Scalability and Load in Online Games

James Gwertzman
@gwertz
Promise & peril of live game operations

- Launched 3/2/12
- Shot to top 10
- Servers crashed
- Offline for 168 days
- $27M lost revenue

- Re-launched 8/17/12
- Top-grossing game for nearly 3 years

LTD revenue: $150M+

Updated more than 45 times

Rank history for top grossing games, iOS in United States

Source: AppAnnie, NPD, ThinkGaming
The old days…
Nowadays…

- **Build game**
- **Build backend**
- **Build tools for ops team**
- **Launch**
- **Deplpy servers**
- **Customer service**
- **User Acquisition**
- **Segment & target customers**
- **Business intelligence**
- **Offers & Promotions**
- **Host in-game events**
- **Update content**
- **Business intelligence**
Characteristics of live games

• Many more writes than reads (primarily driven by analytics)
• Authoritative server-side game logic (to prevent cheating)
• Millions of users (often with transient relationships)
• Peak load often hits on day one of release
• Must be scalable, global, and reliable all at once
• Game studios often inexperienced at large-scale system design
• Testing w/ real user “stories” at scale is hard but critical
Back-end services (cross-platform, one-stop shop)

Live ops tools and dashboards (mission control for the whole team)

Integration with other partners (building out a full ecosystem)

and many more coming...
Typical backend services

- **Player Accounts**
  - Authentication
  - Profile Management
  - Account Linking
- **Data Storage**
  - Per player
  - Per title
  - Per character (under one player)
  - Files / CDN delivery
- **Commerce**
  - Catalog Management
  - Virtual Currencies
  - Player Inventory
  - In-game Purchasing
  - Receipt validation (Apple/Google/Amazon)
  - Marketing and Promotion
- **In-game Marketing**
  - Push Notifications
  - In-game Messaging
- **Product Management**
  - Analytics and Reporting
  - Customer Segmentation
  - Customer Support Tools
  - Abuse Reporting and Banning
- **Social**
  - Friends Lists
  - Player Chat
  - Leaderboards
  - Game forum integration
  - Trading / gifting
- **Multiplayer**
  - Photon integration (real-time / turn-based)
  - Matchmaking
  - Custom game server hosting
  - Server monitoring
- **Game logic**
  - Server-hosted JavaScript
Dealing with elastic load

• Peak load is often launch day
  – Hard to predict actual load
• We solve using elastic compute & storage solutions
• No transaction support in DB, so must solve at logic level
Applying actions to user segments

- Users are grouped into segments based on behavior
- Games apply actions to segments
  - Send a push notification
  - Grant an item
- Need both batch and real-time processing
- We use a hybrid model, with duplicate storage
  - Redshift for bulk actions
  - DynamoDB for individual triggers
Global state for matchmaking

- Matching users for a game is harder than it sounds
  - Millions of players
  - Must be atomic
  - Lots of state and complex rules

- We use a single server, w/ failover
  - State is written through to DB
  - If primary fails, secondary reads in state from DB and takes over
  - This happens very quickly...
Thank you

James Gwertzman (james@playfab.com) @gwertz or @playfabnetwork
Create a free account today at www.playfab.com