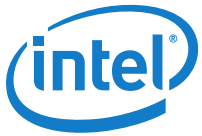


Heterogeneous applications

Using Intel® Threading Building Blocks and Intel® Cilk Plus

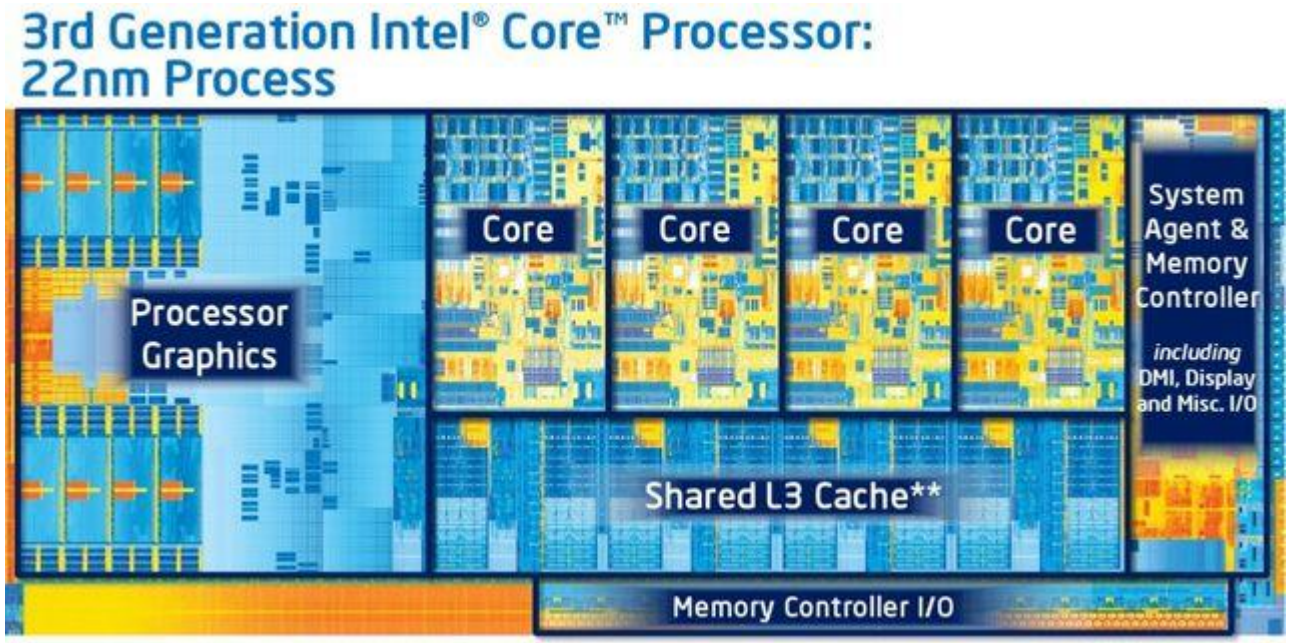
Based on Intel TBB fractal package example

Vladimir Polin

Threading Runtimes Engineering Manager

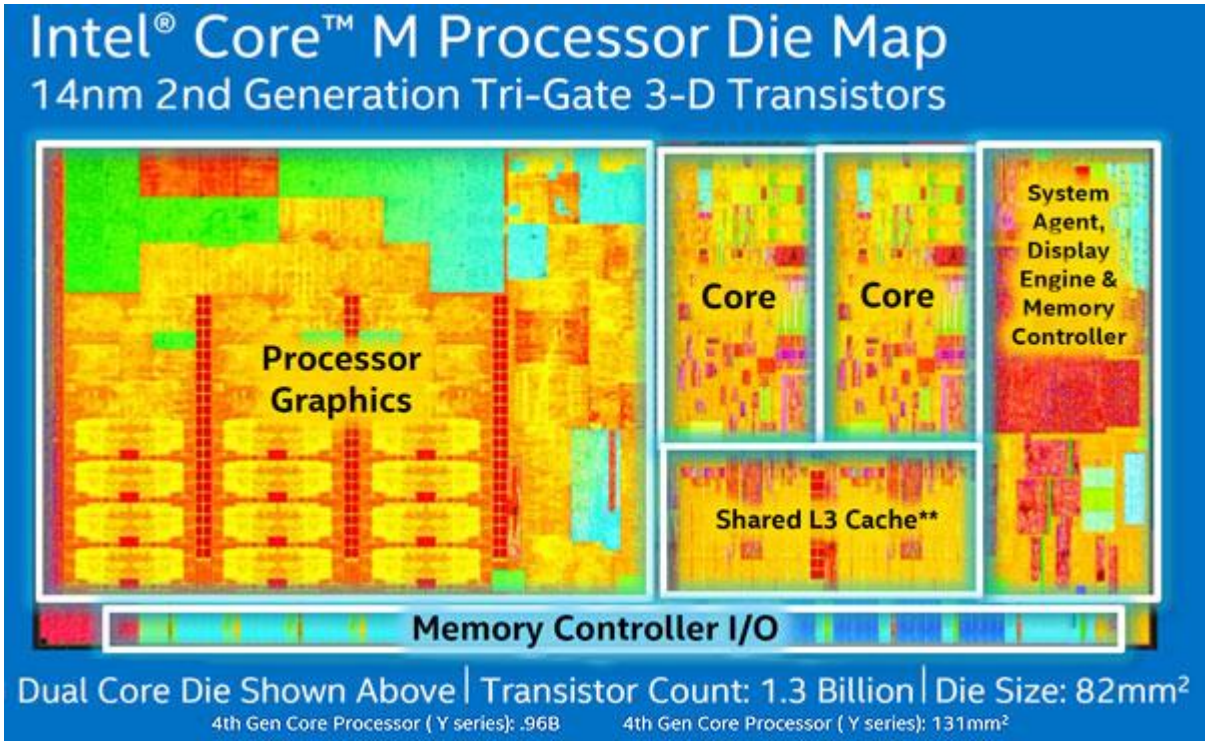
Rock your code.

Past



Rock your code.

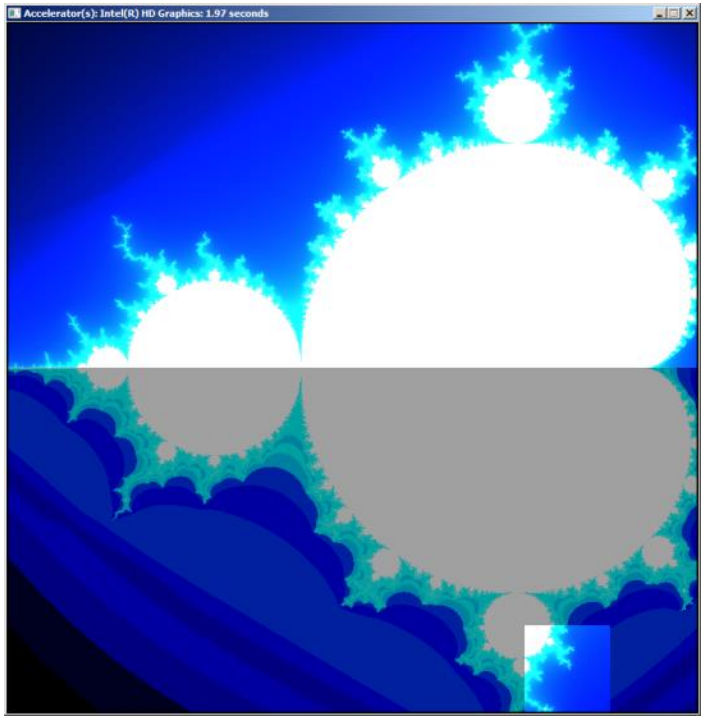
Present



How we can use and utilize this processor graphics?

Rock your code.

Mandelbrot set



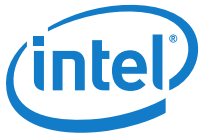
- Simple Mandelbrot set
- Open Source
- Part of the Intel TBB package
- Important:
  - Demo purposes only
  - Vivid picture

Example of parallel_for work

?	**?**	**?**	**?**	**?**	**?**	**?**	**?**
?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?
1	**3**	?	?	?	?	?	?
2	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?

? – means stealing.
Nobody knows where calculation will be started and how stealing will work

Initial Code



Calling calculation of a set parts in parallel

```
tbb::parallel_for(tbb::blocked_range2d<int>(0, size_y, grain_size, 0, size_x, grain_size),
    [&] (tbb::blocked_range2d<int> &r){
    if ( v->next_frame() )
        render_rect(r.cols().begin(), r.rows().begin(), r.cols().end(), r.rows().end());
},
    tbb::auto_partitioner());
```

Processing every set part

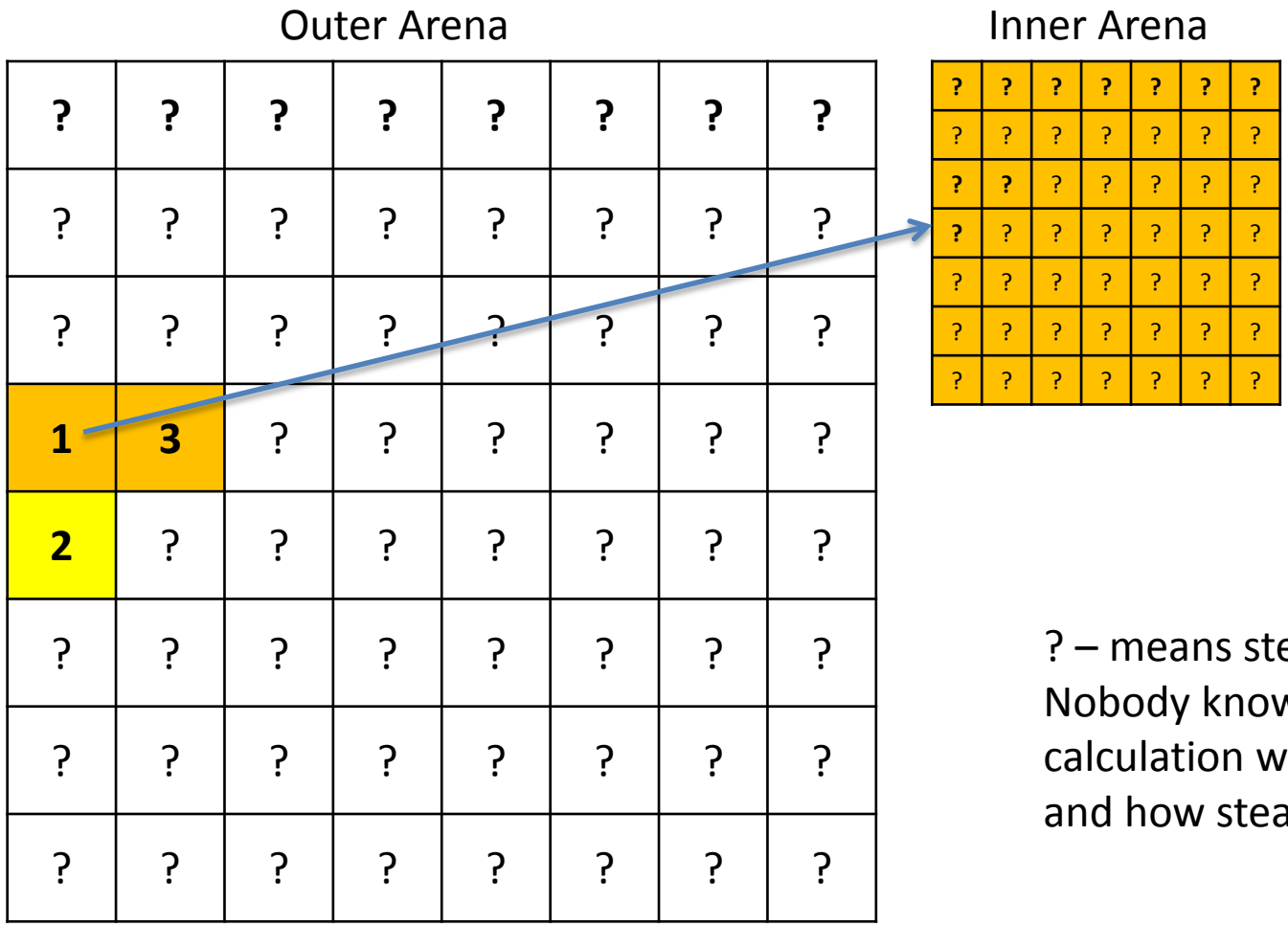
```
color_t fractal::calc_one_pixel(float x0, float y0)
{
    return calc_one_pixel(x0, y0, max_iterations, size_x, size_y, magn, cx, cy, 255);
}

void fractal::render_rect( int x0, int y0, int x1, int y1 ) {
    // render the specified rectangle area
    drawing_area area(off_x+x0, off_y+y0, x1-x0, y1-y0, dm);
    for ( int y=y0; y<y1; ++y ) {
        area.set_pos( 0, y-y0 );
        for ( int x=x0; x<x1; ++x ) {
            area.put_pixel( calc_one_pixel( x, y ) );
        }
    }
}
```

Rock your code.

User-Managed Task Arenas

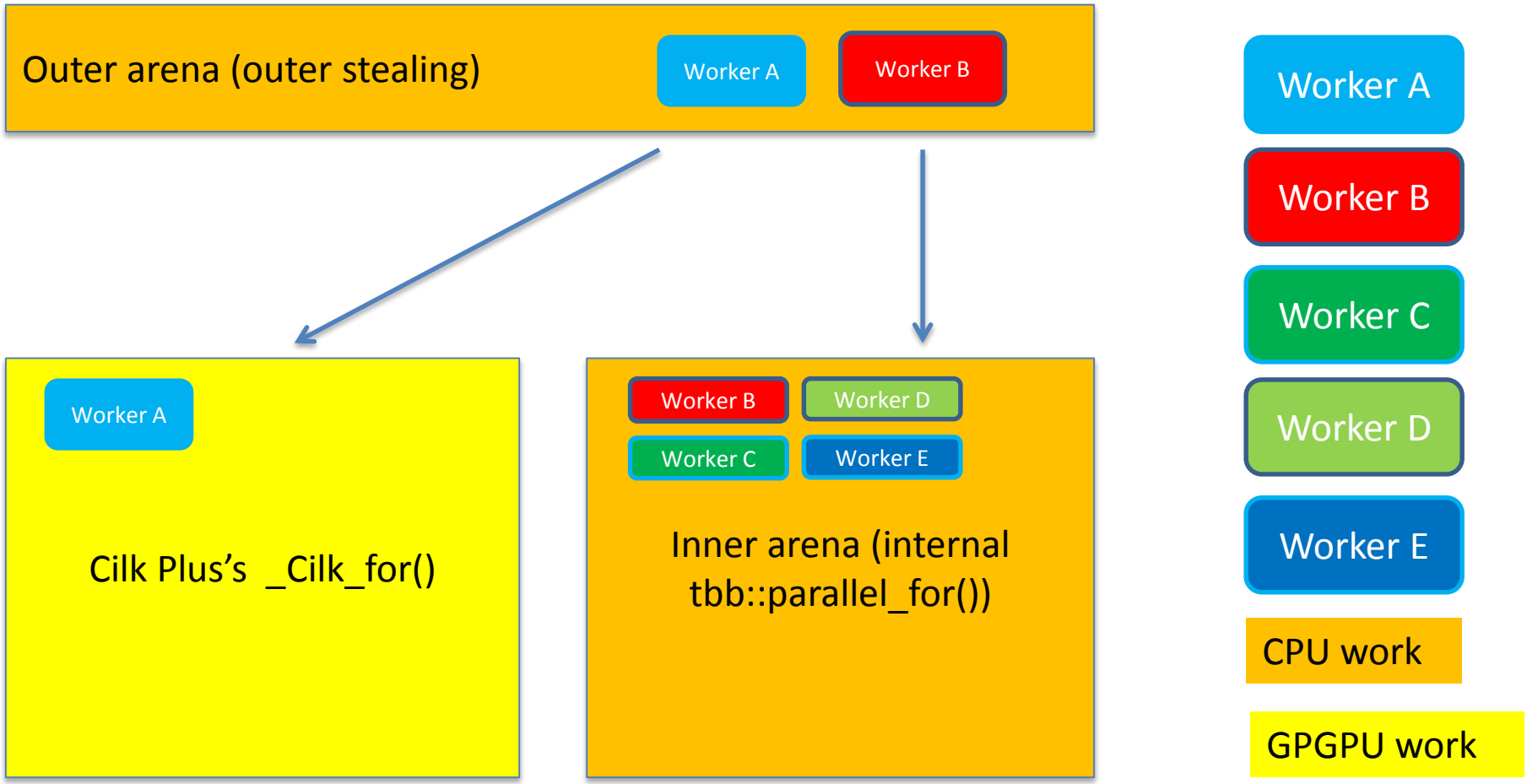
- an arena is a place for threads to share and steal tasks
- Name - task_arena class
- represents an internal task scheduler object where a number of threads, limited by a maximal concurrency level, share and execute tasks.
- The concurrency level of a task_arena is isolated and not affected by previous task_scheduler_init specifications.

Example of parallel_for work with 2 arenas

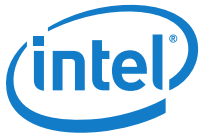


? – means stealing.
Nobody knows where calculation will be started and how stealing will work

Task arenas in action



Outer arena (outer stealing)

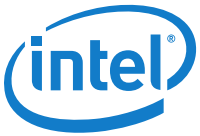



```
tbb::task_arena outer_arena(2);
outer_arena.execute([&] (){
    tbb::parallel_for(tbb::blocked_range2d<int>(0, size_y, grain_size, 0, size_x, grain_size),
        [&] (tbb::blocked_range2d<int> &r){
        if ( v->next_frame() )
            render_rect(r.cols().begin(), r.rows().begin(), r.cols().end(), r.rows().end());
    },
        tbb::auto_partitioner());
});
```

Rock your code.

Worker A

Cilk Plus's _Cilk_for()



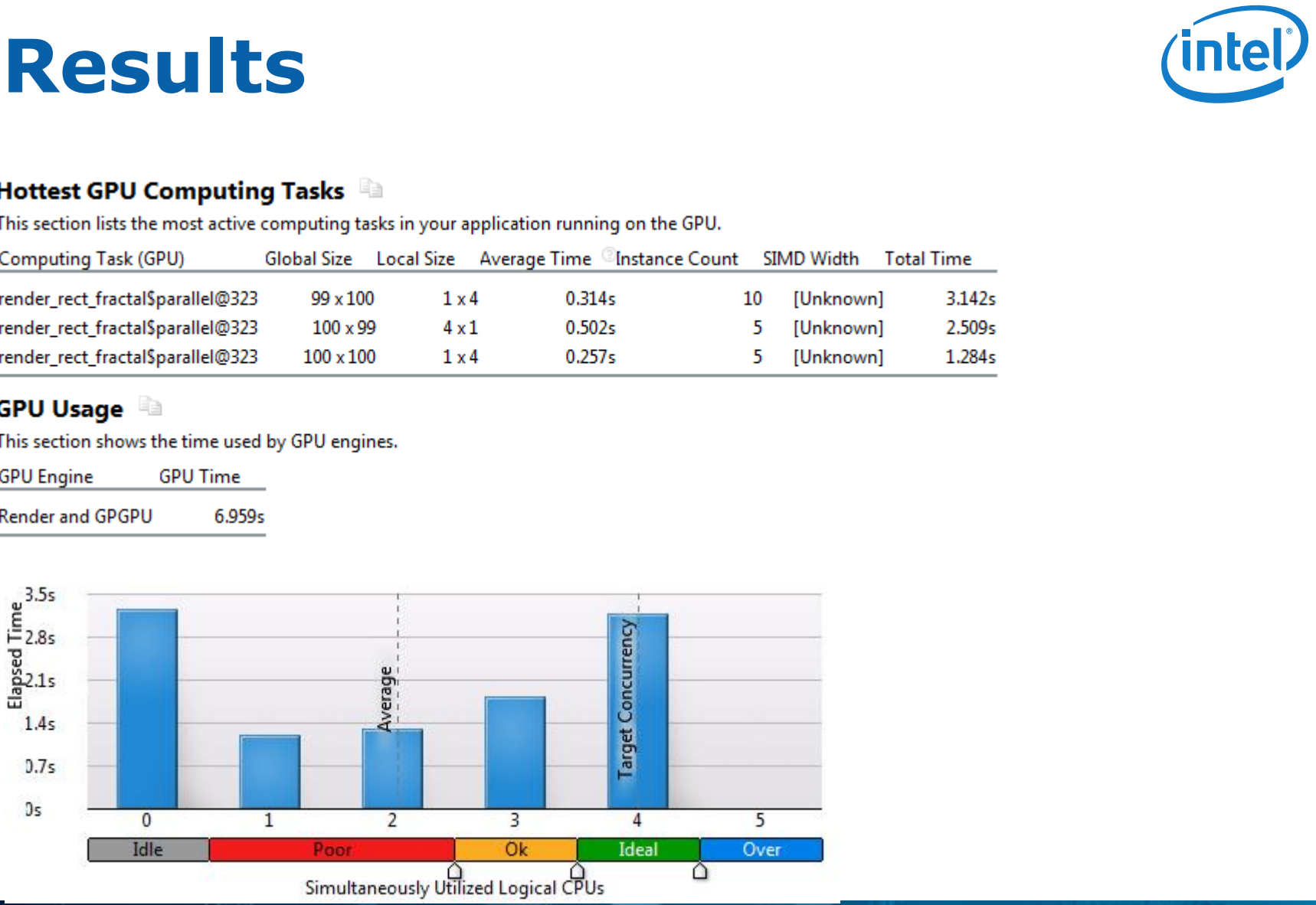
```
    const int delta_x = x1 - x0; const int delta_y = y1 - y0; int delta_y2; int idx = 0; int status;
    unsigned __int32 fractal_data_array[delta_x][delta_y];
#pragma offload target(gfx) pin(fractal_data_array: length((delta_x)* (delta_y)))
    _Cilk_for(int y = 0; y < delta_y; ++y) {
        _Cilk_for(int x = 0; x < delta_x; ++x) {
            fractal_data_array[x][y] =
                calc_one_pixel(tmp_off_x + x, tmp_off_y + y, tmp_max_iterations, tmp_size_x, tmp_size_y, tmp_magn, tmp_cx, tmp_cy, 160);
        }
    }
    drawing_area area(off_x + x0, off_y + y0, x1 - x0, y1 - y0, dm);
    for ( int y = y0, y_temp = 0; y < y1; ++y, ++y_temp ) {
        area.set_pos(0, y - y0);
        for ( int x = x0, x_temp = 0; x < x1; ++x, ++x_temp ) {
            area.put_pixel(fractal_data_array[x_temp][y_temp]);
        }
    }
```

Rock your code.


```
cpu_arena = new tbb::task_arena(<Number of CPU threads>);
cpu_arena->execute([&]{
    tbb::parallel_for(tbb::blocked_range2d<int>(y0, y1, inner_grain_size, x0, x1, inner_grain_size),
        [&] (tbb::blocked_range2d<int> &r){
        int x0 = r.cols().begin(), y0 = r.rows().begin(), x1 = r.cols().end(), y1 = r.rows().end();
        drawing_area area(off_x + x0, off_y + y0, x1 - x0, y1 - y0, dm);
        for ( int y = y0; y < y1; ++y ) {
            area.set_pos(0, y - y0);
            int x;
            color_t pixels_colors[8];
            for ( int x = x0; x < x1; ++x ) {
                area.put_pixel(calc_one_pixel(x, y, tmp_max_iterations, tmp_size_x, tmp_size_y, tmp_magn, tmp_cx, tmp_cy, 255));
            }
        }
    });
});
```

Rock your code.

Results



Hottest GPU Computing Tasks

This section lists the most active computing tasks in your application running on the GPU.

Computing Task (GPU)	Global Size	Local Size	Average Time	Instance Count	SIMD Width	Total Time
render_rect_fractal$parallel@323	99 x 100	1 x 4	0.314s	10	[Unknown]	3.142s
render_rect_fractal$parallel@323	100 x 99	4 x 1	0.502s	5	[Unknown]	2.509s
render_rect_fractal$parallel@323	100 x 100	1 x 4	0.257s	5	[Unknown]	1.284s

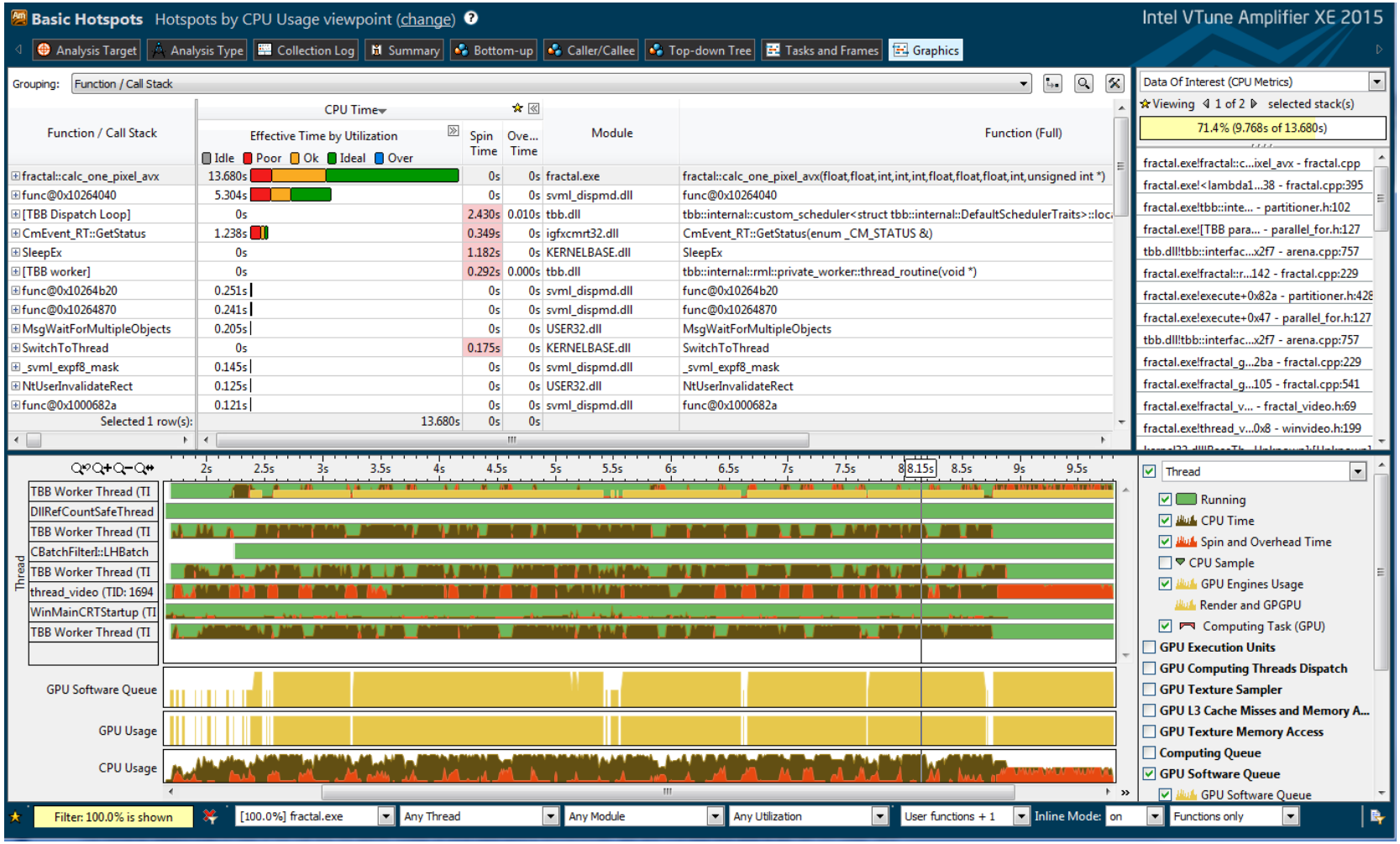
GPU Usage

This section shows the time used by GPU engines.

GPU Engine	GPU Time
Render and GPGPU	6.959s

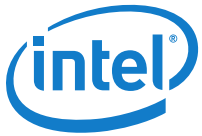
Rock your code.

Results (Graphics view)



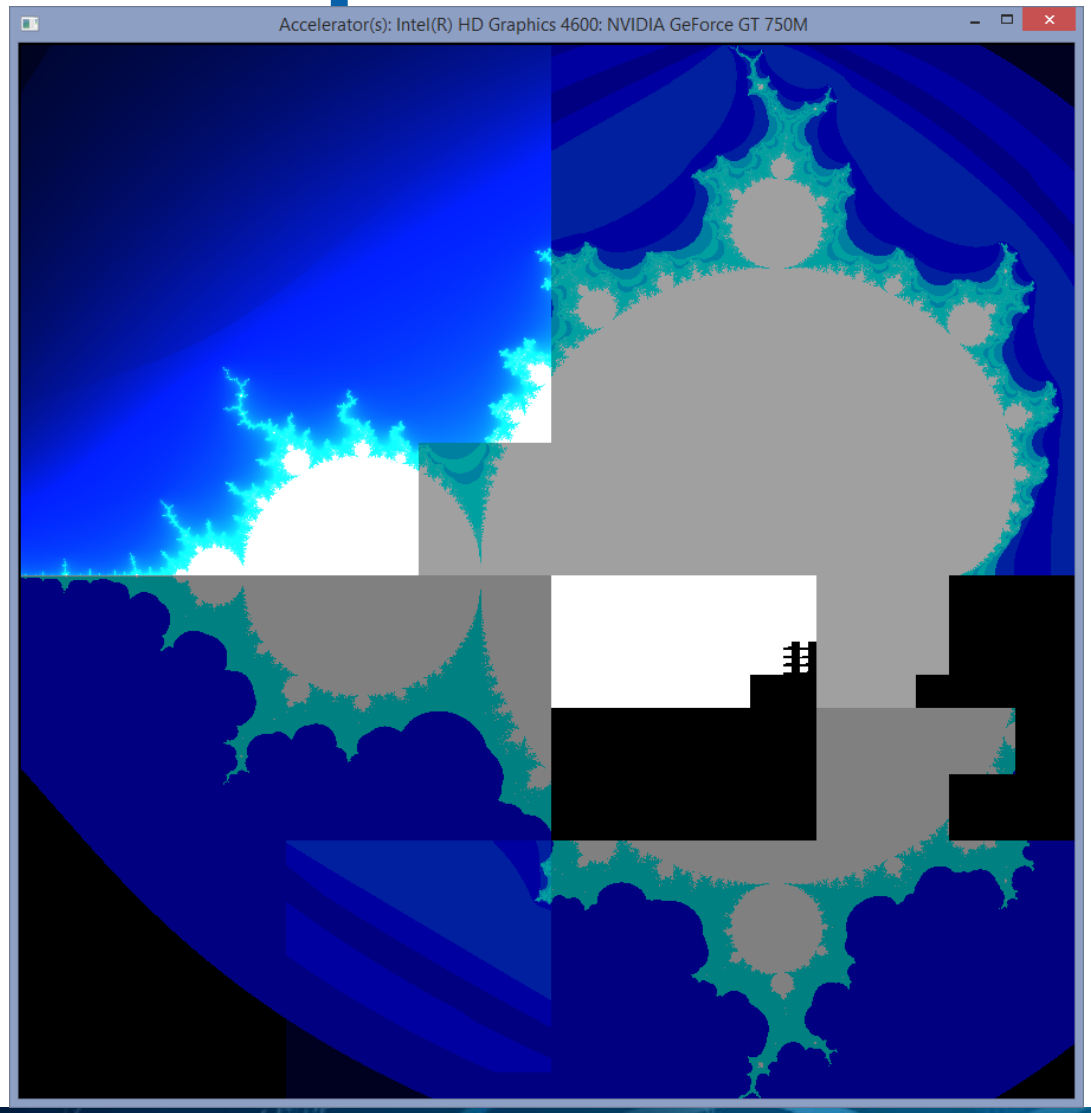
Rock your code.

Tomorrow?



Can you application utilize all resources efficiently?

Backup



Rock your code.

Conclusion

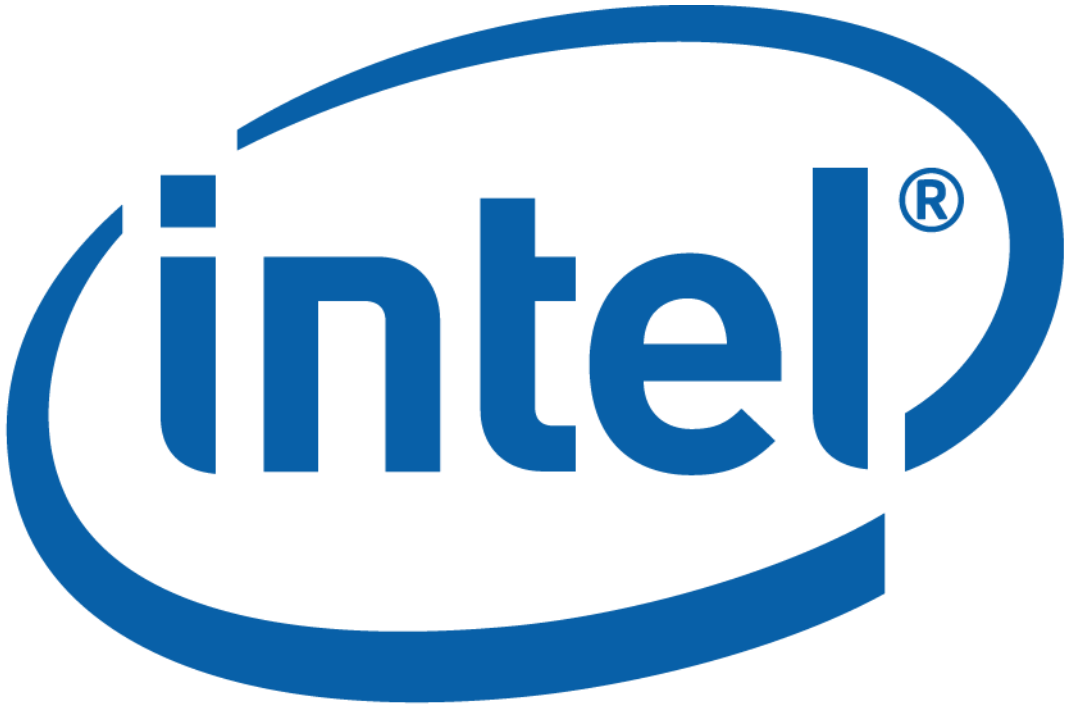
Scale productive
Tune and debug for more cores and nodes

Scale effectively
Keep performance

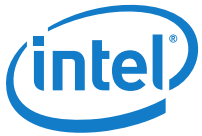
Scale for the future
Multicore today, Many-core tomorrow

Intel® Parallel Studio XE: The Suite for developing of applications for shared and distributed memory models

Rock your code.



Software

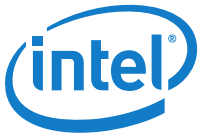


Уведомление об оптимизации

Компиляторы Intel могут не обеспечивать для процессоров других производителей такой же уровень оптимизации для оптимизаций, которые не являются присущими только процессорам Intel. В число этих оптимизаций входят наборы команд SSE2, SSE3 и SSSE3, а также другие оптимизации. Корпорация Intel не гарантирует наличие, функциональность или эффективность оптимизаций микропроцессоров других производителей. Содержащиеся в данной продукции оптимизации, зависящие от микропроцессора, предназначены для использования с микропроцессорами Intel. Некоторые оптимизации, не характерные для микроархитектуры Intel, резервируются только для микропроцессоров Intel. Более подробную информацию о конкретных наборах команд, покрываемых настоящим уведомлением, можно получить в соответствующих руководствах пользователя и справочниках на продукт.

Уведомление, редакция № 20110804

Legal Disclaimer & Optimization Notice



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.

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.

Copyright © 2014, Intel Corporation. All rights reserved. Intel, Pentium, Xeon, Xeon Phi, Core, VTune, Cilk, and the Intel logo are trademarks of Intel Corporation in the U.S. and other countries.

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