Load Balancing Scientific Applications

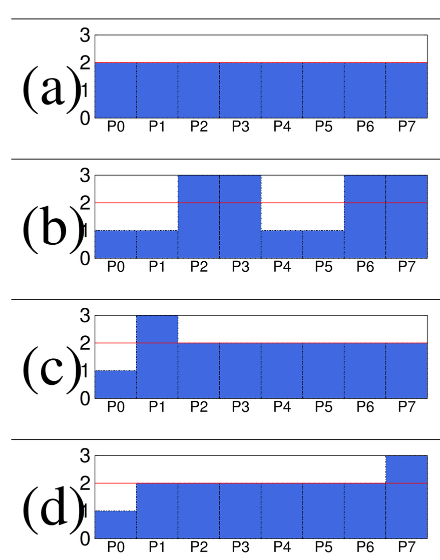
Olga Pearce∗†
∗ Parasol Laboratory, Department of Computer Science and Engineering, Texas A&M University
† Lawrence Livermore National Laboratory

1. Introduction

The dynamic behavior of large modern parallel simulation codes can lead to imbalances in computational load among processors. In this thesis, I address how to evaluate load imbalance at runtime and make its correction affordable.

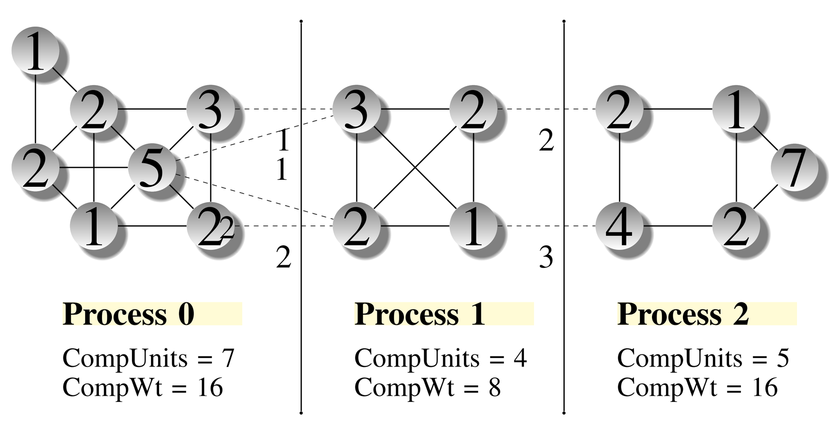
2. Quantifying the Effectiveness of Load Balance Algorithms

- Statistical load metrics do not shed light on how to correct the imbalance
- To evaluate and correct the imbalance, we need to attribute load to the **migratable tasks in the application**

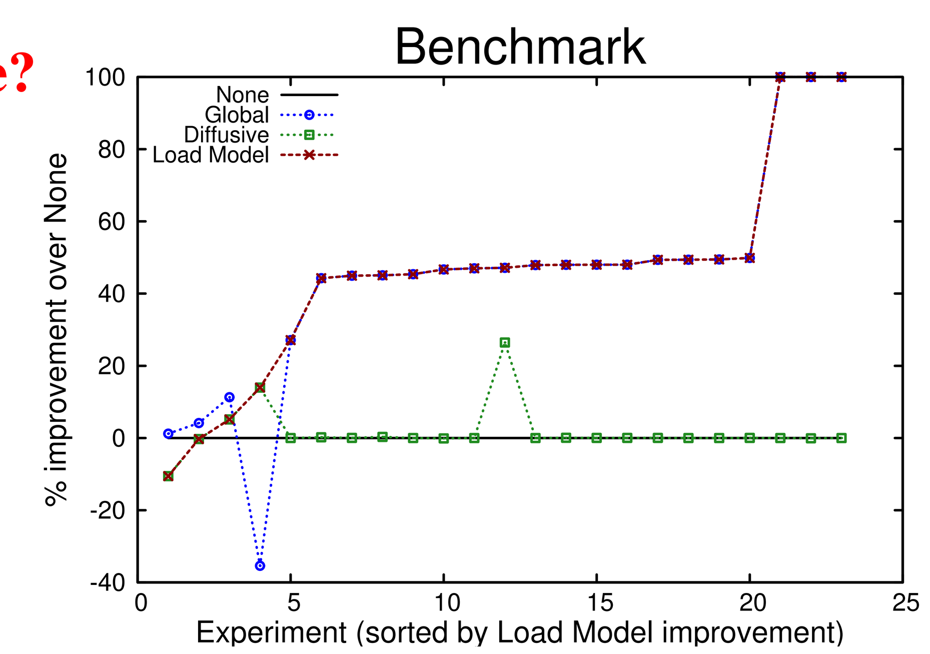


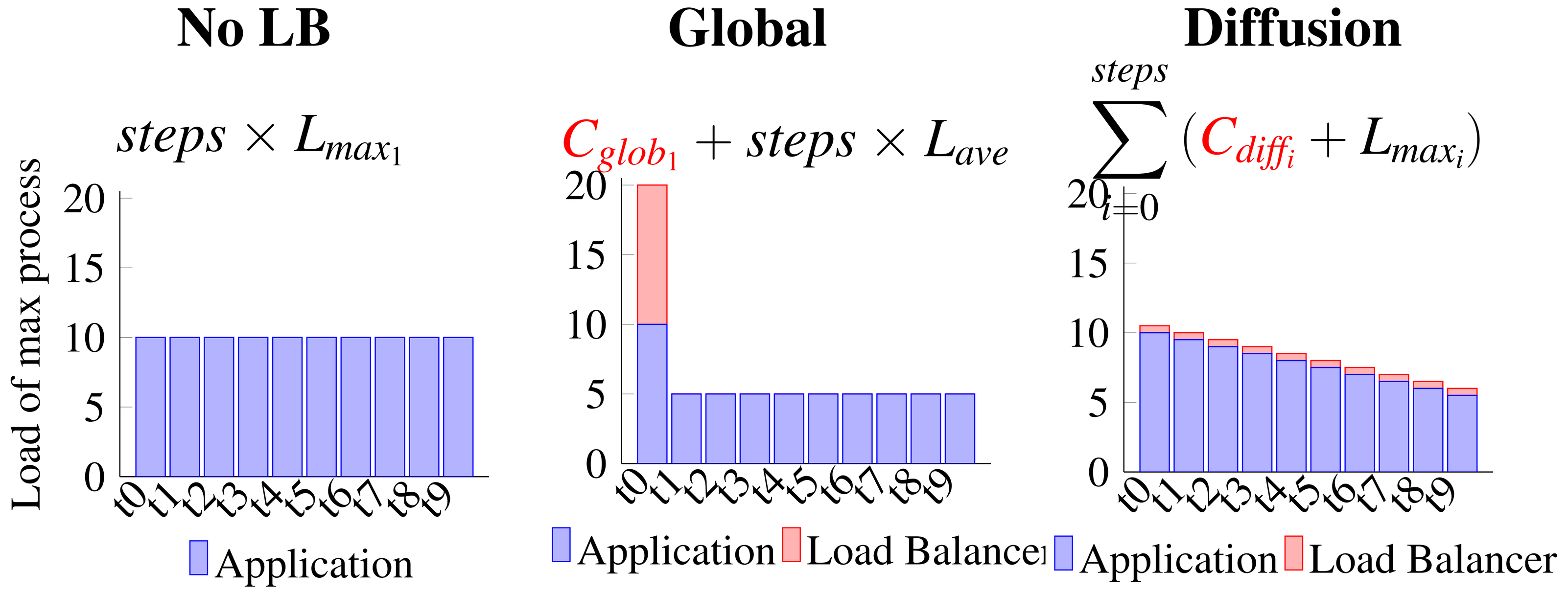
Load/process	mean	imbalance	stdev	skewness	kurtosis
(a)	2	0%	0	0	0
(b)	2	50%	1	1	−2
(c)	2	50%	.5	2	1
(d)	2	50%	.5	2	1

Need a model of:
- Weighted migratable tasks
- Relationships between tasks



When and how should we balance?
- Model costs associated with using different balancing methods
- Use model to select the load balance method that achieves the lowest runtime

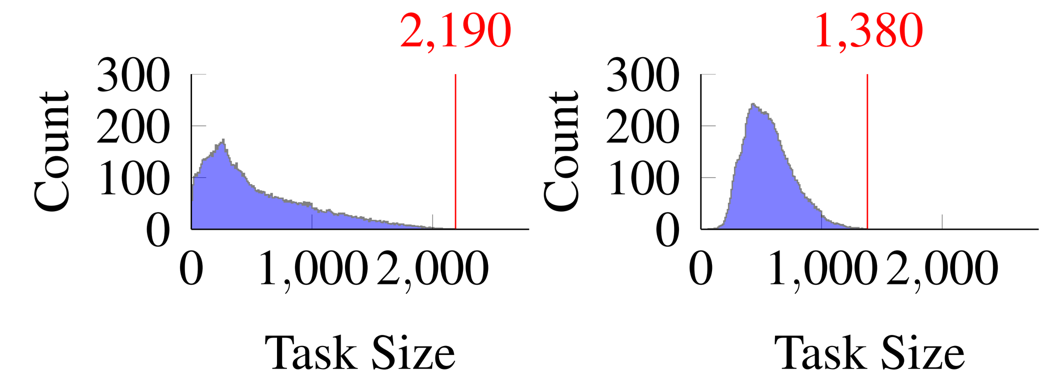




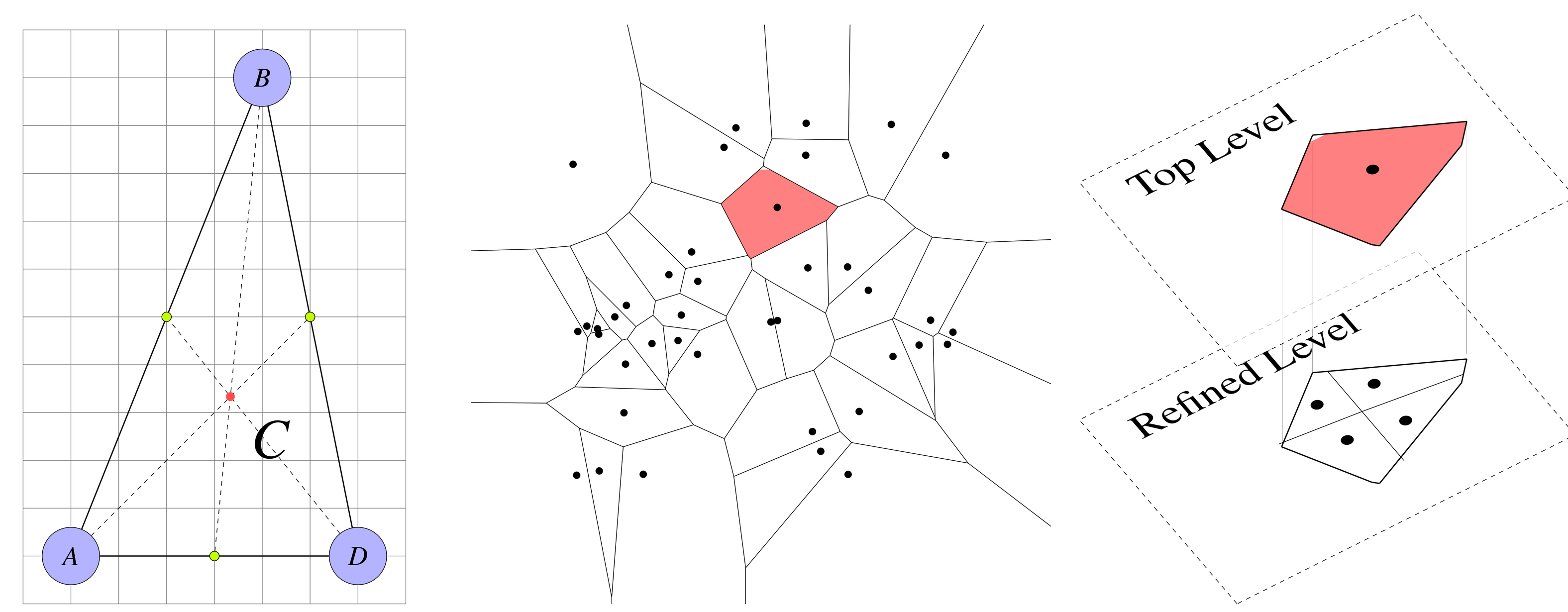
Quantifying the Effectiveness of Load Balance Algorithms, Olga Pearce, Todd Gamblin, Bronis de Supinski, Martin Schulz, Nancy M. Amato, *In Proc. ACM Intl. Conf. on Supercomputing (ICS)*, June 2012.

3. Efficient Load Balance Algorithm for N-Body Simulations with Non-Uniform Density

Tasks of Highly Variable Sizes are Difficult to Balance

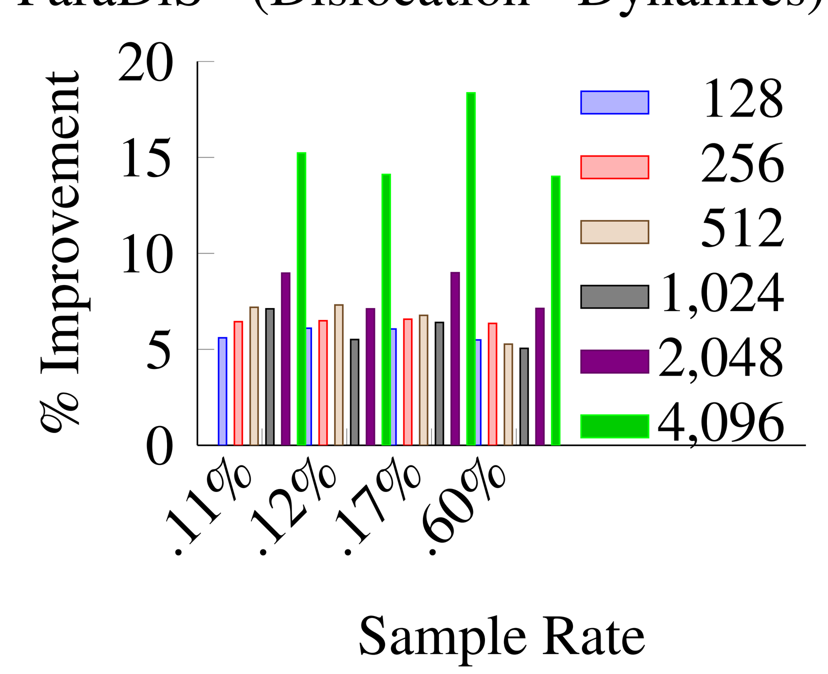


Our Method Uses Aggressive Adaptive Sampling to Define Uniform Tasks in Highly Non-Uniform Density



(a) Centroid *C* of Particles A, B, D (b) Sampled Interactions Define Voronoi Cells (c) Multi-Layer Voronoi Cells

New Balancing Method for N-Body Methods
- Traditional algorithms assign bodies not interactions
- **We developed an accurate and fast method to balance interactions between particles**
- Aggressive adaptive sampling reduces the number of tasks to balance and variance of task sizes
- 6-18% improvement in runtime of ParaDiS as compared to the built-in load balance algorithm

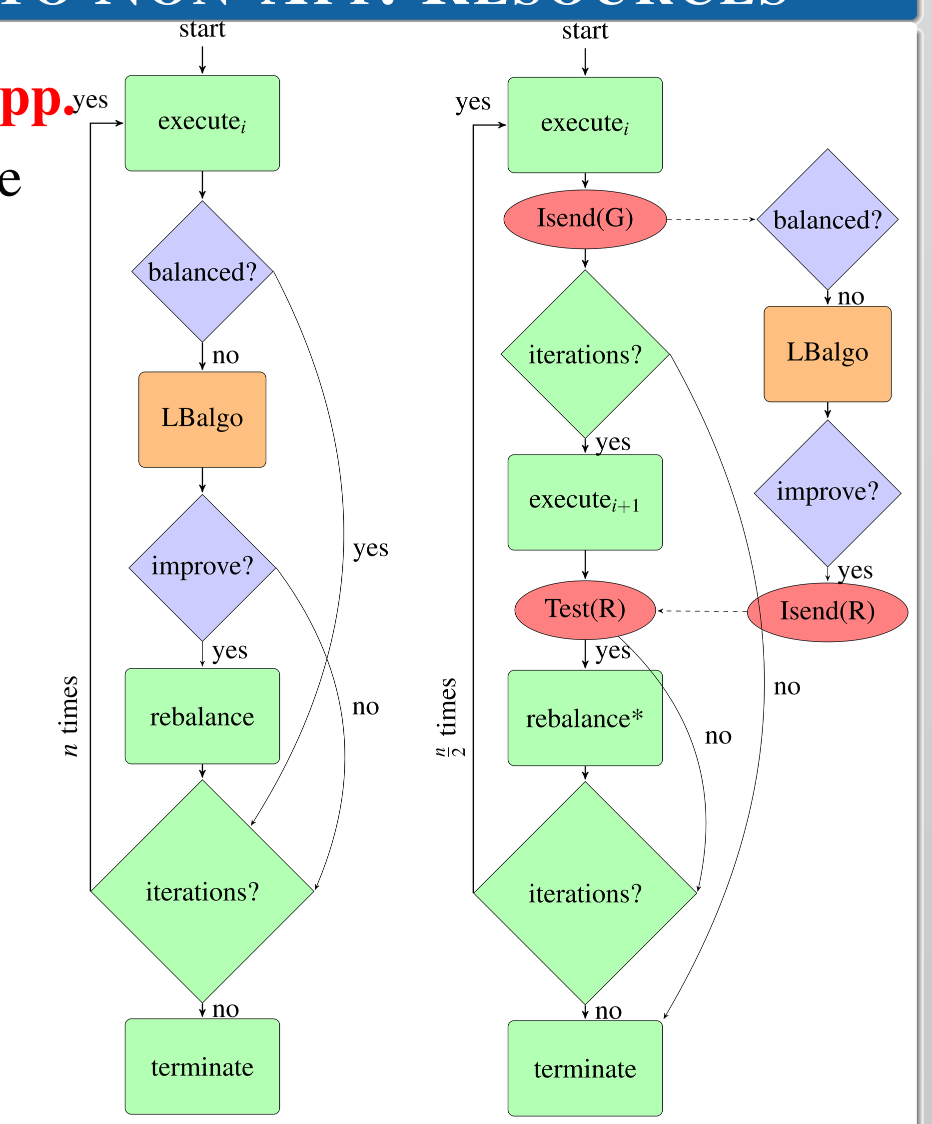


- More improvement on higher process count due to our method's precision

Load Balancing Simulations with Highly Non-Uniform Density, Olga Pearce, Todd Gamblin, Bronis de Supinski, Tom Arsenlis, Nancy M. Amato, *In Proc. ACM Intl.Conf.Supercomputing (ICS)*, June 2014.

4. Lazy Load Balancing: Offloading Load Balance Computation to Non-App. Resources

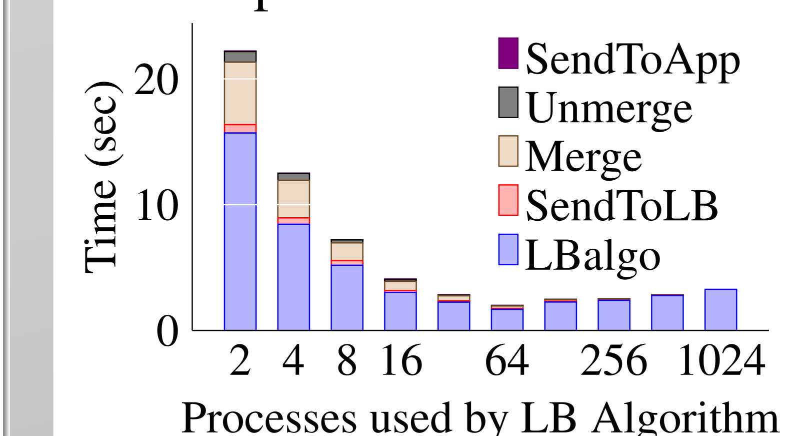
Balance Asynchronously to the App.
- Run load balancer on a separate set of nodes in parallel to application computation
- Evaluate and decide how to correct imbalance without pausing the application
- Overlap LB method with application computation
- Decouple LB method partition size from app. partition size
- Effects of delayed decision?
- Impact of application drift?
- Impact of system scale?

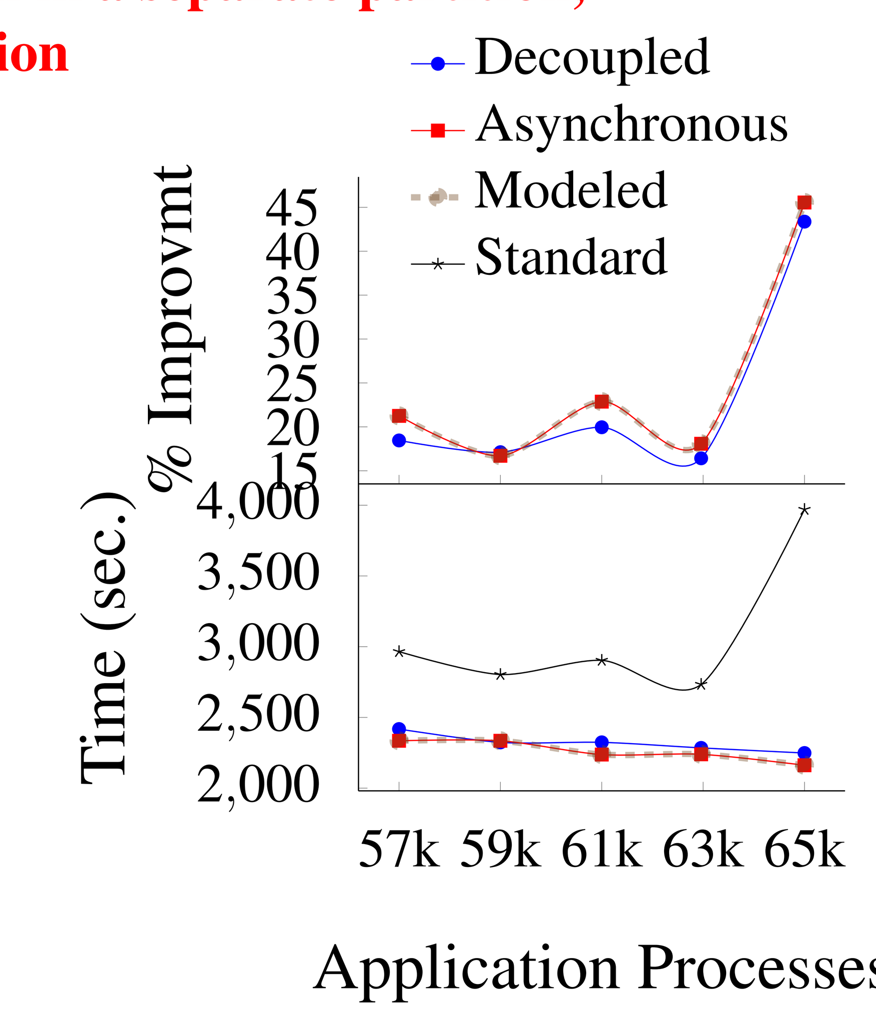


Application and LB Algo scale differently → decouple resources
- LB algorithm performance varies with resources (i.e., graph partitioner)
- Communication overhead depends on resources reserved for LB algorithm

Run the load balance algorithm in a separate partition, asynchronously to the application

- Decoupled LB Costs:





5. Contributions

- Lazy load balancing framework that decouples and offloads the load balance computation to make it affordable at scale
- A model for selecting the right load balance algorithm for the job
- An accurate and fast method to balance N-body applications with highly non-uniform density

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 LLNL-MI-657788
olga@cs.tamu.edu, http://www.parasol.tamu.edu/people/olga/