Research Paper

Operating System Support for Mobile, Interactive, multi fidelity Applications

Submitted By

Muhammad Asaf Khan

Submitted To

Dr. Sanam Shahla Rizvi

Preston University, Islamabad

4th Januray, 2014

Abstract

This research work focuses the multi-fidelity computation and operating system support for mobile interactive applications. Multi-fidelity computation is a viable approach to meeting the needs of interactive applications on mobile systems.Now a days, mobile interactive applications are getting importance. The minimum use of resources such as battery power, memory used, wireless network and user interaction is also a requirement for the efficient use of these mobile computers.

In this paper we will discussed that what are interactive applications? What is multi-fidelity computation and how it supports in mobile computers? How we can handle the use of scare resource in mobile systems? These applications are difficult to support as they require high resource demand and it is difficult to optimize the minimal usage of these resources.

Table of contents

List of Figures

Chapter1

1.1 Introduction

Today Mobile interactive applications are getting importance and use of these applications increasing day by day. These applications must provide a good interactive environment but it is also required to constrain their resource consumption to the battery power, memory capacity, processing power, physical memory, and CPU usage and network bandwidth. User requirements and resource’s consumption vary with time and space. The system behavior should be such that they can adapt their behavior to track these variations.

During the some last time, mobile interactive applications in its own become a field of study and adapt as subject in mobile computing. Mobile computing introduces many new challenges and opportunities which don’t exist in desktop computing. The most significant challenge in mobile computing is the scarcity of the resources and their critical use. As we know that any applications performance depends on its resources usage. An application is as good as it consumes minimum of its resources especially battery power which is a great challenge in mobile computing, as static host have direct access to the wall power.Although a wireless networkenvironment is more reliable as compared to wired or desktop system as they are portable and easier to move and handle at every place but it is also more turbulent as compared to wired one. In some case it is also unpredictable to check the availability of the resources.

Mobile system or machines becoming smaller, cheaper and more wearable but thioer main use will be to interact with the user in such a way as to provise them with some useful functionality. Augmented reality is the most compelling example of the mobile interactive application. Here we will discuss what the augmented reality is.

1.1.1 Augmented reality

Augmented reality isa technology that superimposes a computer-generated image on a user's view of the real world, thus providing a composite view. Or in other words it is a, view of a physical, real-world environment whose elements are *augmented* (or supplemented) by computer-generated sensory. The example of the AR is the advanced movie animations in which user interact with the objects in real world. These applications used the concept of optics. An object is optically distinguished from its physical existence and user can interact with it as he is directly interacting with physical object. Figure depicts the example of augmented reality application. It is the application of the Samsung which uses android OS.

Fig 1.1: Samsung SARI AR SDK marker less tracker used in the *AR EdiBear* game (Android OS)

Mobility and interactivity both comes combine with a new set of challenges. In mobility we try and limit an application's usage of battery power, memory allocation and usage of CPU etc. Excessive use of these resources affects the application's performance. It increases the latency of computations and shortens the battery lifetime. A system that has more usage of CPU or processing faster requires more battery power. In some case excessive use of resources also increase the cost. Thus we need to maintain a balance between the cost and performance of interactive mobile applications. Thus we need to keep the low latency for the resources, high battery lifetime and low cost. It could keep user distraction at its minimum. This thing is very important for the mobile computing environment where user interacts with the real world in addition to mobile applications and games.

1.2 Multi-fidelity

Mutli-fidelity is the concept of the choose best out of the many. In any environment the range of input corresponds to different output quality levels or in other words fidelity levels.

Let take the example of sorting. If there are M inputs or elements in sorted order in array of size N, we will take the ratio M/N to measure the fidelity. The bets result out of many outputs is the result of the multi-fidelity computation

Chapter 2

2.1 Interactive Applications and Resources

2.1.1 Resources

A resource is an entity that is scarce and useful. There is a finite supply of a resource and the supplier can provide the maximum amount of it and its demand is non-zero i.e. the maximum amount of the resource that the consumer can use in the next time interval. It also comprises of a set of cost functions that determines the cost of the resource as the function of supply and demand. Each cost function relates to the performance and battery lifetime.

It is essential condition to consume the resources in such a way that it increases the performance and decreases the final cost. It should be optimized towards the user or interactive applications. It should not work in such a way that distracts the user attention.

2.1.2 Interactive applications

An interactive application is one in which user directly interact with the system or application and final result are determines by the user. A typical interactive application is in idle states and therefore consumes no resource until user interactive with it in some way and it receives the input after that it starting to use its resources. It executes for the sometime and gives user the output and then again becomes idle until it receives the next input from the user. The execution for some user input is called the operation. When there is an operation it utilizes the resources like CPU, memory, battery lifetime and network bandwidth. Most complex and multi user applications allow the second user request while the previous is being in process. In this case operations overlap and run priority wise and allocate and free resources in an optimized way such that applications don’t get hanged or crashes.

Here we will show the model of an interactive application. Figure shows the process of the application. It is the example of a web browser and its associated operations which are received by the user. The application is waiting for the user interaction. When user interacts with the browser it will receive the input. When user clicks on a link it goes in the network bound fetch request phase and CPU bound render phase. The usage of CPU and network bandwidth determines the cost metrice and interactiveresponse time. If CPU processing fast it will take less time to give the response to the user but it will also takes more battery power so we will optimized the response time i.e. CPU processing and battery lifetime.

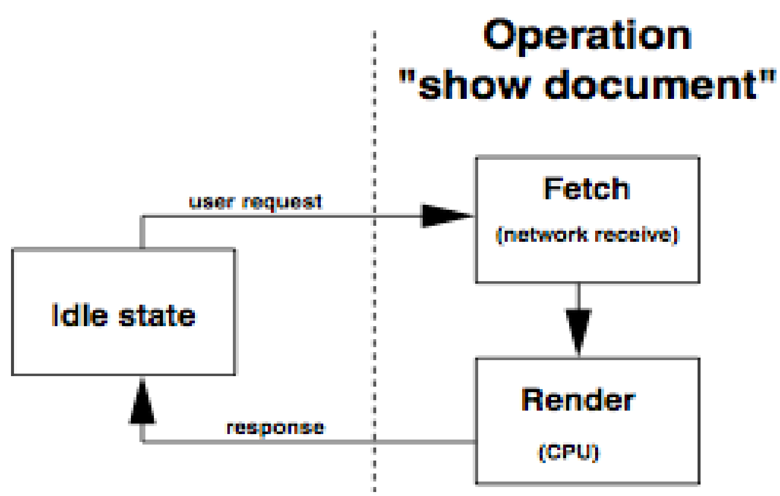

Fig 2.1: An example of interactive application

Interactive applications are described in terms of operations. Let take the example of a speech recognizer. It might have the recognize utterance operation. In augmented reality application a user can generates request or sends inputs by turning its heads or by hand gestures. The applications can also have multi-input or operations and therefore can generated multiple outputs or responses.

Figure shows the resources that are relevant to the mobile interactive systems. It also shows their type whether these resources are time shared or space shared. Demand units and supply units are also describes here. In the figure below the CPU is a measurement of the processor time. The different resources are measure in different demand unit. The same computation might still require different number of cycles for the different processor

Resource	Type	Demand units	Supply units
CPU	Time-shared	cycles	cycles/s
Energy	Exhaustible	Joules	Joules
Network bandwidth (xmit)	Time-shared	bytes	bytes/s
Network bandwidth (recv)	Time-shared	bytes	bytes/s
Physical memory	Space-shared	bytes	bytes
File cache state	Space-shared	(set of objects)	(set of objects)
Disk I/O (read)	Time-shared	bytes	bytes/s
Disk I/O (write)	Time-shared	bytes	bytes/s
Disk space	Space-shared	bytes	bytes

Figure 2.2: Mobile interactive system's resources short description

2.2 Resource dependencies

Resource are depended on one each other e.g. if we increase the processing speed it consumes more battery and shorten the battery life time. So the main issue is to handle these resource dependencies in such a way that it could eliminate all the issues. The dependences of resources might be in terms of the cost and demands for the resources. The memory paging in interactive and other applications results in the disk access, CPU consumes the energy. These dependencies can be viewed in terms of the dependency graph which is shown in figure. Figure shows the dependencies between the resources and the cost matrices in interactive mobile applications

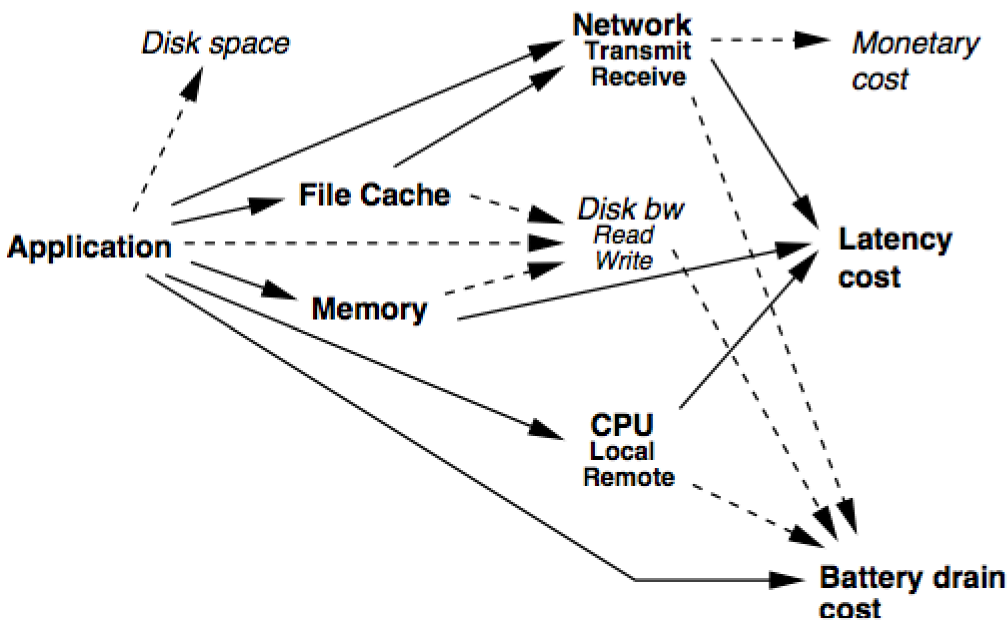

Fig 2.3: Dependencies between resources for mobile interactive application

2.3 Multi-fidelity Computation

Mobile interactive applications must bind the latency and battery drain for the operations and user interactions with the systems. User requires fast response, fast processing and long battery life. Multi-fidelity computes best output from different user inputs and operations and optimizes the resource utilization.

2.3.1 Multi-fidelity algorithms

An algorithm is a well-defined sequence of steps to solve a problem in a managed way. It has a well-defined output specification against inputs. E.g. in case of sorting example, any user for the algorithm preserves all inputs and order them or sort them according to the required criteria. Only algorithm that fulfill these conditions are considered acceptable and we them based on CPU consumption, memory allocation and other resources usage.

Amulti-fidelity algorithm indicates the fidelity level used to generate the result. Multi-fidelity algorithms allow a range of different outputs against same inputs i.e. each has a different fidelity. The essence of a multi-fidelity algorithm is that we can trade this fidelity for resource consumption. The resource consumption affects the battery lifetime, fast response, processing time and latency.

Figure shows the example of an interactive and multi-fidelity application.It shows the multi-fidelity version of the fetch- and-display operation.

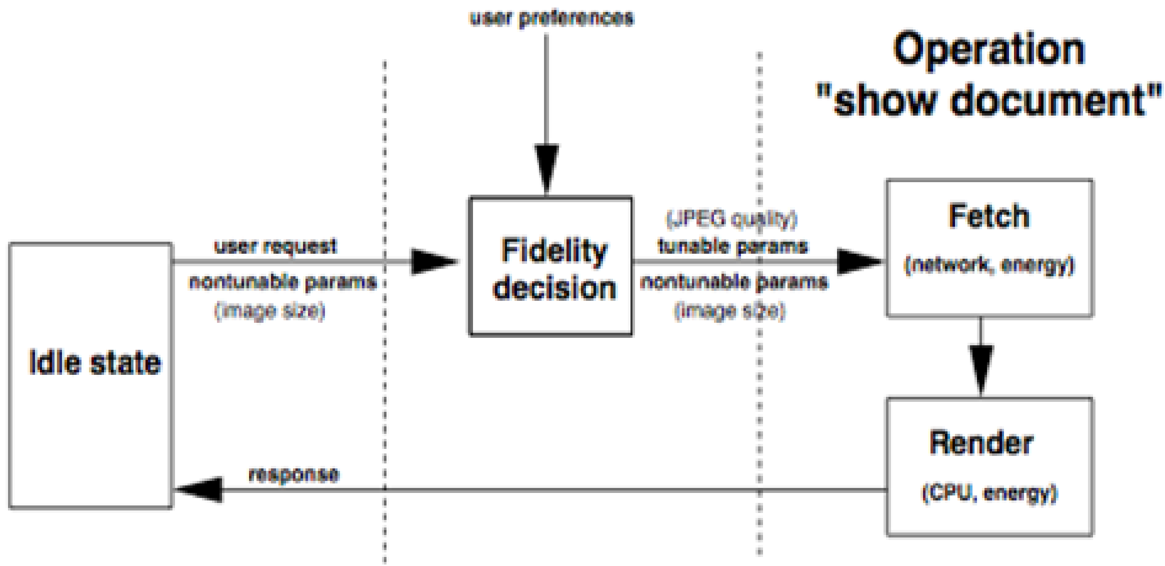

Fig 2.4: Interactive, multi-fidelity operation example

2.4 System support for multi-fidelity mobile interactive application

In this chapter we will focus on the operating system for the mobile interactive application. Application performance is greatly affected by the OS behavior. Most common operating systems used for mobile computing are Google Android, Apple iOS, Symbian OS, Windows 8, Bada OS. Application performance is also affected by the changes in resource allocation, disk space, CPU frequency, and memory size.

Many system are the runtime. The main responsibility of the runtime system is to make fidelity decision and allocation of resources at the beginning of each multi-fidelity operation. The system should monitor the resource demand, battery lifetime for performance debugging purposes and to build predictors for resources requirements in the future or for the next operation.

Figure shows the operating system for the multi-fidelity applications and the mechanism of the resources allocations. Each component is briefly described below.

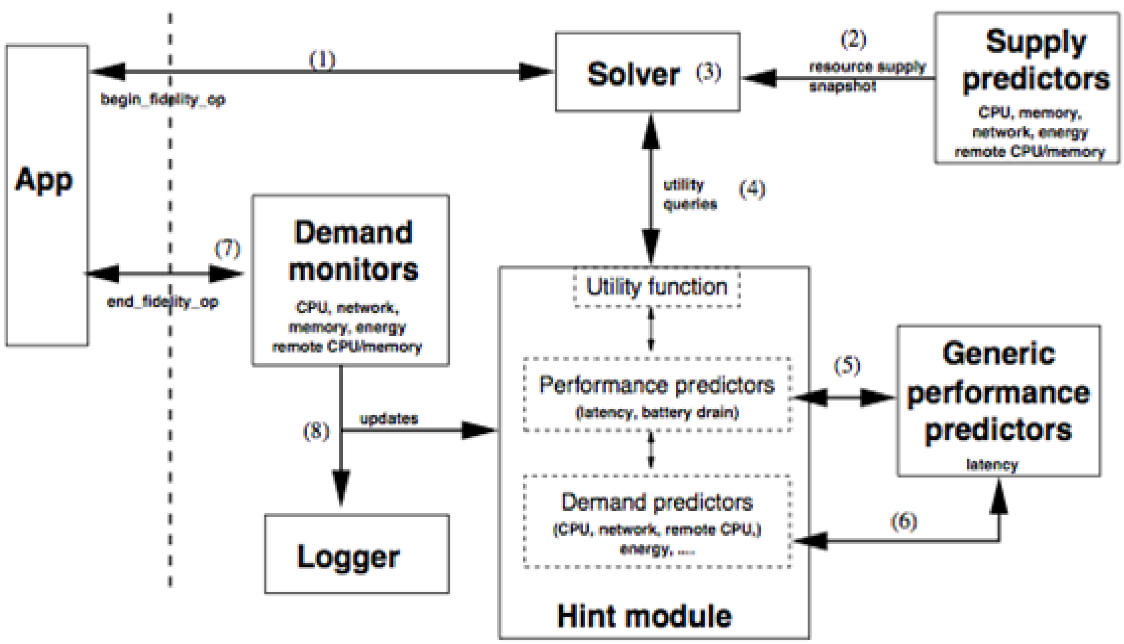

Fig 2.5: Operating system support for interactive, multi-fidelity application

Chapter 3

Conclusion

Interactive mobile computing systems with multi-fidelity applications now become more popular. Through effective adaption applications on mobile systems can overcome challenges such as latency, battery lifetime etc. Multi-fidelity algorithms are reliable solution for adaptive system because they tolerate a range of outcomes. Resources constraints and allocation of resources during processing and execution is the more challenging problem in interactive applications environment.By freeing the system to be more flexible in its responses, multi-fidelity algorithmis a good match for the idiosyncrasies of human users.

References

[1] KlaraNahrstedt, DongyanXu, DuangdaoWichadukul, and Baochun Li. QoS-aware middleware for ubiquitous and heterogeneous environments. *IEEE Communica- tions*, 39(11):140–148, November 2001.

[2] JasonFlinn. *Extending Mobile Computer Battery Life through Energy-Aware Adaptation*. PhD thesis, Carnegie Mellon University, 2001.

[3] Peter A. Dinda. Online prediction of the running time of tasks. In *Proceedings of the 10th IEEE International Symposium on High Performance Distributed Computing (HPDC '01)*, pages 383–394, San Francisco, CA, August 2001. IEEE Computer Society.

[4] D. Narayanan, J. Flinn and M. Satyanarayanan, Using history to im- prove mobile application adaptation, in: Proceedings of the Third IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 2000) , Monterey, CA (December 2000).

[5] Brian D. Noble, M. Satyanarayanan, Giao T. Nguyen, and Randy H. Katz. Trace-based mobile network em- ulation. In *Proceedings of the ACM SIGCOMM Conference : Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM-97)*, volume 27,4 of *Computer Communication Review*, pages 51–62, New York, September 14–18 1997. ACM Press.

[6] M. Satyanarayanan. Mobile Information Access.*IEEE Personal Communications*, 3(1), February 1996.

[7] Jonathan Swartz and Brian Smith. A resolution independent video language.In *Proceedings of ACM Multime- dia '95*, November 1995.

[8] E. Horvitz. Principles of mixed-initiative user interfaces. In *Proceedings of CHI '99, ACM SIGCHI Conference on Human Factors in Computing Systems*, Pittsburgh, PA, May

1999.

[9] David Hull, W. Feng, and Jane W. S. Liu. Operating System Support for Imprecise Computation. In *Flexible Computation in Intelligent Systems: Results, Issues, and Opportunities*, Cambridge, Massachusetts, November 1996.

[10] Michael B. Jones, Paul J. Leach, Richard P. Draves, and Joseph S. Barrera III. Modular real-time resource management in the rialto operating system. In *Proceedings of the Fifth Workshop on Hot Topics in Operating Systems (HotOS-V)*, pages 12–17, Orcas Island, WA, USA, May 1995. IEEE Computer Society.

[11] R. H. Katz. Adaptation and Mobility in Wireless Information Systems.*IEEE Personal Communications*, 1(1), 1996.