CONTACT NAME
Andreas Schafer: andreas.schaefer@informatik.uni-erlangen.de

A Scalable Backend for True MMORPGs
Bringing Supercomputer Tech Back to Gaming

Andreas Schäfer, Konstantin Kronfeldner, Thomas Heller
Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

1. What and Why?

We propose a novel backend for Massive Multiplayer Online Role Playing Games (MMORPGs) which applies software originally designed for supercomputer simulations to alleviate common problems which occurr under high load conditions. The system distributes a game's model (rules, player data, and interactions) across a cluster of servers in a scalable fashion. By using LibGeoDecomp, we can even offload parts of the computations to GPUs if the cluster is equipped with CUDA capable devices.

2. What This Poster is Not About

The described system is not about using GPUs for rendering and streaming games remotely (e.g. Gaikai or OnLive). We do not work on a game engine in the sense of *id Tech 5* or *CryEngine*. Also, this poster is not about offloading physics computations to a local GPU (e.g. Havok Physics) or about online software deployment (e.g. Steam).

3. State of the Art

Today MMORPGs represent one of the most important gaming genres. To cope with the load of thousands of players most games place them on separate servers (called realms in *World of Warcraft* or worlds in *Command and Conquer: Tiberium Alliances*). Players are usually not able to interact with players on other servers, which is detrimental to the *massive multiplayer* exerience they aim to provide. They are thus not true massive multiplayer games. The reason for this situation is that a game server needs to run a simulation which models the game's rules and incorporates all concurrent player interactions. A single server can only be scaled up to a certain degree (vertical scaling). Depending on the game, a couple hundred or at most some thousand participants will generate more load than even a modern server can handle[1].

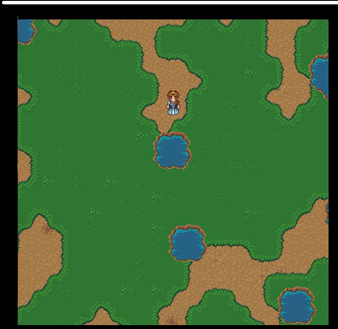
The consequences are disconnects and lag. Even *Second Life*, which explicitly aims at creating a single, alternative world, splits the game world into regions which are bound to distinct servers and are only loosely coupled[2]. This creates a noticable delay when a player moves from one server to another.

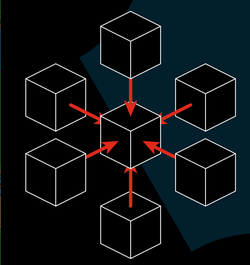
4. Our Architecture

Our idea is to model players and their interactions as a special type of n-body simulation. Unlike scientific simulations, which model the laws of physics, we base our simulation on the rules of a game.

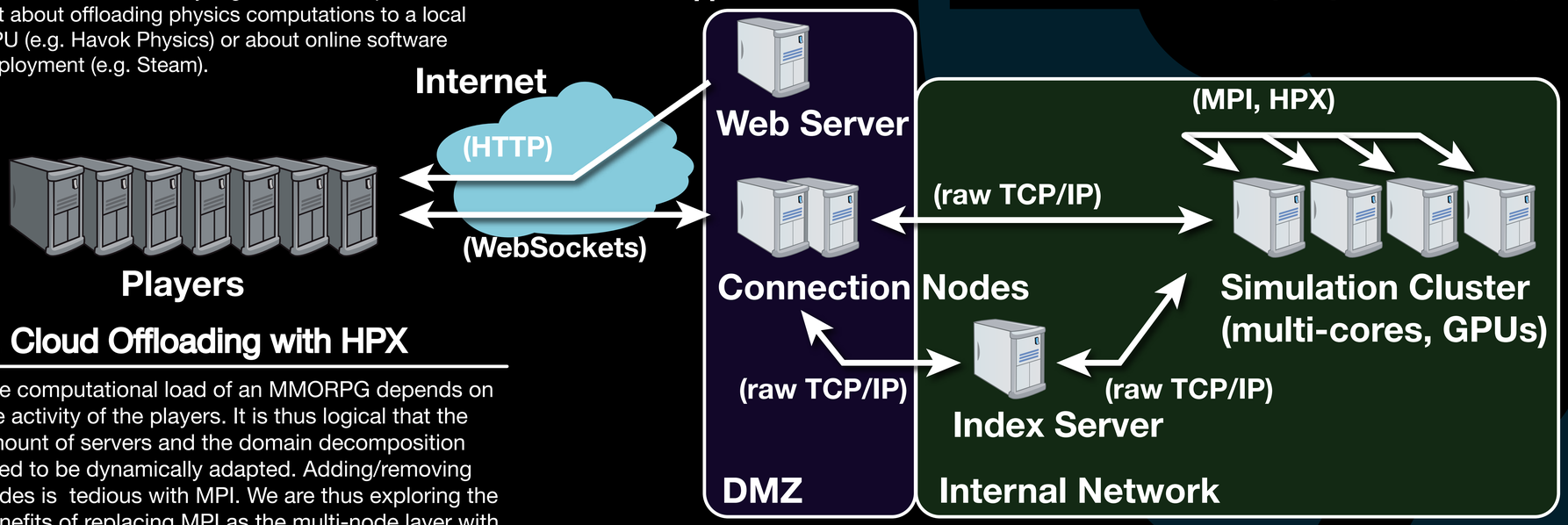
The simulation itself is implemented with LibGeoDecomp (Library for Geometric Decomposition codes[3]), an auto-parallelizing computer simulation library. LibGeoDecomp can harness a variety of HPC hardware. Using it together with live steering and *in situ* visualization for gaming is an unusual setup.

6. The Demo





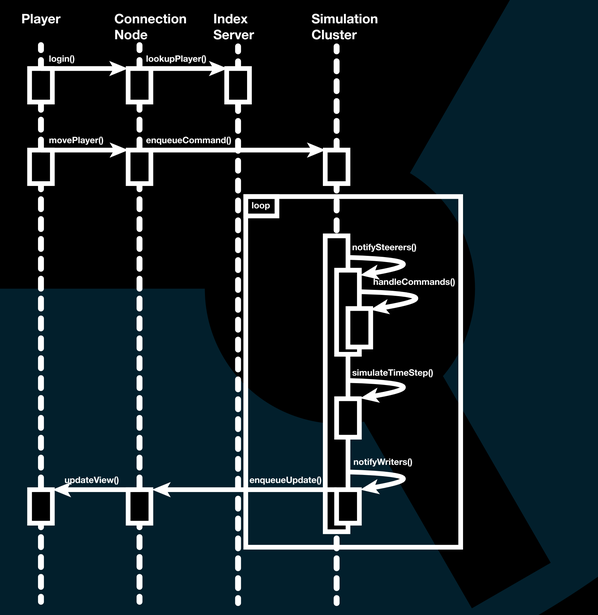
For testing purposes we are currently implementing a simple RPG based on the Liberated Pixel Cup[4] artwork; the framework itself remains flexible and may be used with other models, too.



5. Cloud Offloading with HPX

The computational load of an MMORPG depends on the activity of the players. It is thus logical that the amount of servers and the domain decomposition need to be dynamically adapted. Adding/removing nodes is tedious with MPI. We are thus exploring the benefits of replacing MPI as the multi-node layer with a more flexible HPX[5] (High Performance ParalleX) backend. Its Active Global Address Space (AGAS) approach can be used to transparently migrate load between nodes. We plan to test our engine in a compute cloud and dynamically scale the number of servers to match the varying numbers of players.

6. Interactions of Components



References

[1] Massive 2,800+ Player Battle in EVE Online Took Place Today. http://tinyurl.com/afbcesh
[2] Second Life. http://en.wikipedia.org/wiki/Second_Life#Server
[3] LibGeoDecomp. http://www.libgeodecomp.org/
[4] Liberated Pixel Cup. http://lpc.opengameart.org/
[5] HPX. http://stellar.cct.lsu.edu/

Friedrich-Alexander-Universität Erlangen-Nürnberg
Chair of Computer Science 3
Computer Architecture
Prof. Dr. Dietmar Fey
http://www3.cs.fau.de

Contact:
Andreas Schäfer
andreas.schaefer@cs.fau.de
tel: +49 9131 85 27910

LibGeoDecomp
Self-Adapting Stencil Codes for the Grid