Comparing Incoherent Ray Performance of TRaX vs. Manta

Danny Kopta Josef Spjut Erik Brunvand Steve Parker

School of Computing
University of Utah

ABSTRACT

TRaX (Threaded Ray eXecution) is a highly parallel multi-threaded, multi-core processor architecture designed for real-time ray tracing. One motivation behind TRaX is to accelerate single-ray performance instead of relying on ray-packets in SIMD mode to boost throughput, which can fail as packets become incoherent. To evaluate the effectiveness of this approach we implement a path tracer on the TRaX simulator and measure performance as the secondary rays become less coherent. We are able to show that TRaX exhibits only minor slowdown on highly incoherent rays compared to a well-optimized SIMD-packet based path tracer which suffers significant slowdown as rays become incoherent.

1 INTRODUCTION

We consider ray tracing in the context of primary rays (rays shot from the eye point to determine primary visibility hit points) and secondary rays (rays shot from intersection points within the scene to compute global effects such as shadows, reflections, refractions, and indirect illumination). A common approach to accelerating primary rays (ray casting) is to use SIMD "packets" of rays to amortize cost across sets of rays [2]. However, secondary rays often lose the coherency that makes SIMD packets effective and performance suffers on the image as a whole. Thus, an architecture that accelerates individual ray performance without relying on packets could have a distinct advantage when many secondary rays are desired.

To evaluate the effectiveness of our TRaX architecture on scenes with many non-coherent secondary rays we implemented a path tracer [3] to run on our TRaX simulator. Our path tracer allows us to vary the angle of the sampling cone to control the coherence of the secondary rays. We compared performance against Manta [1], a highly optimized and well-studied packet-based ray tracer that uses Intel SSE SIMD instructions to improve packet performance.

2 TRAX ARCHITECTURE

The TRaX architecture is a multi-core chip architecture described in more detail in [4]. Our overall chip design is a die consisting of an L2 cache with an interface to off-chip memory and a number of repeated identical cores. Each core consists of 32 thread processors, and an L1 cache and some floating point function units shared within the core. As reported in [4] we estimate that in a 150mm square die we could instantiate 22 cores (704 thread processors) in a 130nm technology and 78 cores (2496 thread processors) in a 65nm technology using standard cell based implementations at a conservative 500MHz. Higher densities and speeds would be achievable with more use of custom circuits.

3 PATH TRACER

Our path tracer is written specifically to be able to control the coherence of the secondary rays. We use a single point light source, and limit incoherence to Monte-Carlo sampled Lambertian shading with no reflective or refractive materials. Every ray path is traced to the same depth: there is no Russian Roulette or any other dynamic decision making that could change the number of rays cast. This is all to ensure that we can reliably control secondary ray coherence for these experiments. A more fully functional path tracer with these additional techniques could be written using the TRaX programming language, and we expect it would have similar performance characteristics.

Secondary rays are randomly distributed over the hemisphere according to a BRDF. To compute a cosine-weighted Lambertian BRDF, a random sample is taken on the area of a cone with the major axis of the cone parallel to the normal of the hit geometry and the vertex at the hit point. As an artificial benchmark, we limit the angle of this cone anywhere from 0 degrees (the sample is always taken in the exact direction of the normal) to 180 degrees (correct Lambertian shading on a full hemisphere). By controlling the angle of the cone we can control the coherence of the secondary rays.

4 PERFORMANCE RESULTS

To compare the performance of TRaX on secondary rays to a packet-based ray tracer that relies on keeping SIMD packets coherent, we modified the Manta ray tracer [1] to behave in the same way as our path tracer. We measure our performance results for incoherent secondary rays in terms of relative speed compared to highly coherent primary rays. We measure speed for several different sampling cone angles for both TRaX and Manta using the Cornell Box, Sponza, and Conference scenes. We show that TRaX has only moderate slowdown of 0.87 to 0.75 when secondary rays become highly incoherent (cone angle of 180 degrees) whereas Manta slows down to 0.53 to 0.47 of its full speed. We also estimate (using the TRaX cycle-accurate simulator) that the raw speed of one TRaX chip at 500MHz is ˜40x faster than Manta running on one Intel Core2 Duo at 2GHz. We attribute the difference in performance between TRaX and Manta to the overhead of dealing with small packets and the breakdown of the SIMD operation as the packets become highly incoherent.TRaX performance does not suffer as much because each thread processor within a core is responsible for a single ray-thread. The reason TRaX has a slowdown at all is due to a reduction in cache coherence as rays become incoherent and successive rays access a wider range of the scene data.

As higher realism becomes important for interactive ray tracing, we believe that a hardware solution that does not rely on packets for performance and therefore performs well even for highly incoherent secondary rays will be very interesting.

REFERENCES

[1] J. Bigler, A. Stephens, and S. Parker. Design for parallel interactive ray tracing systems. *Interactive Ray Tracing 2006, IEEE Symposium on*, pages 187–196, Sept. 2006.

[2] S. Boulos, D. Edwards, J. D. Lacewell, J. Kniss, J. Kautz, P. Shirley, and I. Wald. Packet-based Whitted and Distribution Ray Tracing. In *Proc. Graphics Interface*, May 2007.

[3] J. T. Kajiya. The rendering equation. In *Proceedings of SIGGRAPH*, pages 143–150, 1986.

[4] J. Spjut, D. Kopta, S. Boulos, S. Kellis, and E. Brunvand. TRaX: A multi-threaded architecture for real-time ray tracing. In *6th IEEE Symposium on Application Specific Processors (SASP)*, June 2008.