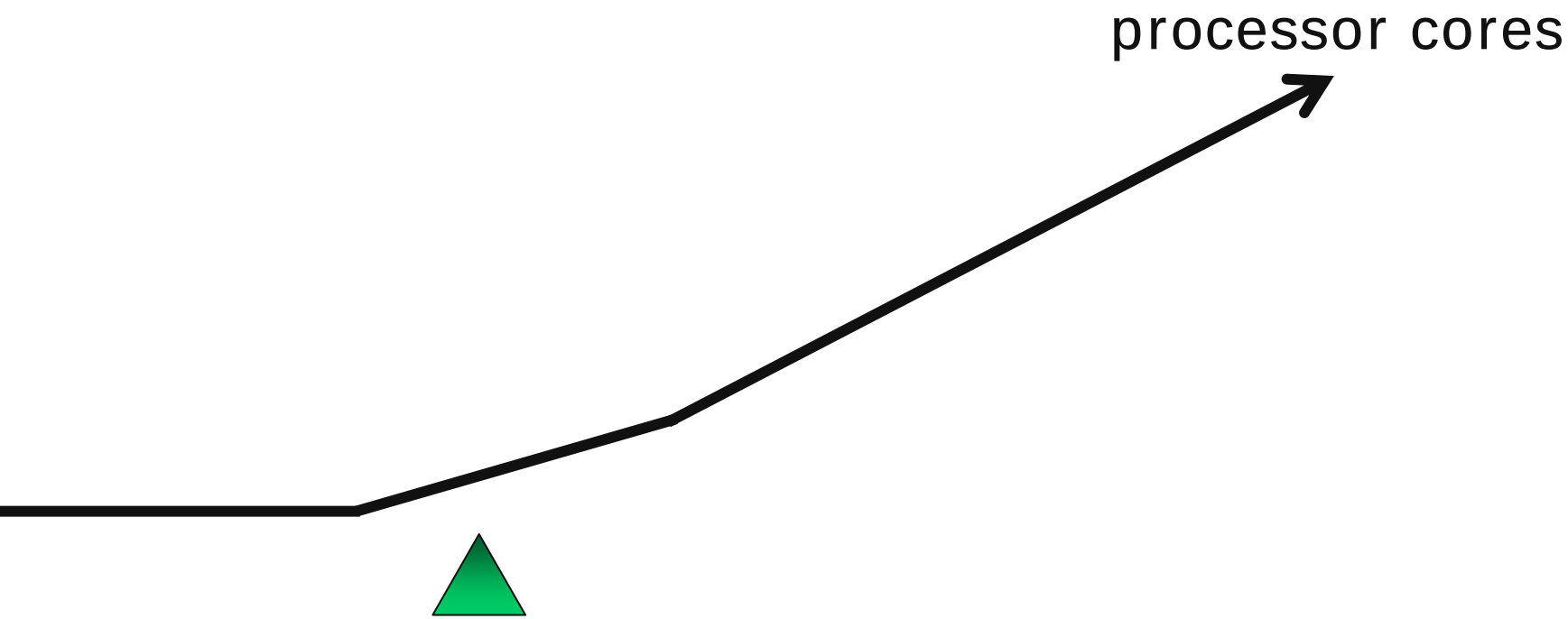
Helping Software Developers Programming for Multi-core

James Reinders
Director of Sales and Marketing
Technical Evangelist
Intel Software Development Products

Intel Developer FORUM

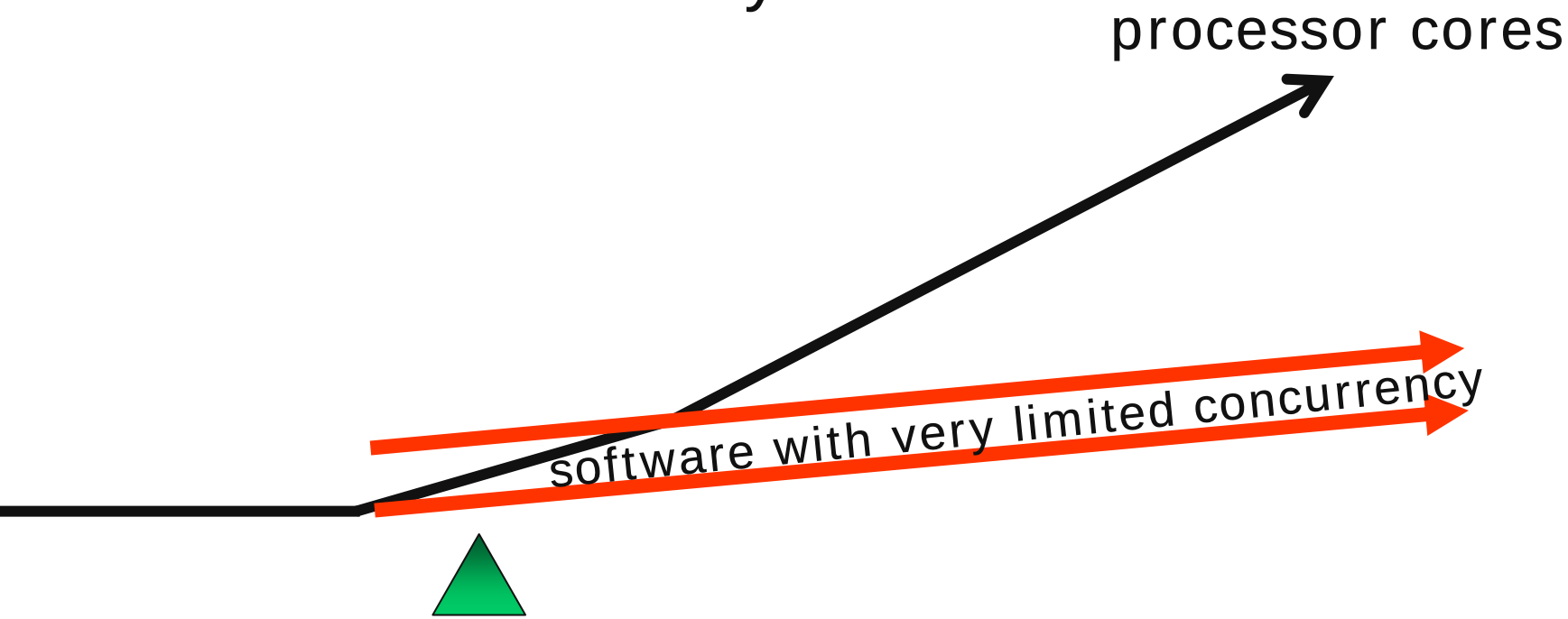
We have a future ahead where processors have more and more cores.



Aside from servers, concurrency is mostly outside applications today.

Multiple tasks. Environment.

Limited concurrency.

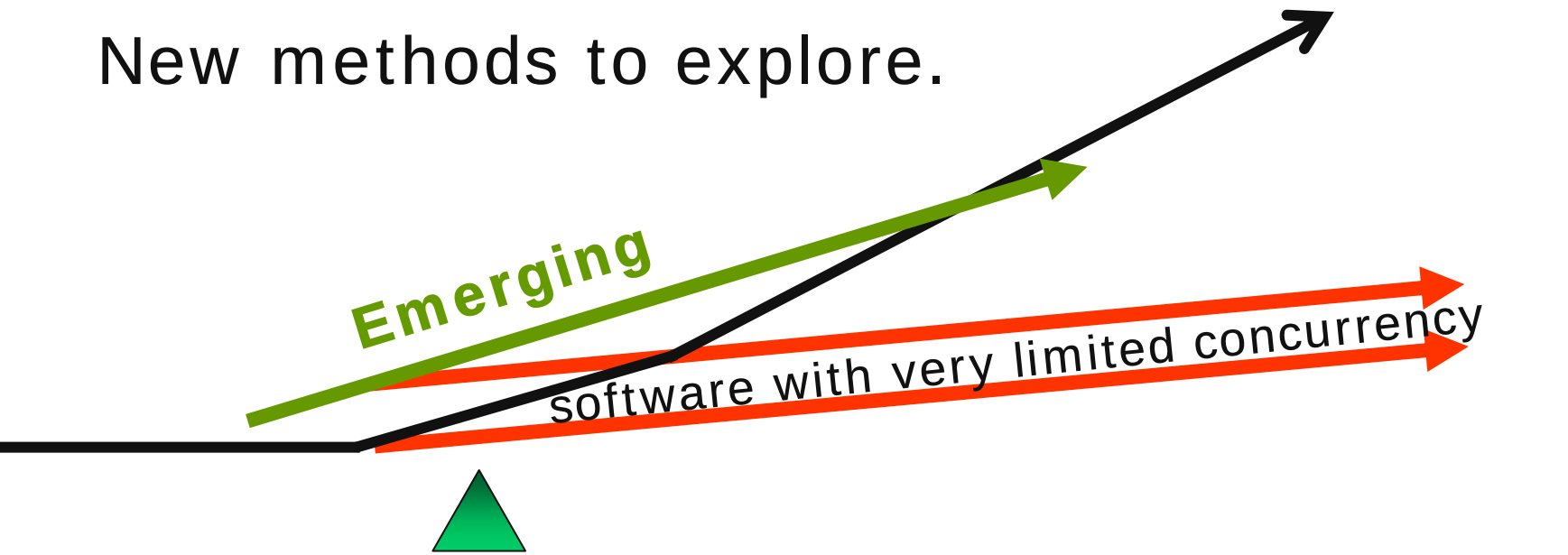


I will talk about the steps being made *today* to put concurrency in more software development.

Learning to THINK PARALLEL.

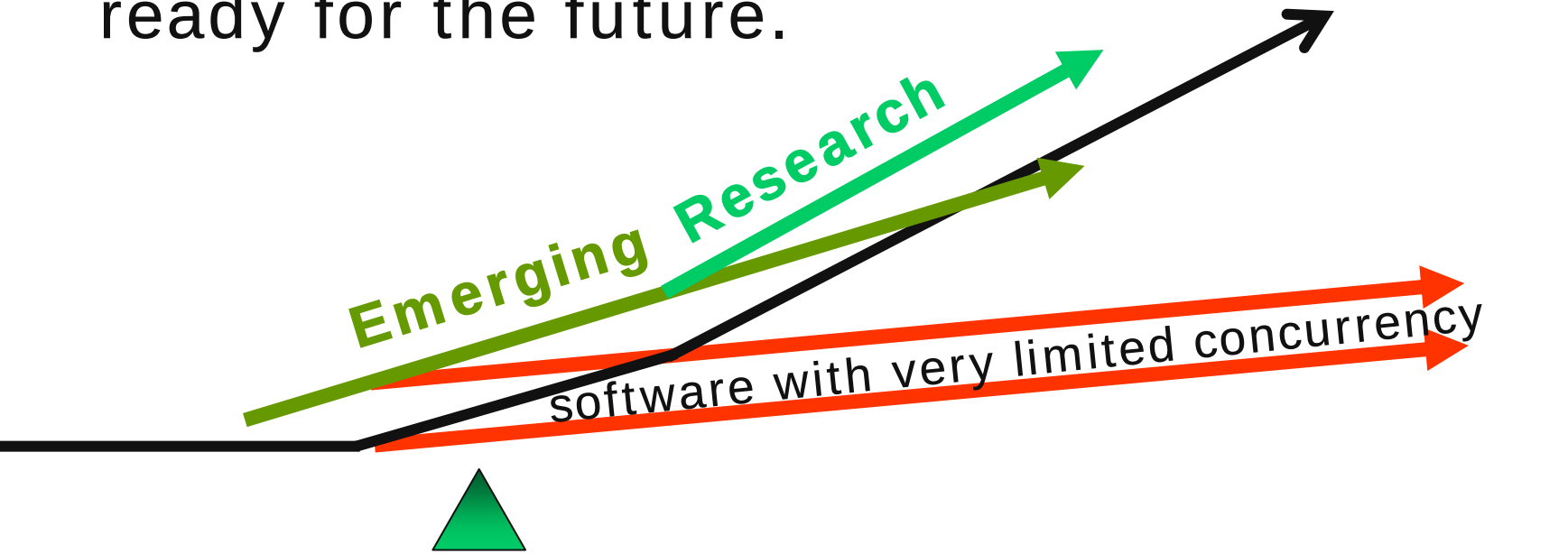
New methods to use.

New methods to explore.



But we already see a future with even more cores.

Jerry will introduce those challenges of the future, and what we are researching today to be ready for the future.



Adding Concurrency to Programs

Interest in Parallel Programming is Very High

- Learning
- Programming

  and
- Researching

Adding Concurrency to Programs

LEARNING

Developing and sharing expertise

Intel instructors develop Intel® Software College

Intel Software College training with a network of trainers

Intel Software College material in/with Universities

Intel Software Network (ISN) forums, papers, books, experts, community, contests

information all in one place:

www.intel.com/software/college

Learning to *THINK PARALLEL*

Update: Intel® Software College training material

For professional programmers:
the opportunity to take trainer led courses worldwide

"Key" courses are used to deliver course worldwide.

- instructor guides, student guides, workbooks, etc.

Key courses expanding:

- started with "Multi-core Programming" last year
- we've added "Introduction to Parallel Programming, OpenMP" and "Introduction to Parallel Programming, Java" this month

Universities have access to all this material

- as well as many other courses (not all have full bound book sets)

Learning to *THINK PARALLEL*

Update: Intel® Software College material in / with universities

41 → 280 → *more* Universities

We announced in August 2006: 41 Universities.

We are over 280 universities participating as of the start of September 2007.

We expect to be over 400 by the end of the year.

32K students reached directly

The program material reached 7000 undergraduates in 2006.

This year, we'll reach more than 25000 directly.

Reaching *undergraduates* is a key objective.
We are *very* pleased with how much we've been able to help accelerate this by making our material available.

Adding Concurrency to Programs

Interest in Parallel Programming is Very High

- Learning

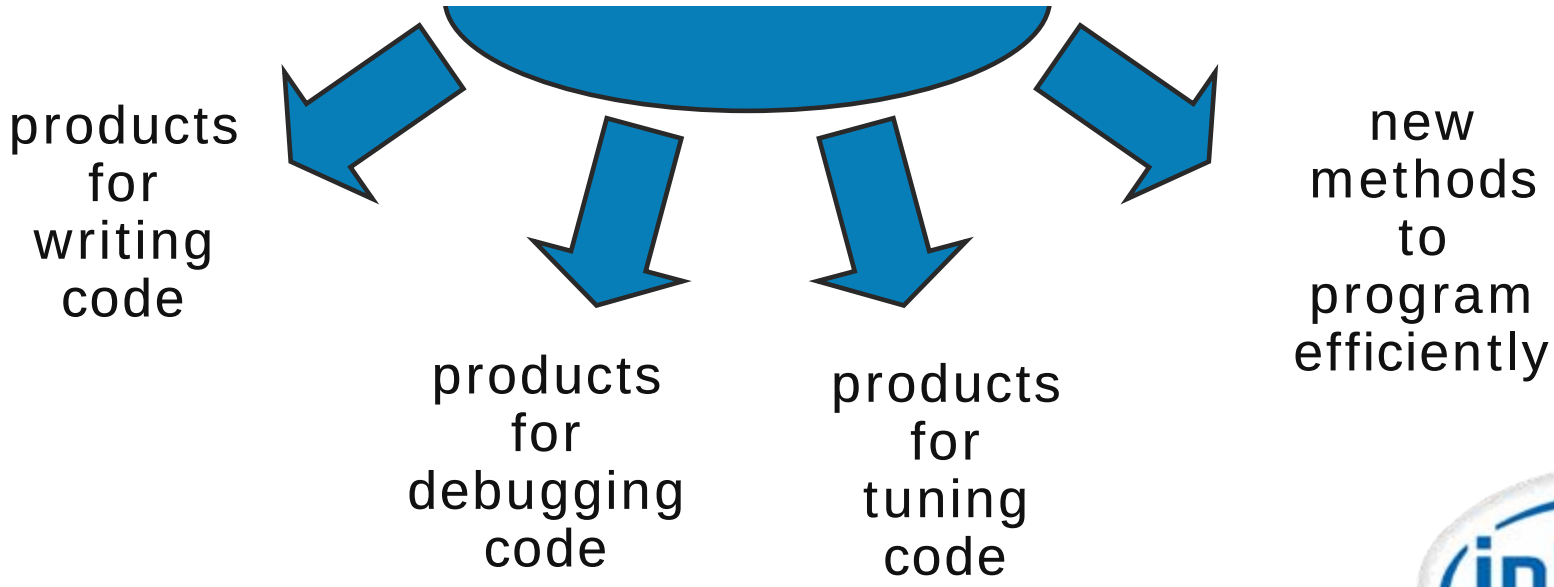
  Programming

  and
- Researching

Adding Concurrency to Programs

PROGRAMMING

- Long time experts have made due with limited tools
- New experts need to get the concurrency job done, not make it a career unto itself



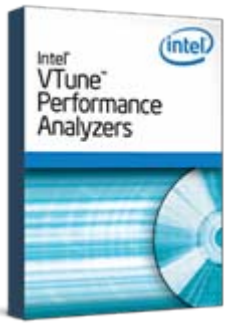
products for writing code

with new methods to program efficiently

Very popular compilers, libraries and analysis tools.

Best suite of products to help with software development for multi-core processors.

products for debugging code



products for tuning code

Intel® Threading Building Blocks

New methods to program efficiently

Contributions, ports, usages, more…

more information on TBB: threadingbuildingblocks.org

Open Source & Commercial Adoption of Intel® Threading Building Blocks

Commercial – large & small

- **Maya** - Digital images, 3D animation, and visual effects. (www.autodesk.com/maya)
- **Knowledge Miner** - Data Mining Engine (knowledgeminer.com)

Many more…

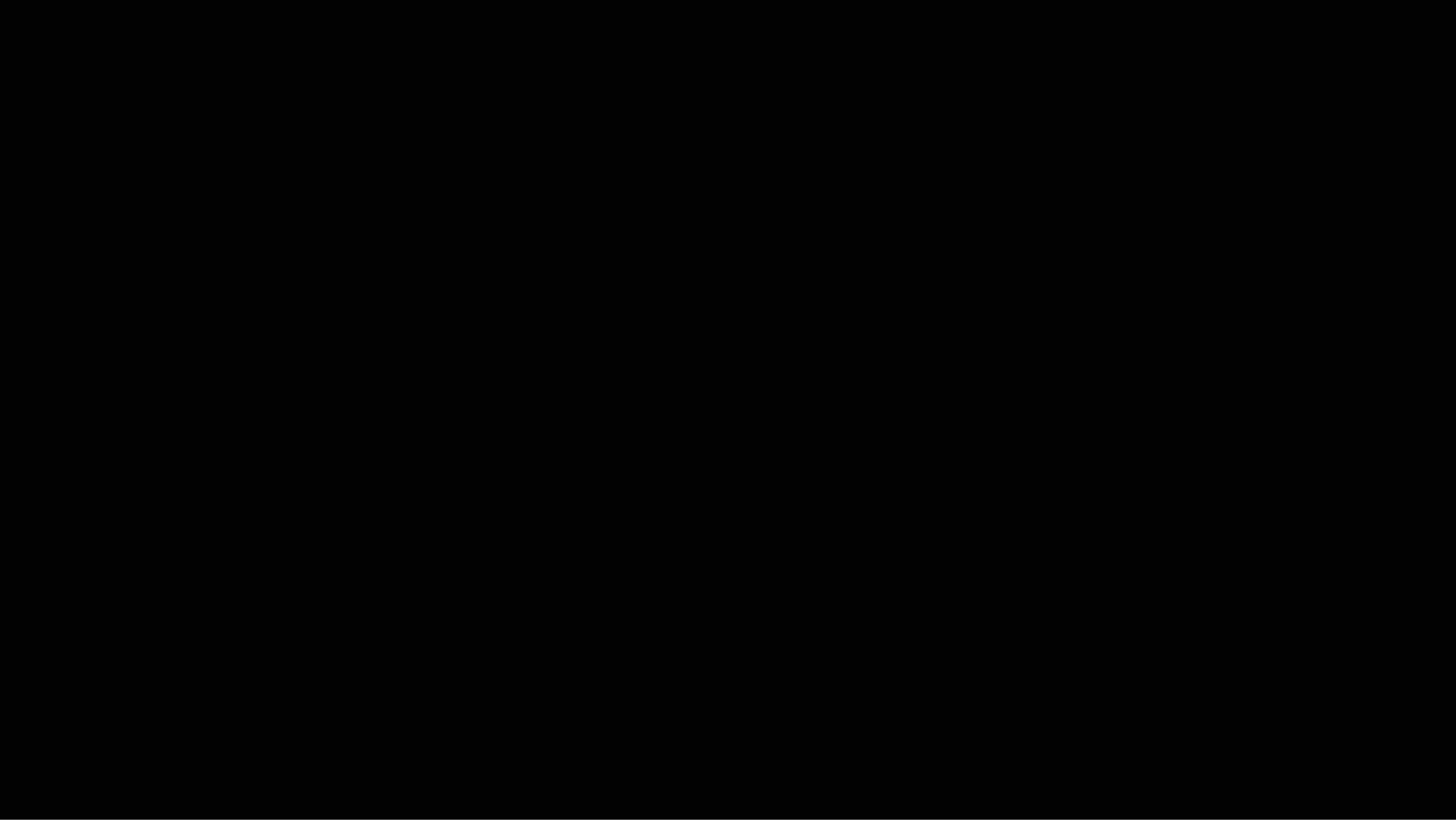
Open source projects

- **Hierarchical Tiled Arrays** (polaris.cs.uiuc.edu/hta)
- **Mandelbrot** – Mandelbrot set on Core Duo (quickwayne.googlepages.com/mandelbrotsetonintelduocore)
- **Par2cmdline** – create & repair data files w/ Reed Solomon coding (chuchusoft.com/par2_tbb/index.html)
- **Yetisim** – C++ discrete event simulation library (www.yetisim.org)

More examples mentioned on our website… threadingbuildingblocks.org

"Threading an application as large as Maya is really difficult. We're talking about an application that's over 10 million lines of C++ code. It pushes the system to its very limits."

- Kevin Tureski
Product Director
Autodesk Media & Entertainment



Kevin Tureski
Product Director
Autodesk Media & Entertainment

"Maya's wrap deformer helps define the outline of a character. Using Intel Threading Building Blocks we're able to improve playback performance by a factor of seven on an eight-core machine.

That's key. Being able to assess a character, to be able to define it, to get a digital character performance, playback speed is essential and we're relying on Intel Threading Building Blocks to help us deliver what our customers are demanding of us."

- Kevin Tureski
Product Director
Autodesk Media & Entertainment

Adding Concurrency to Programs

Interest in Parallel Programming is Very High

- Learning
- Programming

and

Researching

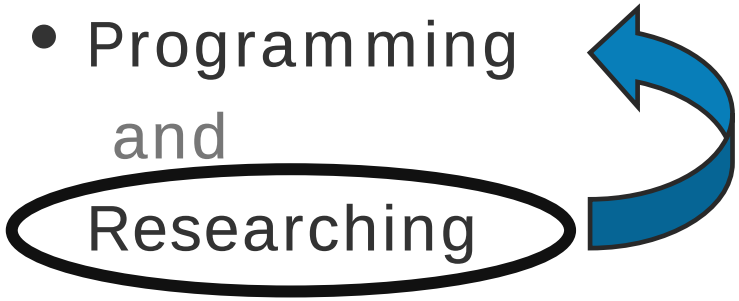
Adding Concurrency to Programs

Interest in Parallel Programming is Very High

- Learning
- Programming

and

Researching



Jerry will speak about the research more.

First let's talk about this "research to product" path.

Research to Product Path

example:

- Intel® Threading Building Blocks
  - draws on university and industry research stretching back for decades
  - draws on code written by Intel Research (McRT)
  - overall product written and supported by product team
  - fully supported Intel product

Research to Product Path

example:

- Intel® Threading Building Blocks
  - draws on university and industry research stretching back for decades
  - draws on code written in Intel Research (McRT)
  - overall product written and supported by product team
  - fully supported Intel product

- **What's next?**
  - What can we do with advanced product development and research – before it is product?

Research to Product Path

example:

- Intel® Threading Building Blocks
  - draws on university and industry research stretching back for decades
  - draws on code written in Intel Research (McRT)
  - overall product written and supported by product team
  - fully supported Intel product

- **What's next?**
  - What can we do with advanced product development and research – before it is product?

What If…

Whatif.intel.com

AVAILABLE STARTING TODAY

vehicle for software developers to experience and shape the future of software

- Opportunity to discuss and offer feedback on promising software technology from Intel
- Interact with Intel researchers, product designers and software professionals
- Starting today: Easy and free access to Intel's STM compiler, a Mixed Mode Debugger, and a new Performance Tuning Tool (learn more SSGS002, ITJ)
- Check back regularly for more additions

Locks – can't live with them, can't live without them

But can we at least get some relief from using locks?

"The most promising technology" is widely believed to be **Transactional Memory (TM)**

Software Transactional Memory – Intel STM Compiler

STM = Software Transactional Memory

- *A promising technology to help accelerate the creation of parallel applications*
- *Will benefit from additional real world testing and feedback to guide us together*

STM = Software Transactional Memory

- *A promising technology to help accelerate the creation of parallel applications*
- *Will benefit from additional real world testing and feedback to guide us together*

Ripe for hype. Ripe for casual dismissal.

Reality is: an opportunity to reduce dependence on locks, when appropriate – and increase 'composability' at least in part.

Intel® STM Compiler is available ***TODAY***

- *Most comprehensive STM support for C and C++ programmers – on Windows and Linux*
- *Build upon Intel's product compiler – fully production ready*
- *Ideal tool for general experimentation, testing and industry dialog around real results*
- *Great opportunity to explore Software Programming Models for parallel programming*
- *This opens a very large opportunity for experimentation and much needed dialog on real results*
- *STM compiler: Freely available to current Intel compiler customers and free trial versions available to everyone else*

AVAILABLE STARTING TODAY

Whatif.intel.com

Intel® C++ STM Compiler Prototype Edition

Locks: can't live with them… can't live without them?

Use at our own risk…

"CARELESS"

```
c[i] = c[i] + sin(c[i]);
```

Conservatively – do one thing at a time…

"CAREFUL"

```
EnterCriticalSection(&lck);
c[i] = c[i] + sin(c[i]);
LeaveCriticalSection(&lck);
```

Say what we *really mean* with TM:

"JUST RIGHT"

```
__tm_atomic {
    c[i] = c[i] + sin(c[i]);
}
```

Intel® STM Compiler is available *TODAY*

- Simple idea
- Promising technology
- An alternative to locking for shared state (locks do *not* go away)

- What next?
  - It aims to make programming EASIER
  - More “composable”
  - **WANTED: Discussion and feedback on how well it achieves that in the real world.**

Intel® STM Compiler is available *TODAY*

- *A promising technology to help accelerate the creation of parallel applications*
- *Will benefit from additional testing and feedback*

- ***Intel® STM Compiler Prototype Edition perfect vehicle for this exploration***

It is about *exploration together*.

What will it solve better than locks?
What are the right role for locks? For TM?
What things can be build on top of TM?

Adding Concurrency to Programs

- Learning
- Programming
- Whatif.intel.com
- Researching

Learn more

- Whatif.intel.com
- intel.com/software (/college, /products)
- threadingbuildingblocks.org
- Renee James keynote on Thursday
- SSGS002

  Event-based Techniques for Software Tuning and Performance Analysis Session

  David A. Levinthal

  Senior Software Engineer,

  Intel Corporation

  Thursday Sept. 20 1:30 - 3:20pm