DiMAVE, Object Sharing in Distributed Multi Agent Volunteer Environment

C. Shanthi and V. Revathi

Abstract--- Goal of the system is utilizing computing power from volunteer network. At first, this system frames a cluster of volunteer's devices like laptop, mobile. Then the system perform complex operation which needs huge computing power by breaking the work as units and sends them across agents installed on volunteer's cluster. Each agent does the work by using volunteer's computing power and sends the results back. Finally the system combines the outputs. Main applicable areas are the work that need huge computing power such as image comparison. The framework includes components that act local servers for dynamic object ownership and a system coordinator for managing and monitoring the system, taking solution oriented responsibilities for unwanted/unexpected situations. An interface is provided to enable users to use simple object access methods in a completely transparent manner.

Index Terms--- Computing Power, Multi Agent Volunteer Environment, Object Sharing, Distributed Systems

I. INTRODUCTION

Distributed systems [1] provide communication and coordination between several computer software and hardware components through messages. The existence of a network and transparency of system details are most distinctive properties of a distributed system. Users work on a multi- processor distributed network but perceive the system as a single virtual machine. Application based operations such as sending and receiving files are executed by users. Developing object sharing applications takes a long time and the heterogeneous structure of the system causes some challenges. The software layer distinguishes distributed systems from other network systems. In general, distributed systems are used because of sharing requirements. The shared resources can be hardware components such as discs, printers or software components such as databases, objects or files, here in our system it's a executable files. Our work concentrates on power sharing, namely objects.

Distributed systems usually use remote object models for object sharing [2], [3]. In a remote object model, object is located on a remote computer and users access this object by using a proxy. Remote object models are usually weak when scalability is considered and their non-flexible structure brings limitations.

Sharing objects on a distributed network and local object access operations are important for load balancing, performance, protection and reusability. In this paper, we present a multi-agent volunteer framework which constructs volunteer distributed network in order to utilizing computing power of volunteer agent devices. If an agent makes too many write accesses or it is already the object owner, object accesses occur locally and therefore are very fast. When ownership changes in distributed network, local object copy spreads around and in case of an error the in system, object copies can be found for restore operations.

II. DISTRIBUTED SHARED OBJECT

Distributed shared object is a software unit that is used in a network which is composed of computers working in collaboration or multi-processor computers. Distributed objects and local objects differ in certain aspects. Life

C. Shanthi, MCA, M.Phil, Research Scholar, Assistant Professor, Vels University, Pallavaram
V. Revathi, Research Scholar, Vels University, Pallavaram

cycle; creation, transport or delete operations should be announced to the network in the distributed systems. Reference; distributed object reference is more complex than local object references. Request latency; object access time is usually longer than local object access. Object activity; in distributed systems shared object may not be available at all times. Parallelism; distributed systems usually provide a high level of parallelism. Communication; distributed systems have various communication alternatives. Failure; in distributed systems, object access failure probability is higher than local object accesses. Safety; objects live on various computers and are transferred on the network so it is difficult to ensure security. Our framework allows local access for owned objects. In some applications objects can be partitioned into sub- objects transparent tousers.

These sub-objects together constitute a single composite object. In these models users can acquire and access parts of objects in which they are interested. In our application, a shared distributed object is a single serializable Java object.

III. MULTI AGENT VOLUNTEER ENVIRONMENT (MAVE)

To frame a MAVE system, we need an application to which volunteers should donate their computing power. They can precisely mention percentage of the computing power they can donate. Computing power can be donated from any device such as desktop, laptop, tablet and mobile. Followed by, volunteer can download an agent which can send volunteers environment details such as processor type, processor speed, and total main memory, to a leader agent.

A. A Leader Agent

A leader agent receives a request that needs huge computing power to process, for example image comparison. The leader can split the tasks that are involved in processing the requests into small number of units. Then it first finds available agents based on the computing power from the volunteer network. Subsequently it assigns the unit of work each available agents and coordinate with them until the unit of work is executed by the corresponding agents. Finally the leader agent merges all outputs and sends the response back.

One of the agents will be selected as leader when distributed environment is up by using leader election algorithm (Bully Algorithm). When anyone of the agents is not getting response from the leader, then, it assumes that leader is dead. Followed by new election will be began and new leader will be elected.

B. Agent Repository System

This is the only centralized system in the distributed environment. From which agent subscription application will be downloaded by volunteers. It also maintains agent details such as agent IP address and its status. Whenever new agent is added he will be registered in the repository system.

IV. CONCLUSION

This paper presents a framework which enables distributed object sharing in a multi agent volunteer environment. Volunteer network will be framed by volunteers' consensus and percentages of computing power from his/her devices. The system solves requests that need huge computing power. The additional benefits of the proposed shared object model are multi- agent system compatibility, utilizing enormous computing power, ease of application development, conserved bandwidth consumption, distribution transparency, and deployment of shared objects, dynamic object ownership, and resistance to adverse conditions.

REFERENCE

[1] Andrew S. Tanenbaum, Maarten Van Steen, "Distributed Systems Principles and Paradigms" 2nd Edition 2007.

[2] Daniel Hagimont, Fabienne Boyer (2001), "A Configurable RMI Mechanism for Sharing Distributed Java Objects". Sirac Laboratory (INPG-INRIA-UJF). IEEE 2001.

[3] Dacian Tudor, Georgiana Macariu, Vladimir Cretu, Wolfgang Schreiner, "Shared Data Grid Programming Improvements using Specialized

Objects", International Conference on Complex, Intelligent and Software Intensive Systems. IEEE 2010.

[4] Guray Yilmaz, Nadia Erdogan, "Distributed Composite Objects: A New Object Model for Cooperative Applications" Springer- Verlag Berlin Heidelberg 2004.