EAMA-2

Proceedings of the 2nd Annual Workshop on Emerging Applications and Many Core Architecture

February 14, 2009

Raleigh, North Carolina

Program

Session 1. Welcome 8:30a - 9:00a

8:30 Welcome + Introduction, Victor Lee, Intel Corporation (30 min)

Session 2. Computational Medicine 9:00a - 10:00a

9:00 ""“Reconstruction and Processing Demands of Magnetic Resonance Imaging”, E. Brian Welch, Philips Medical Systems (30 min)

9:30 Computational challenges of medical image analysis, Stephen Alyward, Kitware, (30 min)

Break 10:00a - 10:30a

Session 3. Data Center Applications 10:30a - 11:45a

10:30 “Optimizing Web Search using Principled Approximation”, Trishul Chilimbi, Microsoft Corporation (45 min)

11:15 "Applying Image Recognition Techniques to Image Retrieval”, Bryan Catanzaro, UC Berkeley (30 min)

Lunch 12:00n - 1:00p

Session 4. 3D Games & Virtual World 1:00p - 3:15p

1:00 “GPU Architecture: Goals, Implications, and Emerging Directions”, David Luebke, Nvidia Corporation (45 min)

1:45 RTfact: Designing Realtime Ray Tracing Engines for Multi- and Many-Core Architectures, Philipp Slusallek, Saarland University (45 min)

2:30 Massively large Crowd Simulations on Multi-Core CPUs, Stephen Guy, UNC (45 min)

Break 3:15p- 3:30p

Session 5. Panel Discussion 3:30p - 5:00p (Pending)

3:30 Which are the biggest challenges that must be solved to allow full use of many-cores?

Title: "GPU Architecture: Goals, Implications, and Emerging Directions"
Dr. David Luebke, NVIDIA Corporation

Modern GPUs have emerged as the world's most successful parallel architecture. GPUs provide a level of massively parallel computation that was once the preserve of supercomputers like the MasPar and Connection Machine. For example NVIDIA's GeForce GTX 280 is a fully programmable, massively multithreaded chip with up to 240 cores, 30,720 threads and capable of performing up to a trillion operations per second. The raw computational horsepower of these chips has expanded their reach well beyond graphics. Today's GPUs not only render video game frames, they also accelerate physics computations, video transcoding, image processing, astrophysics, protein folding, seismic exploration, computational finance, radioastronomy - the list goes on and on. Enabled by platforms like the CUDA architecture, which provides a scalable programming model , researchers across science and engineering are accelerating applications in their discipline by up to two orders of magnitude. These success stories, and the tremendous scientific and market opportunities they open up, imply a new and diverse set of workloads that in turn carry implications for the evolution of future GPU architectures.

In this talk I will discuss the evolution of GPUs from fixed-function graphics accelerators to general-purpose massively parallel processors. I will briefly motivate GPU computing and explore the transition it represents in massively parallel computing: from the domain of supercomputers to that of commodity "manycore" hardware available to all. I will discuss the goals, implications, and key abstractions of the CUDA architecture. Finally I will close with a discussion of future workloads in games, high-performance computing, and consumer applications, and their implications for future GPU architectures.

Biography:

Dr. David Luebke
Manager, NVIDIA Research
NVIDIA Corporation
http://luebke.us

David Luebke joined NVIDIA Corporation in 2006 to help found NVIDIA Research. Previously he spent eight years on the faculty of the University of Virginia. He received his Ph.D. in Computer Science from the University of North Carolina in 1998 under the supervision of Dr. Frederick P. Brooks, Jr. Luebke's principal research interests are general-purpose GPU computing and realistic real-time computer graphics. At the University of Virginia he received both the National Science Foundation CAREER award and the Department of Energy Early Career Principal Investigator award, as well as the Test of Time award ACM Symposium on Interactive 3D Graphics "Test of Time Award" in 2005 for the paper with the most impact from the early years of the Symposium. Together with his colleagues Dr. Luebke has worked on dozens of papers, articles, chapters, and patents; a short film in the SIGGRAPH 2007 Electronic Theater; the book "Level of Detail for 3D Graphics"; and the 2003 Virtual Monticello exhibit seen by over 110,000 visitors to the New Orleans Museum of Art.

Title: Massively Large Crowd Simulations on Multi-Core CPUs

Stephen J. Guy, University of North Carolina – Chapel Hill

Simulations of large groups of agents such as people, robots, animals, soldiers, and aliens have seen a wide variety of uses from movies and video games to virtual training. Furthermore, the rise of virtual worlds, meta-verses, and other connected visual computing applications has brought a new dimension to these simulations with the possibility of thousands of virtual computer controlled agents to interactively inhabit and enliven these virtual worlds. To reach the massive scale necessary to inhabit large virtual worlds with dense virtual crowds it is necessary to spend only a few micro-seconds per agent in each time-step of a simulation.

In order to enable this, we identify the important kernels that need to be executed for obtaining high-fidelity crowd simulations. We discuss efficient implementation of these kernels which allows for the simulation of large numbers of agents simultaneously navigating in virtual environments. Our approach is based on formulating navigation decisions as an optimization problem in velocity-space for each agent. This approach is amenable to thread-level and data-level parallelization, making it an attractive option for current multi-core and future many-core architectures. We apply this approach to several large-scale, complex, heterogeneous crowd simulations and highlight its performance. The overall approach is general and can robustly handle dense scenarios with tens or hundreds of thousands of heterogeneous agents in only a few milli-seconds per frame. As compared to prior algorithms, we observe more than an order of magnitude performance improvement than similar techniques. To the best of our knowledge our work is the fastest published crowd system capable of handling agent collisions and complex obstacles.

Biography:

Stephen J. Guy is a 3rd year Computer Science Ph.D. student at the University of North Carolina – Chapel Hill. His research interests include virtual environments, multi-agent and crowd simulations, and anything interactive. He received his Bachelors of Science in Computer Engineering from the University of Virginia.

Title: RTfact: (Performance XOR Flexibility) --> (Performance AND Flexibility) Designing Realtime Ray Tracing Engines for Multi- and Many-Core Architectures

Philipp Slusallek**,** German Research Center for Artificial Intelligence (DFKI) & Saarland University

Over the past few years Realtime Ray-Tracing has become an interesting alternative to traditional rasterization-based graphics. It offers ease of use to a developer and better image quality due to its physically-based algorithm. Today, both Nvidia and Intel are optimizing their hardware architectures for this new approach, which benefits from both the general push towards Software-Based Graphics as well as from powerful many-core architectures. However, so far all high-performance ray tracing code relied on low-level optimizations in order to make best use of the available parallelism. This greatly limited the flexibility of the ray tracing engines and essentially required rewrites for every new architecture or non-trivial algorithmic change.

In this talk I will discuss RTfact a new very flexible *AND* high performance ray tracing architecture. RTfact uses C++ templates -- which form a functional programming language within C++ -- to perform code transformations during the compilation process. The compiler weaves together code as specified by the programmer at a high abstraction layer. Through inlining it generates big basic blocks that are well suited for today's highly optimizing compilers. At this point RTfact runs on x86/SSE and Cell, with ports to CUDA, AVX, and Larrabee planned or under development.

I will also discuss some ongoing work to use modern compiler technology to better make this kind of program transformations available to application developers without the complexity of C++ templates. Hopefully, this will lead to much better support for software development on emerging new hardware architectures.

Title: "Applying Image Recognition Techniques to Image Retrieval"

Bryan Catanzaro, University of California, Berkeley

The increased computational power afforded by emerging manycore platforms promises to enable new approaches to difficult problems, such as image search. Content based image retrieval systems typically perform limited image analysis, due to computational constraints. Adapting image analysis algorithms typically used in object recognition to highly parallel systems offers the potential to significantly enhance the accuracy of image retrieval. The PALLAS team at Berkeley is investigating algorithmic and implementation issues related to high-quality image analysis on manycore platforms. Specifically, we will relate our experience accelerating Support Vector Machines and algorithms for Image Contour Detection on manycore graphics processors. Our results are encouraging and lead us to conclude that applying deep image analysis algorithms, typically used for recognition problems, is now a possibility for retrieval problems. This will enable increased capabilities in image retrieval systems.

Biography:

Bryan Catanzaro studied Computer Engineering and Russian as an undergraduate at Brigham Young University, where he received his BS in 2003, and a MS in Electrical Engineering in 2005. Since then, he has been pursuing a PhD at the University of California, Berkeley, researching application frameworks for Computer Vision and Machine Learning with Professor Kurt Keutzer.

Title: “Optimizing Web-Search Using Principled Approximation”

Trishul Chilimbi, Microsoft Corporation

Energy-efficient computing is important in several systems ranging from embedded devices to large scale data centers. Several application domains offer the opportunity to tradeoff quality of service (QoS) for improvements in performance and reduction in energy consumption. Programmers sometimes take advantage of such opportunities, albeit in an ad-hoc manner and often without providing any QoS guarantees.

We present a system called Green that provides a simple and flexible framework that allows programmers to take advantage of such approximation opportunities in a systematic manner while providing statistical QoS guarantees. Green facilitates programmer approximation for expensive functions and loops and operates in two phases. In the calibration phase, it builds a model of the QoS loss produced by the approximation. Green uses this QoS model to synthesize an approximate version of the program that attempts to meet a user-specified QoS target. Green also provides a runtime recalibration mechanism that occasionally monitors runtime behavior and automatically adjusts the approximation decision as needed to meet the QoS target. Experiments on a real-world web search application indicate that Green can produce significant improvements in performance and energy consumption with small and statistically guaranteed QoS degradation.

Biography:

Trishul Chilimbi is a senior researcher at Microsoft Research leading the Runtime Analysis and Design research group. Dr. Chilimbi received his B.Tech. from IIT Bombay and PhD in Computer Science from the University of Wisconsin, Madison. His areas of interest are programming languages, compilers, runtime systems, computer architecture, and parallel and distributed systems. He is currently focused on improving the performance and energy-efficiency of web services both from a client and data center perspective.