

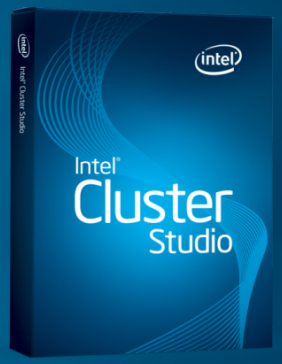
Efficient
Performance



Essential
Performance



Advanced
Performance

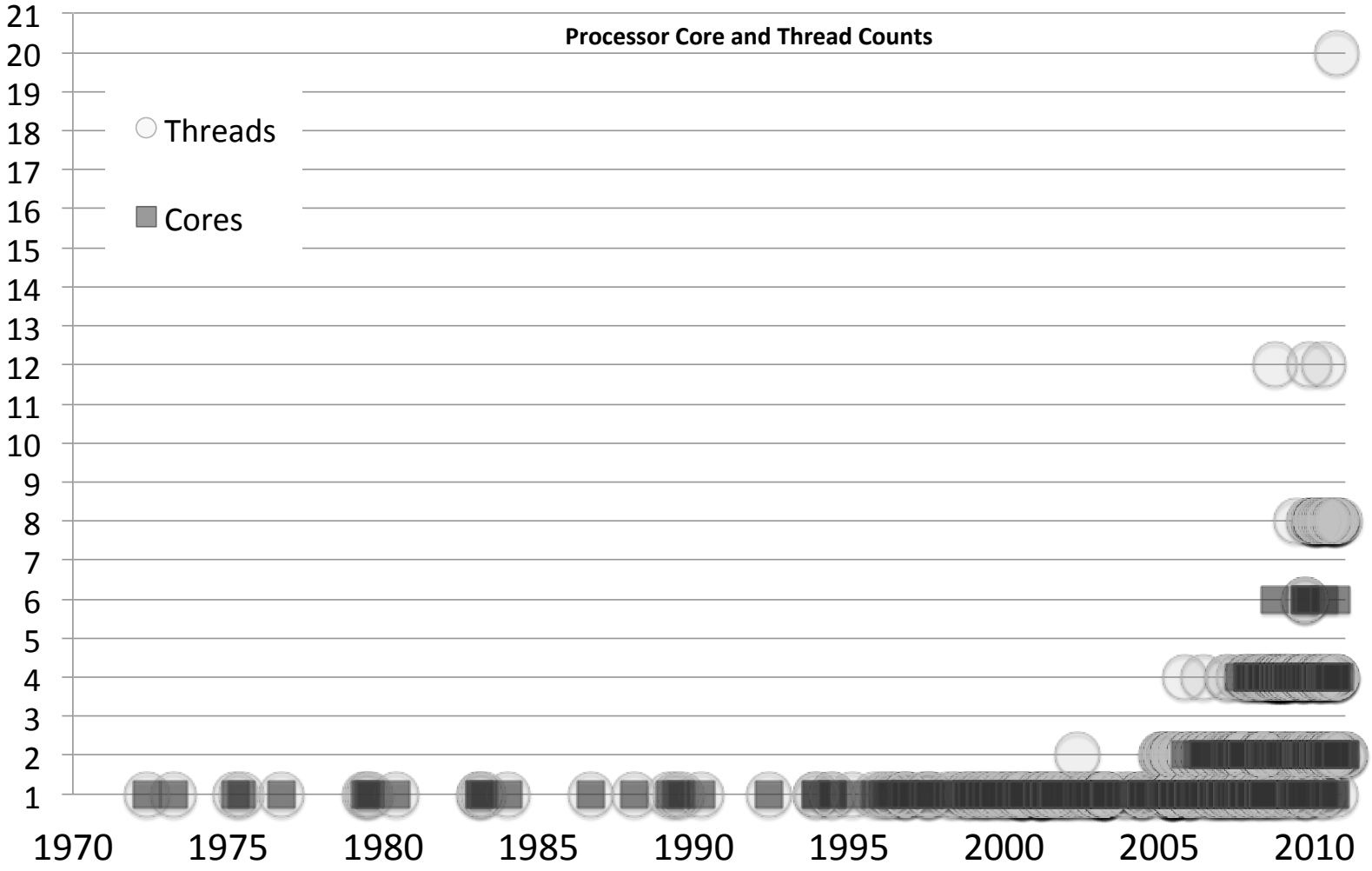


Distributed
Performance

Introduction to Parallel Programming with Intel® Cilk™ Plus

Michael McCool
Intel, Software and Services Group, Principal Engineer
Tokyo Demo Fest, January 14, 2012

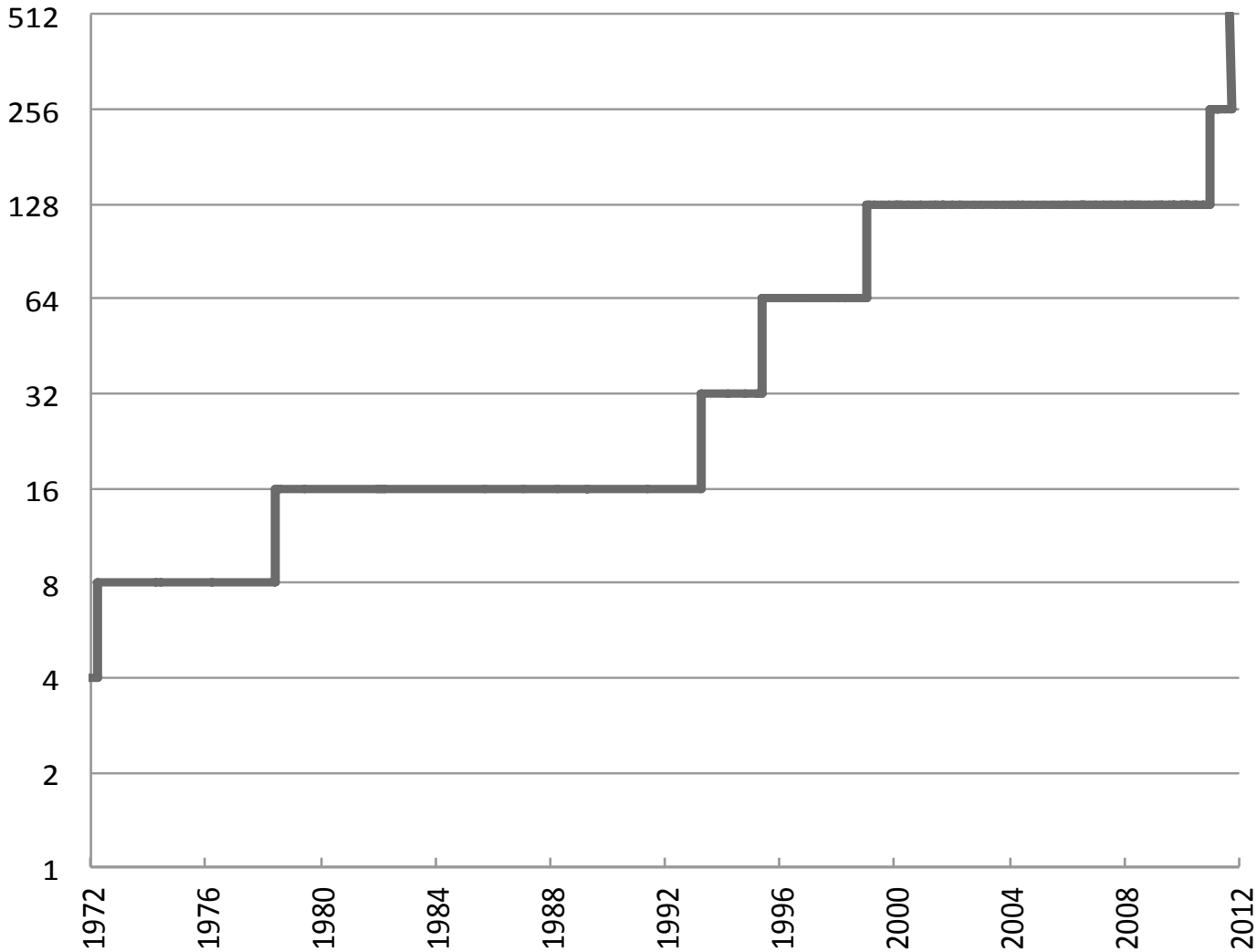
Hardware Parallelism: Cores and Threads



source: James Reinders, Intel

Copyright© 2012 Intel Corporation. All rights reserved.
*Other brands and names are the property of their respective owners.
Optimization Notice

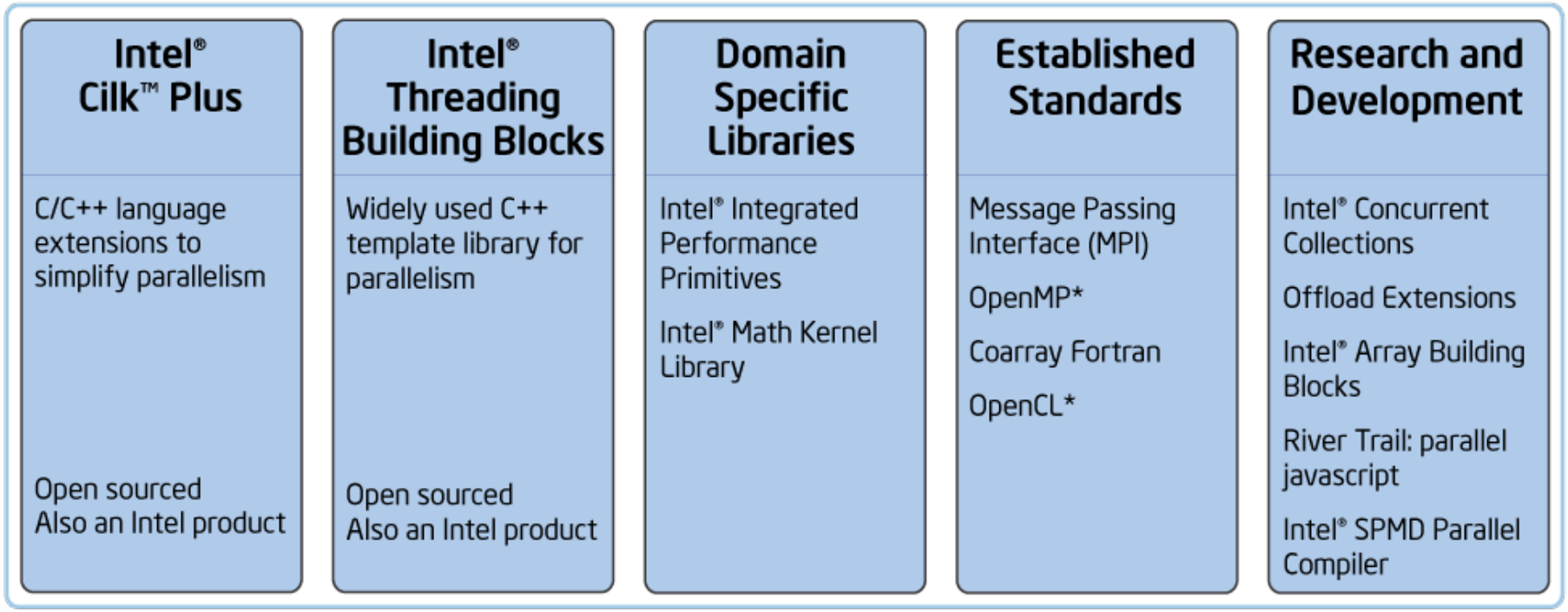
Hardware Parallelism: Register Width



source: James Reinders, Intel

Copyright© 2012 Intel Corporation. All rights reserved.
*Other brands and names are the property of their respective owners.
Optimization Notice

Intel's Parallel Programming Models



Choice of high-performance parallel programming models

- Libraries for pre-optimized and parallelized functionality
- Intel® Cilk™ Plus and Intel® Threading Building Blocks supports composable parallelization of a wide variety of applications.
- OpenCL* addresses the needs of customers in specific segments, and provides developers an additional choice to maximize their app performance
- MPI supports distributed computation, combines with other models on nodes

Copyright© 2012 Intel Corporation. All rights reserved.
*Other brands and names are the property of their respective owners.
Optimization Notice

Summary of Cilk™ Plus

Thread Parallelism
cilk_spawn
cilk_sync
cilk_for

Vector Parallelism
array notation
#pragma simd
elemental functions

Reducers
reducer
reducer_op{add,and,or,xor}
reducer_{min,max}{_index}
reducer_list
reducer_ostream
reducer_string

Copyright© 2012 Intel Corporation. All rights reserved.
*Other brands and names are the property of their respective owners.
Optimization Notice

TBB 4.0 Components

Parallel Algorithms
parallel_for
parallel_for_each
parallel_invoke
parallel_do
parallel_scan
parallel_sort
parallel_[deterministic]_reduce

Macro Dataflow
parallel_pipeline
tbb::flow::...

Concurrent Containers
concurrent_hash_map
concurrent_unordered_{map,set}
concurrent_[bounded_]queue
concurrent_priority_queue
concurrent_vector

Task scheduler
task_group, structured_task_group
task
task_scheduler_init
task_scheduler_observer

Thread Local Storage
combinable
enumerable_thread_specific

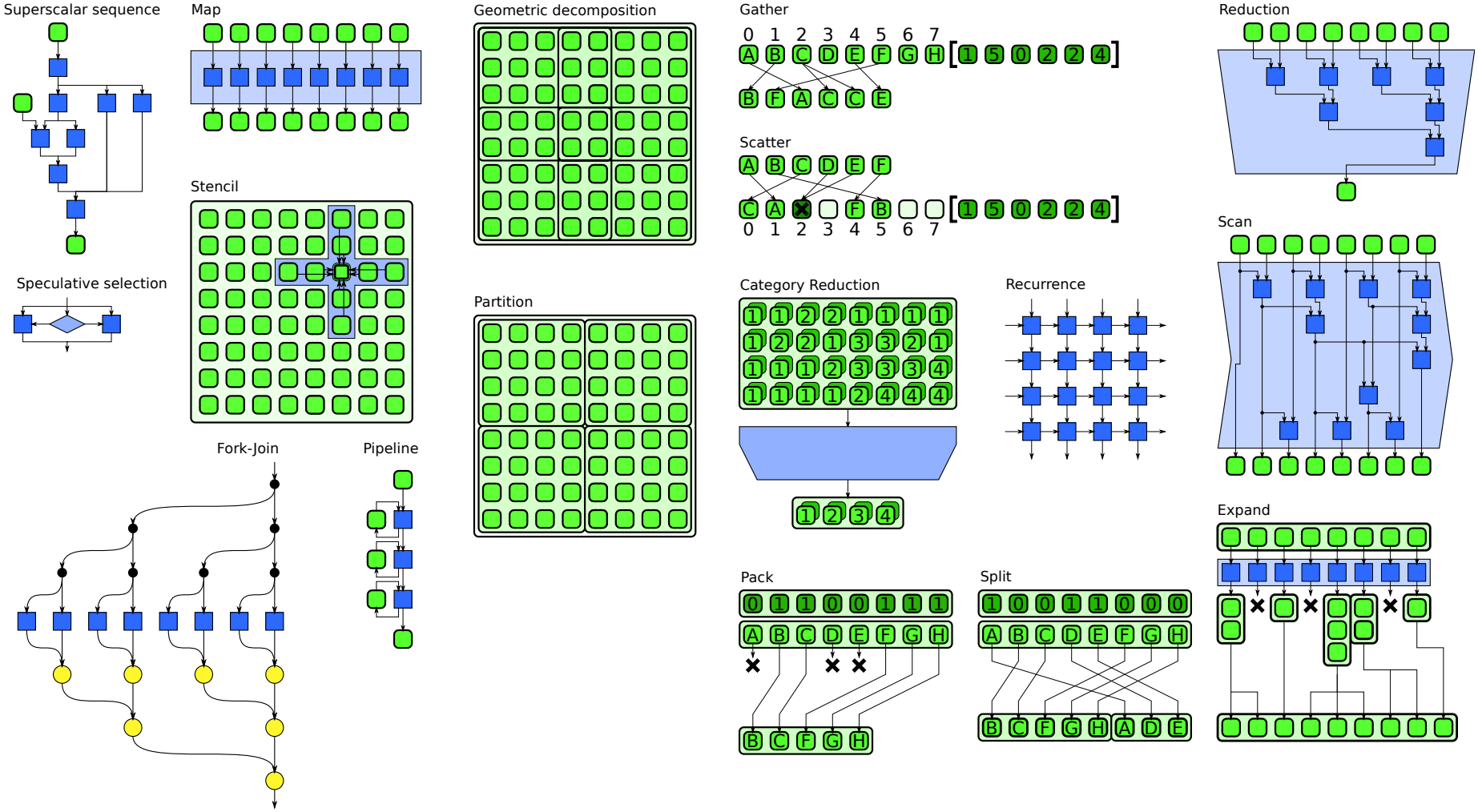
Synchronization Primitives
atomic, condition_variable
[recursive_]mutex
{spin,queuing,null} [_rw]_mutex
critical_section, reader_writer_lock

Memory Allocation
tbb_allocator
zero_allocator
cache_aligned_allocator
scalable_allocator

Threads
std::thread

Copyright© 2012 Intel Corporation. All rights reserved.
*Other brands and names are the property of their respective owners.
Optimization Notice

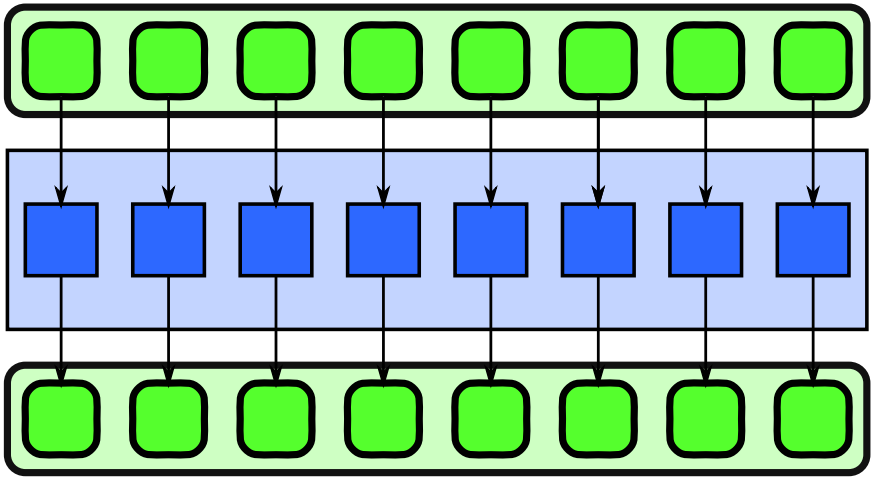
Structured Parallel Patterns: Overview



Structured Parallel Programming, McCool, Robison, Reinders, MK, 2012

Copyright© 2012 Intel Corporation. All rights reserved.
*Other brands and names are the property of their respective owners.
Optimization Notice

Map



- ***Replicates a function*** over every element of an index set
- The index set may be abstract or associated with the elements of an array.

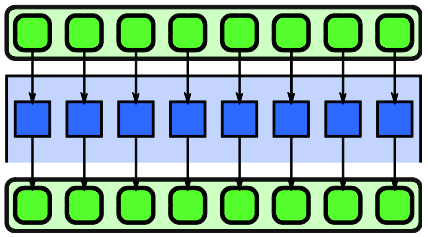
```
A = map(f)(B);
```

- Map replaces usage of iteration for independent operations.

Examples: gamma correction and threshholding in images; color space conversions; Monte Carlo sampling; ray tracing.

Copyright© 2012 Intel Corporation. All rights reserved.
*Other brands and names are the property of their respective owners.
Optimization Notice

Map in Cilk Plus and TBB



Intel® Cilk™ Plus

```
cilk_for( int i=0; i<n; ++i )
    a[i] = f(b[i]);
```

Intel®TBB

```
parallel_for( 0, n, [&]( int i ) {
    a[i] = f(b[i]);
});
```

Thread parallelism

```
parallel_for(
    blocked_range<int>(0,n),
    [&](blocked_range<int> r ) {
        for( int i=r.begin(); i!=r.end(); ++i )
            a[i] = f(b[i]);
    });
```

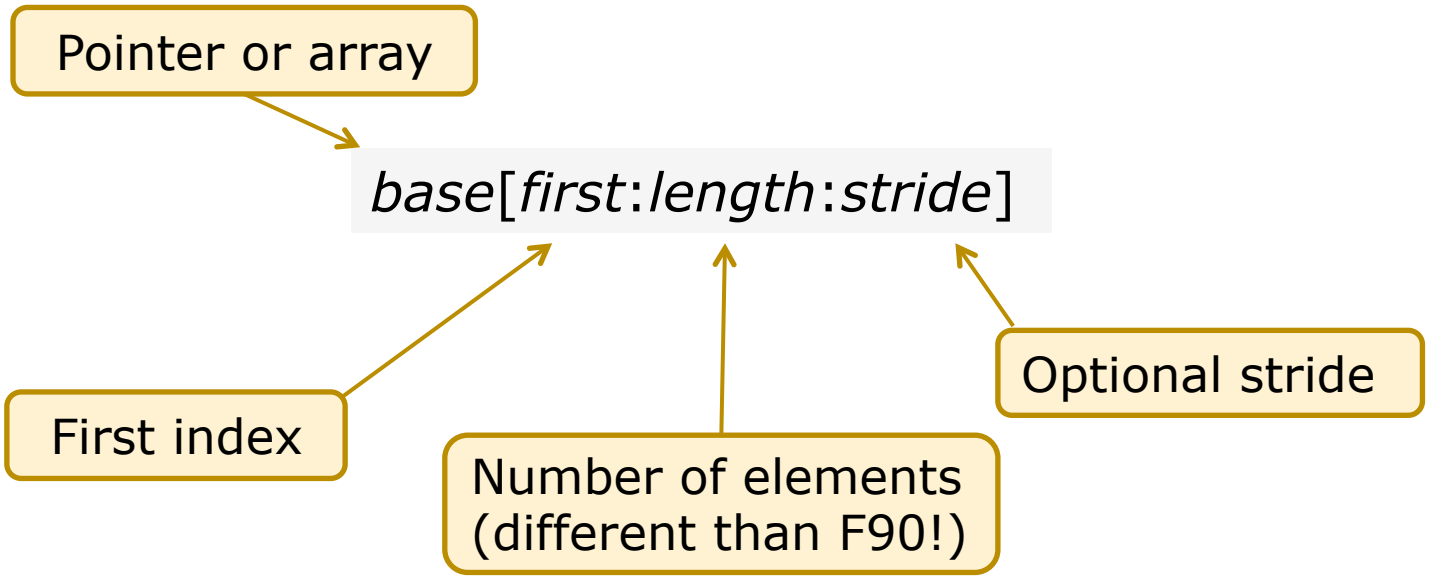
```
a[0:n] = f(b[i:n]);
```

```
#pragma simd
for( int i=0; i<n; ++i )
    a[i] = f(b[i]);
```

Vector parallelism

Copyright© 2012 Intel Corporation. All rights reserved.
*Other brands and names are the property of their respective owners.
Optimization Notice

Array Notation: Sections



Rules for *section*$_1$ *op section*$_2$

- Elementwise application of *op*
- Also works for *func*(*section*$_1$, *section*$_2$)
- Sections must be the same length
- Scalar arguments implicitly extended

Copyright© 2012 Intel Corporation. All rights reserved.
*Other brands and names are the property of their respective owners.
Optimization Notice

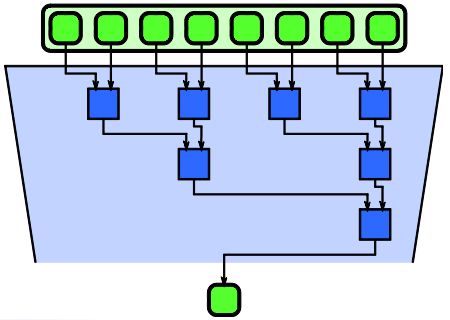
Array Notation: Map Pattern

- Lets you specify parallel *intent*
  - Gives license to the compiler to vectorize

```
// Set y[i] ← y[i] + a ·x[i] for i∈[0..n)
void saxpy(float a, float x[], float y[], size_t n ) {
    y[0:n] += a*x[0:n];
}
```


Copyright© 2012 Intel Corporation. All rights reserved.
*Other brands and names are the property of their respective owners.
Optimization Notice

Reduction



Intel® Cilk™ Plus

```
cilk::reducer_opadd<float> sum = 0;
cilk_for( int i=0; i<n; ++i )
   sum += a[i];
... = sum.get_value();
```

Intel®TBB

```
enumerable_thread_specific<float> sum;
parallel_for( 0, n, [&]( int i ) {
   sum.local() += a[i];
});
... = sum.combine(std::plus<float>());
```

Thread parallelism

```
float sum =
   __sec_reduce_add(a[i:n]);
```

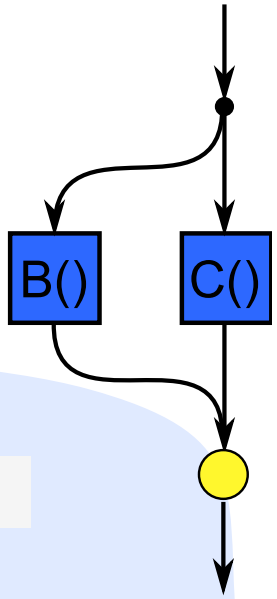
```
#pragma simd reduction(+:sum)
float sum=0;
for( int i=0; i<n; ++i )
   sum += a[i];
```

Vector parallelism

```
sum = parallel_reduce(
   blocked_range<int>(0,n),
   0.f,
   [&](blocked_range<int> r, float s) -> float
   {
      for( int i=r.begin(); i!=r.end(); ++i )
         s +=  a[i];
      return s;
   },
   std::plus<float>()
);
```


Copyright© 2012 Intel Corporation. All rights reserved.
*Other brands and names are the property of their respective owners.
Optimization Notice

Fork-Join Pattern



Intel® Cilk™ Plus

```
cilk_spawn a();
cilk_spawn b();
c();
cilk_sync();
```

Intel®TBB

```
parallel_invoke( a, b, c );
```

Thread parallelism

```
task_group g;
g.run( a );
g.run( b );
g.run( c );
g.wait();
```

```
if( x[0:n] < 0 )
    x[0:n] = -x[0:n];
```

```
#pragma simd
for( int i=0; i<n; ++i )
   if( x[i] < 0 )
      x[i] = -x[i];
```

Fake Fork-Join for Vector parallelism

Copyright© 2012 Intel Corporation. All rights reserved.
*Other brands and names are the property of their respective owners.
Optimization Notice

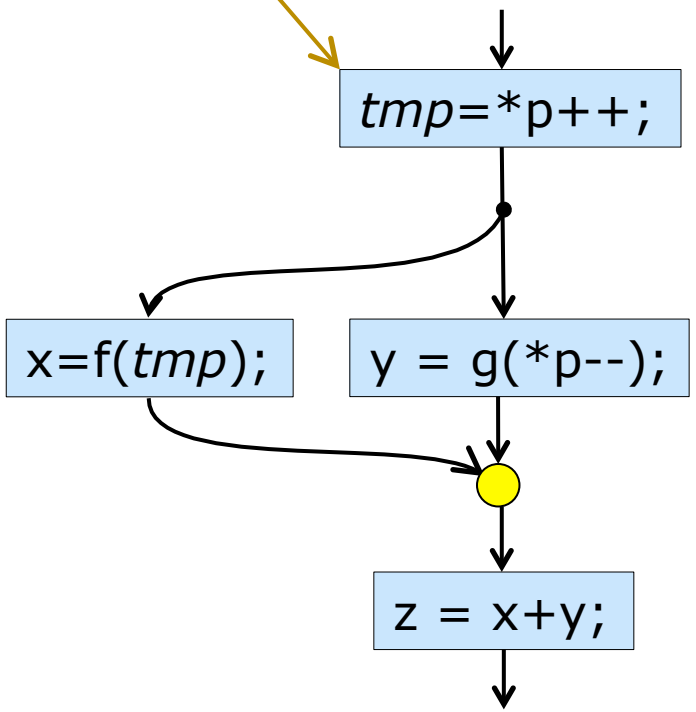
Fork-Join in Cilk Plus

- spawn = asynchronous function call

Optional assignment

```
x = cilk_spawn f(*p++);
y = g(*p--);
cilk_sync;
z = x+y;
```

Arguments to spawned function evaluated *before* fork.

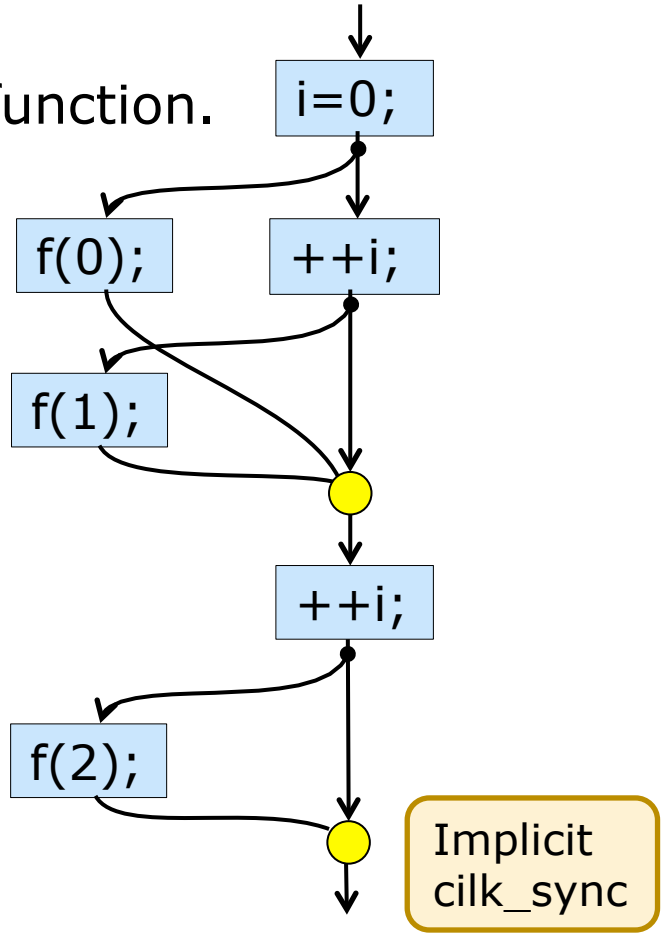


Copyright© 2012 Intel Corporation. All rights reserved.
*Other brands and names are the property of their respective owners.
Optimization Notice

cilk_sync Has Function Scope

- Scope of **cilk_sync** is entire function.
- There is implicit **cilk_sync** at end of a function.

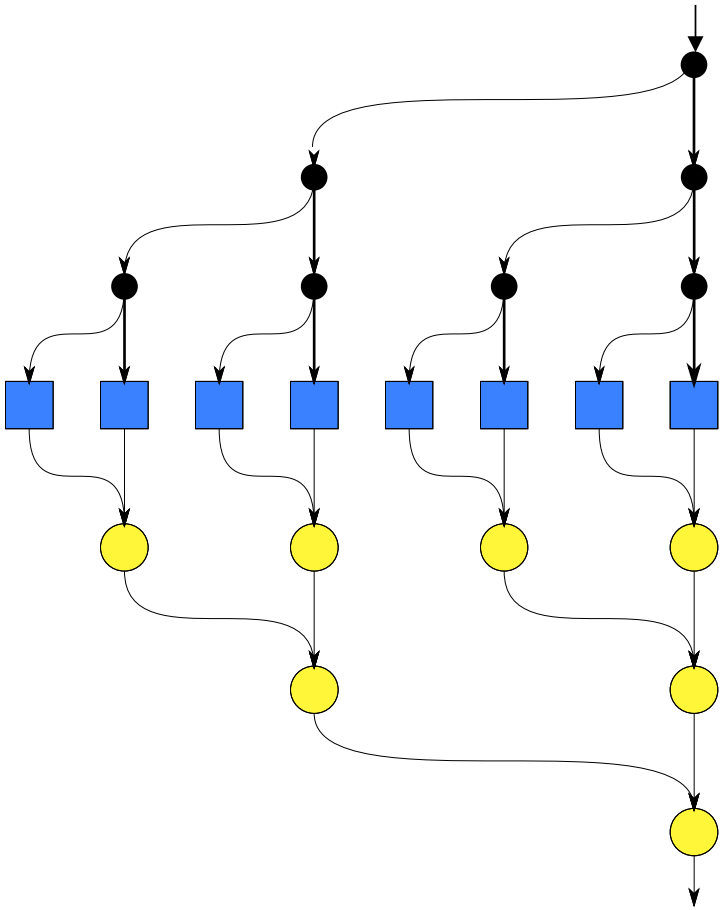
```
void bar() {
    for( int i=0; i<3; ++i ) {
        cilk_spawn f(i);
        if( i&1 ) cilk_sync;
    }
    // implicit cilk_sync
}
```



Serial call/return property:
All Cilk Plus parallelism created by a function completes before it returns.

Copyright© 2012 Intel Corporation. All rights reserved.
*Other brands and names are the property of their respective owners.
Optimization Notice

Fork-Join: Nesting Parallelism and Map



- Fork-join can be nested
- Spreads cost of work distribution and synchronization.
- This is how **cilk_for** and **tbb::parallel_for** are implemented.

Recursive fork-join enables high parallelism.

Copyright© 2012 Intel Corporation. All rights reserved.
*Other brands and names are the property of their respective owners.
Optimization Notice

Summary

- TBB and Cilk™ Plus can express thread parallelism
  - *Programmer specifies tasks, not threads*
  - Threads: mandatory parallelism; mechanism
  - Tasks: potential (optional) parallelism; abstraction
- Cilk™ Plus can explicitly express vector parallelism
  - Array notation, simd pragma, elemental functions
  - Works on both SSE and AVX without code changes
- Cilk™ Plus available as
  - Intel product (is in Intel® C/C++ Compiler)
  - Work is ongoing to build Cilk™ Plus into gcc
- Can try Intel® Compiler version for free:
  - http://www.intel.com/go/parallel
  - Click on "Try" link for Intel® Parallel Studio

Copyright© 2012 Intel Corporation. All rights reserved.
*Other brands and names are the property of their respective owners.
Optimization Notice



Copyright© 2012 Intel Corporation. All rights reserved.
*Other brands and names are the property of their respective owners.

最適化に関する注意事項

インテル® コンパイラーは、互換マイクロプロセッサー向けには、インテル製マイクロプロセッサー向けと同等レベルの最適化が行われない可能性があります。これには、インテル® ストリーミング SIMD 拡張命令 2 (インテル® SSE2)、インテル® ストリーミング SIMD 拡張命令 3 (インテル® SSE3)、ストリーミング SIMD 拡張命令 3 補足命令 (SSSE3) 命令セットに関連する最適化およびその他の最適化が含まれます。インテルでは、インテル製ではないマイクロプロセッサーに対して、最適化の提供、機能、効果を保証していません。本製品のマイクロプロセッサー固有の最適化は、インテル製マイクロプロセッサーでの使用を目的としています。インテル® マイクロアーキテクチャーに非固有の特定の最適化は、インテル製マイクロプロセッサー向けに予約されています。この注意事項の適用対象である特定の命令セットの詳細は、該当する製品のユーザー・リファレンス・ガイドを参照してください。

改訂 #20110804

Copyright© 2012 Intel Corporation. All rights reserved.
*Other brands and names are the property of their respective owners.

著作権と商標について

本資料の情報は、現状のまま提供されます。本資料は、明示されているか否かにかかわらず、また禁反言によるとよらずにかかわらず、いかなる知的財産権のライセンスも許諾するものではありません。製品に付属の売買契約書『Intel's Terms and Conditions of Sale』に規定されている場合を除き、インテルはいかなる責任を負うものではなく、またインテル製品の販売や使用に関する明示または黙示の保証（特定目的への適合性、商品性に関する保証、第三者の特許権、著作権、その他、知的財産権の侵害への保証を含む）をするものではありません。

性能に関するテストや評価は、特定のコンピューター・システム、コンポーネント、またはそれらを組み合わせて行ったものであり、このテストによるインテル製品の性能の概算の値を表しているものです。システム・ハードウェアの設計、ソフトウェア、構成などの違いにより、実際の性能は掲載された性能テストや評価とは異なる場合があります。システムやコンポーネントの購入を検討される場合は、ほかの情報も参考にして、パフォーマンスを総合的に評価することをお勧めします。インテル製品の性能評価についてさらに詳しい情報をお知りになりたい場合は、
http://www.intel.co.jp/jp/performance/resources/benchmark_limitations.htm を参照してください。

Intel、インテル、Intel ロゴ、Cilk は、アメリカ合衆国およびその他の国における Intel Corporation の商標です。

* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

© 2011 Intel Corporation.

http://www.intel.co.jp/jp/software/products/

Copyright© 2012 Intel Corporation. All rights reserved.
*Other brands and names are the property of their respective owners.

Optimization Notice

Optimization Notice

Intel's compilers 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 SSE2®, SSE3, and 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. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804

Copyright© 2012 Intel Corporation. All rights reserved.
*Other brands and names are the property of their respective owners.

Legal Disclaimer

INFORMATION IN THIS DOCUMENT IS PROVIDED "AS IS". NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO THIS INFORMATION INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

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. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, reference www.intel.com/software/products.

BunnyPeople, Celeron, Celeron Inside, Centrino, Centrino Atom, Centrino Atom Inside, Centrino Inside, Centrino logo, Cilk, Core Inside, FlashFile, i960, InstantIP, Intel, the Intel logo, Intel386, Intel486, IntelDX2, IntelDX4, IntelSX2, Intel Atom, Intel Atom Inside, Intel Core, Intel Inside, Intel Inside logo, Intel. Leap ahead., Intel. Leap ahead. logo, Intel NetBurst, Intel NetMerge, Intel NetStructure, Intel SingleDriver, Intel SpeedStep, Intel StrataFlash, Intel Viiv, Intel vPro, Intel XScale, Itanium, Itanium Inside, MCS, MMX, Oplus, OverDrive, PDCharm, Pentium, Pentium Inside, skoool, Sound Mark, The Journey Inside, Viiv Inside, vPro Inside, VTune, Xeon, and Xeon Inside are trademarks of Intel Corporation in the U.S. and other countries.

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

Copyright © 2011. Intel Corporation.

http://intel.com/software/products

Copyright© 2012 Intel Corporation. All rights reserved.
*Other brands and names are the property of their respective owners.