Journal of Theoretical and Applied Information Technology
10th September 2015. Vol.79. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

MOBILE CLOUD COMPUTING & MOBILE'S BATTERY EFFICIENCY APPROACHES: A REVIEW

[1]MUSHTAQ ALI, [2]JASNI MOHD ZAIN, [3]MOHD FADLI ZOLKIPLI, [4]GRAN BADSHAH
Department of Computer System & Software Engineering
University Malaysia Pahang
Gambang, Malaysia
E-mail: sunnyswa@hotmail.com

ABSTRACT

Today, millions of users worldwide experience pervasive email and Web access through IPhones, Black Berries, Windows Mobile, and other mobile devices (Laptops, PDAs). After information at the fingertip, energy efficiency of these devices is paramount which makes the devices first choice of communication for the new generation. On the other hand, Mobile Devices, especially smartphones has limited resources such as, storage, network bandwidth, processor's performance and battery life. Moreover multiple sensors and wireless interfaces drain battery swiftly, thus reducing the operational time. Therefore, extending battery life of these devices has become of crucial research importance at hardware and software levels, both alike. This paper provide an overview of the software side by summarizing the background, techniques of battery augmentation, and on-going research on computational offloading to remote servers (Cloud); sending heavy computation to remote servers and receiving the result back on mobile screen. At the end, summary of the previous research work on computational offloading discussed with a critical analysis. Moreover, suggestions and opinions related to energy efficiency of SIDs given, based on the comparative studies.

Keywords: *Mobile Cloud Computing, Mobile Computing, Cloud Computing, Augmentation Techniques, Context-Awareness*

1. INTRODUCTION

Current Smartphone is the evolved shape of PDA and Cell phone which is gradually enhancing capability with each coming year. Smart Internet Devices (SIDs), especially smartphones provide swelling functionalities such as, graphics, high speed processing, storage capacity, and sensing features. Moreover, the explosion of smart mobile applications such as YouTube, Facebook, Twitters and Google maps make Smartphones the first choice of communication tool for the new generation. On the other hand, emerging high computational intensive applications [1], for instance, speech recognizers, natural language translators, online video games, and wearable sensors in the mobile computing environment increased the user's expectations, while they need a high computing power, battery life and storage memory.

Limited power storage of these devices is a core problem; impeding total operational time of the device. In 2005, a survey conducted in 15 countries found that extended battery time is the most important feature than any other feature of smartphones, including; storage and cameras[2]. A similar survey conducted in 2009, by Change Wave Research, found that short battery timing is the most dislike feature of smartphones, including iPhone 3GS(RFF). Likewise, in 2009 Nokia poll discovered that battery life is one of the greatest concerns for users. On the other hand, most of the developers are concerned with the application side and physical enhancement of smartphones only, for example, processing speed and memory capacity[3], while battery conservation is largely ignored. As a result, smartphones/SIDs sometime fails to provide the reliable functionalities.

In addition, the market demand might also be the prime reason, where smartphones are restricted

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

in size and power resources[4]. Thus, two main factors can be considered main causes of mobile's battery issue, a) Inadequate battery and b) The user demand of energy hunger applications. Two possible solutions could be taking into account to overcome the power efficiency problem in smartphones. Firstly, increasing the battery capacity by considering a mechanical way out but no any smartphone's company has brought about a permanent solution, because they are all facing a fundamental dilemma. The microchip technology that support faster performance, better video and audio communication, higher-speed data, video gaming and a more bright and exhaustive screen, are precisely following the Moore's Law (the number of transistors per square inch on integrated circuits had doubled every year since their invention) [5]. The (Li-ion) lithium- ion Smartphone batteries that power them can't preserve power longer, even though Li-ion batteries are the best available power source in mobile batteries[6]. Secondly, the power consumption of mobile devices can be reduced by managing / utilizing local resources in a way to stop excessive draining of mobile batteries.

The excessive penetration of mobile devices in the market drives a strong demand of new services in order to minimize the ambiguity in reliable communication and services availability. It is vital for the current cloud service providers and infrastructure developer to focus on energy issues and challenges. Thus, it is important to study and analyse the current research on power saving in mobile cloud computing. This paper is written to portrait the past and on-going research and their results, related to energy saving of mobile cloud computing. Also the paper scrutinises the limitations, future needs and issues of the current research. In our survey we analyse and compare the proposed solution for power efficiency of mobile devices at software level by studying the research done between 2008 and 2014.

2. BACKGROUND

This section elaborates the concept of Mobile Computing, Cloud Computing and Mobile Cloud Computing. Furthermore, it describes the mechanism of augmenting battery life of Smartphones.

2.1 Mobile Computing

In 1990s, ideas of ubiquitous computing (Mobile Computing) defined technologies that would bring human computer interaction to an absolutely new level. The pervasive nature of Smartphones proposed by Mark Weiser (1991) with the concept of ubiquitous computing as[7]" After the mainframe era , where people used to share a single machine, personal computers where one-to-one human computer interaction took place, the next era will be ubiquitous computing (the era of calm technology) where the technology will disappear". Mark hoped about a world to be created where people can use and interact with computers without thinking about them (psychologically disappeared). Ubiquitous Computing has provided a complete freedom of mental presence to experience the rich number of services using internet.

By the prompt progress, Mobile Computing became one of the powerful trends in the development of IT, commerce and industrial field. It has revolutionized, how people work and deal with their daily lives. In addition, with the development of wireless technology like WiMax, Ad Hoc Network, WIFI, 3G and 4G, users surfing internet much easier not limited by any physical link with a static position or place as before. Thus, mobile devices have been accepted by an increasing number of people as their first choice for communication, working and entertainment in their everyday lives. The transmission of data without the connectivity of any physical link is one of the basic features of Mobile Computing and that's the reason, the number of mobile users increased.

By the top ten strategic technology trends of Gartner (a famous analytical and consulting firm) "In 2013, mobile devices will pass PCs to be the most common web access tools and by 2015, over 80% of handsets in mature markets will be smart phones" [8].

According to the statistics calculated by BI (Business Insider's) Intelligence "By the end of 2013, global smartphone penetration have exploded

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

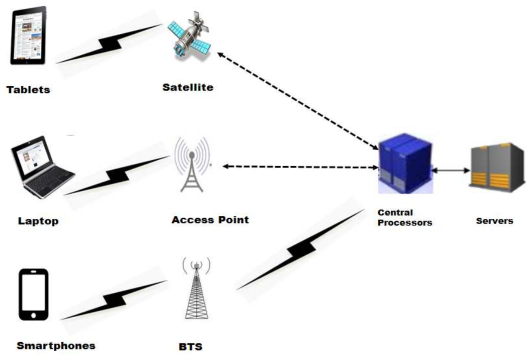

Fig.1. A Framework of Mobile Computing

from 5% of the global population in 2009, to 22%. That's an increase of nearly 1.3 billion smartphones in four years. On average 1.4 billion smartphones, by the end of 2013.

Now, what is Mobile Computing Exactly? By the definition of Wikipedia" mobile computing is a form of human computer interaction, while the computer is expected to be transported during normal usage". Three main components [9] collectively form mobile computing i.e. Hardware, software and communication. Hardware refers to the actual device (smartphone, laptop or their components). Software refers to number of applications running in mobile device, i.e. antivirus, browser and games etc. The communication includes setup of the mobile networks, protocols and the delivery of data in their use. Fig- 1 shows a framework of Mobile Computing. The central processors in the mobile network get user request through Base Station and pass it to the Servers for required Services. The servers release the desired services and the central processors deliver services back to users in response.

2.2 Cloud Computing

In the history of computing, a stepwise evolution can be seen from Mainframe Computing until Cloud Computing (CC). The feature of unlimited resources availability makes CC a superior distributed computing model than grid computing. CC provides the ultimate solution of keeping pace with the development of technology and that's the magic of "Moore's Law"[9]. Since 2007, Cloud Computing grew into a popular phrase and a most significant research topic now-a-days. Due to the different perspective of dozen of developers and organizations, it is difficult to define cloud computing in a distinct way.

Consulting firm Accenture has set a useful, brief definition[10]: "the dynamic provisioning of IT capabilities (hardware, software, or services) from third parties over a network". By NIST (National Institute of Standards & Technology) "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction"[11]. The main objective of cloud computing model is to increase the capabilities of client devices by augmenting the proficiencies of their own resources through accessing cloud infrastructure and software instead of possessing them.

In Cloud computing the services provided by the service providers over internet are commoditized like traditional utilities i.e. water, electricity and telephone etc. Consumers avail the resources on demand fashion and they pay as they use[12]. Amazon Web Services (AWS), Google Apps Engine, Aneka and Microsoft Azure are the examples of public utility computing; deliver at low cost by the Cloud providers (Google, Amazon and Salesforce etc.). AWS allows infrastructure and software as services, which allow users to manage virtualize resources in Cloud datacentre. It decreases the hardware and software cost and efforts of an organization. AWS also allows to utilize S3 (Simple storage services), the unlimited storage capacity for personal data in cloud datacentre by online file storage web services and computation is performed on that data by EC2 (Elastic Cloud Computing) [13]. Amazon S3 is stated to store more than two trillion objects as of April 2013[14]. Google Apps Engine provides a unique powerful application development platform in cloud data centres. The well-known development tools like, Java and Python are used by Apps Engine for the independent development of applications[15]. Windows Azure is an open and extensible cloud computing platform for the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

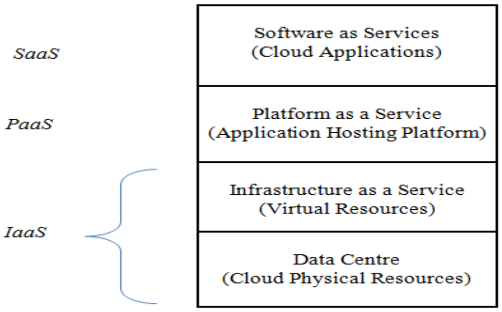

Fig.2 Service-Oriented Layered Architecture of Cloud Computing

development, deployment and operation of applications and services in datacentre. Azure offers a simple, widespread, and a powerful platform for the designing of web applications and services[16]. The service-oriented Cloud Commuting model basically consist of four layers; Data Centres, Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS), as shown in Fig.2.

The cloud physical resources are the hardware resources in datacentres. To access the Physical resources virtual machines are installed. Hypervisor (middleware) used to access the physical resources (Hardware) and is responsible for the placement and management of virtual machines. Both the layers, consists of physical resources and virtual resources fall in the category of Iaas (Infrastructure as a Service). The third layer Platform as a Service (PaaS) comprise of application hosting platform, which provide a cloud programming environments and monitoring tools for example, admission control, QoS negotiation and pricing & billing. The fourth layer, Software as a Service (SaaS) consists of all the cloud applications running on virtual machine instances in a complete secluded form.

2.3 Mobile Cloud Computing

By Eric Schmidt (CEO Google) in 2010 "based on Cloud Computing services development, mobile phones will become increasingly complicated and evolve to a portable super computer[9]" Aepona defines MCC as [17]"a new distributed computing paradigm for mobile applications whereby the storage and the data processing are migrated from the SID,s to resources rich and powerful centralized

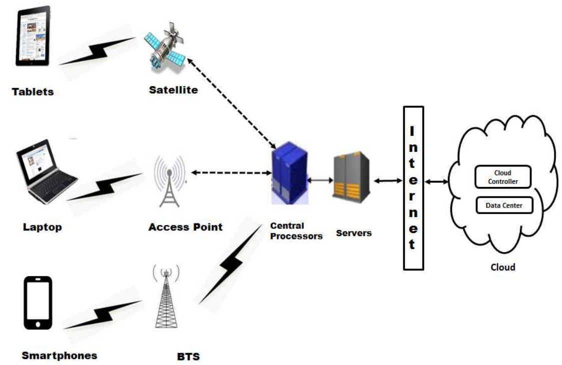

Fig.3 Framework of Mobile Cloud Computing

computing data centres in computational clouds. By Satyanarayanan [18], the mobility feature of devices inherent problems such as, low connectivity, resource scarceness and finite energy etc.

To deal with the low capabilities issues of devices cloud computing turned to be a ruling model which efficiently overcomes the resource scarceness problems by remote computation and utility services. So, by offloading and remote computation techniques, Cloud Computing addresses the inherent issues of mobility using the remote resources, provided by the Service Providers. The big players in the list of service providers are Google, Amazon, Apple, Facebook, Yahoo etc. The Cloud Providers provides such an infrastructure where both, the processing and data storage happens outside of the mobile device and it termed as "Mobile Cloud". Thus, Mobile Cloud Computing is a novel model, encompasses Cloud Computing, Mobile Computing and Networking.

Fig.3 shows a framework of Mobile Cloud Computing. The model composed of mobile computing and Cloud computing bridged by Internet. The mobile devices connect to a network through base stations [19] (i.e. BTS, access points, satellite) that establish and control connections between the networks and mobile devices. The user request is processed and forward by central processors to the servers, providing network services. After that the request are transferred to cloud through internet and the cloud controller process the request and provide the desired services to subscriber.

Journal of Theoretical and Applied Information Technology
10th September 2015. Vol.79. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3. TAXONOMY OF MOBILE'S BATTERY AUGMENTATION TECHNIQUES

Since battery is an element that permits the mobility as a luxury feature in the first place, we obviously need to focus on the improvement of energy usage in order to prevent mobile devices becoming stationary due to low bandwidth and resource-hungry applications. Most of the mobile devices using Lithium-ion batteries [20]. These batteries are the best power source in all the available brands, yet the battery technology shows that the only substitute left to solve the power issue of mobile devices is reducing the power consumption at hardware level and to design more power efficient operating systems and applications. The aim of on-going research, hardware manufacturer and OS designers led to some positive solutions using augmentation techniques at different levels such as Hardware, Wireless technology, Operating System and Applications [20]. Fig.4 shows the taxonomy of smartphone battery augmentation at different level.

3.1 Hardware Level Augmentation

The first step is to enhance the capabilities of mobile local resources i.e. high speed multi-core processors, storage and long lasting batteries[21]. However, the batteries development is unfortunately not progressing at the speed of processors and storage while it is the only un-restorable resource which can't be renewed without of the help of any external resource[22]. Many efforts made since 90's [23]to replenish energy from different sources like human movement, wireless radiation and solar energy, which could not successful enough to enhance energy deficiency, yet are still under research investigation. Similarly large screen and data storage also increases the power consumption due to additional weight and size[24]. Large data storage and retrieval is proven to be application hungry. Hence memory increase donates in faster drainage of battery.

3.2 Software Level Augmentations

Software level augmentation of Smartphone batteries consist three main categories; Operating System, Conserve Resources and Reduce Resources Requirement.

3.2.1 Energy aware operating systems

Two kinds of programs run in a mobile device, Operating System (OS) and applications. Now the question arise, who should be responsible for energy management? Some researchers suggest that application level energy management is best [25], but this approach lacks of a main entity responsible for monitoring and supervising all the

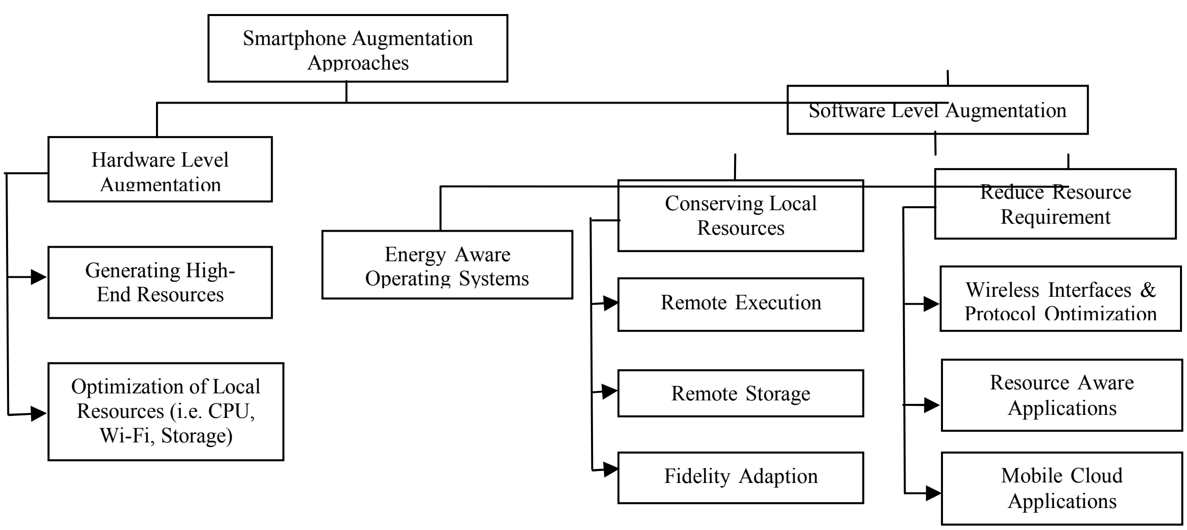

Fig.4 Taxonomy of Smartphones Batteries Augmentation (At Different Levels)

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

resource's consumptions by other applications[20].

On the other hand some researchers prefer OS to monitor and manage energy resources. Considering OS alone responsible for energy management, however this solution face a problem of scalability, that's why some other researchers suggest a hybrid model [20], both the application and operating system should be aware of the resources utilization and supervision. Examples of such Operating systems are Odyssey and Ecosystem.

Normally, the Operating System must know the application's energy demand and the available energy level until the next charging facility. Moreover, new programs, scheduler, models and energy measurement tools should developed in order to reduce energy consumption and support software level energy measurement. Some of the operating system level techniques proposed for power utilization are Hard Disk Management, CPU scheduling, Screen blanking.

3.2.2 Conserving local resources

In 1990's, the best way to conserve the local resources of mobile devices, was to reduce the workload on the local resources. Later on the concept of remote execution, remote storage and fidelity adaptation introduced to conserve local resources.

3.2.2.1 Remote execution

Remote execution is a process involved to transfer the executable codes, control, computational data or any compute intensive application through a network to any local server machine called surrogate that execute the computational task and transfer the results back to the handheld Client.

The remote execution concept presented to divide the load of local resources with the remote stationary devices. Redunko [26], first time in 1998 introduced the idea of remote execution to save labour of the local restricted resources and conserve mobile energy. The same idea introduced by Satyanarayanan [27] in a broader way and proposed the concept of remote execution not only for conserving power, storage, processing efficiency of local devices but also to make possible the execution of compute intensive application which in other words are unable to be process locally. The technique of remote execution is known as process offloading or cyber foraging. The term "cyber foraging" first time introduce by Satyanarayanan [27] for augmenting the computing potential of resource constrained devices by exploiting the potential of rich-computing resource devices available in the local environment. The rich-computing local devices termed as surrogates.

By [28] Aura Project at " Carnegie Mellon University" two techniques are promising in terms of substantial power saving i) Cyber Foraging and ii) Fidelity Adaptation. In the scenario of Cyber Foraging the surrogates can be used by two ways, Data-Staging or by Compute Surrogates. Data Staging is the techniques used with surrogates to get rid of the long latencies (long RTT- Round Trip Time) i.e. instead of mobile device itself to request for web data situated far away from handheld device, and then wait for the distant file to reach, which increases the users response time and more importantly the high latency causes power consumption. In this case surrogates are used to fetch data for the mobile device by data staging. Mobile client send request to surrogates for the distant file, surrogate fetch the file and Client device retrieve the file from nearby surrogates. This reduces the RTT as well as the power consumption. If the data staging technique coupled with some predictable software, then it will let the surrogate to pre-cache all the files which the client device need to use next. This will be more effective in terms of RTT and power consumption. By using the technique "compute surrogate" the client device request to the nearest surrogates for computation on its behalf. In this scenario if the Client device notice that its own battery level is not enough to perform computation locally , it will send compute intensive file to surrogates and will get result on its screen. Research has revealed that remote execution is highly effective in reducing handheld devices energy consumption[29].

3.2.2.2 Remote storage

The outsourcing of data storage at third location, somewhere in the local network or remote clouds, enhances the storage capacity of handheld devices. A number of remote accessible file storage services are available in the cloud servers which

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

extends the storage capabilities by providing off-device storage services. Examples of such services are Amazon S3[14], Drop Box[30] and Google Docs [31] etc. The online storage services (remote storages) not only provide the facility of unlimited storage space but also ensure the safety and reliability of data storage. Thus, the remote storage especially the cloud storage services boost the limited storage capacity of mobile devices and let the mobile users to store and access any kind of data, anywhere in the cloud and can retrieve it from any place by web browser. The writing of data to local storage and retrieval will consume more power as compare to remote storage of such data. Thus it is power efficient to manage big data for remote storage which is not needed in the near future and that can reserve enough battery energy.

3.2.2.3 Fidelity adaptation

The cyber foraging can be effective if the surrogates devices are available but what will happen when no access to any surrogate device in the surroundings? i.e. How the Client devices will divide the load to reduce battery power. Fidelity Adaptation a technique will work in such circumstances. By [32] fidelity can be define as the trade-off between the application's quality and power consumption. For example, a client device is at video call and suddenly the power drop to the lower level, the video automatically shut down while the voice call still continues. Fidelity can also be taking in account in case of releasing CPU load, network bandwidth etc. The run time parameters can be adjust for an application to get lower quality instead of high quality for reservation of power, bandwidth and computational resources. [33] For example a user watching a full colour video from server and it switched automatically to a black and white when the bandwidth drops. Many approaches like [33][34] used both the Cyber foraging and fidelity adaptation to enhance the handheld devices local resources and get better performance.

3.2.2.4 Reduce local resource requirement

Tragically, a lot of software developers have imperfect knowledge with energy-constrained handy embedded systems i.e. Smartphone, PDA etc. As a result, many mobile applications are unreasonably power-hungry. To augment the capabilities of mobile's resources along with manufacturing the high-end hardware devices; a parallel development of resource-efficient application plays a vital role. This approach focuses the design phase of software development to form energy efficient applications.

3.2.2.5 Wireless interfaces & protocol optimization

Among the other components of mobile devices contributing in power consumption, the impact of wireless interface is more than any other component. By [35] 10% of the overall power consumption in laptops is due to wireless interface while in case of Smartphone devices this ratio reached to 50 % of the overall consumption. Mobile devices are now-a-days equipped with several air interfaces such as GSM, Wi-Fi, 3G and 4G, it is vital to manage these interfaces either manually or by resource aware applications to stop the excessive power draining. Although all these wireless interfaces provides a flexible choice for communication/energy saving yet the existing communication protocols[36] i.e. UDP and TCP cannot take any advantages. For example a communication which is established with 3G connection will last till the session end or either the interface unavailable anymore, even though the device is close to a better Interface in terms of speed, energy saving and communication.

New technologies [37] in cellular networks introduced such as 4G LTE to reduce the cost per transmission as compare to the previous standards. In all the available interfaces of a mobile devices Wi-Fi is more power efficient [38] if it stays in continues data transmission for larger data. The only time Wi-Fi drain battery is the idle time or scanning AP (Access Point) to connect. So common solution has been suggested by researchers like Bertozzi [39] explores transport protocols optimization for IEEE 802.11 networks to reduce the energy depletion of IEEE 802.11 standards with very little overhead. His work based on the transport protocol, which tackles flow control of data to regulate the network traffic. It plays an important role to predict workload of the network interface. Further shows that, by tuning parameters in the protocol it forms the activity

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

profile of the network interface and making it more energy efficient.

3.2.2.6 Resource aware applications

To protect Smartphones from energy depletion it is vital to understand the need of power consumption of hardware components and of software installed. Many software developers have limited understanding with energy-constrained portable embedded systems i.e. smartphones, PDAs etc. As a result, quite a lot of smartphone applications are unnecessarily power-hungry[40]. So better understanding of the power consumption of individual mobile components (CPU, Memory, Wireless Interface and Screen) cut a significant amount of energy and leads us to develop a good energy aware system. For instance, the available air interfaces i.e. 2G and 3G, a resource aware application is needed to exploit 2G for voice communication and 3G for FTP services because of their different energy consumption needs[24].

The memory and compute intensive applications are also considered to be power hungry. The compute intensive application engages CPU for long time to process complicated data that is directly suffering mobile battery. Both the increase in compute intensive applications and increasing memory size of mobile devices lead us to more power consumption[24].

3.2.2.7 Cloud-mobile applications

Cloud mobile apps are identical to Web-based applications. The main similarity in both the applications is they run on external servers instead of client device itself. They require a browser on the client device to access them[41]. Moreover, they both are designed for different operating systems and multiple mobile devices unlike native applications, which are design for specific operating system and single device model only.

Native mobile applications are restricted by the battery and processing of mobile devices, which ultimately upset the speedy progress of these devices. The concept of cloud computing bridge this gap by offering cloud-mobile applications to have capabilities of connecting cloud servers for processing and remote storage. The new concept of cloud and mobile agreement generated this new era of rich cloud-mobile applications which are intended to curtail smartphones' resource consumption by utilizing rich cloud resources with no quality changing. Therefore, the development of cloud-mobile apps accelerates code execution by offloading [42] compute intensive data to cloud and thus decreases the overall execution time without using the mobile resources.

Efforts made by numerous researchers for designing perfect Cloud-mobile application to leverages the Cloud resources for mobile devices. Yan Lu et al [43] developed an architecture for rendering mobile screen to the cloud environment, where the remote code execution and online data take place.

Certainly, by the migration of resource hungry and interactive portion of the screen for execution in cloud will decrease the power consumption due to minimizing labour of local CPU, GPU etc. A similar effort made by Byung-gon [44] the CloneCloud service which is uses a smart phone's internet connection to communicate with a full image (clone) of itself that exists on remote servers in the cloud. In case of compute intensive processing the mobile device needs to offload data to the server which hold Clone of device, to process and display result on the Client device's screen. Fig.5 shows a model of CloneCloud.

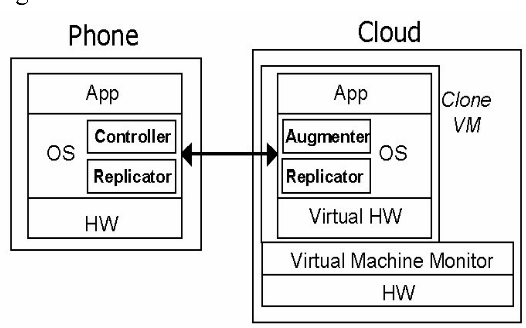

Fig-5 CloneCloud Model [44]

4. HOW COMPUTATIONAL OFFLOADING SAVE ENERGY

The cloud computing is different from the existing models due to a fundamental feature "virtualization". [42] Virtualization allows cloud vendors to provide services of running arbitrary applications of various customers on virtual machines. Thus, a cloud vendor provides computing cycles and customers use these computing cycles to reduce computation of mobile

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

devices. Computational offloading approach facing two challenges listed below.

What will be the optimum condition for computational offloading from client device to cloud?

What can be the factors that are needed to address before starting computational offloading?

Now the question arises, Is Computation Offloading save energy? Karthik Kumar [45] has addressed these issue by analysing the energy consumption using computational offloading. They proposed a formula (1) and by using this formula they derived the amount of energy which can be saving during offloading process.

$$PC \times \frac{C}{M} - Pi \times \frac{C}{S} - Ptr \times \frac{D}{B} \qquad (1)$$

Where, C - is the number of instruction to be offloaded,

S & M - are the speeds instruction /second of Server and mobile device respectively.

PC - mobile power consumption (watts),

Pi -mobile idle power consumption (watts)

Ptr - mobile device power consumption during transmission.

D -data in bytes to be exchanged.

B -network bandwidth.

If, the server speed considered F times faster than mobile speed then

$$S = F \times M \qquad (2)$$

And by substituting eq. (2) in (1), the formula can be rewrite as,

$$\frac{C}{M} \times \left(PC - \frac{Pi}{F}\right) - Ptr \times \frac{D}{B} \qquad (3)$$

Here, the values M, Pi, Pc, and Ptr are constant, and if eq. (3) provides a positive number then offloading will reduce power consumption of mobile device. The formula will provide positive number if $\frac{D}{B}$ is sufficiently small (i.e. B sufficiently large) and F is sufficiently large. In other words, if bandwidth and server speed are sufficiently large then offloading will reduce power consumption. The relationship between B, D and C is important to predict whether to offload task or not i.e. for large Computation C if communication data D is smaller and bandwidth B is enough large then offloading will be beneficial otherwise for small C and low bandwidth B, it is useful to avoid offloading and process data locally. The relationship of B, D and C can be seen in the Fig.6.

On the basis of few elementary metrics i.e. user QoS requirement, availability of local resources, SLA and network availability S. Abolfazli [23] described a sample flow of mobile application execution. The flow chart depicts four process of mobile application to be executed. If the device is capable of running task it will be executed locally otherwise will be offloaded to remote servers. If all the available option go false then application execution request will be killed. Fig.7 illustrates the flow chart of mobile application execution.

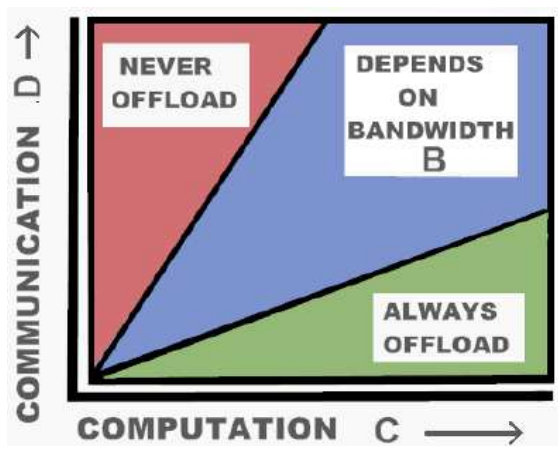

Fig-6 Offloading decision on the values C,D and B. [42]

Journal of Theoretical and Applied Information Technology
10th September 2015. Vol.79. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

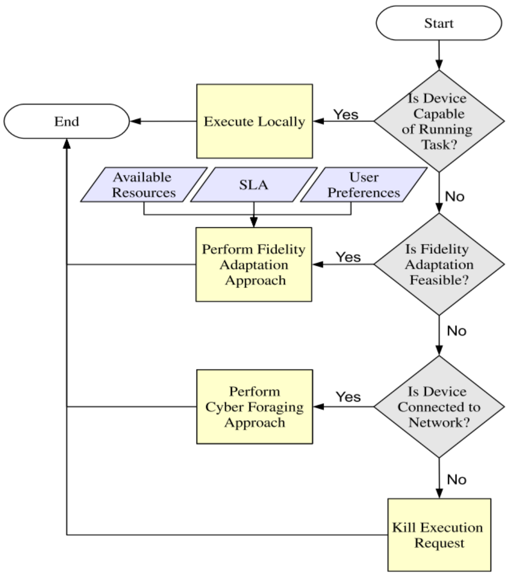

Fig.7 A Sample Flow of Mob Application Execution[23]

The issues related to offloading are the efficient and dynamic offloading [19] under changing environment such as , user movements or communication issues will affect the bandwidth, then what strategy should adopt to offload applications? In case of static offloading the application will be offloaded before runtime to server regardless of environmental changes and user context. By Rudenko [26] static offloading is not always energy efficient approach i.e., if the size of compiling code is small enough then offloading will consume more energy than that of energy consumed in local processing. For instance if the size of compilation codes is 500KB, then offloading use about 5% of mobile's battery for its offloading to server while local processing for the same size of code consumes approx. 10% of the battery for computation. In this case, offloading can save a significant amount of energy (50%). Conversely, if the size of codes is 250KB, then the efficiency reduces up to 30%. Thus, if the size of codes to be executed is small, the offloading will consume more battery than that of local execution.

It is a serious problem for mobile devices to decide whether to offload or not and which portions of the application's codes needed to offload for improving energy efficiency. Moreover, diverse wireless access technologies require different amount of energy and also support dissimilar data transfer rates. These factors are needed to be taken into account.

To overcome these issues the dynamic offloading techniques used. It will decide runtime whether to offload and which portion of the application to offload based on energy consumption as suggested by [42]. The optimal partitioning of programme takes place on the basis of trade-off between computation and communication costs. Several solutions have been proposed for the optimal application partitioning. By [46] "If a device becomes resource constrained at run time and believes it can beneficially use nearby resources, it automatically and transparently offloads part of the service to them". They proposed a dynamic shared distributed environment where in case of remote server unviability can share the application portions to another (surrogate) server.

Runtime dynamic offloading contains the issues[1] related to dynamic application profiling and solver on mobile devices, partitioning of runtime application, migration of compute intensive components to remote servers and constant synchronization for the whole duration in runtime execution platform. Fig.8 depicts the general flow sequence in dynamic offloading of compute intensive components.

5. RELATED WORK

For the last few decades numerous researcher attempted different techniques to delegate the resource-intensive parts of applications to remote servers for minimizing the load of local resources. In this regard two [47] approaches are commonly used. 1) Rely on programmers- they specify how to partition a programme. Which part of a programme need to be remote and how to adjust the programme partitioning scheme with the frequently changing network environment? [48][49]. This approach leads to savings enough energy because it is fine-grained. Now application can be offloaded in sub parts only if the remote execution is beneficial in terms of energy, processing and storage. 2) The migration of entire process [50]or OS (Virtual Machines) [44]to the cloud instead of subparts. This approach exclude burden on programmers i.e.

Journal of Theoretical and Applied Information Technology
10th September 2015. Vol.79. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

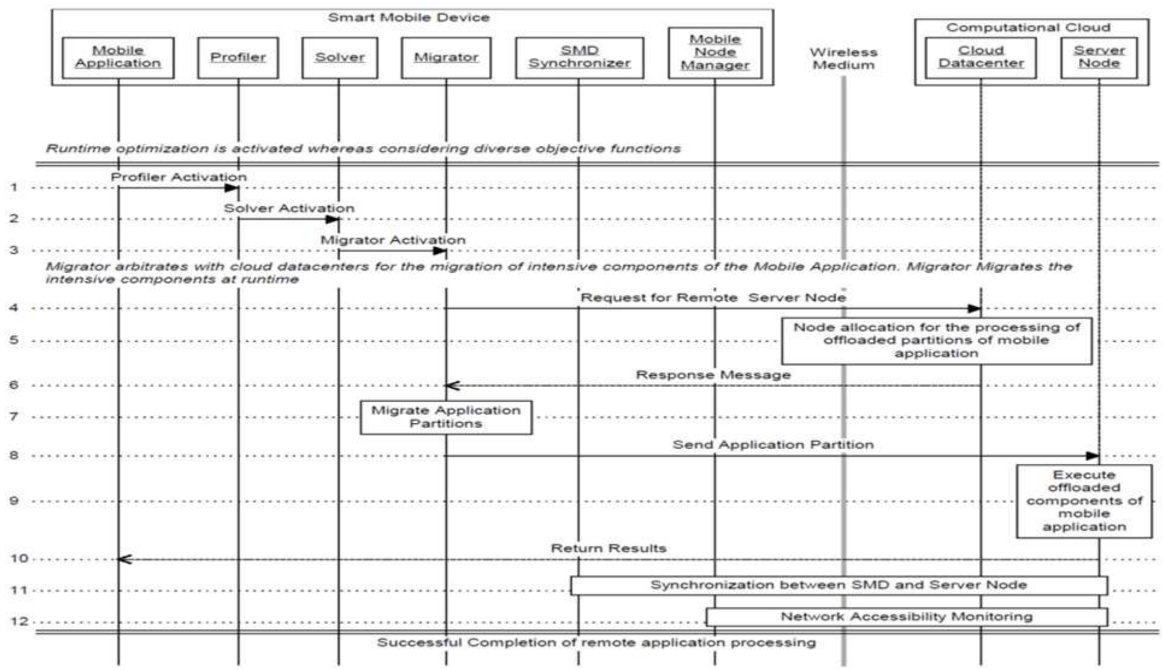

Fig. 8 Sequence Diagram for Dynamic Offloading [1]

application does not need to be partition and the entire process or system is automatically loaded to the remote servers. A review of some typical research projects and the comparison & contrast are presented below.

CloneCloud is introduced in 2011 by B. Chun[44]. The 'Clone' is an image of a mobile device residing on a virtual machine in cloud. In contrast of smartphones a 'clone' is in rich hardware, networks and software environment close to energy efficient resources which is more suitable for the execution of complicated task. The main method used is virtual machine migration, to offload application's execution blocks from resource constrained mobile devices to rich resources pool Cloud flawlessly and partly. Clonecloud system fully or partly offloads the smartphone based execution to a dispersed environment. The CloneCloud model shown in Fig. 9.

Each smartphone's task is divided into 5 different execution blocks. The blocks are divided on resource intensive basis. The block which are more power hungry are then passes to cloud for processing. The energy intensive blocks are colour green in the diagram shown Fig-10. Once the execution of these blocks completed the output then passed from CloneCloud to the smartphone.

A face tracking application is taken as a test by Byung with and without CloneCloud and the result shows that 1 second is taken by CloneCloud processing the task while 100 seconds taken for processing the same in smartphone. Another advantage of CloneCloud System is the reservation of battery life, as Smartphone do not need to process complicated tasks. The disadvantage of this approach is the handover delay and bandwidth limitation. As we know the speed of data transmission is not consistent, therefore the CloneCloud System will be not responsive whenever the user moves to a signal blind area.

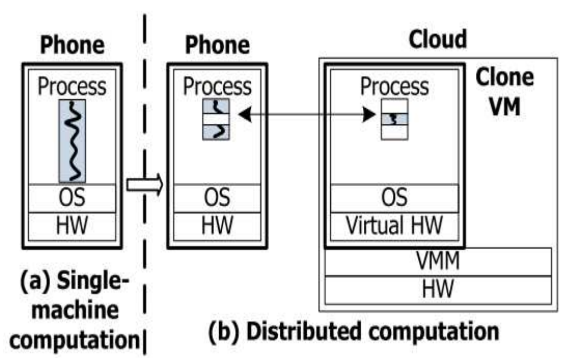

Fig. 9: CloneCloud System Model[44]

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

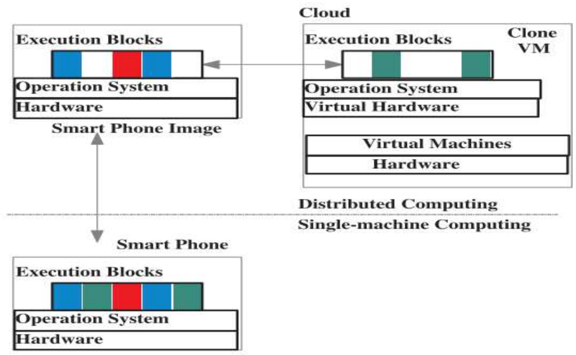

Fig-10 CloneCloud System Architecture[9]

On the basis of CloneCloud X. Zhang [51] introduced an elastic application model to enhance the performance of resource constrained devices by a dynamic execution configuration model of application according to the device current status. This model divides an application into a range of multiple components called weblets as shown in Fig.11. It offers a dynamic adaptation nature of weblet execution configuration. This way the performance of the mobile devices enhanced for more complicated tasks.

Moreover a cost model is provided to adjust the execution pattern yet this model need a mechanism for exchanging weblet between devices, as mobile device changing the communication channel (i.e. 3G to GPRS or Wi-Fi). Another challenge is a media channel or high speed bandwidth is needed to ensure the reliable communication between weblets.

Although both the above approaches are energy efficient for mobile devices yet they still take long communication response time for data transmission between cloud and mobile devices especially when the bandwidth is low. Thus, for light weight applications which can be deploy locally in smartphone, it cannot be justified to offload all the application to cloud. Y.Lu [43] introduced Virtualized Screen in cloud. In this approach the screen rendering is moving from mobile device to cloud as a service and brought as an image to the client device for interactive display. They enable thin-client devices to enjoy various compute-intensive and graphically rich services in cloud. In this approach, screen virtualization does not mean to offload whole rendering task to the cloud but to take offloading decision on the basis of metrics i.e. local device resources efficiency ,network condition, traffic condition , response time, screen resolution etc. Here part of the smartphone's screen is virtualized in cloud which contains collection of data using in display image, audio, video, key board input, text-contents etc. the light weight part of an application is deployed locally to process which effectively diminish the power consumption. So the Y.Lu suggested model for screen rendering partially done in cloud and partially in mobile device.

This way the network bandwidth obstacles can be solved and energy reservation can be achieved. The Y.Lu proposed thin-client computing system depicted in Fig.12. The challenge to remote screen rendering is for real time and high fidelity processing the remote execution of screen might affect by low bandwidth.

To solve the issue of bandwidth delay between mobile device and cloud M. Satyanarayanan [52]

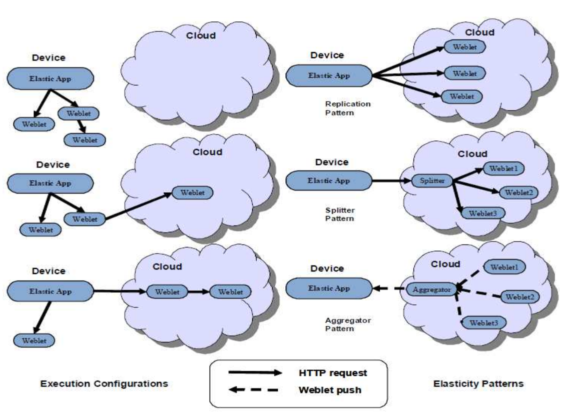

Fig. 11 Elasticity Pattern of Application (Weblets)[51]

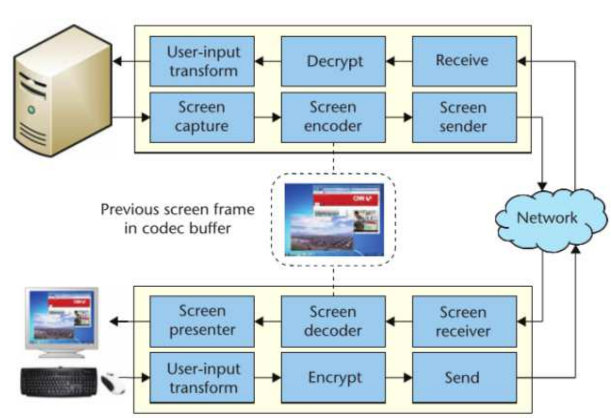

Fig. 12 Remote Screen Rendering by Y. Lu [43]

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

presented the concept of Cloudlet that is a "Micro Cloud" configured in the middle of mobile device and Cloud. The author argued that even though Cloud Computing is the finest solution for resource constraint devices, yet the long WAN latent impede its performance. The rapid changing in computing environment change the accessing bandwidth between mobile device and cloud which leads to different kind of delays, especially when mass data need to transfer and process, the user then feel the presence of such delays. Unfortunately the bandwidth delay can't be avoided completely because of firewall filtering or data checking which are inevitable for security. To overcome the problem virtual machine (VM) technology is used to provide instantaneous customized services to mobile users. Fig.13 shows that a Cloudlet which is a resource-rich computer or cluster of computers, installed in a coffee shop provides the rapid customized services to the client devices by using VM technology through a high bandwidth. Comparative to the distant Cloud, Cloudlet exist in a single hop distance, which provide the fastest processing and transmission bandwidth to the connected devices. In case of no any Cloudlet exist in the surrounding, mobile devices then access the resources of distant cloud or in worst case use their own local resources to handle applications execution. The main challenge in this approach is the compatibility issue related to applications running in Smartphones which are rapidly improving while VM base Cloudlet might not possess such a big range of compatible applications.

Fig.13 Cloudlet Infrastructure [52]

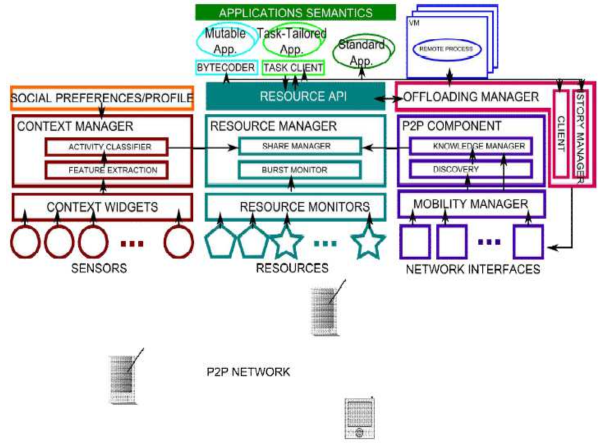

Fig.14 Ad-hoc Mobile Cloud Framework [53]

A bit similar approach has been introduced by G. Huerta-Canepa[53] "ad-hoc mobile cloud framework" a virtual cloud computing platform. He talked about communities built of mobile devices where we can be able to execute shared task. The proposed framework of G.H Canepa allows a small portion of the task to be executed locally while the rest is delegated to the nearest mobile device available in the same vicinity running already the same task. The architecture of ad-hoc mobile cloud depicted in Fig.14, which is consist of five basic components i.e. application manager, resource manager, context manager, offloading manager, and P2P component.

The application manager is responsible for starting and stopping of an application at loading time and also modifying application to take in features according to the current context needed for offloading such as RPC support and proxy creating. The resource manager is in-charge of the application profiling and monitoring of resource in a local device. For each application to execute, a profile is created to keep record of all the remote devices which are needed to build a virtual cloud. The application profile is then checked by application manager every time an application is executed in order to find whether an instance of the required virtual provider needed to be created or not. Context Manager is responsible for the synchronisation of contextual information getting from context widgets and makes them available for other process. Context manager further consist of three sub components. 1) Context widget is

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

responsible to handle communication with the sources of context information. 2) Context manager get new context from the available information. 3) Social manager, which keep record of several type of relationship between users. P2P manger is responsible of informing the context manager of joining a new device in the vicinity or leaving away status of an old device. The offloading manager handle offloading task to the neighbour device for execution, also accept tasks from the other remote devices and processing them.

By the result, this approach is saving energy yet the pervasive nature of nodes needs to have an adopting access mechanism from neighbour. Also a mechanism for dealing the energy consumed in extra computation for making decisions.

A biggest challenge to mobile devices is distributed computing. In such computing new class of applications are needed react to the changes rapidly occurring. Schilit 1994 [54] introduced the term context-awareness in distributed computing (Ubiquitous computing). Application should be aware of the environment they are running in and adapt the changes according to the context. The mobile devices can manage their resources in a better way when devices are aware of contextual computing in the pervasive environment. For example GPS used in Smartphone devices to detect location, but it drain the battery of mobile devices more than other component. The context aware approach will keep the GPS usage in schedule to trigger whenever it need, otherwise will turn it off. Z. Zhuang [55] has developed a framework for location sensing based on the contextual information which is energy efficient as compare to GPS. K. Han Kim [56] developed Wi-Fisense system to sense the environment using low power sensors and previous recorded data to predict the best available network interface for communication and to On Wi-Fi interface on demand fashion to save battery life. R. Herrmann [57] proposed a system Dynamic Power Management (DPM) to avoid the unnecessary sensing of distributed sensing application. This system used the context knowledge to adapt the behaviour of applications. According to the current user's context, the system starts, suspends and change the sampling rate of application used for collecting sample in a sensor network. CABMAN [3] proposed by N. Ravi for battery management of mobile devices. On the basis of user's current context if the system detect a charging opportunity, then warns the users that device battery can run out of power. The system works on the proposed algorithms for processing user's location and call-logs for making some of the predictions.

By using the embedded sensors of mobile device M. Moghimi [58] presented a middleware context aware power management system. Fuzzy inference used in his system for extracting the high level context from low level context, which provide near to accurate results of the user context. The proposed system of M. Moghimi consist of Sensors, CDB (Context Detection Block) and Power Manager as shown in Fig.15.

The system gets raw sensing data from embedded sensors. CDB works as an inference unit, extract the high level context from low level sensing context. This removes the possibility of an application to retrieve same context second time. Power Manager placed in between CDB and applications. It receive the sensing variables registered by the respective applications and tuned them by some defined rules to deliver context-aware energy efficient performance. They get 10-50% lower power consumption using their system. The challenge to this system is to expand the context variables and adopt a dynamic way of determining high level context from low level context.

To make the user interaction limited and build a smart environment D. Sathan [59] proposed a Context Aware Light Weight Energy Efficient Framework (CALEEF). Using this approach the smartphone will be genius enough to decide when

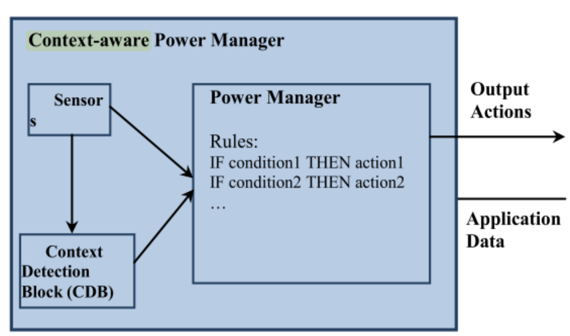

Fig. 15 Context-aware Power Management System [58]

Journal of Theoretical and Applied Information Technology
10th September 2015. Vol.79. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

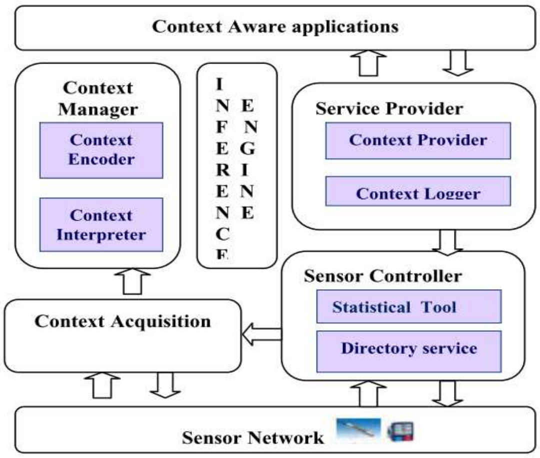

Fig. 16 CALEEF Architecture [59]

to access/ execute the application on the basis of high level contextual information. For example, if user is in the meeting room, the context aware mobile device senses the environment and rejects all the unimportant calls. CALEEF consist of seven components as shown in the Fig.16. A) Context Acquisition is act as a mediator between application and its operating environment. At data acquisition layer specific widget are developed for capturing different kind of information. This layer releases the applications from the issues relating to context sensing by tying the sensor with a single interface. This way it makes independent application design for the method of context sensing.

Context widget continuously updates he context encoder with context information. The context information is then sending to the context service provider for the storage and dissemination of context to the consumers. B) Context Manager is responsible for the conversion of context data received from sensors to the context information that will be provided to application for further action. This component is further divided into two sub-components. Context Interpreter does context processing by logical reasoning; as a result high level context is derived from low level context. It also resolves context conflicts. Context Encoder encodes the context information using OWL and then passes it to the context logger for record. C) Interface Engine performs reasoning on stowed facts and using the past and current context information to define how an application should change its behaviour accordingly. D) Context Logger: context-aware application may change their behaviour using past context along with current context. For this reason the previous context are encoded and stored in the context history that may be queried by applications later whenever need. The context logger is made up of context knowledge base and context history. The context knowledge base provides a set of API's for the components of other services to query, modify, add, delete context information. E) Context Provider is responsible for keeping record of the context consumers and always trigger them whenever get new context information. F) Directory Services register the sensors information of the surrounding. It keeps record of the Sensors attributes i.ee refresh rate, spatial information, correctness etc. by using this mechanism CALEEF pick the most suitable sensor for receiving context data.

G) Context Consumers consumes different kind of context information and adapt their behaviour accordingly. They either listen to the context provider for new context information or query them to get updates.

The main feature of CALEEF is context reasoning. High level implicit context can be derived from low level explicit context. Application confidently uses the high level context information to change their behaviour. The challenging issue in CALEEF framework is, in case of sensor or any component failure need the system to restart and restore itself to the last working state.

Some researchers utilize cloud resources using contextual approach to minimize local resources operation. Y.Xio [60] proposed a framework CasCap (Cloud-assisted Context-aware Power Management). In his work the cloud resources for processing, storage, and networking are utilized to provide an efficient low cost power management of mobile devices. CasCap consist of three main components, mobile devices, internet services and clones. Mobile devices part of CasCap consists of five components, i.e. Resource Manager, Context Manager, Scheduler, Policy Manager and Communicator as shown in Fig.17.

Resource Manger runs in background and responsible for monitoring the device resource's consumption. It also collects the contextual data from sensors like GPS, accelerometer etc. Context

Journal of Theoretical and Applied Information Technology
10th September 2015. Vol.79. No.1

ISSN: **1992-8645** www.jatit.org E-ISSN: **1817-3195**

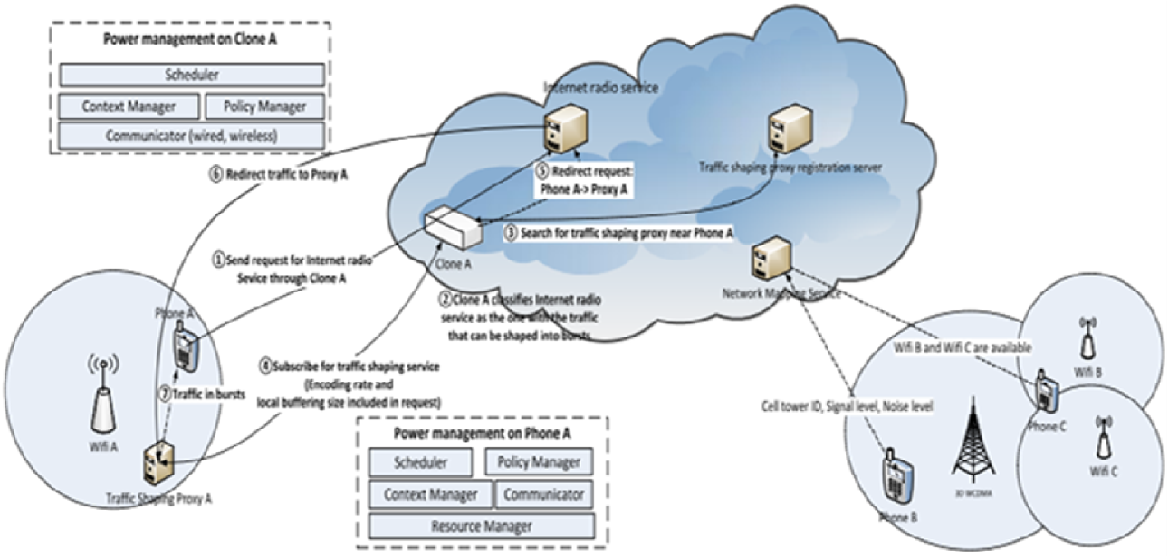

Fig. 17 CasCap Architecture [60]

Manager generates the context information on the basis of data collected by the resource manager from sensors and uploads the context information to the cloud.

A crowed- context monitoring service in the cloud which receive the contextual information from Context Manager of mobile devices and other network elements and then query on them to get meaningful context information. The Scheduler is responsible for keeping track of changes in the context and then adapts the mobile devices according to the changes. The Policy Manager stores all the policies which are made to govern the process of CasCap framework. These policy are specific rules which and actions that should be taken by the device itself or by the internet Cloud whenever a specific changes occurs in the context information. The Communicator component is responsible for providing the wireless communication facility between mobile device and cloud for sharing the policies, context information and using internet services.

The result provided by Y. Xio is significant in some cases, and the main feature of CasCap is the monitoring of crowed-sources context, function offloading to cloud and adaptation as services. They involved first time a third party services to be part of development and deployment of power management services. The system still needs to resolve the challenges in migration of radio stream from one proxy to another whenever the user moves. Also need a filter process for context information stored in the cloud as after some time the stored context information might become invalid.

All of the above approaches brought about different solutions at different levels to minimize the power consumption of handheld devices. The two most popular solutions are, either conserves battery life by resource management or by offloading (migrating load to cloud servers). Singular solution by simply job migration to cloud or only resource management cannot provide an adequate power saving solution as both the approaches having their own limitations. In our research we combine the two approaches i.e. local device resource management and offloading compute intensive application to cloud to save battery life. By resource management part of our research we focus on the maximum utilization of local components using context information to stop drainage of unnecessary battery life. In case of compute intensive applications i.e. graphics, games, language translation, we work on offloading to minimize operational load of local resources. CEEF (Context-aware Energy Efficient Framework) suggested for achieving the combine results of both the approaches.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

6. SUMMARY

The analysis of previous approaches on augmentation of the device's resources with the expected outcomes and limitations are summarised in Table 1.

Table.1 Analysis of the Previous Research Work on Resources Augmentation of Mobile Devices.

Citation	Authors/Years	Focus/Purpose	Framework	Method	Outcomes / Results	Limitations/ Future Work
[44]	B. Chun et al., 2011	Reduce Mobile Resources Labour	CloneCloud	Offloading	21.2x speedup smartphone device application processing Protect battery life by remote application execution	Handover delay and bandwidth limitation
[51]	X. Zhang et al., 2011	Dynamic application execution	Weblet	Elastic application configuration	Performance enhanced by dynamic adaptation nature of complicated tasks. Quick & dynamic access of application reduce the local resources operation i.e. save battery life	Mechanism need for exchanging weblets between devices, with changing comm-unication channels (i.e. 3G to GPRS or Wi-Fi)
[43]	Y.Lu et al., 2011	Reduce Mobile Resources labour	Interactive Screen Remote system	Screen Virtualization in Cloud	Thin-client devices to enjoy various compute-intensive and graphically rich services in cloud Reduces local resources operations & conserve battery life	For real time and high fidelity processing the remote execution of screen might affected by low bandwidth
[48]	Satyanarayanan et al., 2009	Solve the issue of bandwidth delay b/w mobile device and cloud	Cloudlet	Offloading (Cyber Foraging)	Cloudlet exist in a single hop distance, provide the fastest processing and transmission bandwidth to the connected devices Provides the rapid customized services to the client devices by using VM technology through a high bandwidth.	Applications compatibility issues
[53]	G. Huerta-Canepa et al., 2010	Reduce Mobile Resources labour	Virtual Cloud Computing Platform	Offloading (Remote Execution)	The pervasiveness of mobile devices, creating a cloud among the devices in the vicinity, allowing them to execute jobs between the	The pervasive nature of nodes needs to have an adopting access mechanism from neighbour. Also a

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

					devices.	mechanism for dealing the energy consumed in extra computation for making decisions.
[54]	B. N. Schilit et al., 1995	Resource Management	PARCTAB System	Context-aware Computing	Developed unique set of context-aware application which enhances the operation of applications by communication and context information. PARCTAB depends on small cell wireless communication, thus combines portability with information about context.	PARCTAB system has very limited use when disconnected from a network.
[55]	Z. Zhuang et al., 2010	Resource Management	Location Sensing Framework	Context-aware Computing	Reduce GPS usage up to 95 % while increase battery life up to 75 %	As compare to GPS, the proposed system cannot provide accurate location sensing in some cases.
[56]	K. Han Kim et al., 2011	Resource Management	WiFisense System	Context-aware Computing	Increases WiFi usage for various scenarios. Save energy consumption for scanning by up to 79 % while reduces false triggering by up to 4.3 %	The accelerometer is unable to provide the accurate movement information without any location base sensor.
[57]	R. Herman et al., 2012	Resource Management	Context-aware DPM	Context-aware Computing	The tested technique on a real system shows that it can extend smartphone battery life by 5x.	The context detection using sensor is an extra cost that the system has to pay to gain the context knowledge.
[3]	N. Ravi et al., 2008	Resource Management	CABMAN	Context-aware Computing	Predict next charging opportunity on the basis of developed prediction algorithm. Accurate battery life prediction based on a discharge speedup factor. Save battery life for crucial	For those users who spent a very high entropy routine, the prediction al-gorithms may not work very well. As many users charge phones in

Journal of Theoretical and Applied Information Technology
10th September 2015. Vol.79. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

					applications i.e. Telephony	their cars while driving, in such condition it is not always possible to consider location prediction for charging availability.
[58]	M. Moghimi et al., 2013	Resource Management	Context Aware Power Manager	Context-aware Computing	Fuzzy inference used to provide high level context. The results show reduction of energy 13-50% for periodic applications, and for streaming applications 18- 36%.	Need to expand the context variables and adopt a dynamic way of determining high level context from low level context.
[59]	D. Sathan et al., 2009	Resource Management	CALEEF	Context-aware Computing	Reduces the cost and complications of developing context-aware applications by a shared context model of distributed software components. It also get context from a widespread range of sources rather than sensors only that are rooted in the local environment. It enables knowledge sharing among applications entities.	The need of autonomic service-oriented Comput-ing ideas for developing context-aware service frameworks. In case of sensor or any component failure need the system to restart and restore itself to the last working state.
[60]	Y. Xio et al., 2011	Resource Management + Offloading	CasCap	Context-aware Computing	CasCap comprise of crowd-sourced context monitoring, function offloading, and adaptation as service. For the third party service providers the frame work provide a fresh way to develop and deploy power management services.	The system still needs to resolve the challenges in migration of radio stream from one proxy to another whenever the user moves. Need a filter process for context information stored in the cloud as after some time the stored context information might become invalid.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The different approaches analysed in Table.1. [44] used the concept of cloning mobile device in distant cloud to process compute intensive task migrated away from mobile device, yet the mobility feature and low/interrupting communication / bandwidth issues some time causing delay of the services. Moreover synchronization with the clone device each time increases RTT (Round Trip Time).

[43] Presented a concept of the screen rendering instead of migrating whole task to distant cloud. They introduced Virtualized Screen in cloud to overcome bandwidth delay issue. In this approach the screen rendering is moving from mobile device to cloud as a service and brought as an image to the client device for interactive display. Here, screen virtualization does not mean to offload whole rendering task each time to the cloud but to take offloading decision on the basis of metrics i.e. local device's resources efficiency, network condition, traffic condition, response time, screen resolution etc. In this approach part of the smartphone's screen is virtualized in cloud which contains collection of data using in display image, audio, video, key board input, text-contents etc. the light weight part of an application is deployed locally to process.

The same way [48] presented the concept of Cloudlet which is a "Micro Cloud" configured in the middle of mobile device and Cloud. The author argued that even though Cloud Computing is the finest solution to overcome limitation of resource constraint devices, yet the long WAN latent impede its performance. The rapid changing in computing environment alters the accessing bandwidth between mobile device and cloud which leads to different kind of delays, especially when mass data needs to transfer and process. Users feel the presence of such delays. This approach shows that a Cloudlet which is a resource-rich computer or cluster of computers installed in a coffee shop provides the rapid customized services to the client devices by using (Virtual Management) VM technology through a high bandwidth. Comparative to the distant cloud, Cloudlet exist in a single hop distance, which provide the fastest processing and transmission bandwidth to the connected devices. In case, if no Cloudlet exist in the surrounding, mobile devices then access the resources of distant cloud or in worst case uses their own local resources to handle applications execution. The main challenge in this approach is the compatibility issue related to applications running in Smartphones which are rapidly improving while VM base Cloudlet might not possess such a big range of compatible applications.

The approaches [43][44][48] generally facing the same issue of low bandwidth and unnecessary offloading, while mobile device can locally process the task easily. For this reason, to make the user interaction limited and build a smart environment D. Sathan [59] proposed a Context Aware Light Weight Energy Efficient Framework (CALEEF). Here the smartphone become genius enough to decide when to access or execute the application on the basis of high level contextual information. For example, if user is in the meeting room, the context aware mobile device will sense the environment and will reject all the unimportant calls. The shortcomings in this approach is the failure of the sensor which causing disconnection of the whole service. In case of sudden failure of the system, an automatic mechanism needs to restate the system to the previous state. [55] [56][57] [3] [58] [60] used the concept of context awareness to make a precise decision at the time of offloading remote executable parts of different application. The major issue in context-aware approach is of extra sensor and filtration of high level context information from low level sensed context input.

7. SUGGESTION AND OPENINONS

In future research until we reach to the development of power sources (batteries) better than Lithium Ion batteries, the combine approach at software level of two techniques i.e. management of local resources in mobile devices and offloading compute intensive applications to cloud can possibly maximize the efficiency of battery life. At resource management part of research the focus should be on maximum utilization of local components using context information to stop unnecessary processing and reduce battery consumption. In case of compute intensive

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

applications i.e. graphics, games, language translation, the existing offloading techniques can work good to minimize operational load of local resources, yet the shortcoming in context gathering or Long RTT (due to low bandwidth or communication issues) in both the approaches are the significant concerns to address.

8. CONCLUSION

The extremely slow development at battery technology shows that the only substitute left to solve the power issue of mobile devices is to reduce the power consumption at software level. Numerous researchers worked on replenishing the battery power / battery timing of handheld device by different ways, yet the prompt progress of mobile technology (other than battery side) increased the actual power burden. In such circumstances, apart from resource management solutions such as to design more power efficient operating systems and applications, the reliance on cloud services is becoming more essential for various needs (storage, security and computation offloading). Cloud computing, the shared resource computing environment provides a vast pool of on-demand available resources to utilize and enhance capabilities of resource limited devices by computational offloading. MCC offers new openings in energy saving by rendering the computing resources, mobile resources and network resources to increase resource utilization, sharing and virtualization for achieving cost reduction. Thus, to enhance the output of mobile devices, the utilization of cloud services is the best possible solution.

REFERENCES:

[1] M. Shiraz, A. Gani, R. H. Khokhar, and R. Buyya, “A Review on Distributed Application Processing Frameworks in Smart Mobile Devices for Mobile Cloud Computing,” IEEE Commun. Surv. Tutorials, vol. 15, no. 3, pp. 1294–1313, 2013.

[2] K. Kumar and Y. Lu, “Cover feature cloud computing for mobile users : computation save energy ?,” IEEE Computer Society" Issue No.04 , vol.43 , pp. 51–56, April 2010.

[3] N. Ravi, J. Scott, L. Han, L. Iftode, I. Corporation, and S. Clara, “Context-aware Battery Management for Mobile Phones.” Pervasive Computing and Communications, Sixth Annual IEEE International Conference. Pages 224 – 233, 2008.

[4] D. Schmidt and N. Wehn, “A Review of Common Belief on Power Management and Power Consumption,” White Paper" pp. 1–13, 2009.

[5] N. K. (Vice P. at L. Energy), “Six things to know about smartphone batteries.” 29th May 2012 .

[6] M. Ali, J. M. Zain, M. F. Zolkipli, and G. Badshah, “Mobile Cloud Computing & Mobile Battery Augmentation Techniques : A Survey,” IEEE SCOReD 2014.

[7] M. Weiser, “The computer for the 21st Century,” IEEE Pervasive Comput., vol. 1, no. 1, pp. 19–25, Jan. 2002.

[8] “Gartner: Top 10 Strategic Technology Trends For 2014,” 2013. [Online]. Available: http://www.forbes.com/sites/peterhigh/2013/10/14/gartner-top-10-strategic-technology-trends-for-2014/.

[9] H. Qi and A. Gani, “Research on Mobile Cloud Computing : Review, Trend and Perspectives,” Digital Information and Communication Technology and it's Applications (DICTAP), 2012 Second International Conference, Pages 195-202, 2014.

[10] B. Melk and S. Nunn, “Cloud Computing level of capability to the application.”Accenture Cloud Computing", 2009.

[11] P. Mell and T. Grance, “The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology.” Computer Society , Sep 2011.

[12] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, “Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility,” Futur. Gener. Comput. Syst., vol. 25, no. 6, pp. 599–616, Jun. 2009.

[13] M. D. Kristensen, “Enabling Cyber Foraging for Mobile Devices.”Proceedings of the 5th MINEMA Workshop" , 2007.

[14] “Aamazon S3.” [Online]. Available: http://en.wikipedia.org/wiki/Amazon_S3. [Accessed: 25-May-2014].

[15] Wesley Chun, “What is Google Apps Engine?,” 2011. [Online]. Available: https://ep2013.europython.eu/conference/talks/g

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

oogle-app-engine-best-practices-latest-features. [Accessed: 25-May-2014].

[16] "What is Windows Azure?" [Online]. Available: http://www.microsoft.com/bizspark/azure/ [Accessed: 25-May-2014].

[17] "White Paper, 'Mobile Cloud Computing Solution Brief,' AEPONA, November 2010."

[18] M. Satyanarayanan, "Fundamental challenges in mobile computing," Proc. fifteenth Annu. ACM Symp. Princ. Distrib. Comput. - Pod. '96, pp. 1–7, 1996.

[19] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A Survey of Mobile Cloud Computing : Architecture, Applications, and Approaches," Wireless Communication & Computing" no. Cc, pp. 1–38, Oct 2011.

[20] N. Vallina-rodriguez and J. Crowcroft, "Modern Mobile Handsets" IEEE Communications Surveys & Tutorials, vol. 15, Issue no. 1, pp. 1–20 , first quarter 2013" , 2012.

[21] D. Revision, "The ARM Cortex-A9 Processors," pp. 1–11, 2009.

[22] A. D. Batteries, "From the Editor-in-Chief.," Nurs. Leadersh. (Tor. Ont)., vol. 26, no. 1, pp. 1–2, Jan. 2013.

[23] S. Abolfazli, Z. Sanaei, and A. Gani, "Mobile Cloud Computing : A Review on Smartphone Augmentation."1st International Conference on Computing, Information Systems, and Communications (CISCO'12)" , May 2012.

[24] G. P. Perrucci, F. H. P. Fitzek, and J. Widmer, "Survey on Energy Consumption Entities on the Smartphone Platform," 2011 IEEE 73rd Veh. Technol. Conf. (VTC Spring), pp. 1–6, May 2011.

[25] X. Liu, "Chameleon : Application Level Power Management with Performance Isolation," MM'05 Singapore Nov- 6–12, 2005.

[26] A. Rudenko, P. Reiher, G. J. Popek, and G. H. Kuenning, "Saving portable computer battery power through remote process execution," ACM SIGMOBILE Mob. Comput. Commun. Rev., vol. 2, no. 1, pp. 19–26, Jan. 1998.

[27] M. Satyanarayanan, "Pervasive computing: vision and challenges," IEEE Pers. Commun., vol. 8, no. 4, pp. 10–17, 2001.

[28] D. Garlan, D. P. Siewiorek, and P. Steenkiste, "Toward Distraction-Free Pervasive Computing," PERVASIVE Computing, 2002.

[29] J. Flinn and M. Satyanarayanan, "Balancing performance, energy, and quality in pervasive computing," Proc. 22nd Int. Conf. Distrib. Comput. Syst., pp. 217–226, 2002.

[30] "Your stuff, anywhere." [Online]. Available: https://www.dropbox.com/. [Accessed: 03-Jun-2014].

[31] "Google Docs." [Online]. Available: www.docs.google.com, [Accessed: 03-Jun-2014].

[32] R. K. Balan, "Powerful change part 2: reducing the power demands of mobile devices," IEEE Pervasive Comput., vol. 3, no. 2, pp. 71–73, Apr. 2004.

[33] J. Flinn and M. Satyanarayanan, "Energy-aware adaptation for mobile applications," vol. 34, no. 5, pp. 48–63, 1999.

[34] E. De Lara, D. S. Wallach, and W. Zwaenepoel, "Puppeteer : Component-based Adaptation for Mobile Computing." Proceedings of the 3rd USENIX Symposium on Internet Technologies and Systems, March 26-28, 2001.

[35] R. Kravets and P. Krishnan, "Application-driven power management for mobile communication," Wireless Networks, vol. 6, pp. 263–277, 2000.

[36] M. Rahman, J. Gao, and W.-T. Tsai, "Energy Saving in Mobile Cloud Computing*," 2013 IEEE Int. Conf. Cloud Eng., pp. 285–291, Mar. 2013.

[37] S. Frattasi, H. Fathi, F. H. P. Fitzek, R. Prasad, M. D. Katz, and S. Electronics, "Defining 4G Technology from the User's Perspective," Network, IEEE Vol 20, pp. 35–41, Feb 2006.

[38] N. Balasubramanian, "Energy Consumption in Mobile Phones : A Measurement Study and Implications for Network Applications," IMC '09 Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference P-280-293, 2009.

[39] D. Bertozzi, a. Raghunathan, L. Benini, and S. Ravi, "Transport protocol optimization for energy efficient wireless embedded systems," 2003 Des. Autom. Test Eur. Conf. Exhib., pp. 706–711, 2003.

[40] L. Zhang, A. R. Dick, A. Jason, N. Flinn, A. Z. M. Mao, A. Z. Zhang, and P. A. Dinda, "Power , Performance Modeling and Optimization for Mobile System and Applications," 2013.

[41] Bill Claybrook, "Mobile cloud apps vs. native apps: The developer's perspective," 2013. [Online]. Available: http://searchcloudapplications.techtarget.com/feature/Mobile-cloud-apps-vs-native-apps-The-

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

developers-perspective. [Accessed: 08-Jun-2014].

[42] K. Kumar, “Cloud Computing for Mobile Users: Can Offloading Computation Save Energy?,” Computer (Long. Beach. Calif)., vol. 43, no. 4, pp. 51–56, Apr. 2010.

[43] W. Zeng and Y. Lu, “Multimedia at Work Virtualized Screen : A Third Element for Cloud Mobile Convergence, IEEE Computer Society, pp. 4–11, 2011.

[44] B. Chun, S. Ihm, P. Maniatis, M. Naik, and I. L. Berkeley, “CloneCloud: Boosting Mobile Device Applications through Cloud Clone Execution.” Distributed, Parallel, and Cluster Computing (cs.DC); Operating Systems (cs.O) Sep- 2010.

[45] K. Kumar, J. Liu, Y.-H. Lu, and B. Bhargava, “A Survey of Computation Offloading for Mobile Systems,” Mob. Networks Appl., vol. 18, no. 1, pp. 129–140, Apr. 2012.

[46] A. Messer, I. Greenberg, P. Bernadat, D. Milojicic, D. Chen, T. J. Giuli, and X. Gu, “Towards a Distributed Platform for Resource-Constrained Devices computing Towards a Distributed Platform for Resource-Constrained Devices,” 2002.

[47] E. Cuervo, A. Balasubramanian, and D. Cho, “MAUI : Making Smartphones Last Longer with Code Offload, "MobiSys 10 Proceedings of the 8th international conference on Mobile systems, applications and services" , p- 49-62, 2010.

[48] R. Balan, J. Flinn, M. Satyanarayanan, S. Sinnamohideen, and H.-I. Yang, “The case for cyber foraging,” Proc. 10th Work. ACM SIGOPS Eur. Work. beyond PC - EW10, p. 87, 2002.

[49] R. K. Balan and D. Garlan, “Simplifying Cyber Foraging,” Proceedings of the 5th international conference on Mobile systems, applications and services" PP. 272-285, May 2006.

[50] A. R. Reserved, “5th Symposium on Operating Systems Design and Implementation,” 2002.

[51] X. Zhang, A. Kunjithapatham, S. Jeong, and S. Gibbs, “Towards an Elastic Application Model for Augmenting the Computing Capabilities of Mobile Devices with Cloud Computing,” Mob. Networks Appl., vol. 16, no. 3, pp. 270–284, Apr. 2011.

[52] N. Davies, “The Case for VM Based Cloutlets in Mobile Computing, "Journal, IEEE Pervasive Computing" V- 8, PP- 14-23, Oct-2009.

[53] G. Huerta-Canepa and D. Lee, “A virtual cloud computing provider for mobile devices,” Proc. 1st ACM Work. Mob. Cloud Comput. Serv. Soc. Networks Beyond - MCS ’10, pp. 1–5, 2010.

[54] “Want95_PARCTab.pdf.” .

[55] Z. Zhuang, D. T. R, and L. Altos, “Improving Energy Efficiency of Location Sensing on Smartphones," Proceedings of the 8th international conference on Mobile systems, applications, and services”, p. 315-330, 2010.

[56] K.-H. Kim, A. W. Min, D. Gupta, P. Mohapatra, and J. Pal Singh, “Improving energy efficiency of Wi-Fi sensing on smartphones,” 2011 Proc. IEEE INFOCOM, pp. 2930–2938, Apr. 2011.

[57] R. Herrmann, “Context Aware Power Management of Mobile Systems for Sensing Applications” IPSN 12, April 16–20, 2012, Beijing, China”, pp. 4–8, 2012.

[58] M. Moghimi, J. Venkatesh, P. Zappi, and T. Rosing, “Context-Aware Mobile Power Management Using Fuzzy Inference as a Service," 4th International Conference, MobiCASE”, p. 314-327, 2013.

[59] D. Sathan, A. Meetoo, and R. K. Subramaniam, “Context Aware Lightweight Energy Efficient Framework,” "International Journal of Humanities & Social Sciences” pp. 64–70, 2009.

[60] Y. Xiao, P. Hui, and P. Savolainen, “CasCap : Cloud-assisted Context-aware Power Management for Mobile Devices.” MCS '11 Proceedings of the second international workshop on Mobile cloud computing and services, pp- 13-18, 2011.