NATIONAL CONFERENCE ON INNOVATIVE TRENDS IN ENGINEERING & TECHNOLOGY – NITET-18
16-17th March 2018
NOVATEUR PUBLICATIONS
International Journal Of Innovations in Engineering Research And Technology [IJIERT]
ISSN: 2394-3696

ANALYSIS OF A SEQUENTIAL JAVA CODE FOR AUTOMATIC PARALLELIZATION IN DISTRIBUTED NETWORKS FOR HIGH PERFORMANCE COMPUTING –A RUNTIME APPROACH

Ms. Sonam Shaikh
BE CSE

Prof. Mr.Anil Patil
(Assistant. Prof.)
Dept. of CSE

ABSTRACT
The popularity of the Internet and the availability of powerful computers and high-speed networks as low cost commodity components are changing the way we use computers today. This technology opportunity has led to the possibility of using networks of computers as a single, unified computing resource. It is possible to use multiple computers as a single unified resource
for high performance computing. The Java Development Environment (JDE) includes several features which facilitate the production of stable, robust code for parallel processing. Threads provide an efficient and effective paradigm for utilizing tightly coupled systems and socket communication, remote method invocation (RMI) are for distributing environments. The work focuses on an automatic parallelization of a Java code to improve the performance of a program execution. At run time it parallelizes the code for execution. The research focuses on a new parallel execution technique in which the sequential java code is analyzed and parallelized so that each part of code is executed on different system in a network in accordance with the availability of the computers in a network. It speeds up the execution of a particular application to a great extent. The parallelized code should be such that it should not be dependent with each other. Dependencies among the code should be detected at compile time / run time as far as possible.
This proposed work also involves load management as well as maintaining network information at regular intervals. Since java is an object oriented language, the management of the objects requires a special module. The proposed work makes it easier for the programs which have large number of lines of code to be executed and saving a time and is solely used for making the parallelization of java code through java programming. The goal of this work is to describe the design of a system for the automatic parallelization of Java programs. The novelty of the system stems from its combination of compile-time analysis and run-time support to extract and exploit parallelism in programs.

KEYWORDS: Distributed computing, Automatic parallelization, Parallel programming, High performance computing, Java program execution, Code distribution, Load balancing, Network monitoring.