Optimal Distributed Load Balancing Algorithm for Homogeneous Work Units

Akhil Langer (4th year PhD Student) :: Laxmikant (Sanjay) Kale (Advisor)
Department of Computer Science, University of Illinois at Urbana-Champaign

Many parallel applications need dynamic load balancing during the course of their execution because of dynamic variation in the computational load. An ideal load balancer is one that can achieve perfect load balance, while doing minimal data migration (i.e., communication minimizing), is highly scalable to large number of processors and does not have large memory footprint. None of the existing load balancing algorithms - centralized, hybrid, and distributed algorithms - satisfy the criterion of an ideal load balancer. We propose a novel tree-based fully distributed algorithm that is an ideal load balancing algorithm when the work units are homogeneous. We evaluate its performance on Mira and Blue Waters and compare it with several existing load balancing strategies.

### MOTIVATION

<table>
<thead>
<tr>
<th>Centralized and Hierarchical Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Centralized</strong></td>
</tr>
<tr>
<td><strong>Hierarchical</strong></td>
</tr>
</tbody>
</table>

**Problem Statement**

Develop a scalable distributed load balancer that can distribute the work units equally amongst processors while minimizing data communication

**DISTRIBUTED ALGORITHMS**

<table>
<thead>
<tr>
<th>Diffusion</th>
<th>Gossip</th>
<th>Parallel Prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decisions based on localized load information e.g., near neighbor</td>
<td>Gossip protocol to propagate load information followed by probabilistic transfer of work units</td>
<td>Obtain task's global id using parallel prefix, assign to process Optimization: Subtract global minimum load</td>
</tr>
</tbody>
</table>

Disadvantages:
- Improve load balance rather than obtain global balance
- Multiple iterations to converge
- Slow convergence

Disadvantages:
- Multiple attempts to obtain target processor of a load
- User specified balance value
- No guarantees on load balance

Disadvantages:
- Perfect load balance
- Excessive data migration
- Excessive network power consumption, congestion

CONTRIBUTIONS

Communication minimizing optimal distributed load balancing algorithm:  
- Minimal migration of work units, 95% improvement over parallel prefix load balancer  
- Very small memory footprint - maximum list size of 13 at 128 processors  
- Fully distributed - oligo p^2 time complexity, assuming O(oligo) memory footprint  
- Significant reduction in network power consumption  
- Performance comparison with various other strategies  
- High Parallel Efficiencies achieved for strong scaling of 3D Adaptive Mesh Refinement

References


IBM BG/Q

PowerPC A2 1.6GHz

16 cores, 4 hardware threads per core

Cray XE6/XK7

AMD 6276 Interlagos 2.45 GHz

16 FPU/cor