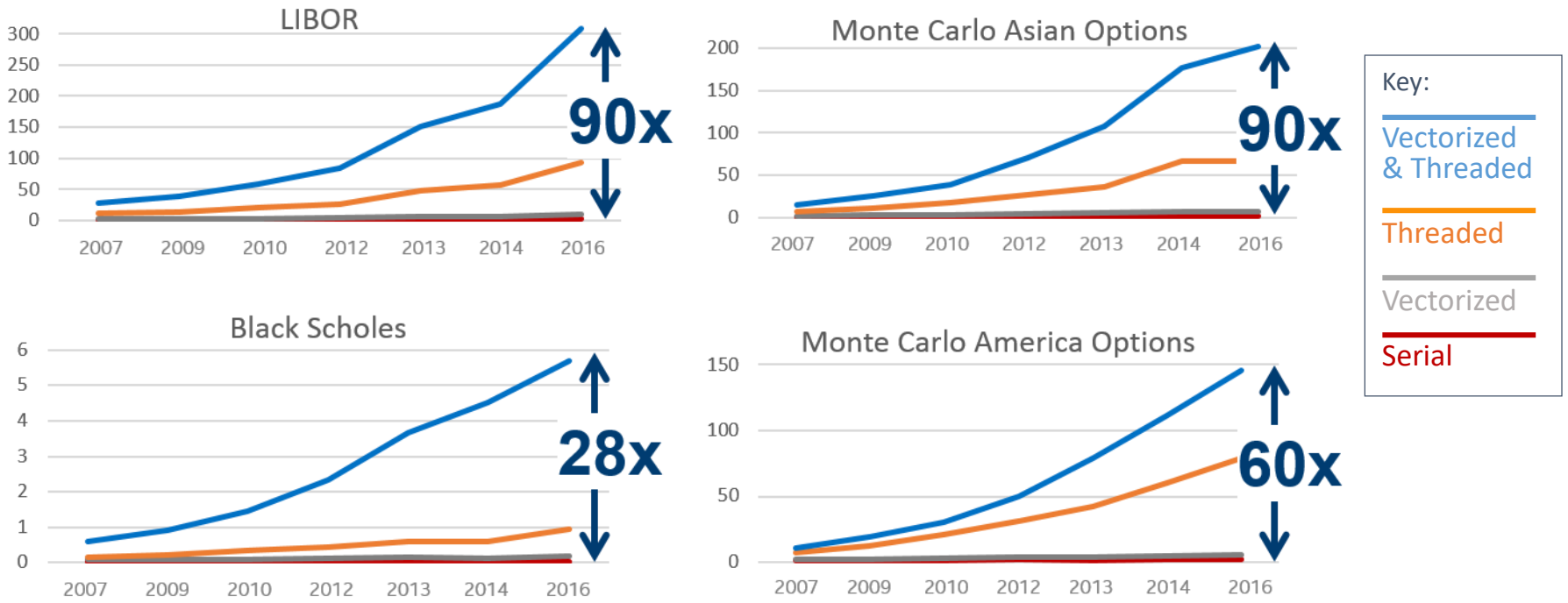
# Intel® Advanced Vector Extensions (Intel® AVX)



# Используйте весь параллелизм

| Intel Advisor XE: | Threading | Vectorization |
|---|---|---|
| | ✓ | ✓ |

# Векторизация и многопоточность критична для современных вычислительных систем




Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more information go to http://www.intel.com/performance Configurations at the end of this presentation.

# Автоматической векторизации часто недостаточно

- Компилятор не всегда может векторизовать ваш код
  - Проверьте наличие цикловых зависимостей используя Intel® Advisor
  - Все понятно? Векторизуйте принудительно. C++: pragma simd,

    Fortran use: SIMD directive
- Не всякая векторизация одинаково эффективна
  - последовательный доступ эффективнее чем доступ с шагом. Анализируйте с Intel® Advisor.
  - Рассмотрите возможность изменения структуры данных. Intel® SIMD Data Layout Templates могут помочь

Бенчмарки на предыдущих слайдах были ускорены не только за счет автовекторизации. Были использованы директивы для принудительной векторизации и т.п.

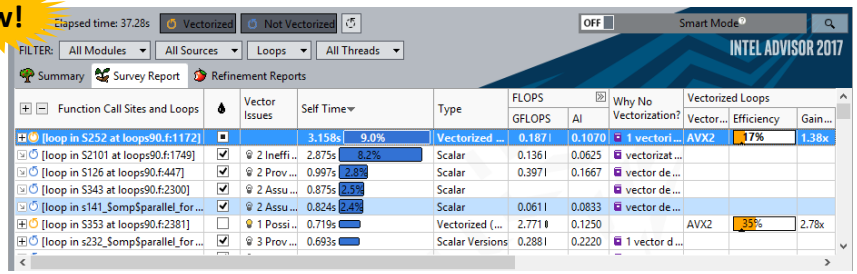
Массивы структур удобны для интуитивной организации данных, но значительно более эффективной для векторизации является организация данных как структура массивов. Использование Intel® SIMD Data Layout Templates (Intel® SDLT) позволяет маппировать данные в более эффективные структуры для векторизации.

# Быстрый код быстрее с дизайном управляемым данными

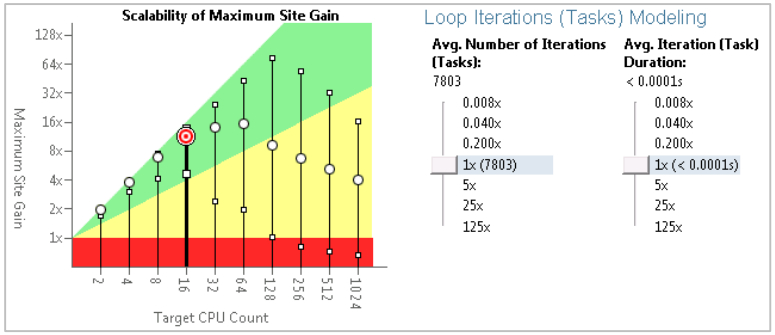
**Intel® Advisor – векторизация и прототипирование многопоточности**

- Наиэффективнейшая параллелизация:
  - Векторизуем там, где это приносит наибольшую выгоду
  - Быстрое определение блокирующих проблем
  - Подсказки для эффективной векторизации
  - Безопасная векторизация
  - Оптимизация доступа к памяти

New!




- Прорыв для многопоточного дизайна:
  - Быстрое прототипирование
  - Масштабирование проекта на больших системах
  - Поиск ошибок синхронизации до имплементации многопоточности
  - Проектирование без утомительной разработки



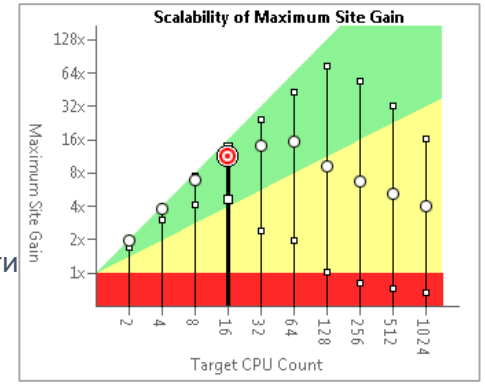

**Добавляйте параллелизм с наименьшими усилиями и риском и наибольшей отдачей**

Part of Intel® Parallel Studio
http://intel.ly/advisor-xe

# Сделай быстрый код быстрее! Intel® Advisor

Прототипирование многопоточности

- Случалось ли у вас, что:
  - Параллелизация приложения дает скромный эффект?
  - Достигли «барьера масштабируемости»?
  - Отложили релиз из-за ошибок синхронизации?
- Параллелизация управляемая данными:
  - Быстрое прототипирование
  - Масштабирование проекта на больших системах
  - Поиск ошибок синхронизации до имплементации многопоточности
  - Проектирование без утомительной разработки




**Добавляйте параллелизм с наименьшими усилиями и риском и наибольшей отдачей**

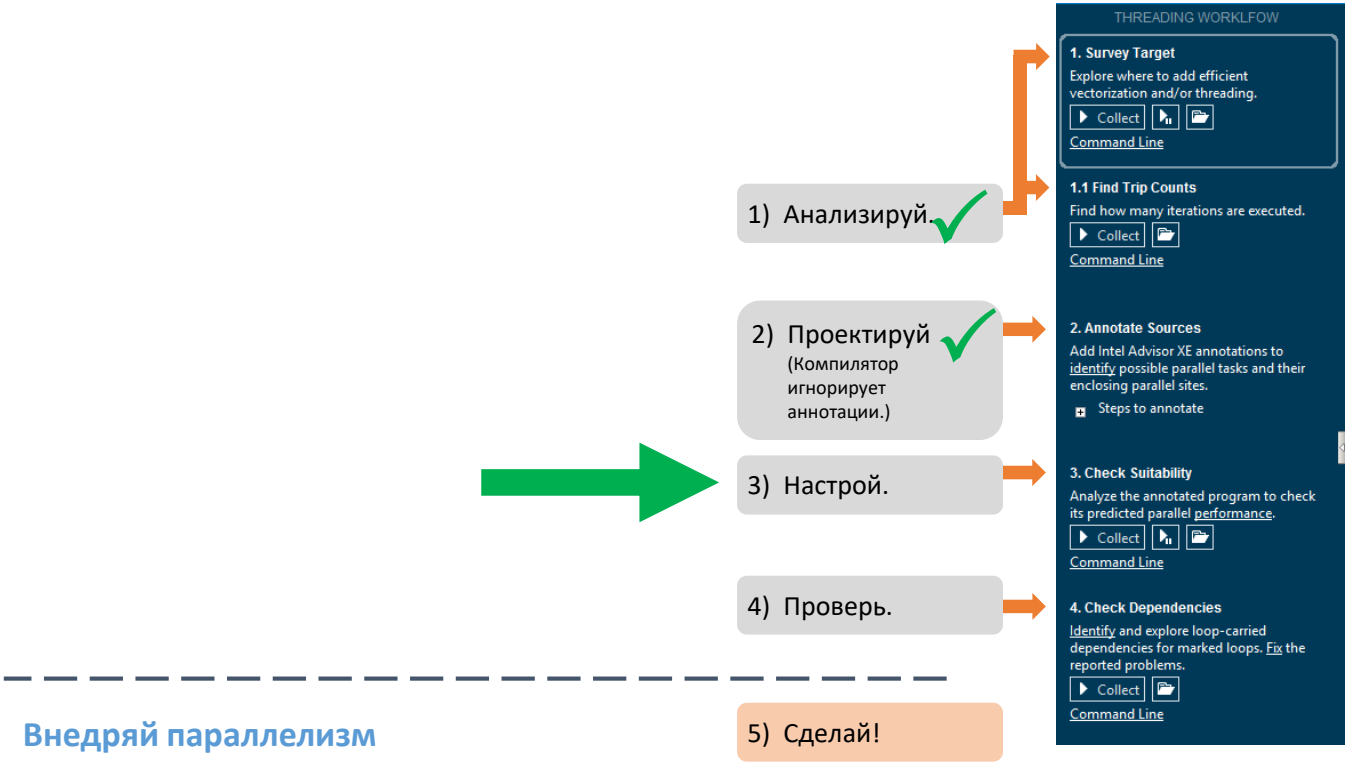
"**Intel® Advisor** has allowed us to quickly prototype ideas for parallelism, saving developer time and effort"

*Simon Hammond*
*Senior Technical Staff*
***Sandia National Laboratories***

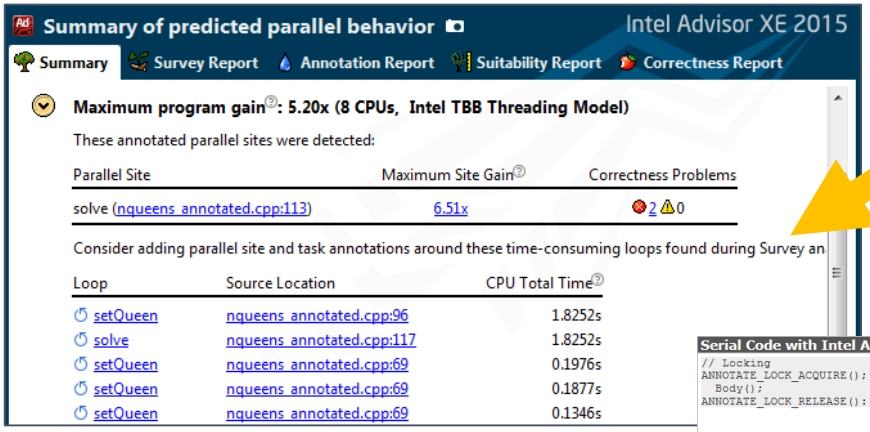
http://intel.ly/advisor-xe

# Спроектируй и затем реализуй

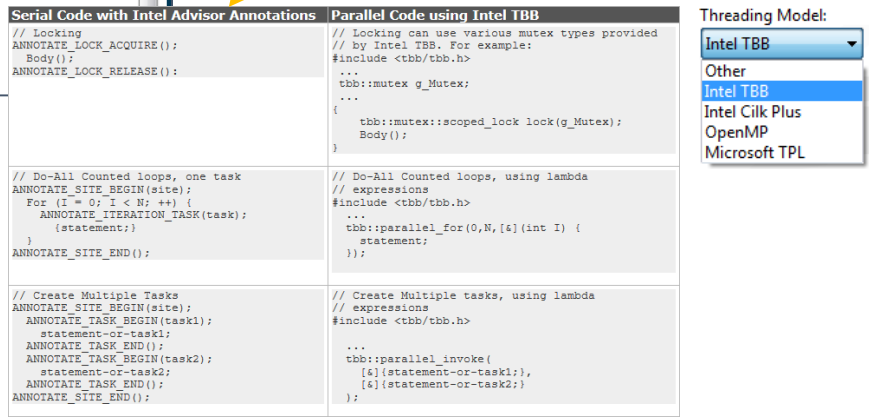
**Intel® Advisor прототипирование многопоточности**



# Добавь параллельную реализацию




- Список позиций в коде
- Шаблоны для популярных методов реализации многопоточности



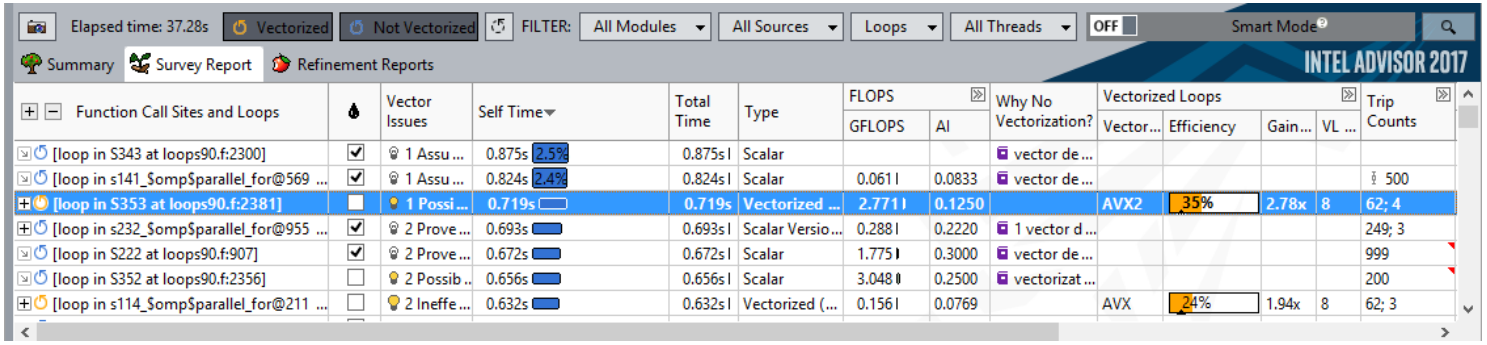

## Intel® Advisor

- Содержит основные метрики для популярных параллельных реализаций
- Быстрое прототипирование и оценка альтернатив
- Детальные подсказки для популярных многопоточных реализаций

# Vectorization Advisor

Intel® Advisor – Векторизация и прототипирование многопоточности

- Наиэффективнейшая параллелизация:
  - Векторизуем там, где это приносит наибольшую выгоду
  - Быстрое определение блокирующих проблем
  - Подсказки для эффективной векторизации
  - Безопасная векторизация
  - Оптимизация доступа к памяти
- Данные и необходимая помощь:
  - Диагностика компилятора + данные о производительности + SIMD эффективность
  - Определить проблемы и рекомендовать их решения
  - Анализ цикловых зависимостей
  - Анализ методов доступа к памяти



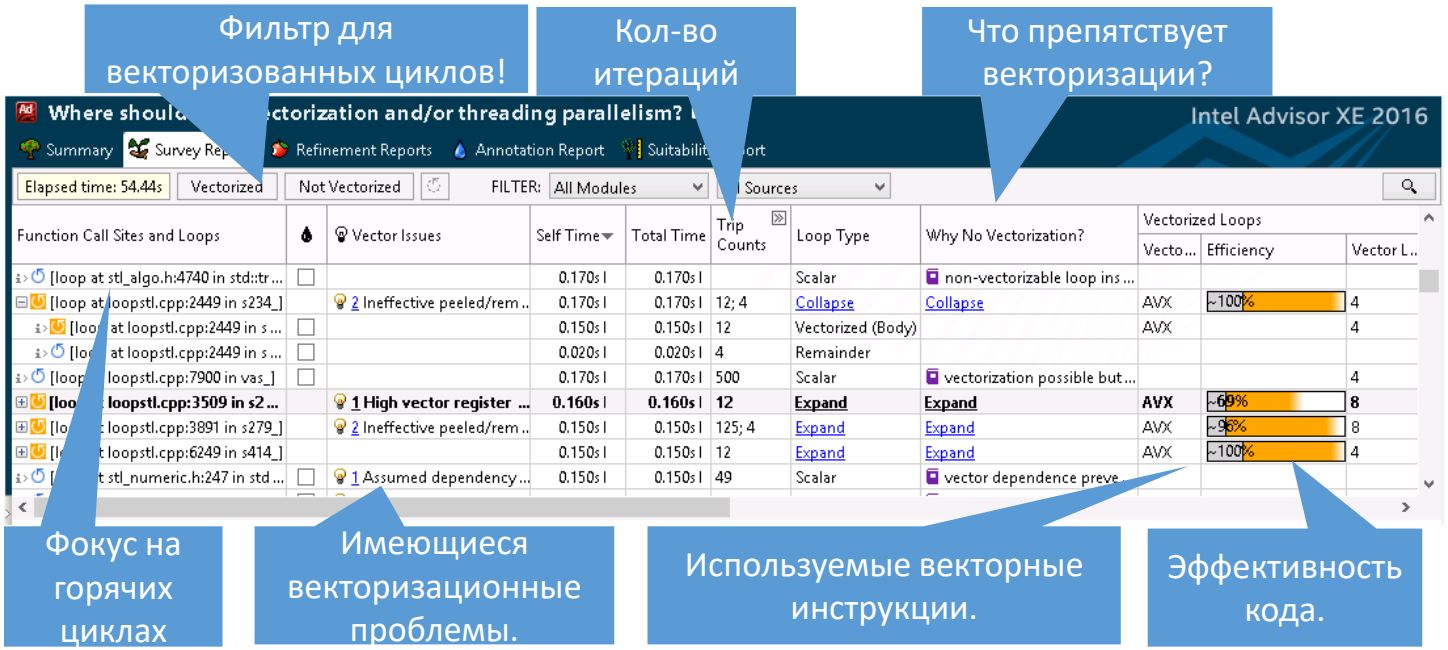

**Оптимизация для AVX-512 с и без доступа к AVX-512 железу**

http://intel.ly/advisor-xe

Part of Intel® Parallel Studio XE

# Правильные данные на кончиках ваших пальцев

Получите всю информацию необходимую для эффективной векторизации




**Сделай быстрый код быстрее!**

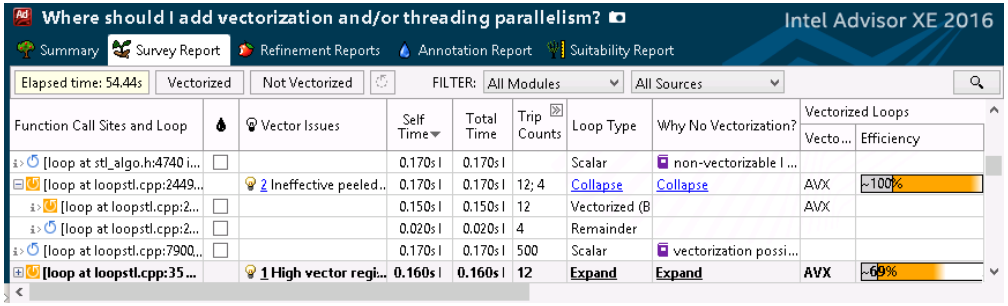
# Сделай быстрый код быстрее! Intel® Advisor

## Векторизация

- Случалось у вас, что:
  - Перекомпилировали для AVX2 с маленьким улучшением?
  - Не понимали где векторизовать?
  - Переписывали интринсики для новых архитектур?
  - Мучались с компиляторными отчетами?

New!

- Векторизация управляемая данными:
  - Что векторизовать в первую очередь?
  - Что и почему мешает векторизации?
  - Мои циклы векторизовались успешно?
  - Можно реорганизовать данные для улучшения производительности?
  - Безопасно ли использование прагмы simd?

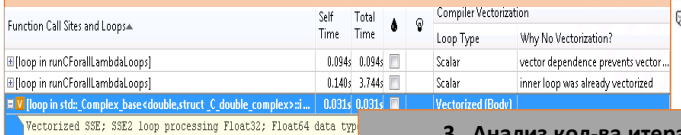



"**Intel® Advisor's** Vectorization Advisor permitted me to focus my work where it really mattered. When you have only a limited amount of time to spend on optimization, it is invaluable."

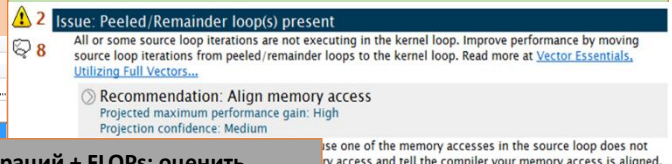
*Gilles Civario*
*Senior Software Architect*
***Irish Centre for High-End Computing***

# 5 шагов к эффективной векторизации - Vector Advisor

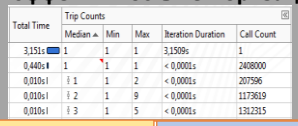
**1. Диагностика компилятора + Данные о производительности + Информация об эффективности векторизации**



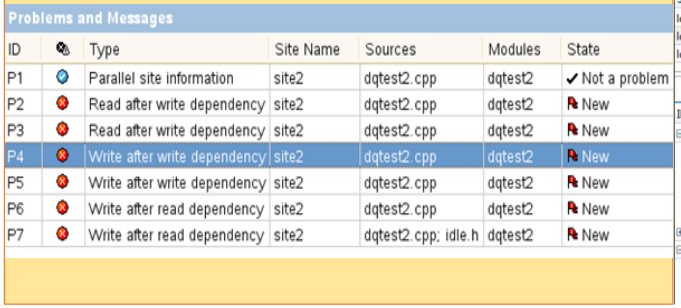

**2. Руководство: определить проблемы и дать рекомендации для их исправления**



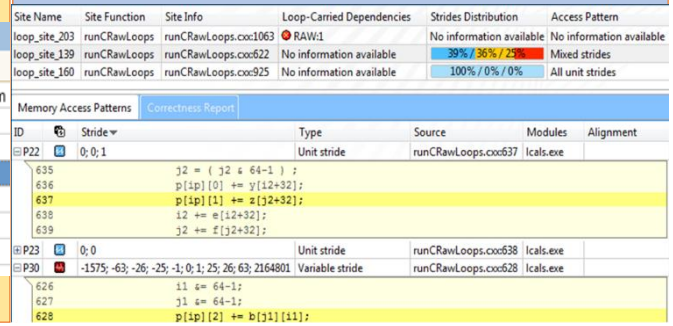

**3. Анализ кол-ва итераций + FLOPs: оценить эффективность векторизации**




**4. Анализ цикловых зависимостей**



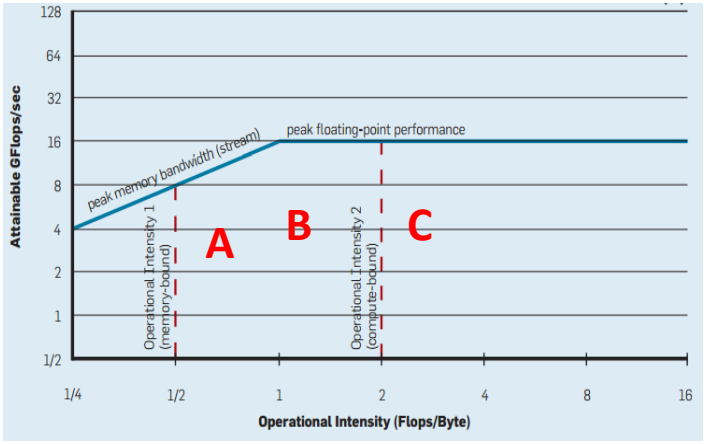

**5. Анализ типа доступа к памяти**



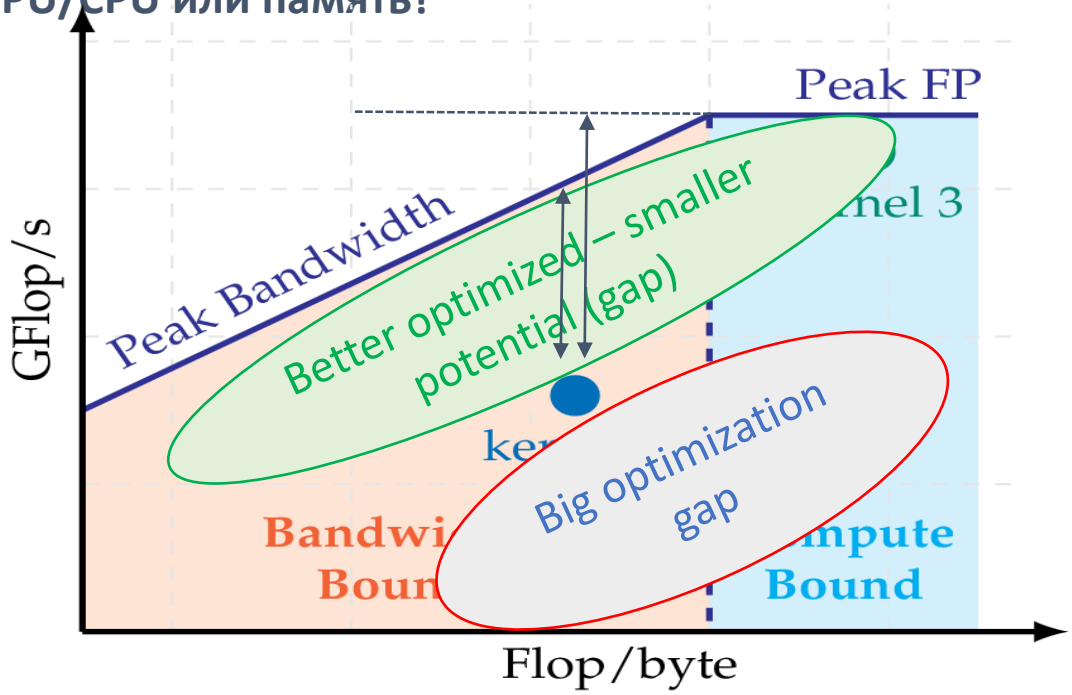
# Дополнительные материалы

- Intel® Advisor:
  - Страница продукта – обзоры, новинки, вопросы, поддержка …
  - Тренинги – фильмы, технические обзоры, документация…
  - Руководства – шаг за шагом через тернии к звездам
  - Обзоры
- Другие инструменты для анализа:
  - Intel® VTune Amplifier – профилировщик производительности
  - Intel® Inspector – исследователь потоков и памяти / отладчик
- Другие продукты для разработчиков:
  - Intel® Software Development Products

# Roofline модель: Что ограничивает производительность? VPU/CPU или память?



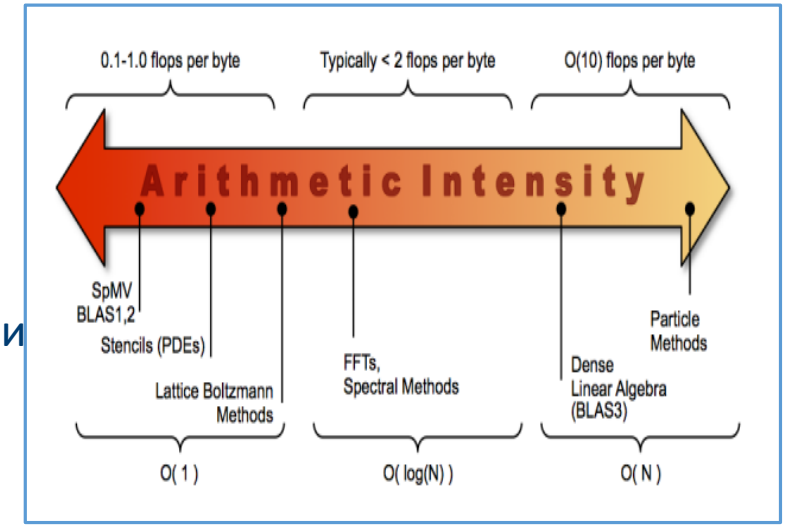

Что делает циклы **A, B, C различными**?



# **Roofline Модель**
# визуально интуитивная модель производительности

Берет в расчет

- Использование/спрос памяти
- CPU использование




Участвуют в анализе производительности и моделировании пространства совместно

**AI = # FLOPs / # BYTEs**

# Advisor Roofline: под капотом

Профилирование приложения:

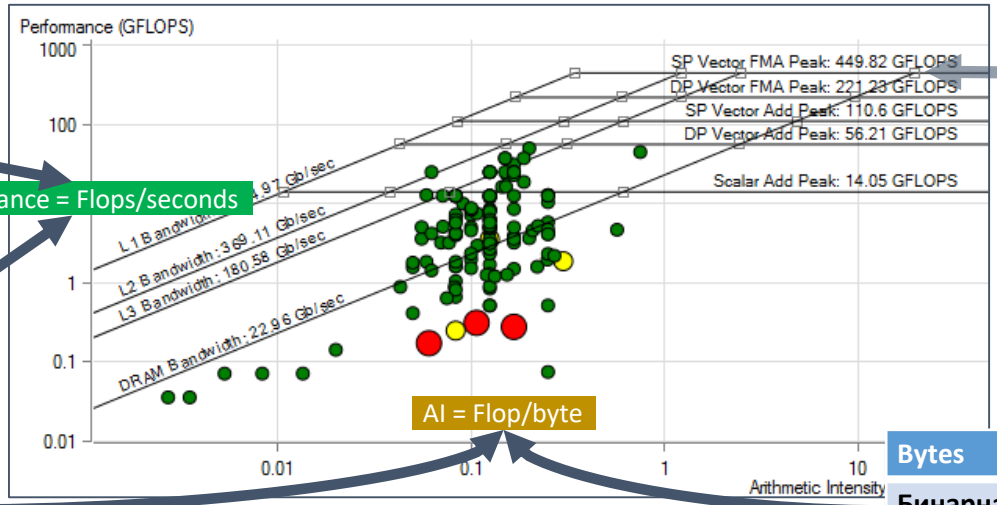
Axis Y: **FLOP/S = #FLOP / #Seconds**

Axis X: **AI = #FLOP / #Bytes**

**Seconds**

Сэмплирование уровня пользователя

*Root access not needed*




Performance = Flops/seconds

AI = Flop/byte

**Roofs**

**Микробенчмарки:**
Максимальная производительность для текущей архитектуры

**#FLOP**

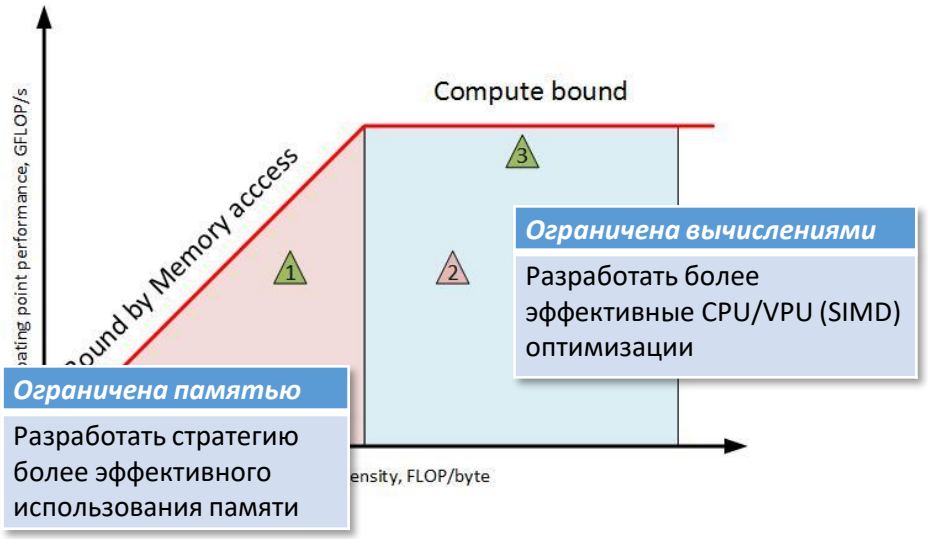
**Бинарная инструментация**
Не зависит от счетчиков CPU

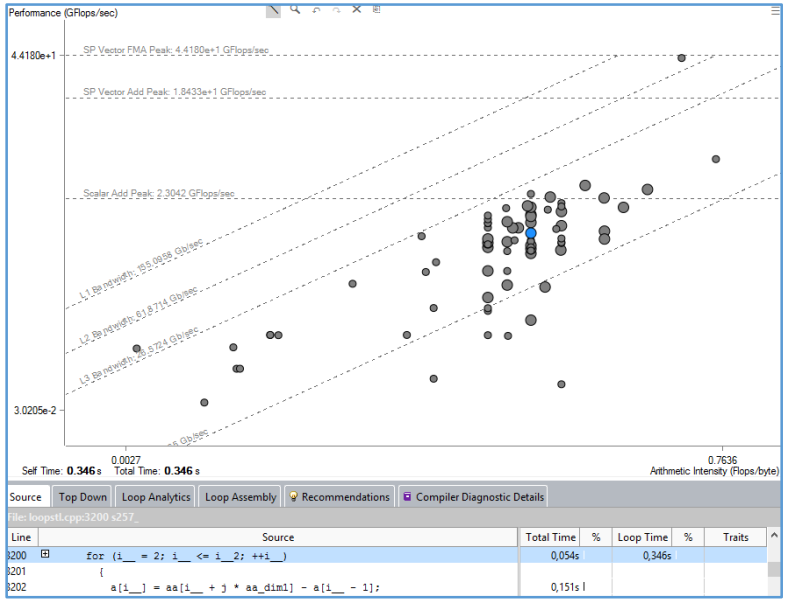
**Bytes**

**Бинарная инструментация**
Считает размер операндов (не кэшлинии)

# Интерпретация данных Roofline анализа.

**Разработка стратегии оптимизации**



# Дополнительные ссылки

Roofline model proposed by Williams, Waterman, Patterson:
http://www.eecs.berkeley.edu/~waterman/papers/roofline.pdf

"Cache-aware Roofline model: Upgrading the loft" (Ilic, Pratas, Sousa, INESC-ID/IST, Thec Uni of Lisbon) http://www.inesc-id.pt/ficheiros/publicacoes/9068.pdf

At Intel:

**Roman Belenov**, **Zakhar Matveev**, Julia Fedorova
SSG product teams, Hugh Caffey,
in collaboration with **Philippe Thierry**

# QA