Win in the Flat World

Condensed Research Note

An Analysis of Application Migration to Grid Environment

– Srujan Kumar Enaganti

Enterprises implementing grid computing have reported dramatic improvements in performance and scalability for certain category of applications. However, the extent of improvement in performance of an application to be migrated to Grid is not apparent before the code re-write actually takes place. Hence, a systematic prior analysis of the benefits of deploying a legacy application on to a grid can be very useful. This note investigates the issue describing a framework to do the same.

Introduction

Grid computing is evolving as a very useful technology in several domains. Several tools and methods have been developed to develop new applications that can directly run over grid. However, a systematic mechanism to migrate older/legacy applications (that were developed for single processor systems) to run over cluster / grid is not yet available. Before applying one such mechanism, one needs to analyze whether performance and hence business value gained through grid-enabling the application is worth the investment. Enterprise applications typically consist of interconnected components or modules of code and data that work in coordination with each other. When the application is migrated to a grid environment, it needs to be analyzed whether or not the different interconnected components can be broken up and distributed on to various nodes of the grid. In this note, we present a framework to systematically analyze legacy code modules for gridizability and performance. The framework takes legacy code as input and spits out suggested modules that could be re-factored for Grid environment. We call it Grid Application Migration Framework (GAMF).

Grid Application Migration Framework

The framework essentially consists of four parts. These are (i) Grid Code Analyzer, (ii) DAG Reducer, (iii) Cluster Generator, and (iv) Grid Simulator.

GCA is very much like a compiler generating intermediate code that can be represented in the form of a Directed Acyclic Graph (DAG) (Note 1). The architectural details of GCA can be seen in Note 2.

Since the number of nodes in a DAG is quite large, it is computationally infeasible to analyze such a big graph further. To manage the task graph we propose a novel DAG reduction algorithm to reduce the total number of nodes in DAG (Note 3).

We then use clustering algorithms (Note 4) to regroup nodes that can be executed on the same processor. The task nodes within a cluster are executed by the same processor of the grid. The order of execution is specified by the clustering algorithm.

After obtaining the clusters, we evaluate the performance gain obtained by looking at the amount of computation required to execute the program over the grid as compared to what it was when the program was running on a single processor. The framework currently works on C and Java code and is being extended for C++. We are further extending the framework to include other parameters that govern the grid application migration effort to create a holistic migration methodology.

We have taken some sample applications and applied our framework to determine the performance gain achieved through parallelization. We found that the proposed framework is quite effective in analyzing applications for Grid enablement and predicting performance gain.

Conclusion
Migrating application systematically and successfully to grid environment is a critical factor in grid adoption. In the note, we introduce GAMF – a framework that systematically analyzes application for grid suitability and predicts performance gains. The initial results look quite promising.

Note 1: Directed Acyclic Graph
DAG is the data structure to depict program components and relationship among the components in terms of tasks and data dependencies. Nodes of the DAG are code components of the program that can be executed sequentially over a single processor. Edges between nodes represent the data dependencies between the code blocks. Edge weights give the communication cost, i.e., the amount of time required for the data to reach from the source node to destination node in course of program execution. Often this weight is taken proportional to the amount of data transferred since a uniform rate of data transfer is assumed.

Note 2: Grid Code Analyzer (GCA)
Grid Code Analyzer (GCA) consists of two major components: (i) Static Analyzer -- the code is analyzed i.e, broken into tokens and parsed and then instruments are inserted at different places that are used by the Dynamic Analyzer; (ii) Dynamic Analyzer – DAG is generated from the instrumented code generated by static analysis.

Note 3: DAG Reduction Algorithm
The principle behind the algorithm is to remove those nodes in the graph that do not contribute to the overall performance of the application by merging them with their parent nodes. Special care has been taken to avoid formation of loops in the DAG. The principle of replication of smaller blocks is also made use of in the design of the algorithm.

Note 4: Clustering Algorithms
There are several clustering algorithms, available in literature, that deal with the problem of scheduling a task graph on to a cluster of computers. Edge-Zeroing (EZ) Algorithm and Dominant Sequence Clustering (DSC) algorithm are two of them that are well known and widely used.

Condensed Research Note (CRN) is a monthly research communiqué from Software Engineering and Technology Labs (SETLabs). All rights reserved.

About Infosys Technologies Ltd. (NASDAQ: INFY)

Infosys (NASDAQ:INFY) defines, designs and delivers IT enabled business solutions. These provide you with strategic differentiation and operational superiority, thereby increasing your competitiveness. Each solution is delivered with the industry-benchmark Infosys Predictability that gives you peace of mind.

With Infosys, you are assured of a transparent business partner, business-IT alignment with flexibility, world-class processes, speed of execution and the power to stretch your IT budget by leveraging the Global Delivery Model that Infosys pioneered.