Chapter 12
High Performance Computing Design by Code Migration for Distributed Desktop Computing Grids

Makoto Yoshida
Okayama University of Science, Japan

Kazumine Kojima
Okayama University of Science, Japan

ABSTRACT

Large scale loosely coupled PCs can organize clusters and form desktop computing grids on sharing each processing power; power of PCs, transaction distributions, network scales, network delays, and code migration algorithms characterize the performance of the computing grids. This article describes the design methodologies of workload management in distributed desktop computing grids. Based on the code migration experiments, transfer policy for computation was determined and several simulations for location policies were examined, and the design methodologies for distributed desktop computing grids are derived from the simulation results. The language for distributed desktop computing is designed to accomplish the design methodologies.

INTRODUCTION

Grid system is a type of distributed systems, and it can be categorized into three classes: computing grid, data grid and service grid (Krauter et al., 2002; Choi et al., 2008). This article deals with the computing grid. The purpose of the computing grid is to use the existing computing powers effectively, those scattered around in one or more organizations. Several hundred PCs, such as Note PCs, Desktop PCs and WSs, are assumed in our model. User jobs, namely transactions, can be executed on the local or remote computer systems depending upon the workload in the system. If

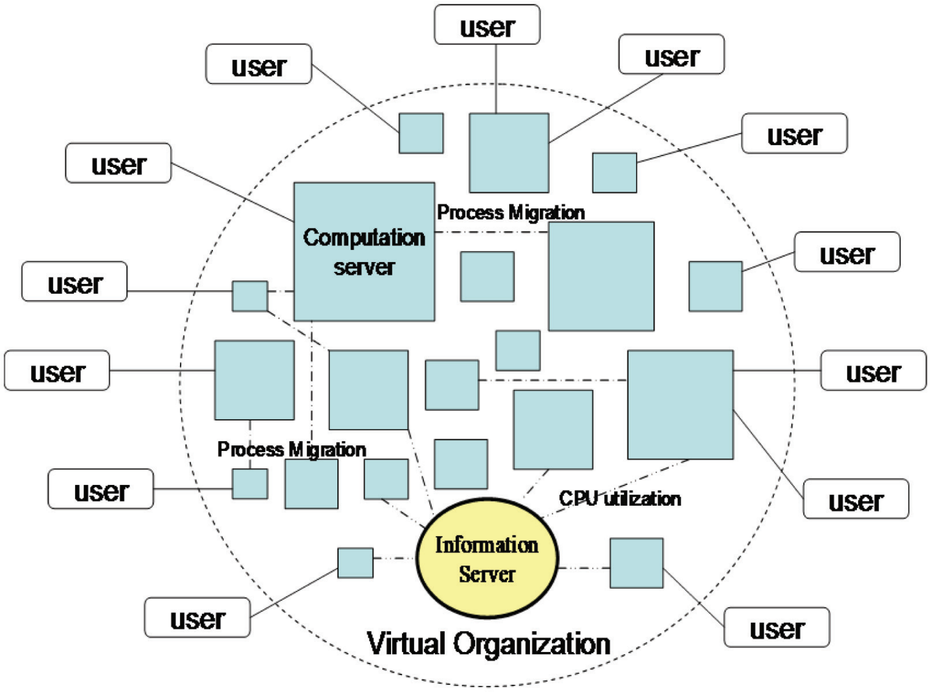
DOI: 10.4018/978-1-4666-2065-0.ch012

the workload of the system that the job entered is heavy enough for processing, it must be properly migrated to the other site through migration. However, the effectiveness of the code migration against the migration overhead must be evaluated (Coulouris et al., 2005).

The network of the computing grids we modeled is shown in Figure 1. It consists of many computing nodes and one information server. On decentralized control, it does not have the information server. Figure 2 shows the taxonomy of the distributed computing systems (Venugopal et al., 2006; Theotokis & Spinellis, 2004). The shadow area is the topics of this article. For migrating the transactions, both the transfer policy that determines the migration decision and the location policy that determines the site to be migrated must be properly provided according to the application environment (Shah et al., 2007). The environment of the system is getting more complex with the increasing and diversifying applications. However, little work for the code migration has been associated with the application environment. Depending upon the migration policy and the environment situated, the performance of the system greatly varies. The scales of transactions and networks, the workload patterns, the power of computers, code migration algorithms, all of these characterize the performance of the system. The relations of these characteristics, the pros and cons, must be analyzed, and the boundaries of the code migration must be specified in designing the distributed computing grids.

We need some methodologies to design the workload of transactions in various complex application environments to have a good performance system. Shah (2007) has shown the theoretical approach for code migration. We develop the practical approach in this article. Code migration overhead was exemplified by implementing the code migration system. Based on the experiments, several centralized and decentralized al-

Figure 1. Computing grid model


user
user
user
user
user
user
user
user
user
user
user
user
user
user
Process Migration
Computation server
Process Migration
CPU utilization
Information Server
Virtual Organization