



# **Designing Extensible and Scalable Virtual World Platforms**

Dan Lake  
Intel Labs

March 2010

# Grand Challenge: Scalability (Interaction)



50 Avatars



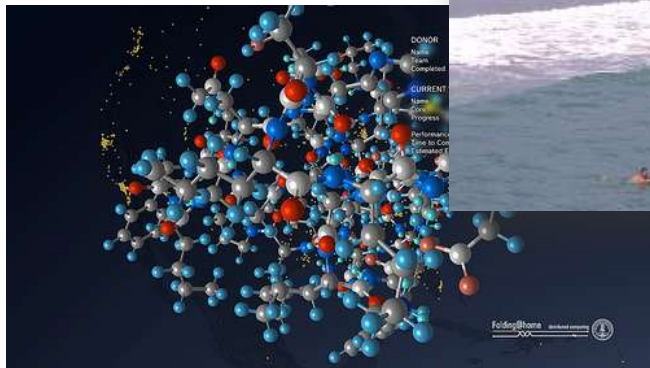
500 Avatars



50,000 Avatars

INTERACTION

# Grand Challenge: Scalability (Granularity)



$10^{-9}$  m



$10^3$  m



$10^{18}$  m

GRANULARITY

# Problem Formulation

- **Scene:** A data structure defining a space and the objects it contains
- **Operator:** a function that observes a scene and/or modifies the objects in a scene
  - Examples: Clients, scripts, physics, other external simulations

# Research Problem

- How to map a scene and its operators to the available compute and communication resources to yield an optimal experience for the maximum number of users?

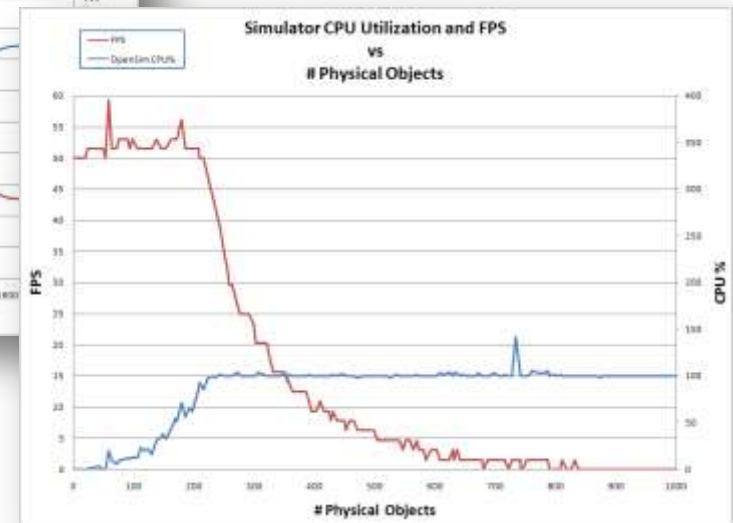
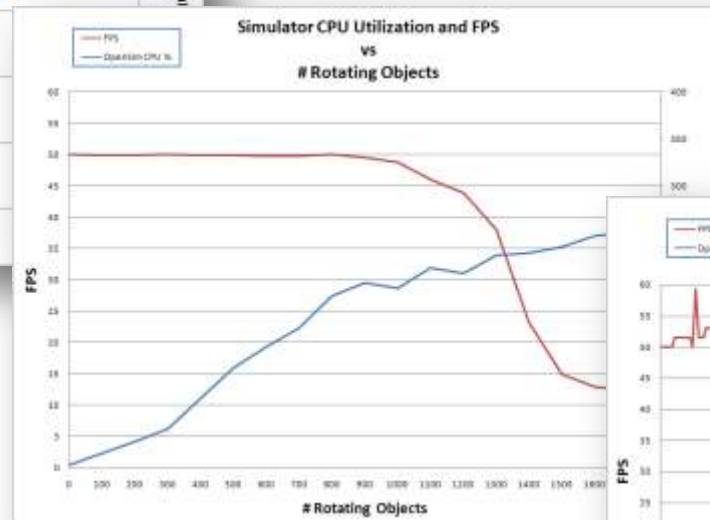
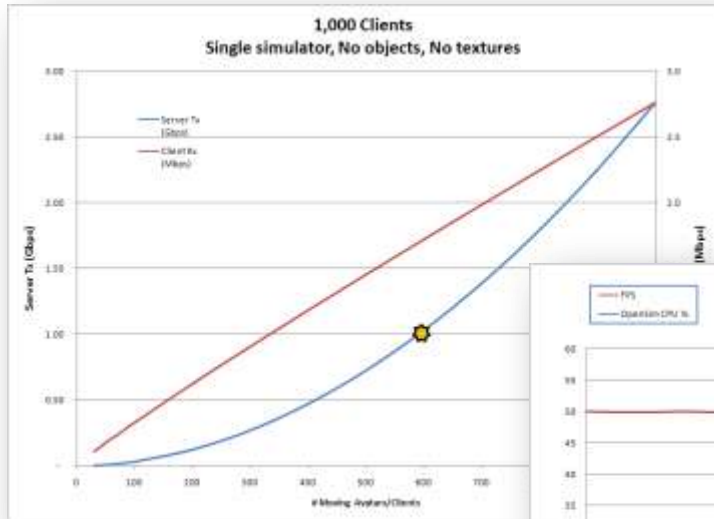
# Current OpenSim Architecture

- Scene and actors are integrated in simulator
  - Script, physics, client stacks and modules
- Heartbeat controls and synchronizes actors
  - Steps Physics
  - Updates objects
  - For each scene presence...
  - Handles client view, map, inventory, chat, etc

# Simulator Scaling Limits

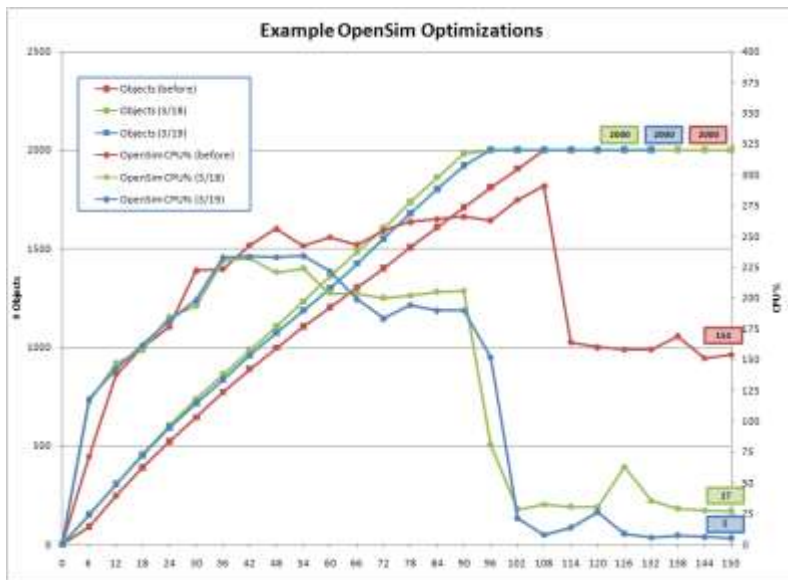
## Avatars, Scripted Updates, Physical objects

OpenSim r8311

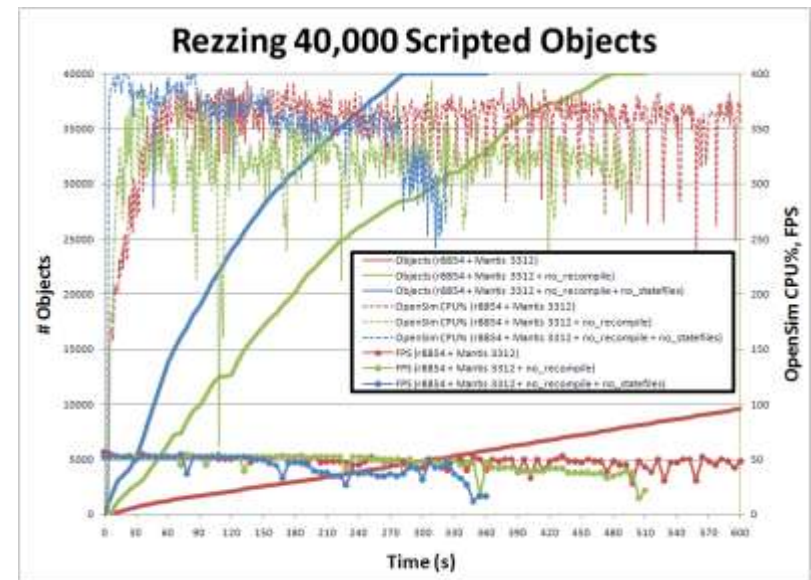


# Simulator Optimizations (2009)

- Increased prim count by 5X, avatar count by 4X
- Reduced startup time of large scenes by 95%, steady state CPU by over 90%, object creation time by 95%, and memory usage by 2/3

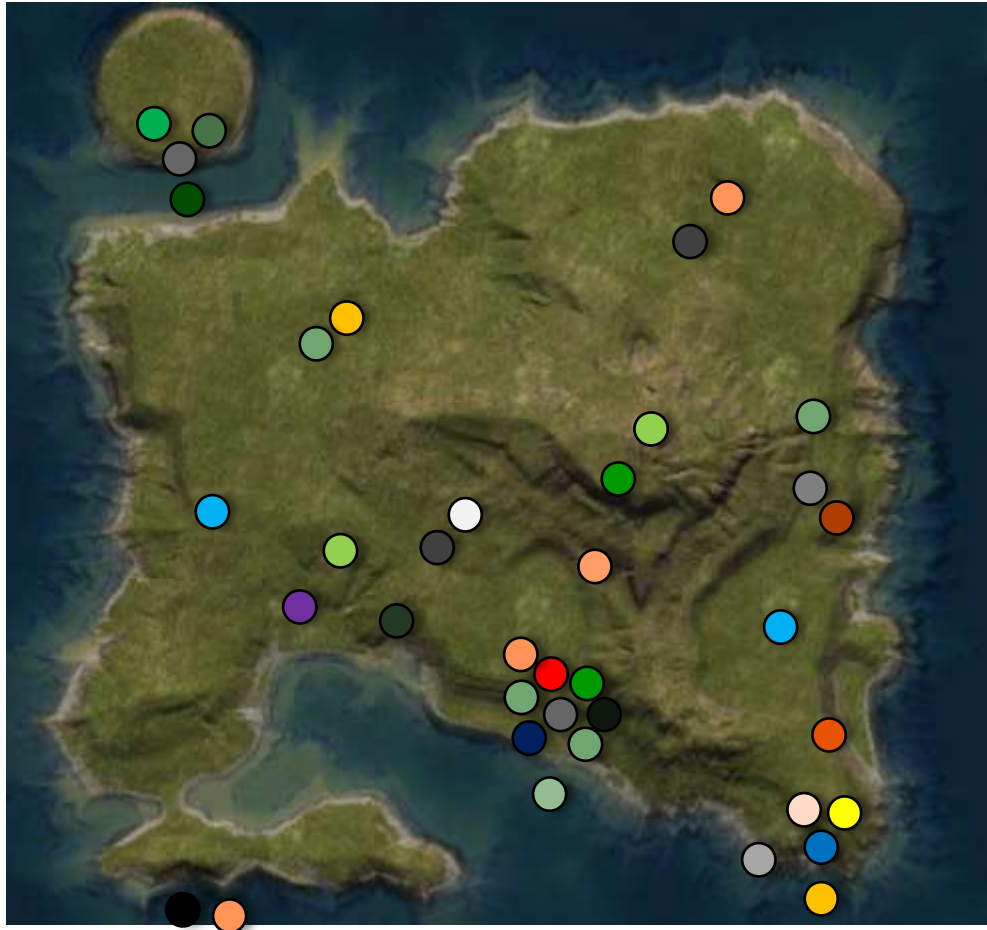


[http://sciencesim.com/wiki/doku.php/opensim/profiling\\_r8536](http://sciencesim.com/wiki/doku.php/opensim/profiling_r8536)



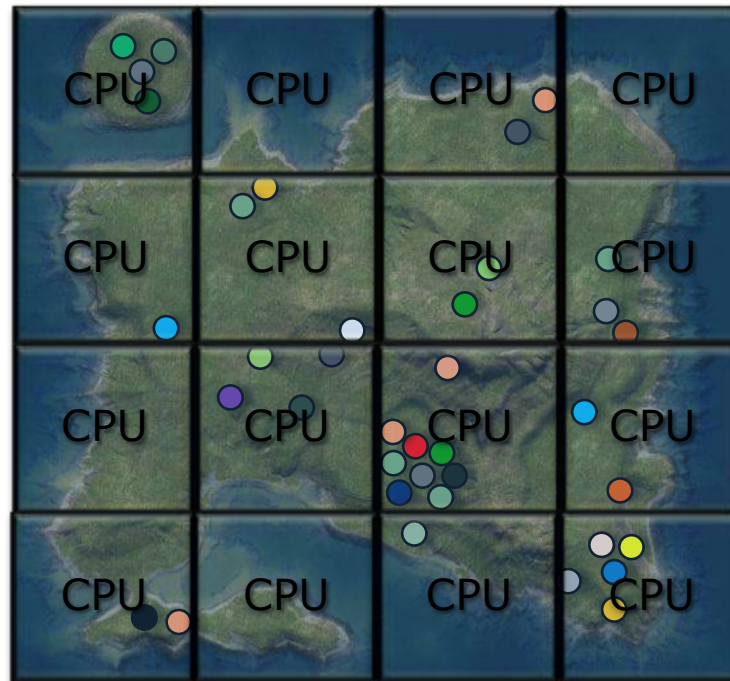
[http://sciencesim.com/wiki/doku.php/opensim/profiling\\_r8854](http://sciencesim.com/wiki/doku.php/opensim/profiling_r8854)

# Scaling the Simulation Architecture



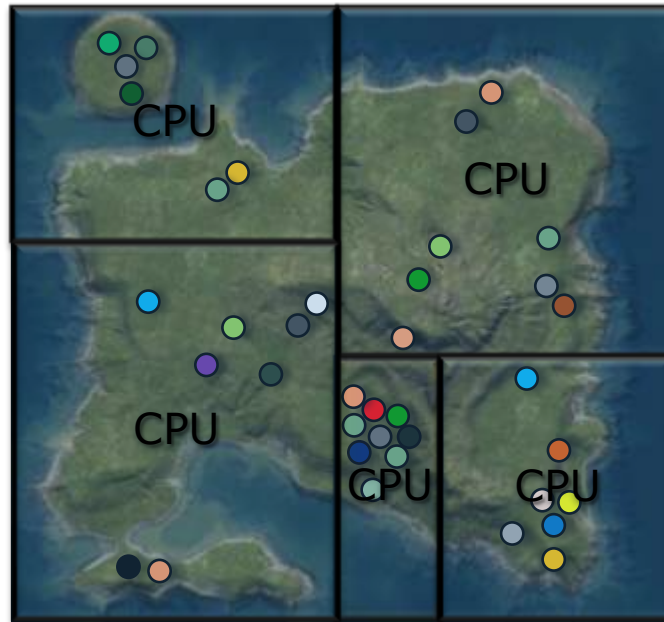
# Current Architecture: Static Partitioning

- Some regions are overloaded, others underutilized
- One actor may become bottleneck to performance
  - Too many clients, scripts, physics or other sim modules
- Interactions, complexity limited to server capacity



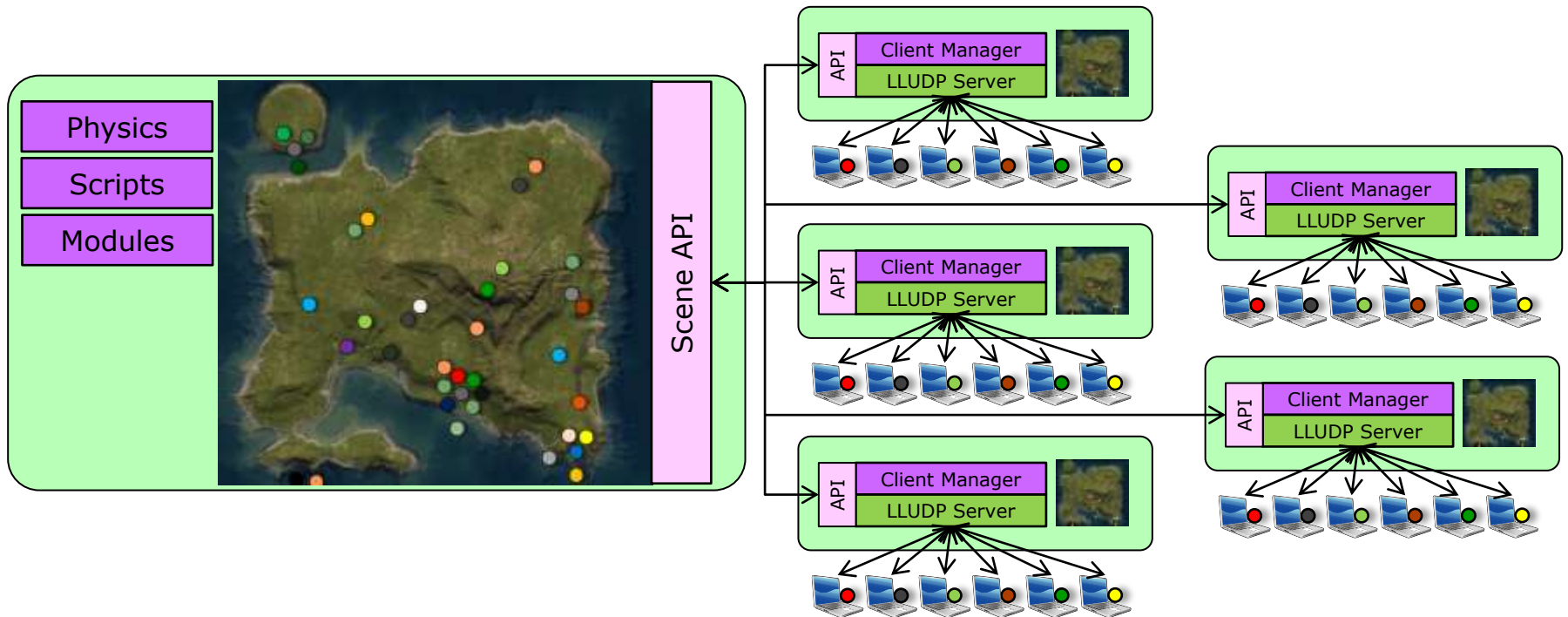
# Another Option: Dynamic Partitioning

- Distributes load across available servers
- More boundary crossings as partitions get smaller
- Still limited by single server simulator capacity
- We hypothesize: The actors limit scene scaling



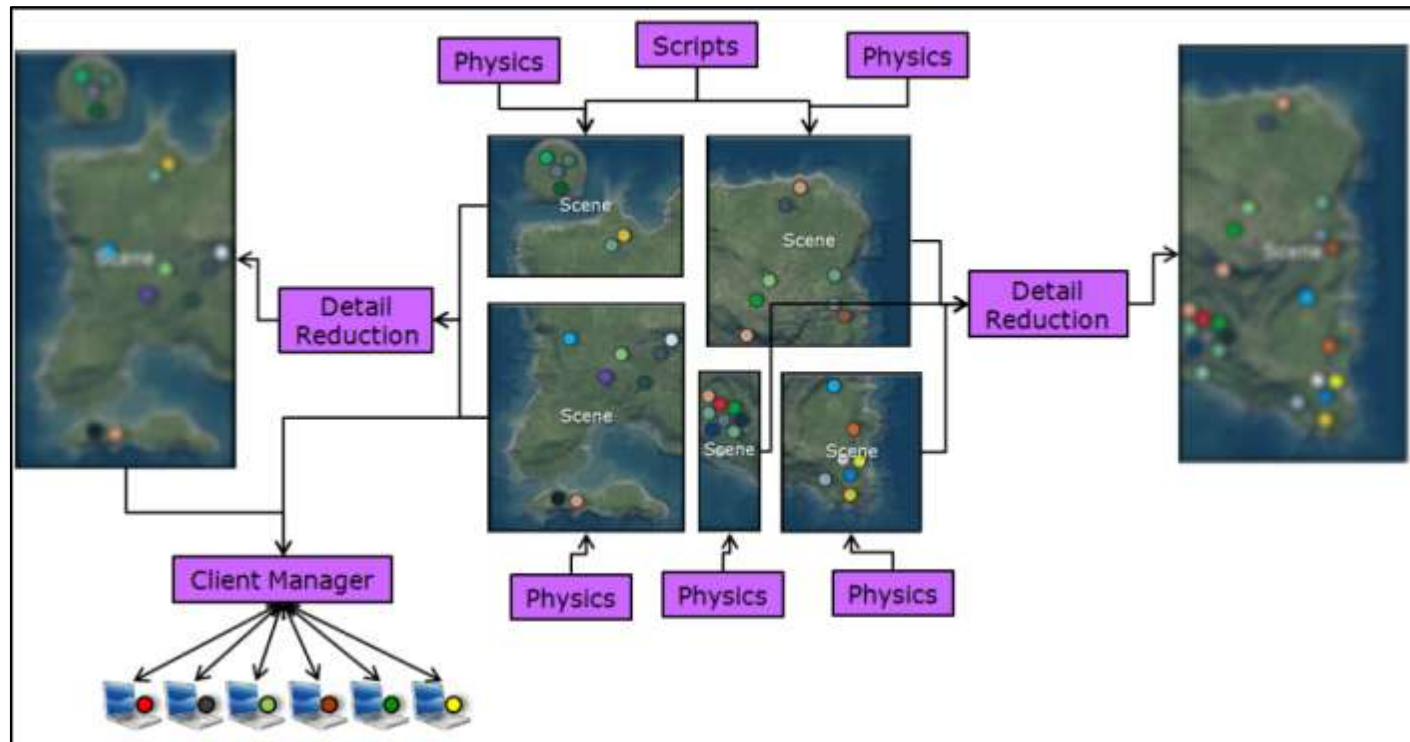
# Proof of Concept: Client Management

- Servers provisioned for high speed networking
- Prototyped 400 clients moving simultaneously
- Simulator serviced 4 connections instead of 400



# Proposal: Distributed scene service

- Scene becomes container with pub/sub service layer
- Physics and scripts run asynchronously on appropriate HW
- Client managers provisioned for communication
- Scene partitions replicated with reduced detail if needed



# Conclusions

- OpenSim is a capable platform for visual applications
- Large, complex environments being built today
- Truly massive scaling will enable new applications
- Requires a distributed architecture
  - Distributed scene service with common API
  - Actors mapped onto appropriate hardware
  - Eliminates synchronous scene control bottleneck
  - Reduces redundant computation of shared perspectives
  - Presents a larger, more interactive scene to clients

