Mobile Grid Computing: Facts or Fantasy?

Ketan B. Parmar[1],* Nalinbhai N. Jani[1], Pranav S. Shrivastav[2], Mitesh H. Patel[2]

[1]Department of Computer Sicence, Kadi Sarva Vishwavidyalaya, Gandhinagar – 382015, Gujarat, India
[2]Chemistry Department, School of Sciences, Gujarat University, Navrangpura – 380009, Gujarat, India

Abstract– **The growing availability of networked mobile devices has created a vast collective pool of unexploited computing resources. In this paper we analyze the challenges for harvesting this potential for grid computing. The inherent limitations of typical mobile devices, such as reduced CPU performance, small secondary storage, heightened battery consumption sensitivity, and unreliable low-bandwidth communication are outraced by number of smart-phones sold annually, suggesting that the concept should not be prematurely dismissed. Given the enormous benefits of mobile grid computing, solutions to compensate for the inherent limitations of these devices must be developed in order to successfully utilize them in the Grid.**

Keywords– **Grid Computing, Mobile Wireless Computing, Pervasive Computing and Network Clusters**

I. INTRODUCTION

Grid computing [1], [2] involves the aggregation of network-connected computers to form a large-scale, distributed system for coordinated computing and resource sharing. By spreading computing workload across this distributed system, grid users can take advantage of enormous computational, storage, and bandwidth resources that would otherwise only be available to traditional multiprocessor supercomputers [3]. Grid is already being successfully used in many scientific applications where huge amounts of data have to be processed and/or stored. Such demanding applications have created, justified and diffused the concept of grid computing, among others, in the scientific community [4]. On the other hand, the availability of wirelessly connected mobile devices has grown considerably in recent years, creating an enormous collective unexploited pool for resource utilization. Thus, the extrapolation of the resource aggregation model of grid computing to mobile computing devices seems only natural. Such a system, which uses computing resources of networked mobile devices (with or without fixed computers) is termed “mobile grid”. Having been constructed from a group of mobile devices, a mobile grid would allow the networked devices to accomplish a specific mission that may be beyond an individual device’s computing or communication capacity [3].

Recent years have experienced a transition towards mobility in the form of mobile devices (like iPod & iPad) most importantly in form of so called "smartphones" (like iPhone, Android mobile, BlackBerry mobile, Palm mobile etc.) and the trend is upward still (see Figure 1). These devices are equipped with not only powerful processors but also large memory and storage - 1 GB RAM and 2 GHz processor is a common trend [5]. With sales of the Apple iPhone topping 1 million in the first weekend alone and approaching 10 million, it may be envisaged that the growth of the smartphone market is huge and it will continue to grow, to the extent that some hail this phenomena as the “next wave in computing” [6]. Smartphones are predicted to become nearly ubiquitous and are thus a major step towards the vision of ubiquitous computing [7] so often dreamed of. The combination of pervasive wireless networks and computational devices able to take advantage of them has created an era of mobile computing, the likes of which have never been seen before. It has been argued [5] that present trend will prevail in future as well because, (1) As Moore’s Law of increasing transistor density results in increased CPU performance of PDAs, the market will see a growth in CPU speed as it has seen for desktop PCs. Such products as Intel’s xScale line of power-efficient, fast CPUs specifically designed for the handheld market bode well for future PDAs. (2) Wireless communication will grow as well for both local-area as well as wide-area. (3) Battery efficiency will not substantially improve. An analogous transition has been observed in the field of communications and networking technologies in the form of flexible wireless networks, among others [5]. Resent time has seen an explosion of consumer electronic devices taking advantage of wireless technology to provide increased productivity because of higher connectivity. When one considers the broad range of wirelessly connected mobile devices used today, from laptops to smartphones with cellular data modems, it is clear that the implications of this ever increasing connectivity will be far reaching.

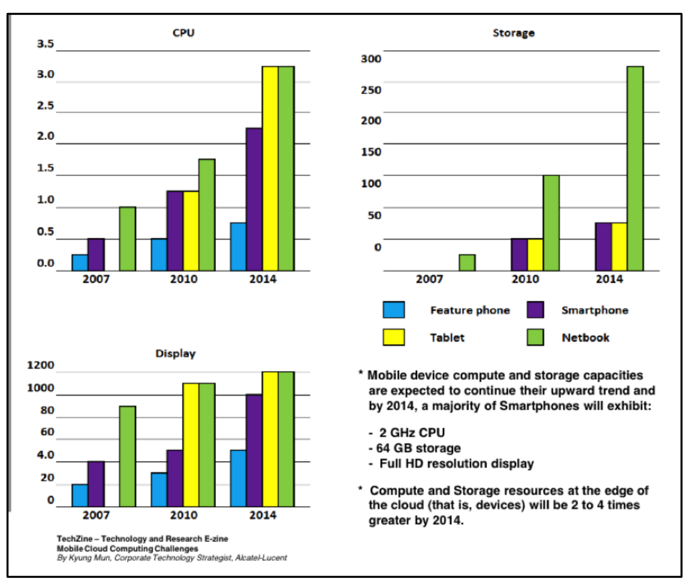

Figure 1. Mobile device computing, storage and display trends

As herein discussed, networked mobile devices represent an already large (and growing) percentage of available worldwide computing power to be leveraged to be used for truly distributed computing, with applications that do not simply rely on the existence of centralized infrastructure but rather may interact directly with each other in more complex and powerful ways. Furthermore, in addition to traditional computer resources like processors, memory, storage, and applications, new generation mobile devices provide support for integrated multimedia equipment, intelligent positioning systems, and a diverse range of sensors. Integration of these mobile devices into the grid could benefit both the mobile and the grid communities. The mobile users will gain access to a huge number of resources, high performance facilities, specialized hardware and software and enormous computing or storage resources. Whereas for the grid community, it is an opportunity to utilize available resources and increase its performance & capacity for a broad range of services/functionality. The most important feature the mobile technology will gift to the grid is virtual ubiquity [8]. Thus, the case for identifying and addressing fundamental issues of mobile grid computing by researchers in order to find ways to harness this abundance is further strengthened, and if there is any 'best' time to take first step in this direction, it is 'now'.
There are four basic approaches by way of which mobile device may be deployed in grid computing:

1. Monitoring and reporting purpose
2. As sensors
3. For computational purpose
4. For data storage.

Out of these, approaches 1 and 2 merely facilitate the accessibility or are to be employed in specialized implementations. In fact, the second approach, of using mobile devices as sensors does not fit in the original concept of grid computing put forward by I. Foster. Thus, approach 3, whereby mobile devices are used for computational purpose leverages the herein discussed potential of mobile devices most efficiently, followed by their use as data storage units (approach 4). Thus, approach 3 is at the center of our present discussion. Current mobile devices have enough processing power and memory to execute complex jobs, plus, communication methods like Wi-Fi, 4G and 3G are fast enough to fulfill communication requirements.

One may wonder though, if the state-of-the-art of mobile technology is consistent with the above description, why have not real life mobile grids materialized, and hence the title of present review. Although, everything seems to be favoring the advent of mobile grids, the concept has not cought on. Prima facie the case seems vary factual and yet it is a mere fantasy for some of us. Sure, there are reasons, some of them recognized and few being worked upon, yet here we are. Over a period of time, some publications have appeared addressing one or more aspects of issues involved in mobile grid computing, but there is a lacunae to be satisfied with an all inclusive reference on the state of the art, and hence this review. In this review we analyze these hurdles before harvesting the increasingly widespread availability of Internet-connected wireless mobile devices to be beneficially used within the emerging field of mobile grid computing. The rest of this review is organized as follows. In the next section, we provide a brief overview of relevant research in this field. Section 3 gives detailed commentary on the challenges the field is facing for integration of mobile devices to the grid followed by concluding remarks.

II. RELATED WORKS

There are two ways to leverage the computational potential of mobile devices for grid computing (aforementioned approach 3).

i). To integrate the mobile devices in existing grid implementations. However, existing implementation are focused for desktop system only. That is, Grid Middleware are developed for desktop hardware and OS only and may not support mobile hardware or OS partially or entirely. Further, communication technology for desktop and mobile devices, network topology and task sizes are different and/or optimized for desktop computing. On the flip side, if above limitations are bypassed/taken care of (e.g. todays windows mobile smartphones), the advantage is that there is no need to design separate middleware for scaling up existing grid by incorporating mobile devices.

ii). To develop separate grid implementation for only Mobile devices. However, this way requires one to develop separate/specialized middleware and applications for mobile grid. This implementation will of course be easy to set-up and maintain, and thus can be leveraged in universities and such institutions, even as volunteer grids.

Foster et al have contributed pioneering work in the field of grid computing over a period. Their concept [1] of grid is "flexible, secure, coordinated resource sharing among dynamic collection of individuals, institutions, and resources what we refer to as virtual organizations". They have highlighted the need for grid technology in virtual organization time and again, but for some reason, are yet to channel their efforts in the field of mobile grid computing. Nevertheless, a number of recent and current research projects and papers by several groups are dealing with challenges of mobile grid computing. It must be noted however, that, most of the research described hereafter is currently in, one may say initial, stage(s) of research only, and no real world implementations have been materialized out of the research lab so far. Notable contributions to the field of Mobile grid computing are, therefore, surveyed briefly in present section with befitting commentary.

J. Hwang et al [9] propose a virtual cluster approach and a middleware to provide peer-to-peer operations. Siegel et al [10] and Datta [11] have advanced conceptual models of ad-hoc mobile grid architectures that deploy intelligent agents and relay on the virtual backbone (constructed from powerful mobile nodes and maintained with a proactive protocol) to organize the network scale resources for a specific mission. They have not addressed resource virtualization or federation of similar resources or even larger scale collaboration and issues of mobility and failure resilience. Similar models have been the theme in many recent papers and research projects lately, [7, 12, 13, 14, 15] but none of these provide any implementation methodology or architecture to support this integration. One contribution worth mentioning here is that of T. Phan et al, [5] who have visited the challenges of integrating mobile devices with computational grid in detail within the scope of their project. They propose, the integration be provided through the use of an Interlocutor (software agent), which acts as proxy for cluster of Minions. They have not exemplified, however, any selection strategy to replace a mobile interlocutor, moving from cell to cell. Implementation considerations or evaluation reports of their approach are not available either.

Kurkovsky et al [16, 17] have proposed an agent based wireless grid architecture to solve computationally expensive tasks. Their architecture enables mobile devices within a wireless cell to form computational grid. Like others, prime limitation of their approach is inadequate consideration for the mobility of the nodes, where, tasks are indiscriminately aborted by subordinates and/or Initiators whenever these mobile agents move to neighboring cells – giving rise to redundancy and performance loss.

Several efforts have also been made to identify the problems and document the contemporary state-of-the-art, not unlike present one. McKnight et al. [18] gives an overview of the field of wireless grid computing. It discusses the additional capabilities offered by wireless grids and the new challenges faced by wireless grids compared to traditional grids. Most importantly, their work identifies five requirements for wireless grid middleware: resource description, resource discovery, coordination, trust establishment, and clearing. Ahuja and Myers [19] provides a survey of wireless grid computing, following a structure similar to McKnight et al.

Mobile OGSI.NET [20] is an implementation of an OGSI based grid container on the .NET hosting environment on mobile devices based on Microsoft's PocketPC. Mobile OGSI.NEt al.lows for Grid service state saving and restoring and distribution of workload among devices with the same types of services, but with the cost of having to change existing services to adhere to the specific Mobile OGSI.NET programming model. Furthermore, Mobile OGSI.NET can only be realized on PocketPCs with the .NET framework installed.

AKOGRIMO [21] is a European funded project that is geared to deal with mobility issues in the Grid. The purpose of the project is to evaluate the mobile grid introducing the notion of mobile dynamic virtual organizations through applications that highlight the challenges present in such mobile environments, like e-health, e-learning and crisis management. AKOGRIMO favors research on ad-hoc mobile Grids and the integration of mobile IPv6 to support mobility in a Grid environment.

Legion [22] is a middleware developed at University of Virginia, and is defined as a meta-system based on resources with billions of hosts and trillions of objects linked together by high-speed networks, workstations, and supercomputers in a system that can aggregate different architectures, operating systems, and physical locations. The support provided by Legion for mobile service characteristics is seen in collaboration, in which the binding system makes possible the collaboration through tuples such as that supply object addressing (LOID) and management (LOA). There also exists resource allocation, provided by LOA (Legion Object Addresses) that incorporates a physical address as the IP and can distribute these resources through multicast.

'Ibis for mobility' project by Palmer et al. [23] applies grid computing techniques to distributed computing on mobile devices, which includes integrating mobile phones into the grid. This included porting the Ibis grid computing platform to run on Android. Also discussed are the challenges of mobile distributed computing and a strong argument for distributed computing on mobile devices is presented based on the growth in the Smartphone market and the pitfalls of cloud computing using proprietary services. Similarly, WIPdroid by Chou and Li [24] is another distributed computing platform for Android. It is based on the Web Services Session Initiation Protocol (WIP), which allows "real-time service-oriented communication over IP". Using WIP, WIPdroid can provide a two-way web service interface similar to that of an online service supported by a "cloud" backend of mobile devices. Like Ibis, WIPdroid is developed and tested on Android emulators. Further, GridGain Systems has succeeded in running the GridGain cloud computing platform on Android phones Kharif [25], but this is still in early stages of development. The GridGain architecture is probably the closest to Hadoop's of all of the grid systems that are being targeted at mobiles. GridGain directly supports deployment on a cloud, and MapReduce is an important feature of the system.

Flechais [26], et al. describe AEGIS, a methodology for the development of secure and usable systems. AEGIS defines a development process and a UML meta-model of the definition and the reasoning over the system's assets. AEGIS has been applied to case studies in the area of Grid computing and we report on one of these.

As can be inferred from this section, much work has been accomplished on various fronts, yet still, much needs to be done. As can be seen, much of the work is carried out on systems wherein the Grid Middleware and Applications are tightly bound with the Operating System (platform). We are yet to see a truly platform independent solution, and that is the need of the hour.

III. CHALLENGES

Although, until now mobile and wireless devices have not been considered useful resources by traditional grid environments, considering the Metcalfe's law, (which claims that usefulness of a network-based system proportionally grows with the number of active nodes), and also considering that mobile device capabilities have substantially improved over the time, it can justifiably be stated that mobile and wireless devices are now of interest to be integrated in the grid [27]. The integration has not been achieved yet, however - not in a meaningful form anyway. This is mainly due to the consideration that current Grid middleware infrastructures don't support mobile devices, not only because (1) middleware have not been devised taking into account pervasive requirements like spontaneity, transparency, context-awareness, pro-activity, and so on; (2) middleware are still too much heavy with respect to computational capacities of mobile devices; (3) middleware are not network-centric; i.e.

they assume fixed TCP/IP connections and do not deal with wireless networks and other mobile technologies; and (4) middleware typically support OGSI interaction paradigm (SOAP based), whereas the Pervasive model requires a variety of mechanisms [28, 29].

While the field of mobile computing is growing quickly, there are still a large number of challenges that these devices face which remain open problems to be resolved [23]. If we scratch the surface a little, It may appear that the unification of mobile wireless consumer devices with high-performance grid computing might not be that simple. After all, grid computing to date has utilized multiprocessors and PCs as the computing nodes within its matrix. Consumer computing devices such as smartphones are typically restricted by reduced CPU, memory, secondary storage, and bandwidth capabilities, (as compared to desktop computers) increased power consumption sensitivity, increased heterogeneity, unpredictable long periods of complete disconnectivity, unreliable, low-bandwidth and high latency communication links; and very dynamic network layout because of devices entering and leaving in a very unpredictable manner [5, 8]. However, these are the challenges to be addressed. The resource-poverty of mobile devices (as compared to desktop computers), in one sense, is a result of a significant cost paid for mobility. A comparison of a Dell Inspiron 580 desktop with the iPhone 4, for example, reveals this compromise. As compared to a typical desktop computer, mobile devices in general have: 1/3rd processing power, 1/8th memory, 1/5th storage capacity and 1/10th network bandwidth. While mobile device performance will continue to improve in absolute terms (Figure 1), the inequality between the resource constraints of mobile and fixed devices will remain (or even increase) and thus, must be accounted for in the types of application selected for mobile computing. Limited computational power (of mobile devices) available to run applications seems the chief drawback. The limiting factor, however, is not so much the computational power but rather the battery power available to run the computations. While improvements in energy densities could fix this situation in the future, few other problems will, more or less, persist. Building distributed systems is notoriously complex and the problems posed by mobile distributed systems are the same as for any other distributed system, just an order of magnitude more complicated.

A. Hardware Challenges

The apparent resource-poverty (when compared with contemporary desktop computers) becomes, more or less, a non-issue when quantity of mobile devices outshines the quality. Thus, the most fundamental challenge arises from the fact that mobile devices are generally powered by batteries, whose capacity is fundamentally limiting [5]. Battery technology has matured slowly over the last decade and has failed to keep up with increased power demands from contemporary high-end mobile devices. That is, the growth of energy density in batteries is not matching the exponential growth in available computational power, memory, storage or networking bandwidth [30]. A cubic millimeter of current battery technology has enough energy to perform about 1 billion 32-bit computations, or send and receive 10 million bits of data [31]. One important thing to realize from this is that computation is cheaper than data transmission by orders of magnitude. One effective technique for addressing the limited computational power is load-balancing by pro rata based (with respect to power availability) task assignment / scheduling. Nonetheless the lack of battery power has a direct effect on the computational capabilities of these devices, thus efficient algorithms and programming techniques are of the utmost importance. The display element is one of the biggest contributors of energy use in a smartphone [31], thus, non-display applications (calculations and analysis) would likely be well suited for mobile grid computing. While research is being done to harness energy from other sources [32] it is unlikely that such technologies will resolve the issue any time soon. This severely affects the availability of mobile devices.

Apart from the physical limitations of hardware, they all run many different Operating Software, and there is no way to make all the hardware uniform with respect to their physical or logical aspects. The fundamental drawbacks of the hardware with respect to their diversity and limitations are not going to vanish in near future. Thus, there is a need to develop a platform independent (which does not bind tightly to any of the diverse operating systems that run on different hardware) grid system to at least partially overcome this Challenge.

B. Network Challenges

"8 fallacies of distributed computing" (7 outlined by Peter Deutsch and one added by James Gosling) depicts a very good illustration of networking challenges posed in front of mobile grid computing. The four most relevant of them deserves explanation here in the scope of present discussion. The first fallacy "The network is reliable" is particularly relevant in the wireless networking world where radio interference is the norm not the exception. Dropped calls are far from a thing of the past, and as devices move they are highly likely to experience network failure. Applications must be written with these disconnections in mind. The second fallacy is that "Latency is zero". This is a particularly prevalent fallacy in the Web 2.0 world with the rise of Asynchronous Javascript and XML (AJAX) type web applications. Latency is critical to responsive applications and this is hard to get right when designing AJAX based web applications. This is also a common problem with remote procedure call (RPC) interfaces and successors such as remote method invocation (RMI) type interfaces where the programming model attempts to hide the communication from the programmer. All too often application designers are tempted by these systems into making many small net- working requests instead of bundling them into one larger request. The problem with the bundling solution to the latency issue is the third fallacy: "Bandwidth is infinite." Thus a careful balance has to be struck. Latency and band- width are not the only concerns when it comes to networking though. There is also the cost of data transport. Perhaps the most important fallacy for mobile distributed computing is the seventh fallacy: "Transport cost is zero." The first interpretation is obvious in the Smartphone area of mobile computing where users may have to pay for every byte that they send over wireless networks. However, even if users have an unlimited data plan commonly provided with the purchase of a smart phone users still have to pay considerable energy for every byte sent. Moreover, it has already mentioned that data transmission is an extremely battery consuming process. This forces application designers to consider carefully every byte they send and its cost to the user not just in terms of money but also in terms of energy.

Apart from the above-mentioned fallacies, which address the reliability part of network connectivity, the efficiency and capacity are other facets of the problem. Multipath disturbances, power-signal degradation and intercellular hand-off, among others, are main culprits preventing mobile devices to achieve "high bandwidth & always-on" characteristics expected of grid nodes. Most wired LANs provide a minimum of 100Mbps and are moving quickly to 1Gbps. On the other hand, the fastest currently available wireless connection available is from a proprietary technology at 108 Mbps [5]. The wireless channels over which mobile nodes communicate can not easily provide as much of theoretical bandwidth as compared to a wired connection, due to the adverse conditions of the wireless channel (e.g., fading, shadowing, interference, collisions, etc) [3].

Network partitioning is another unique challenge in mobile networks. The main reason for network partitioning is the group mobility behavior, in which mobile nodes with similar mobility patterns could eventually form mobility groups exhibiting distinct movement patterns. In the worst cases, network partitioning would break a connected network topology into several separate, disconnected partitions. Further, unlike fixed broadband where a physical link supports consistent network bandwidth, wireless connectivity is characterized by variable data rates and intermittent connectivity due to gaps in coverage. The dynamic nature of application throughput demands, subscriber mobility and uncontrollable factors like weather can cause bandwidth capacity and coverage to vary. Moreover, mobile broadband networks generally have longer network latency than fixed broadband.

Since mobile nodes can join and leave the network at will, the connectivity of links among mobile nodes may vary with time leading to the unpredictability of the network topology [3]. In order to locate destination nodes they wish to communicate with, mobile nodes have to dynamically update their routing tables among themselves as they move around so as to refresh their global vision of the network. Unlike traditional Grid computing platforms, the network topology is neither stable nor predictable. The wireless connections between Grid nodes are normally bandwidth- limited and vulnerable to adverse signal quality.

C. Middleware Challenges

While directly connecting (hard-wired connection) mobile devices to the Grid is very straightforward and requires no modifications to the existing infrastructure, it defeats the very purpose of the mobile devices being 'mobile'. Further, the overall Grid performance of such grids would suffer significant downshift because of the high rate of failures. In case of any such failure, the Grid components would have to reschedule and reallocate resources for the active application, possibly migrating data around the Grid thus reducing the response time and increasing redundancy. Considering that in the mobile edge of the grid, the failure rate is bound to increase, this is not something expected in busy, heavily loaded and complex Grid environments. Another issue to consider is that design of original infrastructure (task size, complexity, communication model, load-balancing etc.) would be very different and far from optimized with respect to mobile workers, and thus would result in far from satisfactory performance gains. Thus, it becomes mandatory to develop/design specialized middleware/architecture for mobile Grids [8].

D. Resource Management

The intermittent availability of the Mobile Devices makes necessary the use of sophisticated mechanisms for resource discovery and selection [4]. The cache of the authorized resources available in the Grid infrastructure must be continuously updated. The selection of those resources that meet the qualifications (to execute the task) has to be based on deterministic criteria and stochastic parameters like resource accessibility, system workload, network performance, etc. resulting in a complex probabilistic models for the topology of the Grids and their capabilities. Of course, the complexity of the design must be in accordance with the intended use cases.

E. Job Management

Job Scheduling is classified as an NP-complete problem [33, 34] due to the fact that an algorithm has to allocate jobs to resources in an efficient and cost effective manner, so as to minimize the resource utilization and maximize output. Mobile Grid environment imposes many more constraints that would make the job scheduling problem even more complicated. The optimization criteria for a job scheduling mechanism Must consider the scalability of the overall system and contemporary availability & reliability of individual resource along with their cost and performance, to help select a specific resources for the execution of a specific job under the given QoS constraints.

Mobile grid (especially volunteer grids) are by nature of dynamic scale and potential. Thus, job migration and re-scheduling, as well as job replication and co-scheduling may both prove to be efficient ways to assure the completion of the jobs [4]. Job monitoring and estimation of task completion time are particularly difficult - if not impossible - in volatile framework such as mobile grid.

F. Security Challenges

Security is most important issue with mobile wireless transmission as it is susceptible to a wide range of attacks. Network layer security protocols, such as IPsec, and Transport layer security protocols, such as SSL, TLS, WTLS, provide sufficient protection between a wireless host and a trusted network. Further, security solutions such as WTLS have been optimized to address new challenges in mobile wireless computing to get rid of overhead computations. Yet, much work remains to be done, such as addressing the lack of security of 802.11 WEP and Bluetooth networks. [5] Further, security imposes bigger performance impact on the Mobile Device than that on a normal computer. The encryption/authentication mechanisms employed in desktop Grid implementation are not very economic with respect to computational use efficiency, and must be addressed accordingly for mobile platform [35].

The GGF OGSA Working Group has proposed a strategy for addressing security with OGSA [36]. According to the group, the security challenges faced in a Grid environment can be grouped into three categories: [37].

• Integration solutions where existing services need to be used, and interfaces should be abstracted to provide an extensible architecture,

• Interoperability solutions so that services hosted in different virtual organizations that have different security mechanisms and policies will be able to invoke each other, and

• Solutions to define, manage, and enforce trust policies within a dynamic Grid environment.

G. Why mobile owners would allow their devices to be incorporated for Grid?

As has been noted above, the computing power to price ratio of Mobile Devices is very poor when compared to specialize computing devices such as desktop computers (naturally, because Mobile Devices are not intended for intensive computational purposes). Thus, using dedicated Mobile Devices for Grid computing is not a reasonable option. The only way one might expect to utilize these devices for Grid computing is through 'volunteer computing' model. Herein lies the last, and perhaps most important, challenge to be addressed for accessing the above revered 'resource pool/potential' of networked Mobile Devices for Grid computing, viz. the permission of the device owner. No matter how many networked Mobile Devices exist in circulation with whatever mighty capabilities, they wont be available for Grid computing until their users allow them to participate.

There are various ways to encourage devise owners to allow their Mobile Device for Grid computing. The approach depends on who wish to consume the 'volunteer' computing services of the devices. If a network provider organization (telecom companies like AT&T) wishes to leverage the resources, they may encourage their consumers to volunteer their devices in exchange of perks/concessions on their services or even monetary benefits. Such an approach is not only within the reach of the service providers, but they are the ones who can accomplish it with best possible outcome. Other approach would be 'true' volunteer computing, where a user would 'donate' the idle computing resources after being convinced of the cause they are to be utilized for. Examples of such volunteer Grid computing (albeit, not in the realm of mobile Grid computing) are seti@home and numerous such projects, which have shown nothing less than remarkable as far as their success stories go.

H. Competing Technologies

Grid computing has traditionally been compared with cluster computing and recently with cloud computing. No doubt, both latter technologies have a few advantages over grid computing, but so does grid computing over them. Most important advantage the grid computing has over other technologies is, unlike grid, the other technologies deploy dedicated resources. On the flip side, the cluster and cloud computing can provide reliability, not a strong feature of volunteer type frameworks such as mobile grid. It is of course not the intention here to establish that volunteer (mobile) grid computing is the only possibility as far as grids are concerned. In fact, there are dedicated grids established and operating without volunteer facet to date. However, particularly in case of mobile grid computing, volunteer computing is the most efficient method of harvesting the idle computing resources, and hence a preferable paradigm.

IV. SUMMERY

The state of the art of the Mobile Grid Computing has been extensively depicted and thrust areas & possible solutions discussed in present survey, however, this survey will not be complete unless a use case is analyzed. Let us consider a University Campus in need of super-computing for student research purpose - say DNA sequencing. DNA sequencing is essentially the art of matching (sub)strings with a reference, as far as computing is concerned - not a significant task. However, the amount of raw data needed to be processed (usually in Giga Bytes) makes it a huge task (comprised of many individually insignificant ones). The individual string matching can very much easily be accomplished by a humble mobile device within a fraction of second, and thus a large number of such mobile devices may execute the seemingly huge task in acceptable time. Assuming the University has 5000 students, the potential volunteer devices may reach up to 7500 (1 smartphone per student and one laptop computer per every other student with average processing speed 1 GHz) connected with a campus-wide wi-fi network. If a sizable portion (say 2/3rd - 5000) of these devices is available for grid computing during working hours (7 AM to 7 PM - 12 hours) with a modest 50% duty cycle (up time), the computational potential of such a system may roughly equal to a cluster of 250 processors of 2.5 GHz each working 24 hours. This is of course a rough (and modest) estimation, but it serves to demonstrate the potential of the Mobile Grid Computing. Now adding the university desktop computers to this grid would even strengthen it, may be up to the extent that the University may delay or even drop the idea of commissioning a dedicated cluster entirely, or pump up computing power of their existing infrastructure.

Further, establishing mobile grids as depicted above has other advantages as well. First and foremost is the multifold decrease in the cost of installation per CPU cycle. Secondly, dedicated system will eventually fall short with respect to increasing demands, however, a volunteer mobile grid will evolve - so to speak - with time with respect to computational power due to inclusion of newer-faster devices and exclusion of inferior devices with student turnover. A rather important and most often overlooked advantage of such grids is their sustainable nature. Using idle computing power of existing devices is 'greener' approach instead of creating new resources. Now, with threats of global warming being more prominent then ever, efficient use of existing devices will help decrease demands from semiconductor Industry - a rather significant source of pollution. Finally, with the advent of HTML5, Mobile Grid computing has an opportunity to register unprecedented growth, as HTML5 along with JavaScript can provide a hardware and software independent (browser based) means of developing grid applications, in essence overcoming many of the aforementioned challenges.

In conclusion, we have discussed the 'issues' related to integration of mobile wireless consumer devices into the Grid (mobile grid computing) in this paper, accompanied with befitting commentary on topics such as the viability of the concept, followed by brief literary sketch of the state of the art. Challenges arising due to the limitations of mobile devices (e.g., reduced CPU performance, small secondary storage, heightened battery consumption sensitivity, and unreliable low-bandwidth communication), demographics, software challenges, security and permissions have been contemplated respectively and possible solutions proposed. We are currently in process of developing solutions to address some of the challenges described in this review, which we shall publish in due course.

ACKNOWLEDGMENT

Ketan Parmar wishes to acknowledge Mr. Devang Rughani of DevIndia InfoWay for the infrastructural support. The authors wish to acknowledge their gratitude towards "Wikipedia", "Open Source Software Community" and "Google" for their efforts in supporting Knowledge Sharing.

REFERENCES

[1] I. Foster, C. Kesselman and S. Tuecke, The anatomy of the Grid enabling scalable virtual organizations. Intl J Supercomputer Applications, 15 (2001), 200-222.

[2] I. Foster, C. Kesselman, J. M. Nick and S. Tuecke, The physiology of the Grid, an open Grid services architecture for distributed systems integration, Open Grid Service Infrastructure WG, Global Grid Forum, 2002. www.globus.org/alliance/publications/papers/ogsa.pdf. Accessed 22 July 2012.

[3] Li, L. Sun and E. C. Ifeachor, Challenges of Mobile Ad-hoc Grids and their Applications in e-Healthcare. Proc. of the 2nd International Conference on Computational Intelligence in Medicine and Healthcare, 2005.

[4] A. Litke, D. Skoutas and T. Varvarigou, mobile grid computing: Changes and Challenges of resource management in a mobile grid environment. Proc. of Practical Aspects of Knowledge Management, 2004.

[5] T.Phan, L. Huang and C. Dulan, Challenge: Integrating Mobile Wireless Devices into the Computational Grid. ACM MOBICOM, 2002.

[6] I. Foster and C. Kesselman, Globus: A Metacomputing Infrastructure Toolkit. Intl J Supercomputer Applications, 11 (1997), 115-128.

[7] G. Forman and J. Zahorjan, The Challenges of Mobile Computing. IEEE Computer, 27 (1994), 38-47.

[8] S. Isaiadis and V. Getov, Integrating Mobile Devices into the Grid: Design Considerations and Evaluation. Proc. of the 11th Int. Euro-Par Conference on Parallel Processing, (2005), 1080-1088.

[9] J. Hwang and P. Aravamudham, Proxy-based Middleware Services for Peer-to-Peer Computing in Virtually Clustered Wireless Grid Networks. Proc. of International Conference on Computer, Communication and Control Technologies, (2003).

[10] D.C.Marinescu, G. Marinescu, Y. Ji, L. Boloni and H.J. Siegel, Ad hoc Grids: Communication and computing in a power constrained environment. Workshop on Energy-Efficient Wireless Communications and Networks, (2003).

[11] A. Datta, MobiGrid: Peer-to-peer overlay and mobile ad-hoc network rendezvous - a data management perspective. the 15th Conference On Advanced Information Systems Engineering, (2003).

[12] Chlamtac and J. Redi, Mobile Computing: Challenges and Potential, Encyclopedia of Computer Science, 4th edition, (1998).

[13] M. Franz, A Fresh Look at Low Power Mobile Computing. http://research.ac.upc.es/pact01/colp/paper15.pdf

[14] B. Chen and C.H. Chang, Building Low Power Wireless Grids. http://www.ee.tufts.edu/~brchen/pub/LowPower_WirelessGrids_1201.pdf

[15] D. Bruneo, M. Scarpa, A. Zaia and A. Puliafito, Communication Paradigms for Mobile Grid Users. IEEE/ACM International Symposium on Cluster Computing and the Grid 669, (2003).

[16] S. Kurkovsky, Bhagyavati and A. Ray, A Collaborative Problem-Solving Framework for Mobile Devices. ACMSE, (2004), 5-10.

[17] S. Kurkovsky and Bhagyavati, Modeling a Computational Grid of Mobile Devices as a Multi-Agent System. Proc. of the 2003 International Conference on Artificial Intelligence, (2003).

[18] L.W. McKnight, J. Howison and S. Bradner, Guest editors' introduction: Wireless Grids–distributed resource sharing by mobile, nomadic, and fixed devices. IEEE Internet Computing, 8 (2004), 24–31. doi: http://doi. ieeecomputersociety.org/10.1109/MIC.2004.14

[19] S.P. Ahuja and J.R. Myers, A survey on wireless Grid computing. Jurnal Supercomputer, 37 (2006), 3–21. doi: http://dx.doi.org/10.1007/ s11227-006-3845-z

[20] D. Chu and M. Humphrey, Mobile OGSI.NET: Grid Computing on Mobile Devices, (2004). http://www.cs.virginia.edu/~humphrey/papers/MobileOGSI.pdf.

[21] The AKOGRIMO project: http://www.akogrimo.org

[22] Legion, World Wide Virtual Computer, http://legion.virginia.edu/

[23] N. Palmer , R. Kemp , T. Kielmann and H. Bal, Ibis for mobility: solving challenges of mobile computing using grid techniques. Proc. of the 10th workshop on Mobile Computing Systems and Applications, (2009) 1-6. Doi:10.1145/1514411.1514426

[24] W. Chou and Li Li, WIPdroid - a two-way web services and real-time communication enabled mobile computing platform for distributed services computing. In SCC '08: Proceedings of the 2008 IEEE International Conference on Services Computing, (2008) 205–212. doi: http://dx.doi.org/10.1109/SCC.2008.113

[25] O. Kharif, A warm welcome for android. BusinessWeek.

[26] Ivan Flechais , Cecilia Mascolo , M. Angela Sasse, Integrating security and usability into the requirements and design process, Intl J Electronic Security and Digital Forensics, 1 (2007), 12-26. Doi:10.1504/IJESDF.2007.013589

[27] L. W. McKnight, J. Howinson and S. Bradner, Wireless Grids. IEEE Internet Computing, (2004).

[28] G. Coulson, P. Grace and G. Blair, D. Duce, C. Cooper and M. Sagar, A Middleware Approach for Pervasive Grid

Environments. Workshop on Ubiquitous Computing and e-ResearchNational eScience Centre, (2005).

[29] M. Ciampi, A. Coronato and G.D. Pietro, Middleware Services for Pervasive Grids M. Guo et al. (Eds.): ISP A (2006), 485 – 498.

[30] W. Lehr and L. McKnight, Wireless Internet access: 3G vs. WiFi? Telecommunications Policy 27, 5-6, (2003) 351–370.

[31] D. Estrin, D. Culler, K. Pister and G. Sukhatme, Connecting the Physical World with Pervasive Networks. IEEE PERVASIVE COMPUTING (2002) 59–69.

[32] J. Paradiso and T. Starner, Energy Scavenging for Mobile and Wireless Electronics. IEEE PERVASIVE COMPUTING, (2005)18–27.

[33] R. Moreno, Job scheduling and Resource Management Techniques in Dynamic Grid Environments, 1st European Across Grids Conference, (2003).

[34] A. Abraham, R. Buyya and B. Nath (2000) Nature's Heuristics for Scheduling Jobs on Computational Grids. International Conference on Advanced Computing and Communications.

[35] Y. Wen, Mobile Grid - Major Area Examination, Department of Computer Science University of California Santa Barbara, CA 93106.

[36] I. Foster, C. Kesselman, G. Tsudik and S. Tuecke, Proc. A Security Architecture for Computational Grids. 5th ACM Conference on Computer and Communications Security Conference, (1998) 83–92.

[37] Y. Xiao, State-of-the-Art Security in Grid Computing. in Security in distributed, grid, mobile and pervasive computing. Auerbach, New York, (2007) pp 207-236.