# Should one incorporate Mobile-ware in Parallel and Distributed Computation?*

Mustafa Sanver, Sathya Priya Durairaju, Ajay Gupta
Department of Computer Science, Western Michigan University
Kalamazoo, MI 49008, USA
{msanver, spdurair}@cs.wmich.edu, ajay.gupta@wmich.edu

## Abstract

In this short paper, we consider issues of integration of mobile devices into grid computing thereby extending the discussion started by Phan et al. in [9]. The dynamic nature of mobile devices increases the cost of integration. We first show via a collection of different surveys that the integration of mobile-ware in the parallel computation is worth the effort to benefit from its aggregate computational power. Main issues that must be addressed are then identified. We also propose possible solutions to some of the issues, such as task, grid and mobile device allocation, buffering, deployment of agents and possible models to convince mobile device owners of the benefits achieved by the integration.

## 1. Introduction

Mobile devices have gained acceptance in the daily life of individuals and organizations all over the world as wireless technology is developed. In this paper, the term *mobile device* refers to wireless devices ranging from lightweight personal digital assistances to low-end and high-end laptops. Increases in the number and power of mobile devices on the market are significant in terms of aggregate computation power. Naturally, the question of integration of mobile devices in distributed computations arises.

In this paper, we identify main issues that must be dealt with if mobile devices were to be integrated into parallel and distributed computing. We discuss possible solutions to some of the main issues of such an integration. We suggest a hierarchical model to facilitate the integration of mobile-ware. In the proposed model not only the performance and the computation power but also the power consumption [8] of mobile-ware is considered. The dynamic and power-hungry nature of mobile-ware increases the cost of integration. Thus, the integration of mobile-ware introduces different issues in parallel and distributed computation. In section 2, we address the question: whether degradation by the integration is worth the effort to integrate mobile devices into the computational grid. After introducing our hierarchical dispatcher model, we elucidate the issue of task allocation to mobile devices in section 4. Intermittent buffering to ameliorate extra cost and computation due to weak connectivity and periodic or

*Research is supported in part by the National Science Foundation under grants ACI-0000442 and ACI-0203776, by the Western Michigan University under FIE02 grant, by the Department of Education grant R215K020362, and a Congressional Award, administered by the US Department of Education, Fund for the Improvement of Education.

frequent disconnections of the mobile devices is addressed in section 5. Finally, section 6 sheds some light on potential economic models to encourage owners of mobile devices to participate in the grid.

## 2. Mobile Devices

Before building a model for the integration, one must address the question if the integration is feasible and worth the effort. Let us consider market statistics on the proliferation of mobile devices and the associated technologies.

### 2.1 Survey about mobile devices

The number of individuals using mobile devices and companies considering switching from wired networks to wireless networks are significant [1]. Some surveys supporting these facts are depicted below. From 2001 to 2005, investment on mobile devices is expected to increase by 41% and reach $31 billion. By 2004, the laptops on the market rises to 39.7 millions. Access point shipment worldwide increases more than 107% of 2003.

**Table 1** Worldwide wireless LAN equipment shipments (1000s of units)

| **Product Segment** | **2001** | **2002** | **2003** | **2004** | **2005** | **2006** | **2007** |
|---|---|---|---|---|---|---|---|
| Adapters | 6,890 | 12,599 | 21,333 | 30,764 | 41,417 | 50,415 | 56,931 |
| Access Points | 1,437 | 1,965 | 3,157 | 3,919 | 4,851 | 5,837 | 6,555 |
| Broadband Gateways | 552 | 850 | 1,906 | 3,365 | 5,550 | 7,941 | 9,472 |
| Other Wireless LAN Equipment | 47 | 59 | 82 | 105 | 132 | 158 | 179 |
| **Total** | **8,926** | **15,473** | **26,478** | **38,153** | **51,950** | **64,351** | **73,137** |

*Source: Gartner Dataquest (July 2002)*

### 2.2 Mobile Device Technology

Battery, storage, memory, bandwidth, and security are major bottlenecks affecting mobile computation metrics. Battery discharge time and its life, need to plug in the mobile devices to recharge, current limit of memory cards, demand for a higher memory throughput, more and faster bandwidth requirement, and security are some of the current limits of mobile world. We suggest some alternative solutions to these deficiencies such as biofuel cell [4] for battery replacement, wire-free electric power [5] support, a power policy [8], near future memory cards [2] and multi-level cell memories [3], ultra-wide band (UWB) [11], and under development security software (Wired Equivalent Privacy WEP, Wireless Protected Access WPA etc.) [7].

At first glance, an individual mobile device may not have sufficient capacity and computation power for the integration. However, if we can harness the a*ggregated mobile power* instead of individual power and consider exponential-rise of mobile units marketed and significantly evolving mobile technology, then one may conclude that it is worth the effort using this vast, pervasive, and untouched computational source for parallel computation. Obviously, we imagine that numerous lightweight tasks and/or mobile agents will be executed asynchronously and in parallel on the mobile devices to accomplish a task.

## 3. Hierarchical Dispatcher Model

We propose a solution, namely hierarchical dispatch model to incorporate mobile-ware into the computational grid as an extension to the Proxy-based architecture discussed by Phan et al. in [9]. In our proposed hierarchical model (Figure 1), the computational grid communicates with a master dispatcher (M-dispatcher), which is a proxy server. This M-dispatcher communicates with a set of access point dispatchers (AP-dispatcher), which are located near the wireless access points and communicate with the mobile devices (m-devices or minions).

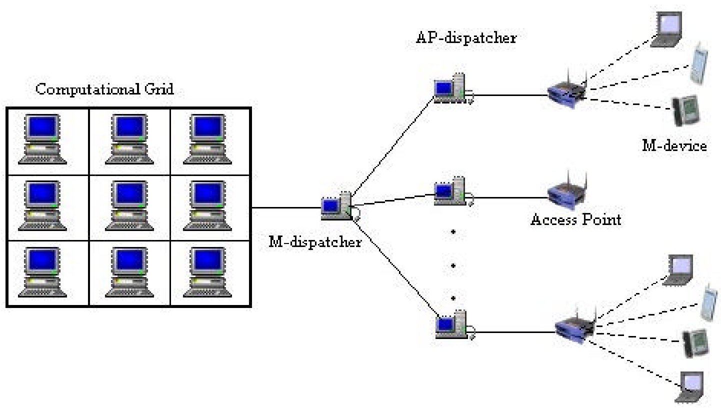


**Figure 1.**

In the above model, all mobile devices can be made transparent to the grid. The hierarchical model reduces congestion at the middleware by distributing the work among the hierarchies thus alleviating some load and improving reliability, adaptability, and fault-tolerance.

Some issues must be further investigated, such as, extra overhead introduced because of addition of M-dispatcher and AP-dispatcher hierarchy; congestion at M-dispatchers, storage requirement at AP-dispatchers for intermittent buffering (explained later), extra computation time for buffering, data exchange between the hierarchies, etc.

AP-dispatchers store some information about minions such as a unique ID, OS, bandwidth, memory, battery life, power consumption, and disk space. A connection history can be logged to profile availability and duration of m-device connections. This information is necessary to compute the available computation power of a device (m-power). M-dispatcher maintains information about AP-dispatchers and estimates their aggregated m-power to coordinate the activities of AP-dispatchers, such as task allocation and load balancing.

## 4.Allocation

Allocation is partitioning of data and workload into processes and mapping the processes to the nodes (processors/grid members) including m-devices. Allocation is one of the main issues arising from the problem of incorporation of mobile devices into grid computing. In the hierarchical model, there are two types of allocations: *grid allocation* and *m-device allocation*.

### 4.1 Grid Allocation Policy

Grid allocation is partitioning of initial workload in the grid and sending a portion to the M-dispatcher. There are three different ways to do the allocation in this level. First, the M-dispatcher is one of the grid members and receives the task from other grid members except the grid controller. Second, the M-dispatcher is one of the grid members and receives the task from the grid controller. We identify 3 different policies based on the size of the partition received: equal, bigger, the entire task. The M-dispatcher could compute itself or send the task to the AP-dispatchers. Third, the M-dispatcher is a middleware or proxy server versus a grid member. The M-dispatcher sends the request to the AP-dispatchers without trying to compute it itself. If M-dispatcher can't compute and there is no adequate m-power for the distribution, it simply rejects the request and returns an error to the requester.

### 4.2 M-device allocation

This allocation takes place at the mobile device level. There are two types of allocation based on the transparency of m-device information to the grid. First, all m-device information is hidden from the grid. AP-dispatchers partition the data parallel type of task into subtasks and send them to available and adequate m-devices. M-devices are logically grouped or *partner*ed to improve reliability. Partitioning will be based on each m-device's m-power, battery life and power consumption ratio. This solution is adequate for the problems having their input from an external data source, such as a file. As an alternative, a middleware can be used for allocation, such as Globus [10]. Second is the allocation where some of the information about m-devices is visible to either the programmer or the grid. Thus, m-device allocation is not fully controlled and executed by AP-dispatchers. Two suggestions to do the allocation are that programmer considers m-device allocation in the code and M-device Id's can be added to the *machinefile*. Second suggestion requires knowledge of user connection patterns and making efficient use of them.

## 5. Intermittent Buffering

Mobility causes intermittent connections. AP-dispatcher maintains intermediate computation results of m-devices in order to reduce the computation time that might be incurred when an m-device leaves during a computation. Re-computation of part of an unfinished job increases inefficiency of the integration of m-devices. We consider buffered intermediate results to eventually finish the computation by assigning the job to the partner m-device along with intermediate results. The intermediate results can be saved either after every update made to the results or periodically. Both situations need extra programming efforts to include milestone points in the code and need support for at least a limited process check pointing. This issue needs further investigation.

## 6. Convincing M-device Owners

Most of m-device owners are individuals rather than organizations. So, it is important to convince the owners to allow their m-device to participate in the grid computation. We suggest followings models to convince the owners: *Power Consumption:* Our suggestion to reduce this deliberation is to increase the availability of wire-free electric bases in the facilities. *Security:* This issue can be resolved if the security is increased, privacy is guaranteed by policies, parallel programs and all client agents are robust, and code verifications and third-party verifications may be adapted. *Economical model*: Suggestions are to attract m-device owners by reducing Internet connection cost for the m-device owners by ISPs and providing free or cheaper services and utilities.

## 7. Conclusions

In this paper, we introduced a hierarchical model as a possible solution to integrate mobile devices into grid computing. Increases in the number of mobile devices on the market and computation power with evolving mobile technology create an untouched source for computation and resource. We strongly argue that integration effort is worth further investigation. Our hierarchical model reduces congestion and facilitates load balancing. We elucidated grid allocation and m-device allocation, intermittent buffering of the intermediate results, and convincing mobile device owners to participate to the grid computing. MPI [6], PVM, Globus [10], etc. can be extended to support mobile devices. Primitives to support for M-dispatcher, AP-dispatcher, and m-devices need to be designed. We have, by no means, tried to address all the issues of integration of mobile-ware into parallel and distributed computing. Our goal in this short paper was to motivate the researchers to further investigate the issues by alluding to a few important issues and proposing an architecture.

## References


[1] Ashley, P., Hinton, H., Vandenwauver, M., *Wired versus Wireless Security: The Internet, WAP and iMode for E-Commerce*. http://www.computer.org/proceedings/acsac/. 17th ACSAC'01 December 10 - 14. New Orleans, Louisiana.

[2] Clove Technology. *Memory*. 2000. http://www.clove.co.uk/. Accessed on April 3, 2003.

[3] Intel Corporation. 2003. http://www.intel.com/design/flash/. Accessed on April 9, 2003.

[4] Johnson, C. R. *Bio-fuel cell can run on ethanol and enzymes*. April 7, 2003. http://www.eetimes.com/. Accessed on April 10, 2003.

[5] MobileWise. *Wire-free electric base*. 2003. http://www.mobilewise.com/.Accessed on April 04, 2003.

[6] MPI. *The Message Passing Interface (MPI) standard*. http://phase.etl.go.jp/mirrors/mpi/mpich/. Accessed on January 2003.

[7] Network Magazine. *XML:The End of Security Through Obscurity?.* April 2003. Volume 18. No 4.

[8] Newman, M., Hong, J., *A look at Power Consumption and Performance on the 3Com Palm Pilot*. 1998. http://guir.cs.berkeley.edu/projects/p6/finalpaper.html.

[9] Phan, T., Huang, L., Dulan, C.*Challenge: Integrating Mobile Wireless Devices Into the Computational Grid.* MOBICOM'02. September 23-26, 2002.

[10]The Globus Project. *The Globus Project, FAQ*. May 2, 2002. http://www.globus.org. Accessed on April 04, 2003.