Game Developers Conference® | March 9-13, 2010 | Moscone Center | San Francisco, CA

iPhone Games SUMMIT

Building the Server Software for Eliminate

ngmoco:)

THINK SERVICES
A DIVISION OF UNITED BUSINESS MEDIA LLC

GDC 10
www.GDConf.com

iPhoneGames SUMMIT

ngmoco:)

Introduction

- Stephen Detwiler
  - Director of Engineering, ngmoco:)
- James Marr
  - Lead Engineer R&D, ngmoco:)

iPhoneGames SUMMIT

ngmoco:)

Introduction

- Build the definitive FPS for iPhone
  - in only 5 months
- Multiplayer deathmatch
  - wifi and 3g
- Free to play

- With three engineers

iPhoneGames SUMMIT

ngmoco:)

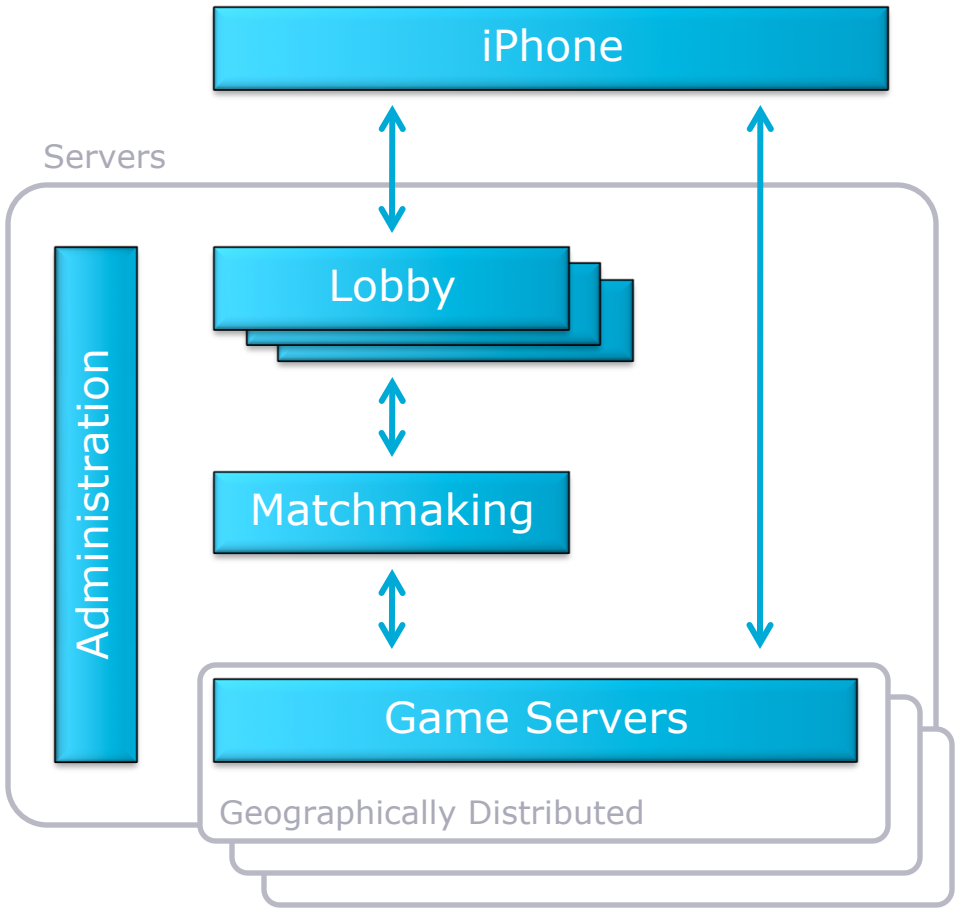
Outline

- Gameplay
- Lobby
- Matchmaking
- Load Testing
- Live Tuning
- Deployment
- Monitoring

iPhoneGames SUMMIT

ngmoco:)

Server Architecture

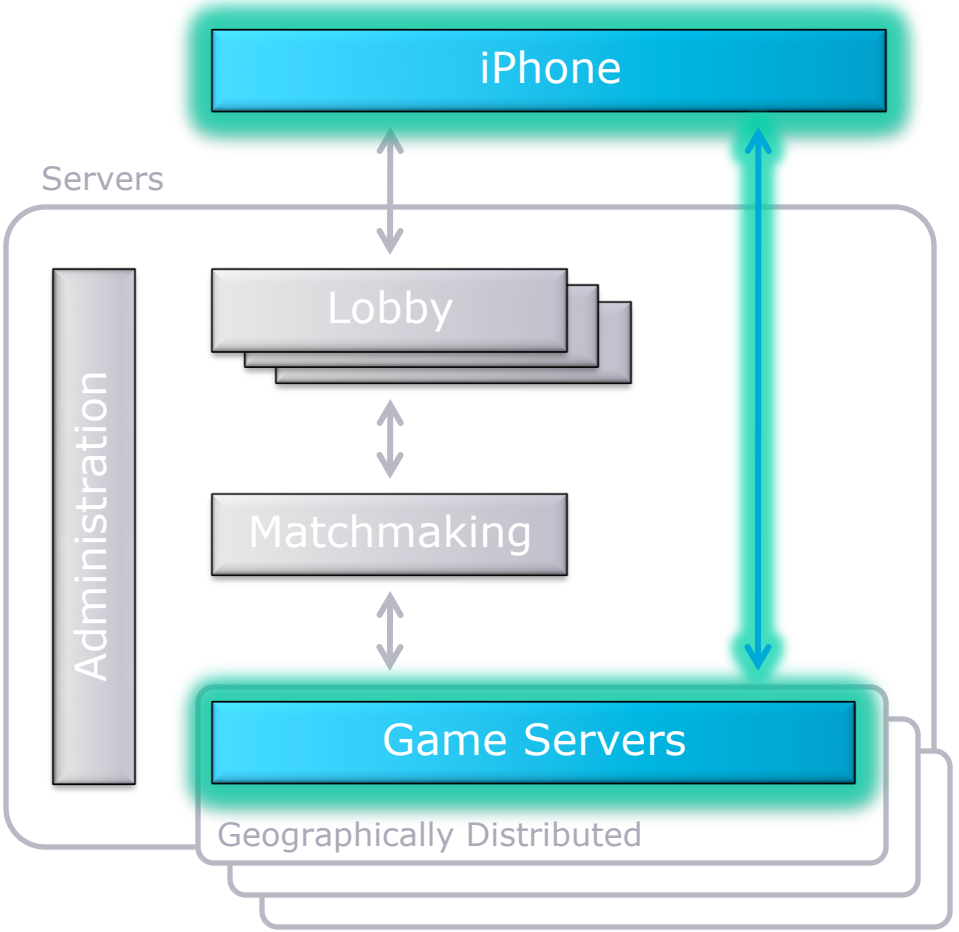


iPhoneGames SUMMIT

ngmoco:)

Gameplay

Topic 1 of 7



iPhoneGames SUMMIT

ngmoco:)

Gameplay: Requirements

- 3G requirement drives decision
  - ~100kbps, 150ms latency
- Aggressive bandwidth optimization
- Prediction to hide latency
- UDP

iPhoneGames SUMMIT

ngmoco:)

Gameplay: Options

- Are there any opensource options?
  - Shipping to clients, so no GPL
- Are there any commercial options?
- Yes, Quake 3
- Dialup from 1999 looks a lot like 3G from 2009

iPhoneGames SUMMIT

ngmoco:)

Gameplay: Q3 Cost

- Source code
- plus full rights
- minus any technical support
- = $10k

- Same cost as a man month

iPhoneGames SUMMIT

ngmoco:)

Gameplay: Q3 Benefits

- Graphics
  - BSP + portals
  - Dynamic lights, static lightmaps
  - Keyframe animation
- Tools
  - Custom map editor (Radiant)
  - 3DS Max model animation exporters
- Lots of information online about how to extend the engine

iPhoneGames SUMMIT

ngmoco:)

Gameplay: Moving On

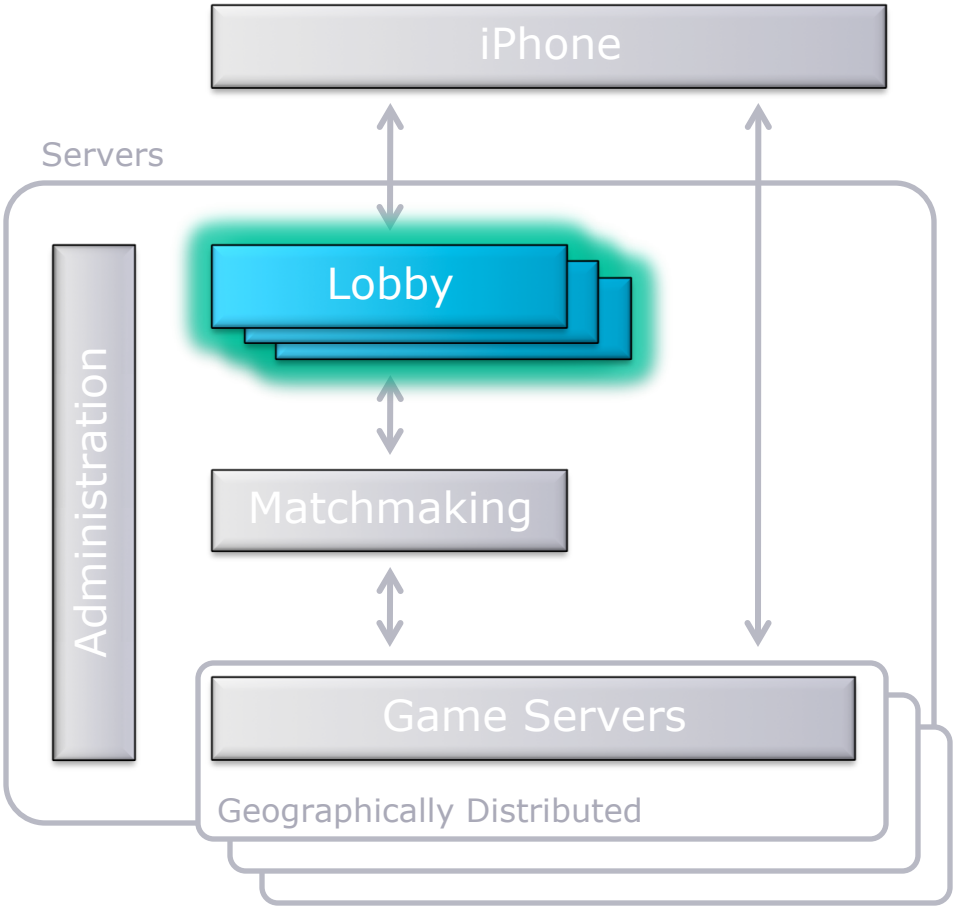
- Purchased solution for "mundane" gameplay networking
- Able to focus on rest of experience

iPhoneGames SUMMIT

ngmoco:)

Lobby

Topic 2 of 7



iPhoneGames SUMMIT

ngmoco:)

Lobby: Requirements

- Handles everything outside of realtime gameplay
  - Inventory and commerce
  - Proxy to Plus+ services
  - Chat
  - Matchmaking requests
  - Party management
- Support 10K+ concurrent users

iPhoneGames SUMMIT

ngmoco:)

Lobby: Approach

- Rejected: Periodic HTTP polling
  - Easy to scale
    - Lots of HTTP front ends
    - Big database backend
  - Latency will be high in many cases
    - TCP socket setup over 3G is slow
      - Sometimes over 2 seconds!
  - Hard to tell when users go away
    - Must have timeout thresholds

iPhoneGames SUMMIT

ngmoco:)

Lobby: Approach

- Chosen: Persistent TCP socket
  - Only one initial TCP setup
  - User is gone when socket closes
  - Much lower message delivery latency
  - Can push messages
  - Harder to scale
    - One socket per user

iPhoneGames SUMMIT

ngmoco:)

Lobby: Implementation

- This will take more than 5 months to build.
  - What can we use off the shelf?
- Yes, XMPP

iPhoneGames SUMMIT

ngmoco:)

Lobby: XMPP

- Jabber/IM/Google Talk
  - Proven to be scalable
- TCP with XML payloads
- Can also route custom messages
- Many off the shelf implementations
  - jabberd, jabberd 2.x, ejabberd , etc.

iPhoneGames SUMMIT

ngmoco:)

Lobby: Evaluating

- jabberd and jabberd 2.x
  - C/C++ codebase
  - Not actively supported
  - Early testing showed it did not scale well past 1000 users
  - Implementation difficult to extend

iPhoneGames SUMMIT

ngmoco:)

Lobby: Evaluating

- ejabberd
  - Highly scalable
    - Load tested to 30K concurrent users
  - Extendable
  - Active community
- But written in erlang

iPhoneGames SUMMIT

ngmoco:)

Lobby: Erlang


```
{Priority, RepackGameServers, IsGameServer} =
case FromSession#ng_session.is_admin of
true ->
        case lists:filter(fun({"isGameServer", _IsGS}) -> true;
                (_) -> false end, OriginalAttributes) of
                [{_, IsGS}] -> {"0", "0", IsGS};
                _ -> {"0", "0", "1"}
        end;
false ->
        AnyEnergy = does_any_player_have_energy(Players),
        case AnyEnergy of
                true -> {"1", "0", "0"};
                _ -> {"0", "1", "0”}
        end
end,
```

iPhoneGames SUMMIT

ngmoco:)

Lobby: Erlang

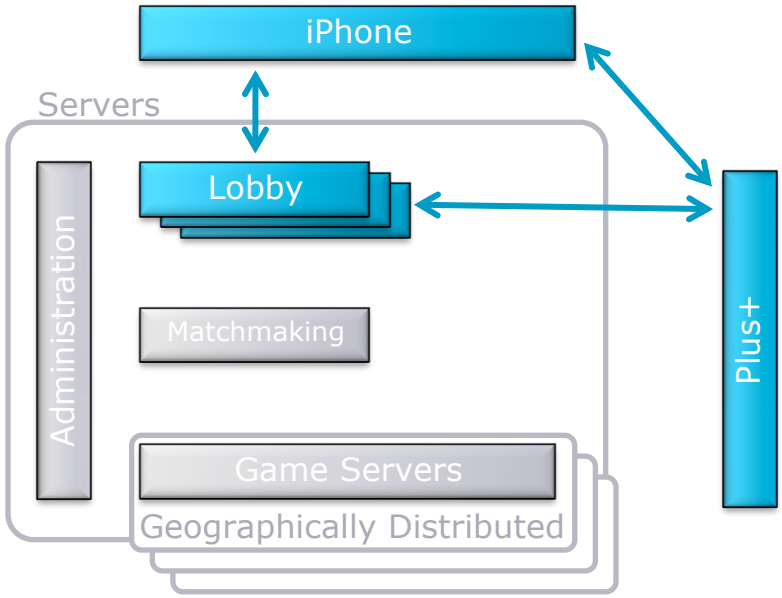
- Functional language
- Crazy syntax
- Distributed message passing built into language
- Data persistence occurs in database

iPhoneGames SUMMIT

ngmoco:)

Lobby: Plus+ Integration

- Users log into XMPP using Oauth credentials from Plus+
- Plus+ Friends and Followers populate user's XMPP roster

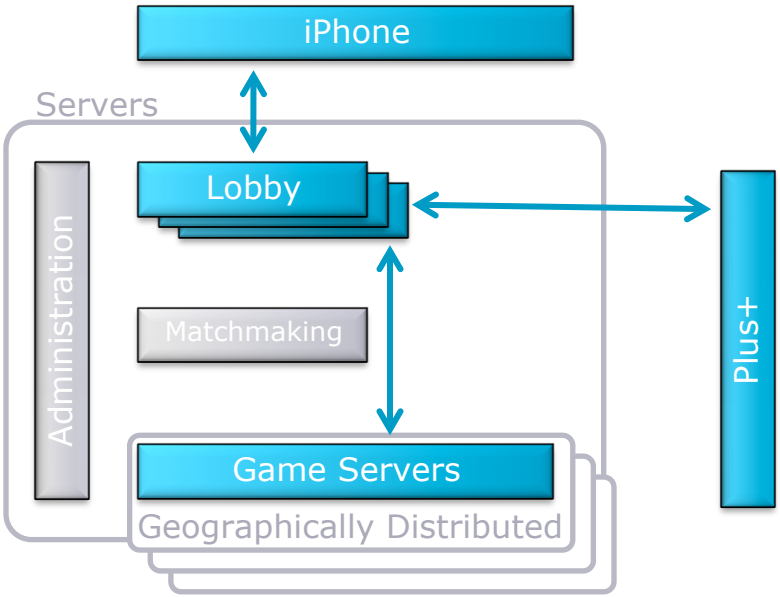


iPhoneGames SUMMIT

ngmoco:)

Lobby: Scaling

- ejabberd clusters well
  - Almost for free using erlang

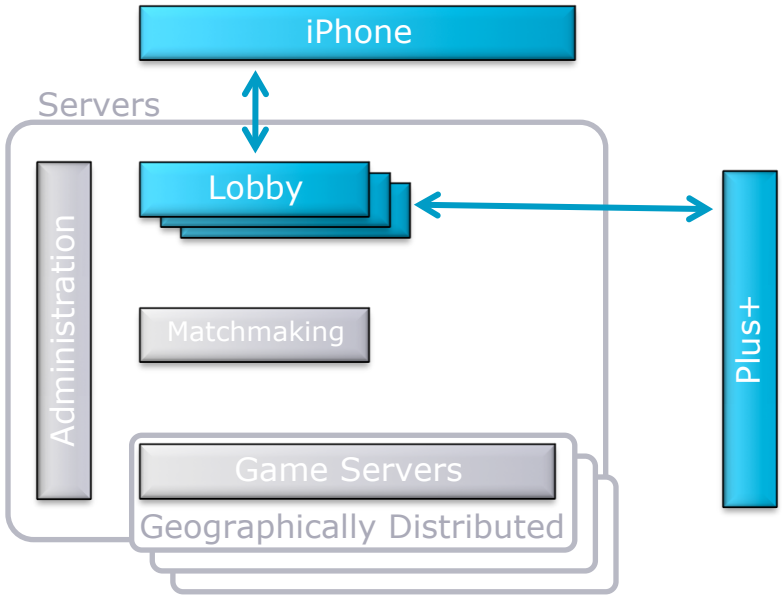


iPhoneGames SUMMIT

ngmoco:)

Lobby: Inventory & Purchasing

- All persistent data stored in Plus+
- XMPP validates and caches data
- XMPP nodes can start and stop at anytime

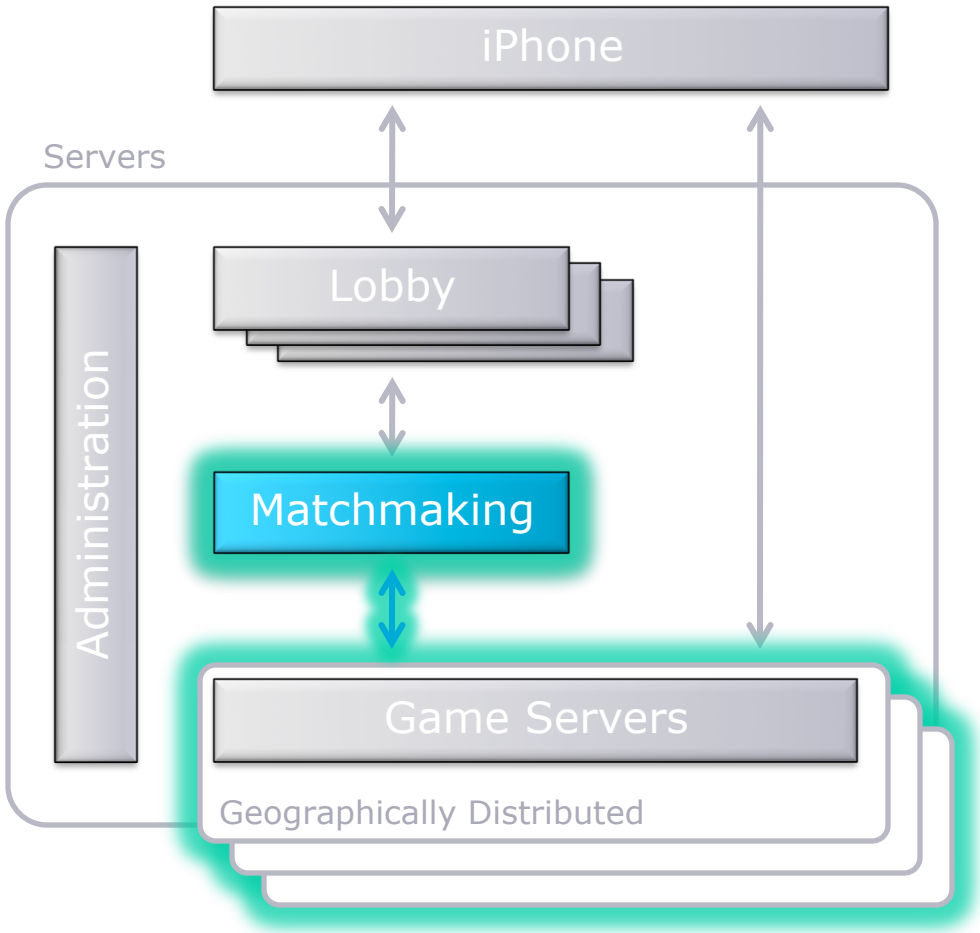


iPhoneGames SUMMIT

ngmoco:)

Matchmaking

Topic 3 of 7



iPhoneGames SUMMIT

ngmoco:)

Matchmaking: Goals

- Console quality matchmaking
- Dirt simple user experience
  - Press a button
  - Play against fun opponents

iPhoneGames SUMMIT

ngmoco:)

Matchmaking: Options

- Are there commercial options?
  - Microsoft? Infinity Ward? Blizzard?
- Are there opensource alternatives?

- No. We're building our own

iPhoneGames SUMMIT

ngmoco:)

Matchmaking: Overview

- Matchmaking server
  - Receives requests from Lobby server
  - Finds a good grouping of players
  - Launches game server instance
  - Inform clients through Lobby server

iPhoneGames SUMMIT

ngmoco:)

Matchmaking: Instances

- Quake 3 dedicated server is one process per concurrent game
- Game manager on each server
  - Talks to matchmaking server
  - Launches instances on-demand
  - Reports max instance capacity

iPhoneGames SUMMIT

ngmoco:)

Matchmaking: Approach

- Rejected: SQL DB
  - All state stored in DB
  - Query DB, process results, repeat
  - Easy to cluster, provide redundancy

  - High data latency
  - Complicated

iPhoneGames SUMMIT

ngmoco:)

Matchmaking: Approach

- Accepted: In Memory
  - All players kept in memory
  - Higher performance
  - Fast to implement

  - Won't cluster, one box must do it all
  - Server crashes lose some data

iPhoneGames SUMMIT

ngmoco:)

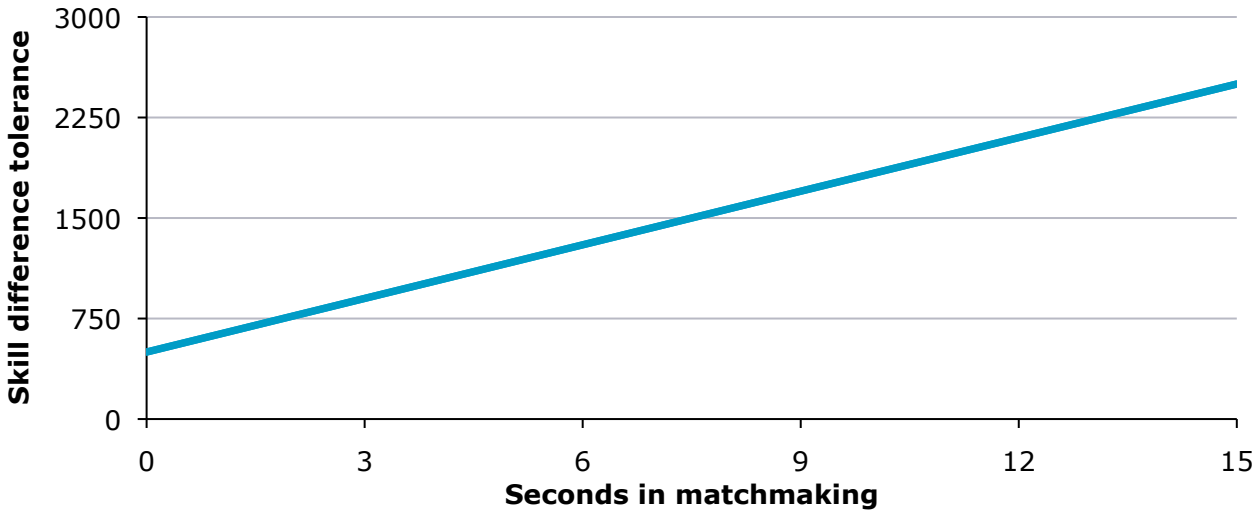
Matchmaking: Qualities

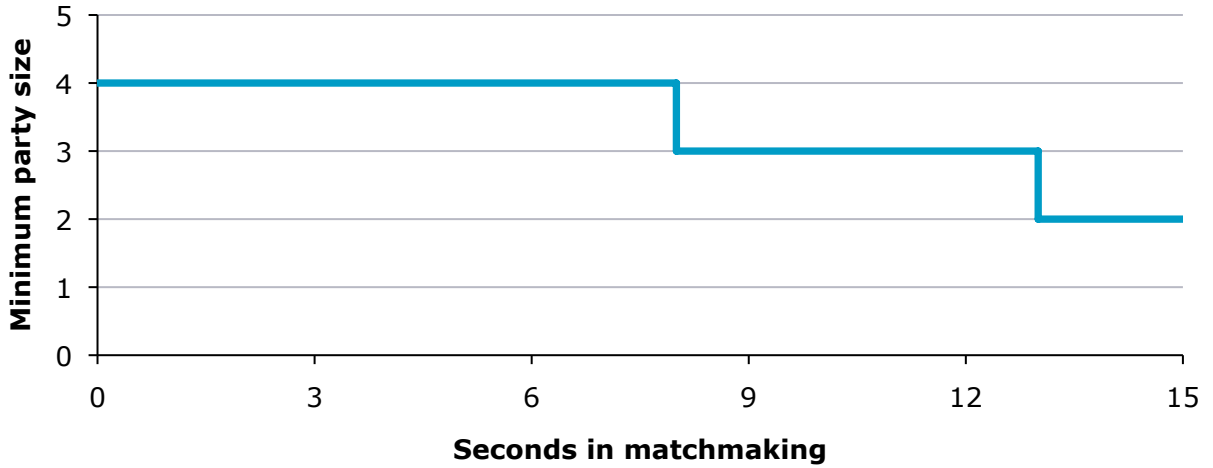
- Each player has qualities
  - Estimated skill
  - Character level
  - Desired party size
  - Ping times to datacenters
  - Time waiting in matchmaking
- Find others with similar qualities
  - Start with narrow tolerances
  - Over time, if can’t find a match, dilate tolerances for qualities

iPhoneGames SUMMIT

ngmoco:)

Matchmaking: Qualities





iPhoneGames SUMMIT

ngmoco:)

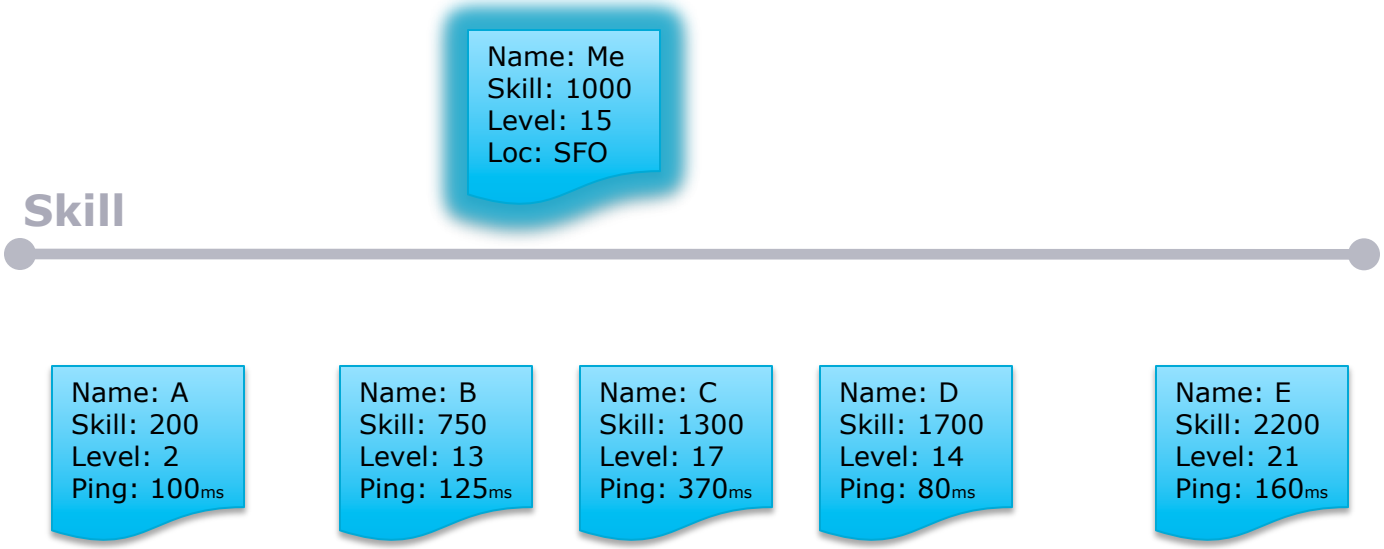
Matchmaking: Algorithm

- Sort players by one quality
  - We choose Estimated Skill
- For each player:
  - Find other candidate players by iterating forward and backwards until outside of skill tolerance
  - Evaluate other quality tolerances for each candidate
  - Form match if enough candidates pass

iPhoneGames SUMMIT

ngmoco:)

Matchmaking: Algorithm



iPhoneGames SUMMIT

ngmoco:)

Matchmaking: Algorithm

Time: 1 second
Skill Tolerance: 500
Level Tolerance: 2

Name: Me
Skill: 1000
Level: 15
Loc: SFO

Skill

Name: A
Skill: 200
Level: 2
Ping: 100ms

Name: B
Skill: 750
Level: 13
Ping: 125ms

Name: C
Skill: 1300
Level: 17
Ping: 370ms

Name: D
Skill: 1700
Level: 14
Ping: 80ms

Name: E
Skill: 2200
Level: 21
Ping: 160ms

Candidate Players

iPhoneGames SUMMIT

ngmoco:)

Matchmaking: Algorithm

Time: 2 seconds
Skill Tolerance: 1000
Level Tolerance: 4

Name: Me
Skill: 1000
Level: 15
Loc: SFO

Skill

Name: A
Skill: 200
Level: 2
Ping: 100ms

Name: B
Skill: 750
Level: 13
Ping: 125ms

Name: C
Skill: 1300
Level: 17
Ping: 370ms

Name: D
Skill: 1700
Level: 14
Ping: 80ms

Name: E
Skill: 2200
Level: 21
Ping: 160ms

Candidate Players

iPhoneGames SUMMIT

ngmoco:)

Matchmaking: Algorithm

Time: 3 seconds
Skill Tolerance: 1500
Level Tolerance: 6

Name: Me
Skill: 1000
Level: 15
Loc: SFO

Skill

Name: A
Skill: 200
Level: 2
Ping: 100ms

Name: B
Skill: 750
Level: 13
Ping: 125ms

Name: C
Skill: 1300
Level: 17
Ping: 370ms

Name: D
Skill: 1700
Level: 14
Ping: 80ms

Name: E
Skill: 2200
Level: 21
Ping: 160ms

Candidate Players

iPhoneGames SUMMIT

ngmoco:)

Matchmaking: Algorithm

Name: Me
Skill: 1000
Level: 15
Loc: SFO

Name: B
Skill: 750
Level: 13
Ping: 125ms

Name: D
Skill: 1700
Level: 14
Ping: 80ms

Name: E
Skill: 2200
Level: 21
Ping: 160ms

iPhoneGames SUMMIT

ngmoco:)

Matchmaking: Skill

- Players start with skill of zero
- After match, update skill estimate based on previous skill estimate and match outcome
- Veteran beating noob
  - veteran += little
  - noob -= little
- Noob beating veteran
  - noob += big
  - veteran -= big

iPhoneGames SUMMIT

ngmoco:)

Matchmaking: Skill

- Math loosely based on Halo 2
  - Early values are positive sum game
  - Middle values are zero sum game
  - Late values are negative sum game

iPhoneGames SUMMIT

ngmoco:)

Matchmaking: Speed

- Need < 10% wait / play ratio
- Status quo
  - ~ 10+ minutes per match
  - ~ 1+ minutes to find opponents
- Eliminate
  - ~ 3 minutes per match
  - ~ 15 seconds to find opponents

iPhoneGames SUMMIT

ngmoco:)

Matchmaking: Capacity

- Can't cluster, must be confident one box can handle load
- Algorithm is worst case $\theta(n^2)$, expected $\theta(n)$
- From unit testing, one box can handle 50k players / second
  - <10% of player time in matchmaking, so supports 500k concurrent users

iPhoneGames SUMMIT

ngmoco:)

Matchmaking: Faults

- Two matchmaking servers
  - Primary, backup
- Clients refresh match request every 4 seconds
- System switches to backup if primary stops responding
- Backup doesn't know how long players had been in matchmaking

iPhoneGames SUMMIT

ngmoco:)

Matchmaking: Wrinkle

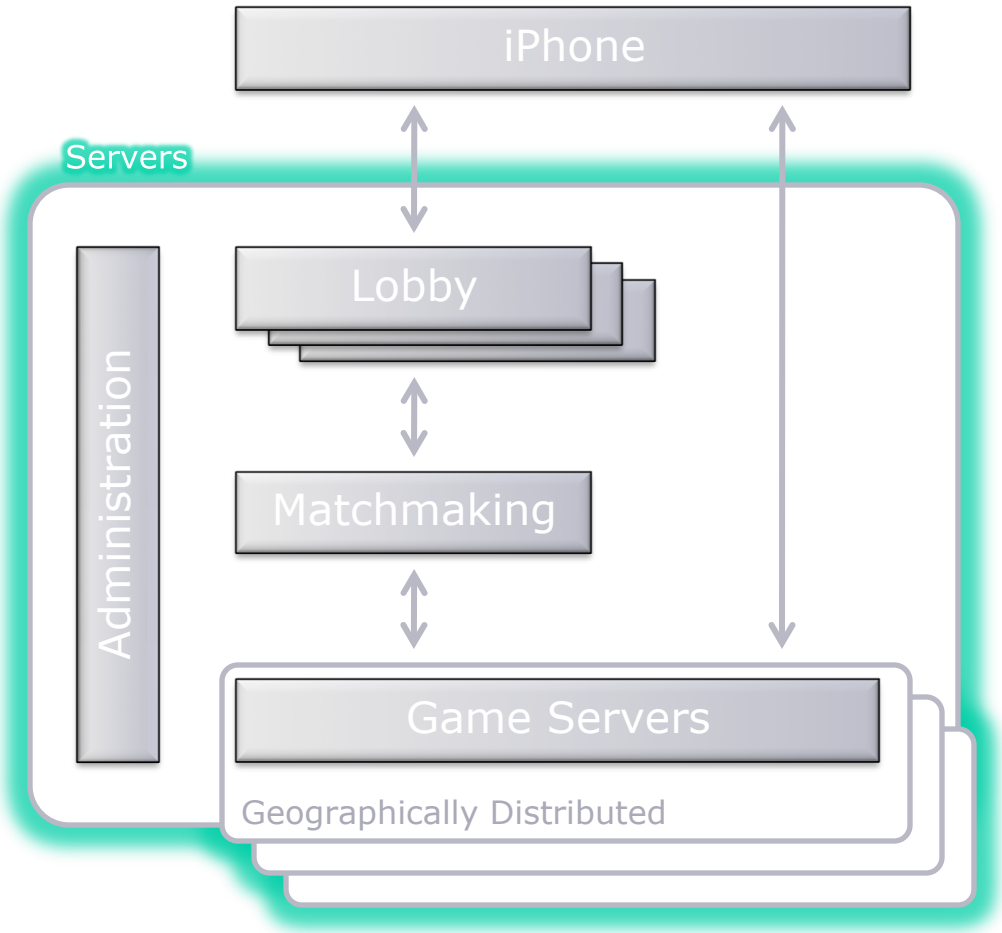
- Initially, character level was ignored by matchmaking
  - Thinking: estimated skill = actual skill + character level
- HUGE outcry from users
- Incorporated character level in 2.0

iPhoneGames SUMMIT

ngmoco:)

Load Testing

Topic 4 of 7



iPhoneGames SUMMIT

ngmoco:)

Load Testing: Why

- Not enough hardware at launch
  - Users won't come back
- Spend all of your money hardware
  - You don't make a sequel

iPhoneGames SUMMIT

ngmoco:)

Load Testing: How

- Build tools to generate load for each component
  - Measure CPU, memory and bandwidth
- Build model to estimate requirements at different usage levels
  - DAUs, Concurrent Users, Session Length
- Re-test often

iPhoneGames SUMMIT

ngmoco:)

Load Testing: XMPP

- Simulate player XMPP actions
  - Login, chat, inventory, etc.
- Reuse actual XMPP client code
- Repurposed game manager hardware
- Ran up to 30K users

iPhoneGames SUMMIT

ngmoco:)

Load Testing: Matchmaking

- Unit test code easily matched 50k players / second on a laptop

iPhoneGames SUMMIT

ngmoco:)

Load Testing: Game Managers Take 1

- Needed to run actual game to generate realistic load
  - Only ran on iPhone
- Built headless version for OS X
- Not enough resources available to stress even one game manager

iPhoneGames SUMMIT

ngmoco:)

Load Testing: Game Managers Take 2

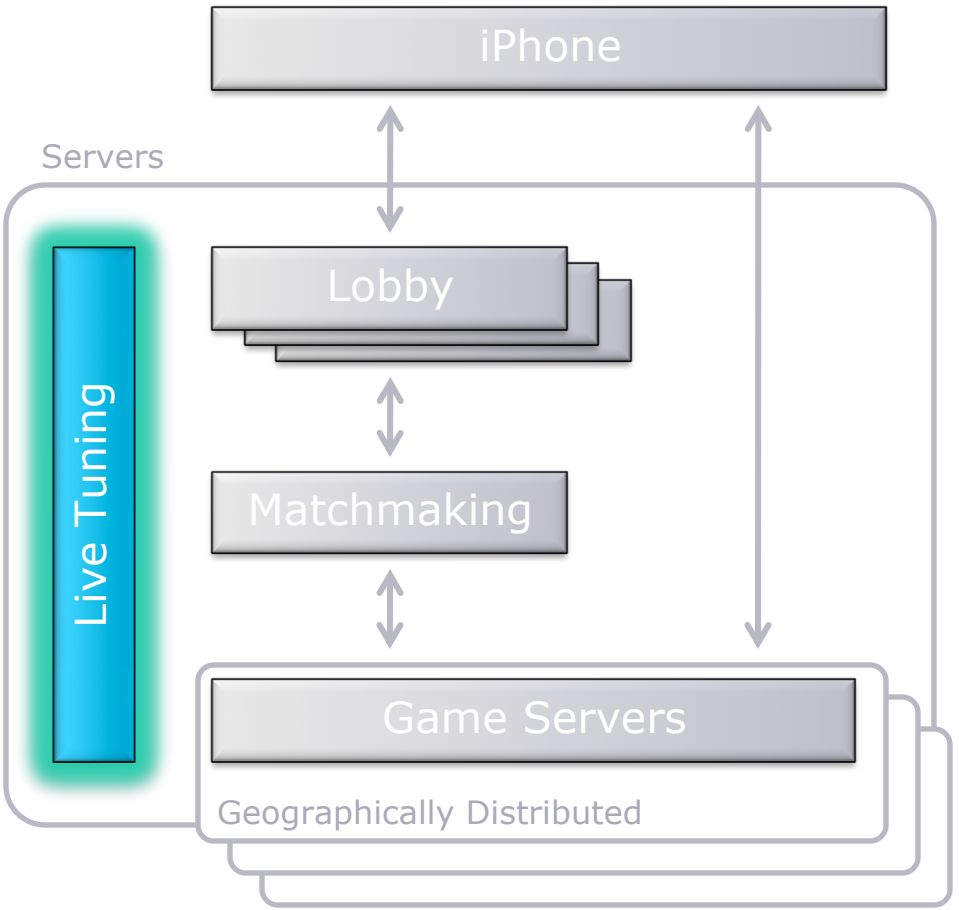
- Measured server load per single game instance
- Created tool to generate matching cpu load
- Continued spawning until OS scheduler fell apart
- Reasonable results but not great
  - Learned more when we went live

iPhoneGames SUMMIT

ngmoco:)

Live Tuning

Topic 5 of 7



iPhoneGames SUMMIT

ngmoco:)

Live Tuning: Overview

- Must be able to tune game experience based on user feedback
  - Weapon and armor strength
  - Items for sale and price in store
  - Regulating stat frequency

iPhoneGames SUMMIT

ngmoco:)

Live Tuning: Plists

- Configuration stored in plist
  - Client downloads latest version to drive UI, modify gameplay
  - Servers consume latest version to configure behavior, validate purchases

iPhoneGames SUMMIT

ngmoco:)

Live Tuning: Problem

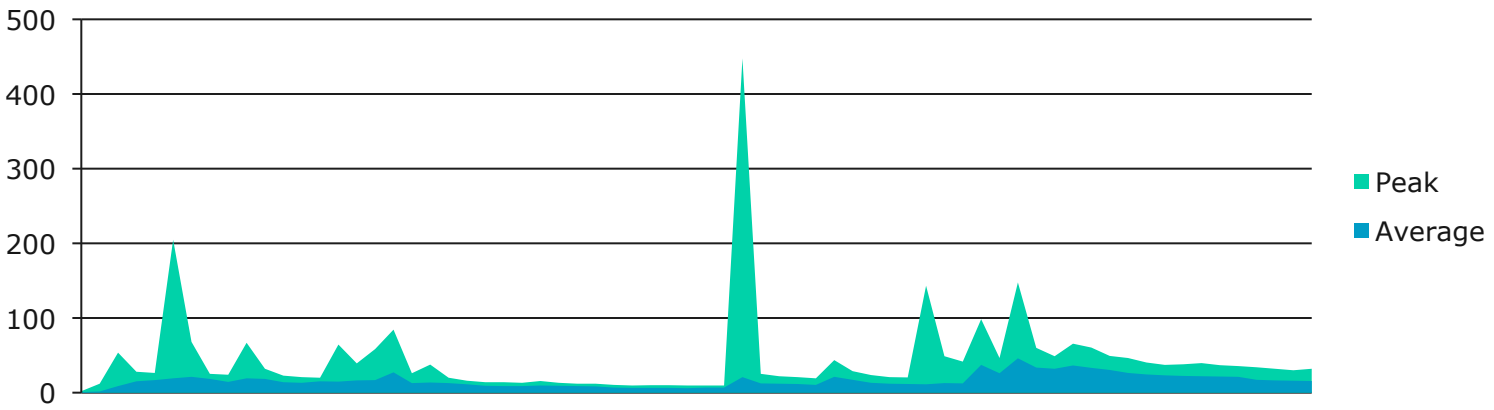
- Initial implementation did not scale
  - XML plist used to make erlang parsing easier
  - Served as base64 encoded XMPP message

iPhoneGames SUMMIT

ngmoco:)

Live Tuning: Problem

- 80KB plist at launch
- Quickly grew past 200KB
- Bandwidth usage spikes when change published
  - 400+Mbps during update

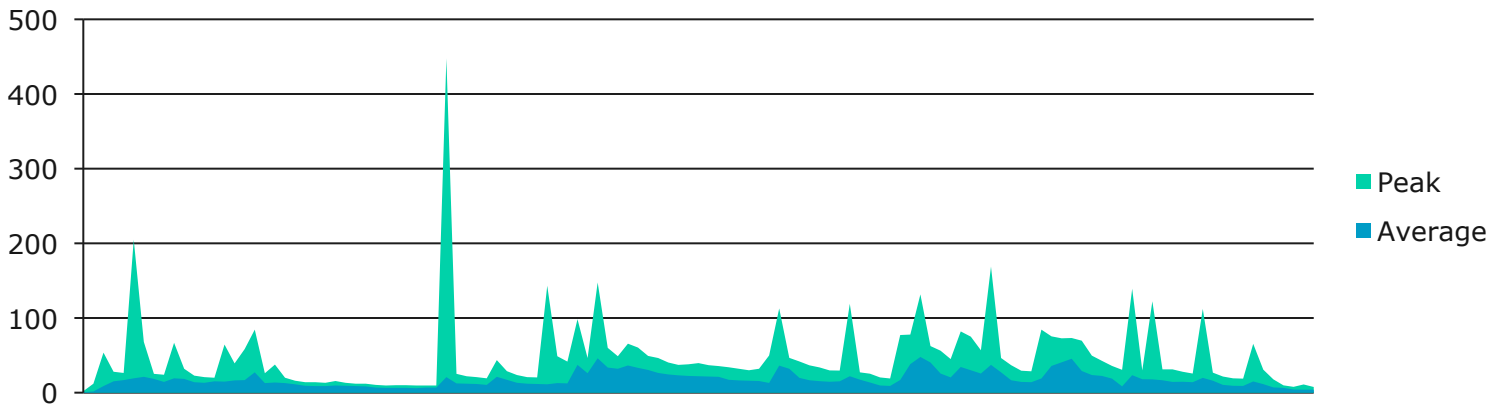


iPhoneGames SUMMIT

ngmoco:)

Live Tuning: Fix

- Eliminate 1.1 added more tuning
  - plist exceeds 400KB
  - New version announced via XMPP
  - Downloaded over gzipped HTTP
    - Bandwidth usage now about 120Mbps

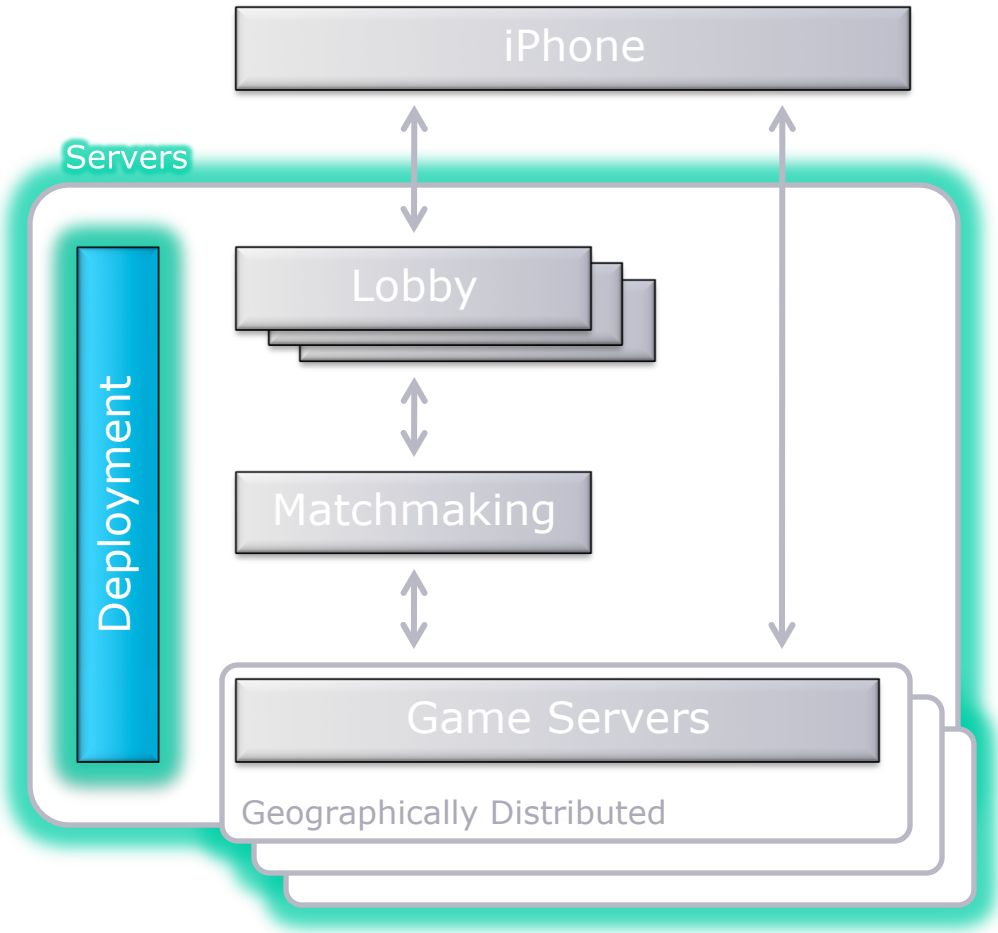


iPhoneGames SUMMIT

ngmoco:)

Deployment

Topic 6 of 7



iPhoneGames SUMMIT

ngmoco:)

Deployment: Overview

- Eliminate uses lots of servers
  - 4 XMPP
  - 2 Matchmaking
  - 8 Game Managers
  - 2 Management
- Production, Staging and Development deployments
- How do we deploy and manage?

iPhoneGames SUMMIT

ngmoco:)

Deployment: Release Management

- Servers run Ubuntu 9.04 64 bit
- Components deployed with apt-get
  - Versioned releases
  - Software dependency tracking
  - Robust upgrade path
- 24 packages for Eliminate

iPhoneGames SUMMIT

ngmoco:)

Deployment: Release Management

- Control script knows about all machines in the cluster
  - Full system upgrades in under 1 minute

```
$ ./control.py upgrade
```

  - Can upgrade subsystems easily

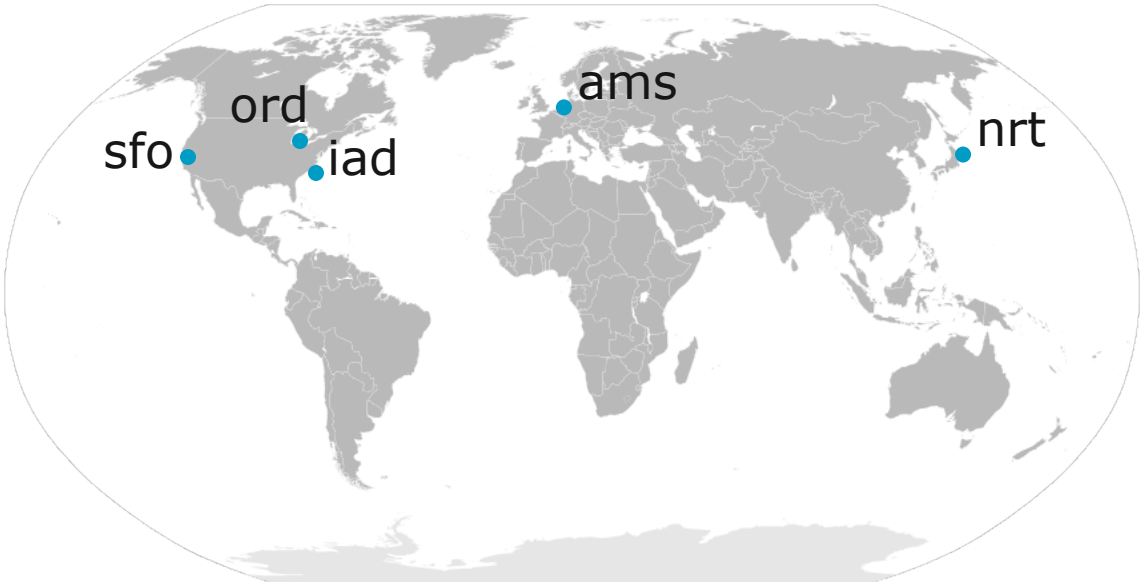
```
$ ./control.py upgrade -c livefire-matchmaking
```

iPhoneGames SUMMIT

ngmoco:)

Deployment: Geography

- XMPP, matchmaking and management servers at ngmoco:)
- Geographically distributed game managers

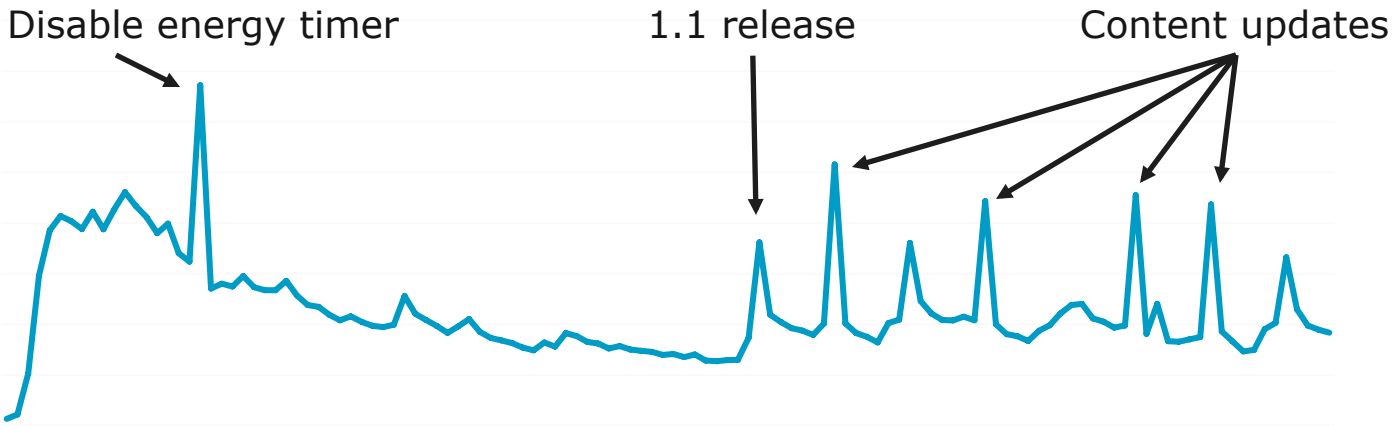


iPhoneGames SUMMIT

ngmoco:)

Deployment: Scaling

- We run hardware to meet our expected daily user load
  - But concurrent user spikes occur
    - Promotions
    - New content creates renewed interest



iPhoneGames SUMMIT

ngmoco:)

Deployment: Scaling

- XMPP deployment can handle 20k concurrent users
  - Can add new capacity in 60 minutes if required
- Matchmaking overbuilt so it never has to scale
  - Match 50K requests/second

iPhoneGames SUMMIT

ngmoco:)

Deployment: Scaling

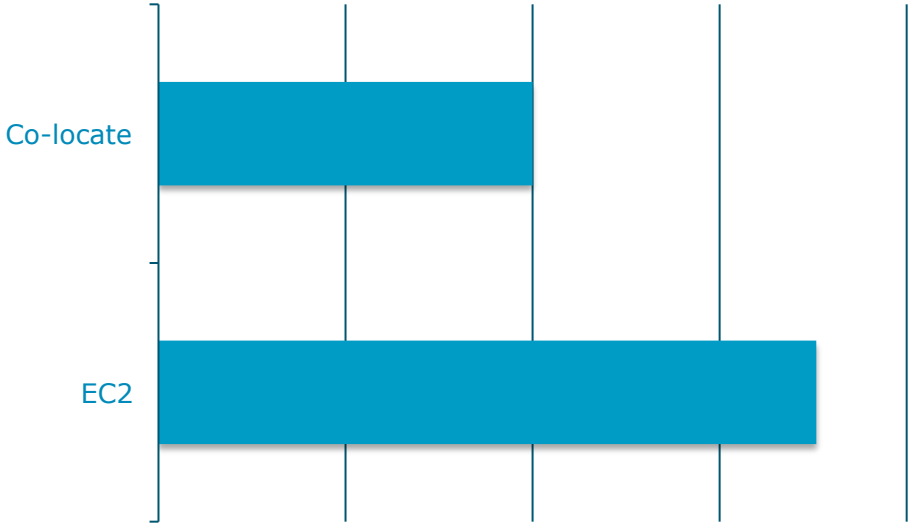
- Amazon EC2 is our safety valve for game managers
- New game managers in 5 minutes
  - High-CPU Extra Large (c1.xlarge)
- EC2 Regions:
  - US-East
  - EU-West

iPhoneGames SUMMIT

ngmoco:)

Deployment: Scaling

- Why not use EC2 for everything?
  - Compute time is cheap
  - Bandwidth is not

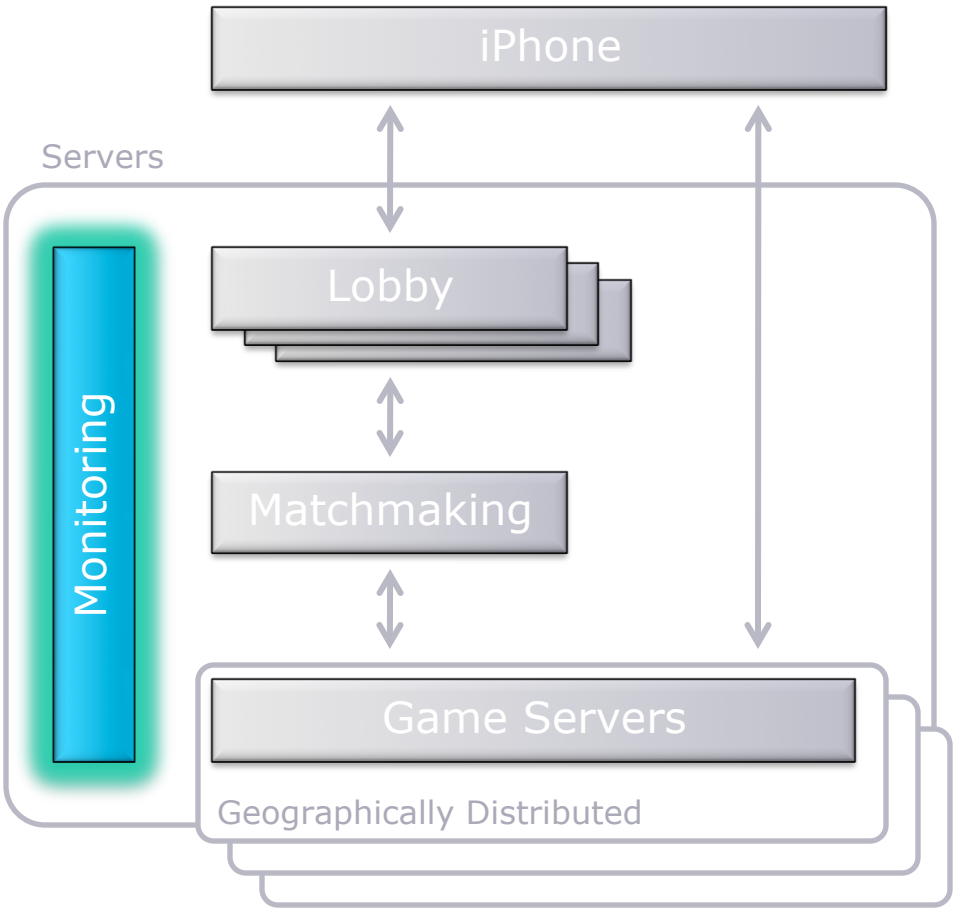


iPhoneGames SUMMIT

ngmoco:)

Monitoring

Topic 7 of 7



iPhoneGames SUMMIT

ngmoco:)

Monitoring: Tools

- Need to track health of the system
- nagios
  - Hardware health checks
  - Text messages on component failure
- munin
  - Visually graphs trends over time
  - Bandwidth
  - CPU
  - Memory

iPhoneGames SUMMIT

ngmoco:)

Monitoring: Custom Tools

- Custom munin plugins
  - Players online
  - People waiting to get in a game
  - Estimated wait time
  - Active games
- Great for long term trends
  - Not good for immediate feedback

iPhoneGames SUMMIT

ngmoco:)

Conclusion

- It took eight months
  - Turns out this is hard
- What we learned that you should know
  - Reuse systems when possible
  - Do load testing early and often
  - Design a system that can scale

iPhoneGames SUMMIT

ngmoco:)

We're Hiring ;)

- Did this sound fun?
- We're looking for exceptional engineers

iPhoneGames SUMMIT

ngmoco:)

Thank You

Questions?