A DISTRIBUTED DYNAMIC LOAD BALANCER FOR ITERATIVE APPLICATIONS

Harshitha Menon (gplkrsh2@illinois.edu)
Laxmikant Kale (kale@illinois.edu)
University of Illinois at Urbana Champaign
MOTIVATION

PROPOSED WORK

EVALUATION
MOTIVATION

Load Imbalance
Dynamic Load Imbalance
DYNAMIC LOAD BALANCER SHOULD...

Perform good load balance

Incur minimum overhead

Be profitable!
LOAD BALANCERS

Centralized

Distributed

Hierarchical
CENTRALIZED LB

Global view of the system

Bottleneck
DISTRIBUTED LB

Local view of the system

Scalable

Poor load balance
HIERARCHICAL LB

Subgroup of processors

Decisions at the root

Scalable
MOTIVATION

Centralized LB

- Global view
- Bottleneck
- Good Load balance

Distributed LB

- Limited view
- Scalable
- Poor Load balance
GRAPEVINE-LB
GRAPEVINE-LB

- Fully distributed
- Partial information about global state
- Scalable and good quality
1. Information Propagation

2. Load Transfer
INFORMATION PROPAGATION

Based on gossip protocol

Underloaded processors start gossip

Randomly sample peers (Fanout)
INFORMATION PROPAGATION

On receiving load information

- Updates its knowledge
- Forwards to random peers

No explicit synchronization

- TTL (Time To Live)
8 Processors
Fanout 2
TTL 3
\( r = O(\log_f n) \)

Number of rounds taken to propagate single update

\[ \begin{array}{c|c|c|c}
\text{System Size (n)} & \text{f=2} & \text{f=3} & \text{f=4} \\
\hline
0 & 0 & 0 & 0 \\
4096 & 4 & 6 & 8 \\
8192 & 8 & 10 & 12 \\
12288 & 12 & 14 & 16 \\
16384 & 16 & 18 & 20 \\
\end{array} \]
INFORMATION PROPAGATION

Naïve
- Random selection

Informed
- Biased selection
- Incorporate current knowledge
LOAD TRANSFER

Distributed

Naïve transfer

• Select processors uniformly at random
• Transfer load until load below threshold

Informed transfer

• Select processors based on load

\[ p_i = \frac{1}{z} \times \left( 1 - \frac{L_i}{L_{avg}} \right) \]
Naïve transfer

Initial load

Probabilities assigned

Work transferred

Final load

Informed transfer

Initial load

Probabilities assigned

Work transferred

Final load
QUALITY OF LOAD BALANCE

Partial information sufficient

Tunable using TTL
TUNABLE PARAMETERS

TTL (Time To Live)

Fanout

Imbalance threshold
BACKGROUND

Application over-decomposed

Load balancer invoked periodically

Using Charm++ load balancing framework

Load balancing framework

- Instruments
- Collects statistics
- Migrates objects
EVALUATION
Applications

LeanMD  (Strong scaling)

Adaptive Mesh Refinement  (Strong scaling)

Machine:

IBM BG/Q, Mira

Comparison

GreedyLB, AmrLB, HierarchicalLB, DiffusionLB
METRICS

Time per step excluding LB time

Load balancing overhead

Total application time
LEANMD: TIME PER STEP

GrapevineLB - good quality
LEANMD: LB OVERHEAD

Centralized LB
- high overhead

Distributed LBs
- low overhead
LEANMD: TOTAL TIME

Centralized-overhead exceeds benefit

GrapevineLB gives best performance
AMR: TIME PER STEP

- NoLB
- HierarchicalLB
- DiffusionLB
- GrapevineLB

GrapevineLB - good quality
AMR - LB OVERHEAD

Centralized LB
-high overhead

Distributed LBs
-low overhead
AMR- TOTAL TIME

Centralized, Hierarchical-overhead exceeds benefit

DiffusionLB-marginal benefit

GrapevineLB-best performance
SUMMARY

Simple strategy

Good quality with less overhead

Tunable
ACKNOWLEDGEMENTS

PPL group

Funded by:
DOE grant DOE DE-SC0001845 and by NSF ITR -HECURA-0833188
THANK YOU!
# LEANMD: TOTAL TIME

<table>
<thead>
<tr>
<th>Method</th>
<th>4096</th>
<th>8192</th>
<th>16384</th>
<th>32768</th>
<th>65536</th>
<th>131072</th>
</tr>
</thead>
<tbody>
<tr>
<td>NoLB</td>
<td>519.19</td>
<td>263.30</td>
<td>131.56</td>
<td>67.18</td>
<td>41.49</td>
<td>27.20</td>
</tr>
<tr>
<td>DiffuseLB</td>
<td>342.15</td>
<td>170.41</td>
<td>99.67</td>
<td>58.47</td>
<td>34.91</td>
<td>24.29</td>
</tr>
<tr>
<td>GreedyLB</td>
<td>336.34</td>
<td>184.09</td>
<td>112.23</td>
<td>90.19</td>
<td>99.51</td>
<td>105.35</td>
</tr>
<tr>
<td>HierarchLB</td>
<td>325.00</td>
<td>163.65</td>
<td>84.62</td>
<td>44.56</td>
<td>33.49</td>
<td>22.43</td>
</tr>
<tr>
<td>GrapevineLB</td>
<td>305.20</td>
<td>152.21</td>
<td>79.94</td>
<td>43.88</td>
<td>31.3</td>
<td>21.53</td>
</tr>
</tbody>
</table>
QUALITY OF LOAD BALANCE

- Quality metric

\[ I = \frac{L_{\text{max}}}{L_{\text{avg}}} - 1 \]
LEANMD-TIME PER STEP

![Graph showing time per step for different number of cores with various load balancing strategies.](image-url)
Load Balancing Evaluation with LeanMD

Load balancing overhead

Centralized have high overhead
Distributed schemes have low overhead

<table>
<thead>
<tr>
<th>Strategies</th>
<th>4K</th>
<th>8K</th>
<th>16K</th>
<th>32K</th>
<th>64K</th>
<th>131K</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hierc</td>
<td>3.721</td>
<td>1.804</td>
<td>0.912</td>
<td>0.494</td>
<td>0.242</td>
<td>0.262</td>
</tr>
<tr>
<td>Diff</td>
<td>0.080</td>
<td>0.057</td>
<td>0.051</td>
<td>0.035</td>
<td>0.027</td>
<td>0.018</td>
</tr>
<tr>
<td>Gv</td>
<td>0.017</td>
<td>0.013</td>
<td>0.014</td>
<td>0.016</td>
<td>0.015</td>
<td>0.018</td>
</tr>
<tr>
<td>Gv+</td>
<td>0.017</td>
<td>0.013</td>
<td>0.013</td>
<td>0.015</td>
<td>0.015</td>
<td>0.018</td>
</tr>
</tbody>
</table>

Load balancing cost (in seconds) of various strategies for LeanMD