# CoDiP2P: A Peer-to-Peer Architecture for Sharing Computing Resources

D. Castellà, I. Barri, J. Rius, F. Giné, F. Solsona, and F. Guirado

Departamento de Informática e Ingeniería Industrial, Universitat de Lleida,
Jaume II 69, 25001 Lleida, Spain
{dcastella,ibarri,jrius,sisco,francesc}@diei.udl.cat

**Summary.** Peer-to-Peer (P2P) computing, the harnessing of idle compute cycles through the Internet, offers new research challenges in the domain of distributed computing. This paper presents CoDiP2P, a Computing Distributed architecture using the P2P paradigm. CoDiP2P allows computing resources from ordinary users to be shared in an open access by means of creating dynamic areas of computing resources in a completely distributed, scalable and fault tolerant way. This paper discusses its system architecture and evaluates its functionality by means of simulation.

**Keywords:** Peer-to-Peer computing, p2p simulation, distributed computing, job scheduling.

## 1 Introduction

During recent years, there has been an important increase in computing power requirements in different research fields (life and earth sciences, chemistry, etc.). Due to the expansion of the computer industry, this need has also been observed in business (i.e. financial models) and domestic scenarios (i.e. games). These new necessities have motivated the development of new distributed computing paradigms that use Internet as a single large virtual computer, reducing the elevated cost of the specialized environments, like supercomputers, needed to execute these applications with high computing requirements.

P2P computation [1, 2, 3] represents an emergent low-cost alternative to supercomputers and cluster systems, providing access to distributed computational resources in a scalable and fault-tolerant way. P2P architectures take advantage of the under utilization of personal computers, integrating them into a platform based on the sharing of computational resources between geographically distributed equals. While the P2P paradigm can not hope to serve as a totally general-purpose efficient parallel computer, it can still serve as an excellent platform with unlimited computational resources for solving a wide variety of computational problems: (1) Allowing processing-limited devices, such as wireless clients, to distribute their processing requirements to other machines in the network or (2) Executing embarrassingly parallel programs or (3) Distributing libraries with high computational requirements (i.e. a rendering library) through Internet.