Gears of War 3 Analytics: Optimizing the Online Experience, or How I Learned to Stop Worrying and Love the Beta

Joe Graf (Dr. Statslove)

Game Connection 2011
December 8, 2011
About Me

• Been at Epic over 8 years

• Primarily focused on online features for our games and engine

• Lead Gameplay Programmer for Gears of War 3
About You

You are at this talk because you are a

- Programmer responsible for matchmaking
- Designer that wants to know more
- Producer that wants to know where the Gears team invested
About Gears of War

• First two games were very popular multiplayer games

• Gears of War 2
  – 6 million units sold
  – More than 1 million simultaneous players

• Gears of War 3 looked to expand upon that via the inclusion of dedicated servers
Overview

• Network infrastructure of Gears
• Types of metrics the Gears beta captured
• How that data was used to pinpoint issues
• How the Gears matchmaking system works
• What we adjusted post beta for a great launch
Why do a beta?

• Gears 2 launch wasn’t seamless
  – Overburdened Xbox Live services caused slow matchmaking at launch
  – Took about 1 month post ship to get everything solid

• Unanswered questions
  – Tested dedicated server support in Gears 2, but not at launch levels of players
Why worry about a beta?

• Risk!!!
  – Betas are seen as “Demos” not work in progress
    • Do it too early and pre-orders are cancelled
  – Schedule impact
    • Essentially shipping the game twice
    • Reduced output from team while beta is running
  – Throwaway work is polish lost on final product
    • Remove content not included in the beta
    • Custom UI flow and exposure to different TCRs
Goals for the Beta

• Find any scalability issues in matchmaking
  – Gears 2 did not exhibit issues with < 150k simultaneous players
  – Gears 2 test validated approach but not scalability
• Determine how many datacenters we need and where to locate them
• Test our data collection pipeline beginning to end
• Verify gameplay balancing with real players
• Test alternative matchmaking algorithms
Network Infrastructure
Unreal MCP

• Epic’s Xbox Live Server Platform (XLSP)
  – Determines best datacenter for each player
  – Handles the playlist population tracking
  – Allows the Gears team to change data on the Xbox client without requiring a patch
  – Responsible for processing all uploads from the game
Data Captured

• Player data
  – Each player uploads their data as XML once per day

• Matchmaking data
  – Party host uploads an XML payload with the details

• Gameplay data
  – Each host uploads a compressed binary file containing all of the events that were not filtered out

• 1 Terrabyte of data collected in first week of beta
Player Data

• Population counts and location data
• Aggregate view of user preferences
  – MP character
  – Starting weapons and weapon skins
  – Button and stick configuration
    • Inverse, southpaw, legacy, etc.
• Player statistics and progress
  – Nearly 2,000 different data points
Player Data

Total Users: 1,289,329

Top Countries by User:
- United States: 132,816
- United Kingdom: 64,814
- Mexico: 52,602
- France: 35,885
- Spain: 22,180
- Australia: 14,680
- Italy: 14,147
- Germany: 10,190
- Ireland: 6,207
Player Data (cont.)

Multi Player Preferences

<table>
<thead>
<tr>
<th>Cog MP Character</th>
<th>Anya</th>
<th>Baird</th>
<th>Cole</th>
<th>ColeThrashball</th>
<th>Dom</th>
<th>Marcus</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>7.98%</td>
<td>4.78%</td>
<td>18.94%</td>
<td>13.78%</td>
<td>34.34%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Locust MP Character</th>
<th>Drone</th>
<th>Flamer</th>
<th>Kantus</th>
<th>SavageHunter</th>
<th>SavageTheron</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>8.06%</td>
<td></td>
<td>14.40%</td>
<td>25.34%</td>
<td>18.85%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Default MP Weapon</th>
<th>Hammerburst</th>
<th>Lancer</th>
<th>RetroLancer</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>14.87%</td>
<td>41.58%</td>
<td>43.56%</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Default Secondary</th>
<th>Gnasher</th>
<th>SawedOffShot</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>29.31%</td>
<td></td>
</tr>
</tbody>
</table>

% of Total Number of Records
Player Data (cont.)

• NAT (network address table) types
  – Data is different from Gears 2 and final product
  – More Open NATs, fewer Moderate or Strict
Matchmaking Data

- Search times for each type of search
- Party details
  - Size, unique ids, average skill, average XP level
- Returned host data
  - Ping, match quality, average skill
- Number of searches
- Number of results per search
Matchmaking Data
Matchmaking Data (cont.)

Size of Parties

<table>
<thead>
<tr>
<th>Party Size</th>
<th>Ranked</th>
<th>Social</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>66.64%</td>
<td>78.25%</td>
</tr>
<tr>
<td>2</td>
<td>15.14%</td>
<td>12.49%</td>
</tr>
<tr>
<td>3</td>
<td>7.88%</td>
<td>4.91%</td>
</tr>
<tr>
<td>4</td>
<td>5.24%</td>
<td>2.79%</td>
</tr>
<tr>
<td>5</td>
<td>5.10%</td>
<td>1.57%</td>
</tr>
</tbody>
</table>

Avg and Max Party Skill Read Time

<table>
<thead>
<tr>
<th>Party Size</th>
<th>Max</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>40.07</td>
<td>0.37</td>
</tr>
<tr>
<td>2</td>
<td>30.61</td>
<td>0.37</td>
</tr>
<tr>
<td>3</td>
<td>10.29</td>
<td>0.33</td>
</tr>
<tr>
<td>4</td>
<td>6.12</td>
<td>0.36</td>
</tr>
<tr>
<td>5</td>
<td>3.30</td>
<td>0.32</td>
</tr>
</tbody>
</table>

Average MM Times by Party Size

<table>
<thead>
<tr>
<th>Party Size</th>
<th>ElapsedSeconds</th>
<th>Ranked</th>
<th>Social</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>78.87</td>
<td>72.86</td>
<td>40.08</td>
</tr>
<tr>
<td>2</td>
<td>72.86</td>
<td>70.70</td>
<td>41.05</td>
</tr>
<tr>
<td>3</td>
<td>70.70</td>
<td>69.60</td>
<td>39.83</td>
</tr>
<tr>
<td>4</td>
<td>66.32</td>
<td>66.32</td>
<td>47.42</td>
</tr>
<tr>
<td>5</td>
<td>50.00</td>
<td>45.15</td>
<td>45.15</td>
</tr>
</tbody>
</table>
Matchmaking Data (cont.)
Gameplay Data

• Map and game mode played
• Weapon data
  – Fired, melee, picked up, dropped
• Player locations recorded for all events
• Each recorded event has a granularity level for throttling via Unreal MCP
Gameplay Data

Games By Hour of Day

- Capture the Leader
- King of the Hill
- Team Deathmatch
Gameplay Data (cont.)

- Games by Map by Host
  - MapName
    - MP_Checkout
    - MP_OldTown
    - MP_Trenches
    - MP_Thrashball
  - Unique Sessions: 0K, 500K, 1000K

- Games by Map Daily
  - MapName
    - MP_Checkout
    - MP_OldTown
    - MP_Trenches
    - MP_Trenches
  - Unique Sessions: 0K, 10K, 20K, 30K, 40K
  - Dates: May 4, 6, 8, 10, 12, 14, 16

- Games By Map
  - Unique Sessions: 0K, 10K, 20K
Gameplay Data (cont.)

Average Match Length Histogram (Last 2 Weeks)

- **King of the Hill**
  - Match Length (minutes) Capped (bin)
  - % of Total

- **Team Deathmatch**
  - Match Length (minutes) Capped (bin)
  - % of Total

- **Capture the Leader**
  - Match Length (minutes) Capped (bin)
  - % of Total

Graph shows distribution of match lengths for different game types over the last 2 weeks.
Gameplay Data (cont.)

• Goal was 10 minutes per match
  – Needed to tweak game rules to bring them closer to that target
Gameplay Data (cont.)

Active Reload by Weapon

<table>
<thead>
<tr>
<th>GearWeap_Name</th>
<th>Reload Failed</th>
<th>Reload Success</th>
<th>Reload Super Success</th>
</tr>
</thead>
<tbody>
<tr>
<td>GearWeap_LocustBurstPistol</td>
<td>14%</td>
<td>70%</td>
<td>16%</td>
</tr>
<tr>
<td>GearWeap_Shotgun</td>
<td>5%</td>
<td>31%</td>
<td>64%</td>
</tr>
<tr>
<td>GearWeap_AssaultRifle</td>
<td>34%</td>
<td>62%</td>
<td>5%</td>
</tr>
<tr>
<td>GearWeap_HeavySniper</td>
<td>40%</td>
<td>59%</td>
<td>9%</td>
</tr>
<tr>
<td>GearWeap_SniperRifle</td>
<td>5%</td>
<td>39%</td>
<td>66%</td>
</tr>
<tr>
<td>GearWeap_Shotgun_SawedOff</td>
<td>13%</td>
<td>36%</td>
<td>52%</td>
</tr>
<tr>
<td>GearWeap_LocustPistol</td>
<td>5%</td>
<td>49%</td>
<td>45%</td>
</tr>
<tr>
<td>GearWeap_HeavyMortar</td>
<td>52%</td>
<td>43%</td>
<td>4%</td>
</tr>
<tr>
<td>GearWeap_LocustAssaultRifle</td>
<td>8%</td>
<td>50%</td>
<td>42%</td>
</tr>
<tr>
<td>GearWeap_Bow</td>
<td>5%</td>
<td>56%</td>
<td>41%</td>
</tr>
<tr>
<td>GearWeap_DiggerLauncher_Content</td>
<td>6%</td>
<td>56%</td>
<td>38%</td>
</tr>
<tr>
<td>GearWeap_FlameThrower</td>
<td>6%</td>
<td>57%</td>
<td>37%</td>
</tr>
<tr>
<td>GearWeap_Boomshot</td>
<td>8%</td>
<td>62%</td>
<td>35%</td>
</tr>
<tr>
<td>GearWeap_COGPistol</td>
<td>11%</td>
<td>60%</td>
<td>31%</td>
</tr>
<tr>
<td>GearWeap_RetroLancer</td>
<td>8%</td>
<td>68%</td>
<td>21%</td>
</tr>
<tr>
<td>Grand Total</td>
<td>8%</td>
<td>46%</td>
<td>47%</td>
</tr>
</tbody>
</table>
Gameplay Data (cont.)
Gameplay Data (cont.)
Gears Matchmaking

• Gears 3 used the same algorithm as Gears 2
  – Minor changes to accommodate dedicated servers

• Player hosted matches are still required
  – Things can and will go wrong
    • Player’s are generally not network engineers
    • Servers can fail and network outages can prevent access to datacenter
    • Some regions have poor infrastructure
Terminology

- **Match Quality**
  - How likely a match ends in a draw

- **QoS – Quality of Service**
  - Network quality between client and host

- **Best search**
  - Exact match based upon party size

- **Any search**
  - Match based upon space greater than party size

- **Empty search**
  - Find a new server to configure for the desired mode
Matchmaking Process

1. Request Datacenter
2. Read Skill Data
3. Search for Matches
4. Rate Matches
5. Join Match
• Use IP to location to find closest datacenter
  – Unreal MCP returns an ID for this datacenter
• Done once per player signin
• Saved in the player’s profile in case the look up fails
• Each playlist has its own skill table
• Done once per search initiation
• Used to provide Xbox Live with a single value to rate advertised sessions against
• Used to calculate match quality
Matchmaking Process (cont.)

1. Request Datacenter
2. Read Skill Data
3. Search for Matches
4. Rate Matches
5. Join Match

- "Best" Search
- "Any" Search
- "Empty" Search

Found?
Matchmaking Process (cont.)

1. Request Datacenter
2. Read Skill Data
3. Search for Matches
4. Rate Matches
5. Join Match
6. Sort Into Ping Buckets
7. Sort Buckets By Match Quality
8. Request Reservation
9. Ok?
Beta Launch Plan

• Access to beta was in multiple phases
  – Epic employees, Microsoft employees, friends and family
  – Bulletstorm Epic Edition purchasers
  – Gears of War Pre-orders

• Phased approach allowed us to manage player population growth
  – Thousands, tens of thousands, hundreds of thousands
Beta Launch

• Employees, friends and family
  – Everything is fine

• Bulletstorm players join in
  – Forums and Twitterverse indicate matchmaking times are slow

• War Room
  – We gather to go over the data and drill into what’s going on
QoS Data Indicates a Problem
Too Many Servers Are Unreachable
Leading To Slow Matchmaking
What’s Going On?

• Contention on QoS results
  – Dedicated servers were configured like Xbox clients
  – Except all clients were funneled to similar sets of servers: different pattern from player hosted

• The fix?
  – Increase the configurable max number of simultaneous QoS probes
QoS Problem Resolved
QoS Problem Over Time
Production QoS Data

### QoS Seconds by Search Type

<table>
<thead>
<tr>
<th>Search Type</th>
<th>Any</th>
<th>Best</th>
<th>Empty</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10K</td>
<td>1K</td>
<td>0K</td>
</tr>
</tbody>
</table>

Searches MatchMode:
- Ranked
- Social

### Percentage of sessions unreachable to QoS

<table>
<thead>
<tr>
<th></th>
<th>Dedicated</th>
<th>Player</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any</td>
<td>17%</td>
<td>32%</td>
</tr>
<tr>
<td>Best</td>
<td>17%</td>
<td>37%</td>
</tr>
<tr>
<td>Empty</td>
<td>15%</td>
<td></td>
</tr>
</tbody>
</table>
QoS was improved, but...

- Matchmaking charts show clients are host starved
  - Not enough hosts are being found
  - Resulting in player hosted sessions
Too Few Search Results
Less is More
QoS and Search Results Change
QoS and Search Results Change
Matchmaking Algorithm Experiment

• Prefer matches by distance from party’s average XP Level
  – Sorting has to be on the backend where the data is available
  – Risky, because it overrides all standard Xbox Live sorting mechanisms

• Why do this?
  – Player perception is XP Level means skill
Experiment Results

- `<SearchResult ... Ping="50" Quality="0.924"/>
- `<SearchResult ... Ping="100" Quality="0.668"/>
- `<SearchResult ... Ping="100" Quality="0.409"/>

- `<Reservation>
  `<Player ... Skill="23" XpLevel="92"/>
  `<Player ... Skill="28" XpLevel="89"/>
  `<Player ... Skill="25" XpLevel="86"/>
</Reservation>
- `<Reservation>
  `<Player ... Skill="19" XpLevel="90"/>
</Reservation>
- `<Reservation>
  `<Player ... Skill="28" XpLevel="87"/>
</Reservation>`
Experiment Results

- `<SearchResult ... Ping="50" Quality="0.924" />`
- `<SearchResult ... Ping="100" Quality="0.668" />`
- `<SearchResult ... Ping="100" Quality="0.409" />`

- `<Reservation>`
  - `<Player ... Skill="23" XpLevel="92" />`
  - `<Player ... Skill="28" XpLevel="89" />`
  - `<Player ... Skill="25" XpLevel="86" />`

- `<Reservation>`
  - `<Player ... Skill="19" XpLevel="90" />`

- `<Reservation>`
  - `<Player ... Skill="28" XpLevel="87" />`
Experiment Results

- <SearchResult ... Ping="50" Quality="0.924" />
  <SearchResult ... Ping="100" Quality="0.668" />
  <SearchResult ... Ping="100" Quality="0.409" />

- <Reservation>
  <Player ... Skill="23" XpLevel="92" />
  <Player ... Skill="28" XpLevel="89" />
  <Player ... Skill="25" XpLevel="86" />
</Reservation>
<Reservation>
  <Player ... Skill="19" XpLevel="90" />
</Reservation>
<Reservation>
  <Player ... Skill="28" XpLevel="87" />
</Reservation>
Beta Results

GEARS OF WAR 3

BETA STATS

23 BILLION BULLETS FIRED
4.9 BILLION RIBBONS
927 MILLION KILLS
291 MILLION MEDALS EARNED
131 MILLION EXECUTIONS PERFORMED
23 MILLION CHAINSAW DEATHS
249 YEARS WORTH OF MATCHES PLAYED
145 DIFFERENT COUNTRIES
Production Changes

• Added the ability to place servers in buckets

• Added looping in addition to bucketing

• Added new QoS configuration options

• Changed Quick Match searching to be XP Level based
Production Matchmaking Process

1. Request Datacenter
2. Read Skill Data
3. Search for Matches 0..N
   - "Best" Search 0..X
   - "Any" Search 0..Y
   - "Empty" Search 0..Z
4. Rate Matches
5. Join Match
6. Found?
Matchmaking Times

Beta

Production
Takeaways

• Record all input matchmaking variables

• Record all output results
  – Don’t assume a detail isn’t important

• You can never have too many configurable knobs to turn
  – We couldn’t change some values in the beta
Why I learned to love the beta

- The extra effort up front was offset by removing the need for emergency patches
- Instead of going from 0 to 1 million, we could learn while managing the growth
- We discovered and addressed bottlenecks in every phase of our data collection pipeline
  - The data doesn’t do you any good if you can’t look at it
- Gears of War 3 launch was our best ever
Gears of War 3 Analytics:
Optimizing the Online Experience, or How I Learned to Stop Worrying and Love the Beta

Special thanks to:
Wes Hunt
Josh Markiewicz
Justin Rowland
Ian Thomas
Sam Zamani

Joe Graf
Epic Games, Inc.
Twitter: @EpicCog