Mobile Software Development Framework: Adaptive Mobile Applications

10/23/2012

Y. Richard Yang
Outline

- Admin
- Mobile cloud services
  - Push notification service
  - Track service
  - Storage service
- Adaptive mobile applications
Admin.

- HW3
- Project ideas
Recap: Example Mobile Cloud Services

- Push notification service
- Location based services, e.g.,
  - Track service (supporting location based services)
- Storage services, e.g.,
  - iCloud, Google Drive, Dropbox
- Proxy services, e.g.,
  - Kindle Split Browser
  - Recognition/synthesis services
Example: Push Notification Service

- A single persistent connection between Push Notification Service (PNS) and each mobile device

- Authorization
  - App developer registers project with PNS
  - App on device registers with PNS and forwards registration to PNS
  - App server sends to registered apps on devices

- Scalability, fault tolerance, generality
  - Send only notification, not data
  - For reliability, use soft state, not hard state
**Example: Track Service**

**Creation**
- TC MakeCollection(GroupCriteria criteria, bool removeDuplicates)

**Manipulation**
- TC JoinTrackCollections (TC tCs[], bool removeDuplicates)
- TC SortTracks (TC tC, SortAttribute attr)
- TC TakeTracks(TC tC, int count)
- TC GetSimilarTracks (TC tC, Track refTrack, float simThreshold)
- TC GetPassByTracks (TC tC, Area[] areas)
- TC GetCommonSegments(TC tC, float freqThreshold)

---

Pre-filter tracks  Manipulate tracks  Fetch tracks
API Usage: Ride-Sharing Application

// get user’s most popular track in the morning
TC myTC = MakeCollection(“name = Maya”, [0800 1000], true);
TC myPopTC = SortTracks(myTC, FREQ);
Track track = GetTracks(myPopTC, 0, 1);

// find tracks of all fellow employees
TC msTC = MakeCollection(“name.Employer = MS”, [0800 1000], true);

// pick tracks from the community most similar to user’s popular track
TC similarTC = GetSimilarTracks(msTC, track, 0.8);
Track[] similarTracks = GetTracks(similarTC, 0, 20);

// Verify if each track is frequently traveled by its respective owner
User[] result = FindOwnersOfFrequentTracks(similarTracks);
Outline

- Admin and recap
- Mobile Cloud Services
  - Push notification service
  - Track service
  - Storage service
Storage/Sync Service

- Store content in cloud to be accessible by multiple selected devices

- Multiple deployed services, e.g.,
  - iCloud
  - Google Drive
  - Dropbox
Storage/Sync Service Example: iCloud

- **Backend**
  - Hosted by Windows Azure and Amazon AWS
  - Uses HTTPS connect to servers

- **More details: see iCloud sessions**
Key Design Points for Mobile Storage

- **Data models**
  - what data structure to put on mobile storage

- **Read miss**
  - stalls progress (user has to wait for data)

- **Synchronization/consistency**
  - user may see outdated data
  - user modification may generate conflicts

- **Good topic for investigation/design projects**
  - See backup slides for some pointers
Outline

- Admin and recap
- Adaptive mobile applications
Adaptive Mobile Applications

- An adaptive mobile application adjusts the amounts as well as the locations of resources that it consumes.

- The objective typically is to
  - Improve responsiveness
  - Reduce energy consumption at device
Amazon Silk Split-Browser

- Dynamically split browsing work between local device and Amazon cloud

Adaptive Mobile App: Big Picture

device

On-device app/ sys adaptation

in-net proxy

Device-aware service partition, delivery

service server
Adaptive Mobile Applications: Overview

- This is a large topic
- We will look at an example on each entity
Outline

- Admin and recap
- Adaptive mobile applications
  - Device adaptation
Discussion

- What resource parameters/aspects of a mobile device can be controlled?

http://www.wired.com/gadgetlab/2012/09/iphone5-spec-spec-showdown/
Example: iphone 5

iPhone 5 controls the CPU frequency

http://www.anandtech.com/show/6324/the-iphone-5-performance-preview
Iphone5 A6 Processor

http://www.zdnet.com/inside-apples-a6-processor-7000004786/
CPU Power Model

- The power consumption rate $P$ of a CMOS processor satisfies

\[ P = kCV^2f \]

where $k$ is a constant, $C$ the capacitance of the circuit, $f$ the CPU frequency, and $V$ the voltage.

- When the supply voltage $V$ is lower, charging/discharging time is longer; thus frequency should be lower

\[ f \propto V \]

\[ \Rightarrow P \sim O(V^3) \]
Question:
- Suppose $P \sim V^3$. Do we save energy by reducing freq/vol to 1/3, but finish the same job?
- What if linear: $P \sim V$?
Dynamic Voltage Scaling

- For convex power consumption rate, to minimize power for finishing a job, one may lower the freq/vol to close to 0

- Constraint
  - bound response time (e.g., deadline of multimedia playback)
Example Architecture: GraceOS

- multimedia applications
  - monitoring
  - requirements
  - scheduling

- profiler
  - demand
  - distribution

- scheduler
  - time constraint

- speed adaptor
  - speed scaling

- CPU
Demand Prediction

- Online profiling and estimation: count number of cycles used by each job

Cumulative probability

\[
F(x) = P [X \leq x]
\]

\[
C_{min} = b_0, \quad b_1, \quad b_2, \quad b_{r-1}, \quad b_r = C_{max}
\]
Demand Stability

Demand distribution is stable or changes slowly.
CPU Resource Allocation

How many cycles to allocate to a multimedia job?

Application should meet $\alpha$ percent of deadlines

$\Rightarrow$ each job meets deadline with probability $\alpha$

$\Rightarrow$ allocate $C$ cycles, such that $F(C) = P[X \leq C] \geq \alpha$

![Diagram showing cumulative probability and cycles allocation](image)
How Fast to Run the CPU?

- **Goal:** provides $C_i$ cycles during a time duration of $P_i$

- **Fact:** since power is a convex function of frequency, if a job needs $C$ cycles in a period $P$, then the optimal frequency is $C/P$, namely the lowest constant frequency.
Why Not Uniform Speed?

Intuitively, uniform speed achieves
- minimum energy if use the allocated exactly

However, jobs use cycles statistically
- often complete before using up the allocated
- potential to save more energy

⇒ stochastic DVS
Stochastic DVS

For each job

find speed $S_x$ for each unit allocated cycle $x$

CDF

$F(x) = P [X \leq x]$
Stochastic DVS

For each job

find speed $S_x$ for each allocated cycle $x$

time is $1/S_x$ and energy is $(1 - F(x))S_x^3$

such that

minimize: $\sum_{x=1}^{C_i} (1 - F(x))S_x^2$

subject to: $\sum_{x=1}^{C_i} \frac{1}{S_x} \leq T_i$
Example Speed Schedule

- **Job 1**: 2.5 x 10^6 cycles
- **Job 2**: 1.2 x 10^6 cycles

Observation: speed up the processor with increasing clock cycles.
DVS

context switch
1. Store speed for switched-out
2. New speed for switched-in

speed up within job

switch back
Implementation

Hardware: HP N5470 laptop
- Athlon CPU (300, 500, 600, 700, 800, 1000MHz)
  - round speed schedule to upper bound

- system call
- process control block
- DVS modules
  - PowerNow speed scaling
  - Soft real-time scheduling
- standard Linux scheduler

Extension to Linux kernel 2.4.18
716 lines of C code
Reduces power consumption
However, limited due to few speed options
Odyssey: An Example Client Architecture

- Application indicates resource capabilities in its request to service
- Operating system maintains/monitors available resources
  - no need to have each application re-implement the monitoring
- An application registers a resource descriptor and an upcall event handler with the OS
- OS notifies the application upon detecting resource changes
- Application adjusts requests to the server
Outline

- Admin and recap
- Adaptive mobile applications
  - Device adaptation
  - Server adaptation
Server Adaptation

- It is a good practice to design your app so that
  - Client notifies the server about its capability
  - Server adapts according to client capability
Server has many dimensions to adapt image/video fidelity:

- frame rate (for video)
- image size
- quality of image

Usage: e.g., data acceleration offered by many carriers
Frame Encoding: Block Transform Encoding
Discrete Cosine Transform

\[
F[u,v] = \frac{4C(u)C(v)}{n^2} \sum_{j=0}^{n-1} \sum_{k=0}^{n-1} f(j,k) \cos \left( \frac{(2j+1)u\pi}{2n} \right) \cos \left( \frac{(2k+1)v\pi}{2n} \right)
\]

where

\[
C(w) = \begin{cases} 
\frac{1}{\sqrt{2}} & \text{for } w=0 \\
1 & \text{for } w=1,2,\ldots,n-1 
\end{cases}
\]
Basis Functions of DCT

An image is a superposition of basis functions.

DCT computes the contribution of each basis function.

\[ F[u,v] \]: for the basis function at position \([u, v]\)

Q: If you want to change encoding rate, what may you do?
Frame Encoding: Block Transform Encoding

Run-length Code → Zig-zag → DCT → Quantize → Huffman Code → 011010001011101...
Example: MPEG Block Encoding

original image

\[
\begin{bmatrix}
139 & 144 & 149 & 153 \\
144 & 151 & 153 & 156 \\
150 & 155 & 160 & 163 \\
159 & 161 & 162 & 160 \\
\end{bmatrix}
\]

DCT

\[
\begin{bmatrix}
1260 & -1 & -12 & -5 \\
-23 & -17 & -6 & -3 \\
-11 & -9 & -2 & 2 \\
-7 & -2 & 0 & 1 \\
\end{bmatrix}
\]

Quantize

AC components

\[
\begin{bmatrix}
79 & 0 & -2 & -1 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

zigzag

run-length and Huffman encoding of the stream

\[
10011011100011...\]

coded bitstream < 10 bits (0.55 bits/pixel)
Examples

Uncompressed
(262 KB)

Compressed
(22 KB, 12:1)

Compressed
(6 KB, 43:1)
Outline

- Admin and recap
- Adaptive mobile applications
  - Client adaptation
  - Server adaptation
  - Proxy and job partition
Discussion

- When may a proxy help?

Diagram:
- Device
- In-net proxy
- In-net server

Discussion:

- When may a proxy help?
Example: Round-Trip-Time Effects
Example: Round-Trip-Time Effects
Measured Benefit of Reduced Latency
Example System: MAUI

- Maui Runtime
  - Client Proxy
  - Solver
  - Profiler

- Maui Runtime
  - Server Proxy
  - Solver
  - Profiler

Application

Smartphone

Maui server

RPC
How Does a Programmer Use MAUI?

- Add .NET attributes to indicate “remoteable”

```csharp
[Remoteable]
public List<Move> GetValidMoves(Square s) {
    if (s.IsEmpty()) {
        return new ArrayList<>();
    }
    if (s.Piece.IsEnemyOf(active)) {
        // this piece does not belong to the active side, no moves possible
        return new ArrayList<>();
    }
    // forward the call to the Rule-class
    return rules.GetMoves(s);
}
```
MAUI Run Time

Maui Runtime

Handles Results

Provides runtime information

Intercepts Application Calls

Synchronizes State

Chooses local or remote

Handles Results

RPC

RPC

RPC

RPC

Server Proxy

Profiler

Solver

Maui Controller

Maui Runtime

Maui Runtime

Application

Smartphone

Maui server
MAUI Profiler

- Execution Time
- State size
- CPU Cycles
- Device Profile
- Network Latency
- Network Bandwidth
- Callgraph

- Computational Power Cost
- Computational Delay
- Network Power Cost
- Network Delay

Annotated Callgraph
A sample callgraph

- **A**: Computation energy and delay for execution
  - Energy: 1000 mJ
  - Delay: 10 ms

- **B**: Computation energy and delay for state transfer
  - Energy: 900 mJ
  - Delay: 15 ms

- **C**: Energy and delay for state transfer
  - Energy: 5000 mJ
  - Delay: 3000 ms

- **D**: Energy and delay for state transfer
  - Energy: 15000 mJ
  - Delay: 12000 ms
MAUI Call Graph Partition Example
Work Partition Example Setting

- **User Interface**: 1000 mJ
- **FindMatch**: 900 mJ
- **InitializeFace Recognizer**: 5000 mJ
- **DetectAndExtract Faces**: 15000 mJ
**Work Partition Alg: Local Greedy**

Compare cost of state transfer and local comp. Pick lower cost

- **User Interface**
  - 1000mJ

- **FindMatch**
  - 900 mJ

- **InitializeFace Recognizer**
  - 5000 mJ

- **DetectAndExtract Faces**
  - 15000 mJ

Cheaper to do local
Compare cost of state transfer and local comp. Pick lower cost.
Work Partition Alg: Global Analysis

User Interface

FindMatch

InitializeFace Recognizer

DetectAndExtract Faces

1000mJ

20900mJ

Cheaper to offload
MAUI Implementation

- **Platform**
  - Windows Mobile 6.5
  - .NET Framework 3.5
  - HTC Fuze Smartphone
  - Monsoon power monitor

- **Applications**
  - Chess
  - Face Recognition
  - Arcade Game
  - Voice-based translator
MAUI/FR Energy Performance

Face Recognizer

- Smartphone only
- MAUI (Wi-Fi, 10ms RTT)
- MAUI (Wi-Fi, 25ms RTT)
- MAUI (Wi-Fi, 50ms RTT)
- MAUI (Wi-Fi, 100ms RTT)
- MAUI* (3G, 220ms RTT)

8x improvement on Wi-Fi
4x savings on 3G
MAUI/FR Latency Performance

Face Recognizer

Execution Duration (ms)

- Smartphone only
- MAUI (Wi-Fi, 10ms RTT)
- MAUI (Wi-Fi, 25ms RTT)
- MAUI (Wi-Fi, 50ms RTT)
- MAUI (Wi-Fi, 100ms RTT)
- MAUI* (3G, 220ms RTT)

9x improvement on wifi; 5x on 3G
MAUI/Game Energy Performance

Arcade Game

Solver would decide not to offload

Up to 40% energy savings on Wi-Fi

Energy (Joules)

- Smartphone only
- MAUI (Wi-Fi, 10ms RTT)
- MAUI (Wi-Fi, 25ms RTT)
- MAUI (Wi-Fi, 50ms RTT)
- MAUI (WiFi, 100ms RTT)
- MAUI* (3G, 220ms RTT)
Backup Slides
Mobile Storage

- Read miss
  - explicit user file selection
  - automatic hoarding/prediction, e.g., CODA, SEER

- Synchronization/consistency
  - keep modification logs and develop merge tools, e.g., Bayou
  - efficient file comparisons and merging, e.g., rsync, LBFS
**Synchronization**

- One approach is to keep modifications as logs
  - a user sends the logs to the servers to update

- If the storage of a client is limited, it may not be able to save logs
  - then upon reconnection, the cache manager needs to find the difference between the stored file and its local cached copy
  - same problem exists for the `rsync` tool!

- Question: how to efficiently compare the differences of two remote files (when the network connection is slow)?
Break Files into chunks and transfer only modified chunks

Fixed chunk size does not work well
  why?
Flexible Chunk Size

- Compute hash value of every 48 byte block
  - If the hash value equals to a magic value, it
Bayou: Managing Update Conflicts

- Basic idea: application specific conflict detection and update

- Two mechanisms for automatic conflict detection and resolution
  - dependency check
  - merge procedure

Bayou Write Operation: An Example

```c
Bayou_Write (update, dependency_check, mergeproc) {
    IF (DB_Eval (dependency_check.query) <> dependency_check.expected_result)
        resolved_update = Interpret (mergeproc);
    ELSE
        resolved_update = update;
    DB.Apply (resolved_update);
}
```
Bayou Write Operation: An Example

```python
Bayou_Write(
    update = {insert, Meetings, 12/18/95, 1:30pm, 60min, "Budget Meeting"},
    dependency_check = {
        query = "SELECT key FROM Meetings WHERE day = 12/18/95
        AND start < 2:30pm AND end > 1:30pm",
        expected_result = EMPTY},
    mergeproc = {
        alternates = [{12/18/95, 3:00pm}, {12/19/95, 9:30am}];
        newupdate = {};
        FOREACH a IN alternates {
            # check if there would be a conflict
            IF (NOT EMPTY (SELECT key FROM Meetings WHERE day = a.date
                AND start < a.time + 60min AND end > a.time))
                CONTINUE;
            # no conflict, can schedule meeting at that time
            newupdate = {insert, Meetings, a.date, a.time, 60min, "Budget Meeting"};
            BREAK;
        }
        IF (newupdate = {}) # no alternate is acceptable
            newupdate = {insert, ErrorLog, 12/18/95, 1:30pm, 60min, "Budget Meeting"};
        RETURN newupdate;
    }
)