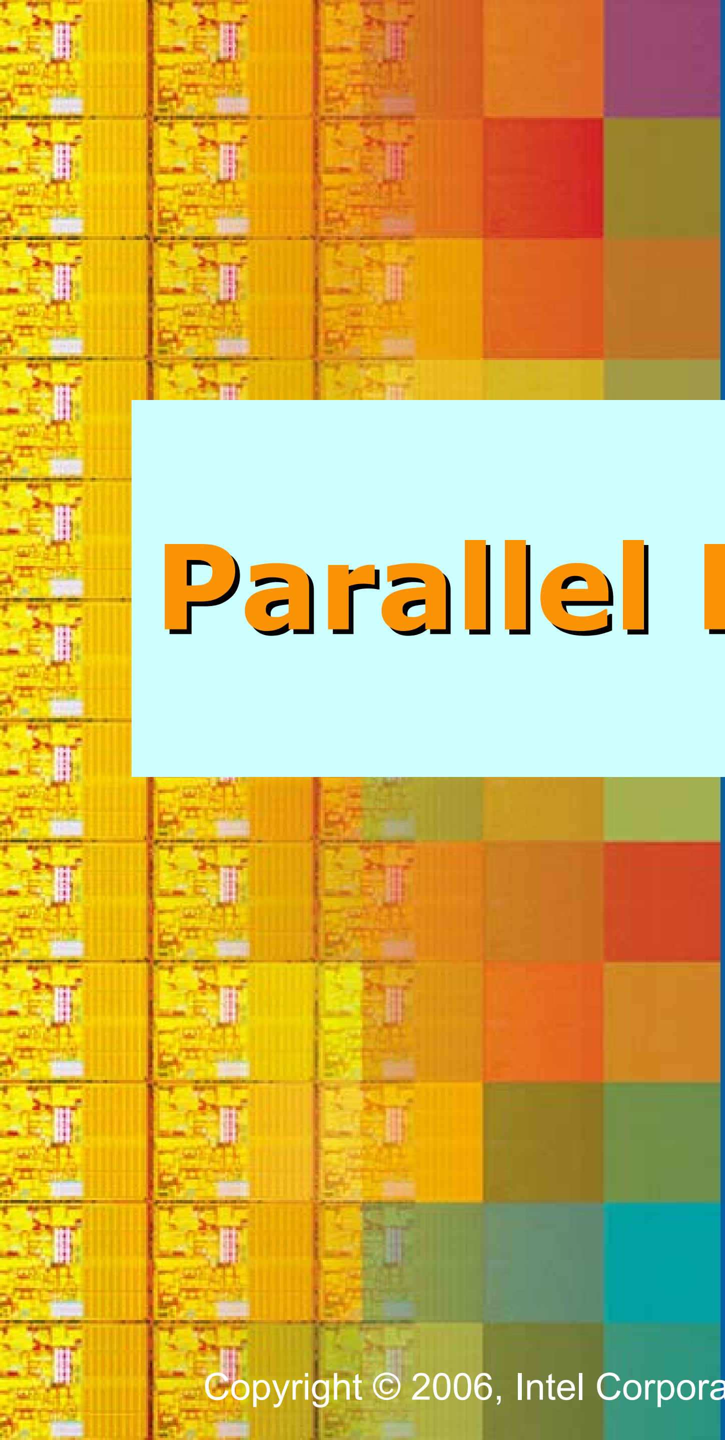


Parallel Programming 2.0

Wei Li
Senior Principal Engineer
Software and Solutions Group
Intel Corporation

CGO’06 Keynote 3/27/06

Agenda

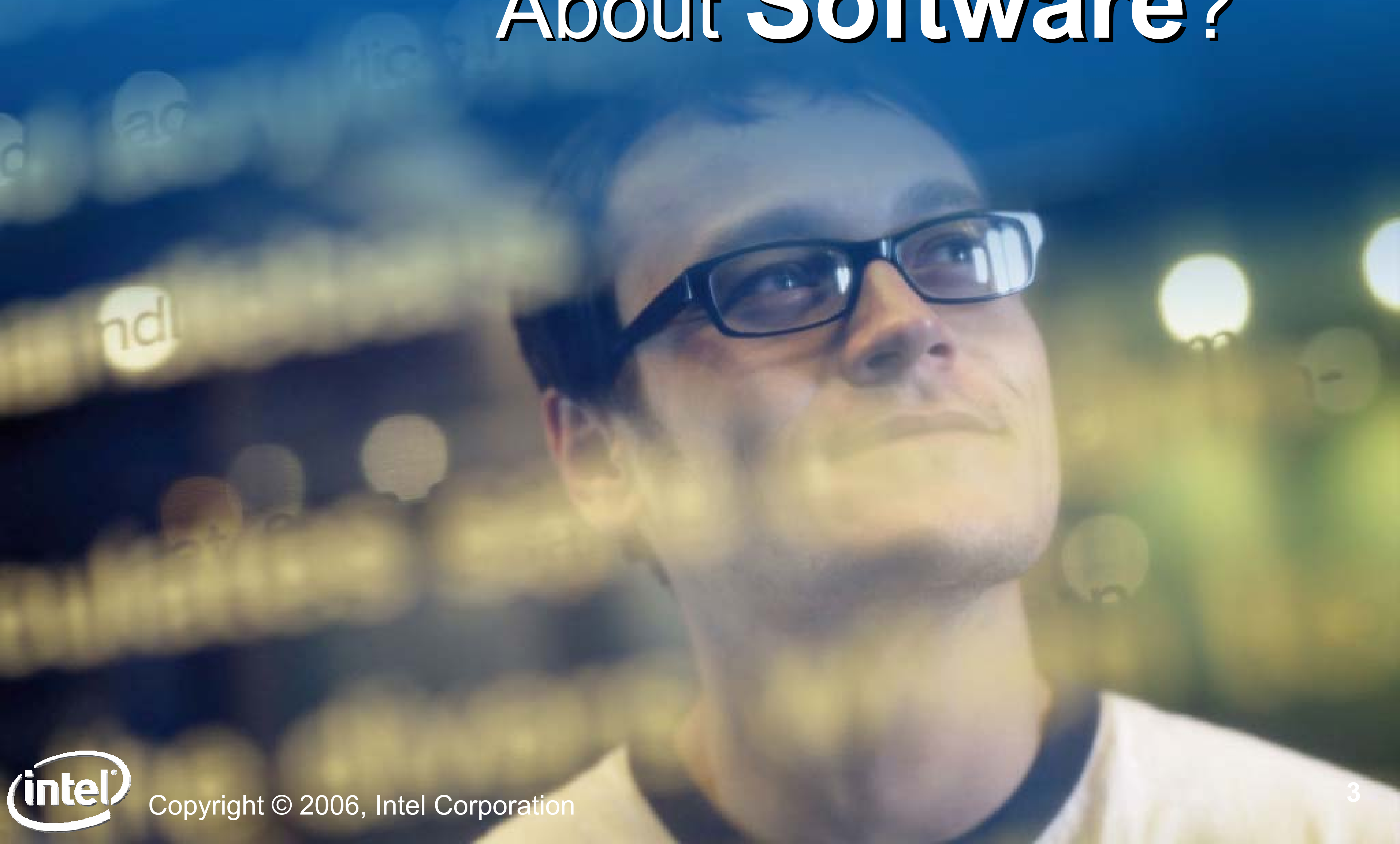
Software at Intel

Major Technological Change

Software Response

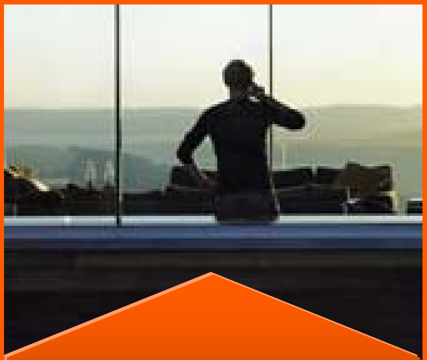
Parallel Programming 2.0

Why Does Intel **Care** About **Software**?



Software At Intel®

Development Products



Enabling

Platform Software

Professional Services

Devel
Produ

Intel® VTune™ Analyzers
Intel® Compilers
Intel® Performance Libraries
Intel® Cluster Toolkit
Intel® Threading Tools
Featuring Intel® Thread Checker and Intel® Thread Profiler

Envir

Devel
Produ

OS’s

Architectures

All Architectures Supported Across All Environments

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

Enabling...

- Intel® Software Network
- Intel® Software College
- Intel® Early Access Program
- Industry Programs and Alliances
- Intel® Competency Centers
- Intel® Software Research
- Intel® Press Publications

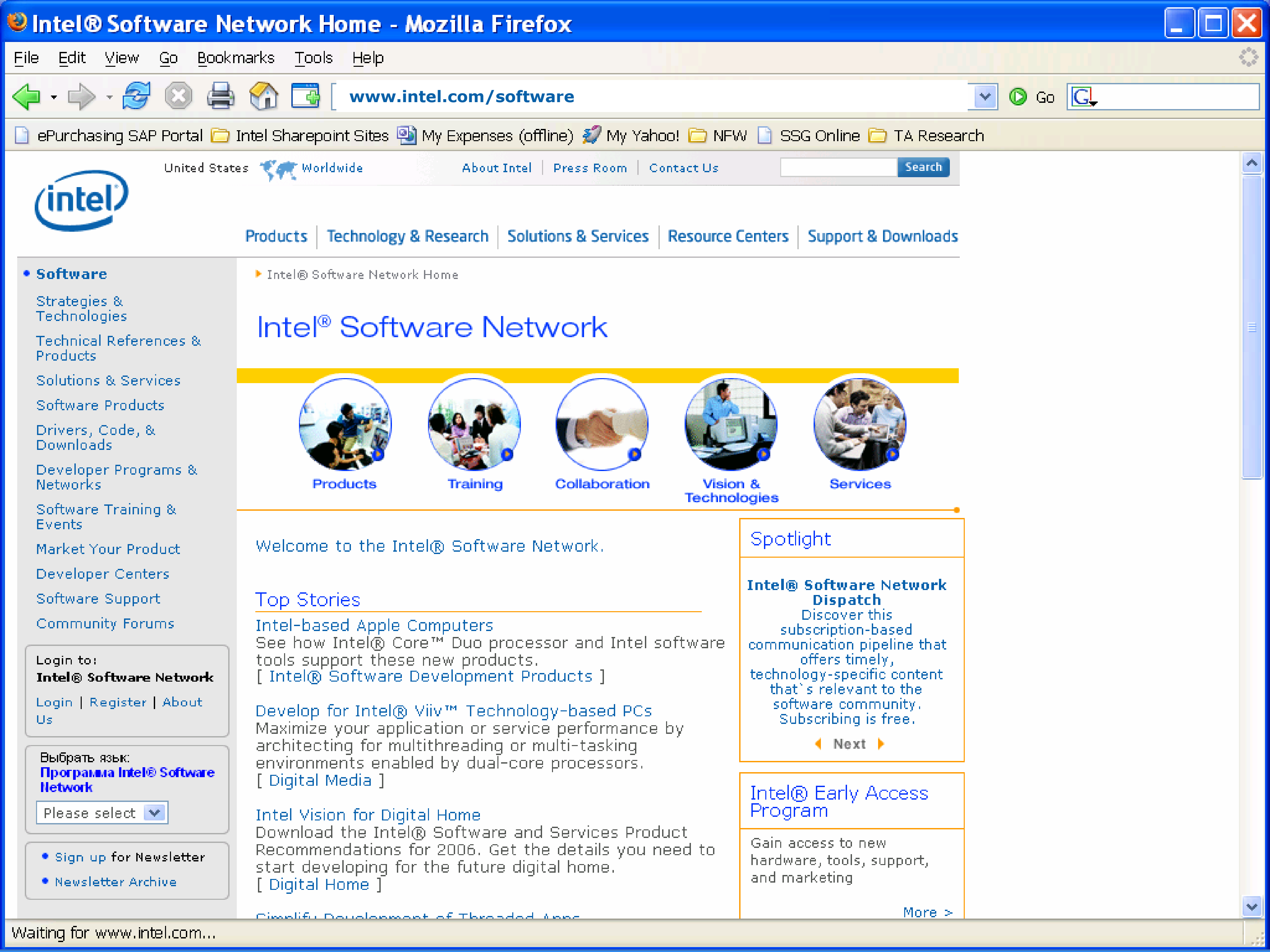
www.intel.com/software

Intel® Software Network

functions. Check out the newly released Intel® Integrated Performance Primitives (Intel® IPP) Library 5.0 for Windows* and Linux*.
[Tools]

Open Source Game Development

Get an overview of Intel® compilers and how to utilize their major optimization option. Find out more.
[Intel® Software College]



Intel® Software Network

Welcome to the Intel® Software Network.

Top Stories

Intel-based Apple Computers

See how Intel® Core™ Duo processor and Intel software tools support these new products.

[Intel® Software Development Products]

Develop for Intel® Viiv™ Technology-based PCs

Maximize your application or service performance by architecting for multithreading or multi-tasking environments enabled by dual-core processors.

[Digital Media]

Intel Vision for Digital Home

Download the Intel® Software and Services Product Recommendations for 2006. Get the details you need to start developing for the future digital home.

[Digital Home]

Simplify Development of Threaded Apps

Spotlight

Intel® Software Network Dispatch

Discover this subscription-based communication pipeline that offers timely, technology-specific content that`s relevant to the software community. Subscribing is free.

Next

Intel® Early Access Program

Gain access to new hardware, tools, support, and marketing

More >

Login to:

Intel® Software Network

Login | Register | About Us

Выбрать язык:

Программа Intel® Software Network

Please select

• Sign up for Newsletter

• Newsletter Archive

….With Worldwide Reach

Over **50** Development **Sites**

Across More Than **20 Countries**

Scaling Innovation on Intel’s Leading Platforms and Tapping the Best Talent Around the World

Hardware

+ Platform Software

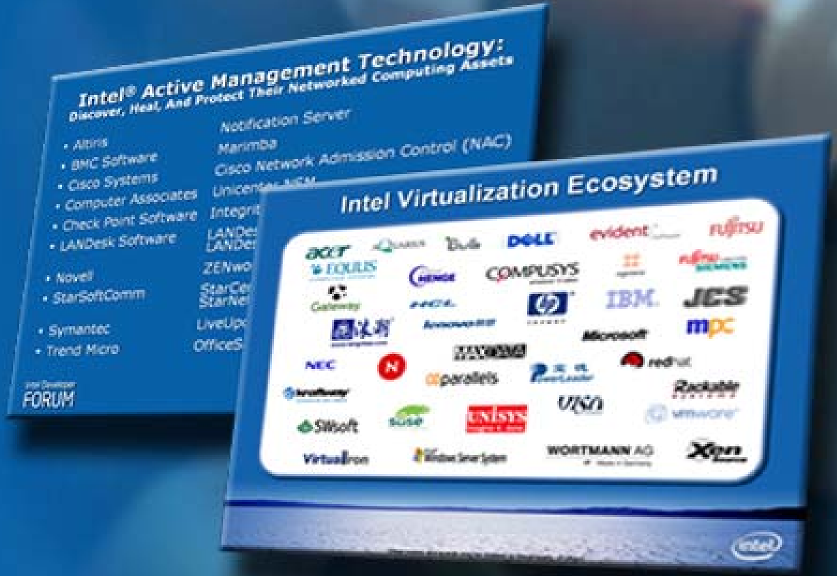
+ ISV Value Add

Platform Solutions

Platform Software

- Intel® Active Management and Intel® Virtualization Technology Solutions
  - Rapid adoption throughout ecosystem
  - Many companies bringing products to market

- Carrier Grade Linux for Telecommunications with AdvancedTCA
  - Both hardware and software technology building blocks available
  - Commercial deployments already underway worldwide



Intel® Active Management Technology:
Discover, Heal, And Protect Their Networked Computing Assets

- Altiris
- BMC Software
- Cisco Systems
- Computer Associates
- Check Point Software
- LANDesk Software
- Novell
- StarSoftComm
- Symantec
- Trend Micro

Notification Server
Marimba
Cisco Network Admission Control (NAC)
Unicen
Integrit
LANDes
LANDes
ZENwo
StarCe
StarNe
LiveUp
OfficeS

Intel Developer FORUM

Intel Virtualization Ecosystem

You Are Here: Home » Lab Activities » Carrier Grade Linux

Carrier Grade Linux

Traditionally, communications and data service networks were built on proprietary platforms that had to meet very specific requirements in areas such as availability, reliability, performance, and service response time. Those proprietary systems were [illegible]

Carrier Grade Linux (CGL) stands at the center of the move to open architectures. [illegible] Requirements Definition.

Platform Software

- Intel® Platform Administration Technology
  - For internet café's and SMB installations

- Tiano / EFI
  - Apple using Tiano and EFI to boot Intel-based Macs
  - Embedded Tiano to replace PC BIOS for initialization and management of high performance clusters

Intel® Platform Administration Technology

Intel® Platform Administrator

overview | product | promotion | support | user registration

Products: Processors, Chipsets, Motherboards, Wireless, A-Z Product Listing, Provide Feedback

Intel® Pentium® D Processor

[illegible] Learn more

Latest News

Intel® Platform Administrator V1.1 release

Faster software provisioning

Better game experience

More friendly user interface

Stronger network management

Origin of Intel® Platform Administration Technology

As the information industry continues to expand in the 21st century, networked computer environments such as internet cafes and e-classrooms have been increasing rapidly. But as networks become larger and their usage models more complex, efficient management and maintenance becomes both increasingly difficult and essential.

Intel has always been dedicated to driving the technological development of the information industry, and after performing thorough market research and analysis of the needs of iCafe administrators, Intel set out to develop Intel® Platform Administration Technology as the ideal solution to the problems facing these large-scale networked computing environments.

Related Links
- Intel® Platform Administrator v1.1 download
- FAQ download
- White Paper v1.1 download
- Product Demo download

Intel® Platform Administration Technology Enabled Desktop Board

Intel® Desktop Board

Intel® Platform Administration Technology

TianoCore

User: Password: Login

Register | Learn more

Welcome to TianoCore.org

TianoCore is the community surrounding the open source components of Intel's implementation of EFI, officially called the Platform Innovation Framework referred to as "the Framework". (To learn more about EFI, please click here.) These components comprise the main projects hosted here at TianoCore.org. The TianoCore community is growing by the day and could use more help in driving and developing the Framework and its capabilities. To learn more about getting involved in the community, click here.

About the Community

There is a lot to learn about the TianoCore community. Without actually joining the community, you still have access to the documentation, source and binaries available on TianoCore. However, your involvement will be limited to read-only access to the tools provided by the community. The real magic of any community is when you join and have full access to the tools that the community drivers use to evolve and make their application and its accompanying documentation and tooling. As a community member, you will have access to the following:

- Forums - Used for a traceable dialog between community members.
- Issue Tracking - This helps keep track of feature requests, tasks and issues with the respective project.
- Mailing Lists - Like forums but handled via email instead of web forms.
- RSS Feeds - You can subscribe to RSS feeds for news, forums and mailing lists.
- Source Control (CVS) - projects use these to keep track of different versions of software source code.

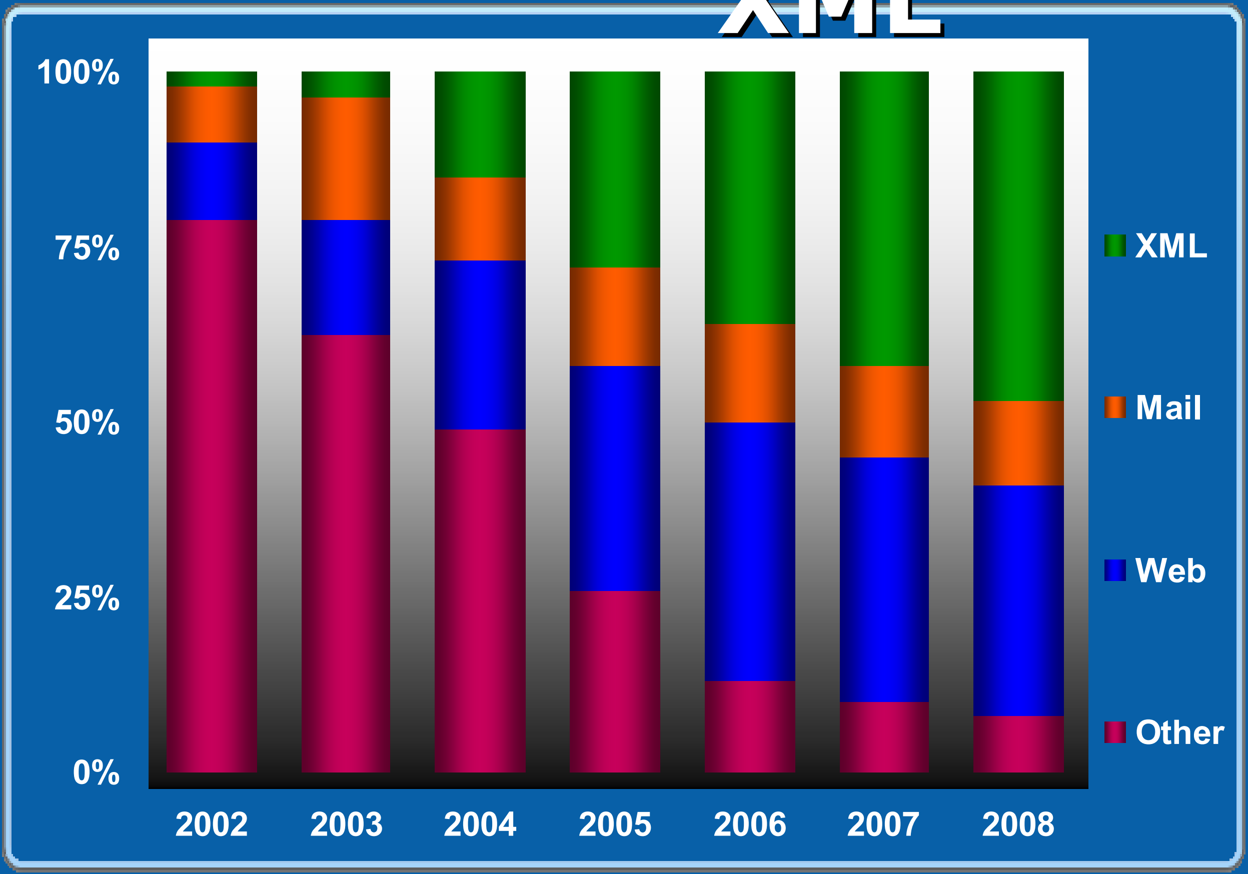
As you can see, there is a lot at your disposal as a community member. Excited yet? If so, learn how to get involved here.

Word on the Street

Our Community at Work

Useful Internet Links

The Next Platform Workload: XML



CONFORMATIVE ‹SYSTEMS›

Sarvega

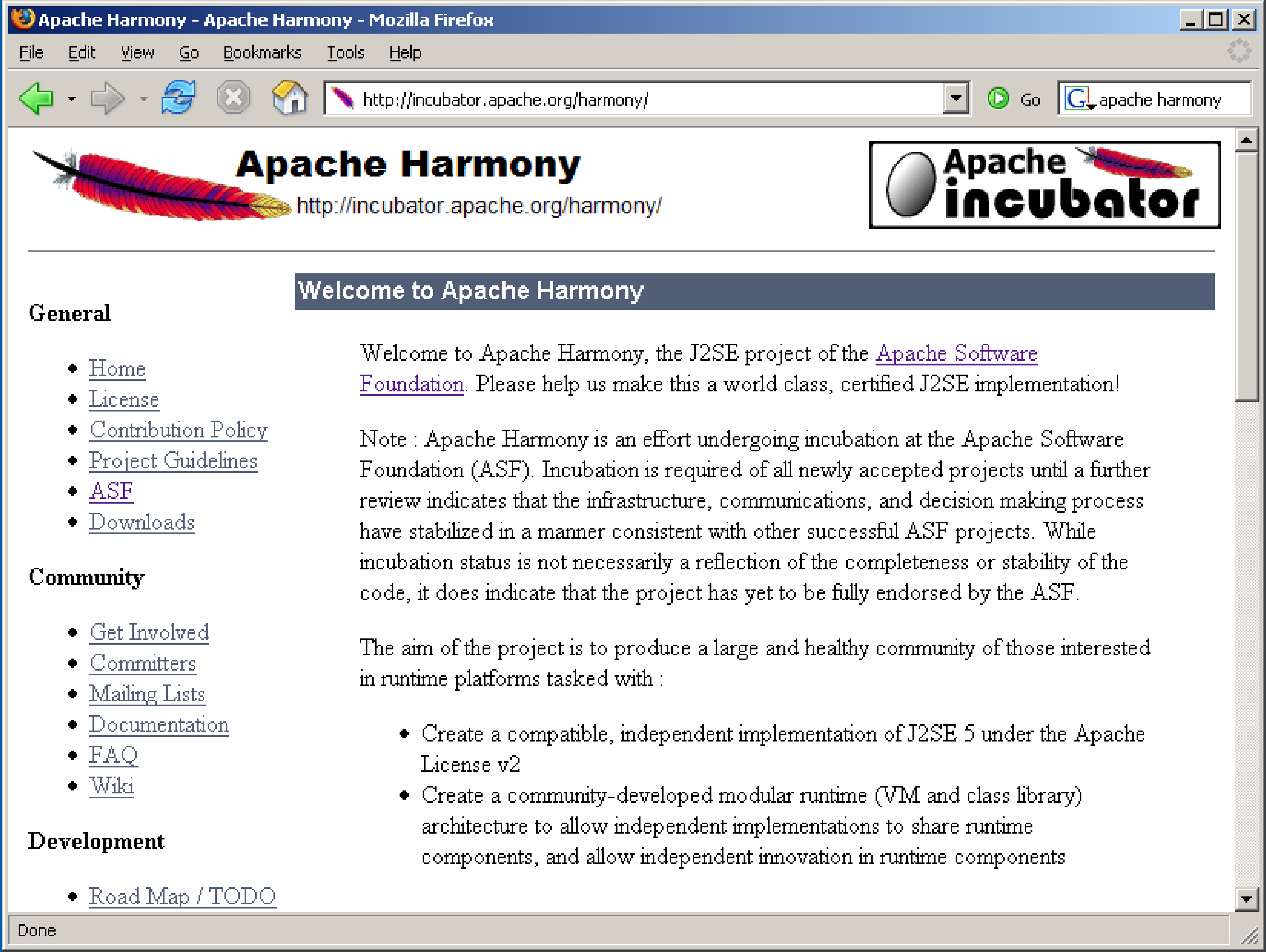
XML Traffic Will Exceed:

All Mail Traffic in 2004 ● All Web Traffic in 2007

Source: Zapthink: Solving the very large messaging problem in the enterprise, February 2005

Managed Runtime

- Develop the best available Java and .NET technology for Intel® hardware
  - help JVMs run better on parallel hardware
  - JITs generate great code for Intel® micro-architectures
- Example projects
  - Apache Harmony open-source JVM. Goals:
    - Create an open-source, compatible implementation of J2SE 5 under the Apache License
    - Create a community to carry the open J2SE forward
  - NuMA-cc Awareness for JVMs; object co-location
  - Profile collection and management
  - Dynamic profile-guided optimization


Apache Harmony - Apache Harmony - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://incubator.apache.org/harmony/
Go
apache harmony
Apache incubator
Done

Apache Harmony

http://incubator.apache.org/harmony/

Welcome to Apache Harmony

General

- Home
- License
- Contribution Policy
- Project Guidelines
- ASF
- Downloads

Community

- Get Involved
- Committers
- Mailing Lists
- Documentation
- FAQ
- Wiki

Development

- Road Map / TODO

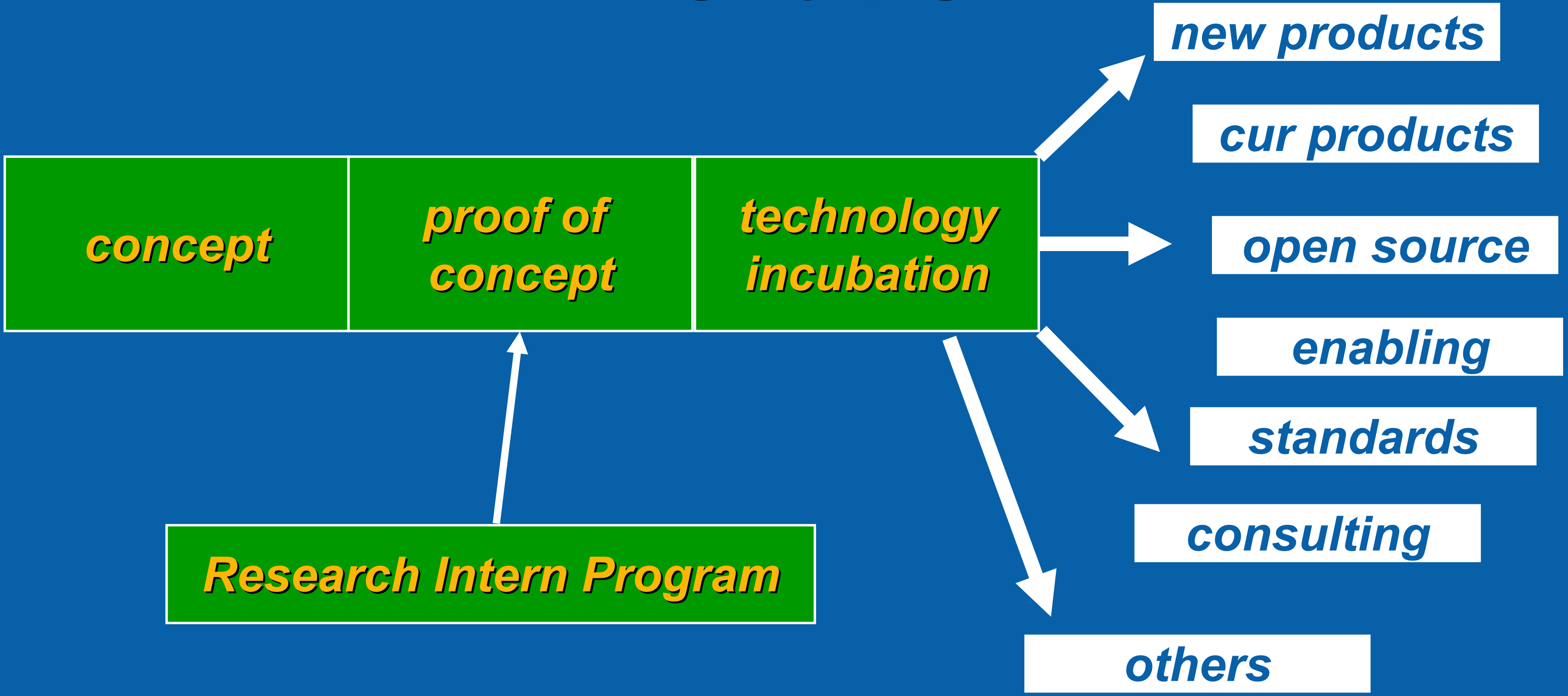
Welcome to Apache Harmony, the J2SE project of the Apache Software Foundation. Please help us make this a world class, certified J2SE implementation!

Note : Apache Harmony is an effort undergoing incubation at the Apache Software Foundation (ASF). Incubation is required of all newly accepted projects until a further review indicates that the infrastructure, communications, and decision making process have stabilized in a manner consistent with other successful ASF projects. While incubation status is not necessarily a reflection of the completeness or stability of the code, it does indicate that the project has yet to be fully endorsed by the ASF.

The aim of the project is to produce a large and healthy community of those interested in runtime platforms tasked with :

- Create a compatible, independent implementation of J2SE 5 under the Apache License v2
- Create a community-developed modular runtime (VM and class library) architecture to allow independent implementations to share runtime components, and allow independent innovation in runtime components

Software Technology Innovation



concept

proof of concept

technology incubation

new products

cur products

open source

enabling

standards

consulting

others

Research Intern Program

Internal Projects as well as Joint Research with Universities

Agenda

Software at Intel

Major Technological Change

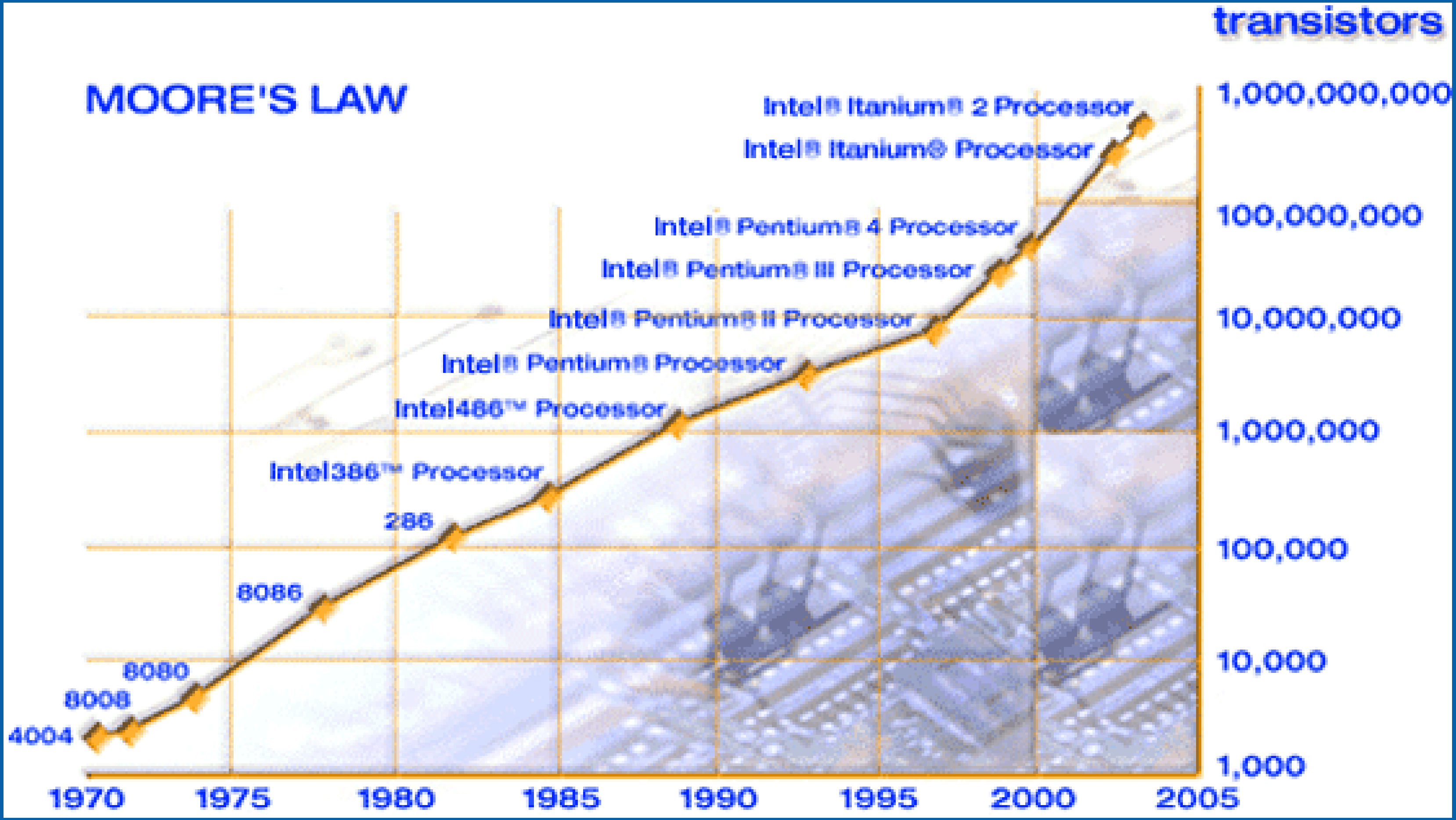
Software Response

Parallel Programming 2.0

Quiz

Moore’s law states which of the following roughly doubles every 2 years?

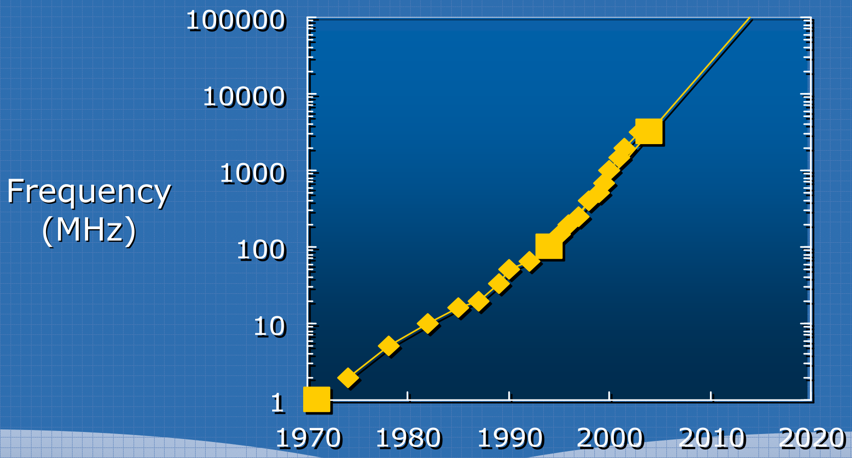
1. Frequency
2. Performance
3. Transistors
4. Transistor Density



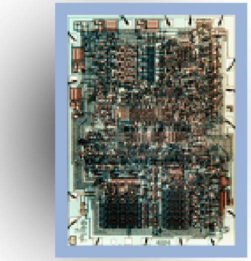
from www.intel.com

Historical Driving Force

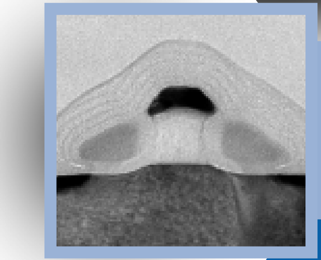
Increased Performance via Increased Frequency



1946
20 Numbers
in Main Memory



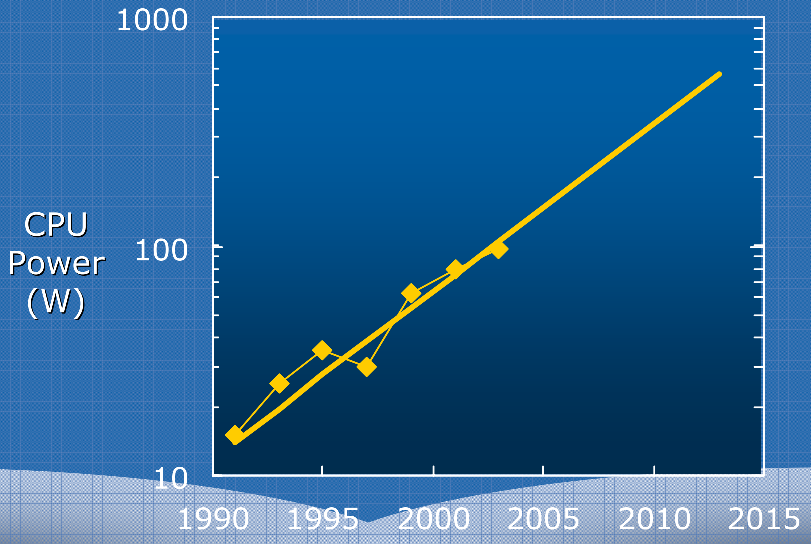
1971
I4004 Processor
2300 Transistors



2005
65nm
1B+ Transistors

The Challenge

Power Limitations

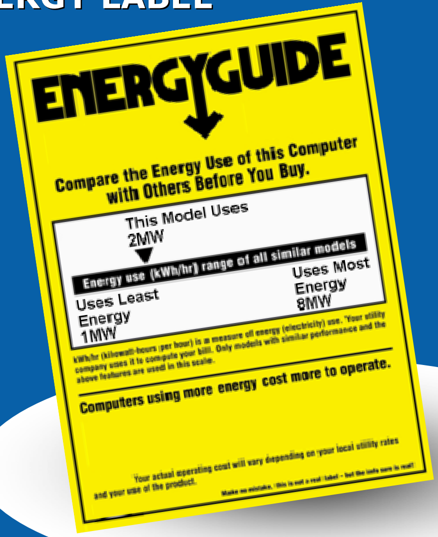


Power = Capacitance x Voltage^2 x Frequency
also
Power ~ Voltage^3

Energy:
The Next Frontier

Energy Efficient Performance – High End

DATACENTER
“ENERGY LABEL”



ENERGYGUIDE

Compare the Energy Use of this Computer with Others Before You Buy.

This Model Uses
2MW

Energy use (kWh/hr) range of all similar models

Uses Least Energy
1MW

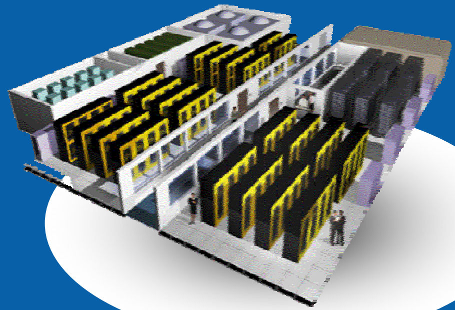
Uses Most Energy
8MW

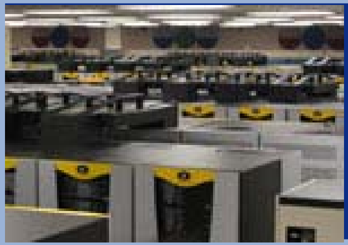
kWh/hr (kilowatt-hours per hour) is a measure of energy (electricity) use. Your utility company uses it to compute your bill. Only models with similar performance and the above features are used in this scale.

Computers using more energy cost more to operate.

Your actual operating cost will vary depending on your local utility rates and your use of the product.

Make no mistake, this is not a real label – but the info sure is real!





NASA Columbia
2 MWatt
60 TFlops goal
10,240 cpus – Itanium II
$50M

Source: NASA

30,720 Flops/Watt
1,288 Flops/Dollar

Computational Efficiency

7,066 Flops/Watt
467 Flops/Dollar

ASC Purple
6 MWatt
100 TFlops goal
12K+ cpus – Power5
$230M

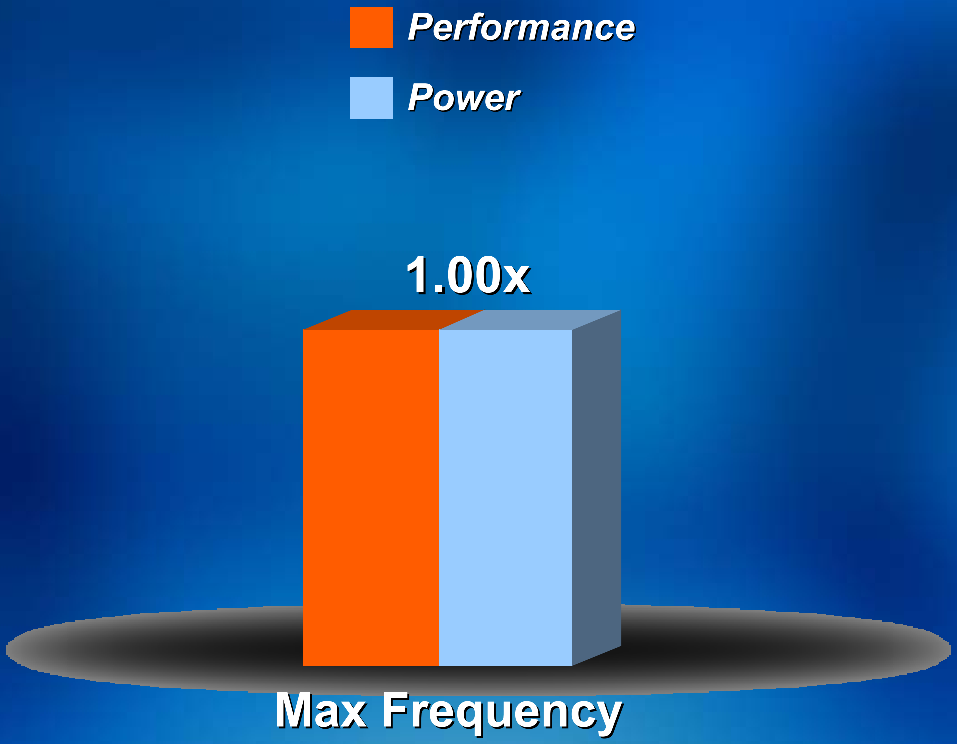
Source: LLNL

The Classic Tradeoff

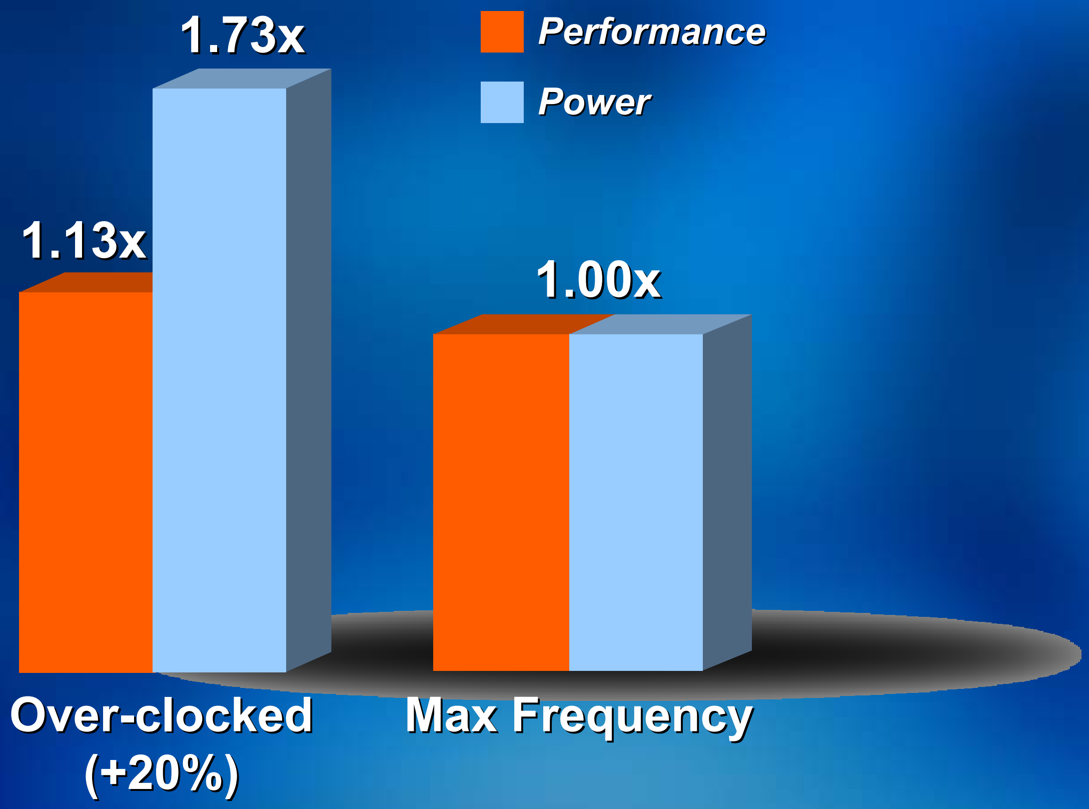
Higher
Top Speed and Acceleration

Increased
Range and Economy

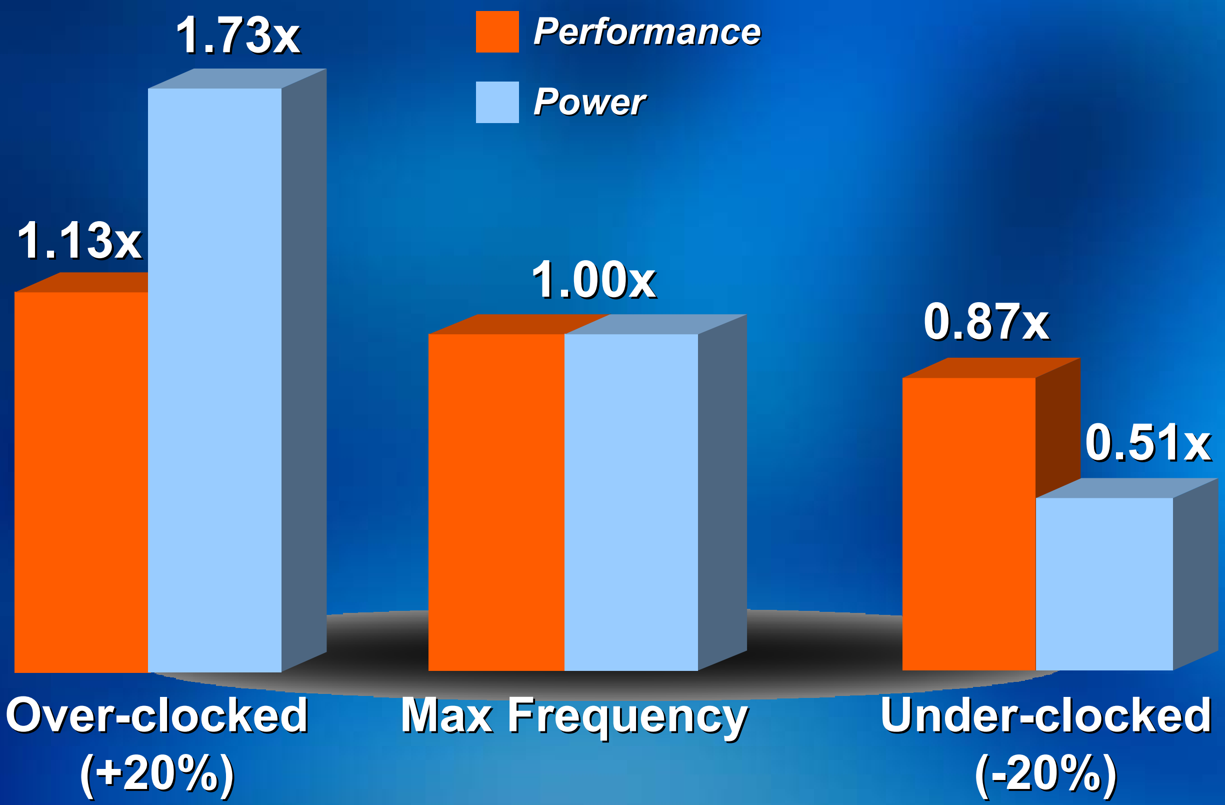
A Simple Example



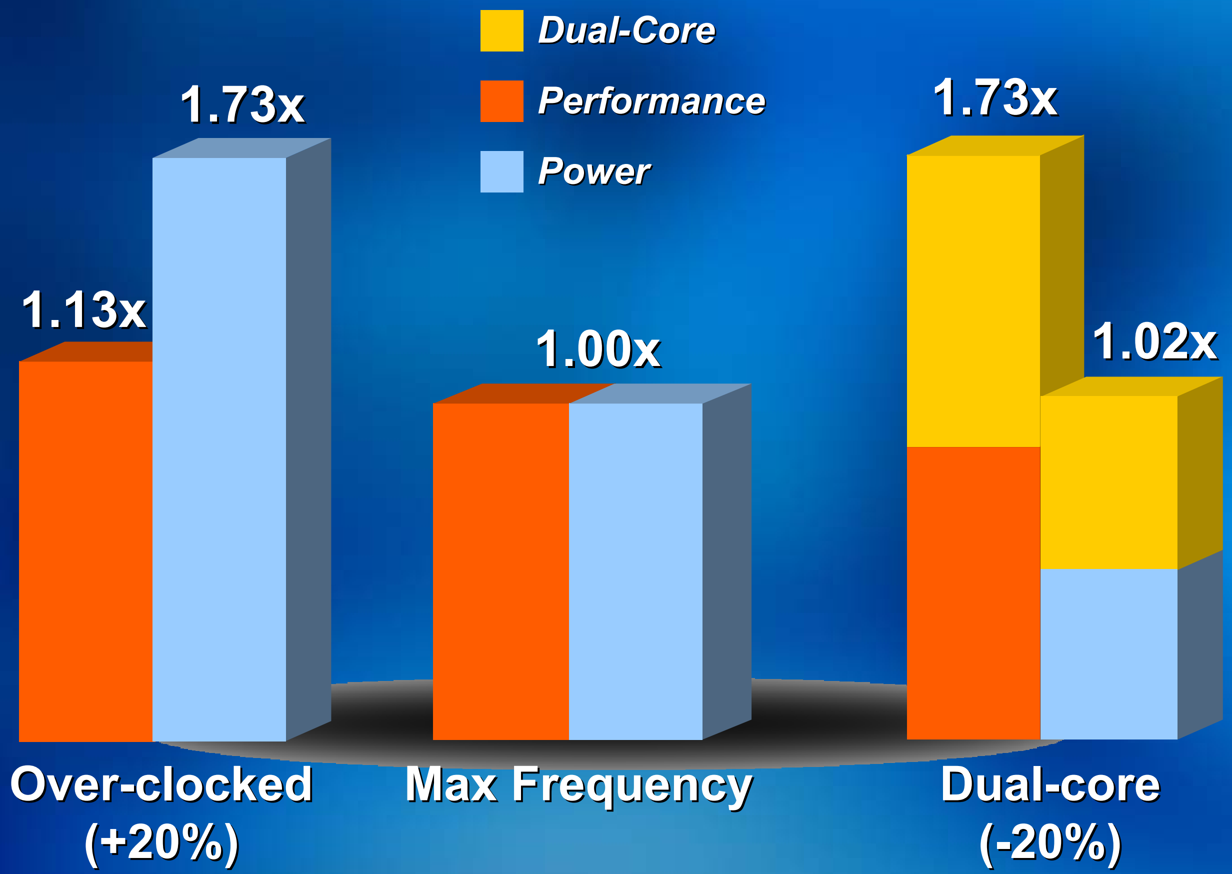
Over-clocking



Under-clocking



Multi-Core
Energy-Efficient Performance



Moore's Law will provide transistors

Intel process technology capabilities

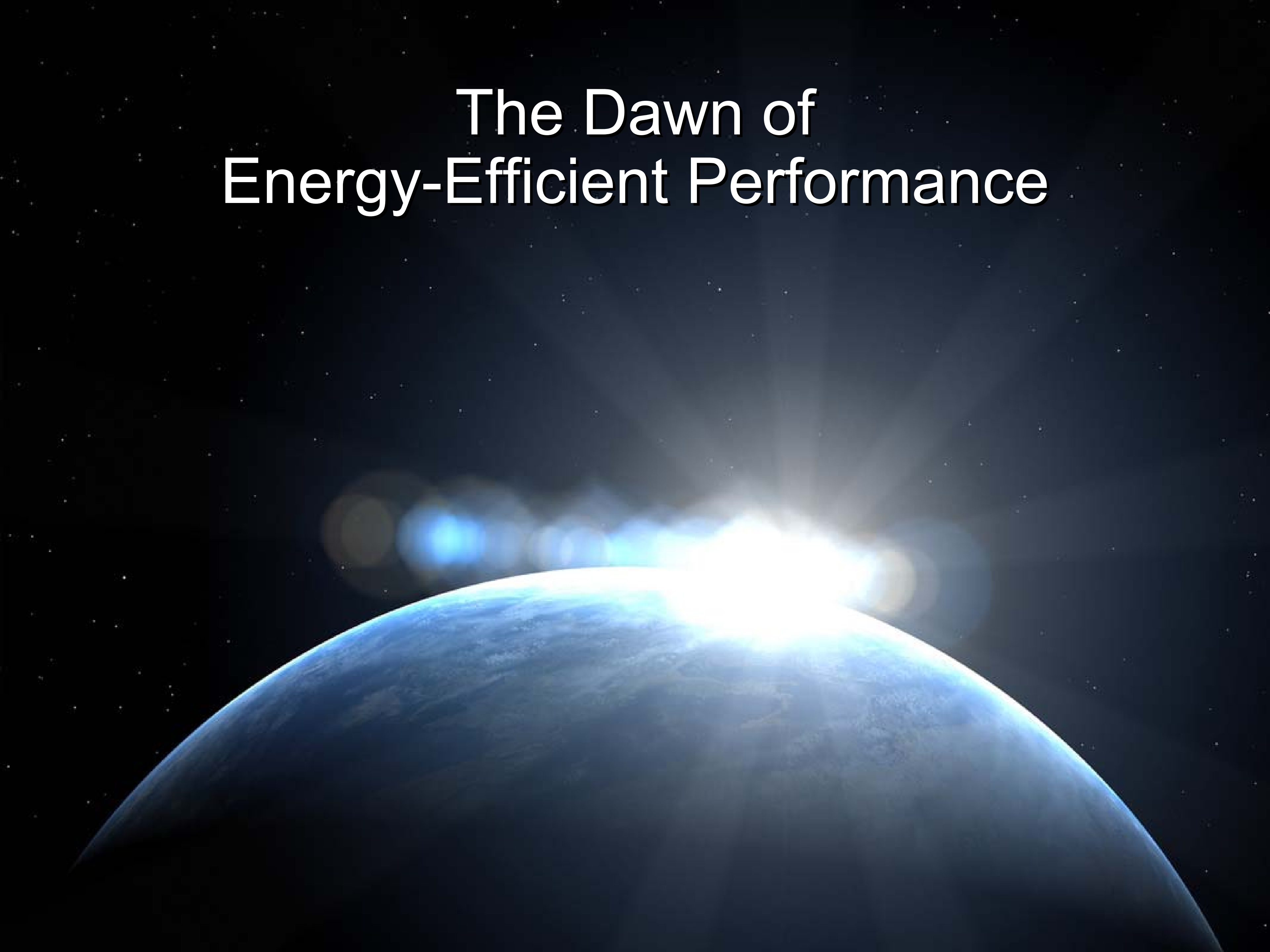
High Volume Manufacturing	2004	2006	2008	2010	2012	2014	2016	2018
Feature Size	90nm	65nm	45nm	32nm	22nm	16nm	11nm	8nm
Integration Capacity (Billions of Transistors)	2	4	8	16	32	64	128	256

Use transistors for

- Multiple cores
- On-core memory (caches)
- New features (*Ts)

Multiple cores and caches address power and memory latency issues

The Dawn of Energy-Efficient Performance



Agenda

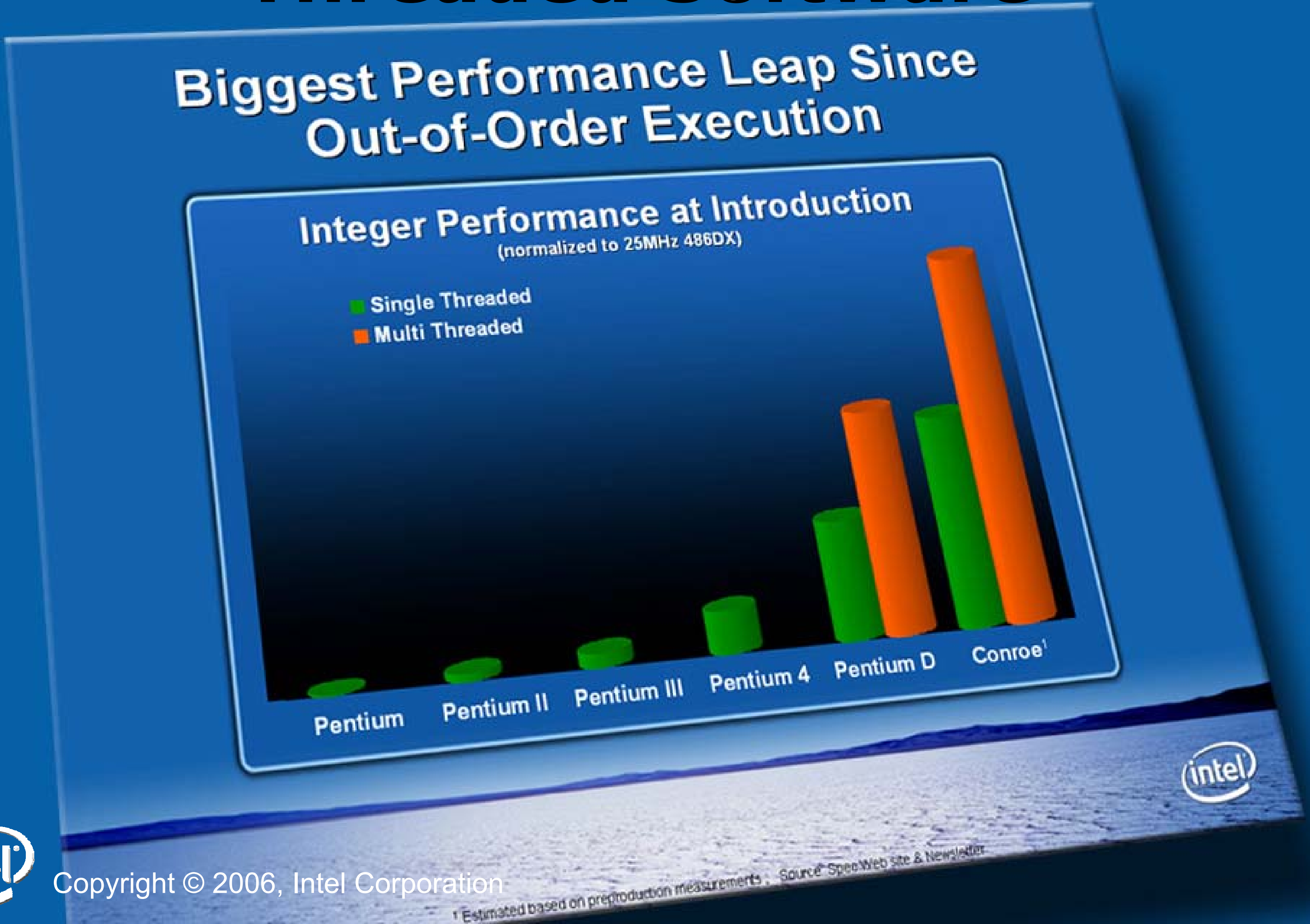
Software at Intel

Major Technological Change

Software Response

Parallel Programming 2.0

Multi-Core Platforms Demand Threaded Software



Biggest Performance Leap Since Out-of-Order Execution

Integer Performance at Introduction

(normalized to 25MHz 486DX)

[1] Estimated based on preproduction measurements , Source: Spec Web Site & Newsletter

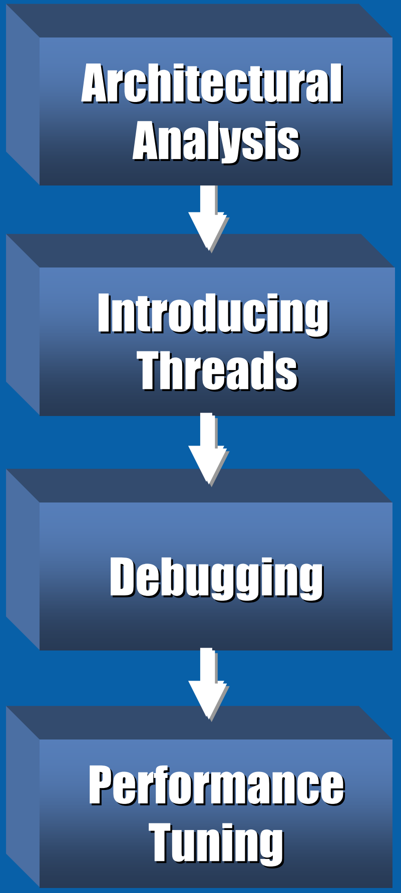
The Importance of Threading

- Do Nothing: Benefits Still Visible
  - Operating systems ready for multi-processing
  - Background tasks benefit from more compute resources
- Parallelize: Unlock the Potential
  - Threaded applications
  - Threaded libraries
  - Compiler generated threads

Unleash Multi-Core Potential

- Intel® C++ and Fortran Compilers
  - Built-in threading support with Auto-Parallelization and OpenMP* support
- Intel® Thread Checker & Thread Profiler
  - Unique product locates hard to find threading errors before they happen!
  - Helps developers optimize threaded applications
- Intel® MKL and IPP Performance Libraries
  - Highly optimized threaded libraries that enable multi-core performance gains even if your application isn’t threaded!
- Intel® VTune™ Performance Analyzer
  - Identifies performance bottlenecks in single or multithreaded applications to maximize performance

Example: Threading for Multi-Core



Architectural Analysis

Introducing Threads

Debugging

Performance Tuning

Threading for Multi-Core

Architectural Analysis

Introducing Threads

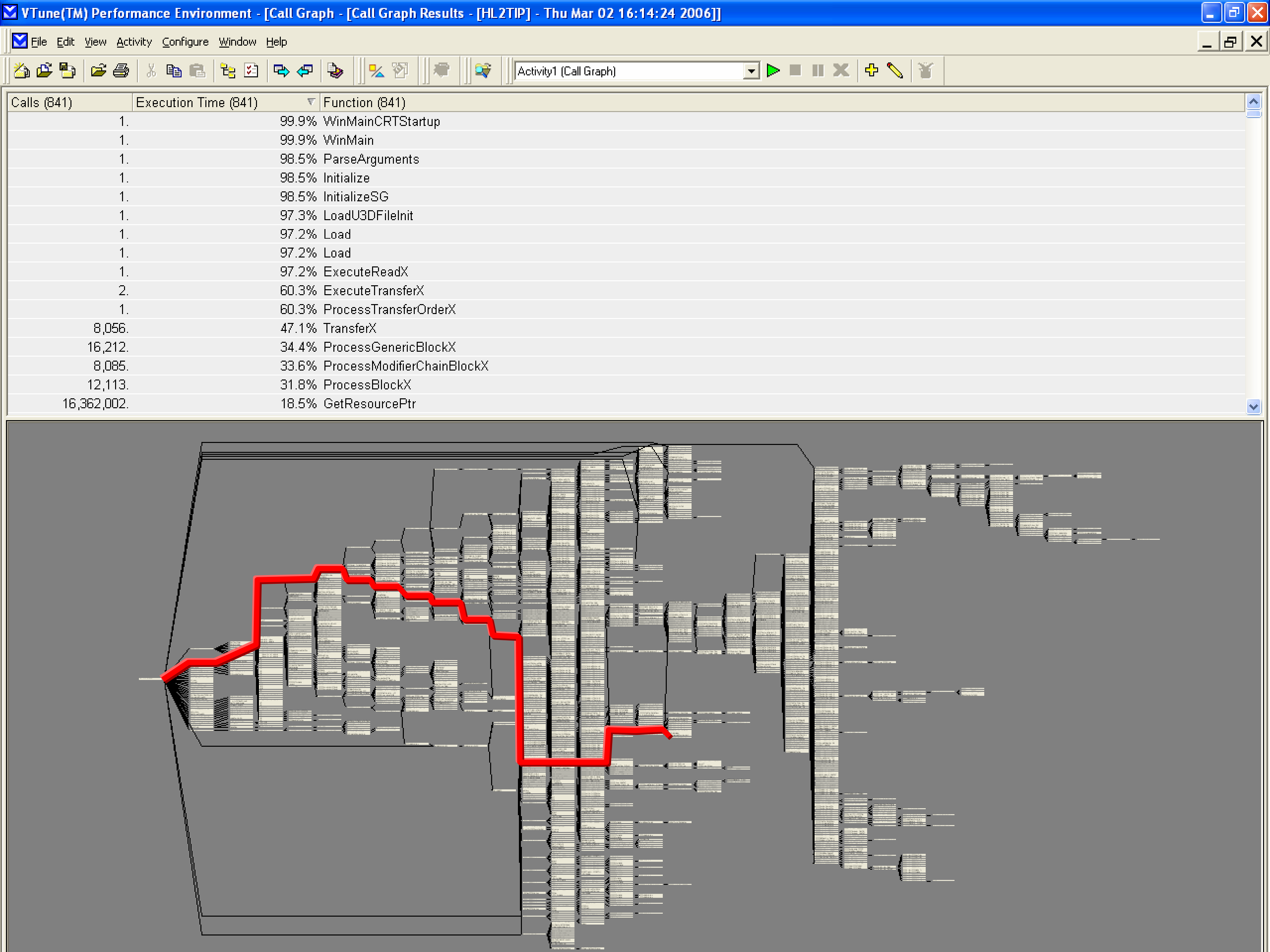
Debugging

Performance Tuning

Intel® VTune™ Performance Analyzer

Call Graph

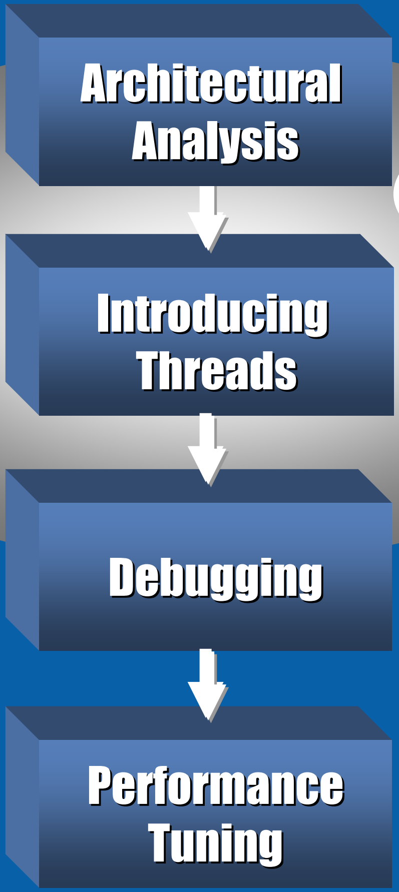
- Functional Structure
- Execution Times
- Counts



VTune(TM) Performance Environment - [Call Graph - [Call Graph Results - [HL2TIP] - Thu Mar 02 16:14:24 2006]]
File
Edit
View
Activity
Configure
Window
Help
Activity1 (Call Graph)

Calls (841)	Execution Time (841)	Function (841)
1.	99.9%	WinMainCRTStartup
1.	99.9%	WinMain
1.	98.5%	ParseArguments
1.	98.5%	Initialize
1.	98.5%	InitializeSG
1.	97.3%	LoadU3DFileInit
1.	97.2%	Load
1.	97.2%	Load
1.	97.2%	ExecuteReadX
2.	60.3%	ExecuteTransferX
1.	60.3%	ProcessTransferOrderX
8,056.	47.1%	TransferX
16,212.	34.4%	ProcessGenericBlockX
8,085.	33.6%	ProcessModifierChainBlockX
12,113.	31.8%	ProcessBlockX
16,362,002.	18.5%	GetResourcePtr

Threading for Multi-Core



Architectural Analysis

Introducing Threads

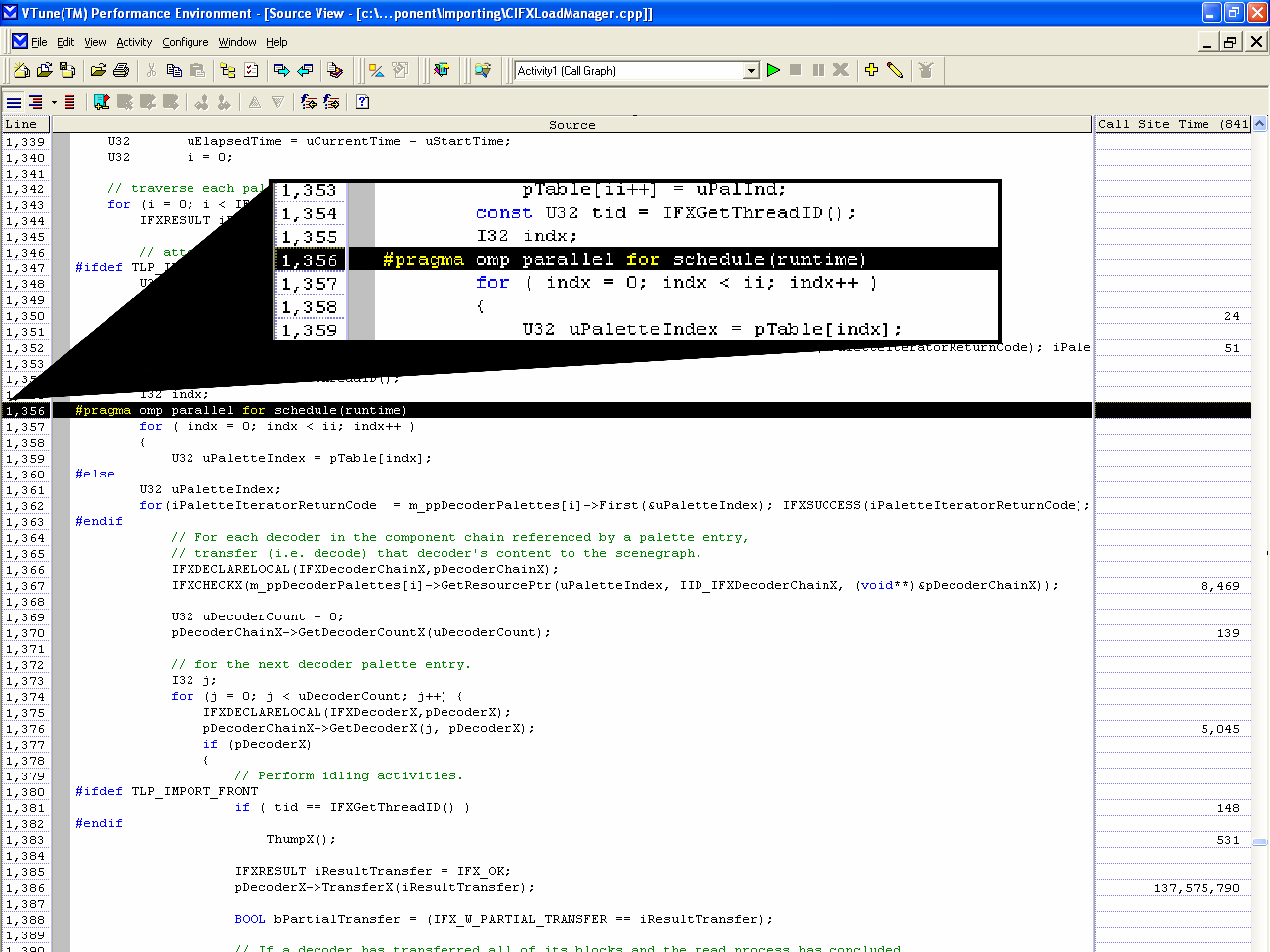
Debugging

Performance Tuning

Intel® Compilers

OpenMP Loop Construct

- Creates one thread per core
- Assigns iterations to threads

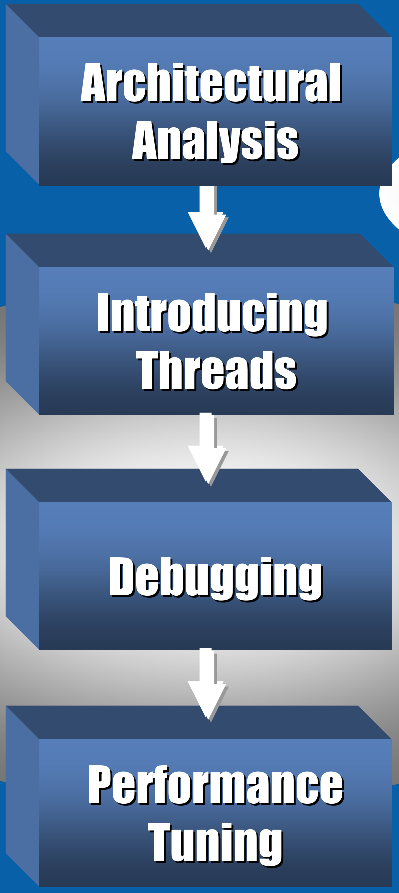

VTune(TM) Performance Environment - [Source View - [c:\...ponent\Importing\CIFXLoadManager.cpp]]
File
Edit
View
Activity
Configure
Window
Help
Activity1 (Call Graph)
Line
Source
Call Site Time (841

```
U32         uElapsedTime = uCurrentTime - uStartTime;
U32         i = 0;

// traverse each pal
for (i = 0; i < IF
    IFXRESULT i

    // att
#ifdef TLP_I
    U3




                                                                eIteratorReturnCode); iPale        51


                                      ThreadID();
    I32 indx;
#pragma omp parallel for schedule(runtime)
        for ( indx = 0; indx < ii; indx++ )
        {
            U32 uPaletteIndex = pTable[indx];
#else
        U32 uPaletteIndex;
        for(iPaletteIteratorReturnCode  = m_ppDecoderPalettes[i]->First(&uPaletteIndex); IFXSUCCESS(iPaletteIteratorReturnCode);
#endif
            // For each decoder in the component chain referenced by a palette entry,
            // transfer (i.e. decode) that decoder's content to the scenegraph.
            IFXDECLARELOCAL(IFXDecoderChainX,pDecoderChainX);
            IFXCHECKX(m_ppDecoderPalettes[i]->GetResourcePtr(uPaletteIndex, IID_IFXDecoderChainX, (void**)&pDecoderChainX));        8,469

            U32 uDecoderCount = 0;
            pDecoderChainX->GetDecoderCountX(uDecoderCount);        139

            // for the next decoder palette entry.
            I32 j;
            for (j = 0; j < uDecoderCount; j++) {
                IFXDECLARELOCAL(IFXDecoderX,pDecoderX);
                pDecoderChainX->GetDecoderX(j, pDecoderX);        5,045
                if (pDecoderX)
                {
                    // Perform idling activities.
#ifdef TLP_IMPORT_FRONT
                    if ( tid == IFXGetThreadID() )        148
#endif
                        ThumpX();        531

                    IFXRESULT iResultTransfer = IFX_OK;
                    pDecoderX->TransferX(iResultTransfer);        137,575,790

                    BOOL bPartialTransfer = (IFX_W_PARTIAL_TRANSFER == iResultTransfer);

                    // If a decoder has transferred all of its blocks and the read process has concluded
```

In the callout, lines 1,350 and 1,352 show Call Site Times of 24 and 51:

```
            pTable[ii++] = uPalInd;
        const U32 tid = IFXGetThreadID();
        I32 indx;
    #pragma omp parallel for schedule(runtime)
        for ( indx = 0; indx < ii; indx++ )
        {
            U32 uPaletteIndex = pTable[indx];
```

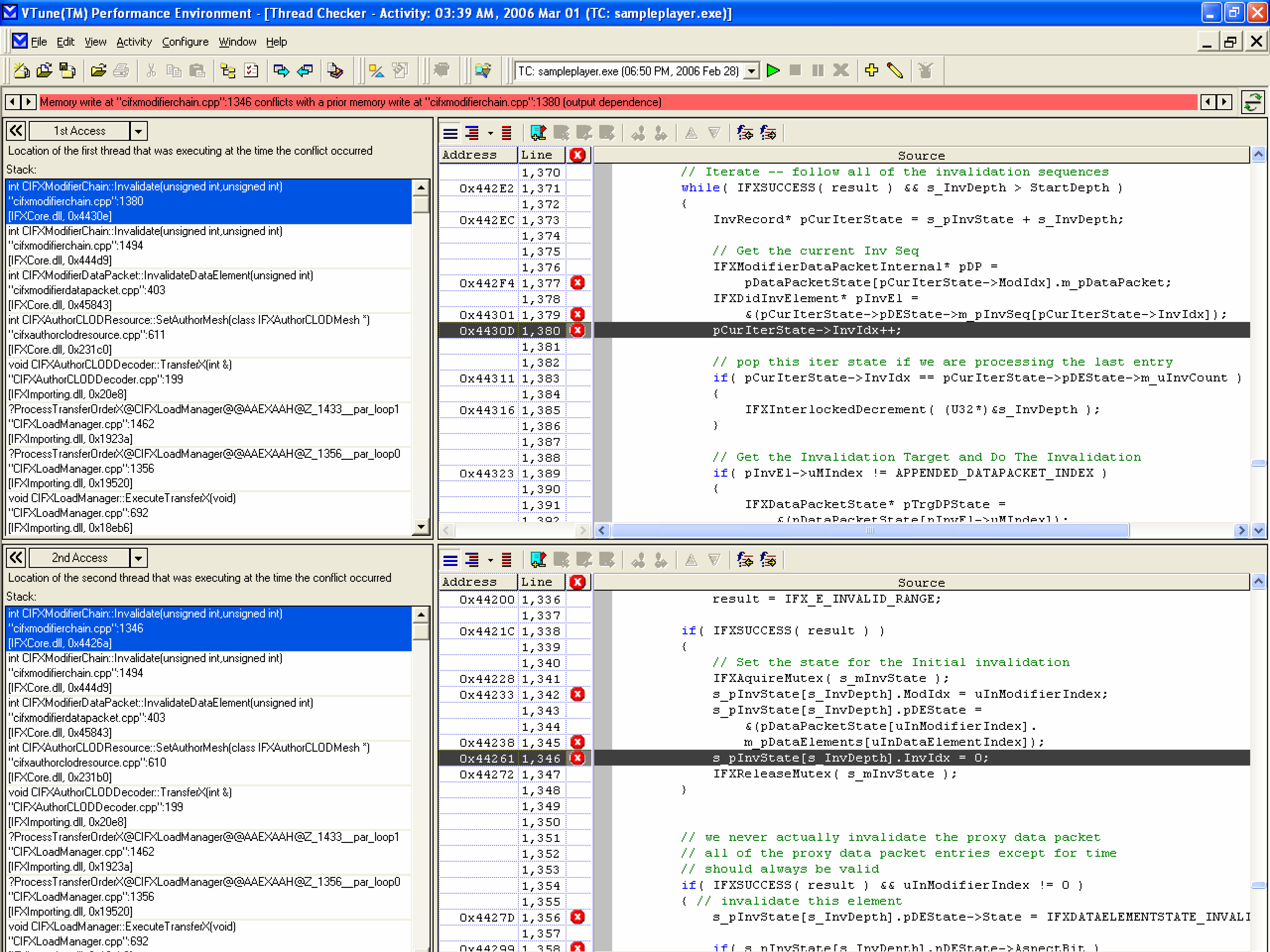
Threading for Multi-Core



Intel® Thread Checker

Thread Safety Issues

- Data Races
- Deadlocks



VTune(TM) Performance Environment - [Thread Checker - Activity: 03:39 AM, 2006 Mar 01 (TC: sampleplayer.exe)]

TC: sampleplayer.exe (06:50 PM, 2006 Feb 28)

Memory write at "cifxmodifierchain.cpp":1346 conflicts with a prior memory write at "cifxmodifierchain.cpp":1380 (output dependence)

1st Access

Location of the first thread that was executing at the time the conflict occurred

Stack:

```
int CIFXModifierChain::Invalidate(unsigned int,unsigned int)
"cifxmodifierchain.cpp":1380
[IFXCore.dll, 0x4430e]
int CIFXModifierChain::Invalidate(unsigned int,unsigned int)
"cifxmodifierchain.cpp":1494
[IFXCore.dll, 0x444d9]
int CIFXModifierDataPacket::InvalidateDataElement(unsigned int)
"cifxmodifierdatapacket.cpp":403
[IFXCore.dll, 0x45843]
int CIFXAuthorCLODResource::SetAuthorMesh(class IFXAuthorCLODMesh *)
"cifxauthorclodresource.cpp":611
[IFXCore.dll, 0x231c0]
void CIFXAuthorCLODDecoder::TransferX(int &)
"CIFXAuthorCLODDecoder.cpp":199
[IFXImporting.dll, 0x20e8]
?ProcessTransferOrderX@CIFXLoadManager@@AAEXAAH@Z_1433__par_loop1
"CIFXLoadManager.cpp":1462
[IFXImporting.dll, 0x1923a]
?ProcessTransferOrderX@CIFXLoadManager@@AAEXAAH@Z_1356__par_loop0
"CIFXLoadManager.cpp":1356
[IFXImporting.dll, 0x19520]
void CIFXLoadManager::ExecuteTransferX(void)
"CIFXLoadManager.cpp":692
[IFXImporting.dll, 0x18eb6]
```

```
Address   Line   Source
          1,370          // Iterate -- follow all of the invalidation sequences
0x442E2   1,371          while( IFXSUCCESS( result ) && s_InvDepth > StartDepth )
          1,372          {
0x442EC   1,373              InvRecord* pCurIterState = s_pInvState + s_InvDepth;
          1,374
          1,375              // Get the current Inv Seq
          1,376              IFXModifierDataPacketInternal* pDP =
0x442F4   1,377                  pDataPacketState[pCurIterState->ModIdx].m_pDataPacket;
          1,378              IFXDidInvElement* pInvEl =
0x44301   1,379                  &(pCurIterState->pDEState->m_pInvSeq[pCurIterState->InvIdx]);
0x4430D   1,380              pCurIterState->InvIdx++;
          1,381
          1,382              // pop this iter state if we are processing the last entry
0x44311   1,383              if( pCurIterState->InvIdx == pCurIterState->pDEState->m_uInvCount )
          1,384              {
0x44316   1,385                  IFXInterlockedDecrement( (U32*)&s_InvDepth );
          1,386              }
          1,387
          1,388              // Get the Invalidation Target and Do The Invalidation
0x44323   1,389              if( pInvEl->uMIndex != APPENDED_DATAPACKET_INDEX )
          1,390              {
          1,391                  IFXDataPacketState* pTrgDPState =
          1,392                      &(pDataPacketState[pInvEl->uMIndex]);
```

2nd Access

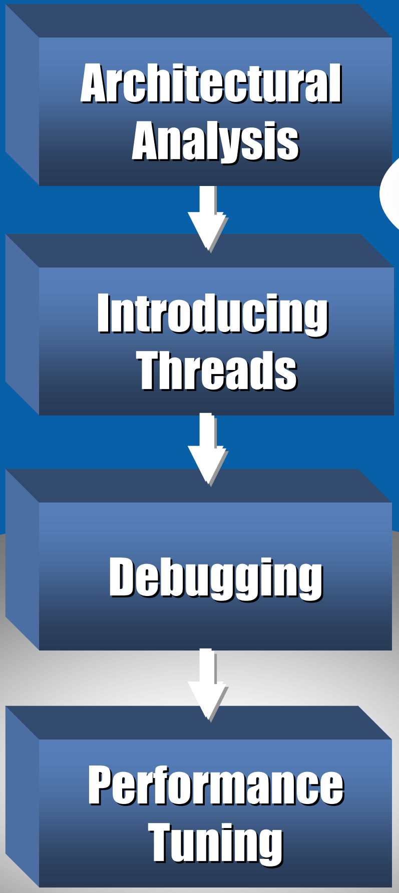
Location of the second thread that was executing at the time the conflict occurred

Stack:

```
int CIFXModifierChain::Invalidate(unsigned int,unsigned int)
"cifxmodifierchain.cpp":1346
[IFXCore.dll, 0x4426a]
int CIFXModifierChain::Invalidate(unsigned int,unsigned int)
"cifxmodifierchain.cpp":1494
[IFXCore.dll, 0x444d9]
int CIFXModifierDataPacket::InvalidateDataElement(unsigned int)
"cifxmodifierdatapacket.cpp":403
[IFXCore.dll, 0x45843]
int CIFXAuthorCLODResource::SetAuthorMesh(class IFXAuthorCLODMesh *)
"cifxauthorclodresource.cpp":610
[IFXCore.dll, 0x231b0]
void CIFXAuthorCLODDecoder::TransferX(int &)
"CIFXAuthorCLODDecoder.cpp":199
[IFXImporting.dll, 0x20e8]
?ProcessTransferOrderX@CIFXLoadManager@@AAEXAAH@Z_1433__par_loop1
"CIFXLoadManager.cpp":1462
[IFXImporting.dll, 0x1923a]
?ProcessTransferOrderX@CIFXLoadManager@@AAEXAAH@Z_1356__par_loop0
"CIFXLoadManager.cpp":1356
[IFXImporting.dll, 0x19520]
void CIFXLoadManager::ExecuteTransferX(void)
"CIFXLoadManager.cpp":692
```

```
Address   Line   Source
0x44200   1,336              result = IFX_E_INVALID_RANGE;
          1,337
0x4421C   1,338          if( IFXSUCCESS( result ) )
          1,339          {
          1,340              // Set the state for the Initial invalidation
0x44228   1,341              IFXAquireMutex( s_mInvState );
0x44233   1,342              s_pInvState[s_InvDepth].ModIdx = uInModifierIndex;
          1,343              s_pInvState[s_InvDepth].pDEState =
          1,344                  &(pDataPacketState[uInModifierIndex].
0x44238   1,345                  m_pDataElements[uInDataElementIndex]);
0x44261   1,346              s_pInvState[s_InvDepth].InvIdx = 0;
0x44272   1,347              IFXReleaseMutex( s_mInvState );
          1,348          }
          1,349
          1,350
          1,351          // we never actually invalidate the proxy data packet
          1,352          // all of the proxy data packet entries except for time
          1,353          // should always be valid
          1,354          if( IFXSUCCESS( result ) && uInModifierIndex != 0 )
          1,355          { // invalidate this element
0x4427D   1,356              s_pInvState[s_InvDepth].pDEState->State = IFXDATAELEMENTSTATE_INVALI
          1,357
0x44299   1,358              if( s_pInvState[s_InvDepth].pDEState->AspectBit )
```

Threading for Multi-Core



Architectural Analysis

Introducing Threads

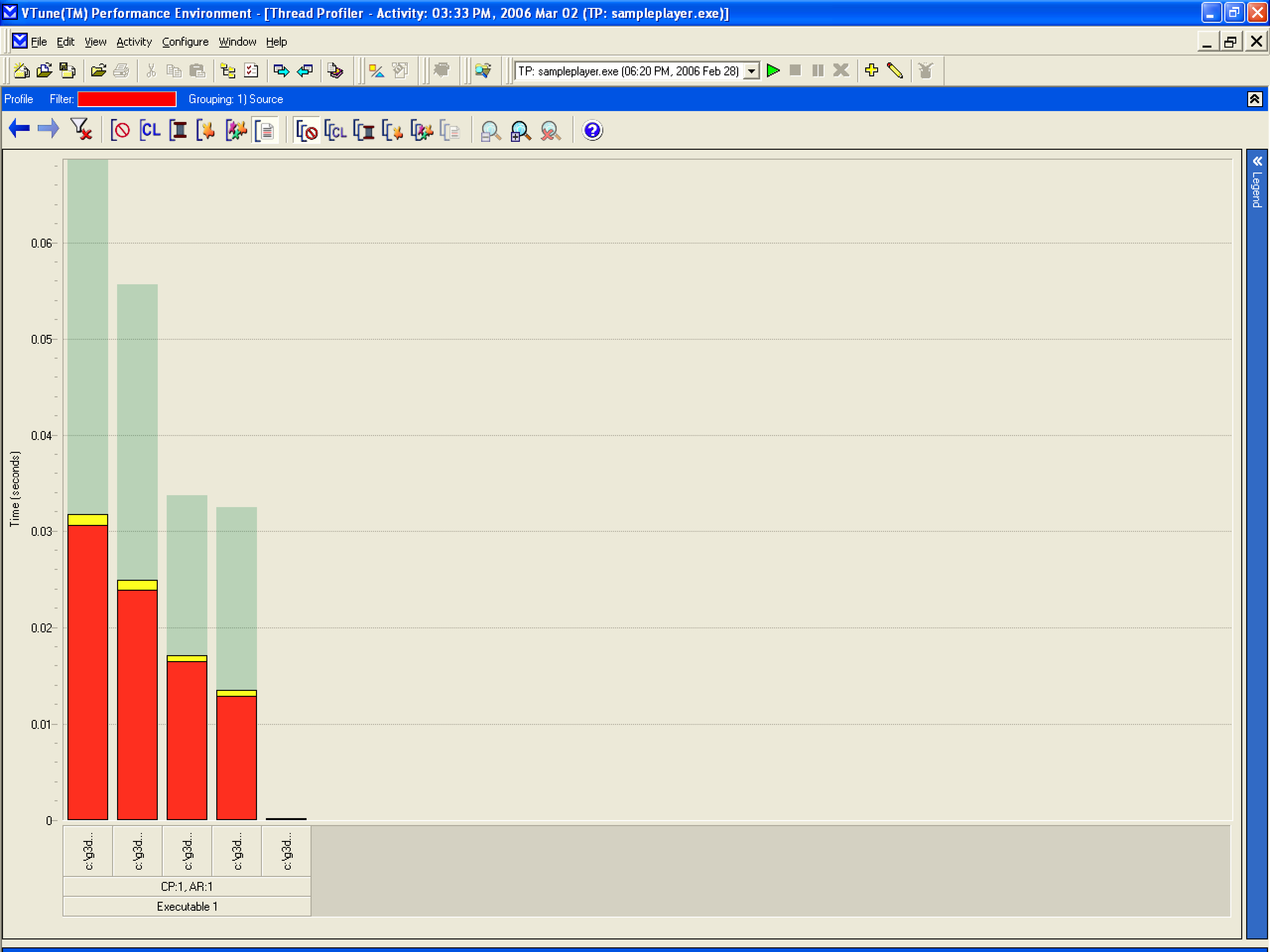
Debugging

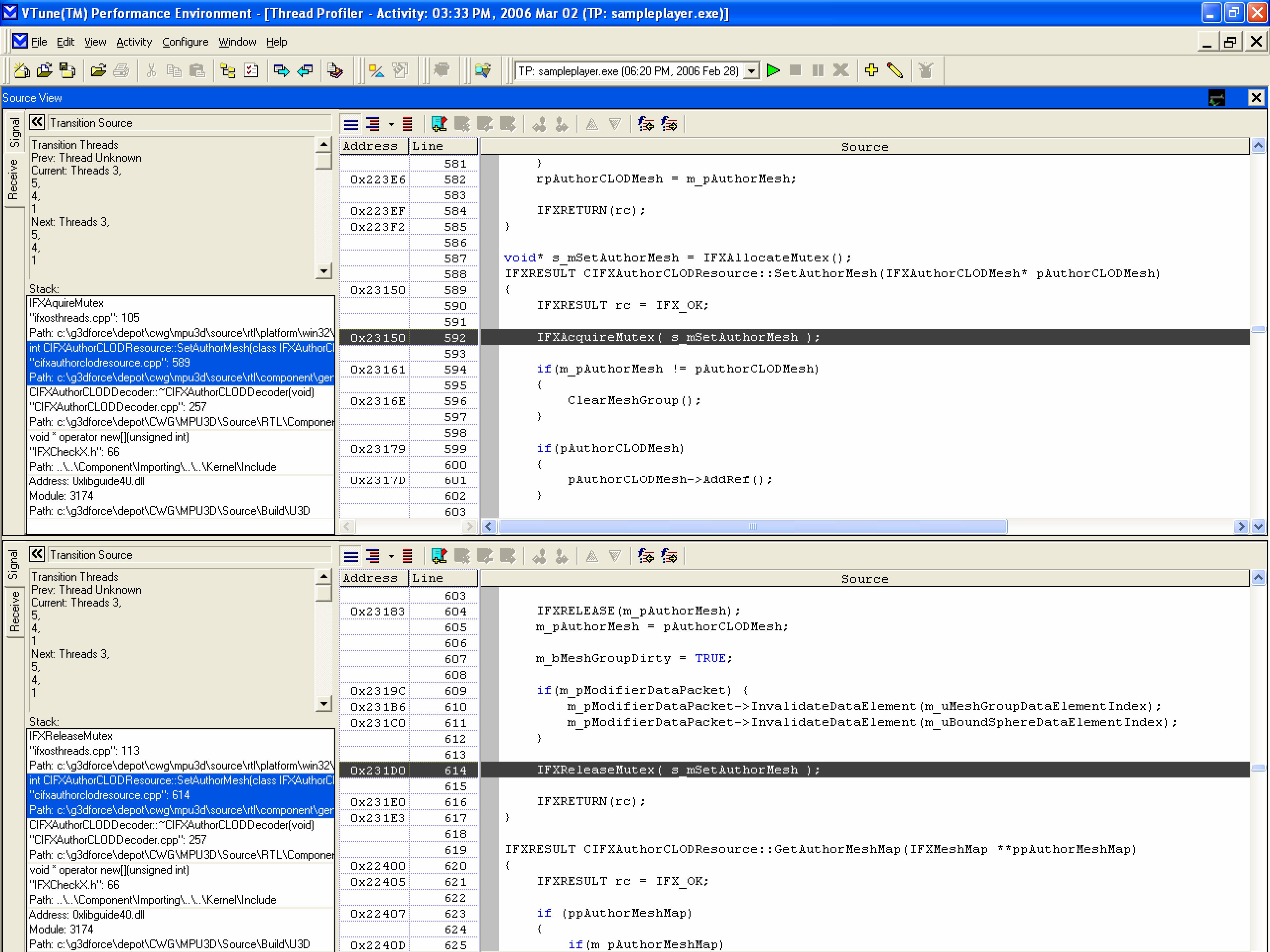
Performance Tuning

Intel® Thread Profiler

Find Contended Locks

- Most Overhead
- Largest Reduction in Parallelism


VTune(TM) Performance Environment - [Thread Profiler - Activity: 03:33 PM, 2006 Mar 02 (TP: sampleplayer.exe)]


VTune(TM) Performance Environment - [Thread Profiler - Activity: 03:33 PM, 2006 Mar 02 (TP: sampleplayer.exe)]
File Edit View Activity Configure Window Help
TP: sampleplayer.exe (06:20 PM, 2006 Feb 28)
Source View
Signal
Receive
Transition Source
Address
Line
Source

Transition Threads
Prev: Thread Unknown
Current: Threads 3,
5,
4,
1
Next: Threads 3,
5,
4,
1

Stack:
IFXAquireMutex
"ifxosthreads.cpp": 105
Path: c:\g3dforce\depot\cwg\mpu3d\source\rtl\platform\win32\
int CIFXAuthorCLODResource::SetAuthorMesh(class IFXAuthorCl
"cifxauthorclodresource.cpp": 589
Path: c:\g3dforce\depot\cwg\mpu3d\source\rtl\component\ger
CIFXAuthorCLODDecoder::~CIFXAuthorCLODDecoder(void)
"CIFXAuthorCLODDecoder.cpp": 257
Path: c:\g3dforce\depot\CWG\MPU3D\Source\RTL\Componer
void * operator new[](unsigned int)
"IFXCheckX.h": 66
Path: ..\..\Component\Importing\..\..\Kernel\Include
Address: 0xlibguide40.dll
Module: 3174
Path: c:\g3dforce\depot\CWG\MPU3D\Source\Build\U3D

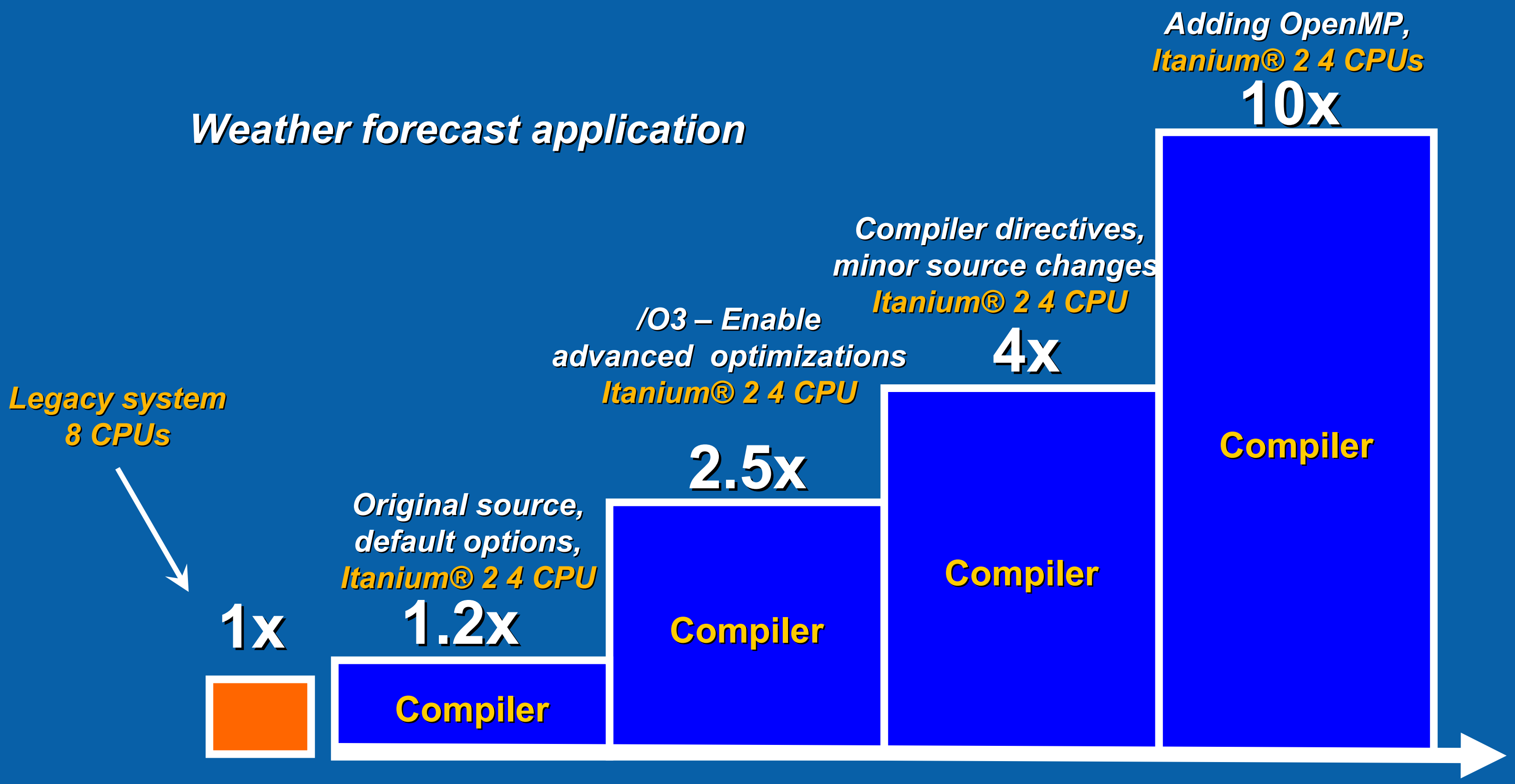
```
Address   Line   Source
           581       }
0x223E6    582       rpAuthorCLODMesh = m_pAuthorMesh;
           583
0x223EF    584       IFXRETURN(rc);
0x223F2    585   }
           586
           587   void* s_mSetAuthorMesh = IFXAllocateMutex();
           588   IFXRESULT CIFXAuthorCLODResource::SetAuthorMesh(IFXAuthorCLODMesh* pAuthorCLODMesh)
0x23150    589   {
           590       IFXRESULT rc = IFX_OK;
           591
0x23150    592       IFXAcquireMutex( s_mSetAuthorMesh );
           593
0x23161    594       if(m_pAuthorMesh != pAuthorCLODMesh)
           595       {
0x2316E    596           ClearMeshGroup();
           597       }
           598
0x23179    599       if(pAuthorCLODMesh)
           600       {
0x2317D    601           pAuthorCLODMesh->AddRef();
           602       }
           603
```

Transition Threads
Prev: Thread Unknown
Current: Threads 3,
5,
4,
1
Next: Threads 3,
5,
4,
1

Stack:
IFXReleaseMutex
"ifxosthreads.cpp": 113
Path: c:\g3dforce\depot\cwg\mpu3d\source\rtl\platform\win32\
int CIFXAuthorCLODResource::SetAuthorMesh(class IFXAuthorCl
"cifxauthorclodresource.cpp": 614
Path: c:\g3dforce\depot\cwg\mpu3d\source\rtl\component\ger
CIFXAuthorCLODDecoder::~CIFXAuthorCLODDecoder(void)
"CIFXAuthorCLODDecoder.cpp": 257
Path: c:\g3dforce\depot\CWG\MPU3D\Source\RTL\Componer
void * operator new[](unsigned int)
"IFXCheckX.h": 66
Path: ..\..\Component\Importing\..\..\Kernel\Include
Address: 0xlibguide40.dll
Module: 3174
Path: c:\g3dforce\depot\CWG\MPU3D\Source\Build\U3D

```
Address   Line   Source
           603
0x23183    604       IFXRELEASE(m_pAuthorMesh);
           605       m_pAuthorMesh = pAuthorCLODMesh;
           606
           607       m_bMeshGroupDirty = TRUE;
           608
0x2319C    609       if(m_pModifierDataPacket) {
0x231B6    610           m_pModifierDataPacket->InvalidateDataElement(m_uMeshGroupDataElementIndex);
0x231C0    611           m_pModifierDataPacket->InvalidateDataElement(m_uBoundSphereDataElementIndex);
           612       }
           613
0x231D0    614       IFXReleaseMutex( s_mSetAuthorMesh );
           615
0x231E0    616       IFXRETURN(rc);
0x231E3    617   }
           618
           619   IFXRESULT CIFXAuthorCLODResource::GetAuthorMeshMap(IFXMeshMap **ppAuthorMeshMap)
0x22400    620   {
0x22405    621       IFXRESULT rc = IFX_OK;
           622
0x22407    623       if (ppAuthorMeshMap)
           624       {
0x2240D    625           if(m_pAuthorMeshMap)
```

Performance Impact



Optimizations may improve performance significantly.

Agenda

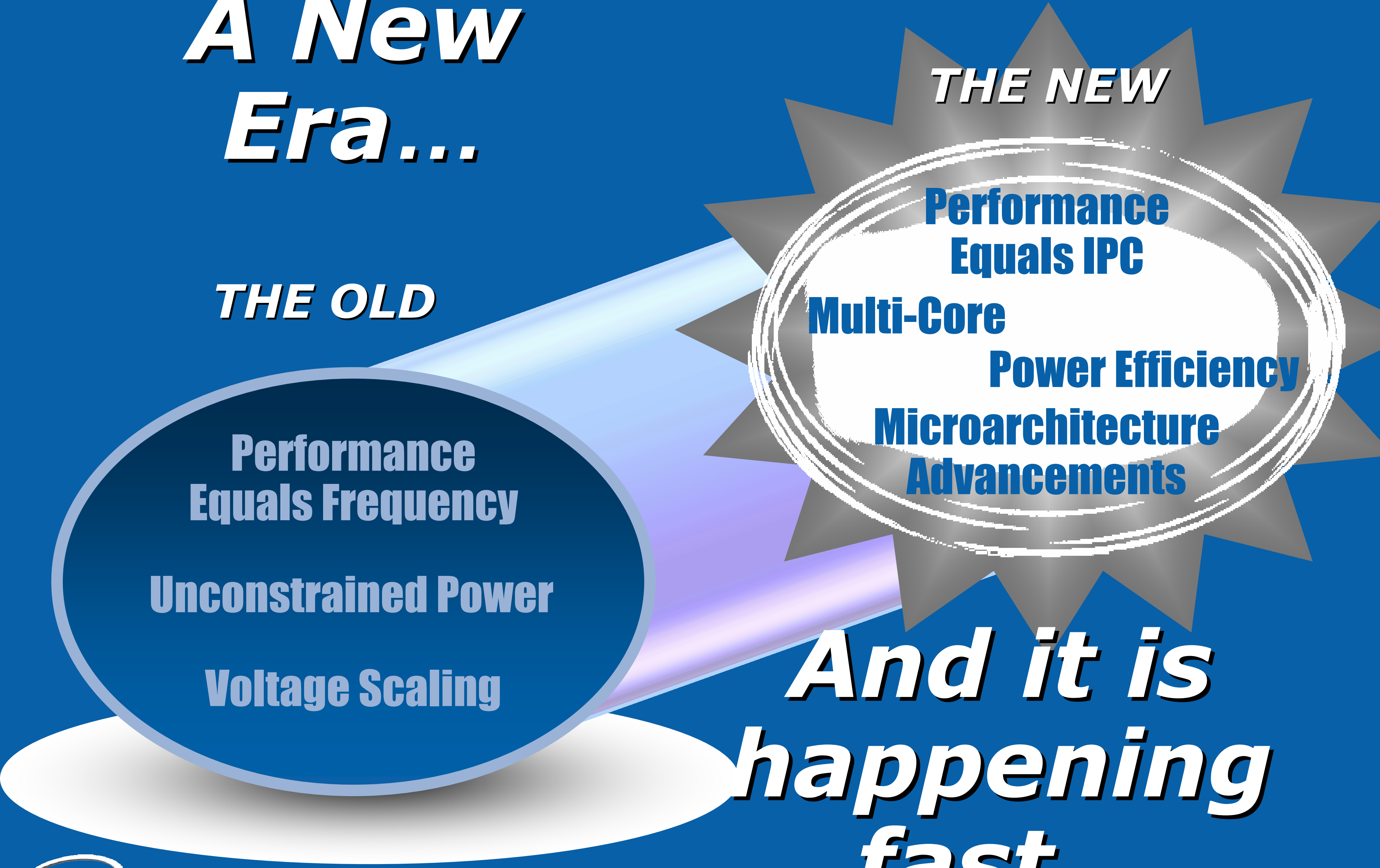
Software at Intel

Major Technological Change

Software Response

Parallel Programming 2.0

A New Era…



THE OLD

Performance Equals Frequency

Unconstrained Power

Voltage Scaling

THE NEW

Performance Equals IPC

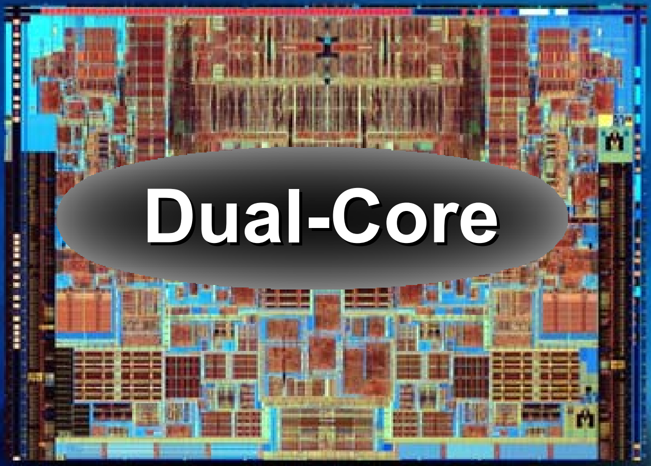
Multi-Core

Power Efficiency

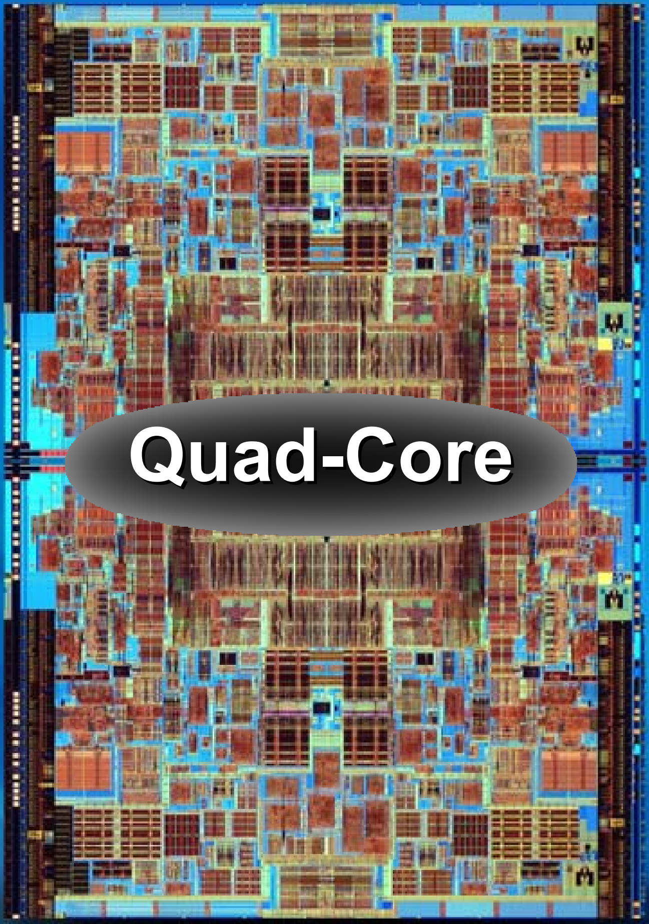
Microarchitecture Advancements

And it is happening fast…

Multi-Core Trajectory

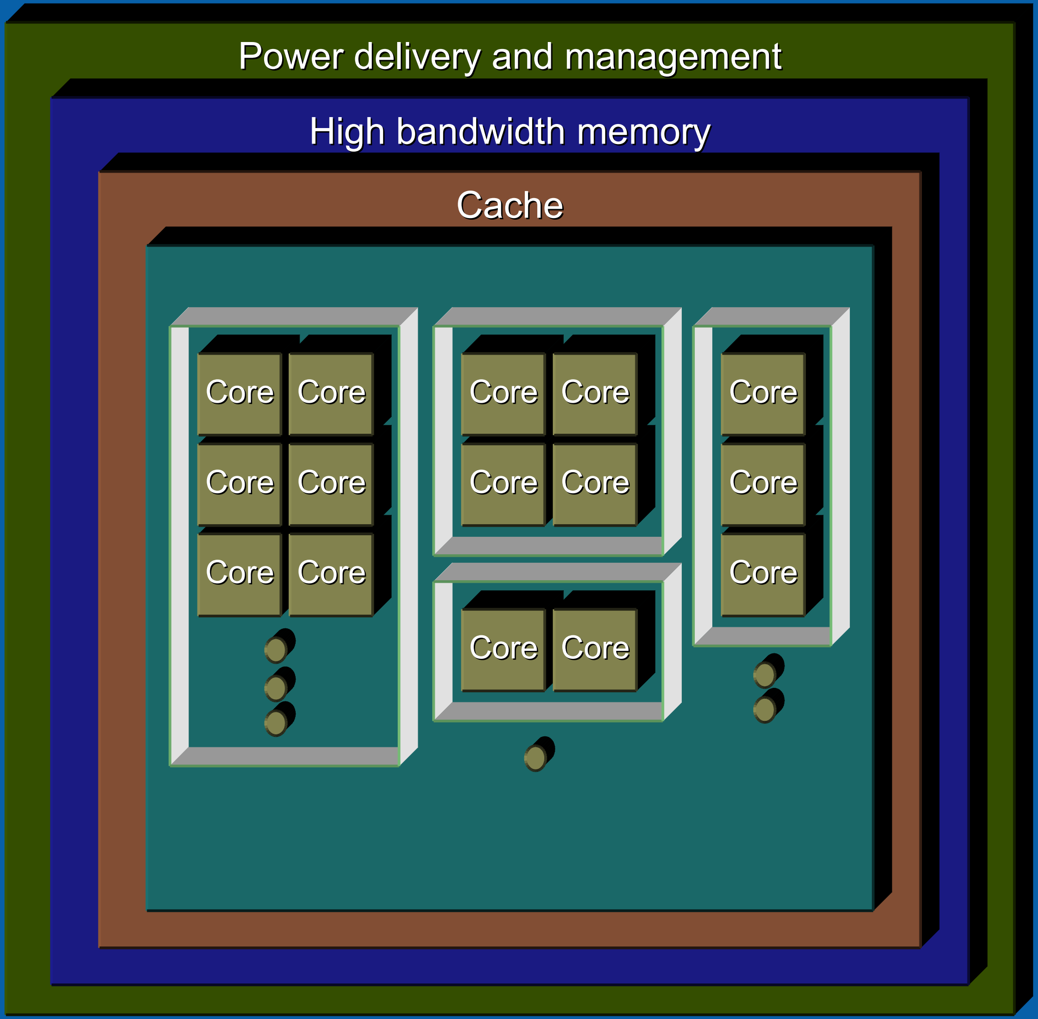


2005/2006



2007

Growing Momentum for Multi-Cores

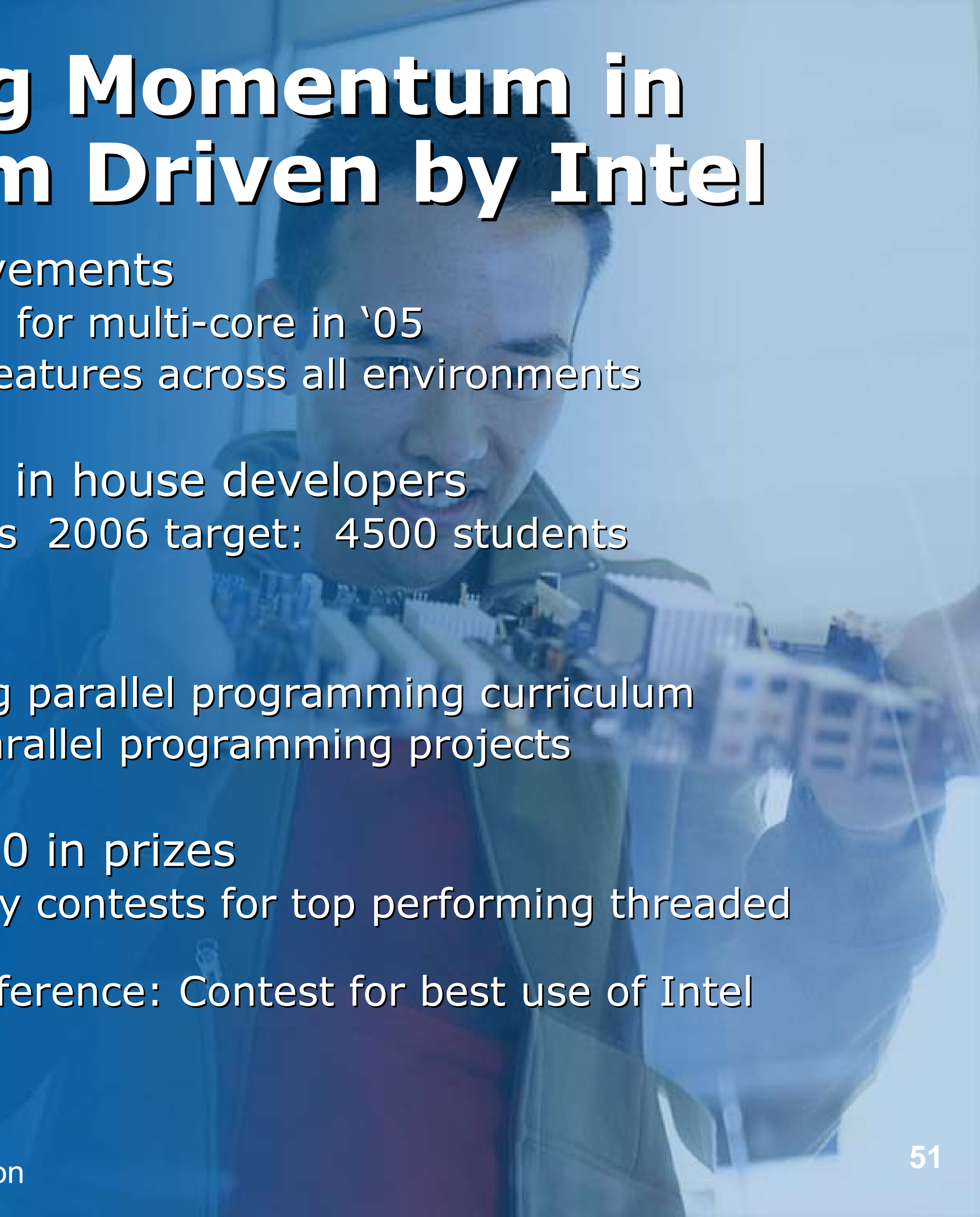


Growing Momentum For Software Parallelization

Activision (Ravensoft)
Adobe
Algorithmics
Alias
Autodesk
Business Objects
Cakewalk
CodecPeople
Computer Associates
Corel (WordPerfect)
Cyberlink
Discreet
IBM
id Software
Landmark
Macromedia
Mainconcept
Maxon
mental images
Microsoft (Office Suite)
Midway
MSC
Novell SUSE
Oracle
Pegasus

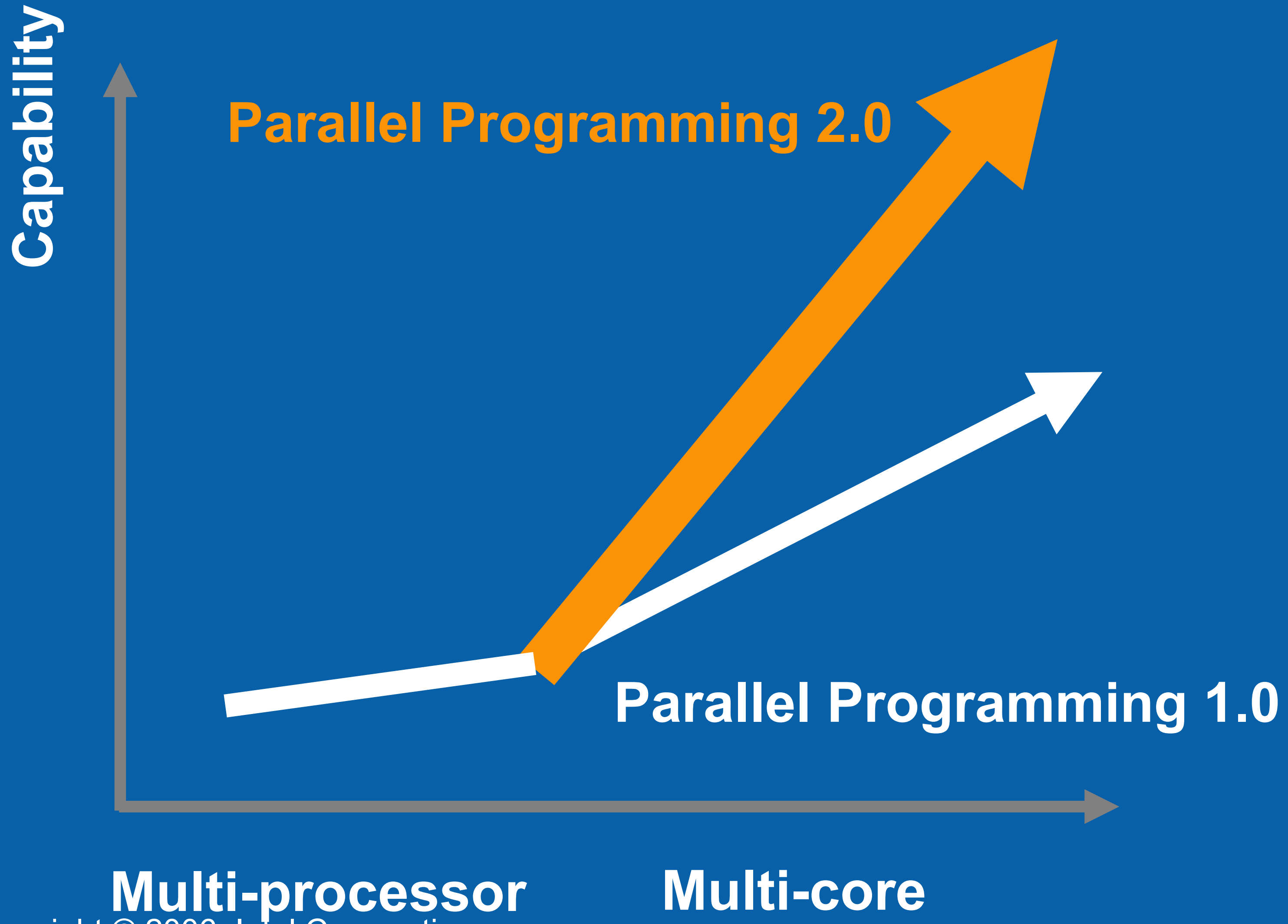
Pinnacle
Pixar (Renderman)
Paradigm
PTC
Red Hat
SAP
SAS
Siebel CRM
Signet
Skype
SLB
SnapStream
Sonic (Roxio)
Sony
Steinberg
SunGard
Sybase
Symantec
Thomson
THQ
Ubisoft
UGS
Valve
Yahoo (Musicmatch)

Growing Momentum in Ecosystem Driven by Intel



- Continuous tool improvements
  - Over 15 tools released for multi-core in ‘05
  - Coming in ‘06: more features across all environments
- Training end users and in house developers
  - 2005: > 1500 students 2006 target: 4500 students
- Academia
  - Universities developing parallel programming curriculum
  - Research grants for parallel programming projects
- Contest: Over $100,000 in prizes
  - Topcoder.com: Monthly contests for top performing threaded software
  - Games Developer Conference: Contest for best use of Intel platform features

Opportunity for Software:
Extract Full Potential of Multi-core



Parallel Programming 2.0

- Ubiquitous parallel software
  - Large spectrum of domains (consumer/wireless vs HPC/database, home vs nuclear labs)
- Scalable software
  - Explosion of cores (e.g. 2X cores every 18-24 months)
- User experience
  - vs. just raw performance
- Greater ease of programming
  - New programming paradigm, programming language, compiler, tools,
- Greater reliability and security
  - Programming for application and system reliability
- Industry-wide vs government-funded
- Higher demand for parallel programming education
  - Mass vs elite

Parallel Programming 2.0

The Beginning of a New Era