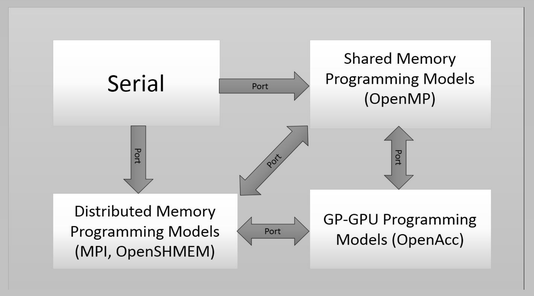
Performance Characterization of Applications Across Different Programming Models Using Similarity Analysis

Md Abdullah Shahneous Bari[1], Abid M. Malik[1], Zhifeng Yun[1], Oscar Hernandez[2] and Barbara Chapman[1]
[1]Department of Computer Science, University Of Houston
[2]Oak Ridge National Laboratory

Background & Motivation



- Different Architectures (Shared Memory, Distributed Memory, GPGPU), different programming models (OpenMP, MPI, OpenAcc)
- No single best model for all the architectures
- Even no single best model for a specific architecture
- Different models bring the best out of different applications
- But no way to know without actually porting the application using the model
- Application developers have to make a choice without really knowing the actual outcome

Proposal

- A framework to predict the application performance behavior on a certain programming model
- Uses Similarity Analysis
- Uses Dynamic Hardware features derived from hardware counters using PAPI [1]
- Similar functions can be ported Similarly [2]
- So these functions will likely perform similarly after being ported to a different programming model

Framework

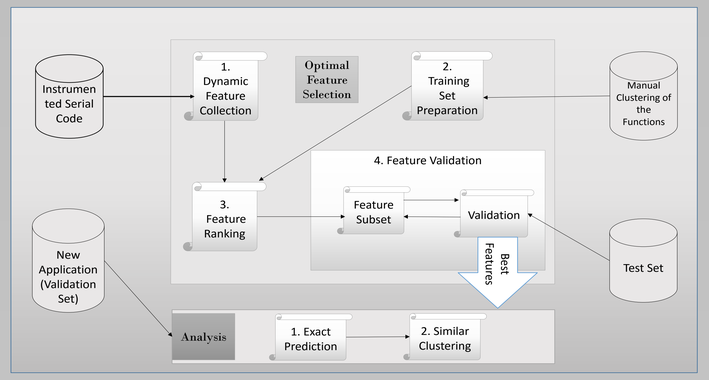


Figure 1: Architectural Design of the Proposed Framework

Two Main Phases:

1) Optimal Feature Selection (4 sub phases)
   - 1.1) Feature Collection
   - 1.2) Training Set Preparation
   - 1.3) Feature Ranking
   - 1.4) Validation

2) Analysis (2 types of results)
   - 2.1) Exact Prediction
     - -Predicts the exact behavior
     - -Typically useful when the training set is large
   - 2.2) Similar Clustering
     - -Not exact prediction
     - -Clusters functions with similar behavior

Experiments and Results

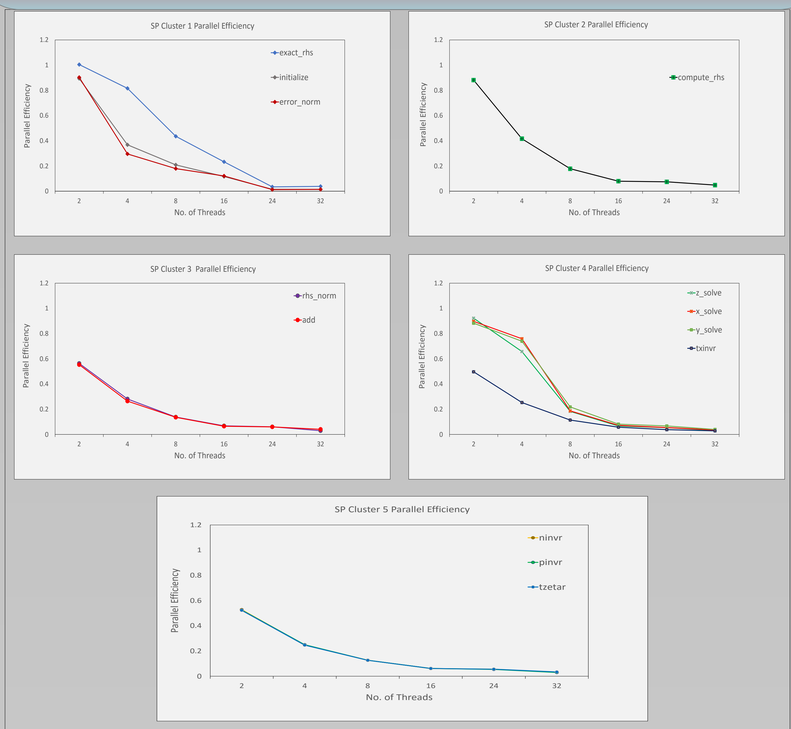


Figure 2: Parallel Efficiency of different clusters for SP

- Used NAS parallel benchmark to validate the framework
- All the functions in the similar cluster has similar parallel efficiency trend
- Only exception was *exact_rhs* in cluster 1 and *txinvr* in cluster 4
- Overall accuracy around 85% in clustering functions with similar behavior

Conclusion

- Proposed a framework to characterize performance behavior on a fine grain level across different programming models
- Validated it by predicting OpenMP parallel efficiency from serial code with 85% accuracy

Future Work

- Characterize other important HPC performance behaviors (eg. Power Consumption, Resilience, I/O)
- Characterize these behaviors across different programming models
- Use static features along with dynamic features

References

1. U. o. T. K. Innovative Computing Laboratory, “PAPI : Performance application programming interface, ” http://icl.cs.utk.edu/projects/papi.

2. W. Ding, C.-H. Hsu, O. Hernandez, B. Chapman, and R. Graham, “Klonos: Similarity-based planning tool support for porting scientific applications,” Concurrency and Computation: Practice and Experience , 2012.

This work was funded by US Department of Energy under contract no. DE-AC05-00OR22725