Distributing Liquids using OpenVDB

Dan Bailey Harry Biddle Nick Avramoussis Matthew Warner
Double Negative*

Distributed Simulation using the Dynamo FLIP Solver (colour indicates the process). Liquid simulations from Interstellar *(©2014 Warner Bros. Entertainment Inc. and Paramount Pictures Corporation) and* Exodus: Gods and Kings *(©2014 20th Century Fox)*

Abstract

From large, stormy oceans to cities being obliterated by tsunamis, demands on liquid simulations in Visual Effects are ever-growing in terms of complexity and scale. Improved performance through multi-threading alone is no longer proving sufficient. We present a light framework built on OpenVDB and OpenMPI that efficiently distributes sparse volumetric and spatially-organised point data. This greatly improves performance and increases our data sizes in production.

1 Introduction

Driven by the need for bigger and faster simulations, we have introduced a distributed simulation framework to our in-house FLIP fluid solver *Dynamo*. Features of our framework include:

- Spatially-organised points coupled to an OpenVDB grid [Bailey et al. 2014].
- Communication of sparse OpenVDB grid topologies between processes.
- Adaptive load-balancing of an evolving point set.
- Identical results regardless of how the data is organised across the machines.

2 Our Distribution Framework

In Dynamo a VDB grid stores which process owns each voxel at any point in time, offering a flexible mechanism through which to load-balance the processes. The sparsity of OpenVDB requires extra computation to determine where voxels should be sent, yet even in the pressure projection we are able to hide network communication behind computation [Höfler et al. 2007]. By coupling points to voxels we are able to share point data between processes as easily and efficiently as volumetric data. A cross-process point sorting algorithm mantains the spatial organisation of the points as the simulation evolves. Our framework targets FLIP liquid simulations, but is built to be extensible to any OpenVDB-based simulation.

*e-mail:{drb,hb,nna,mw}@dneg.com

SIGGRAPH 2015 Talks, August 09 – 13, 2015, Los Angeles, CA.
ACM 978-1-4503-3636-9/15/08.
http://dx.doi.org/10.1145/2775280.2792544

3 Deterministic Simulations

FX artists demand repeatable, deterministic results as they refine a simulation. Our framework provides identical results regardless of the data organisation across machines. To achieve this, order-dependent operations such as global summations are done by a reduction of each leaf of the OpenVDB grid prior to the root process performing an ordered reduction of the resulting data.

4 Results

Sim Machines	1	2	3	4	5
Av Frame Time	2200s	1427s	1082s	856s	741s
Perf Increase	1.00x	1.54x	2.03x	2.57x	2.97x
Mem Per Proc	80GB	42GB	31GB	25GB	15GB

We demonstrate the scaling and memory usage of Dynamo across increasing numbers of machines with a tank-style city tsunami scene. 1.5 billion FLIP points, memory usage is peak.

5 Conclusion

Our framework efficiently and determinstically distributes FLIP liquid simulations using two well-known open-source libraries. Spatially-organising points and adaptive load-balancing enables us to achieve scalable distribution across multiple machines. Distributing our Dynamo fluid solver has enabled us to greatly improve the efficiency and turn-around time of our FX artists in production.

References

BAILEY, D., WARNER, M., AND BIDDLE, H. 2014. Packing the water pipe. In *ACM SIGGRAPH 2014 Talks*, ACM, New York, NY, USA, SIGGRAPH '14, 10:1–10:1.

HÖFLER, T., GOTTSCHLING, P., LUMSDAINE, A., AND REHM, W. 2007. Optimizing a conjugate gradient solver with non-blocking collective operations. *Elsevier Journal of Parallel Computing (PARCO) 33*, 9 (Sep.), 624–633.