Preparing Ray Tracing for Exascale

Ellen A. Porter* Robert R. Lewis†

Washington State University, Tri-Cities
Computer Science

ABSTRACT

Ideas for titles:
Ray Tracing for Exascale Computing.
An approahc for Ray Tracing on Exascale Computers.
Adapting Ray Tracing for Exascale.

To reach exascale computing, significant hardware and software changes will be introduced to high-perfomance computing (HPC). Applications will need to adapt as the architecture of supercomputers change. In this paper we explore the history of ray tracing given architecture changes and propose a methodology using the Intel Concurrent Collections (CnC) Programming Model and Intel Embree Ray Tracing Engine to build scalable ray tracing system.

And this is what references look like [**?**].

Index Terms: K.6.1 [Management of Computing and Information Systems]: Project and People Management—Life Cycle; K.7.m [The Computing Profession]: Miscellaneous—Ethics

1 INTRODUCTION

Reaching exascale performance goals will require changes to node and system architecture.

Programming models and runtime systems are advancing as well to take advantage of the new systems.

Changes to the runtime: tuning hints
Changes to the models: hierarchical programming models (CnC)

Current algorithms will need to be rewritten to take full advantage of the benifits of these new systems in order to scale to exascale.

Ray tracing algorithms have adapted in the past to hardware advances.

Some methods that did not work before due to load balancing may work well.

Summary of what we need in an application designed for future exascale systems and description of how our approach should work well.
Other features of exascale we can take advantage of (hierarchy?)
Future directions.

2 REACHING EXASCALE

Motivation: Moore's law

*e-mail: ellen.porter@wsu.edu
†e-mail:bobl@tricity.wsu.edu

Table 1: Vis Paper Acceptance Rate

Year	Submitted	Accepted	Accepted (%)
1994	91	41	45.1
1995	102	41	40.2
1996	101	43	42.6
1997	117	44	37.6
1998	118	50	42.4
1999	129	47	36.4
2000	151	52	34.4
2001	152	51	33.6
2002	172	58	33.7
2003	192	63	32.8
2004	167	46	27.6
2005	268	88	32.8
2006	228	63	27.6

2.1 Hardware Changes

Heterogeneity, node failures

2.2 Software Changes

x86, MPI, OpenMP
Optimize for data movement, communication avoidance algorithms

3 CNC PROGRAMMING MODEL

Model built for task based runtimes, work moves to data, relies on tag collections

3.1 Tag Collections

Used to organize steps and items

3.2 Steps

Computation to be done

3.3 Item Collections

Data to be used

3.4 Tuning

Hints that can be provided to the runtime

4 ADAPTING RAY TRACING

Modifications made for parallel ray tracing

4.1 History/Previous Work?

Discuss single node to distributed systems

4.2 Changes for Future Systems

Discuss reducing data movement and avoiding communication

5 METHODOLOGY

High level description

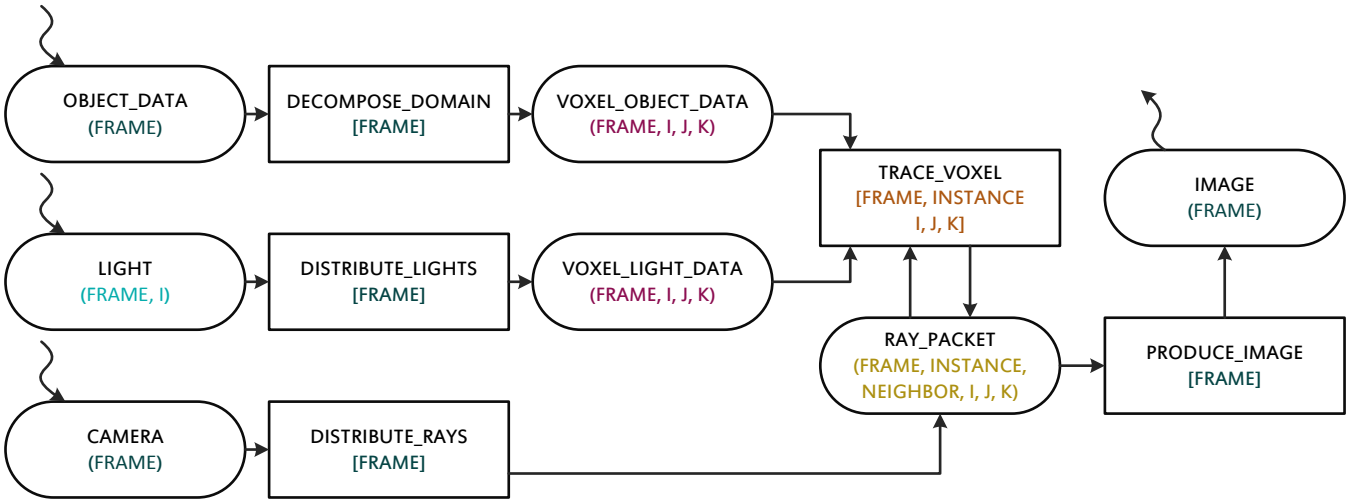

Figure 1: CnC graph

5.1 CnC Steps

Details on the CnC Steps implemented (code?)

5.1.1 Decompose Domain

5.1.2 Distribute Lights

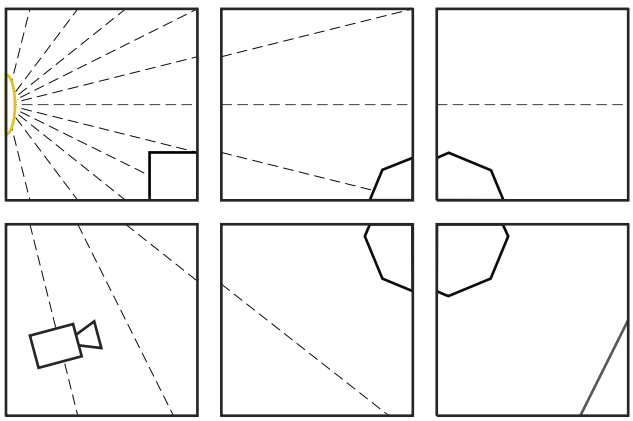

Figure 2: Light Source Rays

5.1.3 Distribute Rays

5.1.4 Trace Voxel

Embree kernel ray tracer

5.1.5 Step: Produce Image

5.2 CnC Item Collections

Details on the data (code?)

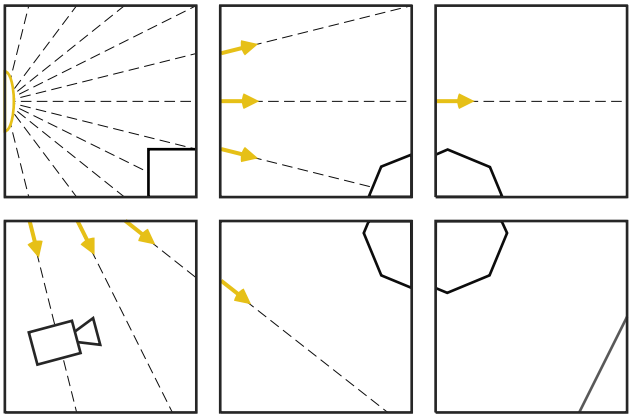

Figure 3: Distributed Point Source Rays

5.2.1 Object Data

5.2.2 Voxel Object Data

5.2.3 Light

5.2.4 Voxel Light Data

5.2.5 Camera

5.2.6 Ray Packet

5.2.7 Image

5.3 Tuning Spec

If we get to adding a tuning spec, describe and show code

5.4 Running the application

Do we need a section describing how to run the code?

5.5 Performance

System where we ran the code, results summary

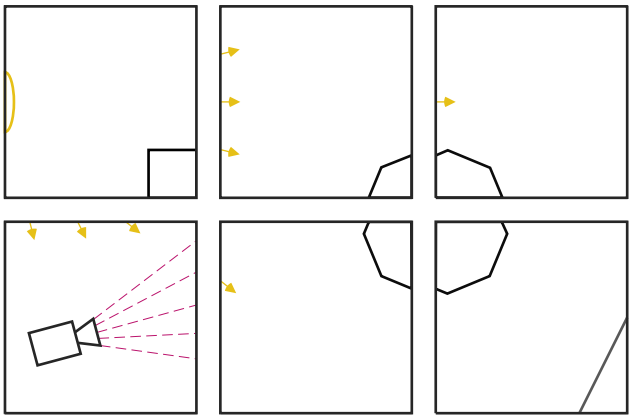

Figure 4: Distribute Rays

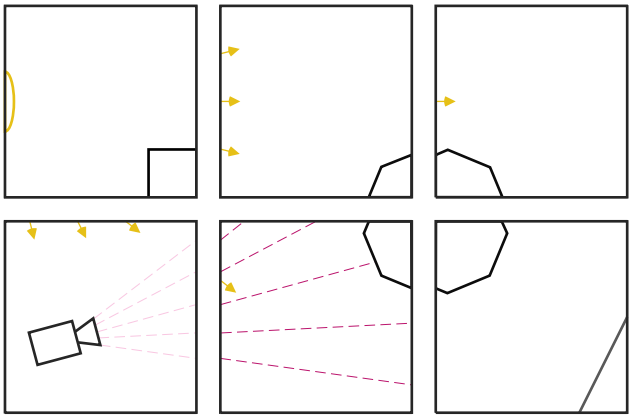

Figure 5: Trace Voxel

6 Other Exascale Features

6.1 Hierarchy

Dynamic scenes, checkpoint and restart

7 Future Directions

Global illumination [1].

8 Conclusion

Summary

Acknowledgements

The authors wish to thank A, B, C. CnC team at RICE? Intel for the cluster?

References

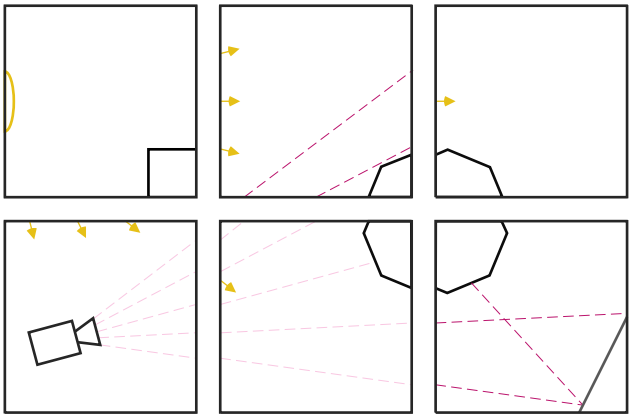

Figure 6: Trace Voxel

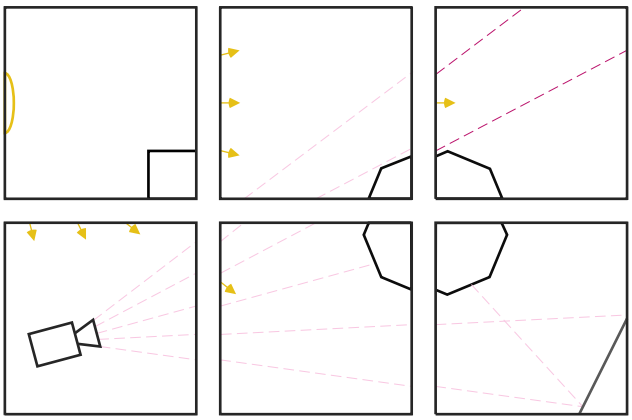

Figure 7: Trace Voxel

[1] Footnotes appear at the bottom of the column