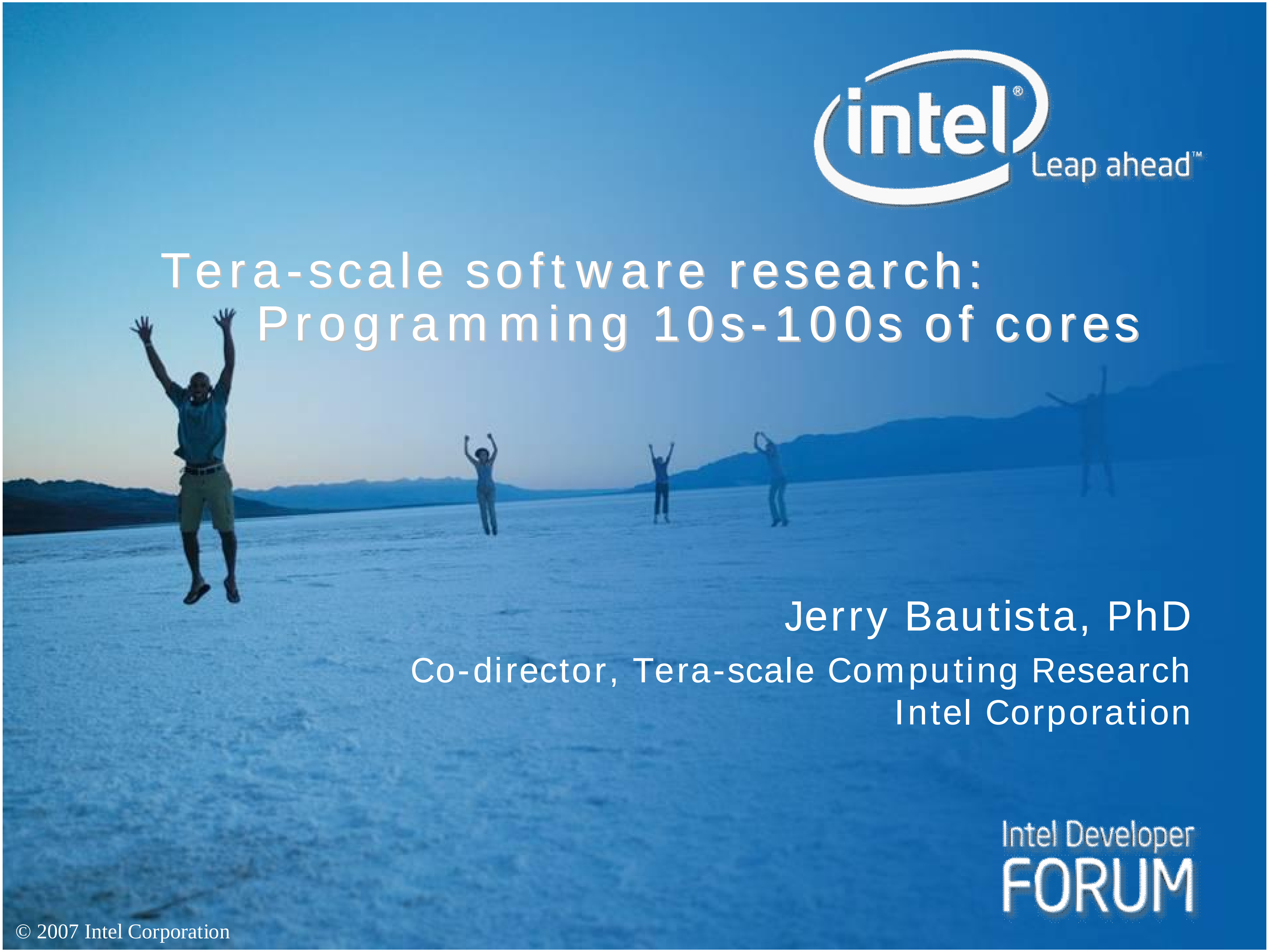
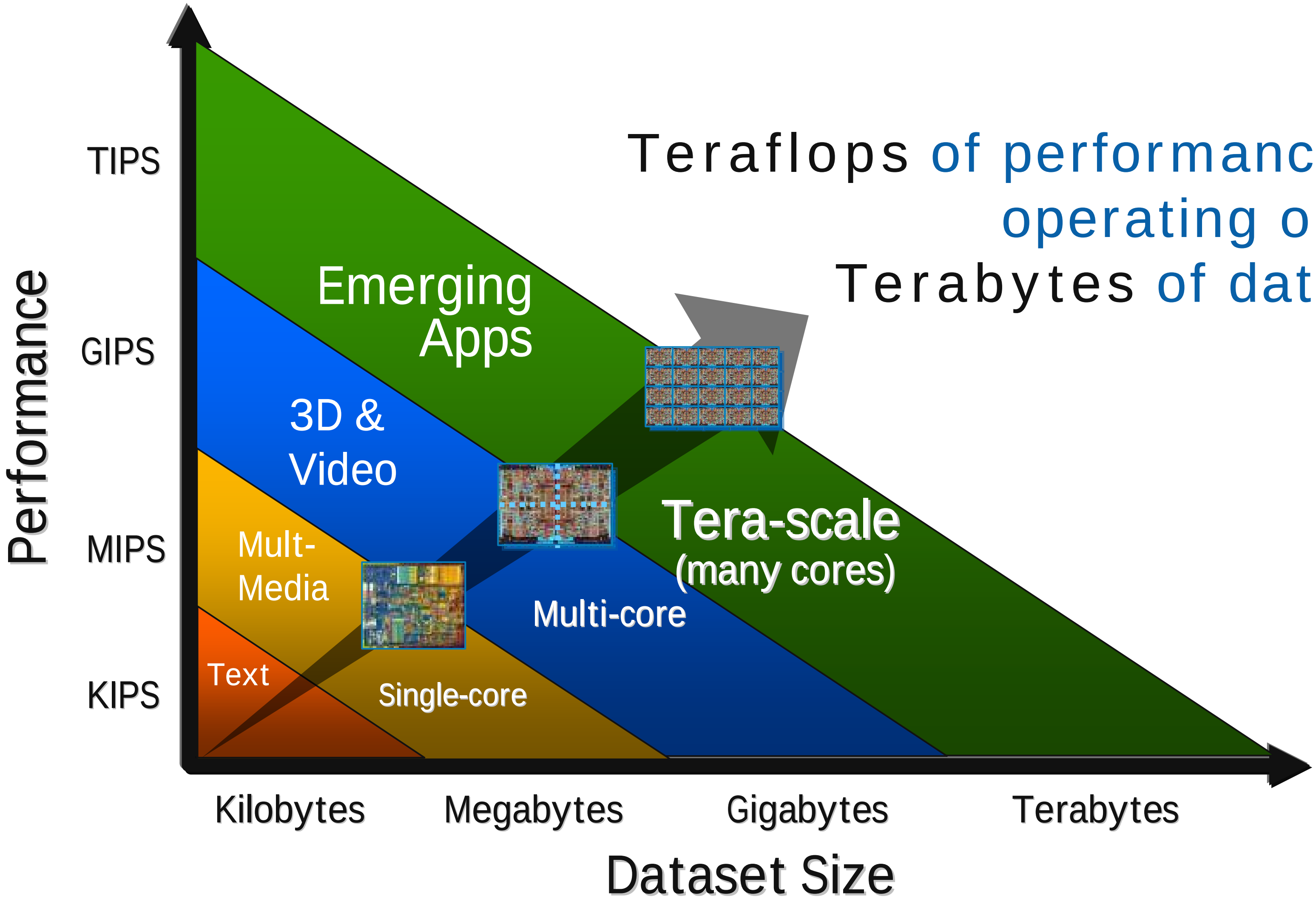


Tera-scale software research: Programming 10s-100s of cores

Jerry Bautista, PhD

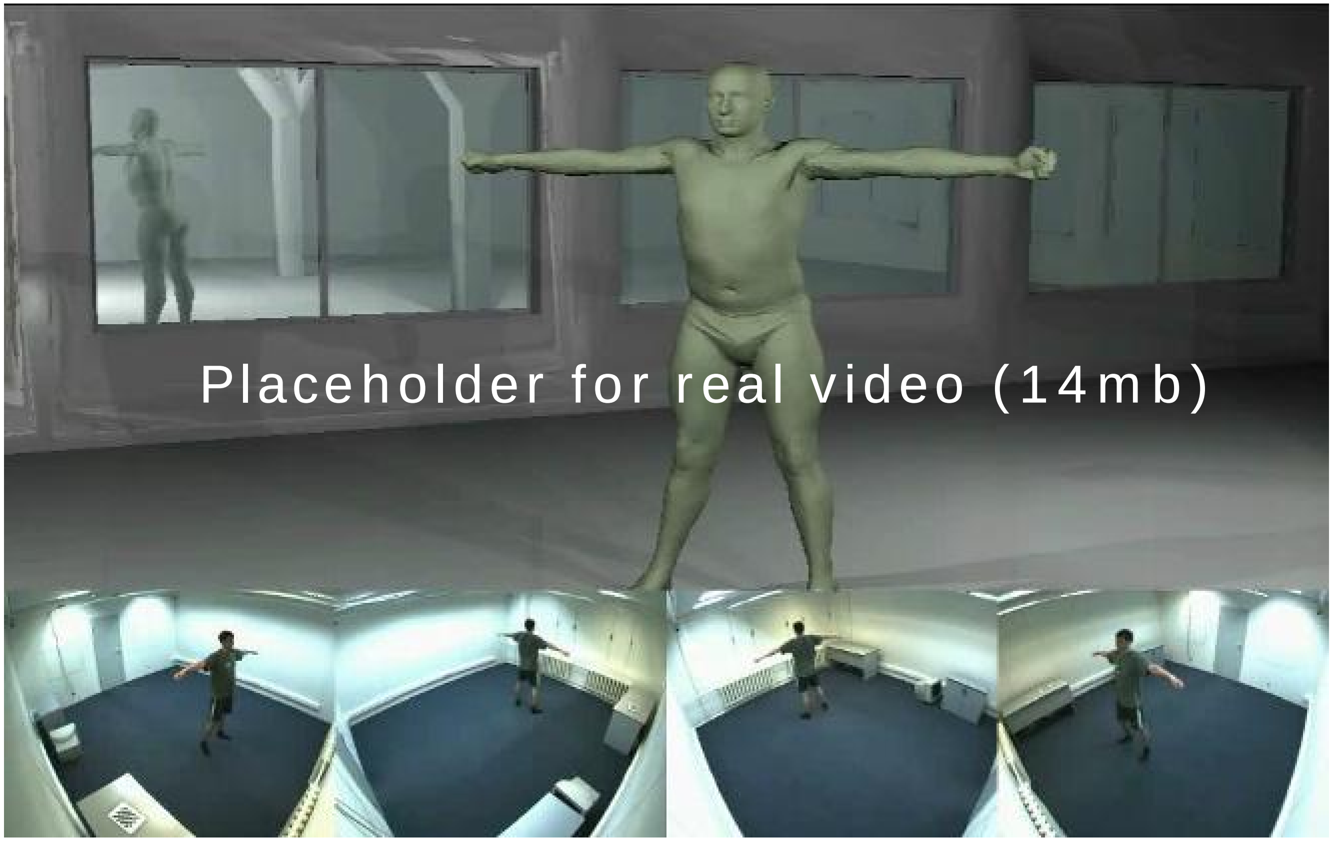
Co-director, Tera-scale Computing Research
Intel Corporation

What is Tera-scale?



Teraflops of performance operating on Terabytes of data

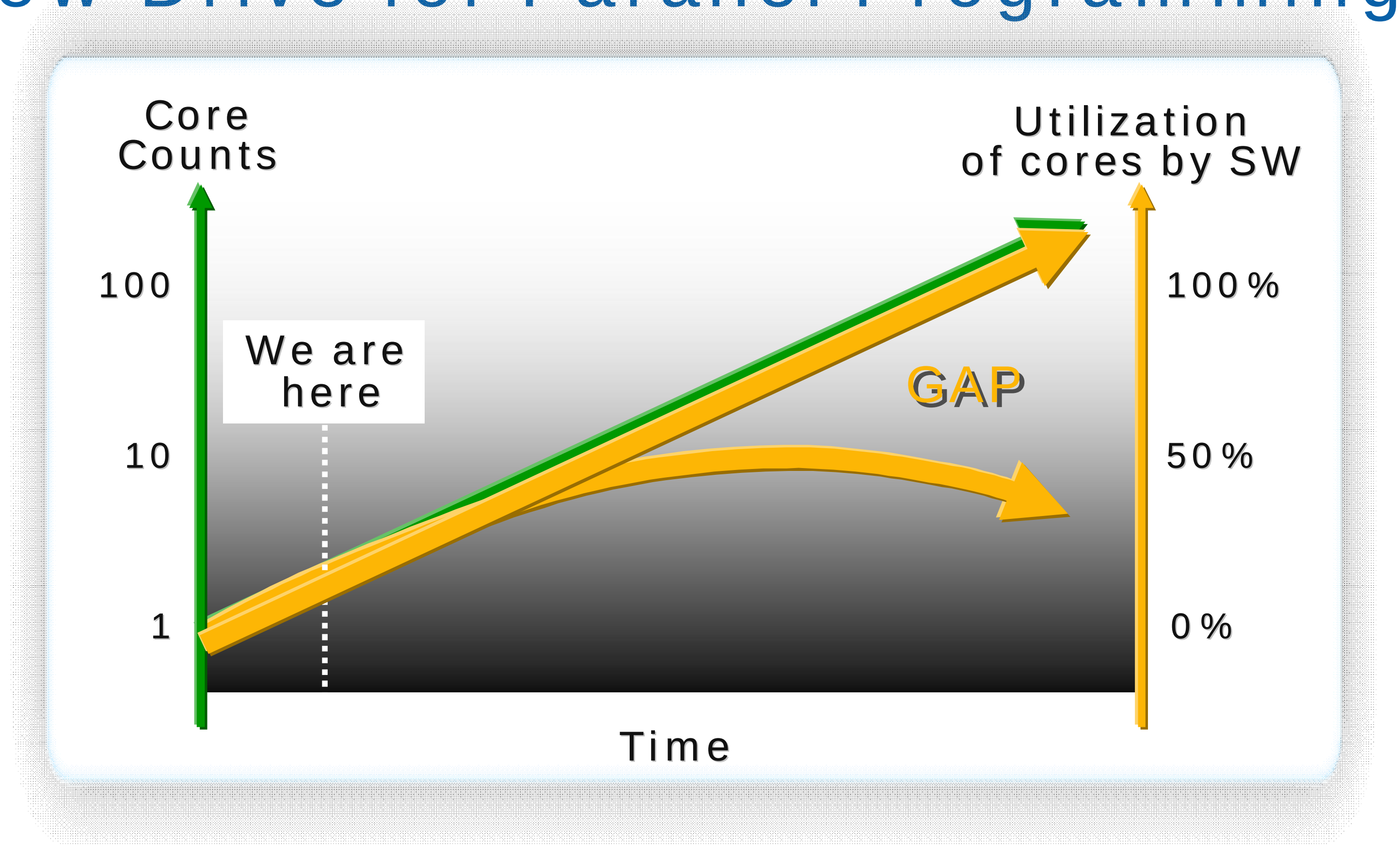
Example emerging Application



Many emerging apps are parallel and “model-based”
Ex: Modeled body, modeled motion, modeled lighting

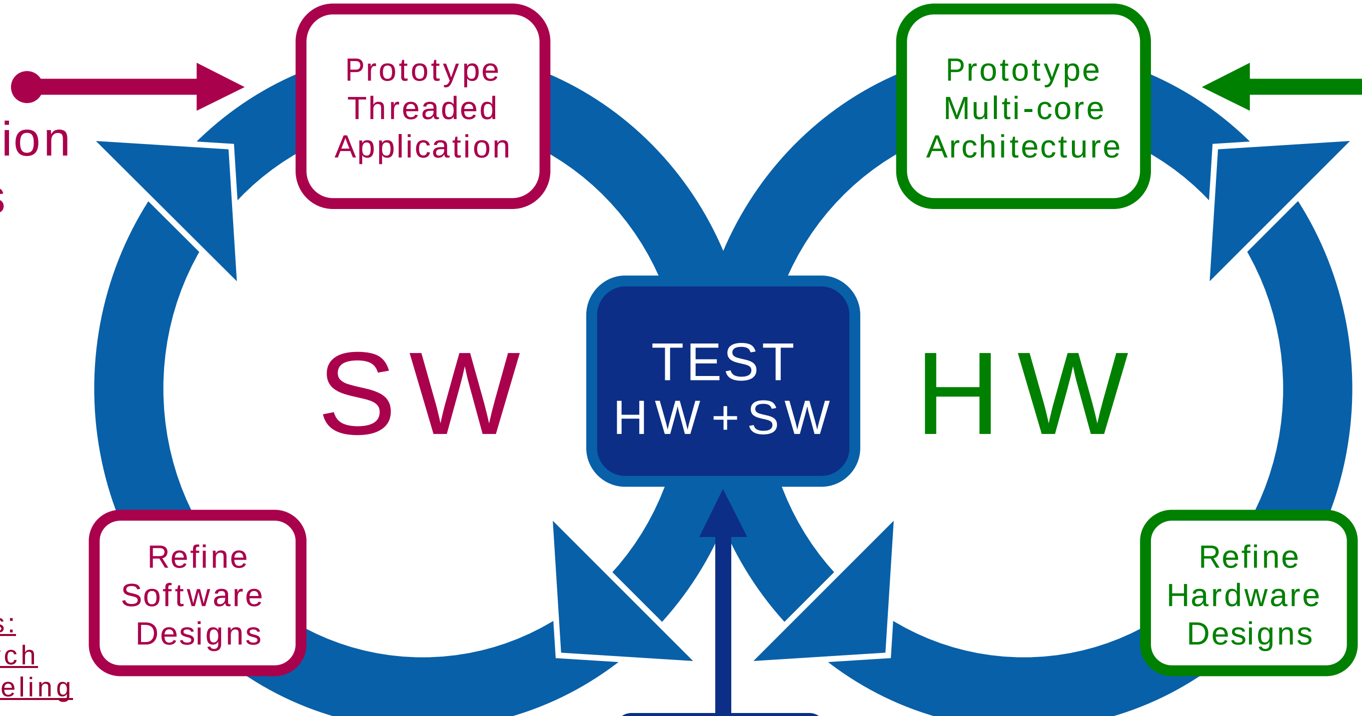
See Justin’s keynote on “Virtual World” Thursday

Tera-scale Computing:
A New Drive for Parallel Programming



Growing need for new parallel programming tools and capabilities

Joint Hardware & Software R&D

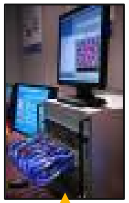


Examples:
Media search
Physical Modeling

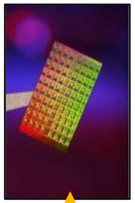
Examples:
Cache hierarchy
Task scheduling

RESEARCH TOOLS

Cycle-accurate Simulators
kHz speeds
~hours to modify



FPGA Emulators
MHz speeds
Days to modify

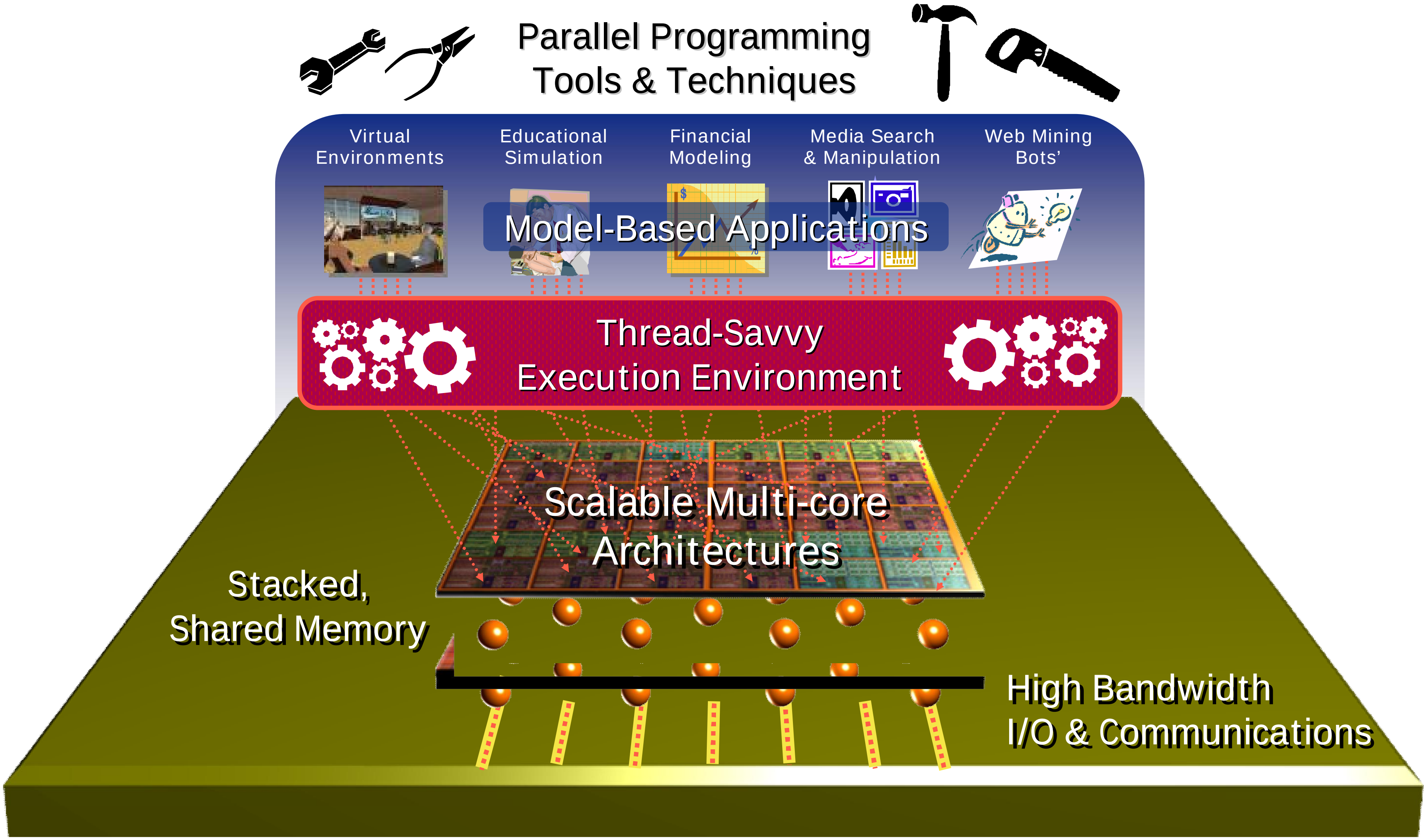


Silicon Prototypes
GHz speeds
Months to modify

See Emulator demo in tech showcase

See 80-core demo in tech showcase

The Tera-scale Computing Vision



Parallel Programming Tools & Techniques

Model-Based Applications

Thread-Savvy Execution Environment

Scalable Multi-core Architectures

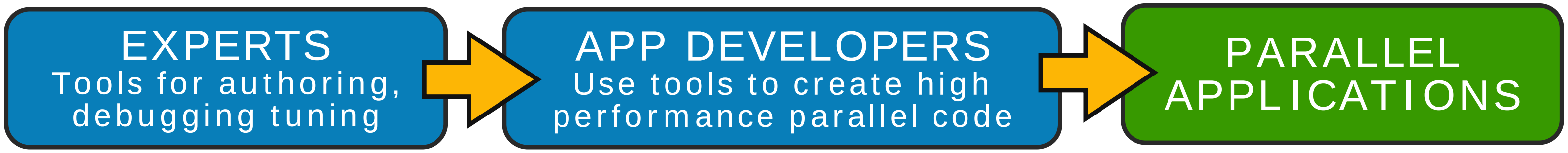
Stacked, Shared Memory

High Bandwidth I/O & Communications

Taking Parallel Programming Mainstream

Used for decades in HPC, parallel programming requires

- Special expertise, not easily automated
- Requires parallel languages/language extensions
- Must co-exist with legacy code

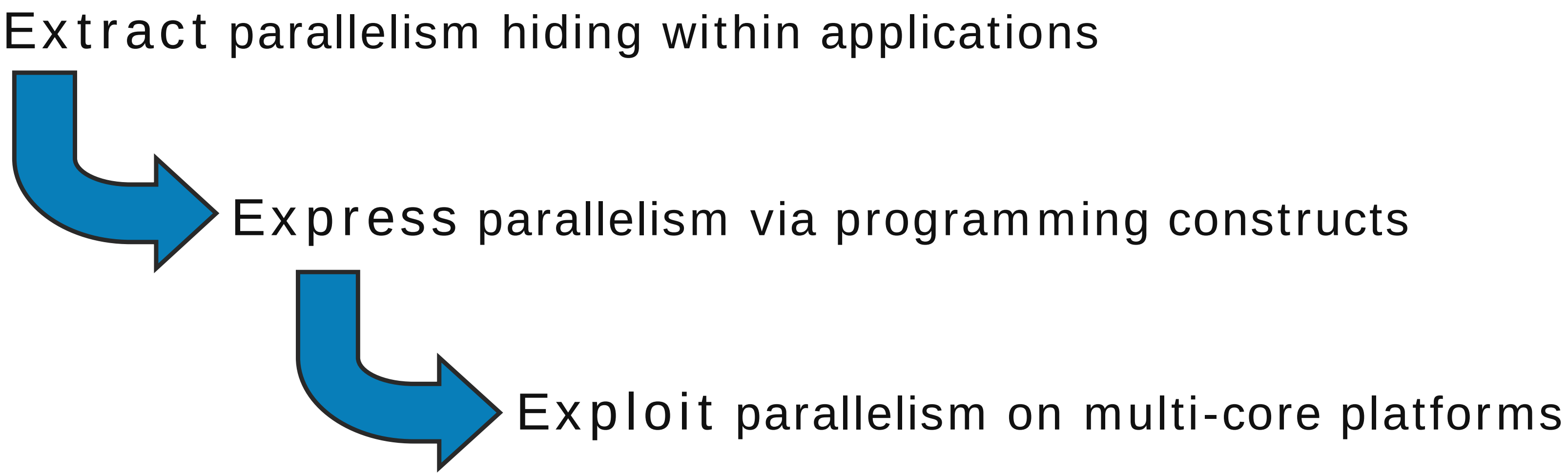


EXPERTS
Tools for authoring, debugging tuning

APP DEVELOPERS
Use tools to create high performance parallel code

PARALLEL APPLICATIONS

Parallel languages or parallel language extensions need to:



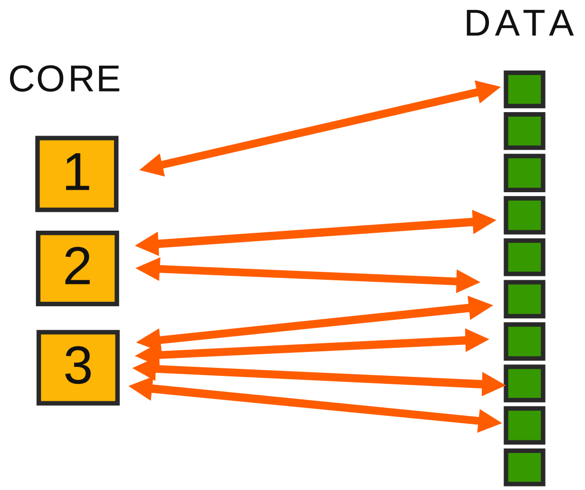
Extract parallelism hiding within applications

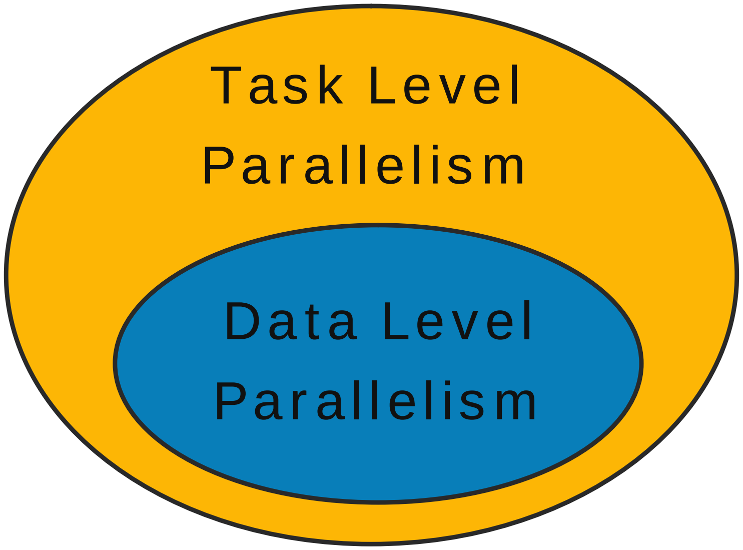
Express parallelism via programming constructs

Exploit parallelism on multi-core platforms

Types of Parallelism

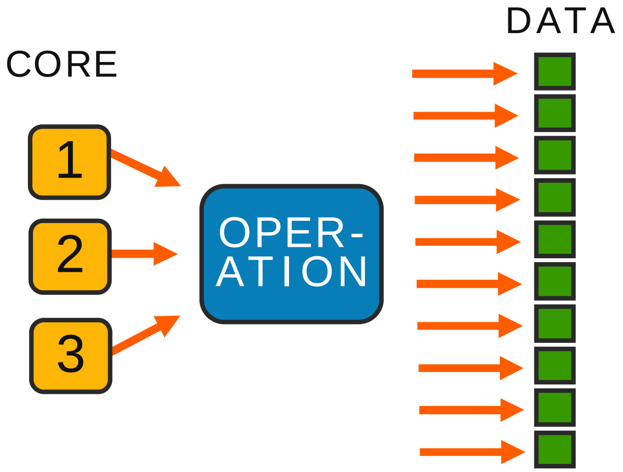
- Task: multiple independent activities, which may or may not share data




Task Level Parallelism

Data Level Parallelism

- Data: one or few tasks operating on a large amount of data
  - Will review Ct later in presentation
  - Subject of demo and class (TCRS0003)



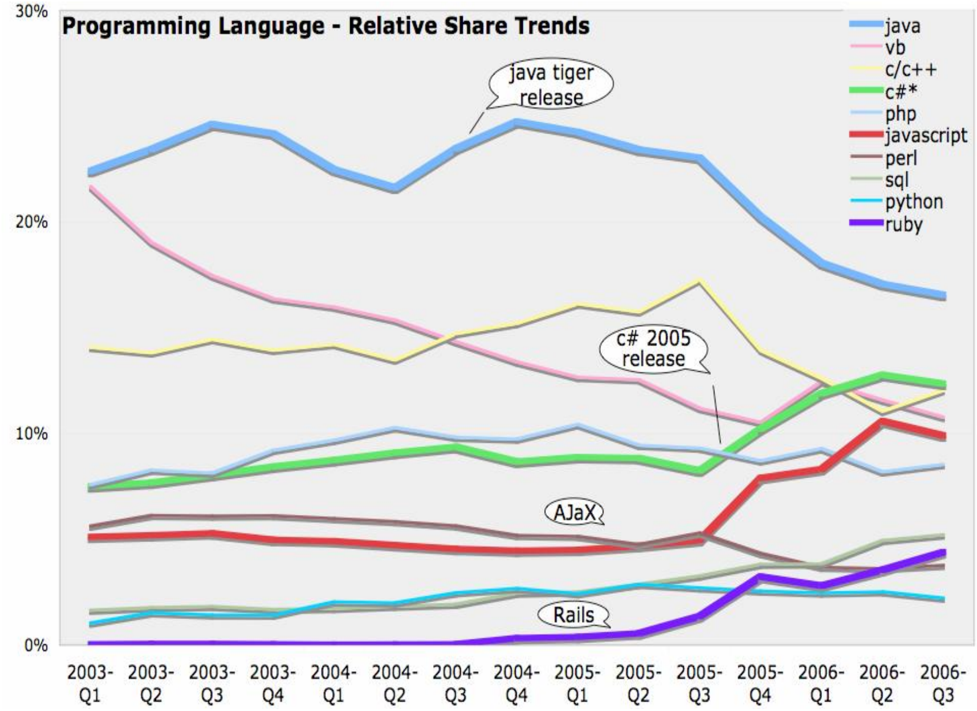
Common Parallel Programming Models

Programming model	Example Application Today	Example Language
Message Passing	HPC type apps on a cluster	MPI
Shared-memory or Thread	General Par Tasks, Database	OpenMP
Data Parallel	Mathematical simulations	MatLab
Streaming	Graphics shaders	C_G

Diverse Programming Environment

Historically, new languages often emerge and compete.

Adoption can be slow and success hard to predict.



Source: Tim O'Reilly Book Index

Lesson: No one language will be a silver bullet that solves the parallel programming challenges

Research to Help Enable Parallelism

- STM (Software Transactional Memory)
  - Shared memory model is dominant but problematic due to synchronization of locks in highly parallel environments
- Ct (C for throughput computing)
  - Relieves need to worry about threads in data parallel execution models providing parallel extensions to C and C++
- Exo (Accelerator Exoskeleton)
  - Bringing IA “look and feel “ of common development environments to parallel, *heterogeneous* environments.

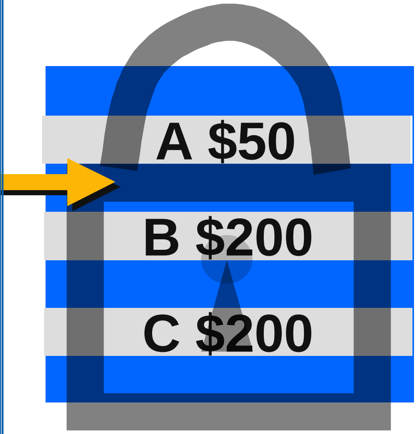
Unlocking Parallelism in a Shared Memory Environment

Must carefully control how multiple threads access shared memory

Today we "lock" memory for one thread at a time.

- Other threads must wait, reducing multi-core benefit
- Locking code scales poorly, must re-do for more threads
- Can cause critical software deadlocks and errors
- We need to fix this...

Account locked during access



A $50

B $200

C $200

2003 Northeast blackout

Courtesy NASA/JPL-Caltech

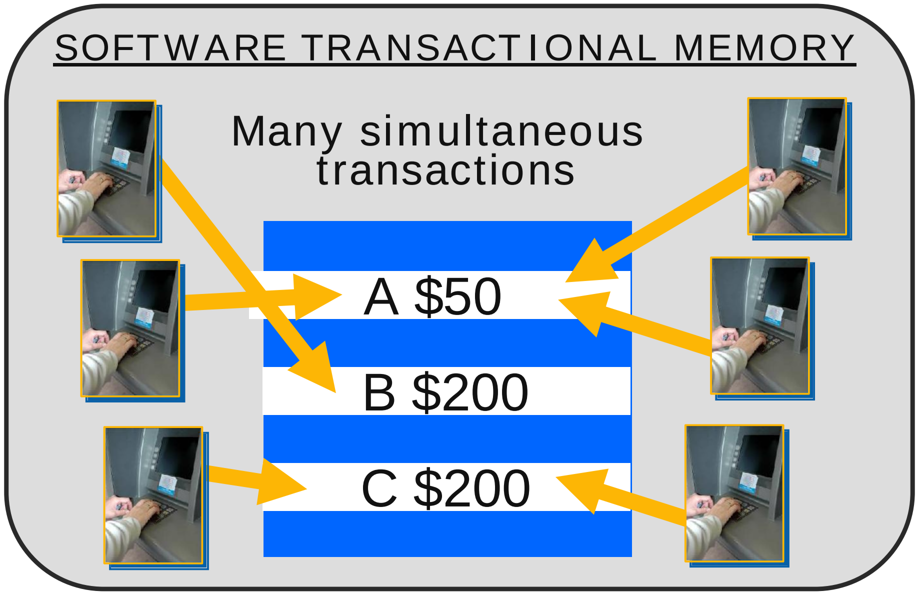
Mars rover problem

Presented by Justin Rattner at IDF Day0, Spring 2006

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

STM: From Research to Reality

Ensures correct parallel memory access without locks



- Greater performance
- Easer to program
- Scales with hardware

Programmers can try it!

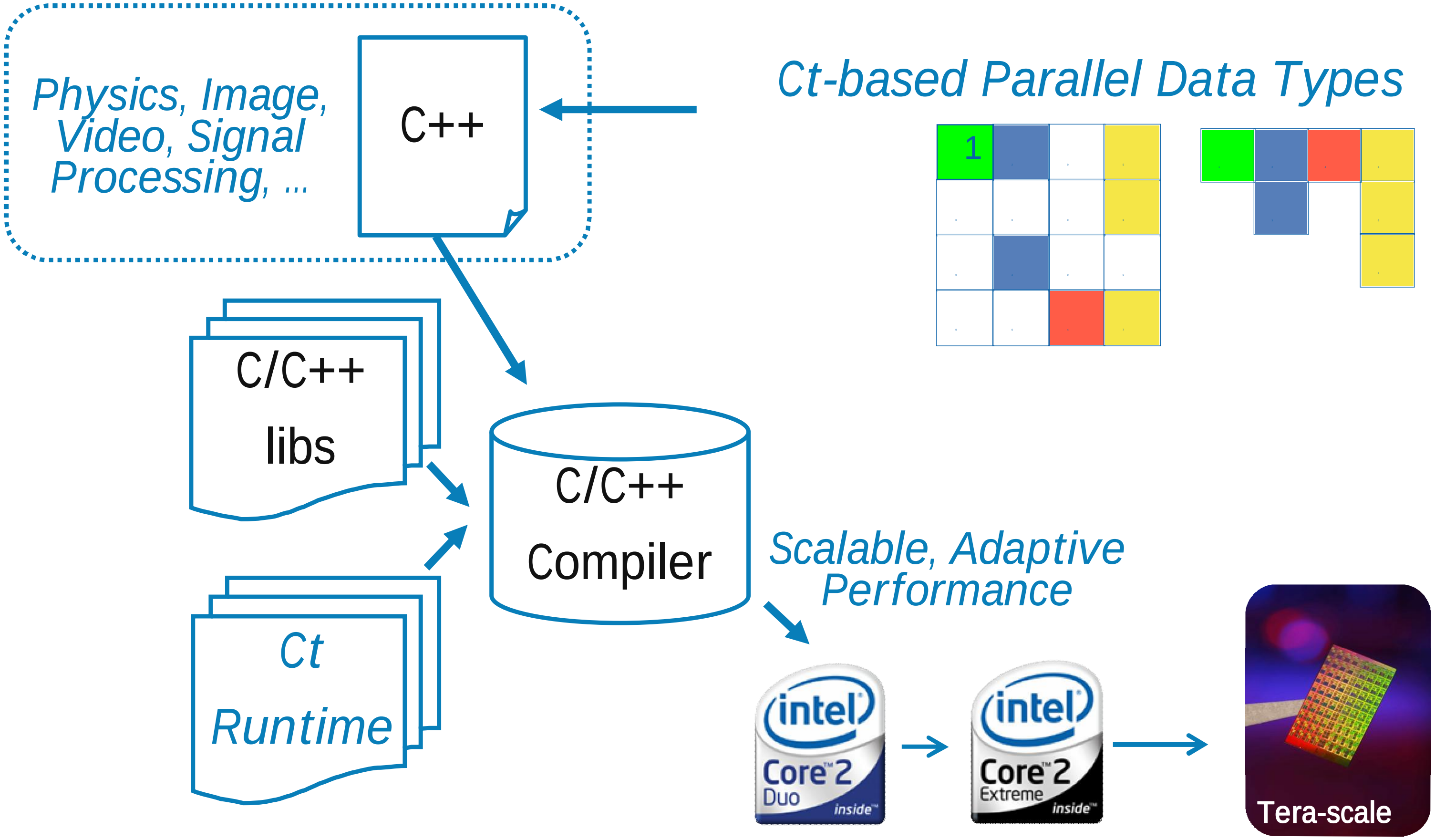
AVAILABLE STARTING TODAY

Whatif.intel.com

Intel® C++ STM Compiler Prototype Edition

Ct: Nested Data Parallel Programming

Ct adds *new parallel* data structures & operators to C/C++;
But uses *existing & unmodified* C/C++ compilers


Ct-based Parallel Data Types
1
Physics, Image, Video, Signal Processing, ...
C++
C/C++ libs
Ct Runtime
C/C++ Compiler
Scalable, Adaptive Performance
intel Core 2 Duo inside
intel Core 2 Extreme inside
Tera-scale

Ct Motivation and Vision

- Make parallel programming easier now:
  - Extend *deterministic* parallel programming models
    - I.e. Data races not possible
  - Express complex behaviors through simple operators
  - Present a simple and predictable performance model
- Provide a forward-scaling programming model that maximizes ISV ROI for new code creation
  - "Future-proof" apps from increasing core count and inevitable ISA evolution

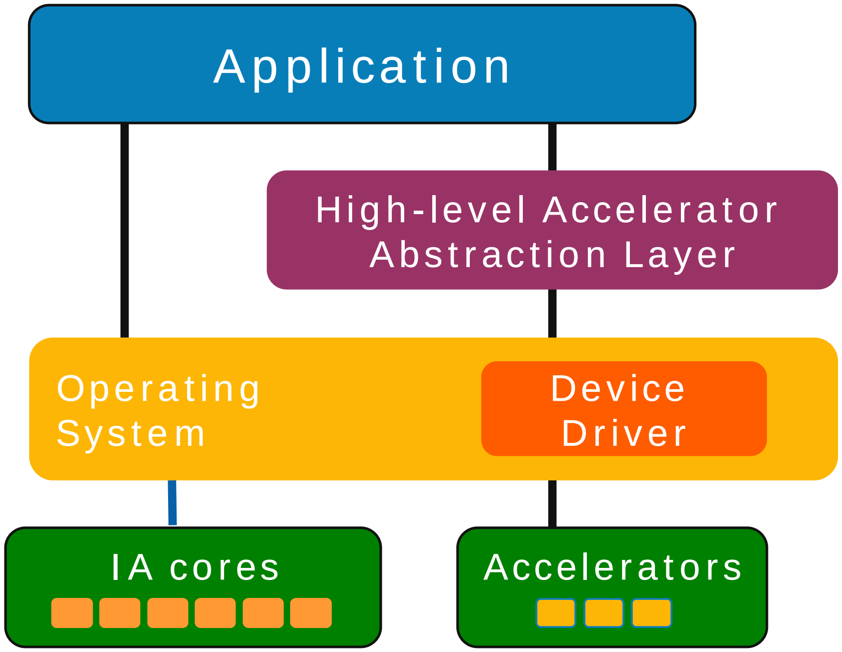
See today's post by Anwar Ghuloum at blogs.intel.com/research for more info

Programming for Heterogeneous Cores

- Multi-core brings the opportunity to integrate fixed function accelerators with IA cores
- Implementation difficult and very HW specific

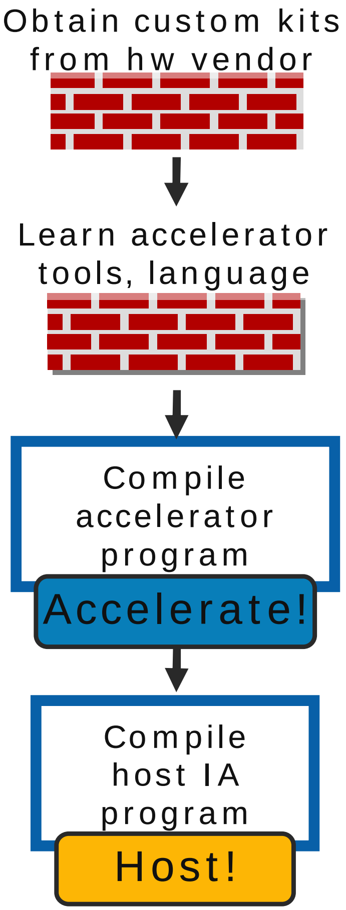
Accelerators today rely on custom drivers bound tightly to the OS.

The software layers between the app accelerator cause performance-limiting overhead



Programming Accelerators Today

SW DEVELOPMENT

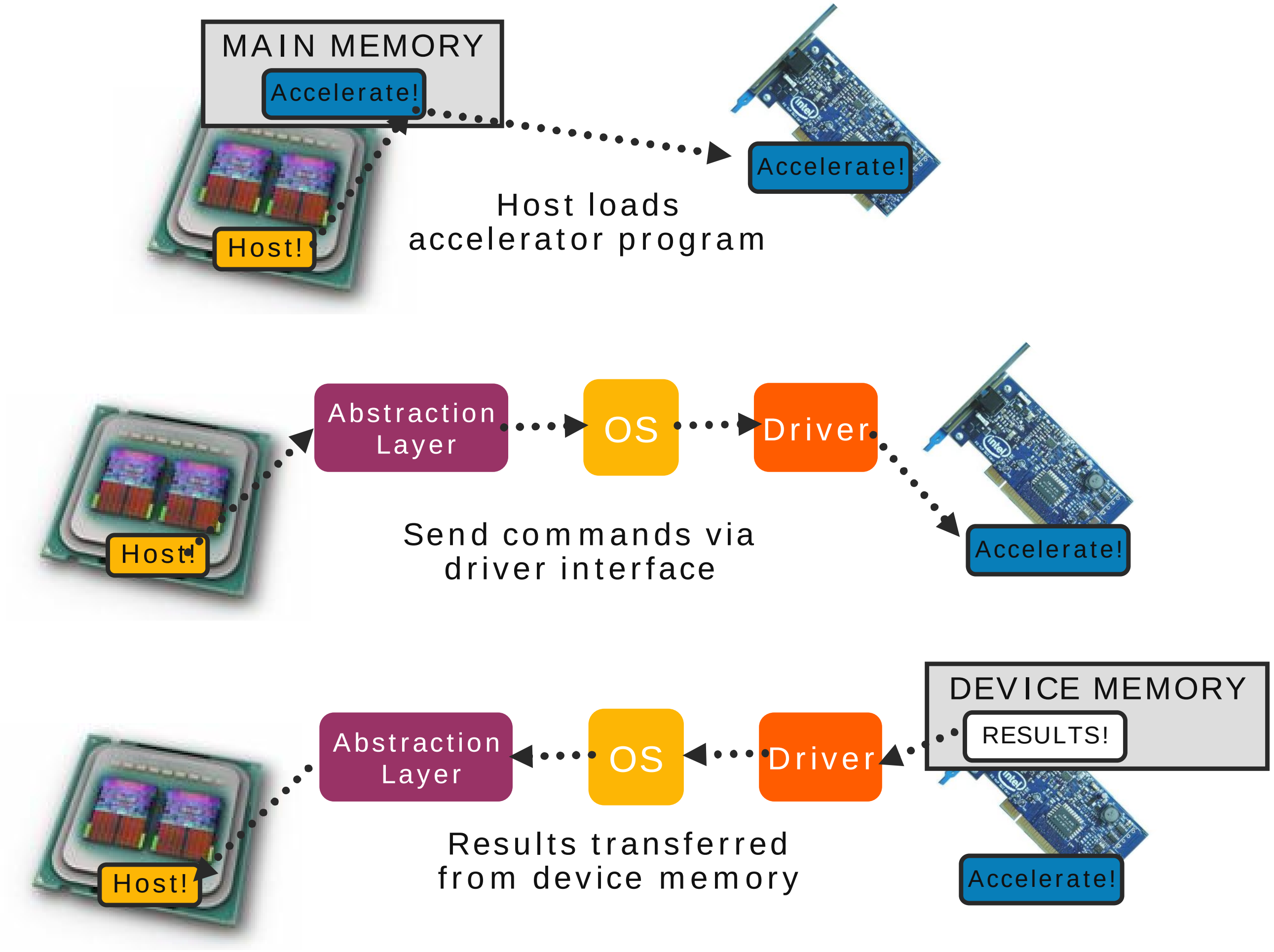

Obtain custom kits from hw vendor

Learn accelerator tools, language

Compile accelerator program

Compile host IA program

SW EXECUTION



Host loads accelerator program

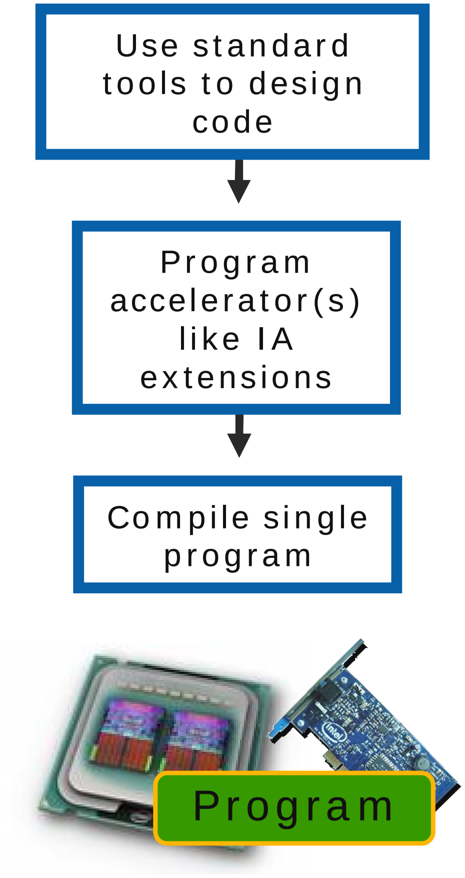
Send commands via driver interface

Results transferred from device memory

Exo - Accelerator Exoskeleton Model

The exoskeleton makes accelerators appear like a part of the processor

SW DEVELOPMENT



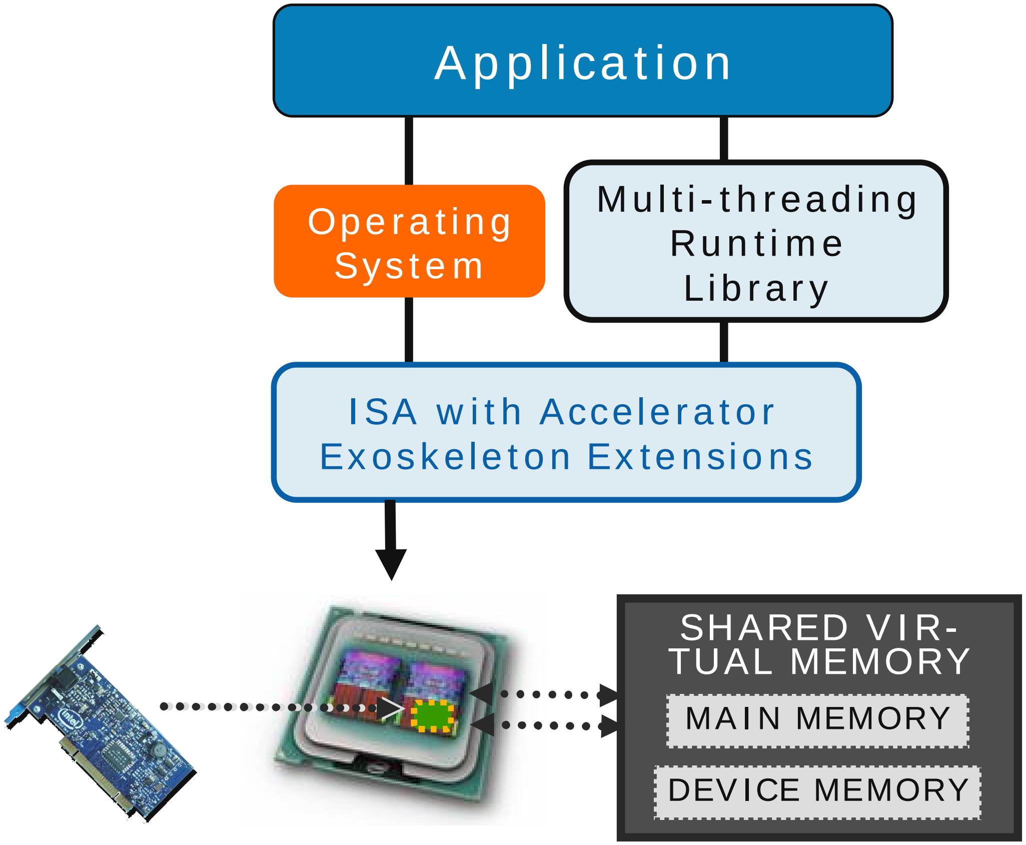
Use standard tools to design code

Program accelerator(s) like IA extensions

Compile single program

Program

SW EXECUTION



Application

Operating System

Multi-threading Runtime Library

ISA with Accelerator Exoskeleton Extensions

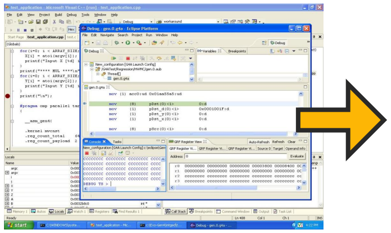
SHARED VIRTUAL MEMORY

MAIN MEMORY

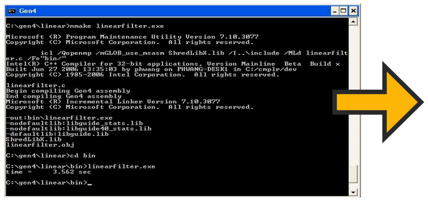
DEVICE MEMORY

Demonstrations

1. Video enhancement application (de-interlacing)

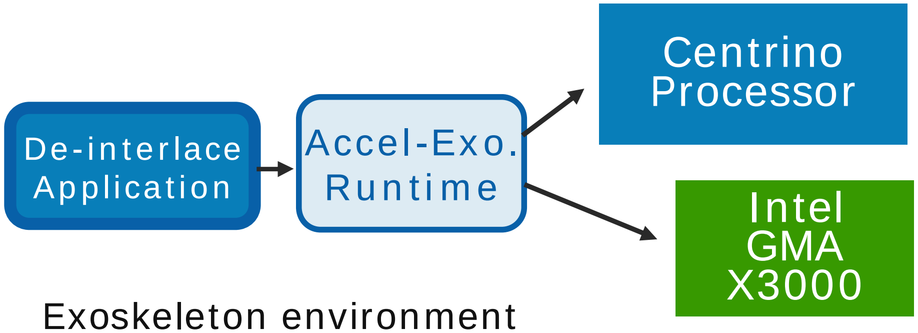


Code for graphics in standard IDE



```
C:\gen4\linear>nmake linearfilter.exe

Microsoft (R) Program Maintenance Utility Version 7.10.3077
Copyright (C) Microsoft Corporation.  All rights reserved.

        icl /Qopenmp /mGLOB_use_msasm ShredLibX.lib /I..\include /MLd linearfilt
er.c /Fe"bin/"
Intel(R) C++ Compiler for 32-bit applications, Version Mainline  Beta  Build x
Built Jun 27 2006 13:35:03 by phwang on PHWANG-DESK1 in C:/cmplr/dev
Copyright (C) 1985-2006 Intel Corporation.  All rights reserved.

linearfilter.c
Begin compiling Gen4 assembly
End compiling Gen4 assembly
Microsoft (R) Incremental Linker Version 7.10.3077
Copyright (C) Microsoft Corporation.  All rights reserved.

-out:bin\linearfilter.exe
-nodefaultlib:libguide_stats.lib
-nodefaultlib:libguide40_stats.lib
-defaultlib:libguide.lib
ShredLibX.lib
linearfilter.obj

C:\gen4\linear>cd bin

C:\gen4\linear\bin>linearfilter.exe
time =      3.562 sec

C:\gen4\linear\bin>_
```

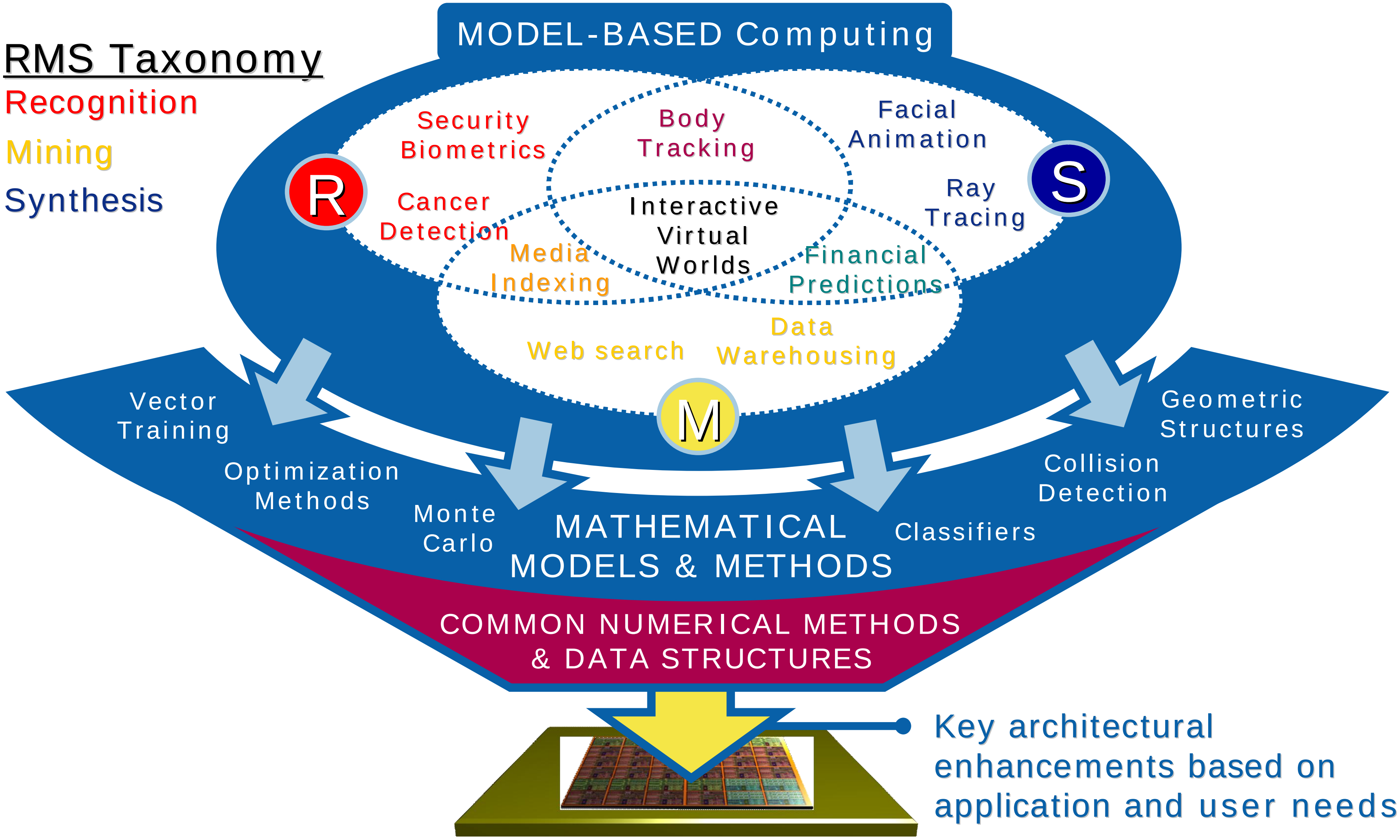
Use Intel compiler to build a single ".exe"



Exoskeleton environment divides the work

2. Similar demo with a financial analytics application

Convergence of Many Parallel Apps



RMS Taxonomy

Recognition

Mining

Synthesis

MODEL-BASED Computing

R

S

M

Security Biometrics

Body Tracking

Facial Animation

Cancer Detection

Interactive Virtual Worlds

Ray Tracing

Media Indexing

Financial Predictions

Web search

Data Warehousing

Vector Training

Optimization Methods

Monte Carlo

MATHEMATICAL MODELS & METHODS

Classifiers

Collision Detection

Geometric Structures

COMMON NUMERICAL METHODS & DATA STRUCTURES

Key architectural enhancements based on application and user needs

Independent IDC Market Analysis

- Spending on computing continues to grow

- New usage models are emerging

- IDC working with Intel and others to understand this space
  - *"IDC believes that new use cases for computing are emerging which will drive significant growth…Our research has shown that highly parallel applications and multi-core processors are key drivers enabling this emerging trend."*

    -Matt Eastwood, Group Vice President, IDC Enterprise Platform Research

Full report to be published in October

Summary

- The key challenges to parallel programming
  - Programmability and scalability
  - Interoperability and integration of heterogeneous execution blocks
- Parallel programming is the key to unlocking the full potential of the Tera-scale platforms
  - Mainstream programmers enabled through broad set of development and optimization tools.
  - Technologies developed to extend well-known programming environments (Ct, Exo and STM)
  - Tools and technologies are already being deployed now:
    - TBB – threadingbuildingblocks.org
    - STM on Whatif.intel.com
- Model-based computing applications are compelling with market projections supporting potential for broad adoption

More information on Tera-scale…

Tera-scale Computing Research Chalk Talk

Chair: Jerry Bautista, Co-Director, Tera-scale Computing Research

TCRC001: Wed, Sept. 19, 4:40 - 5:30, Chalk Talk Room

Other Tera-scale Sessions (see IDF guide for full info)

- TCRS001 Energy Management Innovations for Future Multi-Core Processors
- TCRS002 Intelligent On-chip Interconnects: The 80-core Prototype and Beyond
- TCRS003 Data Parallel Programming for Tera-scale with Ct
- TCRS004 Modeling Reality: Ray Traced Graphics and other Apps of the Future
- TCRS005 Silicon Photonics: Enabling Terabit Data Pipes
- QATS003 Accelerator Exoskeleton: IA Look-n-Feel for Heterogeneous Cores

Tech & Research Pavilion Demos:

80-core Processor, Multi-core emulator, Ct, Log-based Architectures, Ray Tracing

Learn more at www.intel.com/go/terascale and blogs.intel.com/research