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

RASUL KHANBAYOV

Graphics System

TIME BANDIT
(VERSION 2.0)
PRESS F1 & F2 TO CHANGE CONTROLLERS
PRESS "1" OR "2" TO START THE ADVENTURE
ATARI
520

Cloud Computing

- Cloud computing platforms provide users the ability to access thousands of CPUs, featuring terabytes of aggregate memory and hundreds of gigabytes of I/O bandwidth, on demand.

aws

Google App Engine

Microsoft Azure

Cloud Computing

- For example:

- Running 1,000 virtual CPUs with 1.3 TB of RAM and up to 200 Gbit/s of bandwidth to shared storage in the Amazon cloud for one minute costs approximately $1.30 USD.

Helps to achieve low latency path tracing of high-complexity scenes

R2E2 for Cloud Platforms

- Uses Elastic Cloud Platforms:
  - To leverage the unique strengths such as availability of many CPUs/memory in aggregate, and massively parallel access to shared storage
  - Mitigate the cloud's limitations - low per node memory capacity and high latency inter-node communication

What does R2E2 do?

1. Rapidly acquires thousands of cloud CPU cores
2. Loads scene geometry (from a pre-built scene BVH) and texture data into the aggregate memory of these nodes in parallel
3. Performs full path traced global illumination using an inter-node messaging service specifically designed for communicating ray data over commodity interconnect links.

Related work

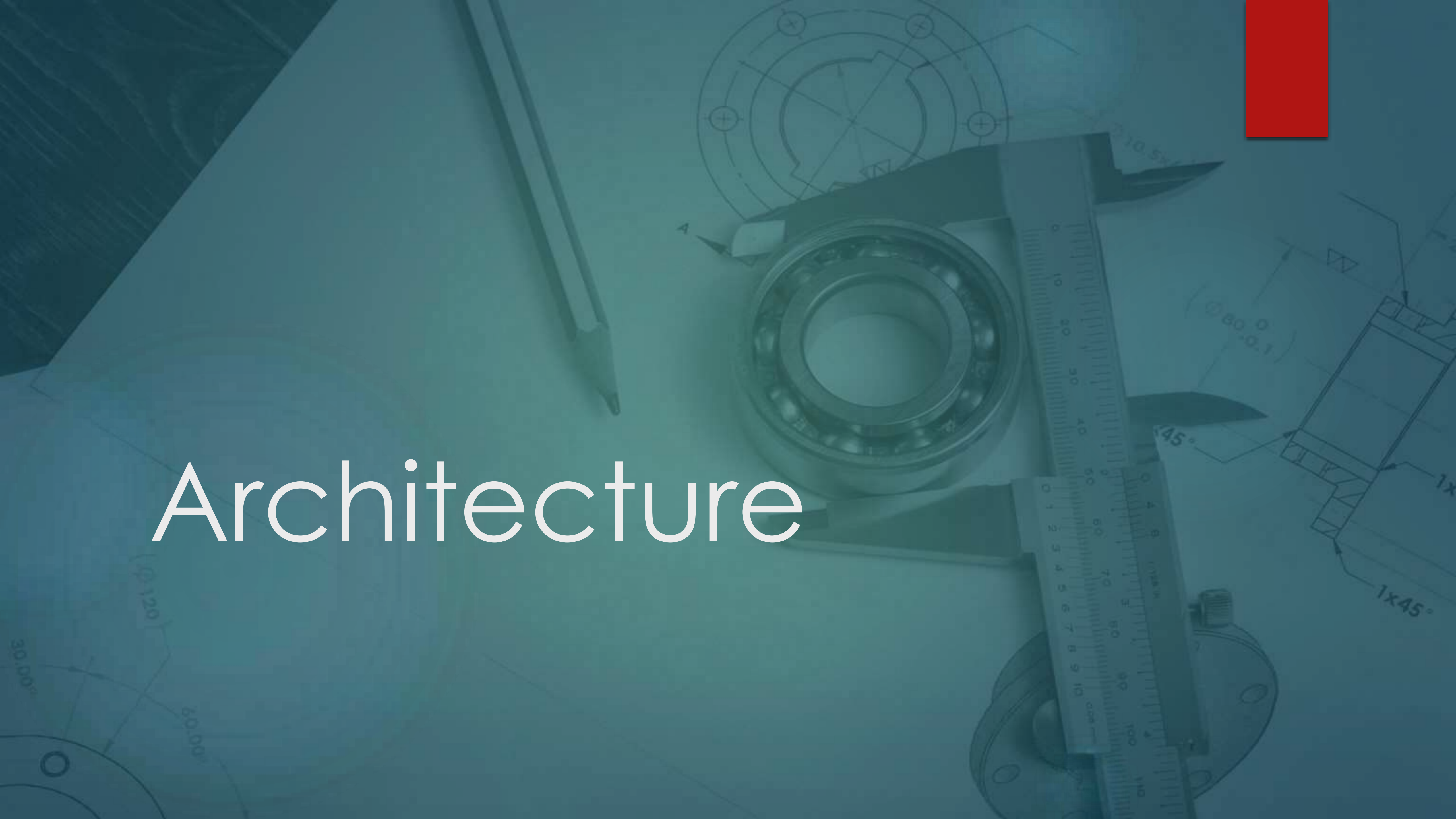
- Cleary et al. 1986; Dippé and Swensen 1984; Kobayashi et al. 1988; Nemoto and Omachi 1986; Priol and Bouatouch 1989; Salmon and Goldsmith 1989; Scherson and Caspary 1988 - Early attempts to design ray tracing algorithms
- Kato and Saito 2002; Navrátil et al. 2014; Pharr et al. 1997; Reinhard et al. 1999; Son and Yoon 2017 - To distribute ray tracing onto commodity clusters
- Parker et al. 2010; Wald et al. 2014; Ylitie et al. 2017a - modern multi-core CPUs/GPUs

Related work

- Christensen et al. 2003; Georgiev et al. 2018 - Lazy loading or lazy generation of scene geometry
- Budge et al. 2009; Eisenacher et al. 2013; Navrátil et al. 2014; Pantaleoni et al. 2010; Pharr et al. 1997; Reinhard et al. 1999; Son and Yoon 2017 - Hybrid techniques
- Benthin et al. 2018; Mahovsky and Wyvill 2006; Ylitie et al. 2017b - Scene data compression techniques
- Burley et al. 2018; Pantaleoni et al. 2010 - Out-of-core rendering methods
- ExCamera [Fouladi et al. 2017] and Sprocket [Ao et al. 2018] for low-latency video processing, PyWren [Jonas et al. 2017] for MapReduce-style data analytics, numpywren [Shankar et al. 2018] for linear algebra, gg [Fouladi et al. 2019] for software compilation and testing, and Cirrus [Carreira et al. 2019] for machine learning workflows.

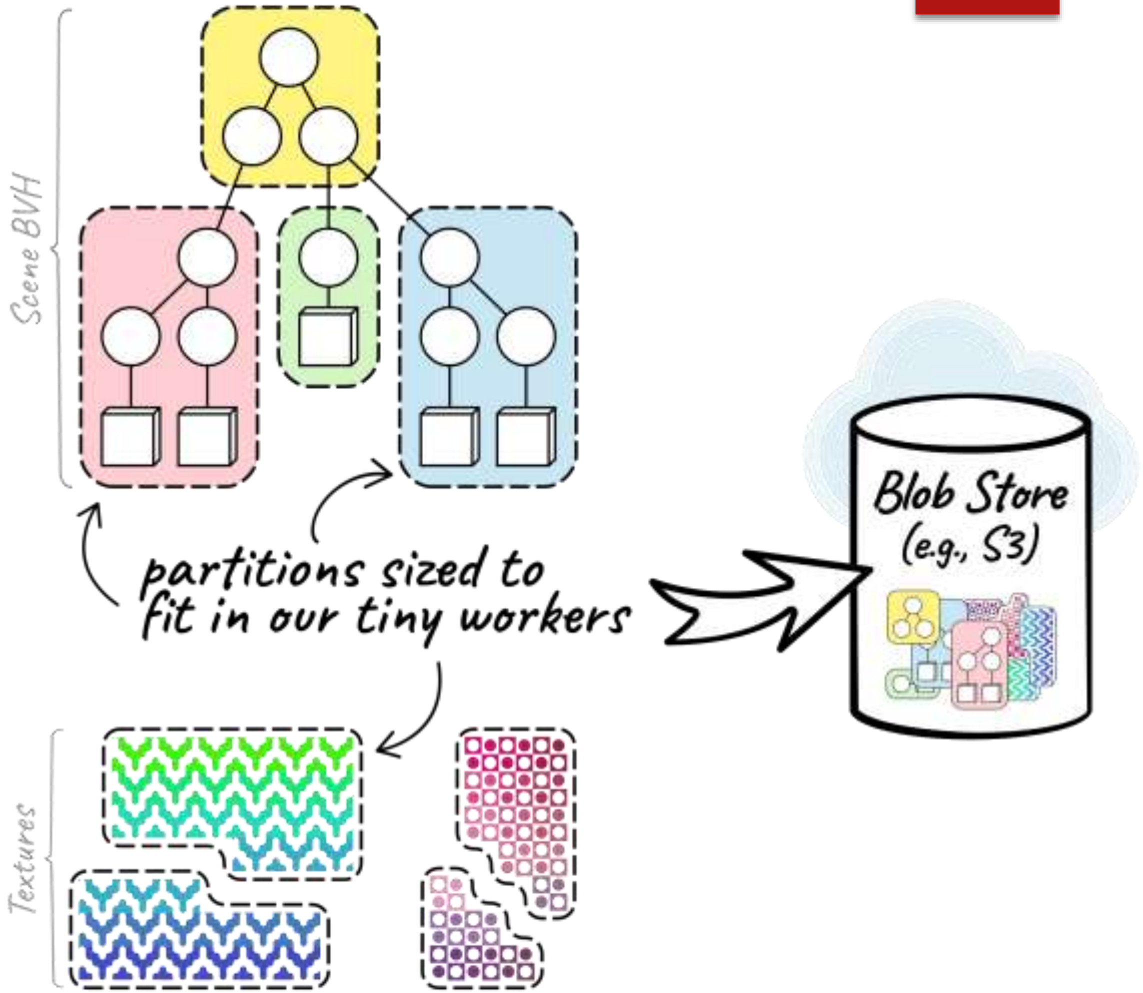
Goals

- R2E2 Goals
- Designing for Cloud Platform Characteristics
  - Many “small” nodes
  - Large aggregate memory capacity
  - Large aggregate I/O throughput
  - High communication latency.

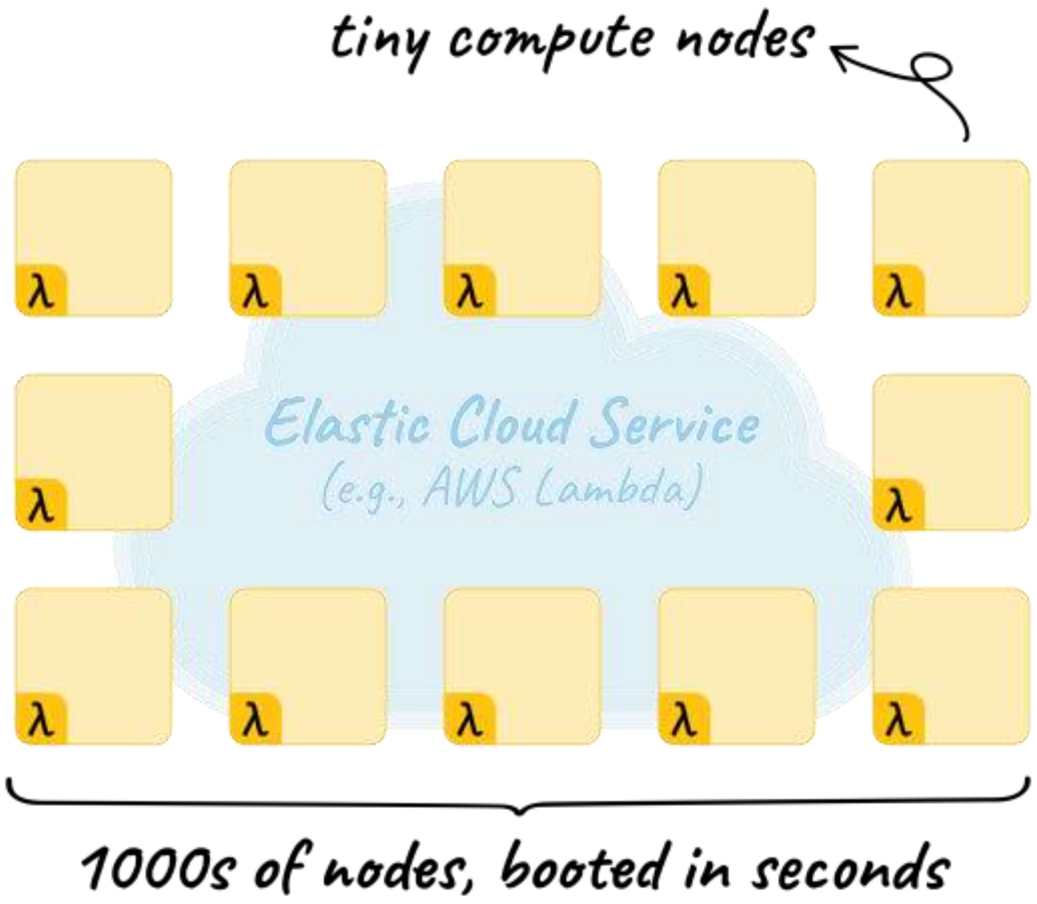


Architecture

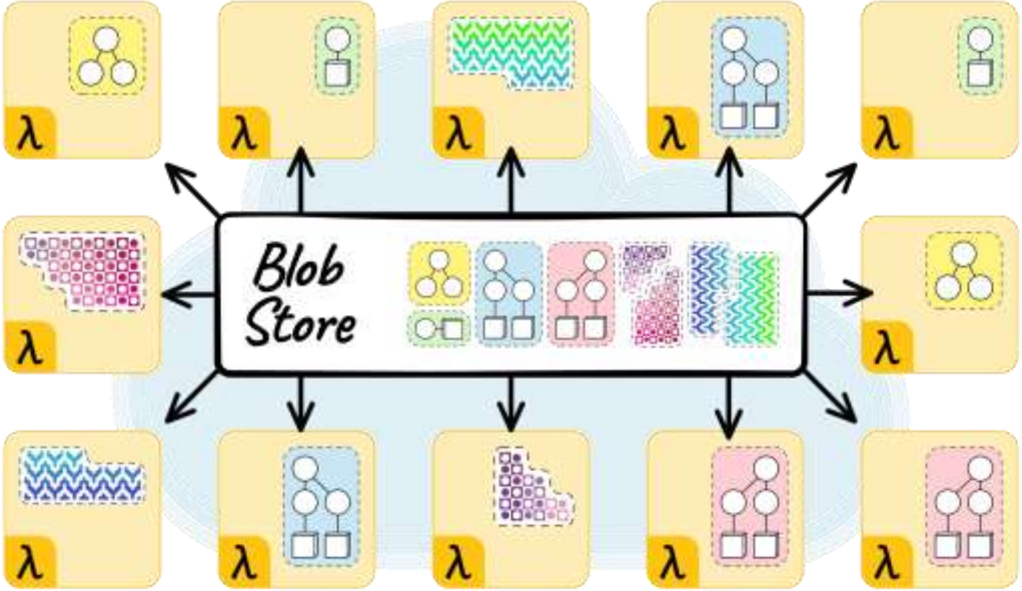
- R2E2 divides the scene's BVH and leaf geometry into treelets, while the scene texture data is divided into texture partitions.
- Then put them in a blob store.



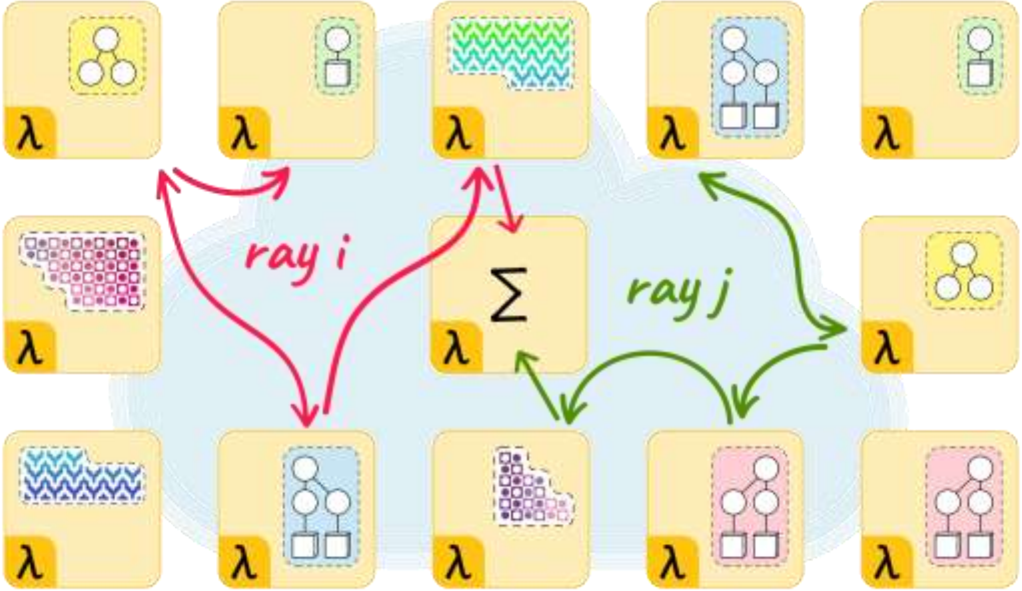
- The system rapidly initiates thousands of small computing nodes.



- The nodes retrieve scene objects from the blob store and assume the responsibility of serving their respective objects

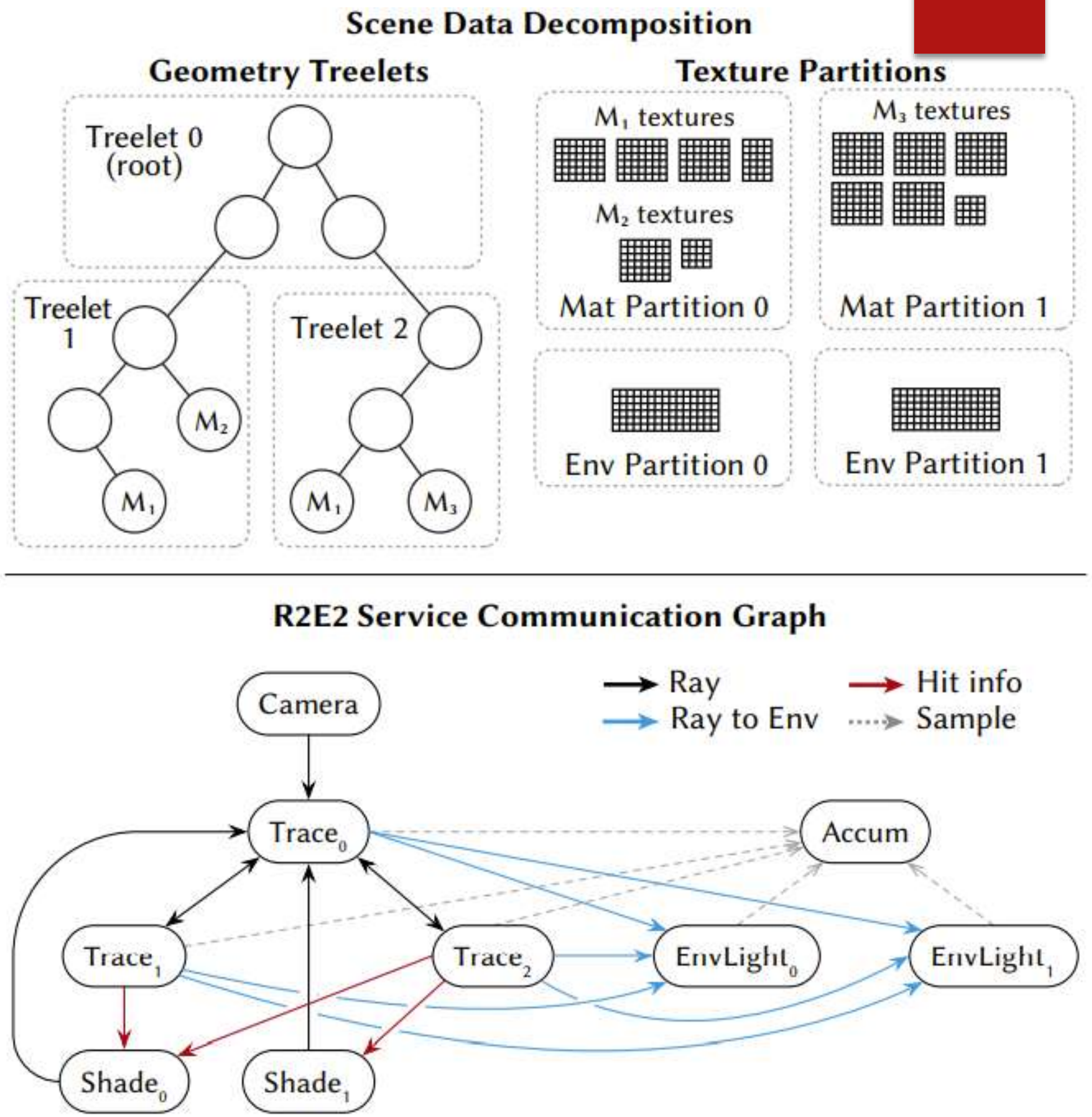


- The nodes collaborate and exchange rays among themselves to execute standard fire-and-forget path tracing



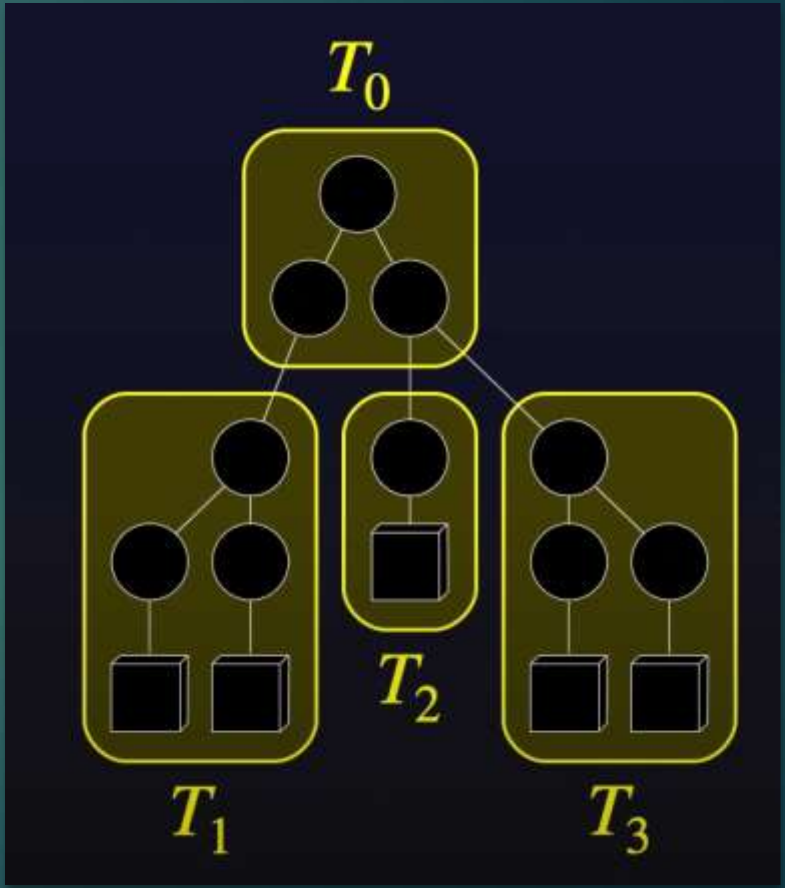
▶There are five types of services:

1. Camera service
2. Trace service
3. Shade service
4. EnvLight service
5. Accum service



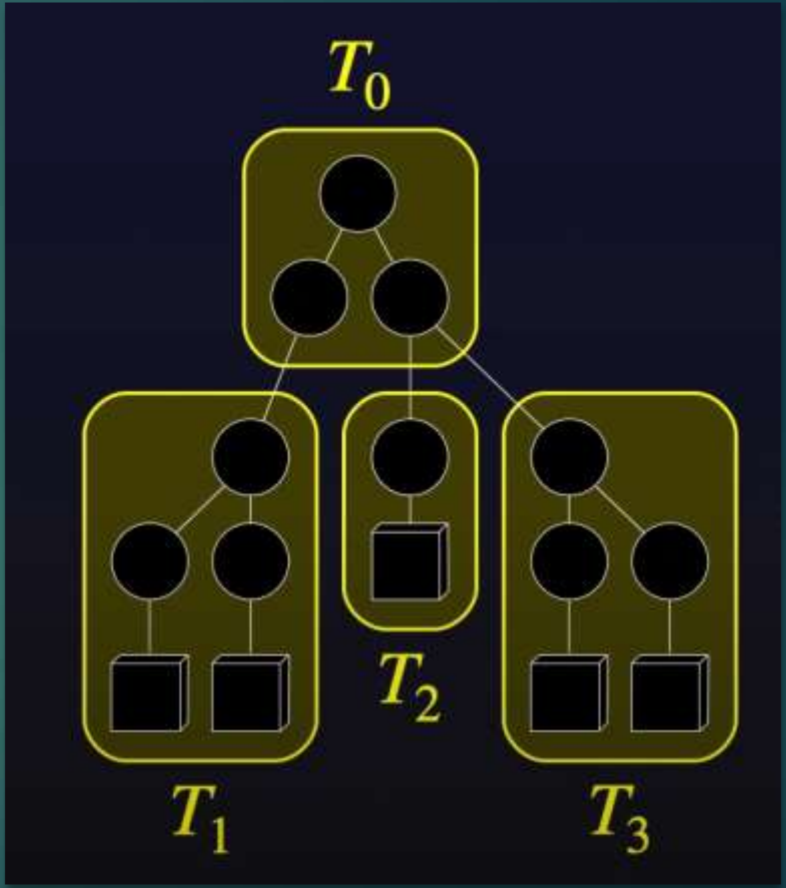
Locality preserving scene partitioning

- Treelet-to-treelet ray transfers during traversal must be infrequent.
- The solution: Algorithm of [Aila and Karras 2010]
- We enhance the partitioning heuristic to account for the communication costs of geometry instancing (extensively used in real-world production scenes) and scale-up target treelet size from a few KB (targeting GPU L1 caches) to ~1 GB to fill the memory of cloud workers



Ensuring good workload balance.

- Generating a good allocation of worker nodes to services is essential.
- Profiling phase performs a low resolution, low path depth rendering of the full scene using one worker per service. Profiling records the total number of rays processed by each service as well as the average computation time for processing requests
- R2E2 allocates workers to treelet services proportionally to the profiling run.



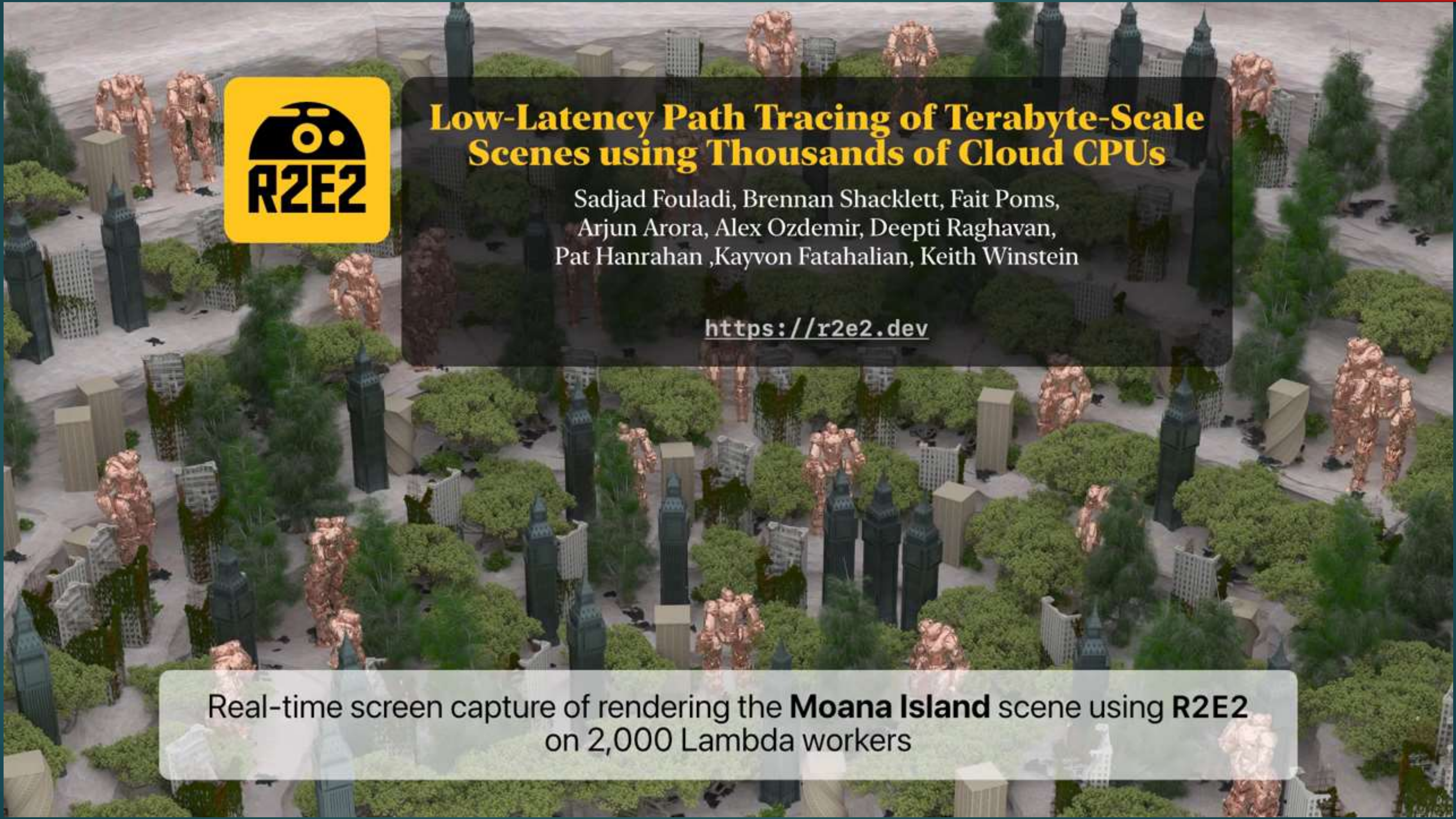
Efficient ray communication.

- In addition to communicating ray data less often, R2E2 must transmit fine granularity ray data efficiently over commodity networking links.

- Off-the-shelf cloud messaging services such as Amazon’s Simple Queuing Service deliver insufficient throughput and too high of latency for this workload

- The service queue store is implemented by a collection of nodes running the memcached in-memory key-value store. A ratio of one ray store server for every 10 R2E2 worker nodes is sufficient to maintain high communication bandwidth across the system.

Demo video



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

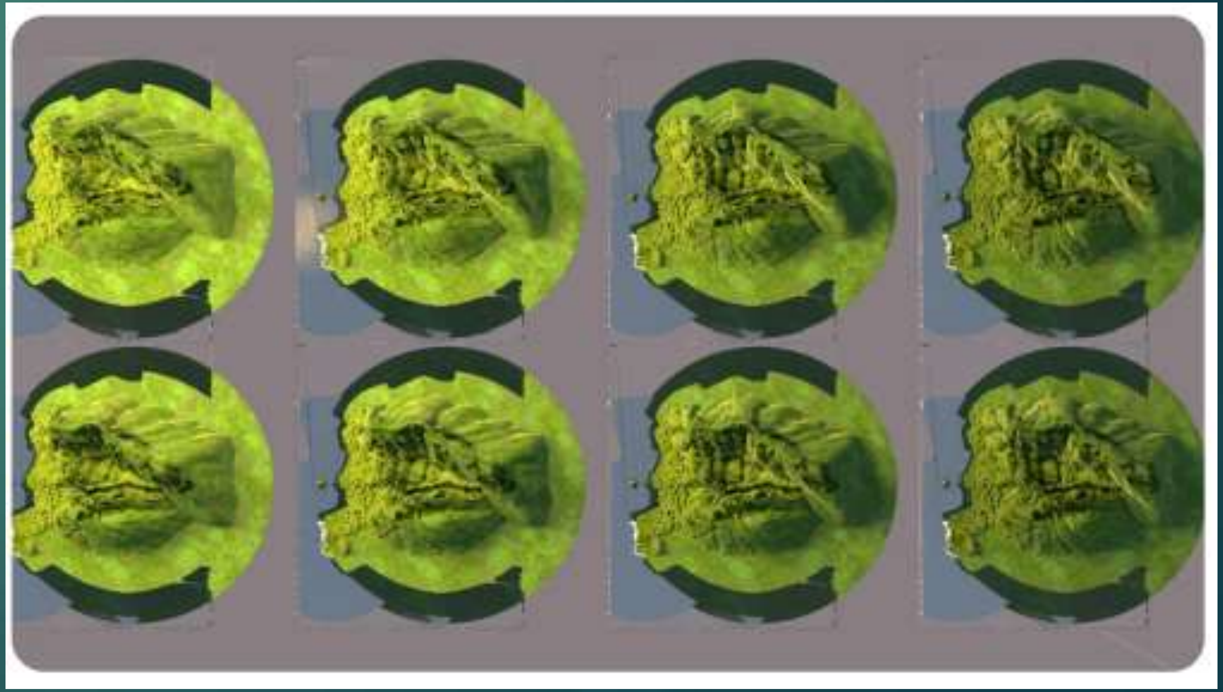
Sadjad Fouladi, Brennan Shacklett, Fait Poms,
Arjun Arora, Alex Ozdemir, Deepti Raghavan,
Pat Hanrahan ,Kayvon Fatahalian, Keith Winstein

https://r2e2.dev

Real-time screen capture of rendering the **Moana Island** scene using **R2E2** on 2,000 Lambda workers

Evaluation

Moana-XL



Moana-XL’s geometry and texture data occupies approximately 1 TB when materialized in memory.

Terrace

Terrace is also sized to require approximately 1 TB of memory.

Configuration

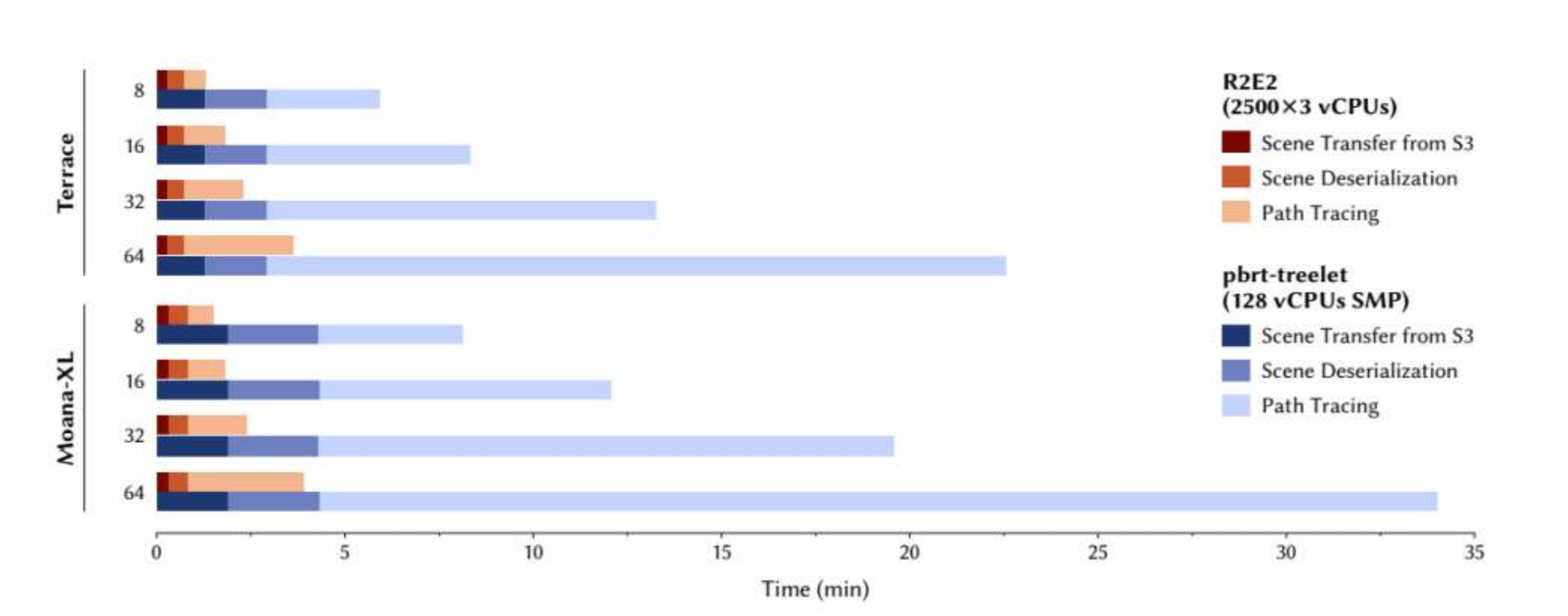
R2E2

- 2500 AWS Lambda nodes
  - 3 CPUs and 4 GB of RAM
- 250 Amazon EC2
  - 2 vCPUs, 5.25 GB of RAM

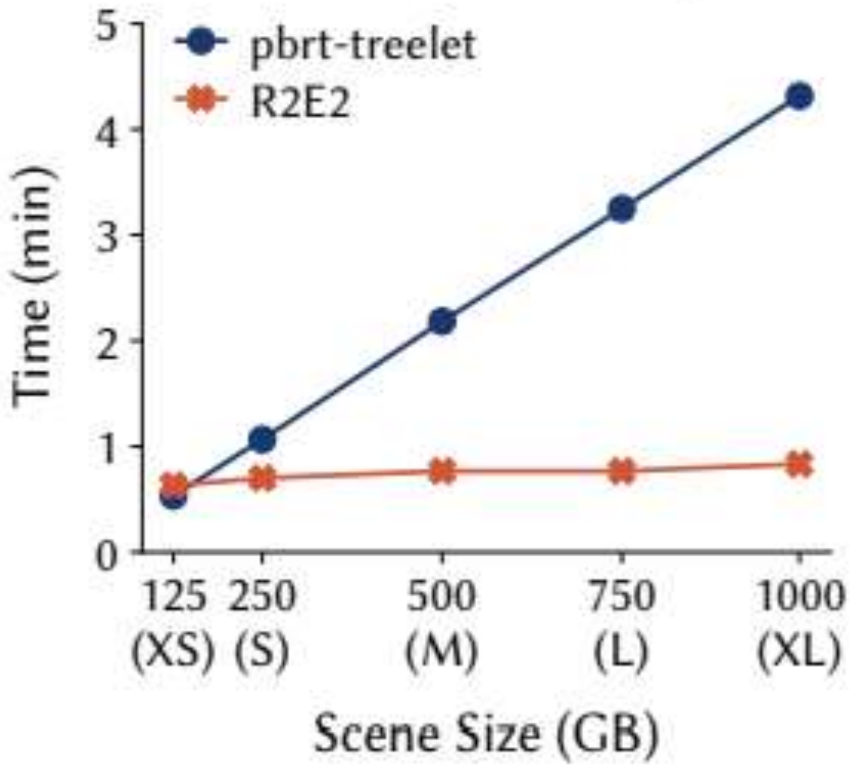
Baseline (pbrt-treelet)

- 1 Amazon EC2 node
  - 128 CPUs and 2 TB of RAM

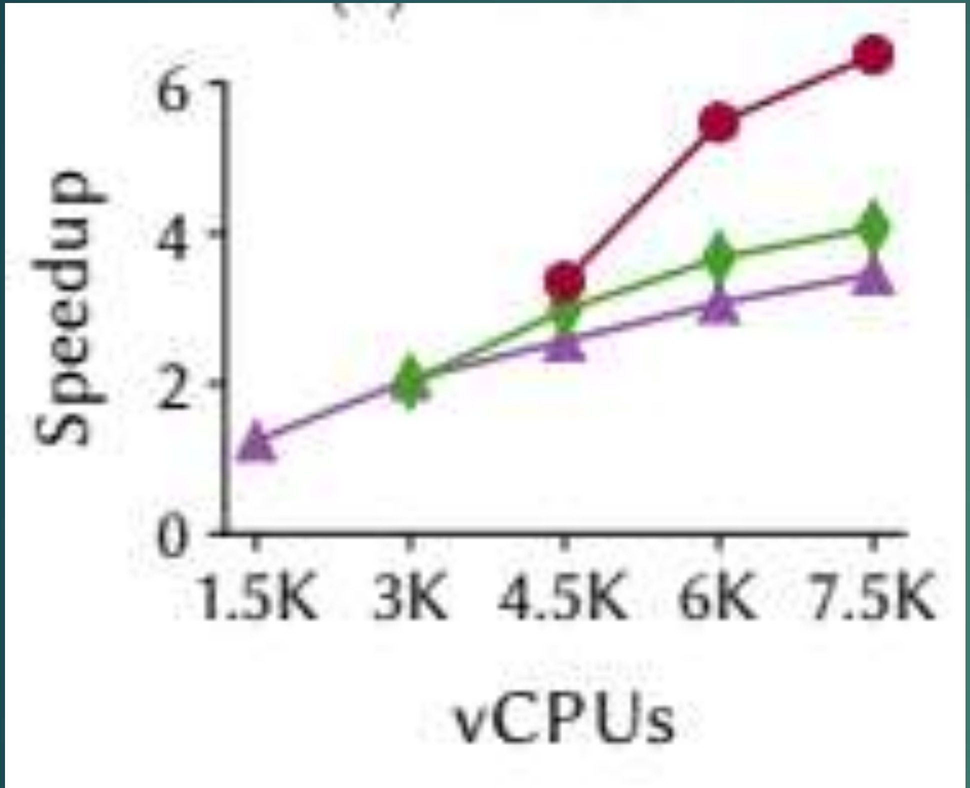
Rendering scenes at 4K resolution



Scene Loading Time



R2E2 – increasing vCPUs



Summary

- We proved that it is possible to construct a “supercomputer on the fly” from many small elastic cloud nodes, and use these resources to reduce end-to-end job latency when path tracing terabyte-scale scenes

- Future work: Continue to explore how additional components of high-quality rendering systems, such as BVH and treelet construction, can be mapped onto widely available, parallel cloud platforms