Peer-to-Peer Desktop Grids Based on an Adaptive Decentralized Scheduling Mechanism

H. Arafat Ali, Mansoura University, Egypt
A.I. Saleh, Mansoura University, Egypt
Amany M. Sarhan, Mansoura University, Egypt
Abdulrahman A. Azab, Mansoura University, Egypt

ABSTRACT

This article proposes an adaptive fuzzy logic based decentralized scheduling mechanism that will be suitable for dynamic computing environment in which matchmaking is achieved between resource requirements of outstanding tasks and resource capabilities of available workers. Feasibility of the proposed method is done via real-time system. Experimental results show that implementing the proposed fuzzy matchmaking based scheduling mechanism maximized the resource utilization of executing workers without exceeding the maximum execution time of the task. It is concluded that the efficiency of FMA-based decentralized scheduling, in the case of parallel execution, is reduced by increasing the number of subtasks.

Keywords: Fuzzy Logic, Grid scheduling, High Throughput Applications, P2P Desktop

INTRODUCTION

Grid has recently emerged as a promising paradigm for high performance or high throughput computing because of the vast development of powerful computers and high-speed network technologies as well as low-cost servers (GIMPS, 2009). Grid aims to aggregate heterogeneous, large-scale, and multiple-institutional resources, and to provide the transparent, secure, and coordinated access to various computing resources (supercomputer, cluster, scientific instruments, database, storage, etc.) owned by multiple institutions by making virtual organization. Grid computing is regarded to be the future of Semantic Web, the next step in distributed networking (SETI@home, 2009).

A peer-to-peer (or “P2P”) computer network exploits diverse connectivity between participants in a network and the cumulative...
bandwidth of network participants rather than conventional centralized resources where a relatively low number of servers provide the core value to a service or application. Peer-to-peer networks are typically used for connecting nodes via largely ad hoc connections. Many peer-to-peer networks are overlay networks because they run on top of the Internet. The overall goal of Peer-to-Peer (P2P) based systems is to provide (share) resources (like computing power, bandwidth or storage). A fundamental principle of the P2P paradigm is equality. P2P computing can be defined as the sharing of computer resources and services by direct exchange.

Desktop Grid

Desktop Grid has recently been an attractive computing paradigm for high throughput applications (Anderson, 2004). However, Desktop Grid computing is complicated by heterogeneous capabilities, failures, volatility, and lack of trust because it is based on desktop computers at the edge of the Internet. In a Desktop Grid computing environment, volunteers (that is, resource providers) have heterogeneous properties such as CPU, memory, network, and so forth.

Desktop Grid has recently received a rapidly growing interest and attraction because of its success of the most popular examples like GIMPS (2009), and SETI@Home (2009). Some studies have been made on Desktop Grid systems which provide an underlying platform: BOINC (Anderson, 2004), XtremWeb (Cappello et al., 2005), Entropia (Chien, Calder, Elbert, & Bhatia, 2003), Bayanihan (Sarmenta, 2002), Javelin (Neary & Cappello, 2005), Computer Power Market (CPM) (Ping, Sodhy, Yong, Haron, & Buyya, 2004), POPCORN (Nisan, London, Regev, & Camiel, 1998), Cluster Computing On the Fly (CCOF) (Lo, Zhou, Zappala, Liu, & Zhao, 2004), Organic Grid (Chakravarti, Baumgartner, & Lauria, 2006), Messor (Babaoglu, Meling, & Montresor, 2002), Paradropper (Montresor, Meling, & Babaoglu, 2002), Condor (Thain, Tannenbaum, & Livny, 2005), and so forth. Although, all of the previous examples had been investigated as valuable techniques concerning Desktop Grid issues, there still remain hindrances in achieving the promised performance especially in fault tolerance and decentralization.

A Grid is a structure with distributed heterogeneous resources offered to users. Users submit tasks that should be efficiently processed using resources available on the Grid (Nabrzyski, Schopf, & Weglarz, 2003). A Desktop Grid is usually built on the Internet platform in which resources are unreliable and frequently turned off or disconnected (El-Desoky, Hisham, & Abdulrahman, 2006). The idea of desktop grid is to harvest the idle time of Internet connected computers, to run very large and distributed applications (Fedak, Germain, Vincent, & Franck, 2001). Machines (i.e., Desktops) in a desktop grid are basically categorized into: a) Client: Machine from which a user can submit grid tasks for execution, and b) Worker: Computing resource on which grid tasks can be executed. In P2P Desktop Grid environment (David, 2002), each node acts as a peer, so that, it can act as a Worker or as a Client.

A Grid worker (i.e., computing resource) is a basic device where tasks are scheduled/processed/assigned (Blazewicz, Brauner, & Finke, 2004). It is a set of cumulative resources (CPUs, memory, and storage space) with limited capacities. In Desktop Grid environment, a worker may be unavailable due to maintenance, disconnection, or a breakdown. A Grid task (job, activity) is a basic entity which is scheduled over the workers. Grid tasks can be categorized into: a) CPU intensive tasks, b) Memory intensive tasks, and c) I/O intensive tasks. A Grid task requirements include: a) Required CPU (i.e., computational power), b) Required memory (i.e., physical memory), and c) Required desk-space. In computational grids, the required desk-space of a task is usually too little to be taken into account, so, it can be neglected. In scientific applications, a task has specific requirements on the amounts and types of workers, or required time intervals on these workers, where the task can be scheduled.
Related Content

Efficient Communication Interfaces for Distributed Energy Resources
[www.irma-international.org/article/efficient-communication-interfaces-distributed-energy/43882/](www.irma-international.org/article/efficient-communication-interfaces-distributed-energy/43882/)

Energy-Efficient Optical Interconnects in Cloud Computing Infrastructures
[www.irma-international.org/chapter/energy-efficient-optical-interconnects-in-cloud-computing-infrastructures/82539/](www.irma-international.org/chapter/energy-efficient-optical-interconnects-in-cloud-computing-infrastructures/82539/)

Visualization of Large-Scale Distributed Data
[www.irma-international.org/chapter/visualization-large-scale-distributed-data/62830/](www.irma-international.org/chapter/visualization-large-scale-distributed-data/62830/)

Algorithms for Maintaining Consistency of Cached Data for Mobile Clients in Distributed File System

Fast Paxos Made Easy: Theory and Implementation
[www.irma-international.org/article/fast-paxos-made-easy/120458/](www.irma-international.org/article/fast-paxos-made-easy/120458/)