



Разработка высокопроизводительных приложений
для архитектуры
Intel® Many Integrated Core

Валерий Черепенников,
Дмитрий Сергеев
Data Center and Cloud Engineering
Software & Services Group

Сентябрь
2011



Для чего нужен Intel® MIC?



Сочетание :

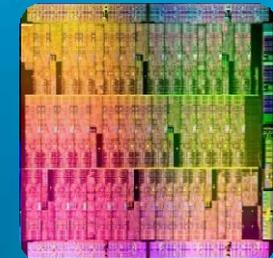
Преимущества программных моделей,
техник и инструментов для X86

+

Высокая плотность вычислительных
элементов, характерная для
акселераторов

=

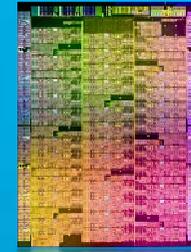
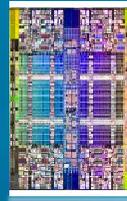
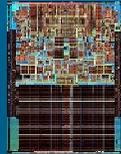
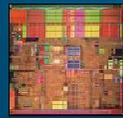
Intel® Many Integrated Core !



Resellers

Rock your sales.

Продукты Intel



Images not intended to reflect actual die sizes

	64-bit Intel® Xeon® processor	Intel® Xeon® processor 5100 series	Intel® Xeon® processor 5500 series	Intel® Xeon® processor 5600 series	Sandy Bridge	Knights Ferry
Frequency	3.6GHz	3.0GHz	3.2GHz	3.3GHz	Not Announced	1.2GHz
Core(s)	1	2	4	6	8	32
Thread(s)	2	2	8	12	16	128
SIMD Width	128 (2 clock)	128 (1 clock)	128 (1 clock)	128 (1 clock)	256 (1 clock)	512 (1 clock)

Resellers

developers of highly parallel applications

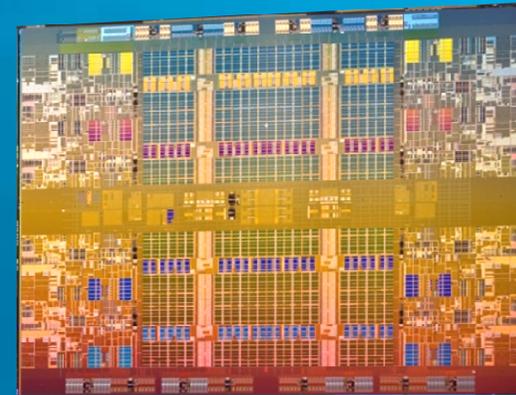
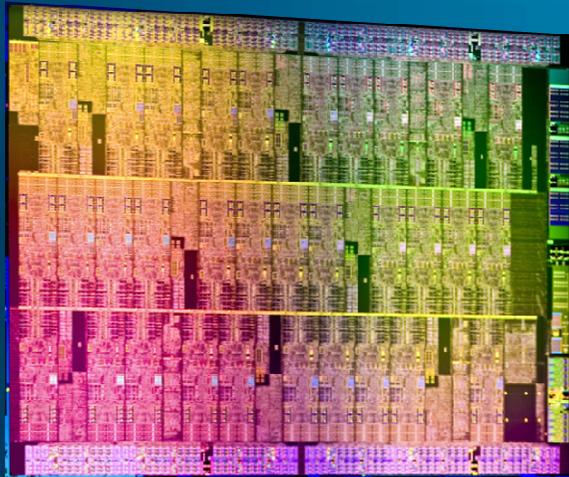
Rock your sales.

Many Core и Multi-Core



Many Integrated Core Knights Ferry at 1-1.2 GHz

Multi-core Intel® Xeon® processor at 2.26-3.5 GHz



- Каждое ядро Intel® MIC архитектуры меньше, экономичнее, имеет более низкую производительность
- Однако суммарная производительность Intel® MIC выше
- Приложения с высокой степенью параллелизма выигрывают от применения архитектуры Intel® MIC

Resellers

Rock your sales.

Сравнение Intel® Xeon® и Intel® MIC для финансовой математики

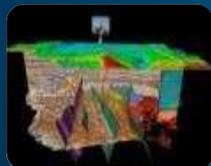


**Оптимизация кода для Intel® Xeon®
Добавление Intel® MIC увеличивает производительность**

Resellers

Rock your sales.

Массивно параллельные приложения



Energy & oil exploration



Digital content creation



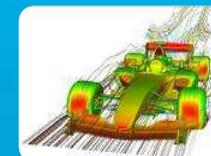
Climate modeling & weather prediction



Financial analyses, trading



Medical imaging and biophysics



Computer Aided Design & Manufacturing

- Сложные вычислительные проблемы, которые могут быть разбиты на части, выполняющиеся параллельно
- Параллельные приложения отличаются по гранулярности и программным моделям
- Массивно параллельные приложения встречаются везде! - workstation, HPC, Data Centers
- Примеры массивно параллельных вычислений: Vector Math, FFTs, Sparse and Dense Matrix Multiplication, Convolution, LU Factorization, Sort, Monte Carlo, Black-Scholes, итд.

Resellers

Rock your sales.

Семейство "Knights"



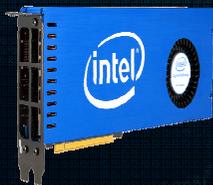
**Будущие
продукты
Knights**

Knights Corner

Первый продукт Intel® MIC
22nm процесс
>50 ядер X86
PCIe

Knights Ferry

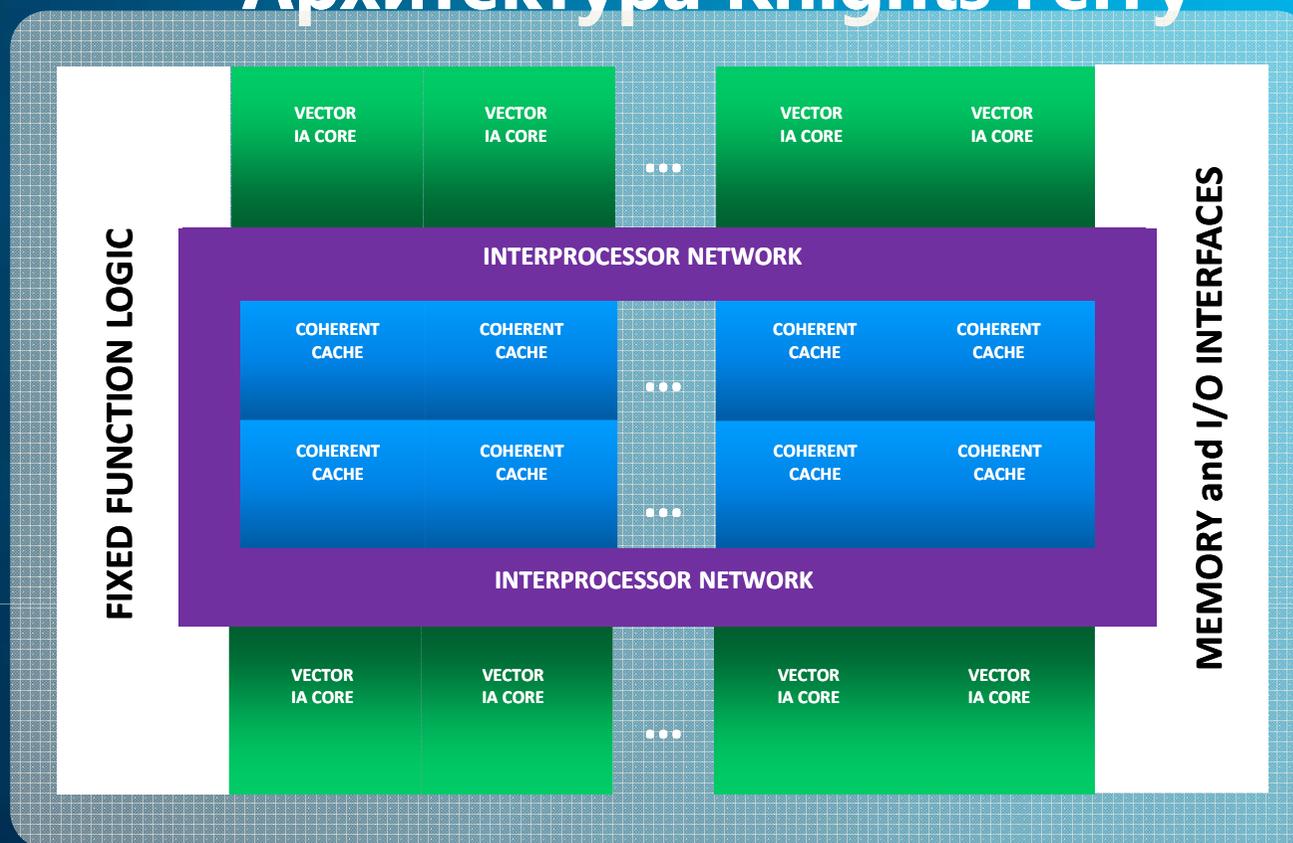
Software Development Platform



Resellers

Rock your sales.

Архитектура Knights Ferry



32 ядра x86

In-order

Короткий конвейер

Многопоточность

16 SP FP SIMD (512 бит)

Полный набор инструкций

Когерентный кэш

кольцевая шина 1024 бит

Память GDDR5

Поддержка виртуальной памяти

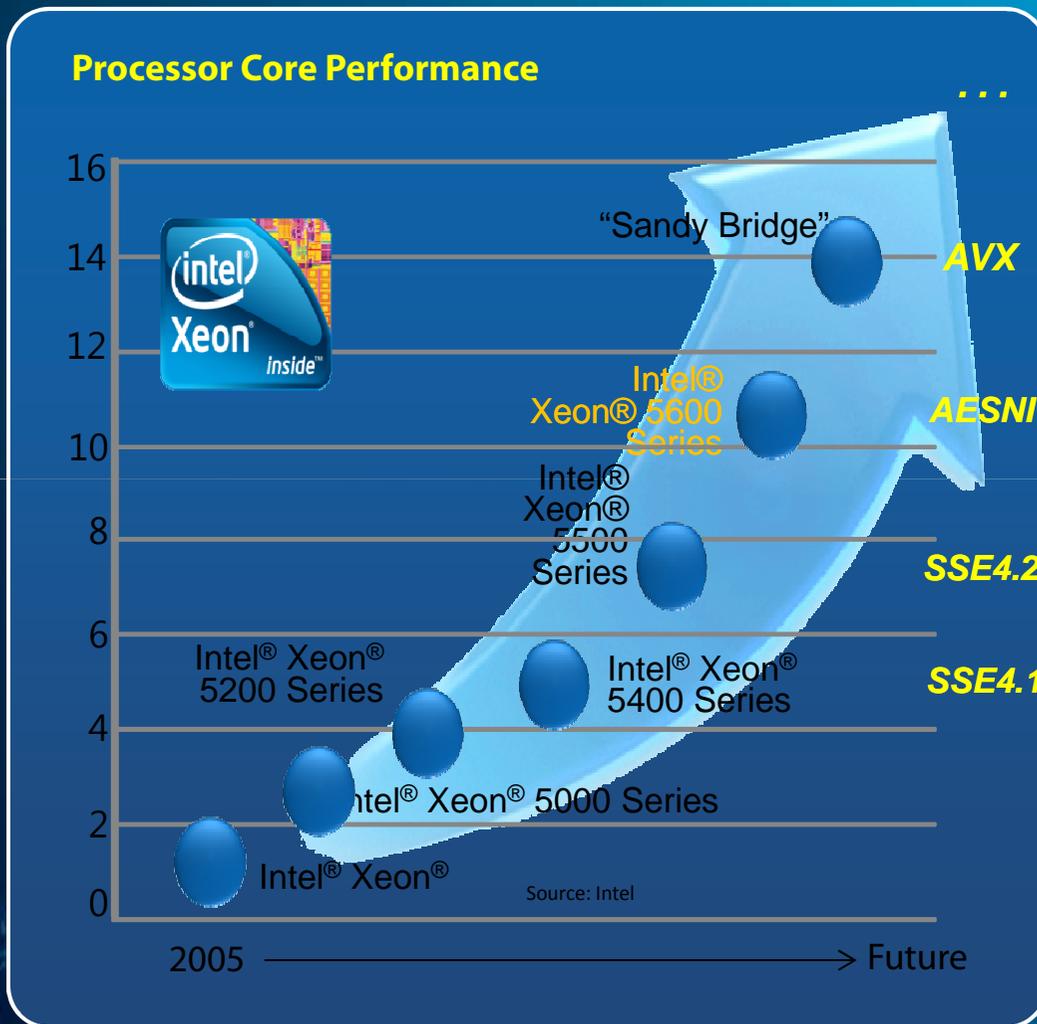
Стандартная программная модель Intel X86

Resellers

Rock your sales



Рост производительности процессоров



Общая производительность приложения

Возросшая производительность одного потока

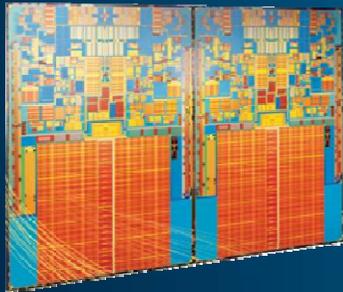
Ускорение Floating Point операций

Гибкий доступ к данным
Системный баланс

Resellers

Rock your sales.

Параллелизм на всех уровнях



Будущие процессоры должны поддерживать существующую базу приложений и минимизировать затраты на оптимизацию

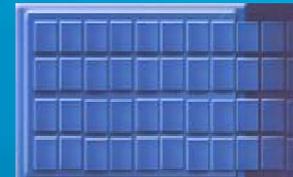
Resellers

Rock your sales.

Multi/Many-Core – индустриальный тренд



Энергоэффективные вычисления на
Архитектурах Multi/Many-Core



Many-Core



Multi-Core
(4+)



Dual-Core



Hyper-Threading



Multi Processor

Resellers

Rock your sales.

Модель программирования

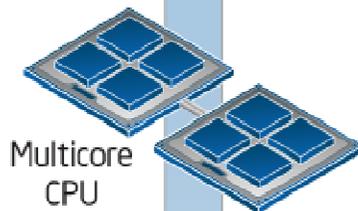


```
REAL SUM(1)
CALL SYNC_WAIT(1)
DO I=1,2000,1000
  IF (MOD(I,1000) .EQ. 0) THEN
    SUM = SUM + 1
  ENDIF
  CALL SYNC_WAIT(1)
ENDDO
```

Source

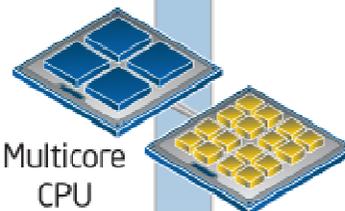
Compilers
Libraries
Parallel Models

Multicore



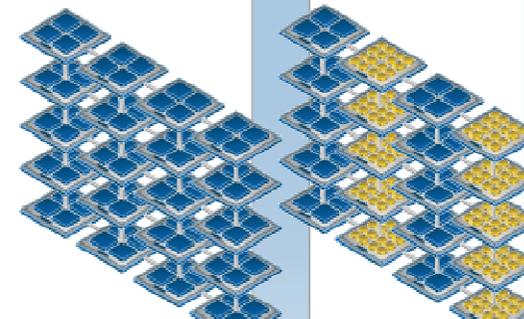
Multicore
CPU

Many-core



Multicore
CPU
Intel® MIC
Architecture
co-processor

Cluster



Multicore
Cluster
Multicore
& Many-core
Cluster

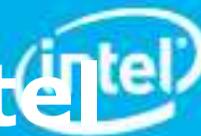
Common with
Intel® Xeon®
processors

Resellers

- Программные модели
- Компиляторы C/C++, Fortran
- Инструменты и библиотеки (MKL, IPP, TBB, ArBB, ...)
- Методы оптимизации с помощью инструментов Интел

Rock your sales.

Инструменты разработки Intel



Оптимизация

Компиляторы C++ и Fortran, библиотеки MKL/IPP, инструменты анализа производительности

Параллелизм

Intel MPI, Параллелизм в MKL, кластерные инструменты

Resellers

Rock your sales.

Технология Intel® Cilk™ Plus



simple keywords	Простая и эффективная система параллельных директив: <ul style="list-style-type: none">• cilk_for – распараллеливание циклов• cilk_spawn – старт параллельного исполнения• cilk_sync – конец параллельной секции
hyper-objects (reducers)	Защита shared переменных без барьеров и синхронизаций
array notation	Параллелизм по данным на уровне массивов и их секций
elemental functions	Поэлементная обработка данных в массивах



Пример кода

```
float a[N][M], b[N][M], c[N][M], d[N][M];

void calc() {
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < M; j++) {
            c[i][j] = a[i][j] + b[i][j];
            d[i][j] = sin(c[i][j]);
        }
    }
}
```

Intel® Cilk™ в действии: Многопоточность



```
float a[N][M], b[N][M], c[N][M], d[N][M];

void calc() {
    cilk_for (int i = 0; i < N; i++) {
        for (int j = 0; j < M; j++) {
            c[i][j] = a[i][j] + b[i][j];
            d[i][j] = sin(c[i][j]);
        }
    }
}
```

Resellers

* Other brands and names are the property of their respective owners

Rock your sales.

Intel® Cilk™ Plus в действии:

Векторизация



```
float a[N][M], b[N][M], c[N][M], d[N][M];

void calc() {
    cilk_for (int i = 0; i < N; i++) {
        // array notation
        c[i][:] = a[i][:] + b[i][:];
        // elemental function
        d[i][:] = sin(c[i][:]);
    }
}
```

Resellers

* Other brands and names are the property of their respective owners. **Rock your sales.**



MIC как акселератор

CPU

C/C++, Fortran

MKL

OpenMP

Cilk Plus

TBB

ArBB

MIC

C/C++, Fortran

MKL

OpenMP

Cilk Plus

TBB

ArBB

Offloading Extensions

Resellers

* Other brands and names are the property of their respective owners

Rock your sales.

Offload на виртуальной Shared Memory



- Одинаковые виртуальные адреса для IA процессора и MIC
 - Комбинация runtime библиотеки и компилятора
 - Автоматическая передача данных между процессором и MIC

Intel® Cilk™ Plus В сочетании с Offload



```
_Shared float a[N][M], b[N][M], c[N][M], d[N][M];  
  
_Shared void calc() {  
    cilk_for (int i = 0; i < N; i++) {  
        // array notation  
        c[i][:] = a[i][:] + b[i][:];  
        // elemental function  
        d[i][:] = sin(c[i][:]);  
    }  
}
```

```
.....  
_Offload calc();
```

Resellers

* Other brands and names are the property of their respective owners

Rock your sales.

Модели программирования с использованием MPI

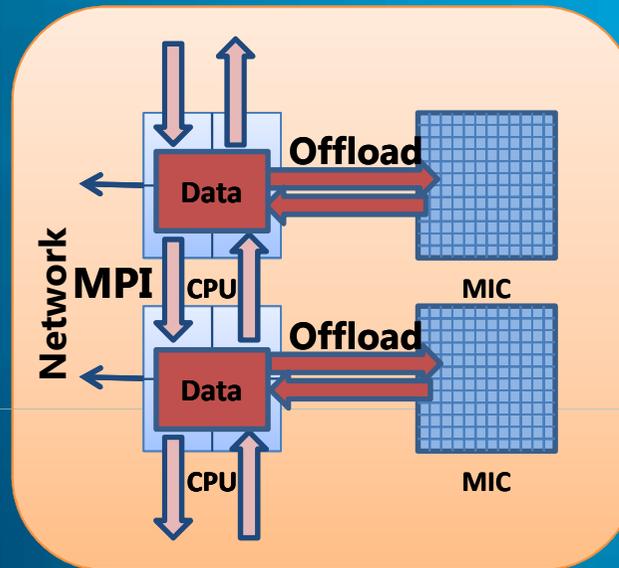


Offload	MIC-only	MIC-hosted	Symmetric
<ul style="list-style-type: none">• MPI процессы только на Xeon• Сообщения между Xeon• MIC как сопроцессор	<ul style="list-style-type: none">• MPI процессы на одной MIC - карте• Запуск на MIC	<ul style="list-style-type: none">• MPI процессы на нескольких картах MIC• Сообщения через Xeon	<ul style="list-style-type: none">• MPI процессы на MIC и Xeon• Сообщения между любыми ядрами

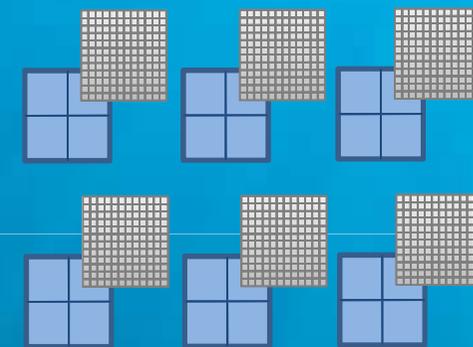
Модель Offload (1/2)



- MPI процессы только на Xeon
- Все сообщения только между процессорами
- Offload используется для ускорения MPI процессов
- PBB, OpenMP, pthreads используются внутри процесса (MIC)



Однородная сеть
неоднородных
вычислителей



Начало использования MIC в HPC

- Простой переход к гетерогенным вычислениям
- Ускорение внутри одного процесса



Модель Offload (2/2)

```
// MPI process
// C/C++ Offload Pragma
#pragma offload target(mic)
#pragma omp parallel for reduction(+:pi)
for (i=0; i<count; i++) {
    float t = (float)((i+0.5)/count);
    pi += 4.0/(1.0+t*t);
}
pi /= count
```

```
! MPI process
! Fortran Offload Directive
!dir$ omp offload target(mic)
!$omp parallel do
    do i=1,10
        A(i) = B(i) * C(i)
    enddo
!$omp end parallel
```

```
// MPI process
// MKL Offload Example
#pragma offload target(mic)
in(transa, transb, N, alpha, beta) \
in(A:length(matrix_elements)) \
in(B:length(matrix_elements)) \
in(C:length(matrix_elements)) \
out(C:length(matrix_elements)alloc_if(0))
sgemm(&transa, &transb, &N, &N, &N, &alpha, A, &N,
      B, &N, &beta, C, &N);
```

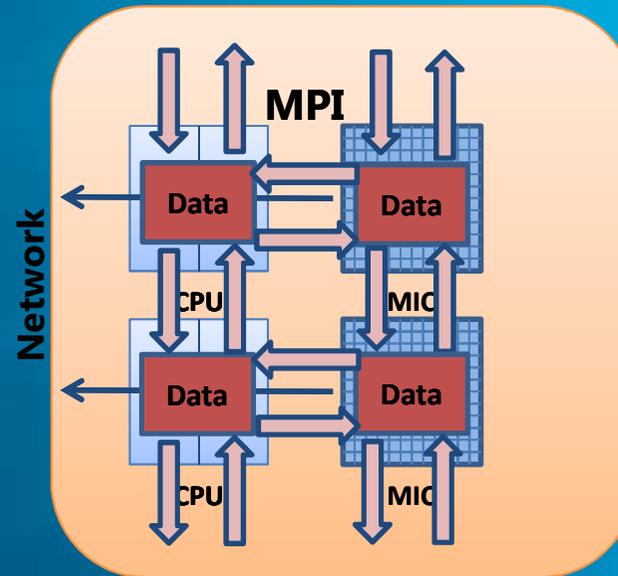
```
// C/C++ Offload Pragma
#pragma offload target(mic)
#pragma omp parallel for
for (i=0; i<count; i++) {
    <...
    MPI
    >...>
}
<>
```

MPI calls can't be offloaded

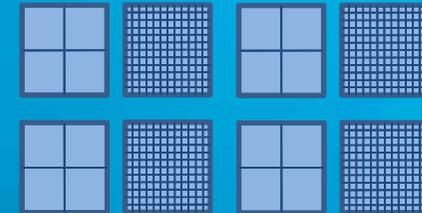
Симметричная программная модель



- MPI процессы на MIC и Xeon
- Сообщения между любыми ядрами
- PWB, OpenMP, pthreads могут быть использованы



Неоднородная сеть



```
$ mpiicc -mic app.c -o app
```

Построение исполняемых модулей с использованием Intel64 и MIC компиляторов

```
$ mpiexec.hydra -mic -n 40 -f hostfile app
```

Запуск на 40 узлах на различных "Xeon+MIC" узлах. Загружает исполняемые модули на MIC

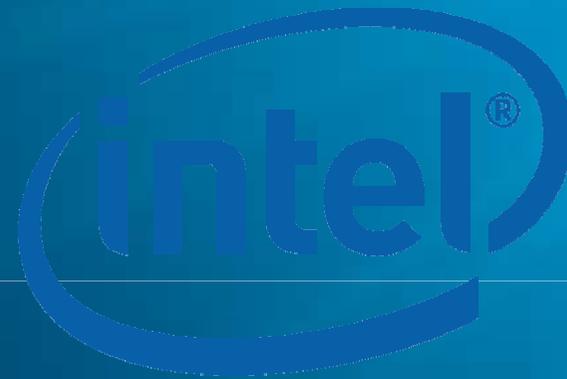
Заключительные замечания



- Тенденция роста числа ядер на кристалле – 4S x 8, 8S x 8, 32 (KNF), >50 (KNC)...
- Расширение регистров SIMD – 128, 256, 512...
- Инструментарий остается тем же Intel C/C++/Fortran компиляторы, библиотеки, Intel® MPI
- Изменение подхода к программированию?
- Будьте готовы к MIC!

Resellers

Rock your sales.



Software

Resellers

Rock your sales.

Legal Disclaimer



Notice: This document contains information on products in the design phase of development. The information here is subject to change without notice. Do not finalize a design with this information. Contact your local Intel sales office or your distributor to obtain the latest specification before placing your product order.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY RELATING TO SALE AND/OR USE OF INTEL PRODUCTS, INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT. Intel products are not intended for use in medical, life saving, or life sustaining applications. Intel may make changes to specifications, product descriptions, and plans at any time, without notice.

All products, dates, and figures are preliminary for planning purposes and are subject to change without notice.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance.

The Intel products discussed herein may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Knights Corner, Knights Ferry, Aubrey Isle and other code names featured are used internally within Intel to identify products that are in development and not yet publicly announced for release. Customers, licensees and other third parties are not authorized by Intel to use code names in advertising, promotion or marketing of any product or services and any such use of Intel's internal code names is at the sole risk of the user.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or by visiting Intel's website at <http://www.intel.com>.

Intel, Xeon, Xeon Inside, Pentium and the Intel logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Copyright © 2011, Intel Corporation. All rights reserved.

**Other names and brands may be claimed as the property of others.*

Resellers

Rock your sales.

Optimization Notice – Please Read

Intel® Compiler includes compiler options that optimize for instruction sets that are available in both Intel® and non-Intel microprocessors (for example SIMD instruction sets), but do not optimize equally for non-Intel microprocessors. In addition, certain compiler options for Intel® Compiler are reserved for Intel microprocessors. For a detailed description of these compiler options, including the instruction sets they implicate, please refer to "Intel® Compiler User and Reference Guides > Compiler Options." Many library routines that are part of Intel® Compiler are more highly optimized for Intel microprocessors than for other microprocessors. While the compilers and libraries in Intel® Compiler offer optimizations for both Intel and Intel-compatible microprocessors, depending on the options you select, your code and other factors, you likely will get extra performance on Intel microprocessors.

While the paragraph above describes the basic optimization approach for Intel® Compiler, with respect to Intel's compilers and associated libraries as a whole, Intel® Compiler may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include Intel® Streaming SIMD Extensions 2 (Intel® SSE2), Intel® Streaming SIMD Extensions 3 (Intel® SSE3), and Supplemental Streaming SIMD Extensions 3 (Intel® SSSE3) instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors.

Intel recommends that you evaluate other compilers to determine which best meet your requirements.

Resellers

Rock your sales.