

# Chapter XXXIX

## Parallel Computing on a Mobile Device

**Daniel C. Doolan**

*University College Cork, Ireland*

**Sabin Tabirca**

*University College Cork, Ireland*

**Laurence T. Yang**

*St. Francis Xavier University, Canada*

### ABSTRACT

*The Message Passing Interface (MPI) was published as a standard in 1992. Since then, many implementations have been developed. The MPICH library is one of the most well-known and freely available implementations. These libraries allow for the simplification of parallel computing on clusters and parallel machines. The system provides the developer with an easy-to-use set of functions for point-to-point and global communications. The details of how the actual communication takes place are hidden from the programmers, allowing them to focus on the domain-specific problem at hand. Communication between nodes on such systems is carried out via high-speed cabled interconnects (Gigabit Ethernet and upwards). The world of mobile computing, especially mobile phones, is now a ubiquitous technology. Mobile devices do not have any facility to allow for connections using traditional high-speed cabling; therefore, it is necessary to make use of wireless communication mechanisms to achieve interdevice communication. The majority of medium- to high-end phones are Bluetooth-enabled as standard, allowing for wireless communication to take place. The Mobile Message Passing Interface (MMPI) provides the developer with an intuitive set of functions to allow for communications between nodes (mobile phones) across a Bluetooth network. This chapter looks at the MMPI library and how it may be used for parallel computing on mobile phones (Smartphones).*

## INTRODUCTION

Bluetooth technology (IEEE 802.15.1) is all around us in the modern age, and it is expected that it will remain so for the foreseeable future. Its use has become so widespread that it would now be difficult to take a walk in any city or town without seeing somebody having a conversation via a hands-free Bluetooth-enabled kit. This gives credence to the estimate that there are well over two billion Bluetooth-enabled devices in the world today. This technology can be seen in everything from hi-fi speaker systems and cars to wireless keyboards and mice. Such applications are usually limited to the simplest type of Bluetooth network that can be formed: a point-to-point interconnect. Bluetooth itself allows far more complex network types to be formed, such as the Piconet (a star network topology) and the Scatternet (a collection of two or more Piconets interconnected by a common node [bridge]).

Why are the more complex network forms far less commonplace? Application development is one key factor affecting the use of these more complex forms. Most of today's mobile devices feature a Java Virtual Machine (JVM) as standard. This ubiquitous use of JVMs and the ease of application deployment makes the development of Java-based MIDlets one of the most attractive platforms to work on. One major factor that has made Java Bluetooth-based applications less appealing to software development companies is economics. When creating a MIDlet, one of the primary aims is that the application/game is capable of running on as many devices as possible, thereby allowing the product to be exposed to a sufficiently large market segment. Many such applications are designed and built so they can execute in MIDP 1.0 environments, ensuring that if the mobile device is Java-enabled, it should be capable of running the application. Even though one may think that if an application is developed as a Java MIDlet it should be capable of running on all devices with a compatible JVM, this could not be any farther

from the truth. At present, there is a myriad of different mobile devices on the market, many having different screen resolutions, bit depth, keypad layouts, reliances on vendor-specific APIs, and so forth. This has given rise to a huge area of porting applications to run on the maximum number of devices possible.

In the case of Macrospace's Dragon Island (Nokia, 2004), 69 versions of the game had to be developed in five languages, totaling 345 different builds of the game. This gives rise to substantial costs and complexity of maintenance. Another reason Bluetooth applications are not readily developed is due to the amount of Bluetooth-specific code that is required to be written for each individual application.

Some Bluetooth-enabled Java applications exist but are generally of a limited number. The MMPI library is one solution to the removal of the need to write reams and reams of Bluetooth-specific code, as well as simplifying the program development process. This chapter discusses in depth the details behind the MMPI library, showing how a fully interconnected mesh network may be formed and how a developer may use the library to achieve parallel computation across a network of mobile devices such as phones.

## BACKGROUND

The Bluetooth wireless communications system was named after Harald Blatand (Bluetooth) who was king of Denmark from about 958 A.D. to 968 A.D. He was renowned for getting people to talk, and thus his name was given to the technology. Sevn Mattisson and Jaap Haartsen working at Ericsson Mobile Platforms, Sweden, in 1994, developed the original specification. Formalized by the Bluetooth Special Interest Group (SIG) in 1998, it is now a standard form of wireless communications the world over. It is divided into three classifications pertaining to the maximum permitted power and effective range. Class two-

16 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

[www.igi-global.com/chapter/parallel-computing-mobile-device/21029](http://www.igi-global.com/chapter/parallel-computing-mobile-device/21029)

## Related Content

---

### A Hierarchical Security Model for Multimedia Big Data

Min Chen (2014). *International Journal of Multimedia Data Engineering and Management* (pp. 1-13).

[www.irma-international.org/article/a-hierarchical-security-model-for-multimedia-big-data/109075](http://www.irma-international.org/article/a-hierarchical-security-model-for-multimedia-big-data/109075)

### Efficient Imbalanced Multimedia Concept Retrieval by Deep Learning on Spark Clusters

Yilin Yan, Min Chen, Saad Sadiq and Mei-Ling Shyu (2017). *International Journal of Multimedia Data Engineering and Management* (pp. 1-20).

[www.irma-international.org/article/efficient-imbalanced-multimedia-concept-retrieval-by-deep-learning-on-spark-clusters/176638](http://www.irma-international.org/article/efficient-imbalanced-multimedia-concept-retrieval-by-deep-learning-on-spark-clusters/176638)

### An Analysis of Human Emotions by Utilizing Wavelet Features

Soo-Yeon Ji, Bong Keun Jeong and Dong Hyun Jeong (2019). *International Journal of Multimedia Data Engineering and Management* (pp. 46-63).

[www.irma-international.org/article/an-analysis-of-human-emotions-by-utilizing-wavelet-features/245263](http://www.irma-international.org/article/an-analysis-of-human-emotions-by-utilizing-wavelet-features/245263)

### Low Complexity Single Color Image Dehazing Technique

SANGITA ROY and Sheli Sinha Chaudhuri (2018). *Intelligent Multidimensional Data and Image Processing* (pp. 116-129).

[www.irma-international.org/chapter/low-complexity-single-color-image-dehazing-technique/207894](http://www.irma-international.org/chapter/low-complexity-single-color-image-dehazing-technique/207894)

### Maxout Networks for Visual Recognition

Gabriel Castaneda, Paul Morris and Taghi M. Khoshgoftaar (2019). *International Journal of Multimedia Data Engineering and Management* (pp. 1-25).

[www.irma-international.org/article/maxout-networks-for-visual-recognition/245261](http://www.irma-international.org/article/maxout-networks-for-visual-recognition/245261)