Real-Time Physics for Cloud-Based Video Game Delivery

Alex Brown

Newcastle University

22 March 2018

Outline

Overview

- Prototype is a Distributed Virtual Environments (DVE)
- Spatially partitioned DVE have attracted a lot of commercial attention recently.[1]
- It's middleware with applications including:
  - Games[2,3]
  - Multi-agent AI
  - City planning
  - VR

---

[1] *SpatialOS*.
[2] *Worlds Adrift - Bossa Studios*.
[3] *Vanishing Stars: Colony Wars by NINPO*.

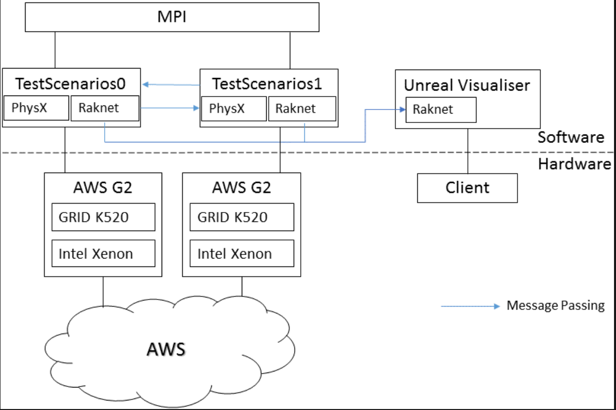
Overview

Using spatially partitioned DVE removes the limitation of the computational power of a single machine

This allows for greater scalability in both the number of users[4] and complexity and size of the Virtual Environment.

[4] Hori et al., ‘Scalability issues of dynamic space management for multiple-server networked virtual environments’.

Overview


MPI
TestScenarios0
PhysX
Raknet
TestScenarios1
PhysX
Raknet
Unreal Visualiser
Raknet
Software
Hardware
AWS G2
GRID K520
Intel Xenon
AWS G2
GRID K520
Intel Xenon
Client
Message Passing
AWS

Overview

Video

Background

- Previous DVE work has focused entirely on players (e.g.[5,6,7,8])

---

[5] Dong and Yue-Long, 'An Overlapping Architecture for ROIA in Cloud'.

[6] Hori et al., 'Scalability issues of dynamic space management for multiple-server networked virtual environments'.

[7] Senna Carneiro and Arabe, 'Load balancing for distributed virtual reality systems'.

[8] Min et al., 'A load balancing algorithm for a distributed multimedia game server architecture'.

Our Contribution

Despite DVE being a well research area, there has been no published research on the use of real-time physics in a DVE. This paper intends to define the problems of using real-time physics in a DVE and propose a technique to solve these problems.

We propose new techniques for tackling the core problems of physics in DVE. The first proposal is Aura Projection, for dealing with time-space consistency issues.

What is Physics?

- Real-time Physics is concerned with:
  - Simulating objects in a physically realistic manner
  - Querying the physical state of the simulation
- Computationally Intensive
- Unrealistic results can easily occur at extremes

Problem Definition

- A naive approach to objects traversing region boundaries causes time-space inconsistencies
- It is important to note that real-time physics engines are non-deterministic[9]

[9] *PhysX Knowledge Base/FAQ — NVIDIA UK.*

Problem Definition

- In Spatially Partitioned DVE a server is only aware of the state of objects owned by other servers through message passing
- Communication delays therefore lead to time-space consistency issues and can result in stability problems

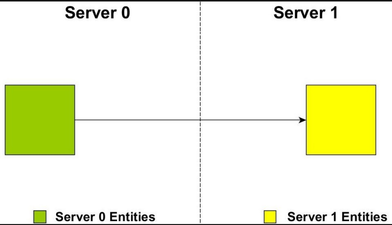
Our Contribution

The core problems of developing real-time physics for a spatially partitioned DVE can be broken down into the following:

- Minimising time-space consistency issues of physical entities across spatial division boundaries, to prevent unstable results
- Minimising network use, to minimise the effect of communication delays and avoid bandwidth bottlenecks
- Distributing the workload across servers in real-time, to avoid performance bottlenecks and maintain the advantage of partitioning the DVE

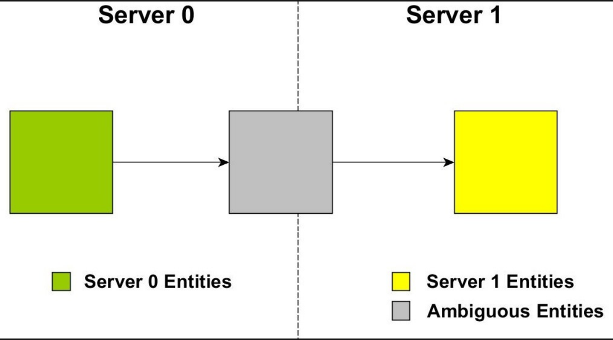
Simple Case

- Two servers run asynchronously
  - 3D simulations, divided into 2 regions
- Asynchronous messages are passed between servers



Simple Case - breakdown

- What happens when the object is on the boundary?
- Which server has ownership when it intersects the boundary?

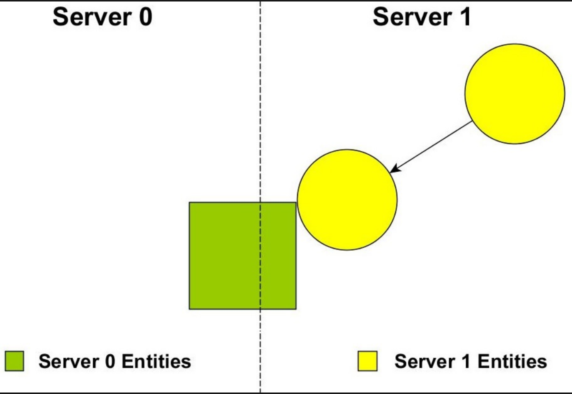


Simple Case - breakdown

- Both servers could simulate object
  - Consensus would need to be reached and the two simulations run synchronously
  - Twice the processing power required
  - Problems with collisions (simulations have non-deterministic collisions for stability reasons)

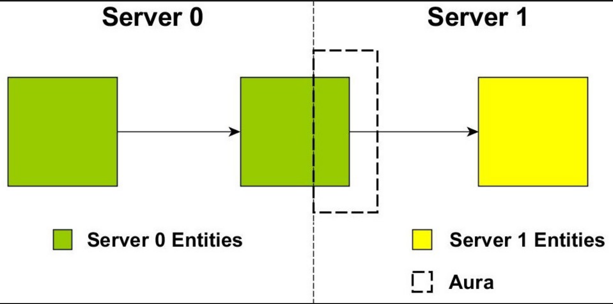
Consistency issues at boundary

What about when objects are interacting over boundaries (N.B. objects interact with objects only on one server)?



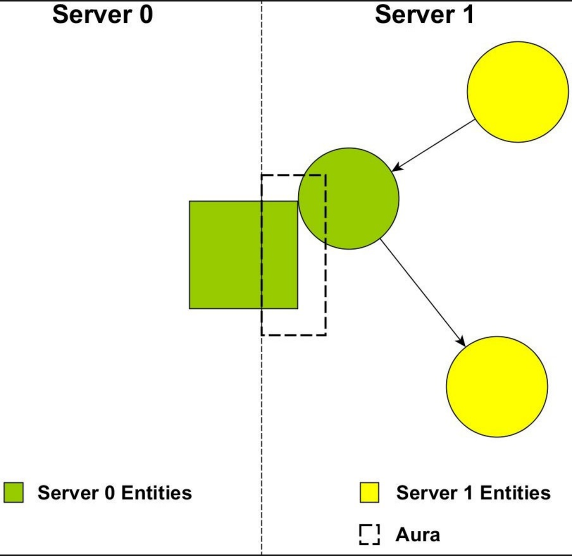
Aura Projection

Boundary between servers gets extended to include bounding area around boundary objects. This is an area known as an aura and can be thought of as being projected onto the other server



Aura Projection

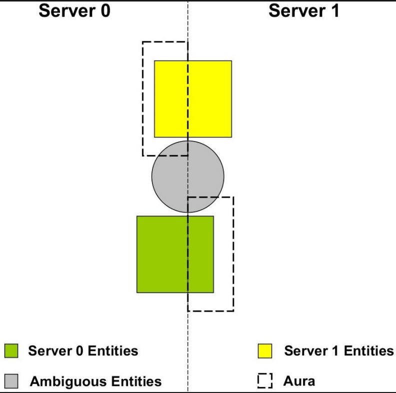
Objects colliding with projected aura get migrated



Challenges with Aura Projection

Thrashing

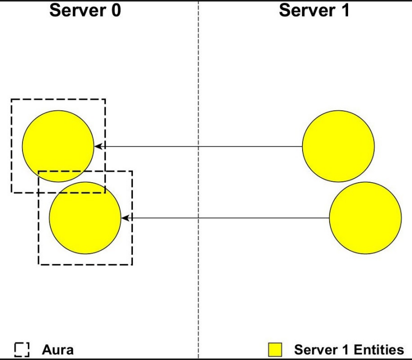
- Object is between two boundary objects on either server, so neither server is correct for the object to be on
- Solution:
  Recursive search through objects in bounds and send whole group
  $\mathcal{O}(n \cdot m)))$



Challenges with Aura Projection

Islands

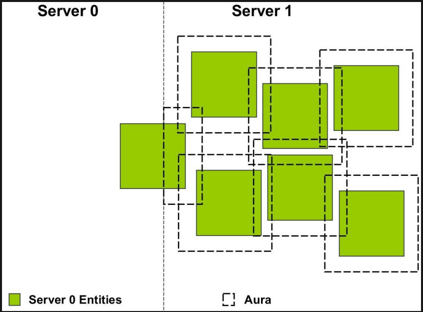
- Two objects in bounding box of each other, both inside region of other simulation. Neither get transferred, as each one is interacting with another object from the original simulation.
- Solution:
  Recursive search through objects in bounds, whenever object leaves boundary simulation $\mathcal{O}(n \cdot m)))$



Challenges with Aura Projection

Sub-optimal load balance

- Many objects can end up on foreign server, causing network traffic overhead, when instead moving all objects to foreign server would be more optimal
- Solution:
  Dynamic boundaries



Results - Optimal Case

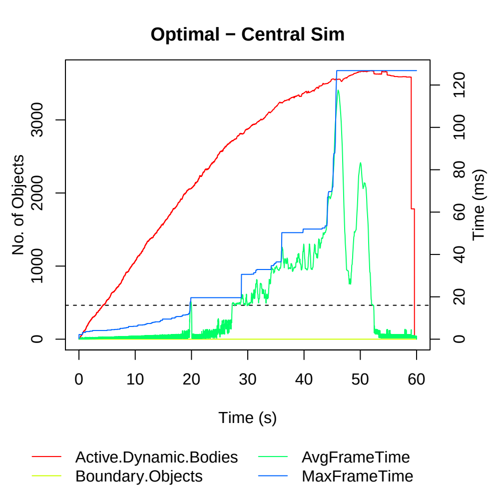


Figure: Performance of a centralised server in the optimal test scenario

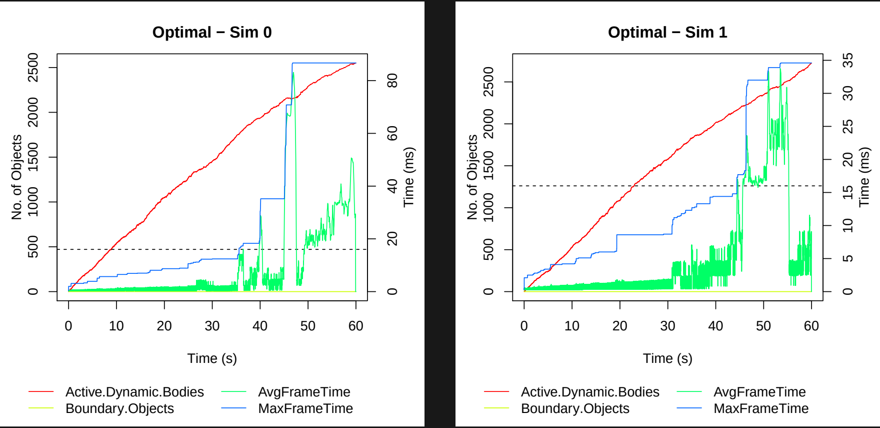


Figure: Performance of two servers in the optimal test scenario

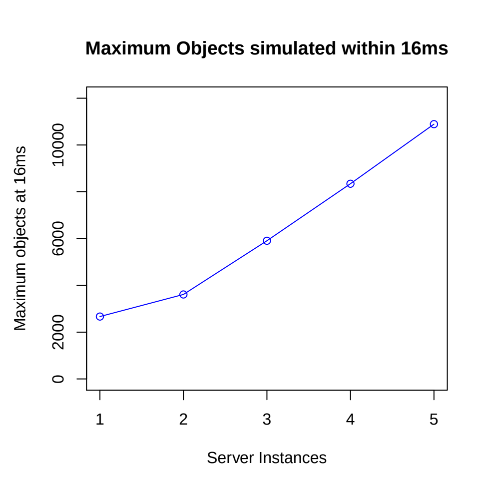


Figure: Scaling across N servers

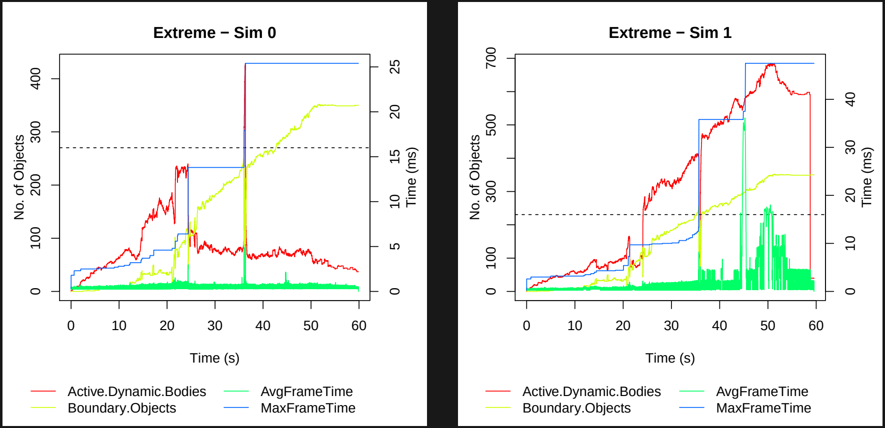


Figure: Performance of two servers in an extreme test scenario

Results - Extreme Case

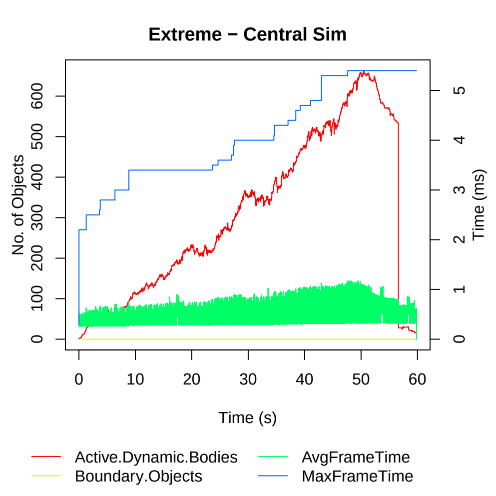


Figure: Performance of a centralised server in an extreme test scenario

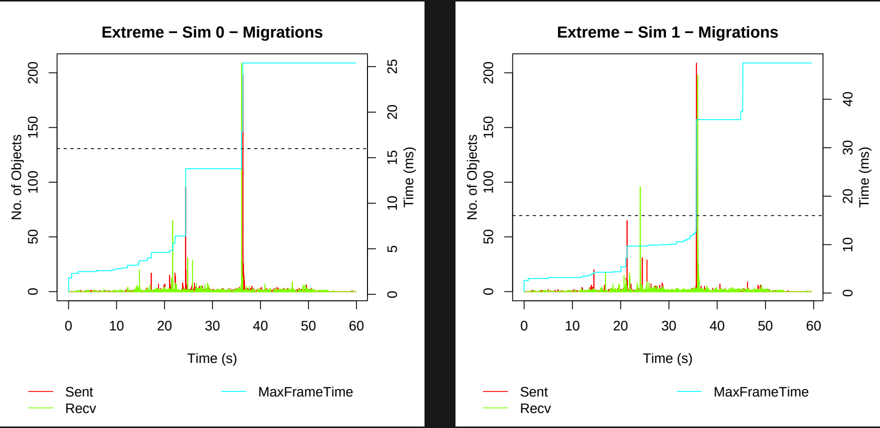


Figure: Performance of two servers in an extreme test scenario

Future

- Towards True Scalability:
  - Aura Projection will need to be extended to deal with corner cases
  - Run-time creation of nodes and regions
- Cross-Boundary Interaction:
  - Allow bodies being simulated on different servers to interact over the network
  - Comparison with Aura Projection
- Dynamic boundaries using quad/oct-tree
  - Routinely adapt boundaries to avoid objects interacting over boundaries
- Simulation Queries:
  - Using spatial division adds an extra level of complexity as queries will need to carried out across server boundaries

Technologies Used

Simulation:

- PhysX (Nvidia, free-source physics engine for games, GPU accelerated)
- MPI (Message Passing Interface)
- AWS (G2 Instances, GPU-enabled)

References I

Bezerra, Carlos Eduardo, Fábio R. Cecin and Cláudio F. R. Geyer. 'A3: A Novel Interest Management Algorithm for Distributed Simulations of MMOGs'. In: *Proceedings of the 2008 12th IEEE/ACM International Symposium on Distributed Simulation and Real-Time Applications*. DS-RT '08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 35–42. ISBN: 978-0-7695-3425-1. DOI: 10.1109/DS-RT.2008.11. URL: http://dx.doi.org/10.1109/DS-RT.2008.11.

Dong, L. and Z. Yue-Long. 'An Overlapping Architecture for ROIA in Cloud'. In: *2013 Fourth International Conference on Emerging Intelligent Data and Web Technologies*. Sept. 2013, pp. 61–65. DOI: 10.1109/EIDWT.2013.15.

Hori, M. et al. 'Scalability issues of dynamic space management for multiple-server networked virtual environments'. In: *2001 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (IEEE Cat. No.01CH37233)*. Vol. 1. 2001, 200–203 vol.1. DOI: 10.1109/PACRIM.2001.953557.

Lu, Fengyun, Simon Parkin and Graham Morgan. 'Load Balancing for Massively Multiplayer Online Games'. In: *Proceedings of 5th ACM SIGCOMM Workshop on Network and System Support for Games*. NetGames '06. Singapore: ACM, 2006. ISBN: 1-59593-589-4. DOI: 10.1145/1230040.1230064. URL: http://doi.acm.org/10.1145/1230040.1230064.

Min, Dugki et al. 'A load balancing algorithm for a distributed multimedia game server architecture'. In: *Proceedings IEEE International Conference on Multimedia Computing and Systems*. Vol. 2. 1999, 882–886 vol.2. DOI: 10.1109/MMCS.1999.778604.

PhysX Knowledge Base/FAQ — NVIDIA UK. (Accessed on 2017-04-20). URL: http://nvidia.co.uk/object/physx_knowledge_base.html.

Senna Carneiro, T. G. de and J. N. Cotrim Arabe. 'Load balancing for distributed virtual reality systems'. In: *Computer Graphics, Image Processing, and Vision, 1998. Proceedings. SIBGRAPI '98. International Symposium on*. 1998, pp. 158–165. DOI: 10.1109/SIBGRA.1998.722746.

References II

SpatialOS. (Accessed on 14/04/2017). URL: https://spatialos.improbable.io/.

Vanishing Stars: Colony Wars by NINPO. http://www.vanishingstars.com/. (Accessed on 27/09/2017).

World of Warcraft. (Accessed on 28/12/2017). URL: https://worldofwarcraft.com/en-us/ (visited on 28/12/2017).

Worlds Adrift - Bossa Studios. https://www.worldsadrift.com/. (Accessed on 27/09/2017).

Summary

- The aim of the project is to develop techniques for real-time physics in DVE
- DVE currently have a lot of commercial interest yet existing academic research has not addressed the issue of real-time physics in DVE
- So far a prototype of real-time physics working across servers in a DVE has been developed with some preliminary results