Interpretive Performance Prediction for High Performance Parallel Computing

by

Manish Parashar

Abstract of Dissertation

July, 1994

Abstract

The key factor contributing to the complexity of parallel application development and the poor utilization of current high performance computing (HPC) systems is the increased degrees of freedom that have to be resolved in such an environment. The primary objective of our research is to address this software development bottleneck.

In this research we develop the *interpretive* approach to performance prediction. The essence of this approach is the application of interpretation techniques to performance prediction through an appropriate characterization of the HPC system and the application. A comprehensive *system characterization* methodology is defined to hierarchically abstract the HPC system into a set of parameters which represent its performance. A corresponding *application characterization* methodology is defined to abstract a high-level application description into a set of parameters which represent its behavior. Performance prediction is then achieved by interpreting the execution costs of the abstracted application in terms of the parameters exported by the abstracted system. Models and heuristics are defined to handle accesses to the memory hierarchy, overlap between computation and communication, and user experimentation with system and run-time parameters. This thesis concentrates on distributed memory HPC systems and uses such a system to illustrate and validate the developed approach.

An interpretive toolkit is designed and implemented to support HPF/Fortran 90D application development. It incorporates the following three systems: (1) *ESP*:

An Interpretive Framework for HPF/Fortran 90D Performance Prediction; (2) *ESP-i*: A HPF/Fortran 90D Functional Interpreter; and (3) *ESPial*: An Integrated Environment for HPF/Fortran 90D Application Development & Execution. The toolkit is supported by an interactive, graphical user interface (*ESPView*) and provides the developer with the following functionality: design evaluation capability, functional verification capability, performance visualization support, experimentation capability, compilation support, and execution support.

A set of application codes and benchmarking kernels are used to validate the accuracy, utility, cost-effectiveness, and usability of the interpretive framework.

The interpretive approach provides an accurate and cost-effective (in terms of time and resources required) evaluation methodology that can be used by any tool supporting HPC (e.g. intelligent compilers, mapping and load-balancing tools, and system design evaluation tools) which has to optimize available design options.