Mobile Computing with Distributed Processing

1Mamtha T K, M. Tech. student
2Vidya N A, Assistant Professor
1,2Department of Electronics and Communication Engineering, BTLIT, Bangalore
mamthamurugan@yahoo.com
Vidyamohan.123@gmail.com

ABSTRACT
Many mobile applications retrieve content from remote servers via user generated queries. Processing these queries is often needed before the desired content can be identified. Processing the request on the mobile devices can quickly sap the limited battery resources. Conversely, processing user-queries at remote servers can have slow response times due communication latency incurred during transmission of the potentially large query. We evaluate a network-assisted mobile computing scenario where mid-network nodes with “leasing” capabilities are deployed by a service provider. Leasing computation power can reduce battery usage on the mobile devices and improve response times. However, borrowing processing power from mid-network nodes comes at a leasing cost which must be accounted for when making the decision of where processing should occur. We use the dynamic programming framework to solve for the optimal processing policies that suggest the amount of processing to be done at each mid-network node in order to minimize the processing and communication latency and processing costs. The real time application that is being considered in this paper is an android search application which allows users to find sites (eg. apartments) using Global Positioning System (GPS) and ranking methods. These ranking methods consider spatial preference queries based on the qualities of features of the sites in their spatial neighborhood.

Keywords: Dynamic Programming, Feature-based Spatial Queries, Global Positioning System, Mobile Augmented Reality, Network Optimization.

1. INTRODUCTION
The processing and storage capabilities of mobile consumer devices are becoming increasingly powerful. A gamut of new mobile applications has thus emerged for providing a better quality of experience for the end users. A class of such applications commonly referred to as mobile augmented reality, includes one that enable delivery of content in response to the user-generated queries for enhancing user’s experience of the environment. Text to speech conversion and optical character recognition (OCR) based applications for mobile devices follow a similar paradigm. Several interesting usage scenarios thus arise. A user clicks a picture or shoots a video of a desired object—a building, painting in a museum, a CD cover, or a movie poster—through a camera phone. The video or image is then processed and sent over the network to an application server hosting a database of images. The extracted query image is then matched with a suitable entry and the resulting content—object information, location, title song from a CD, or movie trailer—is then streamed back to the user. A number of existing commercial products provide this type of service. The processing of query image or video on the phone often involves computationally demanding processes like pattern recognition, background extraction, feature extraction, and feature matching, which when done often can diminish the battery lifetime of the mobile device.

Alternatively, the raw data could be transmitted to the application server where the processing could be done. However this would increase the bandwidth demand over the network with several users using such an application and competing for spectrum along with voice and data traffic generated by users of the wireless network. The first-hop wireless link between the mobile device and base station is often bandwidth constrained and backhaul connections in mobile networks have high capital and operation expenditures per bit. Several wireless carriers have also reported a staggering increase in data traffic over mobile networks because of unprecedented use of mobile data applications. Backhaul links that carry the traffic from edges to the core using copper, fiber or wireless links are associated with significant cost for the carriers. Moreover, the transmission latency on the uplink will be higher as larger query data is transmitted through the network. Thus there is an inherent tradeoff between battery usage and latency. As mobile devices become more sophisticated with higher resolution image and video capabilities, the query data will continue to grow resulting in more demand for intelligent navigation of this tradeoff.

A user request originates at the Mobile Station (MS). In order to be completed, the request must be
transmitted upstream to a remote Application Server (AS) via a Base Station (BS) and a series of relay nodes. We refer to the node at the first hop as the base station, but emphasize that the links between the BS, relay nodes, and AS may be wired or wireless. If the request processing is entirely done at the MS, the limited battery power can be drained. On the other hand, if the processing is done at the AS, communication latency can be high due to limited bandwidth of the wireless access link and large query size. There are a number of systems which enable distributed processing across multiple nodes. We consider systems with leasing servers which are deployed at mid-network nodes to offer processing capability for the user queries before they reach the AS. Deployment of servers constitutes an instance of server leasing capabilities in the network, where uplink queries requesting content are processed without these uplink data having to travel all the way to backend servers.

We consider how to utilize network assisted computing to alleviate the processing burden on the Mobile Station thereby reducing its battery consumption and extending its operational lifetime. Leasing processing power from mid-network nodes can help lower communication latency because rather than transmitting the entire, large request message over multiple congested links to the Application Station, mid-network processing will reduce the message size. Introducing the ability to lease processing power from mid-network nodes brings in the tradeoff of leasing cost. As discussed, battery consumption and latency can be reduced by leasing processing power. However, if leasing is costly because of scarce processing capability available at the mid-network nodes or if the users are averse to their data being accessed by the leasing servers, then battery usage and latency will increase. Depending on the relative costs between battery usage, latency, and leasing, it may or may not be beneficial to lease. We examine these tradeoffs in this paper. Using the dynamic programming framework, we solve for the optimal processing policies that suggest amount of processing to be done at a node in the network. The optimization objective is to minimize the processing and communication latency and processing costs. We consider cases where the processing times and leasing costs have linear or concave variation with the amount of processing and assess the properties of the optimal processing policy and the core tradeoffs between leasing cost, latency, battery power, and communication over the wireless access link.

2. BACKGROUND WORK
2.1 Mobile Augmented Reality
Augmented reality is a live, direct or indirect, view of a physical, real-world environment whose elements are augmented by computer-generated sensory input such as sound, video, graphics or GPS data. It is related to a more general concept called mediated reality, in which a view of reality is modified possibly even diminished rather than augmented by a computer. As a result, the technology functions by enhancing one’s current perception of reality. With the help of advanced augmented reality technology like adding computer vision and object recognition, the information about the surrounding real world of the user becomes interactive and digitally manipulable. Artificial information about the environment and its objects can be overlaid on the real world. Hardware components for augmented reality are: processor, display, sensors and input devices. Modern mobile computing devices like smart phones and tablet computers contain these elements which often include a camera and MEMS sensors such as accelerometer, GPS, and solid state compass, making them suitable augmented reality platforms.

2.2 Dynamic Programming/Network Optimization
Dynamic programming is a method for solving complex problems by breaking them down into simpler sub problems. The key idea behind dynamic programming is quite simple. In general, to solve a given problem, we need to solve different parts of the problem (sub problems), then combine the solutions of the sub problems to reach an overall solution. Often, many of these sub problems are really the same. The dynamic programming approach seeks to solve each sub problem only once, thus reducing the number of computations: once the solution to a given sub problem has been computed, it is stored the next time the same solution is needed, it is simply looked up. This approach is especially useful when the number of repeating sub problems grows exponentially as a function of the size of the input. The optimization problem is to find the amount of processing to be done at each node in order to minimize the total latency and processing costs. The total cost is given by the processing latency, processing costs and communication latency. The goal is to minimize the expected costs to process the entire request.

2.3 Feature-based Spatial Queries
Consider finding top-k sites (apartments) based on their influence on feature points (e.g. water facility, hospital facility, school facility, bus facility). As an
example, Figure 1 shows a set of sites (white points), a set of features (black points with weights), such that each dotted line links a feature point to its nearest site. The influence of a site $p_i$ is defined by the sum of weights of feature points having $p_i$ as their closest site. For instance, the score of $p_1$ is $0.9 + 0.5 = 1.4$. Similarly, the scores of $p_2$ and $p_3$ are 1.5 and 1.2 respectively. Hence, $p_2$ is returned as the top-1 influential site.

Figure 1. Top-k influential site

2.4 Global Positioning System
The Global Positioning System (GPS) is a space-based satellite navigation system that provides location and time information in all weather conditions, anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites. The system provides critical capabilities to military, civil and commercial users around the world. It is maintained by the United States government and is freely accessible to anyone with a GPS receiver.

A GPS receiver calculates its position by precisely timing the signals sent by GPS satellites high above the Earth. Each satellite continually transmits messages that include

- the time the message was transmitted
- satellite position at time of message transmission

The receiver uses the messages it receives to determine the transit time of each message and computes the distance to each satellite using the speed of light. Each of these distances and satellites' locations define a sphere. The receiver is on the surface of each of these spheres when the distances and the satellites' locations are correct. These distances and satellites' locations are used to compute the location of the receiver using the navigation equations. This location is then displayed, perhaps with a moving map display or latitude and longitude. Latitude ($\phi$) is a geographic coordinate that specifies the north-south position of a point on the Earth's surface. Lines of constant latitude run east-west as circles parallel to the equator. Latitude is an angle which ranges from $0^\circ$ at the Equator to $90^\circ$ (North or South) at the poles. Longitude is a geographic coordinate that specifies the east-west position of a point on the Earth's surface. It is an angular measurement, usually expressed in degrees and denoted by the Greek letter lambda ($\lambda$). Points with the same longitude lie in lines running from the North Pole to the South Pole. Latitude is used together with longitude to specify the precise location of features on the surface of the Earth.

3. PROBLEM DEFINITION

3.1 Existing System

Consider the scenario in Fig 2. A user request originates at the Mobile Station (MS). In order to be completed, the request must be transmitted upstream to a remote Application Server (AS) via a Base Station (BS) and a series of relay nodes. We refer to the node at the first hop as the base station, but emphasize that the links between the BS, relay nodes, and AS may be wired or wireless. If the request processing is entirely done at the MS, the limited battery power can be drained. On the other hand, if the processing is done at the AS, communication latency can be high due to limited bandwidth of the wireless access link and large query size.

3.2 Proposed System

There are a number of systems which enable distributed processing across multiple nodes. We consider systems with leasing servers which are deployed at mid-network nodes to offer processing capability for the user queries before they reach the AS. Deployment of servers constitutes an instance of server leasing capabilities in the network, where uplink queries requesting content are processed without these uplink data having to travel all the way to backend servers.

Figure 3. Proposed System Model
MS thereby reducing its battery consumption and extending its operational lifetime. Leasing processing power from mid-network nodes can help lower communication latency because rather than transmitting the entire, large request message over multiple congested links to the AS, mid-network processing will reduce the message size. As discussed, battery consumption and latency can be reduced by leasing processing power. Using the dynamic programming framework, we solve for the optimal processing policies that suggest amount of processing to be done at a node in the network. The optimization objective is to minimize the processing and communication latency and processing costs.

As shown in Fig 3 the request processing is shared between the Base Station and the Application Server.

4. REAL TIME APPLICATION
ARCHITECTURAL COMPONENTS

The above fig 4 shows the architecture of real time application. It lists the apartments which are geographically located in certain radius using GPS calculations and ranking methods. Ranking is based on the qualities and features of the apartments like water facility, hospital facility, school facility, bus facility etc. It can be seen from the system architecture that the processing is distributed between the Base Station and the Application Server. The apartment filtering is done at the Base Station and listing of the apartments based on the ranking is done at the Application Server. Thus we see that the request processing is shared between the base station and application server.

5. CONCLUSION
The popularity of mobile applications is steadily increasing. Many of these applications require significant computation power, especially in the case of multimedia applications. As the demand, as well as the sophistication and required computation power, for these types of applications increases, battery and communication bandwidth limitations may prevent the use of many of these applications. By “leasing” processing power from mid-network nodes, the battery drain and communication latency may be diminished. Network-Assisted Mobile Computing can help alleviate the processing burden off the Mobile Station without increasing the service latency. Using dynamic programming, we identified the optimal processing policy. We identified some important properties of the optimal policy which can be used to guide future system design. A number of factors must be considered for deployment of such a network-assisted mobile computing system. While there exist technology for collaborative networks, one must consider the amount of processing and data that will be permitted to be shared at mid-network nodes. If high security is required, there may be additional costs required to handle mid-network processing. The design challenges will be application and system dependent. For instance, if the processing only requires transcoding, this can be done on fully encrypted data by simply dropping packets, making mid-network processing simple and secure. However, it is certainly the case that query partitioning will be limited if the data must remain encrypted during the whole query processing.

REFERENCES