ADAPTIVE TECHNIQUES FOR SCALABLE OPTIMISTIC PARALLEL DISCRETE EVENT SIMULATION

Eric Mikida
Presentation Overview

• PDES / GVT Overview
• GVT Framework Description
• New GVT Algorithm
• New Load Balancing Work
• Summary and Future Work
PDES / GVT Overview

• Simulation driven by discrete, time-stamped events
• Logical Processes (LPs) store state and execute events
• Charades is optimistically synchronized
  – Events executed speculatively
  – Incorrect events rolled back via reverse computation
  – Event efficiency = committed / total
• Global Virtual Time (GVT) required for synchronization
  – Virtual time passed by every processor and event in flight
GVT Framework Description

• Separated GVT Management from Scheduler
  – Each encapsulated into separate chare groups
  – Common API between base classes
• Allows for multiple different GVT implementations
• Work and communication automatically overlapped with Scheduler and LPs
GVT Framework Description

Scheduler

resume()
gvt_done(gvt)

GVTManager

gvt_begin()

Event Exec

gvt_begin()
resume()

gvt_done(gvt)

Event Exec

FC

GVT Work

GVT Work

...
New GVT Algorithm

- Adaptive Bucketed GVT algorithm
  - Virtual time divided into buckets
  - Completion detection per bucket
  - CD is timestamp aware
  - Buckets included in a given computation can increase/decrease based on simulation conditions
Adaptive Bucketed GVT Algorithm

Virtual Time

Sending an event (increment $s_4$)

Receiving an event (increment $r_6$)
Adaptive Bucketed GVT Algorithm

Formally, a bucket $b$ is completed iff:

1) $\text{sent}[b] = \text{recv}[b]$

2) $\text{lvt}_p > b \times \text{bucket\_size}$ for all processors $p$

3) bucket $x$ is complete for all $x$ in $\{1 \ldots b-1\}$
Adaptive Bucketed Performance

Speedup over Blocking

Speedup over Phase-Based
## Adaptive Bucketed Interval Analysis

<table>
<thead>
<tr>
<th>All-Reduces for Phase-Based</th>
<th>Total</th>
<th>Per GVT</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHOLD Base</td>
<td>3887</td>
<td>4.11</td>
</tr>
<tr>
<td>PHOLD Work</td>
<td>4270</td>
<td>4.31</td>
</tr>
<tr>
<td>PHOLD Event</td>
<td>5553</td>
<td>4.28</td>
</tr>
<tr>
<td>PHOLD Combo</td>
<td>6890</td>
<td>4.33</td>
</tr>
<tr>
<td>Traffic Base</td>
<td>------</td>
<td>--------</td>
</tr>
<tr>
<td>Traffic Src</td>
<td>------</td>
<td>--------</td>
</tr>
<tr>
<td>Traffic Dest</td>
<td>------</td>
<td>--------</td>
</tr>
<tr>
<td>Traffic Route</td>
<td>------</td>
<td>--------</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>All-Reduces for Adaptive Bucketed</th>
<th>Total</th>
<th>Per GVT</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHOLD Base</td>
<td>2005</td>
<td>1.98</td>
</tr>
<tr>
<td>PHOLD Work</td>
<td>2024</td>
<td>1.98</td>
</tr>
<tr>
<td>PHOLD Event</td>
<td>2011</td>
<td>1.98</td>
</tr>
<tr>
<td>PHOLD Combo</td>
<td>2040</td>
<td>1.99</td>
</tr>
<tr>
<td>Traffic Base</td>
<td>1276</td>
<td>1.58</td>
</tr>
<tr>
<td>Traffic Src</td>
<td>1965</td>
<td>1.92</td>
</tr>
<tr>
<td>Traffic Dest</td>
<td>1350</td>
<td>1.57</td>
</tr>
<tr>
<td>Traffic Route</td>
<td>2027</td>
<td>1.99</td>
</tr>
</tbody>
</table>
Adaptive Event Throttling

• SPEEDES halted event sending to flush network for continuous GVT [1]
  – Execution was allowed to continue
• Anti events a source of significant overhead
• Adaptive Bucketed GVT monitors all event sends (both regular and anti events)
Adaptive Event Throttling

Approach

- Track events by offset from GVT (in buckets)
- Add tracing for off-line analysis
- Analyze cancellation frequency and lag
- Hold events based on offset

![PHOLD Combo Event Stats](image)
Adaptive Event Throttling

Model Event Rates

Event Rate (millions of events/s)

PHOLD Base
PHOLD Work
PHOLD Event
PHOLD Combo
Dragonfly Worst
Dragonfly Trans
Traffic Base
Traffic Src
Traffic Dest
Traffic Route

1.1x
1.2x
1.75x
1.2x

Cutoff (# of Buckets)
- No Cutoff
- 16
- 8
- 4
- 2

5/2/19
Adaptive Event Throttling

Dragonfly Remote Events

- **Cutoff (# of Buckets)**
  - No Cutoff
  - 16
  - 8
  - 4

Billions of Events Sent:
- Dragonfly Worst
- Dragonfly Trans

Traffic Remote Events

- **Cutoff (# of Buckets)**
  - No Cutoff
  - 16
  - 8
  - 4
  - 2

Billions of Events Sent:
- Traffic Base
- Traffic Src
- Traffic Dest
- Traffic Route

5/2/19
Adaptive Event Throttling

Dragonfly Event Efficiency

Traffic Event Efficiency

Cutoff (# of Buckets)
- No Cutoff
- 16
- 8
- 4

Event Efficiency

Event Efficiency
How does this differ from SPEEDES?

SPEEDES

• Throttling *required* for the GVT computation to complete

• Once throttling starts, *all events* are all held until next GVT cycle

CHARADES

• GVT computation runs regardless of messages in flight – throttling just to improve performance

• Choice to hold an event is per event – holding one does not preclude us from sending another
Load Balancing with Bucketed GVT

• Don’t want to stop the simulation
• No obvious synchronization points
  – GVTManager runs independently of Scheduler
• Exploit anytime migration in Charm++
• Throttling improves event efficiency to aid LB
Load Balancing with Bucketed GVT

Traffic Speedup

Speedup

Traffic Src  Traffic Dest  Traffic Route

CPU Time  Committed  Past  Executed  Current VT  Latest VT  Next VT  Active  W. Pending

5/2/19
Summary

• Proposed the Adaptive Bucketed GVT algorithm
  – Timestamp aware to adapt to simulation conditions
  – Less communication required
  – Allows for adaptive communication throttling
• Load balancing can improve event efficiency
  – Metric effectiveness depends on model
• Best performance comes with decoupled solution
  – GVT: sync cost, Throttling: event efficiency, LB: balance
Future Work

• On-line tuning for adaptive event throttling
• Lightweight graph partitioning strategies
• Vectors of load metrics
• ML for load metrics
THANK YOU!