Available online at www.sciencedirect.com
ScienceDirect
Future Generation Computer Systems 23 (2007) 939–955
FGCS FUTURE GENERATION COMPUTER SYSTEMS
www.elsevier.com/locate/fgcs

A dependable Peer-to-Peer computing platform

Hong Wang[a], Hiroyuki Takizawa[a,b], Hiroaki Kobayashi[b,a,*]

[a] *Graduate School of Information Sciences, Tohoku University, Sendai 980-8579, Japan*
[b] *Information Synergy Center, Tohoku University, Sendai 980-8578, Japan*

Received 22 August 2006; received in revised form 6 March 2007; accepted 8 March 2007
Available online 24 March 2007

Abstract

This paper discusses a dependable and widely applicable Peer-to-Peer (P2P) computing platform. As the existing P2P computing platforms are limited due to the lack of support for various computational models, this paper proposes a workflow management mechanism to support task dependency in parallel programs while increasing computing efficiency. In general, task dependency leads to a serious performance degradation for failed task re-execution because of volatile peers. Therefore, it results in low dependability. Here, dependability is defined as a comparison of the actual performance with task failures to the theoretical one without failure on a P2P computing platform. Redundant task dispatch and a runtime optimization method are proposed to guarantee high dependability even with highly volatile peers. Large-scale simulation results indicate that the computing platform efficiently solves the problem of P2P computing due to volatile peers.

Keywords: Peer-to-Peer; Workflow management; Redundant task dispatch; Runtime optimization

1. Introduction

The recent research and development in both academia and industry spearhead the demand for more computing power. However, the world's computing power is no longer primarily concentrated in supercomputer centers. Instead, it is distributed in hundreds of millions of personal computers and game consoles. Internet-connected individual computers have the potential to provide more computing power than any supercomputer, cluster, or grid, and the disparity will grow over time.

The aim of P2P computing [1] (also known as "Global Computing" or "Public-resource Computing") is to use Internet-connected individual computers to solve computing problems, especially large-scale scientific computational problems. In the mid-1990s, there were two pioneering research projects, including the GIMPS [2] (The Great Internet Mersenne Prime Search) and Distributed.net [3]. GIMPS is a distributed computing project researching Mersenne prime numbers. Distributed.net is a general purpose computing platform. In 1999, SETI@home [4,5] was launched by the Space Sciences Laboratory, at the University of California, Berkeley. The purpose of SETI@home is to analyze incoming data from the Arecibo radio telescope and search for possible evidence of radio transmissions from extraterrestrial intelligent forms of life.

These projects are rather successful. GIMPS has already found a total of 9 Mersenne primes, each of which was the largest known prime number at the time of discovery. While SETI@home has not found any conclusive signs of extraterrestrial intelligence, it has identified several candidate spots for further analysis. Distributed.net has successfully providedthe solutions for the DES, RC5-32/12/7 ("RC5-56"), and RC5-32/12/8 ("RC5-64") RSA secret-key challenges.

Nowadays, there are several new P2P computing platforms such as BOINC [6] (Berkeley Open Infrastructure for Network Computing), XtremWeb [7], Entropia [8], Alchemi [9], and JNGI [10]. By November 2006, BOINC could provide a sustained processing power of 350 Teraflops [11]. In contrast, the fastest conventional supercomputer, BlueGene/L achieves a maximum LINPACK performance of 280.6 Teraflops [12]. The number of Internet-connected PCs is projected to grow from approximately 150 million to 1 billion by 2015 [13]. Thus, it has the potential to provide many Petaflops of computing power.

* Corresponding author at: Information Synergy Center, Tohoku University, Sendai 980-8578, Japan. Tel.: +81 22 795 3415; fax: +81 22 795 6096.
E-mail address: koba@isc.tohoku.ac.jp (H. Kobayashi).

0167-739X/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.future.2007.03.004

Embarrassingly parallel computation is a kind of computation that can obviously be divided into a number of completely independent tasks, each of which can be executed by a separate processor (or computer). Embarrassingly parallel problems are ideally suited to P2P computing. However, they constitute only a small minority of all computational problems. As many computational problems cannot be divided into independent tasks, P2P computing platforms have to support the task dependency for most computational jobs.

To make the P2P computing platforms more applicable for general uses, computational problems with task dependencies have to be supported. One method to solve this problem is to define the workflow of a computational job with the dependency among tasks, and to control the task process sequence with the workflow.

The dependency among tasks will result in a status that none of the un-dispatched tasks can be dispatched while the dependency is not satisfied. This status will lead to a performance degradation for idle workers.[1] In addition, because P2P computing systems consist of volatile peers, the frequent peer failure results in further serious performance degradation. A mechanism that makes use of the idle workers to guarantee a low performance degradation for task failure is required to enhance both efficiency and reliability of P2P computing platforms.

This paper designs a dependable and widely applicable P2P computing platform. To extend the applicable area, we first introduce the workflow management concept to P2P computing. In order to improve the efficiency of the P2P computing platform with workflow management, we propose a redundant task dispatch mechanism to avoid the overheads caused by the task failures. Redundant task dispatch uses the computing power from the idle workers to process duplicated tasks; thus the task failure rate can be reduced. Therefore, the performance degradation can be reduced. However, unlimited redundant task dispatch may result in a performance degradation, because too much computing power is wasted due to the processing of duplicated tasks. As the optimal value for the number of redundant tasks depends on the runtime parameters that are unknown in a real computing environment, we finally introduce a runtime optimization for the redundant task dispatch to find the optimal value.

The rest of the paper is organized as follows. Section 2 explores related work and addresses the important research issues to be discussed throughout this paper. Section 3 proposes the design of a dependable P2P computing platform. Section 4 evaluates the proposed P2P computing platform using a simulator, in terms of its dependability. Section 5 concludes and summarizes the paper.

2. Related work

To our knowledge, no workflow management mechanism exists for P2P computing platforms. This is due to the fact that P2P computing platforms are mostly used for solving embarrassingly parallel problems and not for other kinds of computational problems. Workflow management mechanisms have been studied in depth for grid systems.

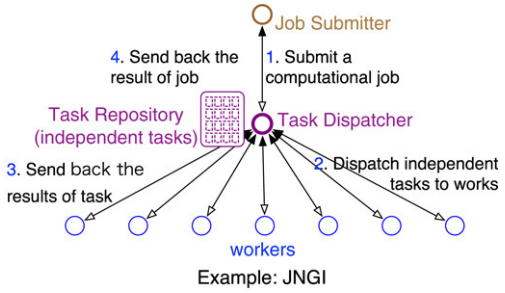

Fig. 1. General architecture of P2P computing platforms.

2.1. P2P computing platforms

BOINC [6] harnesses hundreds and thousands of PCs across the Internet to process a massive amount of data from different scientific computing problems, including radio telescope data analysis, protein shapes analysis, and searching for spinning neutron stars. It works on heterogeneous computers running Windows, Mac, and variants of UNIX-like operating systems.

XtremWeb [7] consists of three kinds of peers: the coordinator, the workers, and the clients. Compared to some other platforms, it allows multiple coordinators and clients (workers keep a collection of coordinators' and clients' IP addresses).

Entropia [8] is a Windows P2P computing system for enterprise networks. It is distinguished from the others for its sandboxed task execution. Binary modification is used to intercept all important Windows API calls.

Alchemi [9] is a.NET-based framework for constructing desktop grids and developing grid applications. It supports an object-oriented application programming model in addition to a file-based job model. Cross-platform support is provided via a web service.

JNGI [10] is a P2P computing platform based on JXTA [14,15]. The platform utilizes the JXTA peer-to-peer communication protocol [16] to construct the overlay network. To improve scalability, it organizes computational resources into groups through the support of JXTA peer groups. Thus, the top level monitor will not be the bottleneck.

The general architecture of existing P2P computing platforms is shown in Fig. 1. Using the master–worker model, it usually consists of two kinds of peers: master peers and worker peers. For most existing platforms, the master peer is a specified server. Worker peers are dynamic peers of the P2P platform.

Here, JNGI is used as an example of a P2P platform. A computational job of JNGI consists of independent tasks of the embarrassingly parallel computation model. A task is a Java program with its own data. Tasks are distributed among worker peers to process. A computational job is processed by JNGI in the following four steps.

[1] A worker is a volunteer peer that provides its computing power.