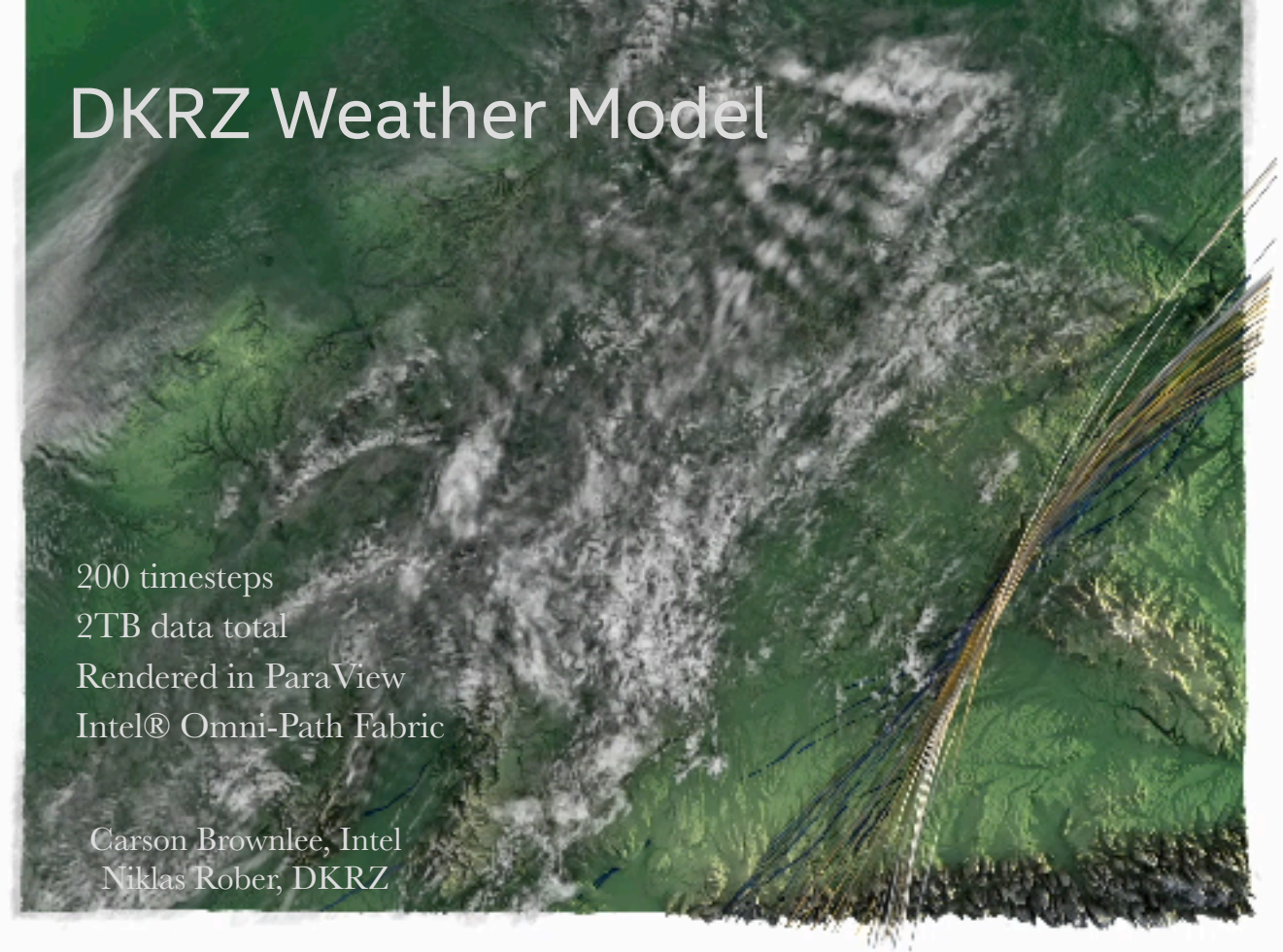
DISTRIBUTED RAY TRACING WITH INTEL® OSPRAY & PARAVIEW

Carson Brownlee - Intel

DKRZ Weather Model



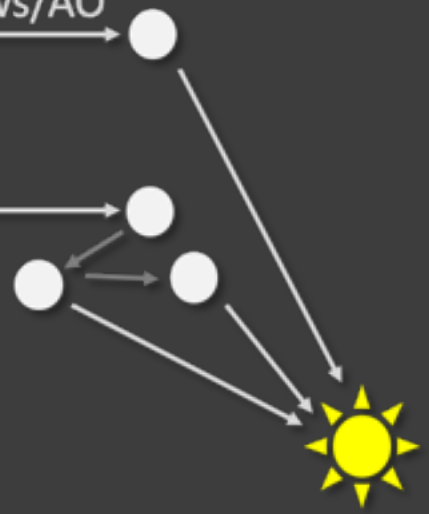
200 timesteps
2TB data total
Rendered in ParaView
Intel® Omni-Path Fabric

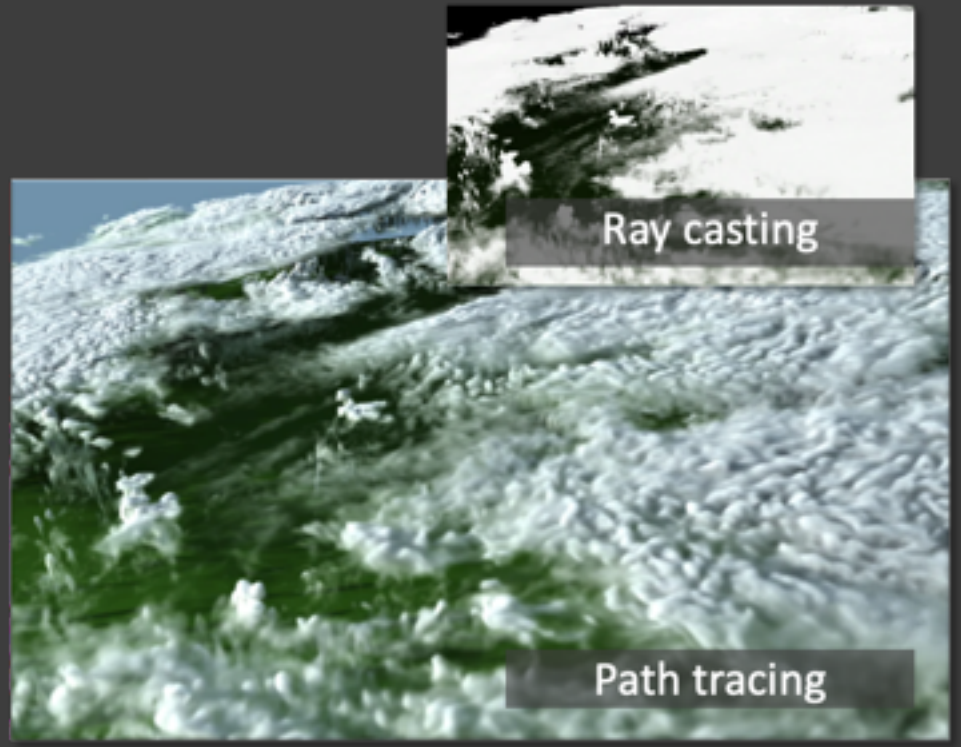
Carson Brownlee, Intel
Niklas Rober, DKRZ

Ray casting

Ray casting + shadows/AO

Path tracing





NASA AMES

Rendered on Pleiades
Vis wall cluster

dataset: parachute; simulation: Dr M. Barad, NASA Ames; visualization: Tim Sandstrom, NASA Ames

AMR - Cosmic Strings

Paul Shellard - Amelia Drew - Kacper Kornet
Carson Brownlee

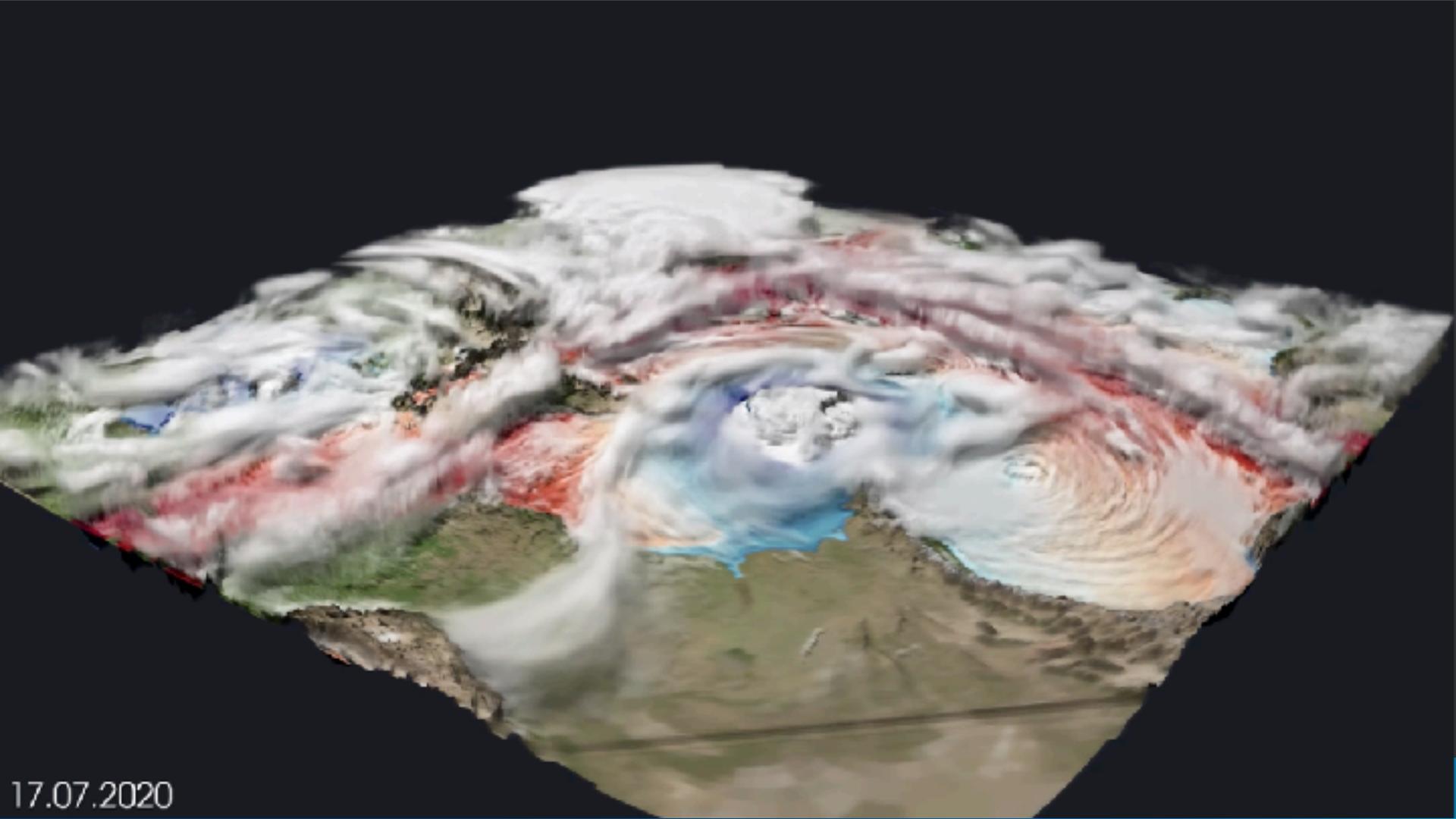
TACC, *Data courtesy Eastern Graphics

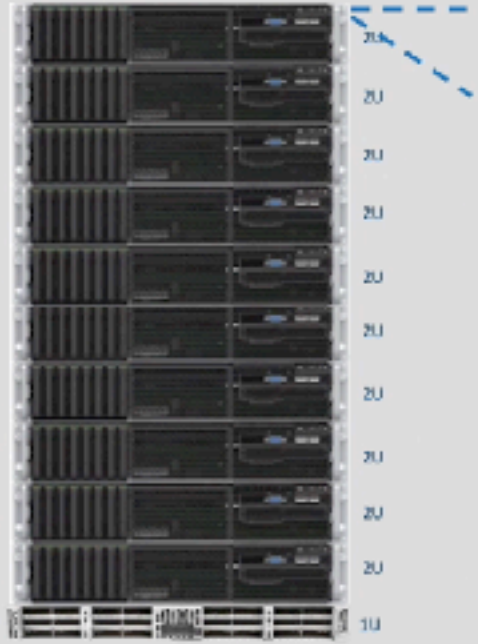
Distributed Data

```
// Determine the bounds of this rank's region in world space
const vec3f distribBrickDims = worldBounds.size() / vec3f(distribGrid);
box3f localRegion(distribBrickId * distribBrickDims + worldBounds.lower,
    distribBrickId * distribBrickDims + distribBrickDims
        + worldBounds.lower);

// Special case for the ospray test data: we might have geometry right at
// the region bounding box which will z-fight with the clipping region. If
// we have a region at the edge of the domain, apply some padding
for (int i = 0; i < 3; ++i) {
  if (localRegion.lower[i] == worldBounds.lower[i]) {
    localRegion.lower[i] -= 0.001;
  }
  if (localRegion.upper[i] == worldBounds.upper[i]) {
    localRegion.upper[i] += 0.001;
  }
}

// Set our region that represents the bounds of the local data we own on
// this rank
world.setParam("region", cpp::CopiedData(localRegion));
world.commit();
```


17.07.2020



56 CPU Cores - 112 Threads

Intel® Xeon® Platinum 8280L CPU; @ 2.70GHz

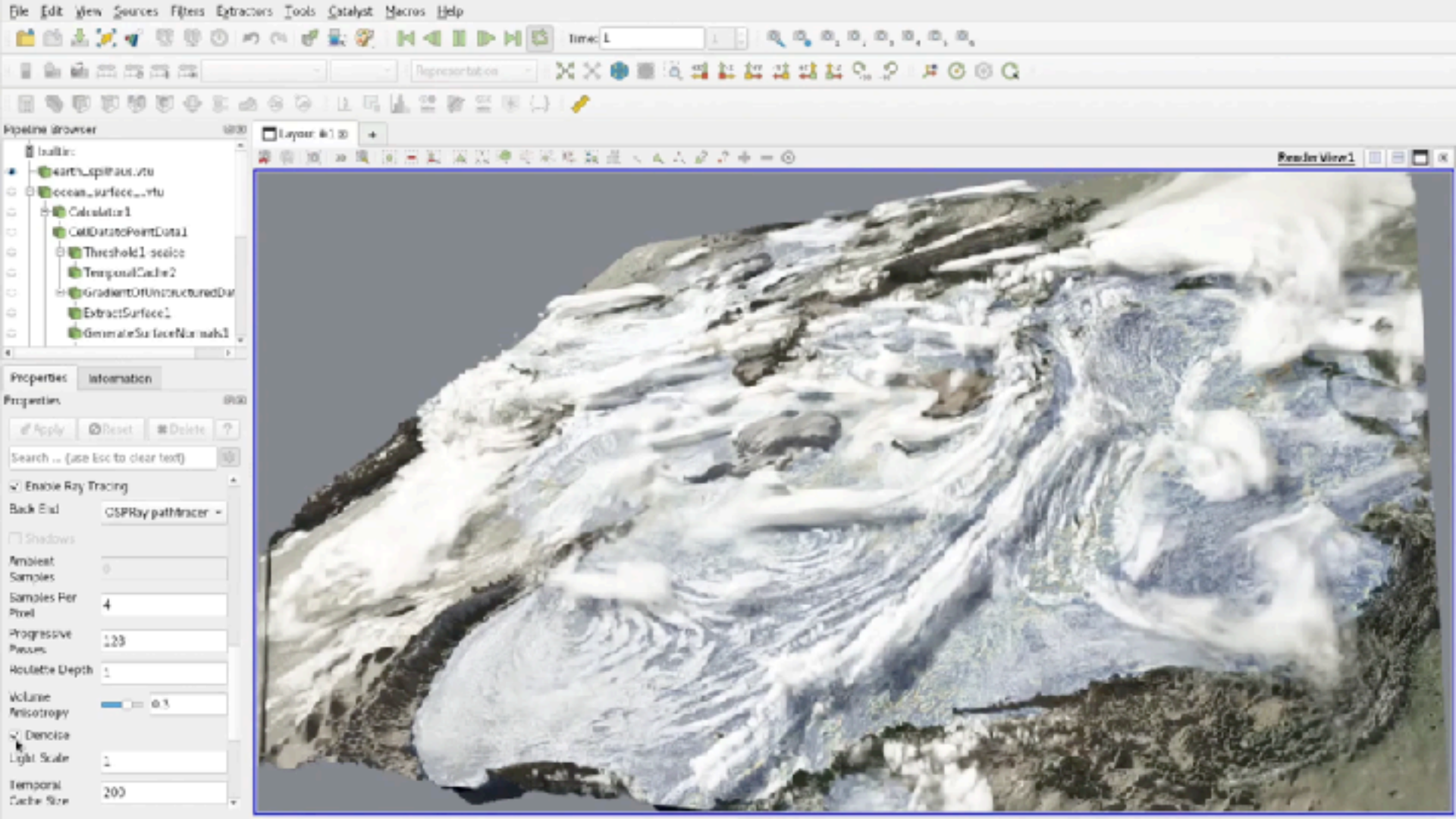
3 TB Memory Mode

Intel® Optane™ DC Persistent Memory

Intel® Omni-Path Edge Switch

Cluster Totals

- 3840GB DDR RAM (10 x 384GB)
- 30TB Optane™ DC NVDIMM (10 x 3TB)
- 20 Intel® Xeon 8280L CPUs
- 560 Cores – 1120 Threads
- Total size: 21U


File Edit View Sources Filters Extractors Tools Catalyst Macros Help
Pipeline Browser
Properties
Information
Enable Ray Tracing
Back End
OSPRay pathtracer
Shadows
Ambient Samples
Samples Per Pixel
Progressive Passes
Roulette Depth
Volume Anisotropy
Denoise
Light Scale
Temporal Cache Size

Default ParaView AMR Volume Rendering

AMR Volume is resampled into a regular grid

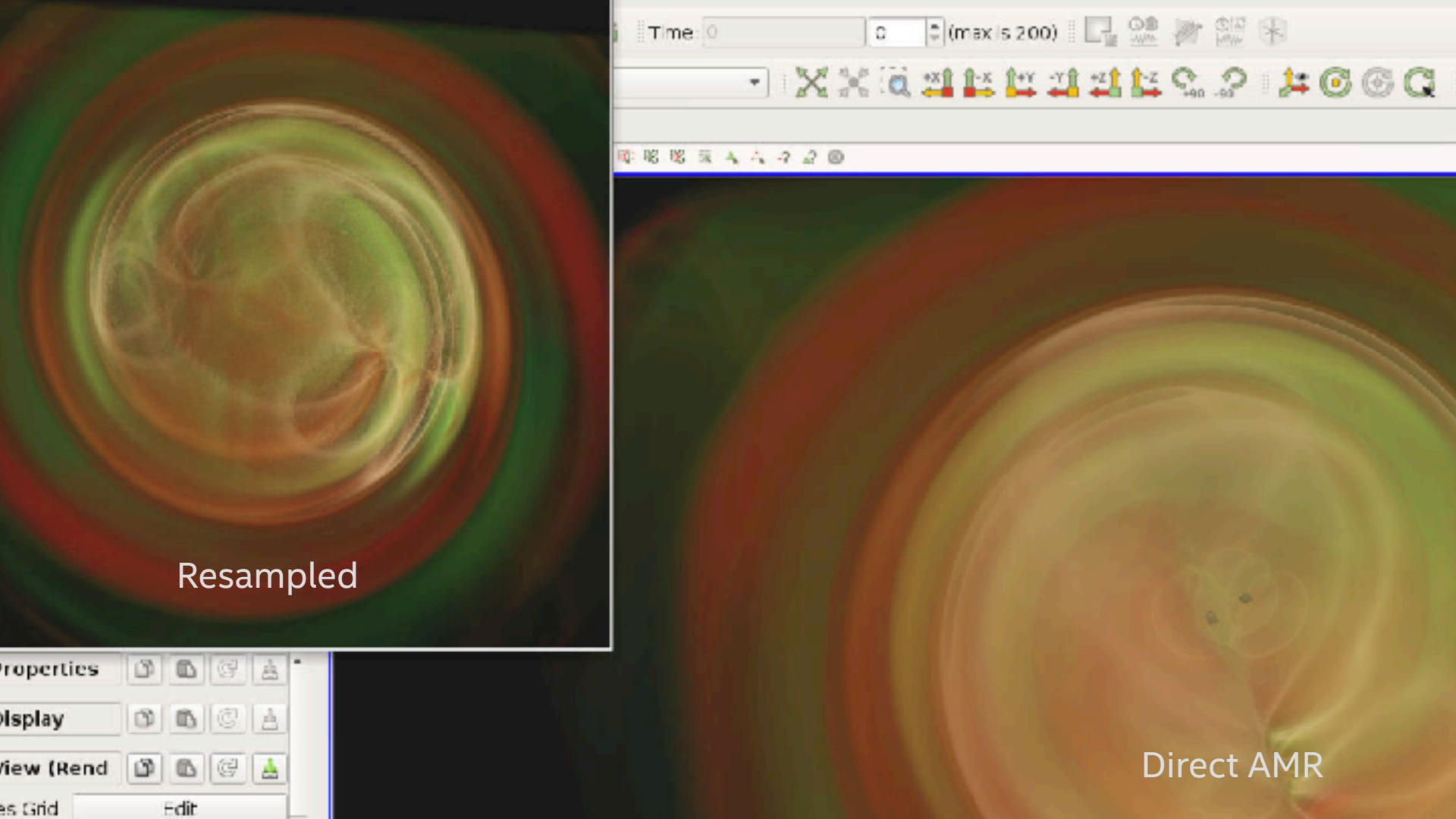
- Processing time
- Memory
- Grid artifacts

Direct AMR Volume Rendering with OSPRay

Berger-Colella overlapping AMR

Rays directly sample AMR grid

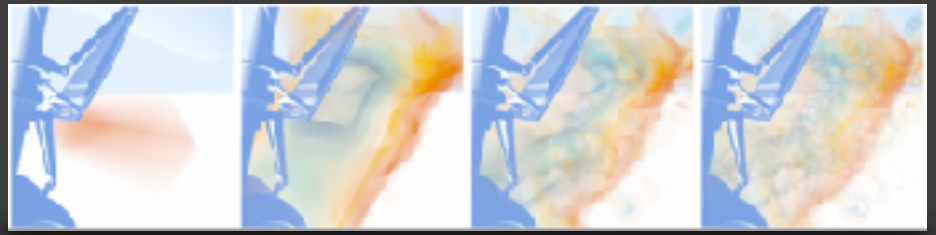
* “CPU Volume Rendering of Adaptive Mesh Refinement Data”, Wald and Brownlee ’17.


Time
(max is 200)
Resampled
Direct AMR

Performance

FPS shown at 1MP resolution.

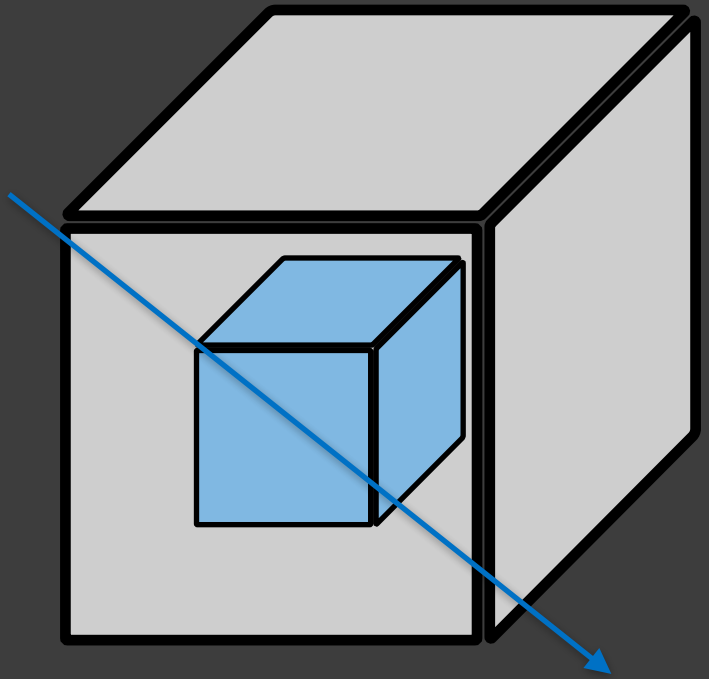
Older hardware - Two Intel Xeon E5-2699 v4 CPUs. 44 cores (88 threads) running at 2.2 GHz. 256 GB of RAM.

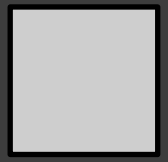


lvl1 - lvl4 refinement levels of landing gear

LandingGear 57GB	lvl0	lvl1	lvl2	lvl3	lvl4	knl4
Cells	4	8	52	560	2056	2056
Current	13.12	13.09	8.44	5.35	4.29	2.36
Basis	11.40	11.34	5.79	2.54	1.83	0.99
Blend	12.44	12.12	7.78	4.84	3.88	2.08
Finest	14.65	13.91	9.67	6.31	5.28	2.94
BHM 28GB	lvl0	lvl1	lvl2	lvl3	lvl4	knl4
Cells	4096	4098	4106	4114	-	4114
Current	25.00	24.96	25.26	25.58	-	14.52
Basis	29.57	28.59	28.52	27.49	-	15.86
Blend	24.94	25.64	25.36	24.52	-	14.32
Finest	30.48	29.17	26.36	26.25	-	14.93
Sphere 6GB	lvl0	lvl1	lvl2	lvl3	lvl4	knl4
Cells	1024	1034	1214	-	-	1214
Current	3.79	3.64	3.57	-	-	2.22
Basis	3.87	3.89	3.76	-	-	2.43
Blend	3.76	3.59	3.47	-	-	2.15
Finest	4.36	4.27	4.19	-	-	2.77
LLNL 8GB	lvl0	lvl1	lvl2	lvl3	lvl4	knl4
Cells	31k	56k	29M	-	-	29M
Current	4.89	4.31	3.36	-	-	1.833
Basis	5.51	4.35	2.76	-	-	1.53
Blend	4.77	4.07	3.06	-	-	1.68
Finest	5.74	5.72	4.65	-	-	2.61
DNS 7GB	lvl0	lvl1	lvl2	lvl3	lvl4	knl4
Cells	6144	399k	24M	-	-	24M
Current	44.51	22.55	5.67	-	-	2.88
Basis	50.71	14.08	2.75	-	-	1.12
Blend	45.13	22.31	5.52	-	-	2.78
Finest	52.81	24.71	6.53	-	-	3.49

The problem of Parallelism



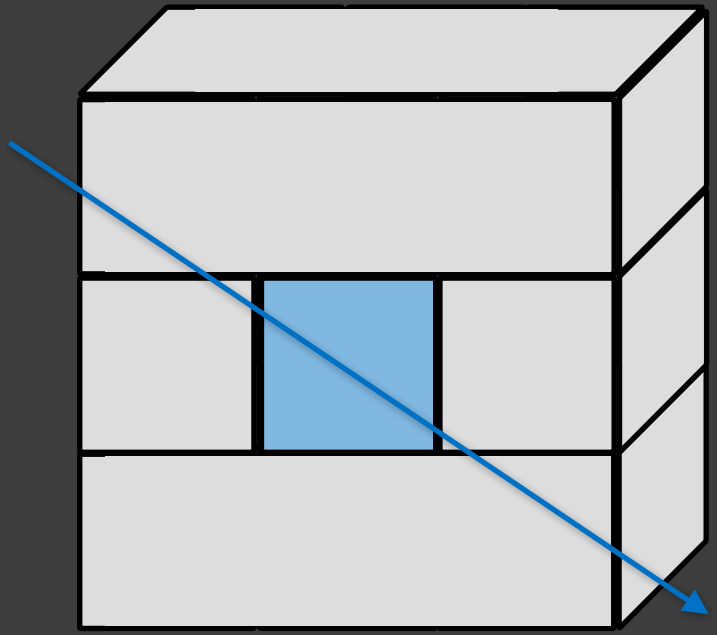


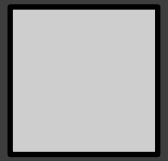
Resident Data

Remote Data

Wombat (WIP)

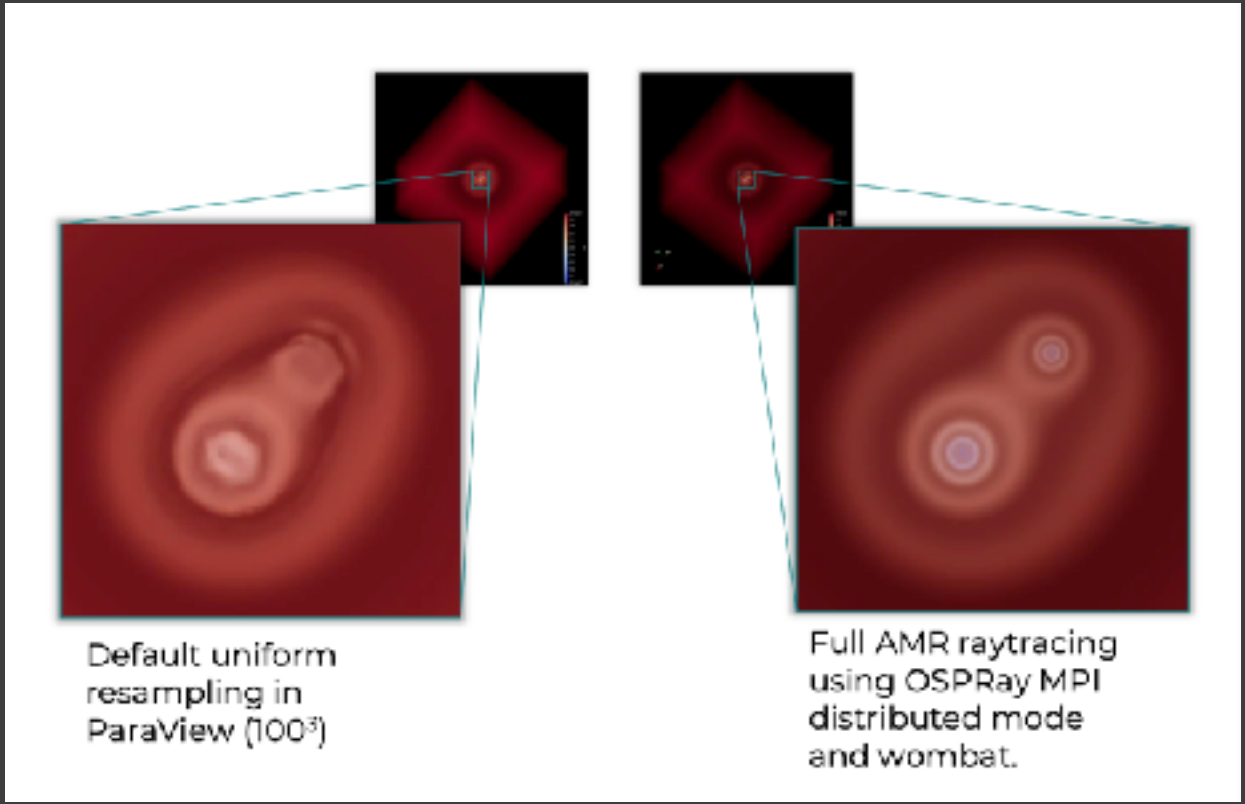
"Convexify" the regions





Resident Data

Remote Data


Default uniform resampling in ParaView (100^3)
Full AMR raytracing using OSPRay MPI distributed mode and wombat.

Miren Radia

Wombat

Pros

- Parallel AMR
- Only minor modifications to host application (ParaView)
  - * currently implemented in OSPRay
- Convexify operation can be done as independent middleware in the future for any application

Cons

- “Convexify” overhead
- Host Application needs to use OSPRay for rendering **and** compositing

Questions?