mobile gaming

Victor Bahl

8.13.2012
internet & devices growth (obligatory slide)

Apps are ~$10 Billion market, growing at ~100% per year

Fun Fact: Getting to 1 M users:
AOL: 9 years; Facebook: 9 months; “Draw Something”: 9 days
bandwidth demand!

~ 10 billion mobile devices in 2016¹
(1.4 devices / human)

2011-2016 ~ 18X growth in mobile data traffic²
(~ 10 exabytes / month)

gaming today

Source: Strategy Analytics

TechCrunch

Angry Birds Catapults Itself To One Billion Downloads

Source: Strategy Analytics, AppTRAX 2011

Source: Flurry Analytics, Jan - Feb 2012, n = 64 billion sessions

Source: Strategy Analytics - Appttrax
...but you already knew that 😊

Some things I heard today:

- how game analytics was used to increase dwell time
- how in-the-wild user behavior may be modeled (& used)
- The challenges in getting to MMOG games
- wireless peer-to-peer games
- power management by making use of saliency

All great stuff, let me say a few words about some things I didn’t hear .....
services behind the games

Fun fact: in 2011 ~$12 billion was spent on social/mobile games
in 2015 revenue is projected to be ~24 billion (19% CGR)

Apps that connect to backends receive higher rankings and more downloads because they are likely dynamic with more fresh content and are more social and contextual

- Kinvey Inc., 2012
Xbox LIVE

2.1 billion hours played per month

30% growth year over year

40+ Million Users

35 Countries

176,802,201,383 Gamer Points scored

In US: subscribers spend 84 hours/month

Since 2010, $$ spent advertising has increased 142%
Xbox LIVE is coming to Windows 8

Xbox LIVE provides a comprehensive set of services for Windows 8 game developers that are proven to be useful

https://services.xboxlive.com
XBox SmartGlass

services to enable multi-device gaming experience

HTML5 apps are pushed to your device via a backend service
Xbox LIVE services
first generation of cloud services for games

Games and Apps

Xbox LIVE on Windows Runtime

Xbox LIVE Services
- User Profile & Settings
- Presence
- Avatars
- Friends
- Achievements
- Avatar Awardables
- Leaderboards
- Multiplayer
- Text Messaging
- Roaming Game State
- Beacons
- Real-Time Counting

Mobility & Networking, Microsoft Research
let’s talk about - next gen. services
perennial problems

- network bandwidth
- battery life
- computation latency
- network latency
- fine-grain localization
- sensor accuracy, ....

Take a look at MobiGames’12 Chairs’ welcome message

Several panelist spoke about these as well
compute and energy limitations can destroy mobile gaming experience
energy scarcity: silver bullet seems unlikely

Contrast with

CPU performance improvement during same period: 246x

trade-offs
- Higher voltage batteries (4.35 V vs. 4.2V) – 8% improvement
- Silicon anode adoption (vs. graphite) – 30% improvement

lagged behind
- Fast charging = lower capacity
- Slow charging = higher capacity

Li-Ion Energy Density

Year

Wh/Kg
compute scarcity

Augmented Reality
Too CPU intensive

3D Interactive Gaming
Not on par with desktop counterparts

....limits the gamer’s experience
promising direction: offload computation

remote execution can reduce energy consumption and improve performance
opportunistic use of the cloud

research challenges

• what to offload?
• how to dynamically decide when to offload?
• how to minimize programmer effort?

important for adoption: a simple programming model

• app developer community has varying expertise & skills
  – Cannot require app developers to become experts in distributed systems

“I just want to write game logic on the server – I don’t want to be concerned with scaling, DBs, figuring out how many servers I need, etc.

-- Game Developer-Magazine (Survey of Mobile & Social Technology, May 2012 Issue)
### programming model choices

- **MAUI**: exploits .NET framework to dynamically partitioning & offload method execution [Microsoft, MobiSys’10]
- **Odessa**: creates a data-flow graph to exploit parallelism [USC, MobiSys 2011]
- **CloneCloud**: supports existing applications, but requires tight synchronization between cloud and phone [Intel, EuroSys 2011]
- **Orleans**: a new programming model based on grains [SoC’11]

<table>
<thead>
<tr>
<th></th>
<th>MAUI</th>
<th>CloneCloud</th>
<th>Odessa</th>
<th>Orleans</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remote execution</td>
<td>Methods</td>
<td></td>
<td>Tasks</td>
<td>Grains</td>
</tr>
<tr>
<td>unit</td>
<td>(RMI)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Programming Model

- Dynamic partitioning made simple for the programmer via semi-automatic partitioning
  - Programmer builds app as standalone phone app
  - Programmer adds .NET attributes to indicate "remoteable" methods / classes
- MAUI runtime: partitions (splits) the program at run-time
  - Can optimize for energy-savings, or performance

Why not use a static client/server split?
- Developers need to revisit application structure as devices change
- Failure model: when phone is disconnected, or even intermittently connected, applications don’t work
- The portion of an app that makes sense to offload changes based on the network conn. to the cloud server
Application Partitioning

Dynamic offloading

Client/server split, can be extended to multiple tiers

Mobility & Networking, Microsoft Research
Profiler:
Handles dynamics of devices, program behavior, and environment (Network, Server Load)

Decision Engine:
Partition A Running App

We use an Integer Linear Program (ILP) to optimize for performance, energy, or other metrics...

Example – Maximize:
\[
\sum_{v \in V} (I_v \times E_v) - \sum_{(u,v) \in E} (|I_u - I_v| \times C_{u,v})
\]
energy saved cost of offload

Such that:
\[
\sum_{v \in V} (I_v \times T_v) + \sum_{(u,v) \in E} (|I_u - I_v| \times B_{u,v}) \leq \text{Lat.}
\]
execution time time to offload
and
\[
I_v \leq R_v \quad \text{for all } v \in V
\]
performance benefits

Energy Benefits:
Interactive arcade game w/physics engine:

Energy measurements from hardware power monitor

Arcade game benefits:
• Up to double the frame rate
• Up to 40% energy reduction
How about a service that virtualizes the screen?

A real-time, low-delay cloud server technology for remoting display rendering and user interactions.
offloading in the real-world

existing players:

offload the CPU- and GPU-intensive tasks to a remote render farm, then beam the gameplay as a streaming video

Comments

“System requires a 3-5Mbps connection to work, and can be unforgiving of spotty Internet access speeds. Wi-Fi is still wonky on the PC client”

“If your machine is connected via Wi-Fi, an error message will pop up. System requirements include a dual-core microprocessor and a 5-Mbit/s wired connection.”
thin-client approach

Large bw requirement (5+ Mbps), can’t play disconnected

GPU sitting idle
can we use the mobile GPU to save bandwidth?
collaborative rendering

basic approach

- client computes **low-fidelity** output using mobile GPU
- server ships additional information
- client combines information sources for **high-fidelity** result

intuition of why it would work

- low fidelity game output contains **most** scene information
- missing details are expensive to compute, but missing details are relatively small percent of total info
Doom 3 (low fidelity)
Doom 3 (high fidelity)
Approach 1: delta encoding

Low detail, High FPS

Low game

High game

Video stream

H.264

Delta

High quality, < 1 Mbps

Mobile device

Server

Mobility & Networking, Microsoft Research
Approach 2: i-frame rendering

- Low game
- i-frame merger
- High detail, Low FPS
- i-frame
- H.264
- Low game
- H.264
- raw frames
- video stream
- i-frame filter
- Fraction of thin client’s bw
- Mobile device
- Server

Mobility & Networking, Microsoft Research
bandwidth versus quality

- Thin-client needs 5x bandwidth
- Excellent quality for 1 Mbps or less
- Good quality threshold

SSIM: Structural Similarity Index Metric
code offload allows developers to bypass resource limitations of handheld devices

with dynamic offload, programmers no longer worry about *where* their code runs

- Encourages developers to build applications they would never have considered possible

- but for all this to work we need to be able to scale

“There seriously needs to be an entirely new category of commodity internet infrastructure designed for scaling games “

-- Game Developer-Magazine (Survey of Mobile & Social Technology, May 2012 Issue)
generic offloading should be cloud service, but what else?
how about a service that helps match up gamers

<table>
<thead>
<tr>
<th>type</th>
<th>latency threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>Halo3</td>
<td>≈ 150 ms</td>
</tr>
<tr>
<td>sports</td>
<td>≈ 500 ms</td>
</tr>
<tr>
<td>strategy</td>
<td>≈ 1000 ms</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>location</th>
<th>median latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Los Angeles</td>
<td>≈ 120 ms</td>
</tr>
<tr>
<td>Durham</td>
<td>≈ 175 ms</td>
</tr>
<tr>
<td>Kauai</td>
<td>≈ 180 ms</td>
</tr>
<tr>
<td>Seattle</td>
<td>≈ 180 ms</td>
</tr>
</tbody>
</table>
the match-making cloud service

- matchmaking abstracts away 2 hard problems for game developer
  1. estimating network latency of players
  2. grouping of players into viable games

3G measurement study:
- Phone-to-phone latency stable over 15 minute intervals
- Can share latency profiles between phones using same cell tower
the matchmaking problem

End-to-end Latency Threshold

Connection Latency

Mobility

Clients

Match to satisfy total delay bounds

earch
matchmaking for mobile devices is much harder

- cellular latency is highly variable
- scale is larger (e.g. # of phones vs. # of consoles)
Example game (old)

What are these people doing?
Example: relay service
future: phones that see

who?

what?

where?

Video credits:
Matthai Philipose, MCRC Intel Labs

Mobility & Networking, Microsoft Research
“service store” for game developers

... build world-class cloud services that enable game application developers to easily realize the full potential of their vision

Examples:
- computation offload
- rendezvous: Lookup for relay endpoints
- relay: Phone to phone data transfer
- matchmaking
- social mobile sharing for ad hoc groups
- gesture recognition
- object recognition
- ....

Toolbox of services

sophisticated resource intensive algorithms running in the cloud typically CPU, memory & storage intensive battery and/or bandwidth hungry
latency:
poor latency can be kill certain mobile games
## Game Type vs. Latency Thresholds

<table>
<thead>
<tr>
<th>Game Type</th>
<th>Latency Threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>First-person, Racing</td>
<td>≈ 100 ms</td>
</tr>
<tr>
<td>Sports, Role-playing</td>
<td>≈ 500 ms</td>
</tr>
<tr>
<td>Real-time Strategy</td>
<td>≈ 1000 ms</td>
</tr>
</tbody>
</table>

*Example: Halo is a fast action, low latency game.*

---

*Mobility & Networking, Microsoft Research*
latency: a simple experiment

### iPhone via Wi-Fi: 11 hop

*Wi-Fi -&gt; 209.85.225.99*

1. (10.0.2.1) 8.513 ms 8.223 ms 9.365 ms
2. (141.212.111.1) 0.913 ms 0.606 ms 0.399 ms
3. (192.122.183.41) 11.381 ms 6.054 ms 5.975 ms
4. (192.12.80.69) 7.038 ms 7.353 ms 7.026 ms
5. (198.108.23.1) 11.381 ms 6.054 ms 5.975 ms
7. (216.239.48.154) 9.974 ms 209.85.250.237 10.295 ms 9.405 ms
8. (10.0.2.1) 8.513 ms 8.223 ms 9.365 ms
9. (141.212.111.1) 0.913 ms 0.606 ms 0.399 ms
10. (192.122.183.41) 11.381 ms 6.054 ms 5.975 ms
11. (192.12.80.69) 7.038 ms 7.353 ms 7.026 ms
12. (198.108.23.1) 11.381 ms 6.054 ms 5.975 ms
14. (216.239.48.154) 9.974 ms 209.85.250.237 10.295 ms 9.405 ms
15. (10.0.2.1) 8.513 ms 8.223 ms 9.365 ms
16. (141.212.111.1) 0.913 ms 0.606 ms 0.399 ms
17. (192.122.183.41) 11.381 ms 6.054 ms 5.975 ms
18. (192.12.80.69) 7.038 ms 7.353 ms 7.026 ms
19. (198.108.23.1) 11.381 ms 6.054 ms 5.975 ms
20. (198.110.131.78) 12.715 ms 9.424 ms 9.315 ms
21. (216.239.48.154) 9.974 ms 209.85.250.237 10.295 ms 9.405 ms
22. (10.0.2.1) 8.513 ms 8.223 ms 9.365 ms
23. (141.212.111.1) 0.913 ms 0.606 ms 0.399 ms
24. (192.122.183.41) 11.381 ms 6.054 ms 5.975 ms
25. (192.12.80.69) 7.038 ms 7.353 ms 7.026 ms

### iPhone via 3G: 25 hop

*3G -&gt; 209.85.225.99*

1. 
2. (172.26.248.2) 414.197 ms 698.485 ms 539.776 ms
3. (172.16.0.66) 539.712 ms 809.954 ms 689.547 ms
4. (10.0.2.1) 8.513 ms 8.223 ms 9.365 ms
5. (141.212.111.1) 0.913 ms 0.606 ms 0.399 ms
6. (192.122.183.41) 11.381 ms 6.054 ms 5.975 ms
7. (192.12.80.69) 7.038 ms 7.353 ms 7.026 ms
8. (198.108.23.1) 11.381 ms 6.054 ms 5.975 ms
10. (216.239.48.154) 9.974 ms 209.85.250.237 10.295 ms 9.405 ms
11. (10.0.2.1) 8.513 ms 8.223 ms 9.365 ms
12. (141.212.111.1) 0.913 ms 0.606 ms 0.399 ms
13. (192.122.183.41) 11.381 ms 6.054 ms 5.975 ms
14. (192.12.80.69) 7.038 ms 7.353 ms 7.026 ms
15. (198.108.23.1) 11.381 ms 6.054 ms 5.975 ms
17. (216.239.48.154) 9.974 ms 209.85.250.237 10.295 ms 9.405 ms
18. (10.0.2.1) 8.513 ms 8.223 ms 9.365 ms
19. (141.212.111.1) 0.913 ms 0.606 ms 0.399 ms
20. (192.122.183.41) 11.381 ms 6.054 ms 5.975 ms
21. (192.12.80.69) 7.038 ms 7.353 ms 7.026 ms
22. (198.108.23.1) 11.381 ms 6.054 ms 5.975 ms
23. (198.110.131.78) 12.715 ms 9.424 ms 9.315 ms

![Hop Count Chart](image_url)

traceroute to 209.85.225.99 (one of the server IPs of www.google.com)
RTT comparison for 3G networks

(f) CDF of Ping latency to the first hop
try it out for yourself: TestMyNet

Available on Windows Phone Marketplace
101 Reviews, average review rating of 4.75/5 stars
why? heavyweight architecture
potential solution: Cloudlets
reducing latency: cloudlets

a resource rich infra-structure computing device with high-speed Internet connectivity to the cloud that a mobile device can use to augment its capabilities and enable applications that were previously not possible.
sample deployment scenario

augment Wi-Fi hot spots or femtocells with cloudlets.

advantages
- does not use cellular spectrum
- short round-trip-times between mobile & cloud(let)
- optimal performance

research challenges
- Offload framework
- caching
- security & privacy
... now for something different
what if a cloud does not exist?
ad hoc multiplayer gaming

high resolution real-time continuous location

HLPP: a new class of games & applications
High-Speed, Real-Time, Locational Phone-to-Phone

3D position from audio cues within centimeters
Looking ahead....

- multi-player mobile gaming will continue to be a huge revenue generator
- new devices (e.g. HUD) will tax the networks & infra-structure even more...
- perennial challenges: bandwidth management, latency reduction, energy management, localization, etc. remain
- Prediction: there will be a large number of extensive cloud services for games
what is the killer app you say...

...it’s killing time
Thanks!