Massively Multi-player Games and the Systems That Love Them

Mark Coway Wirt
Chief Technology Officer
Butterfly.net, Inc.

USENIX '05
Anaheim, CA
What is “Massive” anyway?
A Simple Definition of MMOGs

1000's of players simultaneously connecting
Sharing actions and state
Persistent worlds
MMOGs are not limited to...
  Specific Game Genres
  Deep/long term commitment
  3D immersive environments

MMOs are breaking out of their molds
The History of MMOGs

MIT Spacewar
Year: 1962
Platform: PDP-1 + CRT
Programmers: Steve Russell et. al.

Multiplayer games in pre-history
The History of MMOGs

Empire/Avatar
Year: Late 70's early 80's
Platform: Plato Computer system (CDC)

Multiplayer games in pre-history
The History of MMOGs

Multi-User Dungeon (MUDs)
Year: 1978 onward
Platform: DECSys-10
Programmers: Roy Trubshaw, Richard Bartle

Multiplayer games start to evolve to their present form.
The History of MMOGs

Meridian 59
Year: 1996
Platform: Intel/PC
Programmers: 3DO

First “Modern” MMO
The History of MMOGs

Ultima Online
Year: 1997
Platform: Intel/PC
Origin Systems

The MMO is truly born
What makes an MMOG Unique

MMOGs may have 100,000's of simultaneous players
Increasingly (de)-optimized for Broadband
Proliferation of Objects/Assets
Potentially Hostile Customers ("griefers")
Cheating
DOS attacks
Transforming the very fabric of the Internet into a high-performance video game machine

The Growth of the Market
Massively-multiplayer (MMP) Subscribers Doubled from 2002-2003
5 million at the end of 2001
10 million at the end of 2002

Subscription price-points and subscribers Q1 2003

MuOnline/Legend of Mir (Asia) 5,000,000 $ 8.00/mo
Lineage (S. Korea) 4,000,000 $15.00/mo
EverQuest 435,000 $12.95/mo
Star Wars: Galaxies 300,000 $12.95/mo
Ultima Online 250,000 $ 9.95/mo

Also: 25,000,000 gamers playing free “session-based” games

(sources: Mercer Consulting & Informedia Group, Company Sources)
The Growth of Games

Identify the Curve

Total MMOG Active Subscriptions (Excluding Lineage, Lineage II, and Ragnarok Online)

R² = 0.9865

(sources: MMOGCHART.COM, Sir Bruce)
The Size of On-line Economies

Some popular on-line MMOGs have economies (as determined by real-life financial transaction) larger than many countries.

Based on a study of Ebay auctions, in 2002 Edward Castronova characterized the economic size of Sony's *Everquest.*

Average player earns over $3.00US an hour.
Annual gross exports of more than $5,000,000US.
77th richest "country" in the world, with a per capita gross income greater than Russia's.
20th Century Topology
Figure 12
Cable Modem Deployment in the United States

Source: FCC cable system registrations and aeronautical frequency notifications; "Advanced Telecommunications in Rural America," NTIA, RUS.
The 21st Century Evolving Geography

(Source: www.websiteoptimization.com)
Transforming the very fabric of the Internet into a high-performance video game machine

Technical Challenges
Transforming the very fabric of the Internet into a high-performance video game machine

Technology

IRL, one builds scalable, reliable, distributed systems in a well defined environment

In MMOGs, one builds scalable, reliable, distributed systems in a dynamic, non-deterministic environment

Operating Platform

Artists

Programmers

Gateway Servers

Game Servers

Database Servers

Development Sandbox

Client Libraries

Capacity on Demand
Bits and Bytes: The Wire Protocol

Protocols supporting MMOGs must be well-chosen
TCP vs. UDP?
Low latency vs. Reliability (optional)?
Base on existing protocols? (ex. Datagram Congestion Control Protocol (DCCP), Stream Control Transmission Protocol (SCTP, RFC 2960))
Security (message hashing, encryption, anti-spoofing)?
Heterogeneous platform support?
Distributed ACID Transactions

Distributed Transactions utilizing cryptologically strong encapsulation, and allowing for remote authorization provide for conditional commitment “in game”.

Distributed Transactions utilizing cryptologically strong encapsulation, and allowing for remote authorization provide for conditional commitment “in game”.
Distributed State

State for millions of objects on thousands of clients must be distributed and synchronized

- Low latency requirements
  - As play styles move away from traditional role-playing games, latency requirements become even more challenging.

Calculation of occlusion, visibility

State as perceived on any client must be internally consistent and in consonance with other clients ("good enough").

Dynamic, stochastic, and capricious nature of play is, er, problematical

(Imagine doing CFD on a mesh which is controlled by thousands of users. At the same time. And they like to break things.)

State not only needs to be distributed; it must also be transformed.
People cheat, so all systems must be secure

Transport Layer: packets' veracity and integrity must be maintained. Spoofing must be prevented. Packet rewriting must be prevented.

Protocol Layer: Sensitive data must be protected through encryption or blind protocols – if it is clear-text, it will be compromised.

Resource Level: If your machines can be compromised, they will be. The player's machines will be compromised.

Service Level: Denial of Service attacks will be directed at your service and your users.

Software exploits. They exist. They will be found. Plan for it.
When Security Fails

Ultima OnLine (Wildly Popular MMP Role Playing Game)

On August 8th, 1977, Lord British (alter-ego of game designer Richard Garriot) was murdered. The weapon used was Fire Field. Murderer was banned from world. Early on, a server bug allowed the cloning of Gold on some Ultima Worlds. Hyper-inflationary economics resulted
Artificial Intelligence and Non-player Characters

Not only do you have 100,000 players, you have 100,000 automated Self-transforming Machine Elves, and they don't pay you any money.

- Level-of-detail AI in a distributed environment
- Highly efficient algorithms and implementations
- How to make stupid processes seem smart?
- State transmission between AI processes, in a distributed environment.
Advanced Clustering Techniques

In a dynamic environment, static clustering may not work.
Load balancing may necessitate ad-hoc clusters.
Functions and processes may need to be migrated in real-time.
The complexity of the content (and the domain) necessitate very complex interactions between physical resources (this isn't MPI!).
Resource management in a profoundly heterogeneous environment.
Game clusters can easily generate/receive Gigabits of data per second.
Malleable network topologies.
Game Instancing

Create games where 1,000s of players can have private experiences.

- User-driven spawning of game spaces and computational resources
- E.g. Every war party gets their own musty dungeon, complete with dragons and kobolds.
- Seamless transitions between shared and private spaces.
Content Abstraction

The current state of the art is "automatic legacy"

All games are custom crafted, with game content tied to infrastructure (but some companies, such as Butterfly.net, are addressing this)

How does one abstract functionality, engender generality, and create content-agnostic technology without loosing performance?

(Imagine trying to address the previous technical challenges with a general solution, for multiple game types/play styles.)

Game are difficult, expensive, and slow to develop. Until technology can be leveraged across titles, the industry will not flourish.
Episodic Content

The current state of the art are game snapshots with monolithic updates.

As games evolve, the industry is moving towards more episodic content.

Configuration management on a distributed system, with multiple concurrent versions.

- Distribution of content
- Compatibility with client software
- Non-deterministic paths of unfoldment
Transforming the very fabric of the Internet into a high-performance video game machine

**Testing: Development**

The distributed nature of the beast makes replicability “problematical at best.”

- 10's – 100's of servers, 100's – 1,000's of clients

Combinatorics of interactions daunting

- 10's of player classes, 100's of interaction, 1,000's of objects. Tested daily if possible

Many types of development tests needed

- Regression, both in unit and end-to-end
- Stress/Loading
- Live testers and scripted clients

Instrumentation of systems key to testing in non-deterministic/heterogeneous environment.
Testing: Beta

The scale of MMOs makes it well impossible to test at scale.

“This 7-day test will help Blizzard optimize World of Warcraft to deliver the best online-gaming experience possible when it launches later this year. Gamers in North America will have the opportunity to become one of over 100,000 people chosen to participate in the stress test and try out World of Warcraft before its release. “

--Blizzard Press Release, WoW

(In the first day of open play, over 200,000 players created accounts, and 100,000 played at the same time)
Testing: Live

A variety of testing must be performed with the live system. Playability cannot be entirely predicted beforehand.

What are players doing?
What are they *not* doing?

What anomalies are seen?
A Bug?
Design Flaw?
Cheating?

Can you predict what will break next?
Transforming the very fabric of the Internet into a high-performance video game machine

Testing: An Example from TSO
Instrumentation is Key for finding Bugs

Source: Larry Mellon, Executive Director of Development, Butterfly.net (formerly EA/Maxis)
Testing: An Example from SOL (Continued)
Instrumentation is Key

Source: Larry Mellon, Executive Director of Development, Butterfly.net (formerly EA/Maxis)
An Example from SOL (Continued)

Live testing

Total Inflationary Income (Money From Maxis)

Source: Larry Mellon, Executive Director of Development, Butterfly.net (formerly EA/Maxis)
Transforming the very fabric of the Internet into a high-performance video game machine

Challenges, Not completely Technical
The Fundamental Distinction between Games and so-called Real Life

In Games, the computer knows what is true, and the challenge is to convince the player he is wrong.

In Real Life, the universe knows what it true, and the challenge to convince the funding source that you're "right enough."
Transforming the very fabric of the Internet into a high-performance video game machine

The Fundamental Distinction (continued)

Real Life - Chemistry

Games - Alchemy
Economics

In Real Life, one models economies, usually to predict economic behavior.

In MMOGs, one creates real economies.

(Source of Figure: The In-game Economics of Ultima Online, Zachary Booth Simpson)
The Challenges of Economies

The availability of supplies and the controlling of the supply curve
Shadow economies
NPC's (Non-player characters) and other forms of slavery
Hording
Taxes and fees
Methods of exchange and transactions
  Transaction and contract support
  Auctions and Markets
Regulation vs. free-market economies
Lest you doubt the Reality of Game Economies...
In real life, one models some aspect or aspects of society to gain sociological insight.

In MMOGs, one creates societies.
The Challenges of Building Societies

Designing (and implementing) for emergent behavior
  IRL, the boundary conditions are well defined and well known
  In MMPs, discrete boundaries and limits degrade game-play
Punishment and Law (mechanized karma)
  Banning, Policing, Limiting, Incarceration, Conflict Resolution, etc.
Economies
Delineating Allowed behavior in an open-ended environment
  Many different play styles must be accommodated
  (Bartle's Killer, Socializer, Achiever, Explorer)
IRL, Bad things happen: People Starve; Economies Fail; People Die.

In MMPs, We Can Help Prevent This

"The Serious Games Initiative is focused on uses for games in exploring management and leadership challenges facing the public sector. Part of its overall charter is to help forge productive links between the electronic game industry and projects involving the use of games in education, training, health, and public policy."

-- http://www.seriousgames.org
Lastly...

Game are fun, and the challenges are interesting!
Transforming the very fabric of the Internet into a high-performance video game machine

Proof that it *is* fun: Disney's *Toontown*

"Making an MMORPG is hard! Anyone who has worked on one of these will tell you this. We had to develop several core competencies from scratch, rewrite our development software, work for years, and are even still revising our operations plan on a regular basis. The number of skill sets necessary to produce and run an MMORPG surpasses anything we had experienced before."

"Unfortunately, an MMORPG requires a persistent, relatively low-latency connection that is reliable and sustainable for up to hours at a time. This is extremely difficult to deliver using today's Internet."
Thank You

Mark Wirt, CTO
Mark.Wirt@Butterfly.net
Transforming the very fabric of the Internet into a high-performance video game machine

Q&A