Meta-Balancer: Automating Load Balancing Decisions

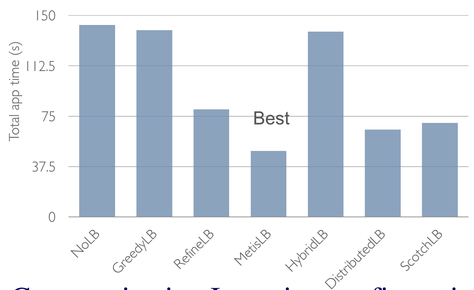
Harshitha Menon, Kavitha Chandrasekar, Laxmikant V. Kale

University of Illinois at Urbana-Champaign

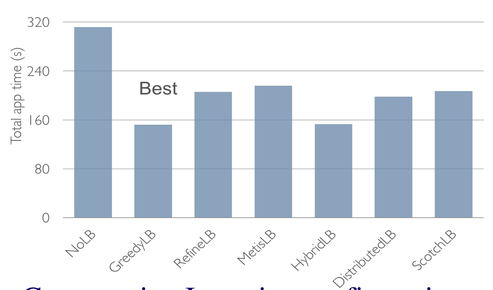
HPC applications are increasingly becoming complex and dynamic. Many applications require dynamic load balancing to achieve high performance and system utilization. Different applications have different characteristics and hence need to use different load balancing strategies. There are many load balancing algorithms available. However, invocation of an unsuited load balancing strategy can lead to inefficient execution. Most commonly, the application programmer decides which load balancer to use based on some educated guess. We propose Meta-Balancer, a framework to automatically decide the best suited load balancing strategy. Meta-Balancer monitors application characteristics and based on that, it chooses an ideal load balancing algorithm to use. In order to predict the best load balancing strategy, Meta-Balancer uses a supervised random forest machine learning technique with the application characteristics as the features. Using this, we are able to achieve high prediction accuracy of 82% on the test set to demonstrate performance benefits of up to 3X.

Motivation

- Load imbalance is a critical factor that can affect the performance of an application
- Different applications require different load balancing strategies
- Application programmer has to choose from numerous load balancing algorithms
- Puts a burden on the user and may not be most efficient



Communication Intensive configuration

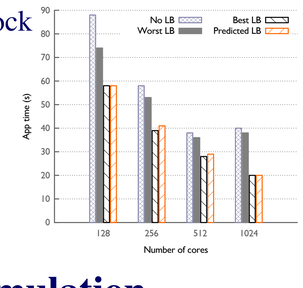


Computation Intensive configuration

Applications

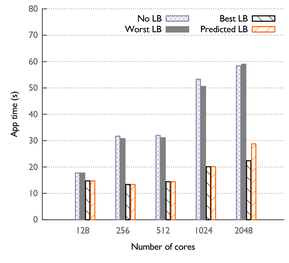
Lassen – LLNL Proxy App

- Used to study denotation shock dynamics
- Suffers from load imbalance
- Meta-Balancer is able to choose a good load balancer
- Meta-Balancer improves performance



Particle-In-Cell Simulation

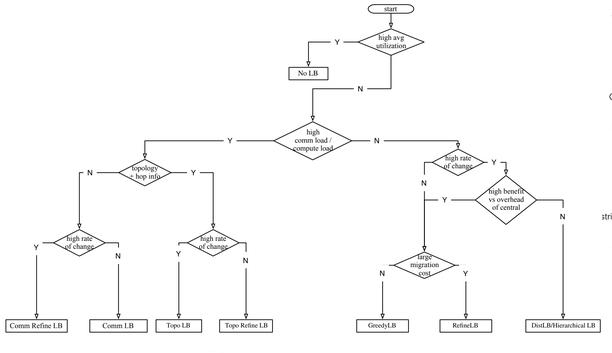
- Used for simulation of plasma particles
- Load imbalance results from imbalanced distribution of particles and particle motion
- Meta-Balancer is able to improve performance with weak-scaling



Meta-Balancer

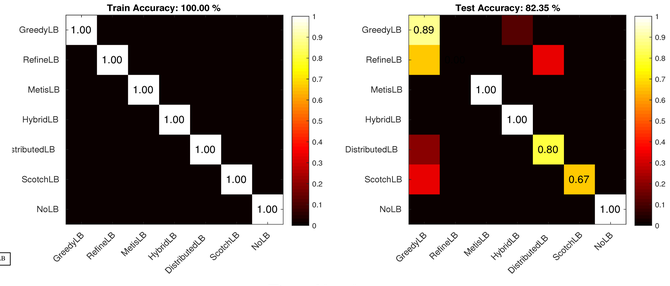
Framework to automate the load balancing decision

- Monitors system and application characteristics
- Statistics collected from the application is used to predict the most suitable load balancer
- Decision tree can be used to guide the selection of statistics



Decision tree

- Supervised Random Forest machine learning technique is used to predict load balancing strategy
  - Choices are : GreedyLB, RefineLB, MetisLB, HybridLB, DistributedLB, ScotchLB, NoLB
- Application & machine characteristics used as features
- Synthetic Benchmark used to generate training set
- Accuracy on test set is 82%
- Trained model is used to predict on other mini-apps

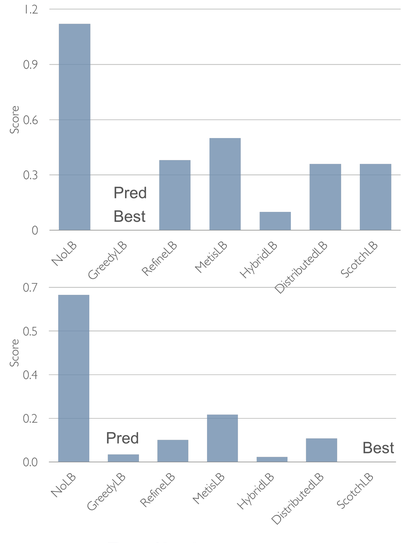


Prediction accuracy

Predictions

$score = T_{predicted\_lb} / T_{best\_lb} - 1$

Larger the score, worse the performance



Prediction scores

Summary

- Proposed Meta-Balancer to automate the load balancing decision
- Meta-Balancer strategy selection collects statistics of the application and machine to choose the best suitable load balancer
- Meta-Balancer uses random forest machine learning technique to choose the load balancing strategy
- Meta-Balancer is able to predict the best suitable load balancer with 82% accuracy on the test set