

AMD Developer Central

Tools & Resources for Software Developers



What AMD Developer Central Offers



Software Tools & Libraries

CPU

AMD CodeAnalyst™

A powerful suite of tools that analyzes software to identify opportunities for optimization and thread analysis

Framewave

A broad array of routines and functions designed to save you development time and effort, as well as to provide you with an easy path to faster, multi-threaded applications

AMD SimNow™ Simulator

A platform emulator for AMD's processor family, providing fast simulation of an entire computer system plus standard debugging features

GPU

GPU PerfStudio

A real-time performance analysis tool designed to help increase the graphics performance of D3D and OpenGL applications

GPU ShaderAnalyzer

A tool for analyzing the performance of pixel and vertex shaders on ATI graphics cards

AMD Tootle

A mesh optimization library that improves on existing mesh preprocessing techniques

RenderMonkey

A rich shader development environment for both programmers and artists which facilitates the collaborative creation of real-time shader effects

...and many more downloads



Best Practices & Technical Tips

Performance Optimization of Windows Applications on AMD Processors, Part I

AMD Developer Central > Documentation & Articles > Articles & Whitepapers > Performance Optimization of Windows Applications on AMD Processors, Part I

Part I: 64-bit optimization, multi-core, the vectorization, software profiling using AMD assembly code

Introduction

Michael Wall, Principal Member of Technical Staff, AMD Micro Devices Inc.

» Performance Optimization of Windows Applications, Part II

The Mandel project is a Visual Studio C/C++ project that illustrates several key concepts in Windows programming progressively through different performance optimization levels, it shows how to create and compile a simple Visual Studio 2005 or 2008. This floating-point is compiled using the 32-bit compiler, or built for the x64 compiler included in Visual Studio.

Dramatic performance gains are attained by using optimization tricks, then by using a series of full compiler SSE vector intrinsic functions. True not optionally integrated into the project, and assert. Finally, the newly supported OpenMP multi-thread multi-thread the application and double the performance desktop PCs or two-processor workstation machines.

What you need to get started

- PC based on AMD Athlon™ 64 or AMD Opteron™ core, multi-core or 3P preferred
- Microsoft Windows XP Professional x64 Edition or Visual Studio 2005 beta 2 or later, or Visual Studio 2008
- Standard and Express versions do not or OpenMP threading
- Microsoft DirectX SDK, June 2005 or later
- Mandel project files

Performance Optimization of Windows Applications on AMD Processors, Part II

AMD Developer Central > Documentation & Articles > Articles & Whitepapers > Performance Optimization of Windows Applications on AMD Processors, Part II

Part II: Cache and Memory

Michael Wall, Principal Member of Technical Staff, Advanced Micro Devices, Inc.

6/16/2008

- » Introduction
- » Memory optimization
- » Conclusion

Introduction

Multi-core microprocessors offer more raw CPU horsepower than ever before, but software developers need to face several challenges to really deliver on the promise of higher performance. One of those challenges involves the increased demands on the cache and memory system.

Memory latency and bandwidth historically have been a limiting factor for many applications, even with single-core processors. But when two, four or even more processors (cores) reside in the same CPU package and compete for cache and memory resources, the problem becomes even more acute. This paper describes techniques developers can apply to make more efficient use of those resources.

Memory system basic description

Back in the early days of computers, memory systems were simple. The processor issued a memory request, waited while the data was read or written, then resumed execution. This kind of sequential processing is easy to understand, but is no longer accurate: the never-ending quest for higher performance has produced modern systems with far more complex behavior. Instructions are executed out-of-order, and many memory operations can be "in flight" simultaneously. In other words, the hardware attempts to hide memory latency by implementing greater concurrency in the memory system.

Cache system basic description

To reduce the effective latency of memory even further, processors include a relatively small amount of fast on-chip memory called the cache. Details vary, but virtually every modern processor includes some kind of cache. Typically, memory data which has recently been read or written resides in the cache. The assumption is that recently used data is likely to be needed again soon, and the cache provides faster access to that data.

The smallest unit of data that moves between the cache and the main memory is called a **cache line**. The cache line is a

» Performance Optimization of Windows Applications on AMD Processors, Part I

» Download cacheandmemory.zip

Coding Tips using Microsoft Visual Studio 2008 Targeting Quad-Core AMD Opteron™ Processors

With new "Barcelona" (2ND Gen), 3RD processors, including 3rd Generation AMD Opteron processors which feature a million quad-core architectures and L3 cache, software developers will enjoy improved instruction execution, optimized register allocation, and enhanced 32-bit floating-point performance when using Microsoft Visual Studio 2008.

Visual C++ Optimization Flags

- /O2: Optimizes for maximum speed
- /O1: Optimizes for size rather than /O2 (optimize for speed) may result in better performance when executing cache sensitive code. Make sure to test both flags depending on your scenario
- /GL: Whole-program optimization, especially interprocedural optimizations
- /fp:fast: Fast floating point (only consider if your application does not require full floating-point precision)

Tips for Barcelona Processors

- /arch:amd64: Compiles for 64-bit architecture
- /x64: Compiles for 64-bit architecture
- /x86: Compiles for 32-bit architecture
- /x86_64: Compiles for 64-bit architecture
- /x64-nostric: Compiles for 64-bit architecture without strict aliasing
- /x64-nostric-nostric: Compiles for 64-bit architecture without strict aliasing and without strict aliasing
- /x64-nostric-nostric-nostric: Compiles for 64-bit architecture without strict aliasing, without strict aliasing, and without strict aliasing

The cache-block optimized algorithm

```

for each cache-size block in the set
do operation A on the block // must read from memory
do operation B on the block // it's still in cache! fast!
do operation C on the block // it's still in cache! fast!
next block
    
```

...and more

Visual C++ Optimization Flags

/O2

→ Optimize for maximum speed

/O1

→ (Optimize for size) rather than /O2 (optimize for speed) may result in better performance when executing cache sensitive code. Make sure to test both flags depending on your scenario

/GL

→ Whole-program optimization, especially interprocedural optimizations

/fp:fast

→ Fast floating point (only consider if your application does not require full floating-point precision)



Insight from AMD Experts



General Discussions

Don't see an appropriate topic category for your post? Discuss it here first. New categories will be added as topics grow in popularity.

RSS



AMD CodeAnalyst

Collaborate with the CodeAnalyst user community. Share tips, best practices, success stories and more!

RSS



AMD Core Math Library (ACML)

Collaborate with the ACML user community. Share tips, best practices, success stories and more!

RSS



AMD Performance Library (APL)

Collaborate with the APL user community. Share tips, best practices, success stories and more!

RSS



AMD FireStream™

Collaborate with the Firestream user community. Share tips, best practices, success stories and more!

RSS



Dev Forums Archive

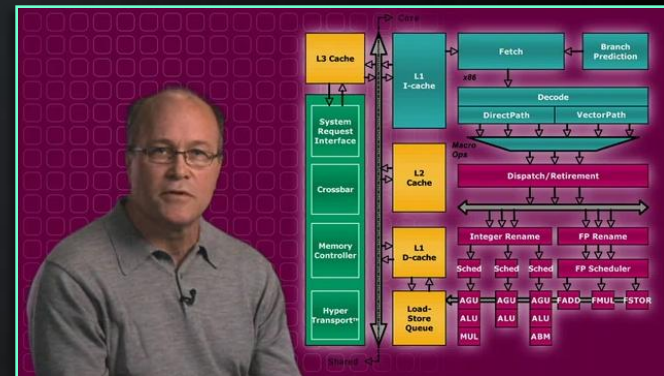
Browse read-only posts from our previous forums.

Blogs from software teams across AMD

Categories

- > [Inside Dev Central](#) (13)
- > [Hard-Core Software Optimization](#) (3)
- > [AMD High Performance Computing \(HPC\)](#) (5)
- > [AMD Libraries](#) (5)
- > [AMD "Barcelona" \(Family 10h\) Processor Software Visible Features](#) (11)
- > [AMD Java Labs](#) (1)
- > [AMD Operating System Research Center \(OSRC\)](#) (1)

Lively forum discussions with AMD engineers



Webcasts with in-depth explanations of software optimization techniques

...and more



Practical Guidance

"Shanghai" Resources

"Shanghai" Zone

AMD Developer Central > Technology Zones > "Shanghai" Zone



You can't actually see it, but there's lots of software support in there!

The new AMD "Shanghai" processors build on the foundation laid by the AMD "Barcelona" Family 10h processors with some key technology advancements. With "Barcelona," we introduced an array of innovations in processor design and features, including native quad-core architecture and a new L3 cache. The AMD "Shanghai" release brings additional enhancements for software developers. There are a number of software visible features that can be leveraged to make your applications perform better and be ready to scale across multiple cores. Visit this page regularly for updated information and practical guidance on how to take advantage of all the new features in the "Barcelona" and "Shanghai" Family 10h processors.

- » [Software Development Tools and Resources](#)
- » [Overview of Software Visible Features](#)
- » [Documentation](#)
- » [Technical Articles & Blogs](#)
- » [Related Resources](#)

Software Development Tools and Resources

The following software development tools and resources have been optimized for AMD "Barcelona" and "Shanghai" Family 10h processors:

AMD Core Math Library (ACML)

ACML is specifically designed to support multi-threading and other key features of AMD's next-generation processors. ACML currently supports OpenMP, and features hand-tuned "Barcelona" and "Shanghai" support for SGEMM and DGEMM matrix multiplication routines, and the CFFT complex-complex Fast Fourier Transforms. The newly released ACML 4.2.0 includes further tuning of DGEMM and improved performance on 3D FFTs. The newly released ACML 4.2.0 includes further tuning of DGEMM and improved performance on 3D FFTs.

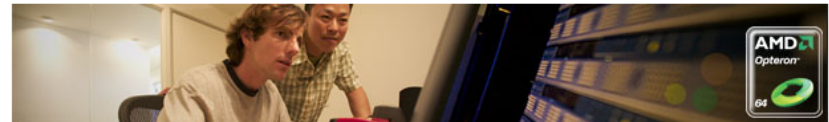
AMD CodeAnalyst™ Performance Analyzer

Shanghai builds upon the Instruction-Based Sampling (IBS) functionality that was introduced in Barcelona. Shanghai adds a new mode of operation for Instruction-Based Sampling. This mode enhances IBS op sampling. In addition to using processor cycles to select ops for monitoring and sampling, the new mode counts ops as they are dispatched and uses the count to decide when an op

Windows Zone

Windows® Zone

AMD Developer Central > Technology Zones > Windows® Zone



Microsoft Visual Studio® 2008 is the link between software and hardware as it enables software to leverage platform innovations. AMD and Microsoft closely collaborate during the development phase of our products to ensure that Visual Studio Tools generate optimized code for the latest AMD products, including "Barcelona" based platforms.

- » [Tools & Downloads](#)
- » [Documentation from MSDN](#)
- » [Related Resources](#)
- » [Technical Articles](#)

Tools & Downloads

AMD CodeAnalyst Performance Analyzer version 2.8 is integrated with the Visual Studio Tools set to help you get the optimum performance out of your code. Get an in-depth analysis and identify the hot spots of your code from the CodeAnalyst tab within Visual Studio. Just download AMD CodeAnalyst to get started.

- » [Download CodeAnalyst for Windows®](#)



Benchmarking and collecting performance data under Microsoft Windows Vista and Windows Server 2008

For the highest and most repeatable performance data on AMD processors running Windows Vista or Windows Server 2008, AMD strongly recommends using the "High performance" power plan. Follow the link below for more information including guidance on a scriptable way to manage power plans.

- » [Learn more](#)

Visual Studio 2008 Coding Tips for targeting Quad-Core AMD Opteron™ Processors

With new "Barcelona" (CPUID Family 10h) processors, including

...and more



Join AMD Developer Central

The screenshot shows the AMD Developer Central website. At the top left is the AMD logo with the tagline "The future is fusion". The main header includes "AMD Developer Central" and navigation links for "Home", "Register", and "Login". A search bar is located on the right. A left sidebar contains a menu with categories like "Drivers & Downloads", "CPU Tools", "GPU Tools", "Optimized Partner Tools", "Technology Zones", "Documentation & Articles", "Sample Code & Apps", "Community", "Programs", and "Support & Training". The main content area features a "Welcome to AMD Developer Central" banner with a photo of a man and the text "//Code Faster, Faster Code". Below this is a paragraph describing the site as a central resource for tools, technologies, and best practices. A "What's New" section lists several featured items with dates, including "AMD CodeAnalyst v2.8 Official Release Now Available" and "AMD CodeAnalyst v2.8 Official Release Now Available". A "Quick Poll" section asks "What subjects do you want to know more about from AMD?" with radio button options for "Multithreading best practices", "Cache and memory optimization techniques", "64-bit porting", "Virtualization", "Roadmap schedules", "Software visible processor features", "AMD software development tools and libraries", "AMD Developer Program", and "Other".

JOIN TODAY, IT'S FREE!

developer.amd.com



AMD CodeAnalyst™ Performance Analyzer for Windows®

AMD CodeAnalyst™ Performance Analyzer is a suite of powerful tools that analyze software performance on AMD microprocessors and help developers optimize application performance. Developers can get visibility into overall system performance, and can navigate into different modules and functions to locate hotspots within the target application.

AMD CodeAnalyst™ Performance Analyzer

Features

- System-Wide Profiling
- Timer-Based Profiling
- Event-Based Profiling
- Thread Profiling
- Instruction-Based Sampling
- Call Stack Sampling
- Pipeline Simulation
- Light weight profiling
- User-friendly graphical display
- Current version plugs directly into Microsoft® Visual Studio 2005 and 2008

Benefits

- Quickly identify optimization opportunities by pinpointing the functions and code that take the most execution time
- Discover which code regions are causing data cache misses and other performance-robbing events
- Get the most out of your parallel code by identifying poor thread affinity and remote memory access
- Get a graphical representation of core utilization and the extent to which threads can execute in parallel
- Free download for AMD Developer Central Members! Visit <http://developer.amd.com>



Framework

A collection of popular low level software routines beginning with simple arithmetic and extending into rich domains such as image and signal processing

- Leverages multi-core processors
- Helps accelerate application development
- Available as static and dynamic libraries



AMD Core Math Library (ACML)

Suite of highly tuned math functions for high performance computing

Double, Single, Single Complex, Double Complex

BLAS – Basic Linear Algebra Subprograms

- Full Level 1, 2, and 3 support
- Highly optimized DGEMM, other Level 3 BLAS
- OpenMP support for key routines

Lapack – Linear Algebra package

- Uses calls to BLAS to solve linear algebra systems
- Matrix factorization/solve, eigenvalue solutions
- OpenMP support for key routines

FFTs – Fast Fourier Transforms

- Time-to-frequency domain
- Hand-tuned assembly
- OpenMP support for 2D, 3D transforms

Fast/vector transcendental math library

- 1, 2, 4, or N values per call
- Single, Double precision (IEEE754)

RNGs – Random Number Generators

- Comprehensive reference implementation



AMD Math Libraries

LIBM – Math library functions

- Supplied to Microsoft, Novell, Sun, IBM, PGI, Pathscale, etc.
- Reference C floating point math functions
 - ⇒ C written by NAG
- Transcendental functions
 - ⇒ sin, cos, exp, log, pow, etc.
- Fastmath and vector/array versions
 - ⇒ Hand tuned assembly



Microsoft Visual Studio 2008 Tips

Visual C++ Optimization Flags

`/O2`

→ Optimize for maximum speed

`/O1`

→ (Optimize for size) rather than `/O2` (optimize for speed) may result in better performance when executing cache sensitive code. Make sure to test both flags depending on your scenario

`/GL`

→ Whole-program optimization, especially interprocedural optimizations

`/fp:fast`

→ Fast floating point (only consider if your application does not require full floating-point precision)

`/arch:SSE2`

→ For AMD64 platforms (for the 32-bit compiler only--SSE2 is already assumed by the x64 compiler)

`/OPT:icf,ref`

→ Linker flag that discards unreferenced modules and removes redundant functions

Use Visual Studio's Program-Guided Optimization

Step 1: Compile your program with the `/GL` switch and link with the `/LTCG:PGI` switch

Step 2: Now train your program, run it in real world scenarios multiple times to instrument the program

Step 3: Re-link with `/LTCG:PGO` switch to re-optimize the link order and segment positioning and favor the usage pattern represented by the instrumentation results

Other Visual Studio Tips

`/LARGEADDRESSAWARE`

→ Linker option to enable a 32-bit process running under 64-bit Windows (WOW64) to increase the available Virtual Address space to 4GB

Tips for Barcelona Processors

`/favor:blend` (for 64-bit compilation)

→ This flag is the best switch to use for Barcelona – and happens to be the default!

New SSE4a and bit manipulation intrinsics

`__lzcnt16, __lzcnt, __lzcnt64`

→ Counts the number of leading zeros in a 16-, 32-, or 64-byte integer.

`__popcnt16, __popcnt, __popcnt64`

→ Counts the number of one bits (population count) in a 16-, 32-, or 64-byte unsigned integer.

`__mm_stream_ss, __mm_stream_sd`

→ Writes 32-bit data (ss) and 64-bit (sd) to a memory location without polluting the caches.

`__rdtscp`

→ Generates the rdtscp instruction, writes `TSC_AUX[31:0]` to memory, and returns the 64-bit Time Stamp Counter (TSC) result.

`__mm_extract_si64, __mm_extracti_si64`

→ Generates the extrq instruction to extract specified bits from the low 64 bits of its first argument.

`__mm_insert_si64, __mm_inserti_si64`

→ Generates the insertq instruction to insert bits from its second operand into its first operand.

Visit <http://forums.amd.com/devblog> for more details on using the SSE4a intrinsic functions



DISCLAIMER

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions and typographical errors.

The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION.

AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY DIRECT, INDIRECT, SPECIAL OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

ATTRIBUTION

AMD, the AMD Arrow logo, ATI, the ATI logo, AMD Opteron and combinations thereof are trademarks of Advanced Micro Devices, Inc. in the United States and/or other jurisdictions. Windows is a registered trademark of Microsoft Corporation in the United States and/or other jurisdictions. HyperTransport is a licensed trademark of the HyperTransport Technology Consortium. Other names used in this presentation are for identification purposes only and may be trademarks of their respective owners.

©2008 Advanced Micro Devices, Inc. All rights reserved.

