Multi-level, Multi-core Distributed Trace Synchronization

Masoume Jabbarifar
Masoume.jabbarifar@polymtl.ca

Supervisor: Michel Dagenais

DORSAL
11 May 2011
Outline

- Optimization on Offline Synchronization
  - Convex-Hull
  - Architecture
  - Results
- Online Synchronization
  - Interval based Aposteriori Synchronization
  - Sliding Window based Synchronization
  - Incremental Online Synchronization
- Conclusion
- References
Synchronization Algorithm

Convex-Hull

1) Sent and Received sets
   - Guarantee no message inversion
2) Two lines with Max & Min slop
   \[ L_{\text{max}}(t_A) = a_{1}\text{\scriptsize{\text{max}}} t_A + b_{0}\text{\scriptsize{\text{min}}} \]
   \[ L_{\text{min}}(t_A) = a_{1}\text{\scriptsize{\text{min}}} t_A + b_{0}\text{\scriptsize{\text{max}}} \]
   Accuracy = \( a_{1}\text{\scriptsize{\text{max}}} - a_{1}\text{\scriptsize{\text{min}}} \)
3) The bisector of the angle formed by these two lines
Architecture

Multi-core Multi-level Distributed Trace Synchronization
Network Features

1) Physical distance
2) Quality of network path
3) Network latency
4) Delays
5) Hop count
6) Network traffic
Multi-core Multi-level Distributed Trace Synchronization
Result of NS2 (1/2)

Network Traffic (Packet Count)

Total Time ($)
### Result of NS2 (2/2)

<table>
<thead>
<tr>
<th>No. of Nodes</th>
<th>Total No. of Packets</th>
<th>Previous Sync. Time</th>
<th>Optimized Sync. Time</th>
<th>Saved Time (s)</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1437</td>
<td>8.67</td>
<td>6.04</td>
<td>2.5</td>
<td>30 %</td>
</tr>
<tr>
<td>5</td>
<td>2098</td>
<td>13.39</td>
<td>7.94</td>
<td>5.5</td>
<td>40 %</td>
</tr>
<tr>
<td>6</td>
<td>13044</td>
<td>79.60</td>
<td>69.06</td>
<td>10.5</td>
<td>13 %</td>
</tr>
<tr>
<td>16</td>
<td>209070</td>
<td>1127.28</td>
<td>765.22</td>
<td>362.06</td>
<td>32 %</td>
</tr>
<tr>
<td>19</td>
<td>141123</td>
<td>882.43</td>
<td>588.89</td>
<td>293.54</td>
<td>33 %</td>
</tr>
<tr>
<td>21</td>
<td>173985</td>
<td>1157.051</td>
<td>921.3</td>
<td>235.75</td>
<td>20 %</td>
</tr>
</tbody>
</table>

**Average**  
28 %
Outline

- Optimization on Offline Synchronization
  - Convex-Hull
  - Architecture
  - Results
- Online Synchronization
  - Interval based Aposteriori Synchronization
  - Sliding Window based Synchronization
  - Incremental Online Synchronization
- Conclusion
- Reference
Interval based Aposteriori Synchronization

- Incremental interval
- Save and reuse previous points
- Analysis on the whole data from the start point of tracing
- No need to repeat processing and matching of packets
Interval based Aposteriori Synchronization

• The advantage:
  • The highest level of accuracy

• The disadvantages:
  • Scalability

• Optimization:
  • Consider particular no. of previous intervals (e.g. 5 intervals)
LTTV Integration Challenges

- Add/remove new/old node at the entrance/leave time
- Gather trace files
- Synchronization delays (Network, Algorithm)
- Buffering
Sliding Window based Synchronization

- \( L \): size of window (static interval)

- Accurate packet is replaced as soon as detected
Sliding Window based Synchronization

- The advantages:
  - Guarantee high accuracy all the time
  - Improve accuracy over time
  - No buffering
Incremental Online Synchronization

• Self-Managing Method
• Optimize performance of the synchronization
• Dynamic window size
Multi-core Multi-level Distributed Trace Synchronization

Architecture

Linux Trace Toolkit Viewer (Online Sync. Part)

Gathering
- Sliding Window Manager
- Gather Stream Traces

Synchronization
- Measure Accuracy
- Synchronize

Management
- Matching Module
- Extract Dynamic Network Topology

Feedback (window size)

1
2
n
Gathering Component

• **Sliding window manager:**
  - Wide window = high accuracy & time consuming
  - Narrow window = performance & low accuracy
  - Network situation = dynamic/stable

• **Gathering Stream Traces:**
  - Network traffic load
  - Streaming latency
Management Component

• Match Packets:
  – Manage unknown received packets

• Extract Dynamic Network:
  – Extract set of nodes:
    • Communications
  – Select neighbors one by one randomly
  – Feedback network changes
Synchronization Component

• **Measure Accuracy:**
  – Balance speed and accuracy
  – Change window size

• **Synchronize**
  – Synchronize all nodes
Conclusion and Future Work

- Accurate online synchronization
- Scalable online synchronization
- Incremental online synchronization for large-scale dynamic systems
References (1)


References (2)


References (3)


References (4)


References (5)


[52] http://www.igi-global.com/bookstore/titledetails.aspx?TitleId=1123