Challenge: Integrating Mobile Wireless Devices Into the Computational Grid [*]

Thomas Phan
Pervasive Computing Laboratory
Computer Science Department
The University of California, Los Angeles
Los Angeles, CA 90095
phantom@cs.ucla.edu

Lloyd Huang
Semiconductor Manufacturing
International Corp.
Shanghai, China 201203
lloyd_huang@smics.com

Chris Dulan
Vizional Technologies, Inc.
Santa Monica, CA 90405
cdulan@vizional.com

ABSTRACT

One application domain the mobile computing community has not yet entered is that of grid computing – the aggregation of network-connected computers to form a large-scale, distributed system used to tackle complex scientific or commercial problems. In this paper we present the challenge of harvesting the increasingly widespread availability of Internet-connected wireless mobile devices such as PDAs and laptops to be beneficially used within the emerging national and global computational grid. The integration of mobile wireless consumer devices into the Grid initially seems unlikely due to the inherent limitations typical of mobile devices, such as reduced CPU performance, small secondary storage, heightened battery consumption sensitivity, and unreliable low-bandwidth communication. However, the millions of laptops and PDAs sold annually suggest that this untapped abundance should not be prematurely dismissed. Given that the benefits of combining the resources of mobile devices with the computational grid are potentially enormous, one must compensate for the inherent limitations of these devices in order to successfully utilise them in the Grid. In this paper we identify the research challenges arising from this problem and propose our vision of a potential architectural solution. We suggest a proxy-based, clustered system architecture with favourable deployment, interoperability, scalability, adaptivity, and fault-tolerance characteristics as well as an economic model to stimulate future research in this emerging field.

Categories and Subject Descriptors

C.2.4 [**Computer Systems Organisation**]: Distributed Systems; D.2.11 [**Software**]: Software Architectures; J.7 [**Computer Applications**]: Computers in Other Systems—*Consumer products*; K.4.4 [**Computing Milieux**]: Electronic Commerce—*Distributed commercial transactions*

General Terms

Design, Economics

Keywords

Grid computing, mobile wireless computing, pervasive computing, network clusters, economic model

1. INTRODUCTION

In recent years the mobile computing community has been successful in utilising academic and industry research efforts to bring products to the commercial market. We have seen a proliferation of consumer electronic devices taking advantage of wireless technology enrich our daily lives with increased productivity thanks to higher connectivity. When one considers the broad range of wirelessly connected mobile devices used today, from 802.11-connected laptops to personal digital assistants (PDAs) with cellular data modems, it is clear that such network-enabled devices will continue to be increasingly important and widely used.

Although these devices play vital roles in personal and business productivity, one application domain the mobile computing community has not yet entered is that of *grid computing* – the utilisation of aggregate computing resources for computational distribution. In this paper we present the challenge of harvesting the increasingly widespread availability of Internet-connected wireless mobile devices to be beneficially used within the emerging national and global computational grid.

Grid computing [15] [2] is an important developing computing initiative that involves the aggregation of network-connected computers to form a large-scale, distributed system used to tackle complex scientific or commercial problems. By spreading workload across a large number of computers, the grid computing user can take advantage of enormous computational, storage, and bandwidth resources that would otherwise be prohibitively expensive to attain within traditional multiprocessor supercomputers. Previous grid computing efforts have included computation for the Search for Extraterrestrial Life project, AIDS research, the Human Genome Project, molecular visualisation, and multimedia content distribution. Leading work by the academic Globus research effort, along with industry support from companies like IBM, HP, and Sun, will raise grid computing to a global scale.

At first glance, it seems that the marriage of mobile wireless consumer devices with high-performance grid computing would be an unlikely match. After all, grid computing to date has utilised multiprocessors and PCs as the computing nodes within its mesh. Consumer computing devices such as laptops and PDAs are typi-

[*] This research was supported by National Science Foundation grant EIA-0103764, “Performance Driven Adaptive Software Design and Control.”

MOBICOM’02, September 23–26, 2002, Atlanta, Georgia, USA.
Copyright 2002 ACM 1-58113-486-X/02/0009 ...$5.00.

cally restricted by reduced CPU, memory, secondary storage, and bandwidth capabilities. However, therein lies the challenge. The availability of wirelessly connected mobile devices has grown considerably within recent years, creating an enormous collective untapped potential for resource utilisation. To wit, recent market research shows that in 2001, 28 million laptop PCs [16] and 13 million PDAs [17] were sold worldwide. Although these individual computing devices may be resource-limited in isolation, as an aggregated sum, they have the potential to play a vital role within grid computing.

We believe it would be remiss of researchers to discount the enormous potential benefit to be gained from utilising this vast number of devices on the Grid. The fact that mobile devices represent an already large – and growing – percentage of available worldwide computing power should be leveraged by researchers in order to find ways to harness this abundance. As mobile devices' CPU performance and wireless connectivity both continue to grow, the argument in favour of using such devices is strengthened as well, making research now in identifying and potentially addressing fundamental challenges important. Therefore, in this paper we pose two central questions. First, given that the benefits of integrating the resources of mobile devices like laptops and PDAs into the computational grid are potentially enormous, how can we compensate for the inherent limitations of these devices in order to successfully utilise them in the Grid? Second, how can we present a compelling case to already-skeptical owners of these devices in order to persuade them to contribute their devices to the Grid?

The purpose of this paper is to promote further thought into this challenge; we believe a useful way of doing so is to provide a sample solution which we can evaluate to gain further insight. Specifically, we suggest a proxy-based clustered infrastructure solution to provide mobile computing devices with favourable deployment, interoperability, scalability, adaptivity, and fault-tolerance characteristics. In our design we create groups of mobile devices that are clustered around a nearby device serving as their proxy. Unlike contemporary mobile ad hoc routing approaches that also utilise clustering around a proxy or gateway node, in our design the proxy additionally serves the important roles of service negotiator and resource request partitioner. Additionally, to motivate users to take part in our architecture, we turn to elements of game theory and suggest a stimulus model that provides mutual benefits for the device owners, the service providers, and the grid computing users, as well as a faster return-on-investment for all parties involved.

The work presented here lays the foundation for emerging research. The Leveraging Every Existing Computer out tHhere, or LEECH, project will investigate the use of consumer devices as nodes in the computational grid. Like its blood-sucking parasitic namesake, the LEECH is meant to siphon computational resources from participating nodes in order to contribute capabilities to the grid. Such an investigation is timely and will complement the growing field of pervasive/ubiquitous computing [35].

This paper is organised in the following manner. In §2 we briefly describe the computational grid along with applications that can benefit from this architecture. We discuss the problems endemic in the integration of mobile devices in the Grid in §3 and offer our LEECH technical and economic infrastructure to support our vision in §4. We conclude our paper with future plans in §5.

2. GRID COMPUTING

2.1 Background

Grid computing has its roots within the field of high-performance parallel computing, which has traditionally been successful on massively parallel processor (MPP) systems designed following a NUMA or UMA architecture. Such MPPs have utilised multiple CPUs within a single chassis to produce higher performance manifest through increased throughput. However, such systems become prohibitively expensive for large CPU configurations.

Three different approaches have emerged within the last decade that provide alternatives to the MPP platform. Local-area Networks of Workstations (NOW) [1] take advantage of clusters of uniprocessor workstations connected via a network such as Myrinet or Ethernet. Taking advantage of such commodity parts, NOWs can provide high performance at low cost. For example, Beowulf systems [3] look to leverage low-cost, high-performance Linux PCs with commodity networking. Additionally, Condor [26] provides the capability to share processing jobs across a Unix NOW to achieve load-balancing.

At a much larger scale of distribution, metacomputing [32] [21] links geographically diverse supercomputing resources via a high-speed network. This conglomeration of gigaFLOP-capable centers into a teraFLOP-capable virtual ubercomputer can yield vastly increased performance for applications that can take advantage of this architecture. Work using this architecture focused on computationally-intensive tasks that could be naturally distributed, such as dynamic macromolecular visualisation and meteorological prediction.

The third approach, grid computing, emerged directly from the metacomputing concept but has now morphed into a resource-sharing paradigm akin to the current peer-to-peer concept. Whereas contemporary peer-to-peer applications such as the commercial Napster, Gnutella, and KaZaa programs concentrate on file distribution, current grid computing generalises the peers' available resources to include computation, bandwidth, and storage.

2.2 Grid Computing Infrastructure

Much pioneering work in grid computing was done with the Legion [20] and Globus [13] [15] [19] research efforts. Globus has emerged as the middleware standard for a number of different grid projects and provides a 4-layer stack to control hardware, communications, resource sharing, and collective coordination.

Grid computing has attained prominence of late thanks to the importance of several grid facilities intended for computation- and data-intensive scientific work. Some facilities currently in operation or opening in the intermediate future include the Department of Energy Science Grid, the NSF Distributed Terascale Facility, the Particle Physics Data Grid, the NASA Information Power Grid, the North Carolina Bioinformatics Grid, and the UK National Grid.

The commercial sector has also begun seeing the importance of this emerging architecture. Companies developing grid computing infrastructures include IBM, HP, Platform Computing, and Sun.

2.3 Grid Applications

Unlike applications intended to be run on tightly-coupled multiprocessors, grid-enabled applications cannot depend on low-latency communication between processing entities due to the inherent high latency of distributed networks. Instead, recent applications have leveraged the computation capability of nodes but do not require much communication. With such applications, the problem dataset is decomposed and distributed across the processing nodes.

Perhaps the most famous of these applications is the radio signal analysis program used for the Seti@home [31] project. Provided to device owners as either a screen saver or stand-alone application, it processes data in a disconnected fashion: 0.25 Mbytes of data are received from a server, the application proceeds to analyse the data offline, and the results are sent back. Other applications in a similar vein include projects for the Human Genome Project [10], AIDS research [11], encryption challenging [8], pharmaceutical drug design [5], and stock market prediction [30]. Some companies, such as Entropia and DataSynapse have emerged within the last few years focused on distributed computation.

Other uses for grid computing have also been developed. Companies like AllCast, Uprizer, and Kontiki have utilised grid-like infrastructures for multimedia content distribution. Furthermore, distributed computing is beginning to make its way into general e-commerce. Distributed clustering has previously been utilised to achieve fault-tolerance, but now its use to boost performance of web services during peak periods has gained wider acceptance.

3. MOBILE DEVICES AND THE GRID

In the previous section we described the attractiveness of contemporary grid computing using always-connected computers like desktop PCs. We suggest that the next logical step in expanding the Grid lies with the use of heterogeneous mobile consumer devices connected through a potentially unreliable wireless network. In this section we present two opposing views to this assertion.

3.1 A Baseline Acronym

We first establish the useful acronym of BASELINE to represent Barely Adequate Systems Leveraging Internet NEtworking. We shall refer to Baseline devices as the collective family of mobile laptop computers, PDAs, and perhaps future Internet appliances that can contribute limited resources, such as CPU cycles or storage. These devices communicate through the Internet using possibly high-latency, low-bandwidth, unreliable network channels; in this paper we concentrate on wireless communication. These machines are fundamentally different from their desktop PC and supercomputer brethren, which are rich in CPU, storage, memory, and communication resources.

3.2 The Case Against Baseline Units on the Grid

A number of problems hinder the use of contemporary Baseline devices on the Grid. For PDAs, hardware and OS heterogeneity issues are pervasive as Palm and PocketPC compete aggressively for market share. Mobile computing devices are also well-known for inherent disadvantages [12], such as slower CPUs, unreliable low-bandwidth wireless connectivity, unpredictable extended periods of complete disconnectivity, heightened power-consumption sensitivity, software noninteroperability, small secondary storage, and security.

In an ideal world, wireless networks would provide as much bandwidth and work as reliably as wired connections. Unfortunately, real world conditions such as multipath disturbances, power-signal degradation, and intercell hand-off, among others, do not facilitate the high bandwidth, always-on characteristics expected of Grid nodes. Present grid computing applications typically target idle desktop PCs that receive portions of a larger problem, perform computation, and return results within bounded time. Unreliable connectivity and prolonged periods of intended disconnectivity break this expectation. Even when connectivity is not at issue, present wireless technology cannot provide the high bandwidth typical of wired connections. 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.

Other problems are prevalent. Battery technology has matured slowly over the last decade and has failed to keep up with increased power demands from contemporary PDAs and laptops. Recent developments in lithium polymer replacements for lithium ion show promise in this field [9]. Little to no investment has been made in developing software that support Baseline devices on the Grid, resulting in such problems as Grid integration, service discovery, and application-level interoperability. In terms of secondary storage, the limitation of flash memory in handhelds is a major factor against using Baseline devices in the Grid. Applications need storage to place temporary and permanent data for reuse or aggregation, but contemporary PDAs typically come with only 32 MB of memory. Perhaps the use of micro hard drives, such as IBM's 1GB Microdrive, will become more prevalent in the near future. This, however, adds to the higher power requirements of the device. Finally, security is always an issue with mobile wireless devices since wireless transmission is susceptible to a wide range of attacks. From the network's perspective, it is appealing to have end-to-end protection where wireless traffic is protected unless the device itself is compromised. Network layer security protocols, such as IPsec, readily provide qualitative protection between a wireless host and a trusted local area network or a trusted host. Transport layer security protocols, such as SSL, TLS, WTLS, provide similar protection for user sessions. In addition, many security solutions have been customised to address new challenges in mobile wireless computing. For example, WTLS uses potentially more CPU-efficient elliptic-curve cryptography to reduce computation overhead [37]. Nevertheless, much work remains, such as addressing the the lack of security of 802.11 WEP and Bluetooth.

3.3 The Case For Baseline Units on the Grid

In addition to the contemporary technological issues just presented against the use of Baseline devices on the Grid, a number of other socio-economic problems become evident. In this subsection we raise these issues in turn and address them directly.

First, it may not be immediately clear why one would even consider the use of such resource-constrained Baseline devices on the Grid at this time, especially when only a small fraction of Internet-connected desktop PCs currently contribute to the Grid and Grid-like applications. The argument for including Baseline devices in the Grid is anchored by the sheer weight of numbers. The ubiquity of computing devices in people's pockets and briefcases has potentially become a vast new source of processing power. According to Gartner Dataquest, a market research firm, a projected 15.5 million PDAs will be shipped worldwide in 2002 [17], an 18% increase over the 13 million worldwide shipments in 2001, which itself saw an 18.3% growth over 2000. The recent economic downturn as well as market saturation have clearly stunted the *114%* PDA shipment growth of 2000 over 1999. Of those sold annually in 2001, between 975,000 to 1.6 million were devices that run Microsoft's PocketPC operating system. In Table 1 we list the hardware specifications of some contemporary Pocket PC PDAs. (At the time of this writing, we chose to omit PalmOS PDAs because the current generation of such PDAs running on the Motorola Dragonball CPU are unable to perform floating-point arithmetic in hardware, essentially negating their potential utility. However, Palm has recently announced that their next generation of PDAs will utilise the ARM family processor design, an encouraging development.) As can be seen from the Table, the raw processing power of the handhelds is not trivial given their mobility.

The argument for laptop PCs follows more intuitively. Infor-

System	Entry year	CPU	Storage	Connectivity
Casio Cassiopeia E-125	2000	150 Mhz NEC VR4122	32 MB RAM, Compact Flash TII	56K modem via CF
Compaq iPAQ 3650	2000	206 Mhz Intel StrongARM	32 MB RAM, Compact Flash TII expansion	56K modem via CF
HP Jornada 548	2000	133 Mhz Hitachi SH3	16 MB RAM, Compact Flash TI	56K modem via CF
Compaq iPAQ 3975	2002	400 Mhz Intel X-Scale	64 MB RAM, Secure Digital Card	Built-in Bluetooth

Table 1: System specifications for contemporary personal digital assistants. Most Pocket PC PDAs can further add on a PCMCIA slot allowing them to utilise such products as 802.11 cards.

	Bluetooth	802.11a	802.11b	802.11g	Atheros	HomeRF	Ultra-Wideband	former Metricom	Verizon	3G
Bandwidth	1 Mbps	54 Mbps	11 Mbps	22 Mbps	108 Mbps	10 Mbps	100 Mbps	128 kbps	144 kbps	2 Mbps
Range	10 metres	50 metres	100 metres	100 metres	33 metres	50 metres	10 metres	cellular	cellular	cellular

Table 2: Wireless LAN and cellular technologies. Sources: RHR, IBM, Verizon, Metricom, and The Economist.

mally speaking, we observe that laptops are typically 0.75 generations behind desktop PCs in terms of CPU speed and storage capacity; at the time of this writing, 1.7 Ghz CPUs are now available in high-end laptops. Market research showed that in 2001, 22 percent of the 128 million PCs sold worldwide were laptops, a percentage figure that has grown by 1-2 points each year since 1999 [16]. A user who owns a Baseline device can wirelessly connect to the Internet and potentially to the Grid by using any of the current or emerging wireless LAN or cellular standards shown in Table 2. The emergence of new products utilising the 3G standards CDMA2000 or WCDMA will only further strengthen the argument in favour of inclusion. An evaluation of the potential aggregate power of these machines is indeed compelling.

We add to our argument by considering four trends we believe will be prevalent in the future: (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 for the handheld market bode well for future PDAs. (2) Wireless communication will grow as well for both local-area (using 802.11, Bluetooth, or Ultra Wideband) as well as wide-area (using 3G technology or perhaps ad-hoc meshes of wireless LANs). (3) Battery efficiency will *not* substantially improve. (4) Grid applications will be more widely used.

We firmly believe that careful anticipation of such future developments will lead to better preparation for later research down the road. The time is ripe to start investigating the use of Baseline devices for the Grid, due largely to the expected development of mobile processors and wireless communication of the first two trends. An architecture will be needed to mitigate the third trend of limited battery efficiency as well as to address issues of availability, interoperability, security, and network latency. Finally, all of this is in favour of meeting the potential widespread adoption of Grid technology as stated in the fourth trend.

Hence, although the use of these resource-poor Baseline machines can be considered much too premature given the current state-of-the-art in mobile technology, we posit it is exactly this stage, when an upcoming, potentially important technology clearly emerges, that requires thought to be invested. While the infrastructure is not currently ideal, we are providing a small glimpse into what a future grid of completely heterogeneous machines can look like. By identifying the key technological design issues now, we lay the foundation for future research.

As a second issue against Baseline devices on the Grid, it may be argued that research in this area should wait until such devices gain sufficiently powerful CPUs and other resources so that their contribution is more meaningful. Unfortunately, there will always be tiered heterogeneity no matter what year it is. Our research addresses the problem of dealing with the lowest rung of the technological ladder, the current Baseline device, in order to address the technological issues that arise. Similar problems may be evident in the future for whatever PDA-like device may exist at that time. Research performed now helps us anticipate the long-term utilisation of "lowest-rung" devices on the Grid in the future.

Third, by their very nature, it may be doubtful that users will ever want to give up their power-limited Baseline devices for others' use. Slow improvements in battery technology only compound the problem. There are two ways we address this problem in this paper. A system architecture can be designed to hopefully assuage the problem of Baseline device "overusage" as perceived by the user. Our architecture hopes to accomplish that, as we shall explain in §4.1. Additionally, the Baseline owner must be given a persuasive incentive to contribute his device. The economic model we present in §4.2, almost a game theory approach if you will, provides a potentially compelling rationality for owners to grant use of their Baseline wares.

Fourth, users typically do not leave their Baseline devices on all the time and thus allow these machines to automatically shut off. This may substantially reduce the potential number of resource contributors. If users are motivated enough to want to contribute to the Grid in the first place (as we have suggested in the previous point), they will be able to allow such devices to be "always on," a trait confluent with upcoming "always on" 3G wireless technology. People who demand always-on, always-connected mobile devices can thus obtain savings by putting their machines in "semi-standby mode," where, for example, the CPU clockspeed can be reduced and the energy-draining LCDs can be turned off while the machine continues with computations. With these techniques, battery conservation can be increased along with the amount of work that can be done in the background. Two other points are noteworthy. In contemporary society, users at their desk, either at home or work, tend to leave their Baseline device plugged into a rechargeable cradle or into the wall socket anyways. Also, although many Baseline may be shut down, there will most likely always be active devices to be utilised due to the potentially large number of users involved,

Finally, there may not be a clear Grid application domain which can leverage the use of Baseline devices. Grid computing, in general, has already established the context for its own existence: resource sharing and distributed computation. Our research looks to preserve the Grid abstraction by simply contributing Baseline device resources for contemporary and future Grid applications. The most significant issue is that, as we have mentioned, Baseline devices are typically constrained in hardware, software, and network connectivity. Applications intended to leverage Baseline devices must be written (or be adapted retroactively) such that their problem space can be decomposed and distributed among Baseline devices accordingly to fit these limitations, as we shall show next.

4. SYSTEM ARCHITECTURE

In the previous section we proposed that utilising the enormous number of wirelessly-connected Baseline units is compelling enough to potentially merit their integration into the Grid. Conversely, we had also described how Baseline devices do not fit well into current grid computing. The Globus middleware standard for the Grid currently runs only on Unix and Windows systems, but the potential lies for it to be ported to PDAs and certainly to laptops. However, Baseline devices' inherent characteristics associated with mobility complicate their inclusion. In this section we suggest a possible system architecture solution for integrating mobile devices as well as an economic model to foment its growth.

4.1 A Proxy-Based Clustered Architecture

An architecture to support large numbers of mobile devices in a computational grid must address the issues presented in the previous section: device heterogeneity, low-bandwidth, high-latency connectivity; possibly extended periods of disconnectivity; device power consumption; and software interoperability. Additionally, a solution for these problems must provide service in a scalable manner.

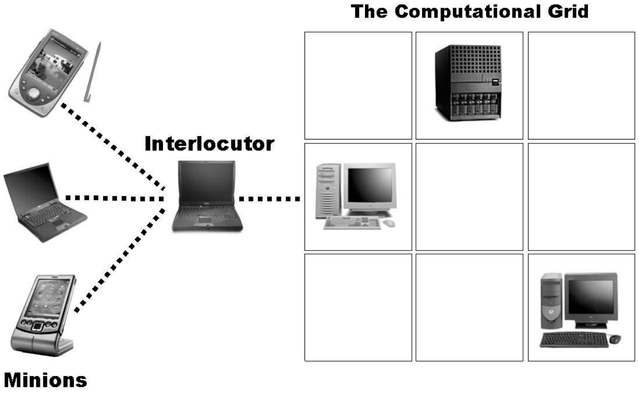

Figure 1: A broad view of the proxy-based clustered architecture.

In order to present these challenges in a tangible manner and to suggest appropriate research directions in our LEECH project, we offer an architecture we can utilise in our analysis. We present a broad overview of our proposed proxy-based clustered design in Figure 1 and then subsequently address its characteristics. We initially establish the scenario wherein there exist multiple Baseline units that are to be integrated into the computational grid. In our architecture, we create clusters of N Baseline units. Each cluster is centered around a proxy which can be either another Baseline device, a non-Baseline node within the Grid, or a dedicated middleware server. For wireless network configurations, the proxy would ideally be co-located with the wireless access point. The proxy, which we call an *interlocutor*, will be chosen such that it has adequate computing resources to handle its requisite responsibilities. The devices are said to be the *minions* of the interlocutor. The N Baseline devices are not visible to the rest of the computational grid. The interlocutor represents these devices to the Grid on their behalf, thereby freeing the minions from a number of responsibilities, which we shall discuss shortly.

The interlocutor can represent any number of devices (although in the future we will try to determine a scalable limit through empirical analysis). Within the Grid, the interlocutor runs appropriate Grid middleware, such as Globus, to publish itself as a node that can contribute a certain amount of computational, networking, and storage resources. This amount is the aggregate total of the resources of the interlocutor's minions. When a resource request arrives at the interlocutor from a resource consumer, the request is handled by the interlocutor. For simplicity, we proceed in this example with the assumption that the request is for CPU time to process incoming data. The interlocutor must decompose the request accordingly among its minions; this problem is accentuated due to the typical limited RAM capacity of handheld PDAs. Problem and data partitioning is known to be a difficult task within the parallel computing community [25], but we assume that a subsystem will provide the tools needed to successfully distribute the problem (e.g. a descriptive hint to distribute a 2-D array using block partitioning). After the problem has been distributed to the minions, the interlocutor waits for results and sends them back to the requester. The interlocutor has the option of aggregating the data before responding with the result in order to amortise the cost of per-message communication overhead. Requests to the interlocutor for storage or data distribution can be handled in a similar fashion.

We immediately note the ostensible similarity between our approach with that of other clustering techniques intended to rein in a loosely assembled group of devices. For example, many ad hoc routing schemes [4] utilise such clustering. Bluetooth-enabled devices assemble themselves into *piconets* of seven or less nodes centered around a master device [23] [27]. Landmark routing [28] suggests a similar approach. Mobile IP [29] facilitates the integration of mobile computers into the Internet by using *home agents* to act on behalf of mobile nodes. ALICE [22] provides a similar capability but at the application layer to support CORBA-enabled applications. However, these approaches are used only for routing data; the clusters are not leveraged to provide request partitioning.

Clustering is also used by the file-sharing peer-to-peer KaZaa program using the Fast Track infrastructure [33] to facilitate scalable searching. In this system, peer nodes are clustered around so-called *supernodes*. These supernodes serve as indexing repositories for search requests from peers, thereby negating the need for multicast searches in infrastructures such as Gnutella or centralised search indices as with Napster. However, these supernodes perform only indexing and do not involve themselves with resource partitioning (since the central objective of KaZaa is file-sharing).

The Control of Agent-Based Systems (CoABS) research effort [7] also uses a proxy architecture but in the context of software agents. Here, heterogeneous agents use a custom communication channel to communicate with a representative proxy, which in turn utilises a standardised communication API to talk with a larger agent grid. However, this architecture is more focused on interoperability via the hiding of agent heterogeneity and does not utilise the proxies for problem decomposition in the computational grid.

Our proxy-based clustered architecture potentially addresses a number of important research challenges.

- The Baseline nodes delegate much responsibility to their interlocutor, which acts as a representative agent for N nodes. The interlocutor publishes the availability of the nodes to the rest of the Grid and negotiates requests for service on behalf of its minions. If the cluster and the requester are physically distant from one another, this scheme reduces N long-haul request negotiations down to N short-haul advertisements (from the minions to the interlocutor) and 1 long-haul negotiation (between the interlocutor and the requester).

- The Baseline nodes can autonomously decide and publish their availability within the Grid through the interlocutor.

Consider the fact that most Baseline units are under different ownership. A unit's availability can be determined either through the owner's decision or through automatic evaluation of metrics (e.g. the device's CPU load or its bandwidth). Once the decision is autonomously made, the minion informs the interlocutor that it is no longer available to contribute, so the interlocutor allocates future requests accordingly among the remaining nodes. Likewise, the interlocutor can preemptively prepare for its minions' unreliability by publishing only a portion of the resource capacity. If the requested service demands more available resources than are available from the cluster, the interlocutor responds with an appropriate error message to the requester. Coordination among minions and the interlocutor to determine if a request can be met can be performed with a scheme such as a two-phase commit. Furthermore, this autonomous decision-making scales well by insuring that the interlocutor does not need to have global knowledge of its minions' resource availability.

- The importance of bandwidth availability and even of periods of disconnectivity is marginalised. By definition, grid applications are intended to be written such that they should not depend on low-latency communication (although efforts have been made to run communications libraries like MPI traditionally found on multiprocessors instead on a wide-area distributed system [14] [24]). Furthermore, the interlocutor, after receiving a resource request and partitioning it, can cache individual requests to particular minions, thereby partially hiding connectivity deficiencies. Similarly, results from the minions can be cached until the aggregate total is collected if need be.

- The interlocutor further shields the Grid computing requester from the heterogeneity of the cluster minions. The only information available to the requester is the aggregate resource total provided by the interlocutor. The underlying heterogeneity of the Baseline devices is not known. However, the price to be paid is that the interlocutor is responsible for appropriately partitioning the work among the minions. In the future we shall investigate the issue of partitioning, for instance, a number of needed FLOPs among a high-end laptop, a low-end laptop, and two PDAs.

- The always-persistent issue of power consumption is mitigated as well. As mentioned earlier, fewer long-haul communication is needed by the Baseline devices. Additionally, the device's power consumption metrics can be provided to the interlocutor to aid its partitioning decisions.

- Service discovery is simplified with the interlocutor. Cluster minions are only responsible for discovering the interlocutor, which is typically near. Many options for this discovery already exist, including DHCP, Jini, the Service Location Protocol, and expanding ring IP multicast. The interlocutor is then responsible for the interaction needed to discover and interoperate with other entities on the Grid on behalf of its minions.

Discussion: The architecture we presented was intended to shield the deficiencies of Baseline units in order to support their use within the Grid. We now analyse our proposal by raising a number of questions which we hope future researchers pursuing this topic will additionally try to address.

The core problem at hand is determining how Baseline units are manifest in the Grid. We have argued that since such devices are characterised by heterogeneous capabilities, unreliable availability, and power consumption sensitivity, these devices should be hidden behind a proxy; in our design, the interlocutor negotiates on behalf of other Baseline units. Is this a reasonable approach? Following our arguments, hiding Baseline units behind an opaque agent provides a number of advantages. Heterogeneity of the minions is hidden, and they are free to enter and leave sessions of Grid participation of their own volition. From the viewpoint of the Grid application programmer, this situation means that he can partition his problem according to the number of visible nodes in the Grid and leave further decomposition among an interlocutor's minions to the system. However, this obviously implies that the interlocutor must be responsible for automatic micro-granularity problem space partitioning, an inherently application-specific proposition. We suggest that perhaps the programmer can provide problem decomposition hints to the interlocutor or even utilise downloadable code responsible for problem space partitioning to specify how the interlocutor can perform this task. We will look to leverage previous research in the field of mobile code to facilitate this goal. Nonetheless, the problem remains difficult if one follows our argument that Baseline units are to be hidden within the Grid.

On the other hand, completely revealing the heterogeneity and unreliable availability of such units in the Grid would allow interlocutors to have far less responsibility but would place the burden of problem decomposition entirely on the programmer prior to compile-time. We note that this case is obviously nothing new; programmers of parallel/distributed applications must deal with problem decomposition in current systems. However, if we follow this argument, the issues of heterogeneity and availability again surface. Extending this approach even further by eliminating the interlocutor altogether, a third approach would be to have each unit represent itself on the Grid. We anticipate our own future work and other researchers to come up with new solutions and tradeoffs.

4.2 An Economic Model

Needless to say, economic plans are typically not a major concern of computer science research. However, because Baseline units are typically owned by consumers, as are PCs used in contemporary distributed applications like Seti@home, we must consider almost a game theory approach to promote the relationship between consumers and Grid service providers in order to see our model come to fruition.

Much work has already been done to describe market strategies for distributed systems (e.g. [36]) and economic models for grid computing (e.g. [6] [34]). It has been suggested that economic modelling is appropriate in order to: enforce general strategies, schedules, and procedures for resource management and allocation; provide incentives for consumers and producers to participate in the computational grid; and regulate supply and demand. Furthermore, since grid computing provides computing resources as a service in much the same way that electrical companies provide electricity, capitalistic policies should be enforced in the future to ensure long-term financial viability of commercial Grids. In pursuit of these goals, grid economic models have been devised following conventional market paradigms, such as models for a commodity market, auctioning, contract tendering, and bartering.

However, these models have been not taken into consideration the inherent characteristics of mobile computing devices, especially Baseline devices intended for the mainstream consumer market. As such, our proxy-based clustered architecture suggests a new model based on representative agency that provides faster return-on-investment for all parties. We offer the following viewpoints based on two characteristic divergences of Baseline devices away from contemporary PC- or server-based grid computing:

Baseline units are typically owned by one person. In contrast, contemporary clustered computing approaches are based on local-area networks of workstations found at universities or companies. In such environments, bandwidth is typically bought from a service provider as an aggregate total, so the cost is amortised across a number of computers. Owners of one (or a few) Baseline units do not have such a luxury, so steps must be taken to minimise network utilisation cost, especially during grid service negotiation. We suggest the following model. The interlocutor can negotiate with the Grid computing user to receive an appropriate lump amount of compensation in return for resources. The interlocutor proceeds to divide the resource request to its minions and allots them proportionally divided compensation in return. The interlocutor can be seen to have subcontracted work out to its minions.

Owners of Baseline units typically have a much more restricted operating budget than do operators of local area networks. This restricted budget conflicts with the need for the owner to invest sizable time and effort in setting up software to participate in the Grid. Thus, return-on-investment is a desirable trait for the Baseline unit owner (the resource producer), not to mention for the Grid computing users (the resource consumers). We leverage the fact that since Baseline unit owners are restricted financially, the cost of service from Internet service providers (ISPs) or application service providers (ASPs) is a much more important factor than for LAN operators in a business or university.

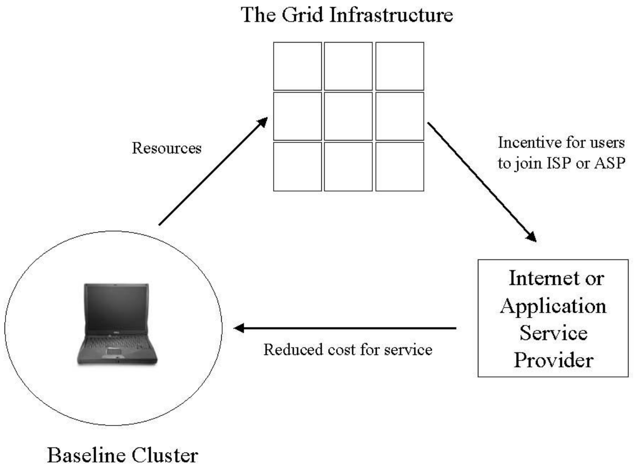
The Grid Infrastructure
Resources
Incentive for users to join ISP or ASP
Internet or Application Service Provider
Reduced cost for service
Baseline Cluster

Figure 2: A self-sufficient economic model for Baseline devices involved in grid computing.

We suggest a self-sufficient triangular fair trade policy that interconnects the entities or the resource provider, the resource consumer, and the ISPs. As shown in Figure 2, a Baseline cluster provides resources to the grid computing consumers. In turn, the grid applications and the Grid itself provide incentive for Baseline users to sign up for service from ISPs. Finally, ISPs grant Baseline cluster users with cost-reduced Internet service. The ISPs, after all, utilise the Internet service itself as a loss leader and make money from advertisements; this approach would provide them a new revenue stream.

Discussion: Is the preceding proposal compelling enough to motivate consumers to contribute their resource-limited, power-hungry Baseline units to the Grid? In one sense, this question has already been answered by the strong success of the Seti@home project: users are indeed willing to contribute their machines for “the greater good” of distributed computing. However, users will be less likely to allow their Baseline machines to be utilised than they would their resource-rich desktop PCs. As mentioned earlier, users need to be motivated on two fronts. First, a sufficiently persuasive support architecture of technical merit must be available that is proven to mitigate the issues inherent in Baseline devices. Such technical matters will provide reassurance to the users that their limited machines are being efficiently used. Second, users will need commercial and financial incentive to contribute what they may perceive to be their Baseline units’ limited resources. The economic model we presented here tried to address this issue. We assert that by its very nature, the problem of motivating user acceptance and participation will require future researchers in this topic to put considerable thought into such a non-traditional area of concern.

5. CONCLUSION AND RESEARCH PLANS

Compared to PCs and certainly to multiprocessor computers, mobile computing devices for the consumer marketplace are hampered by weaker hardware and high battery consumption. Mobility additionally brings the principal challenge of unreliable low-bandwidth connectivity. In this paper we assessed the difficulties involved in integrating mobile computers into the computational grid, presented possible solutions to work around these problems and to even leverage them as strengths, and examined the plausibility of success from technological and economic viewpoints. Our approach was based on the guiding tenet of leveraging the principle of economy of scale to utilise the enormous body of available mobile devices. We opined that research now in identifying the fundamental challenges and potential solutions for the current generation of hardware will serve the community well as increasingly faster and more powerful systems become available in the intermediate to far future. To paraphrase Machiavelli, in the beginning the problem with utilising Baseline devices is easy to solve but hard to diagnose; with the passage of time, having gone unrecognised and untreated, it becomes easy to diagnose but hard to solve.

During our discussion, several questions were left unanswered, providing a foundation for future work. It is still unclear whether participating Baseline units ought to be visible to the rest of the Grid. In this paper we argued that a proxy can represent a number of Baseline units on their behalf; although this shields the devices’ heterogeneity, it comes with the cost of having the proxy perform problem decomposition. Another issue is the motivation of user to contribute Baseline resources to the Grid. In this paper we suggested an economic model that could potentially encourage future growth. As our investigation in this area continues, we look forward to new research challenges yet to be uncovered.

Our research plans include the following. We will form testbeds of interlocutor and minion clusters using currently available mobile consumer devices. We shall port Globus middleware standards onto interlocutors to facilitate interoperability with network of workstation clusters. Grid applications will be surveyed and evaluated to determine which types of programs can fully utilise our proposed architecture. Scalability and throughput will be measured through implementation and simulation of our models. Finally, we will investigate the issue of network security in our context.

6. ACKNOWLEDGMENTS

The authors would like to thank Richard Meyer, Mark Yarvis, Jiejun Kong, and Professor Rajive Bagrodia for their valuable feedback during the writing process.

7. REFERENCES

[1] T. Anderson, D. Culler, D. Patterson, and the NOW Team. “A Case for NOW (Networks of Workstations),” *IEEE Micro*, Feb. 1995.

[2] M. Baker, R. Buyya, and D. Laforenza. “The Grid: International Efforts in Global Computing,” In *Proceedings of the International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet*, July 31-August 6, 2000.

[3] D. Becker, T. Sterling, D. Savarese, J. Dorband, U. Ranawak, and C. Packer. “Beowulf: A Parallel Workstation for Scientific Computation,” in *Proceedings of the 1995 International Conference on Parallel Processing*.

[4] J. Broch, D. Maltz, D. Johnson, Y.-C. Hu, and J. Jetcheva. “A Performance Comparison of Multi-Hop Wireless Ad-Hoc Network Routing Protocols,” *Mobile Computing and Networking*, pp. 85-97, 1998.

[5] R. Buyya, K. Branson, J. Giddy, and D. Abramson. “The Virtual Laboratory: Enabling On-Demand Drug Design with the World Wide Grid,” In *Proceedings of the IEEE International Symposium on Cluster Computing and the Grid*, May 21-24, 2002.

[6] R. Buyya, D. Abramson, J. Giddy, and H. Stockinger. “Economic Models for Resource Management and Scheduling in Grid Computing,” *Special Issue on Grid Computing Environments, The Journal of Concurrency and Computation: Practice and Experience (CCPE)*, Wiley Press, May 2002.

[7] *The Control of Agent-Based Systems (CoABS) homepage*. `coabs.globalinfotek.com`

[8] *The distributed.net homepage*. `www.distributed.net`

[9] *The Economist, Technology Quarterly*, “Hooked on Lithium,” June 22, 2002.

[10] *Folderol: bringing the Human Genome Project to your desktop*. `www.folderol.org`

[11] *The FightAIDSatHome homepage*. `www.fightaidsathome.org`

[12] G. Forman and J. Zahorjan. “The Challenges of Mobile Computing,” *IEEE Computer*, vol. 27, no. 4, April 1994.

[13] I. Foster and C. Kesselman. “Globus: A Metacomputing Infrastructure Toolkit,” *International Journal of Supercomputer Applications*, vol. 11, no. 2, 1997.

[14] I. Foster and N. Karonis. “A Grid-Enabled MPI: Message-Passing in Heterogeneous Distributed Computing Systems,” in *Proceedings of the 1998 Supercomputing Conference*, November 7-13 1998.

[15] I. Foster, C. Kesselman, and S. Tueke. “The Anatomy of the Grid: Enabling Scalable Virtual Organizations,” *International Journal of Supercomputing Applications*, 2001.

[16] “Mobile PC Sales Reach Historic Highs,” *Gartner FirstTake report FT-15-4807*. `www3.gartner.com/resources/104300/104392/104392.pdf`

[17] “Gartner Dataquest Says Worldwide PDA Shipments Will Increase 18 Percent in 2002,” *Gartner Press Release*, April 4, 2002. `www4.gartner.com/5_about/press_releases/2002_04/pr20020403a.jsp`

[18] *The Genome@home homepage*. `genomeathome.stanford.edu`

[19] *The Globus homepage*. `www.globus.org`

[20] A. Grimshaw, W. Wulf, J. French, A. Weaver, P. Reynolds, Jr. “Legion: The Next Logical Step Toward a Nationwide Virtual Computer,” *University of Virginia Technical Report No. CS-94-21*, 1994.

[21] A. Grimshaw, A. Ferrari, G. Lindahl, and K. Holcomb. “Metasystems,” *Communications of the ACM*, vol. 41, no. 11, November 1998.

[22] M. Haahr, R. Cunningham, and V. Cahill. “Supporting CORBA Applications in a Mobile Environment,” In *Proceedings of the 5th International Conference on Mobile Computing and Networking*, August 1999.

[23] J. Haartsen. “BLUETOOTH - the Universal Radio Interface for Ad-Hoc Wireless Connectivity,” *Ericsson Review*, no. 3, 1998.

[24] T. Kielmann, R. Hofman, H. Bal, A. Plant, R. Bhoedjang. “MagPIe: MPI’s Collective Communication Operations for Clustered Wide Area Systems,” in *Proceedings of the Seventh ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, May 4-6 1999, in Atlanta Georgia.

[25] V. Kumar, A. Grama, A. Gupta, and G. Karypis. Introduction to Parallel Computing, The Benjamin Cummings Publishing Company, 1994.

[26] M. Litzkow, M. Livny, and M. W. Mutka. “Condor - A Hunter of Idle Workstations,” in *Proceedings of the 8th International Conference of Distributed Computing Systems*, June 1988.

[27] G. Miklos, A. Racz, Z. Turanyi, A. Valko, and P. Johansson. “Performance Aspects of Bluetooth Scatternet Formation,” In *Proceedings of the 1st Annual Workshop on Mobile Ad Hoc Networking and Computing*, 2000.

[28] G. Pei, M. Gerla, and X. Hong. “LANMAR: Landmark Routing for Large Scale Wireless Ad Hoc Networks with Group Mobility,” In *Proceedings of IEEE/ACM MobiHOC*, August 2000.

[29] C. Perkins and D. Johnson. “Mobility Support in IPv6,” In *Proceedings of the 2nd Annual International Conference on Mobile Computing and Networking*, November 1996.

[30] *The SaferMarkets homepage*. `www.safermarkets.org`

[31] *SETI@home homepage*. `setiathome.ssl.berkeley.edu/`

[32] L. Smarr and C. Catlett. “Metacomputing,” *Communications of the ACM*, June 1992.

[33] K. Truelove and A. Chasin. “Morpheus Out of the Underworld,” `www.openp2p.com/pub/a/p2p/2001/07/02/morpheus.html`

[34] S. Vazhkudai and G. Laszewski. “A Greedy Grid - the Grid Economic Engine Directive,” In *Proceedings of the International Parallel and Distributed Processing Symposium*, April 2001.

[35] M. Weiser. “The Computer for the Twenty-First Century,” *Scientific American*, September 1991.

[36] M. Wellman and P. Wurman. “Market-Aware Agents for a Multiagent World,” *Robotics and Autonomous Systems*, volume 24, 1998.

[37] “Wireless Transport Layer Security specification,” `www1.wapforum.org/tech/documents/WAP-261-WTLS-20010406-a.pdf`