R2E2: Low-Latency Path Tracing of Terabyte-Scale Scenes using Thousands of Cloud CPUs

Rasul Khanbayov

Technische Universität München

May 25, 2023

1 Cloud Computing

Cloud computing platforms offer users the flexibility to access many CPUs, extensive memory, and high bandwidth. Setting up this environment is easy, and it will cost around 1.30 $ a minute to run 1000 CPUs in Amazon Cloud. With the help of abundant computing resources, new opportunities arise to reduce the runtime of graphics jobs and increase productivity in content creation workflows. Elastic cloud resources can accelerate one of the challenging tasks of graphics and visualization, achieving low latency path tracing for high-complexity scenes that are too large for individual render farm machines.

High-performance path tracing is usually carried out on servers equipped with multiple cores and large, unified memory, either on CPUs or GPUs. Render farm machines often need hundreds of gigabytes of memory, and users must make modifications to fit the scene into the limited memory capacity of these nodes. These modifications resolve the issues of larger scenes when they exceed available memory and cause the performance of path tracing.

On the other hand, cloud platforms are optimized for dynamic demand and efficient resource allocation. As a result, cloud platforms perform well at distributing resources in smaller units, and users can rapidly acquire numerous cloud computing resources by utilizing many smaller nodes. To achieve low-latency job completion using cloud resources, path tracers must be designed for massive scale-out execution across distributed nodes, each handling a fraction of the scene.

2 Goals of R2E2

R2E2 aimed at rendering scenes containing hundreds of gigabytes to terabytes of geometry and texture data in just a few minutes by performing low latency path tracing using serverless computing nodes in the cloud. Usually, we encounter high-complexity scenes in film productions or even in scientific data visualization areas which most single-machine CPU/GPU platforms cannot handle high-bandwidth. These scenes can appear in high-resolution CAD models, 3D scans of large environments, and more. R2E2 enables users to render a large scene with path-traced global illumination quickly. We will only focus on reducing the end-to-end time of the path tracing computation, including time to acquire processing resources and load the scene from persistent storage.

3 Architecture

R2E2 system maps path tracing onto the massively parallel cloud infrastructure. It adopts a domain partitioning approach, which divides the scene's BVH and leaf geometry into treelets, while the scene texture data is divided into texture partitions. These partitions are sized to fit within the limited memory of worker nodes.

R2E2 is structured as a collection of asynchronous services that communicate through queues. Overall there are five types of services. These services include the Camera service for generating camera rays, the $\mathrm{Trace_t}$ services for tracing through specific treelet t, and the $\mathrm{Shade_p}$ and $\mathrm{EnvLight_p}$ service for BRDF evaluation and environment map sampling using texture partition p, respectively. In addition, it includes the Accum service for aggregating ray radiance contributions into the frame buffer.

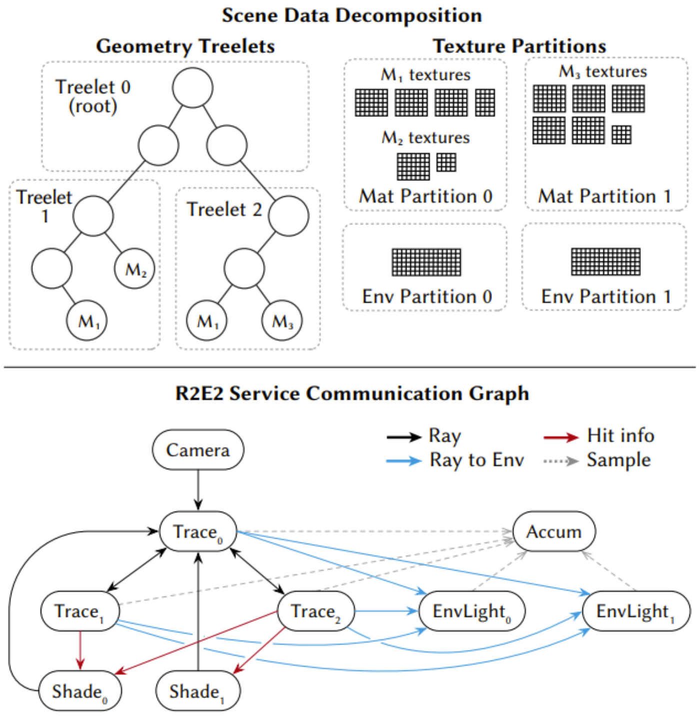

FIG 1. *Top: Example Geometry Treelets and Texture Partitions Bottom: The R2E2 service communication graph resulting from Top scene decomposition.*

The system's scened data decomposition is depicted in FIG. 1, illustrating scene partitioning into treelets and texture partitions, along with the corresponding R2E2 service communication graph. Each treelet has a corresponding trace service, while each texture partition has a Shade or EnvLight service.

After the scene, treelets, and texture partitions are ready, it is followed by performing fire-and-forget path tracing. In this phase, the Camera service generates rays and communicates them to the root treelet's tracing service (Trace_0). Rays can traverse to other treelets, or intersect geometry within the current treelet. Once it hits, the ray is directed to the shading service associated with the relevant texture partition. Shading generates two rays: a bounce ray to continue the path and a shadow ray containing the shading result. These new rays are then sent to Trace_0 for further traversal. Rays that miss the scene and require environment map sampling are forwarded to the appropriate EnvLight service. Finally, radiance along unoccluded shadow rays and the results of environment map sampling are sent to the Accum service, which accumulates samples into the render target.

To sum up, R2E2 utilizes domain partitioning and asynchronous services to achieve efficient and scalable path tracing on cloud infrastructure, enhancing the performance and capabilities of the system.

4 Evaluation

In the evaluation of R2E2, its performance is assessed in terms of wall-clock latency and scalability for large scenes and numerous worker nodes. Findings demonstrate that R2E2 achieves considerably faster rendering speeds (up to 8.6 times) for terabyte-scale scenes compared to a high-end shared memory server.

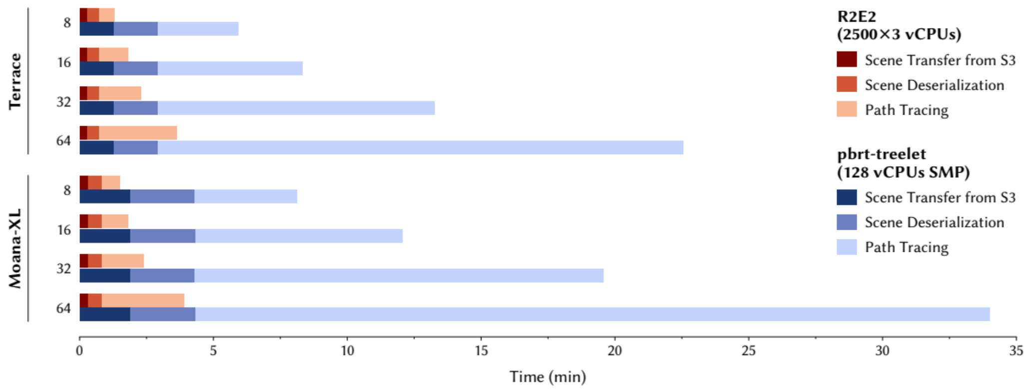

FIG 2. *The table compares the performance of R2E2 and pbrt-treelet on the Moana-XL and Terrace scenes. R2E2 demonstrates faster rendering times compared to pbrt-treelet in the Moana-XL scene, while in the Terrace scene, R2E2 shows comparable rendering times and superior performance compared to pbrt-treelet.*

The figure above compares R2E2's performance against a version of pbrt that requires the entire scene to be materialized in memory during ray tracing. R2E2 is configured with 2500 AWS Lambda nodes with 3 CPUs and 4 GB of RAM. In addition, 250 Amazon EC2 nodes are also accompanied for ray queening support, where every 10 AWS Lambda nodes are connected to one EC2 node. On the other hand, the pbrt-treelet is configured with 1 Amazon EC2 node with 128 CPUs and 2 TB of RAM. As it is seen from the table, R2E2 is 3.5–7.8× faster than pbrt-treelet's single-machine in-memory path tracer. The reason for fast path-tracing is the sheer number of cores that R2E2 used. Except for the Terrace 64 spp configuration, R2E2 finishes the entire path tracing task before pbrt-treelet initiates the tracing of a single ray. This indicates that employing a different approach of parallelizing pbrt-treelet across multiple large 128-vCPU machines using image-decomposition rendering would not surpass the performance of R2E2 on these jobs, regardless of the number of machines utilized.

5 Conclusion

R2E2 demonstrates the possibility of constructing a cloud-supercomputer using multiple small elastic cloud nodes. By utilizing these resources, R2E2 reduces the overall time it takes to complete path tracing for scenes of massive scale. The project still has future work on additional components of high-quality rendering systems. Such as BVH and treelet construction, can be mapped onto widely available and parallel cloud platforms. Other than that, the project proves how elastic platforms enable us to build graphics applications that are impossible on the single-machine CPU and GPU platforms today.