Grid Enabled Data Analysis on Handheld Devices

Ahsan Ikram[1], Arshad Ali[1], Ashiq Anjum[1], Conrad Steenberg[2], Harvey B. Newman[2], Julian J. Bunn[2],
Michael Thomas[2], Tahir Azim[1]

[1]National University of Sciences and Technology, Rawalpindi, Pakistan.
[2]California Institute of Technology, 1200 East California Boulevard, Pasadena, CA 91125, U.S.A.
E-mail: {ahsan,arshad.ali,ashiq.anjum}@niit.edu.pk, {conrad,newman}@hep.caltech.edu, Julian.Bunn@caltech.edu,
thomas@hep.caltech.edu, tahir@niit.edu.pk

Abstract – In the present era, the requirement for information on portable, handheld devices is demanding the realization of increasingly complex applications for increasingly small and ubiquitous devices. This trend promotes the migration of technologies that were originally developed for desktop computers to handheld devices. With the onset of Grid computing, users of handheld devices should be able to accomplish much more complex tasks, by accessing the processing and storage resources of the Grid. This paper describes the development, features, and performance aspects of a Grid-enabled analysis environment designed for handheld devices. We also describe some differences in technologies required to run these applications on desktop machines and hand held devices. In addition, we propose a prototype agent-based distributed architecture for carrying out high-speed analysis of physics data on handheld devices.

Index Terms – Grid computing, Handheld Computing, Performance Enhancement, Physics Analysis.

I. INTRODUCTION

In this paper, we present a brief description of a Grid enabled analysis environment for handheld devices that we have developed specially for the high-energy physics (HEP) community, and then describe a distributed analysis architecture for carrying out high-speed analysis of data over a Grid, in order to overcome the limited processing power of handheld devices.

Handheld devices, considering their present rate of growth, are expected to replace desktop machines in the near future. This research focuses on enabling the technology of mobile, handheld devices, having the typically inherent limitations of reduced CPU performance, small secondary storage, heightened battery consumption sensitivity, and unreliable low-bandwidth communication, to analyze the data in enormous databases, efficient access to which is made possible by the Grid.

The CMS (Compact Muon Solenoid) [1] at the European Center for Nuclear Research (CERN), going online in 2006, will use such Grid-based databases to store the gigabytes of data it will generate each minute. In its raw form, data cannot be used to generate any significant results, because of its sheer quantity and complexity. The only way of analyzing this data is by using analysis applications to render it in the form of 2D and 3D diagrams, which scientists can use much more effectively in deriving conclusions about events taking place in the CMS. In this paper, we present a physics analysis environment for handhelds that is optimized for maximum performance on handheld devices. To enable these applications to access a computational or data Grid, we have developed a Grid-enabled portal named JClarens, which provides simplified programmatic interfaces to access various services offered by the Grid. This allows even the resource-constrained devices running the analysis applications to harness the power of Grid computing.

II. RELATED WORK

Mobile computing and grid computing, if merged, have a great deal of technological potential. This would allow users of mobile devices and Personal Digital Assistants (PDAs) to submit jobs on the Grid, and get access to its tremendous processing power at their fingertips. It would also allow them to access data being generated by the Grid, and analyze it on their handheld devices. This research mainly targets PDAs such as the Pocket PC and Palm. Due to the processing power and resources generally required by analysis applications, none of them has yet been ported to low resource and slow processing (typically 200MHz to 400MHz) handheld devices. Moreover, slow, unreliable, and intermittent nature of wireless connections has always been a concern.

For desktop machines, on the other hand, a variety of Grid enabled physics analysis applications is available. Some of these include JAS [2], WIRED [3] and ROOT [4]. JAS (Java Analysis Studio) was developed at the Stanford Linear Accelerator Center (SLAC), and is used for 1D and 2D graphical display of histogram data obtained from particle accelerators. Along with graphical display, it also offers various mathematical functions to fit along the displayed data. The user can also view various statistics related to the analyzed data set. WIRED (WWW Interactive Event Display) was developed at CERN in collaboration with SLAC. WIRED uses files based on the Extensible Markup Language (XML) for graphical rendering of events and sub component geometry information from various particle experiments. Also developed at CERN, ROOT analyzes special format ROOT files in which data is arranged in a highly efficient, hierarchical structure.

Although there are no currently existing physics analysis applications for handhelds, research is underway to integrate the two groundbreaking fields of Grid computing and

mobile computing [5]. Even Java-enabled mobile phones have been targeted for possible integration with the Grid environment, in order to provide more computationally intensive features and capabilities on mobile phones [6].

Our Pocket PC based analysis applications are basically built around the Java Analysis Studio (JAS) and WWW Interactive Remote Event Display (WIRED) software, and have been optimized for the Personal Java [7] environment on the Pocket PC 2002. We will use these two applications for specific performance analysis.

III. ANALYSIS ENVIRONMENT ARCHITECTURE

In a mobile computing environment, the user does not carry all of the data that he needs; instead data generally has to be fetched from some data servers that, in our case, will be connected to the Grid. To enable data from the Grid to be provided to analysis applications, an interface was required.

A Grid enabled portal named JClarens has been developed to serve the purpose. JClarens is basically an enhanced Java-based version of Clarens [8], a web services framework originally developed in Python at the California Institute of Technology. JClarens provides us our required features of remote data access, job submission, and GSI [9] based security authentication. JClarens authenticates users by certificate exchange, whenever the user logs on for the first time. After successfully logging in, the user can access the wide range of services hosted by JClarens. An architectural overview of the analysis environment is shown in Figure 1.

Clients comprise of handheld and desktop devices, which communicate with JClarens, using either XML-RPC (XML Remote Procedure Call) or SOAP (Simple Object Access Protocol). Once logged on to JClarens, the clients can access the services offered by it. A detailed overview of the services offered by Clarens/ JClarens, and the way its clients communicate with it is described in [10].
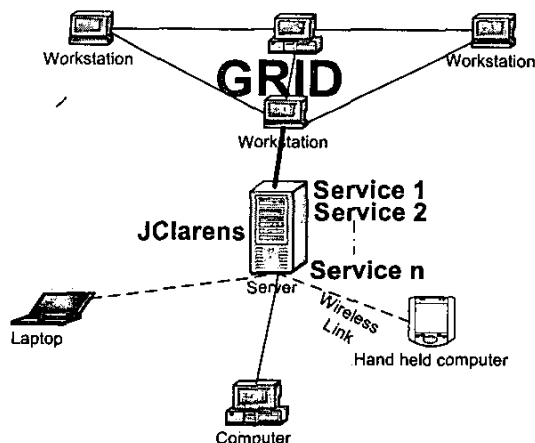

Figure 1. A general architectural overview of the analysis environment.

IV. TECHNOLOGIES AVAILABLE FOR HANDHELD DEVICES

Java was chosen at the beginning as the development platform due to its portability and short development time. However, the Pocket PC was our actual deployment platform, and the selection of a suitable and compatible JVM for it proved to be a major obstacle. We did not have many options in that regard and most of the JVMs we tested did not suit our requirements in one way or the other. We started with IBM Device Developer [11], which appeared to be a useful tool for PocketPC based application development, but it was not long before we had to reject it, because it only supported Personal Profile, and lacked certain crucial components we required. We next opted for Super-Waba [12] as it offered some interesting features, such as Java based syntax and open source availability, but soon we had to drop SuperWaba as an option due to some bugs detected during evaluation and complete lack of support for Java Swing[1]. Continuing with our search for a better JVM, we found Savaje [13]. However, a prerequisite for Savaje was the installation of SavajeOS in place of WinCE on the Pocket PC. Since portability between different operating systems was one of our major software requirements, we had to abandon Savaje as well. MIDP [14] was considered an alternative for some time, but it was mainly designed for mobile phone devices and not PocketPCs.

In the end we decided to go for Personal Java, as after initial evaluation and comparison with previously tested JVMs, we found it suited our needs more than any other JVM. There were some minor hurdles that came along with PersonalJava, such as the setting of the CLASSPATH environment variable on the PocketPC, but that was solved by using a special registry editor tool named PHM Registry Editor [15]. To further suit our needs we found Insignia Jeode [16], a Personal Java compliant Runtime environment. At the time of writing this document, Insignia's Jeode Personal Java Runtime Environment is still being used as our development platform. Recently, NSICom has launched a new Personal Java compatible platform named CrEme, whose performance is optimized for Pocket PC's, and is comparable to that of J2SE 1.4. However, we have not completely abandoned Jeode in favor of CrEme so far, due to some licensing issues associated with CrEme.

V. JAVA ANALYSIS STUDIO (JAS) AND JASONPDA

Java Analysis Studio (JAS), as stated before, is developed at Stanford Linear Accelerator (SLAC). JAS is a physics analysis tool used for analyzing data obtained from linear accelerators in the form of 1D-2D histograms. Apart from 1D-2D analysis, JAS offers numerous other facilities, which include comparison of displayed histograms with predefined mathematical functions (Quadratic, Cubic,

---

[1] Other platforms also do not support Swing directly, but still allow us to use it with some modifications in Swing's source code.

Gaussian, polynomial, Lorentzian etc), the fitting of these functions over the displayed histograms for statistical analysis (peak value, average value etc), and executing your own analysis code on selected datasets.

JASOnPDA was our first application for the PocketPC. JASOnPDA is the scaled down version of Java Analysis Studio and is especially designed for constrained handheld devices. JASOnPDA provides essential analysis utilities of Java Analysis Studio on PocketPC devices, and was developed using J2SE 1.1. As it has been built for mobile users, JASOnPDA has the facility to sign on to JClarens server using a certificate-based authentication procedure. Once successfully authenticated, the user is allowed to access files stored at the server. Remote browsing facility allows users to browse the directories served by JClarens and look for desired ROOT files. The selected ROOT file is analyzed and a tree structure displays the hierarchy of objects in the ROOT file. The user can move along the tree, selecting any object from the tree structure, and the selected object will be displayed in the form of 1D-2D Histograms in the display panel. Statistical fitting features from JAS are also available. The user can also view statistics information related to the histogram displayed on the screen. Keeping in view the small screen size of the PocketPC, different viewing options are also provided. A screen shot of the software running on the PDA is shown in Figure 2.

To overcome the issue of intermittent, unreliable connections during the transfer of large datasets and files, the downloading process for large files is carried out by dividing the size of the file into smaller chunks, and downloading those smaller file chunks in steps, rather than using a single connection to transfer the entire file. This allows us to check point the file transfer process and ensure that if there is a disconnection at any stage, the entire dataset is not lost, and downloading can be resumed from the latest check point.

The software has been designed to ensure that it is as extensible as possible. Interfaces have been exposed that allow users to write their own analysis classes for different file formats. This allows users the flexibility to select the procedure by which their file will be analyzed and its contents displayed. To specify the class to be used for analysis, the user only has to give the name of the class in a simple properties file. In this way, the user can easily plug in the classes that he wants to use for handling new file formats and user-specific custom file formats.
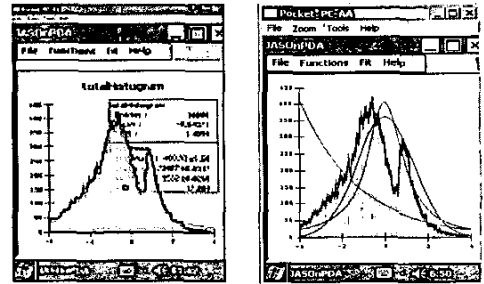

Figure 2. JASOnPDA running on a PDA showing its features of histogram plotting, function fitting, and statistics calculation.

VI. WWW INTERACTIVE REMOTE EVENT DISPLAY (WIRED) AND WIREDONPDA

WWW Interactive Remote Event Display (WIRED) was a joint venture of Stanford Linear Accelerator (SLAC) and European Organization for Nuclear Research (CERN). WIRED is one of the first Event Displays written in Java for use on the World Wide Web. It provides a framework for writing event displays and visualizations of detector geometry information. WIRED data is stored in special format files known as HepRep (HEP representable) and HepRep2.

WiredOnPDA is another of our analysis applications developed for PocketPC devices. As the name suggests, WiredOnPDA provides analysis features of WWW Interactive Remote Event Display (WIRED) on PocketPC. WiredOnPDA accesses data using JClarens using the same procedure as JASOnPDA.

Once the user selects a file from the JClarens server, the file is downloaded into the RAM and a SAX XML parser parses the information stored in the file. Again for large files, a check-pointed data transfer process similar to JASOnPDA is employed. "Drawables" are then extracted from the parsed data and are displayed in a hierarchical tree structure in WiredOnPDA's tree panel. The user can then select any "drawable" from the tree and it will be displayed in the display console. Screen shots are shown in Figure 3 and Figure 4.
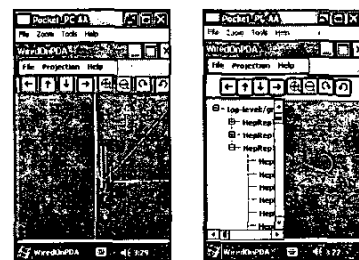

Figure 3. Two views of WiredOnPDA displaying event data and the structure of a HepRep2 file in separate panes.
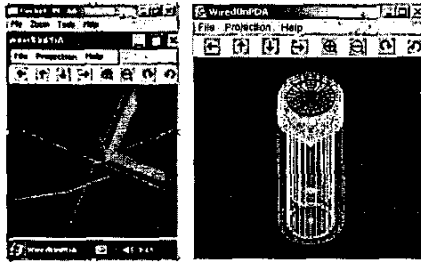
Figure 4. Two views of WiredOnPDA displaying events from a HepRep2
file (left) and detector geometry (right).

Due to the small screen size of the PocketPC, various display options are provided in order to utilize maximum screen space for event display. As shown in the figures, application is provided with a tool bar, which allows the user to scale, rotate or zoom the displayed event for better analysis.

WiredOnPDA has had some issues of concern especially regarding performance. The main reason for inefficient performance was the poor parsing speed of SAX parsers on PersonalJava. The main reason for this slow parsing is because of the differences in implementation of PersonalJava virtual machine compared to J2SE virtual machines. To address this issue various parsers were tested which included Xerces, Crimson, KXML and Piccolo. Piccolo has so far proved to be the fastest performer, with best possible results out of all the other parsers.

VII. PERFORMANCE ANALYSIS

It has been observed that the relatively fast clock speed (200 MHz) of the PocketPC or its large RAM (64 MB) does not actually give an accurate description of its performance. In reality, the PocketPC is a much slower device than desktop PCs with the same clock speeds and RAM, because of the difference in processor architecture and technologies used in manufacturing the RAM and other integrated circuits. As a result, we have tried to ensure that the algorithms used in the Pocket PC software are optimized as much as possible, and this has resulted in significant improvements in performance in successive iterations.

The PocketPC used in the performance comparisons was an iPAQ h3955, equipped with a 200 MHz Intel processor and 64 MB of RAM, running Windows CE 3.01, and using Insignia Jeode as the JVM for PersonalJava. The desktop machines (PCs) included a 400 MHz Intel Pentium-II with 256 MB of RAM, and a 2.4 GHz Intel Pentium-IV with 512 MB of RAM (DDR). Both the PCs were equipped with Sun's PersonalJava Emulation Environment 3.01, and J2SE 1.4, over Windows XP Professional. All the statistics related to the number of threads and memory usages were collected using the task manager utility available in Windows XP.
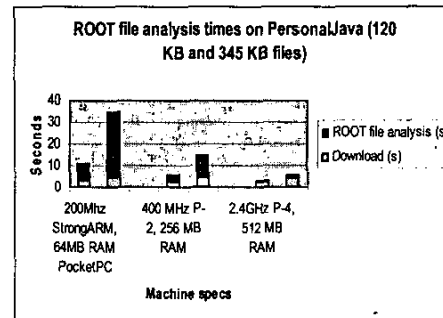

Figure 5. The graph shows the large difference in analysis times for two
ROOT files on machines with varying processing resources.

The graph in Figure 5 (above) demonstrates the vast difference in performance between the PocketPC and the two PCs in analyzing two ROOT files downloaded from the JClarens server. The smaller file contains only four histograms, and so is analyzed much more quickly than the second file, which contains almost 300 small histograms.

The difference is even more marked in case of HepRep2 file analysis, as can be seen from Figure 6. In this case, we have observed that a Pentium-IV can parse such a file in less than eight seconds, whereas the PocketPC can go up to 125 s for the same process. The difference in analysis times for the two files is because the larger file has much larger number of drawing primitives than the smaller one. The marked difference in analysis times clearly illustrates the slow performance of the Pocket PC as compared to desktop PCs.

Another issue that has been observed is the poor performance of PersonalJava as compared to J2SE. This difference is especially marked in the case of HepRep2 analysis, as XML parsing of HepRep2 is many times slower in PersonalJava than J2SE. Parsing times measured for the two platforms are given in Figure 7.
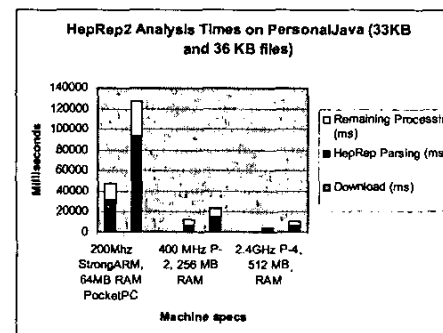

Figure 6. The graph shows the time analysis for HepRep2 files on different
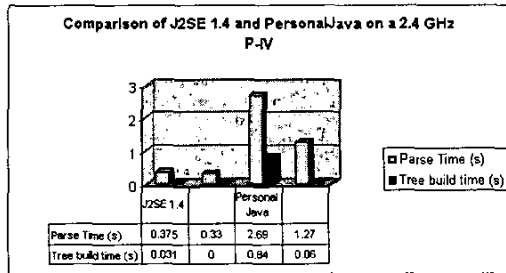machines and architectures.

161

Figure 7. This chart shows that the performance disparity between Personal Java and J2SE 1.4 persists even on faster PCs (a P-IV in this case).

Another issue that has to be considered is the fact that the memory footprint of PersonalJava has been kept purposely small as compared to J2SE. This is to allow Java programs to run on the relatively limited RAM available in PDAs. The small footprint is also a reason behind the slower performance as it causes less heap and stack space to be allocated, and forces the process to run in a smaller address space. The differences in memory usages by the two runtime environments are highlighted by Figure 8.

Also, there were several cases where functionality in J2SE was embedded in native code, whereas the same task was carried out using pure Java in PersonalJava. This was especially evident in the java.io classes. These performance issues are another reason why optimization has to be accorded extra importance while coding in PersonalJava, as compared to J2SE.

VIII. DISTRIBUTED ANALYSIS

A. Distributed Analysis Using Agents

Until now, we have been able to develop a few analysis applications for PocketPC devices, but during the development period we came to the conclusion that performance issues are a constant issue for PocketPC based applications. In order to solve these issues for our already built applications and for future needs, we have created a distributed analysis architecture based on mobile agents. We are now in the process of implementing this architecture that should suit the requirements of almost all kinds of portable devices. Figure 9 gives a general overview of this agent-based distributed analysis environment.
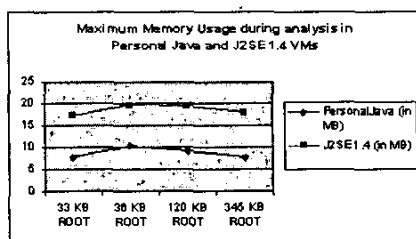

Figure 8. The heap space used by PersonalJava is almost half that of the J2SE 1.4 VM.
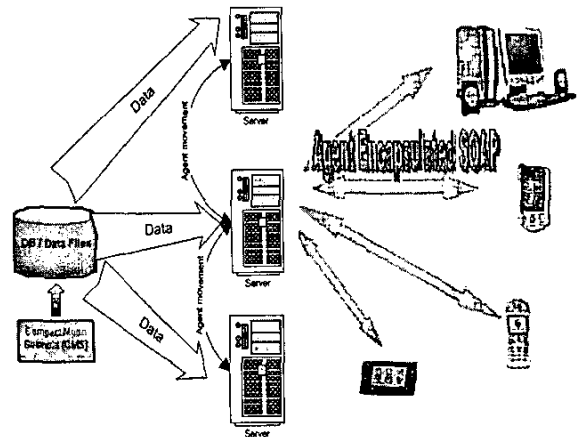

Figure 9. The architecture proposes agents that will communicate with PDAs and other devices using SOAP. This interaction will be mostly limited to data and request transfer. The importance of the agents' mobility will lie mostly on the server side, where they will handle load balancing and session replication issues.

A detailed performance analysis of mobile agents on wireless networks has been carried out by Kotz et al. [17], which proves that using mobile agents can result in significant performance benefits, provided that the number of mobile agents using the available bandwidth is kept within certain limits, and the overheads of mobile-agent communication (agent movement) and computation (overhead of the mobile agent platform) is kept small.

In our architecture, the agents reduce the load on PocketPC by carrying the analysis requests to analysis servers such as JClarens, which will reside on high resource devices connected to wired networks. The idea is that analysis on high resource devices will be more efficient in terms of time consumption. After the analysis is done, the agent sends the results to its host device and waits for any further requests. Since analysis on the servers is faster and results are smaller as compared to the actual file size, therefore considerable improvement is achieved in the performance of the PocketPC applications. The use of agents is meant to create a high performance analysis environment where response times are as small as possible and users are able to access and analyze data as soon as possible resulting in a richer, more interactive experience.

However, since agents are extremely resource constrained on the client end, we have kept the agents as light as possible in terms of storage and processing. On the client side, they are responsible only for handling the request and transferring results of analysis requests between client and server. This limitation on agents helps in keeping the memory and processing load as small as possible on the Pocket PC.

B. Distributed Analysis Using Condor

Condor is a job scheduling and job execution system developed at the University of Wisconsin, Madison. The

modular and service-oriented architecture of JClarens has helped us to provide an interface to PDAs to submit jobs to Condor. This enables the resource-constrained handhelds to launch lengthy and time-consuming jobs on a large number of files on a Condor pool, get periodic information on job status, and retrieve outputs.

The architecture of the Condor-based analysis environment centers on a server machine running JClarens. The machine is also configured to act as a job submitter on a Condor pool. Handheld clients select data files on JClarens on which they wish to launch their jobs. They, then, send job submission requests to JClarens, which in turn submits these jobs into Condor's queue. Each job submission request to Condor (through JClarens) consists of the binaries (or byte-code) of the executable to run, the name of the executable, the names of the input files and the submit file describing the characteristics of the job.

On receiving the job submission request, JClarens creates a set of separate directories as a staging area for Condor. It then copies the executable, the submit file and the input files to each of the directories (each input file in a separate directory). Once the staging directories have been prepared, JClarens calls "condor_submit" on each of the submit files.

Condor then takes over the allocation of the individual jobs to different machines in the Condor pool, and ensures that the jobs are completed and the results returned to the respective directories. While Condor executes the job on the pool, the client can repeatedly poll JClarens to check the status of his jobs. Once all the jobs are complete, the client can retrieve the outputs of all his jobs through JClarens. In this way, a complex and time-consuming set of jobs can be rapidly executed on the Condor pool, and its results displayed and visualized on the PDA. In this way, even users on slow-processing PDAs connected to wireless networks can carry out remote processing of lengthy operations on powerful machines in the Condor pool and thus, exploit the power of Grid computing.
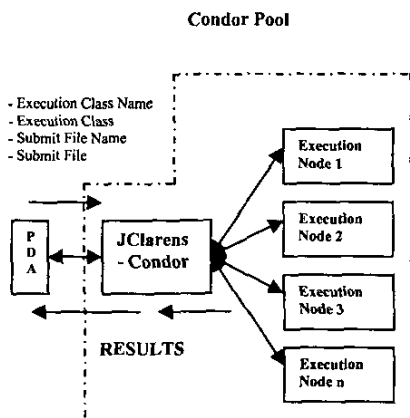

Figure 10. An architectural overview for Grid computing using Condor from handheld devices.

IX. CONCLUSION

The analysis environment has been developed to contribute to the scientist and physicist community all around the world, to help them in their quest for ubiquitous access to data. But it is yet to reach the high performance standards that are so easily achieved in case of desktop PC-based applications. The slow performance of the handheld devices is the major barrier that has to be overcome to achieve true interactivity. However, as described earlier, a combination of efficient and novel programming techniques has already enabled us to speed up our analysis applications to a great extent.

Our PocketPC based applications were demonstrated in ITU Telecom World 2003 as a part of the "GRID-Enabled Physics Analysis" demonstration. Also JASOnPDA was presented at the first Grid Analysis Environment (GAE) workshop at Caltech in June, and attracted a great deal of attention, due to the fact that it was the first physics analysis application ported to the PDA.

Our work on this analysis environment, however, is far from over. Already our current work proves that resource-constrained devices such as the PocketPC can be interfaced with the Grid, and can play a vital role in the realization of the idea of a Grid-Enabled Analysis Environment (GAE). Mobile and ubiquitous computing has yet to reach a mature level, where it can compete with desktop PCs in the kind of applications it can offer. The achievement of the above mentioned goals would prove to be a giant leap towards the attainment of this level of maturity.

REFERENCES

[1] CERN, 1 September 2003 [Online], "Welcome to the Compact Muon Solenoid Outreach Activities", Available: http://cmsinfo.cern.ch, 29 April 2004.

[2] Tony Johnson, 30 April 2003 [Online], "Java Analysis Studio", Available: http://jas.freehep.org, 29 April, 2004.

[3] FreeHEP, 13 July 2003 [Online], "WIRED - WWW Interactive Remote Event Display", Available: http://wired.freehep.org, 28 April 2004.

[4] Rene Brun & Fons Rademakers, 11 February 2003 [Online], "The ROOT System", Available: http://root.cern.ch, 30 April 2004.

[5] Thomas Phan, Lloyd Huang and Chris Dulan, "Challenge: Integrating Mobile Wireless Devices Into the Computational Grid", *Proceedings of the 8th annual international conference on Mobile computing and networking*, Atlanta, Georgia, USA, pp. 271-278, September 2002.

[6] Miika Tuisku, "Wireless Java-enabled MIDP devices as peers in Grid infrastructure", unpublished, November 2003.

[7] Sun Microsystems Inc., 10 September 2003 [Online], "J2ME Personal Profile", Available: http://wireless.java.sun.com/personal, 28 April 2004.

[8] Conrad D. Steenberg, Eric Aslakson, Julian J. Bunn, Harvey B. Newman, Michael Thomas, and Frank van Lingen, "The Clarens Web Services Architecture", *Proceedings of Conference on High Energy Physics (CHEP) 2003*, La Jolla, California, paper MONT008, March 2003.

[9] V. Welch, F.Siebenlist, I. Foster, J. Bresnahan, K.Czajkowski, J. Gawor, C. Kesselman, S. Meder, L. Pearlman, and S. Tuecke, "Security for Grid Services", *Twelfth InternationalSymposium on High Performance Distributed Computing (HPDC-12)*, IEEE Press, pp. 48-57, June 2003.

[10] Conrad D. Steenberg, Eric Aslakson, Julian J. Bunn, Harvey B. Newman, Michael Thomas, Frank van Lingen,"Clarens Client and Server Applications", *Proceedings of Conference on High Energy Physics (CHEP) 2003*, La Jolla, California, paper TUCT005, March 2003.

[11] IBM, 4 February 2003 [Online], "IBM WebSphere - e-business on demand, middleware, application server, portal, business integration, infrastructure software", Available: http://www.ibm.com/websphere, 30 April 2004.

[12] SuperWaba, 19 September 2003 [Online], "Super-Waba: The Real Power of Mobile Computing", Available: http://www.superwaba.com.br, 30 April 2004.

[13] MochaWorks, 17 September 2003 [Online], "SavaJe XE", Available: http://www.mochaworks.com, 29 April 2004.

[14] Sun Microsystems Inc., 29 August 2004 [Online], "Mobile Information Device Profile (MIDP)", Available: http://java.sun.com/products/midp, 29 April 2004.

[15] Philippe Majerus, 16 May 2004 [Online], "PHM Registry Editor", Available: http://www.phm.lu/Products/PocketPC/RegEdit, 30 April 2004.

[16] Insignia Solutions Inc., 10 May 2004 [Online], "Insignia Jeode Virtual Machine for Handhelds", Avaialble: http://www.insignia.com, 3 Februray 2004.

[17] David Kotz, George Cybenko, Robert S. Gray, Guofei Jiang, and Ronald A. Peterson, "Performance Analysis of Mobile Agents for Filtering Data Streams on Wireless Networks", *International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems (Proceedings of the 3rd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems)*, Boston, Massachusetts, USA, pp. 85-94, 2000.