Zero to Millions: Building an XLSP for Gears of War 2

Dan Schoenblum
Senior Engine Programmer
Epic Games
dan.schoenblum@epicgames.com
About Me

- Working in online gaming for over 10 years
- At GameSpy from 1999-2008
  - “Powered by GameSpy” technology
- Joined Epic Games early in 2008
  - Part of the online team
About Epic Games

- Gears of War Franchise
- Unreal Franchise
- Unreal Engine
  - 100+ games
Gears of War

- Released November 2006
- Multiple Game of the Year awards
- #1 Xbox LIVE game of 2006
- #2 Xbox LIVE game of 2007
Gears of War 2

- Build a better game
- More visually stunning
- More fun
Gears of War 2

- Build a better game
- More visually stunning
- More fun
- “Bigger, better, and more badass”
Gears of War 2

- Great online community
- New online functionality
- Better for gamers
- Better for Epic
- Build our own online backend
- Add features not supported by Xbox LIVE
Gears of War 2
Online Backend

- What features?
- How will it work?
- How will we build it?
- What technologies will we use?
Gears of War 2
Online Backend

- Starting from scratch
- Small team
  - 3-4 programmers
  - Also doing client-side work
  - Little backend experience
    - (aside from me)
- Lots of data to handle
- Less than a year
- Team at Microsoft working on backup
Unreal MCP
Unreal MCP

Xbox LIVE

Data Center

App Servers

Processing

Database Cluster

Gamers

gearsofwar.com

Epic
Unreal MCP

Xbox LIVE

Data Center

App Servers

Processing

XLSP Proxy

Load Balancer

Web Servers

Data Processing Servers

Database Cluster

Xbox 360
Gears 2

Web Browsers

Gearsofwar.com

Gamers

Epic

Internal Data Visualizers

OLAP Database

Post Processing Services

Epic Employees

SQL Database
Unreal MCP
XLSP

- Xbox LIVE Server Platform
- Used when adding custom online features to Xbox LIVE games
- Provides a secure and trusted channel of communication
Getting Started

- XLSP
- Game to backend data
- Backend to game data
Getting Started
Web Server

- Game’s interface to the backend
- HTTP is a simple protocol
  - Less work
  - Less risk
Getting Started
SQL Database

- Stores incoming data
- Stores outgoing data
Getting Started
Web Server & Database

- Mature
- Well-known
- Quick startup
Service Types

- **Bidirectional Service**
  - Game sends a request
  - Game receives data in response
  - Example: message of the day

- **Asynchronous Service**
  - Game sends data
  - Fire and forget
  - Example: game stats reports
Bidirectional Services
Example: MOTD
Bidirectional Services
Example: MOTD

Game sends request to the web server
Bidirectional Services
Example: MOTD

- Game sends request to the web server
- MOTD is pulled from the database
Bidirectional Services
Example: MOTD

- Game sends request to the web server
- MOTD is pulled from the database
Bidirectional Services

Example: MOTD

- Game sends request to the web server
- MOTD is pulled from the database
- Game receives MOTD via HTTP response
Asynchronous Services
Example: Game Stats
Asynchronous Services
Example: Game Stats

Game uploads a stats report
Asynchronous Services
Example: Game Stats

- Game uploads a stats report
- Web handler puts it in a queue
Asynchronous Services
Example: Game Stats

- Game uploads a stats report
- Web handler puts it in a queue
- Processing service pulls it
Asynchronous Services
Example: Game Stats

- Game uploads a stats report
- Web handler puts it in a queue
- Processing service pulls it
- Game info is stored in the database
Asynchronous Services

Example: Game Stats

- The data is replicated to two other databases
  - Epic for data analysis
  - www.gearsofwar.com
www.gearsofwar.com

![Gears of War 2 stats screenshot](http://gearsofwar.xbox.com/WebStats/default.htm?c=1033)
Data Analysis Internal

- Internal uses
  - Website
  - Custom reports

- Visualizations
  - Charts
  - Graphs
  - Heatmaps
  - Numbers
Data Analysis Internal

Default Weapon by Experience Level
Data Analysis
Internal

Weapon Kill Trends
Data Analysis
Internal

Game Type Trends
Data Analysis
Internal

🔥 Shotgun Kills on Day One
Data Analysis
Post-Processing

- Asynchronous services do minimal processing in production
- For further analysis, more processing is needed
- Custom post-processing apps dig further into the data
- Use the Epic replicated database
- No direct effect on production backend
Data Analysis
Post-Processing
Data Analysis Post-Processing

Replicated data arrives
Data Analysis
Post-Processing

- Replicated data arrives
- Apps post-process the data
Data Analysis
Post-Processing

- Replicated data arrives
- Apps post-process the data
- Store details back in the database
Data Analysis
Post-Processing

- Replicated data arrives
- Apps post-process the data
- Store details back in the database
- Post-processed data used for analysis
Data Analysis

SQL Trouble

- Initially used SQL for analysis
- Trouble after Gears 2 release
- Queries were very slow
- Huge table of weapon data
- SQL-based analysis was impractical
- Internal website was unusable
Data Analysis
SQL Trouble

- Initially used SQL for analysis
- Trouble after Gears 2 release
- Queries were very slow
- Huge table of weapon data
- SQL-based analysis was impractical
- Internal website was unusable
Data Analysis

OLAP to the Rescue

SQL
- Relational database
- Great for storage
- Bad for analysis

OLAP (OnLine Analytical Processing)
- Complements SQL
- Aggregates data in “cubes”
- Great for analysis
Data Analysis
OLAP to the Rescue

- SQL
  - Relational database
  - Great for storage
  - Bad for analysis

- OLAP (OnLine Analytical Processing)
  - Complements SQL
  - Aggregates data in “cubes”
  - Great for analysis
Scalability & Performance

- Estimates
  - 100s of transactions per second
  - Gigabytes of data per day
Scalability & Performance

- Game clients
- First line of defense
Scalability & Performance

- Load balancer
- Second line of defense
Scalability & Performance
Application Servers

- Horizontally scalable machines
- Each application server has:
  - Web server
  - For each Asynchronous Service
    - One queue
    - One processing service
- Self-contained
- Only talk to the database
- Add servers to add capacity
Scalability & Performance
Multi-threaded Apps

- Multi-core servers
- Web server (IIS)
- Queues (MSMQ)
- Database (MS SQL Server)
- Processing services (custom C#)
Scalability & Performance

Asynchronous Services

- **Local queues**
  - Each web server has a queue

- **Minimize processing**
  - Example: game stats upload
  - Originally XML
    - Too large
  - Then compressed XML
    - Too slow
  - Finally, custom binary
    - Small and fast
Scalability & Performance
Bidirectional Services

- Cache when possible
  - Reduces DB load
  - Reduces turnaround time
  - Example: MOTD caching
Scalability & Performance
Bidirectional Services

- Cache when possible
  - Reduces DB load
  - Reduces turnaround time
  - Example: MOTD caching
Scalability & Performance
Bidirectional Services

- Cache when possible
  - Reduces DB load
  - Reduces turnaround time
  - Example: MOTD caching
Scalability & Performance

Bidirectional Services

- Cache when possible
  - Reduces DB load
  - Reduces turnaround time
  - Example: MOTD caching
Scalability & Performance

- Multiple application servers
- Single database is a bottleneck
Scalability & Performance

Database - Replication

- Production database
- Replicated to:
  - www.gearsofwar.com
  - Epic internal
- Expensive queries don’t run on production
Scalability & Performance
Database – Profile Caching

- Matches a player to a profile ID
- Performance bottleneck
- Cache profiles when possible
  - Web handlers
  - Asynchronous processing services
  - Post-processing services
- Pro: Increased performance
- Con: Cache management
Scalability & Performance
Database – Profile Caching

GetOrAddProfile(Player)
{
    if(PlayerInLocalCache)
        return ProfileID from Cache
    if(PlayerInProfilesTable)
        return ProfileID from Table
    AddPlayerToProfilesTable(Player)
        return ProfileID from Table
}
Scalability & Performance
Database – Profile Caching

BeginTransaction()
    ProfileID = GetOrAddProfile(Player)
    AddProfileIDToLocalCache(ProfileID)
    DoOtherDatabaseOperations()
CommitTransaction()
Scalability & Performance
Database – Profile Caching

BeginTransaction()
    ProfileID = GetOrAddProfile(Player)
    AddProfileIDToLocalCache(ProfileID)
    DoOtherDatabaseOperations()
CommitTransaction()
Scalability & Performance
Database – Profile Caching

BeginTransaction()
   ProfileID = GetOrAddProfile(Player)
   AddProfileIDToLocalCache(ProfileID)
   DoOtherDatabaseOperations()
CommitTransaction()
if(TransactionSucceeded)
   AddProfileIDToLocalCache(ProfileID)
Stress Testing

- Stress test client
- Simulates game operations
  - MOTD requests
  - Game stats uploads
  - etc.
- Configurable
  - Operations per second
  - Number of threads
  - Length of run
  - etc.
Stress Testing

- Find bottlenecks
- Test optimizations
- Example: game stats uploads
  - Identify problem area
  - Test alternatives
  - Measure performance change
Stress Testing

- Find bugs
- Test fixes
- Example: database deadlock
  - Identify and fix bug
  - Verify fix
Administration

- Web-based
- No direct access to production
- Did not know who would be administering MCP
- Already using web server
- Web browser as admin client
Administration
Features

Message of the day

- Thanks for playing Gears 2 Valentine’s Day double XP event! Watch here and geardsofwar.com for upcoming events.

- It’s all about sticking close to that special someone in the Valentine’s Day double XP event! From Friday, February 12, through Monday, February 15, earn double XP in Wingman and Horde, as well as double points for revives in all game types.

- We hope you enjoyed the 12 Days of Gearsmas! Please keep an eye on this space as well as www.geardsofwar.com for news of upcoming events!
Administration

Features

• Custom game settings
Administration Features

- Edit MCP configuration
Administration
Special Events

- Schedule custom game settings
- Schedule MOTDs
- Valentine’s Day Event
- Old School Weekend
- Fourth of July Weekend
Administration
Special Events

- 12 Days of Gearsmas
  - Different MOTDs and game settings each day
  - 150+ MOTDs scheduled in multiple languages
Administration

- Can be cumbersome for common uses
- Setting up MOTD can be time consuming
- Post-release updates helped
- Uses cases are important
Hosting
Hosting

- Microsoft hosts Gears 2 MCP
- We do not have direct access
- Only web admin access
- Changes can take weeks
- Update checklist
Hosting Update Checklist

- Database Scripts
- Functional Tests
- Stress Tests
- Front End
- Web Backend
- Upgrade Doc
- Health Model
- Nightly MOM Data
- Perf Counters
- Error Handler
- Replication
- Data Aging
- Post Processing
- Reporting/Charting
Hosting MCP Front End

- Deployment and verification tool
- Helps with MCP installation
- Also used for local development
What could possibly go wrong?
Data Center Problems

- Failed cooling system
  - Machines overheating
  - Multiple day downtime
  - Luckily, not production
Data Center Problems

- Failed cooling system
  - Machines overheating
  - Multiple day downtime
  - Luckily, not production

- Failed power supply
  - Redundant backup failed
  - Outage and lost data
  - Unfortunately, production
Monitoring

- MOM (SCOM)
- Health model
  - Performance counters
  - Event logs
- Lots of iteration
  - Warning thresholds
  - Error thresholds
- Hard to predict real world
Monitoring Problems

- Thresholds were set too low
  - We did not want to miss any issues
  - But we ended up with false alarms

- Event log was not cleared before release
  - Simulation had filled the event log
  - Alerting was turned on
  - Flood of false alerts
  - We crashed a phone
  - SMS charges $$$
Monitoring
Ongoing...

- Problems can always happen
- Need to continue monitoring
  - For the life of the game
  - Or as long as online is supported
Monitoring Ongoing...

- Problems can always happen
- Need to continue monitoring
  - For the life of the game
  - Or as long as online is supported
Launch

- Testing had been done
  - Local, PartnerNet, Production
- But Production testing was done from inside the network
- External connections had not been tested – and did not work
- We could only sit and wait
- Was fixed less than 2 minutes before our midnight release
Success!

- Great for Gears 2
  - New features
  - Gameplay feedback
  - Special events
  - Held up under load
- Platform for future products
  - Using with the UDK (Unreal Development Kit)
- Available to Unreal Engine licensees
Q&A

Dan Schoenblum  

dan.schoenblum@epicgames.com

Epic Games  

Booth BS200, South Hall  

www.epicgames.com

Unreal Technology  

www.unrealtechnology.com