Generating Synthetic Traffic for Heterogeneous Architectures

Mario Badr | Natalie Enright Jerger | Riken Gohil | Radhika Jagtap | Matteo Andreozzi
What is Synthetic Traffic?

Perform Once-Few Times

1. Workload
2. Trace

Perform Many Times

1. Results
2. gem5 Traffic Generator
3. Other Heterogeneous System(s)
4. Statistical Profile
5. Results

Heterogeneous System

Other Heterogeneous System(s)
What is Synthetic Traffic?

Perform Once-Few Times

Perform Many Times

Workload

Heterogeneous System

Trace

Statistical Profile

gem5 Traffic Generator

Other Heterogeneous System(s)

Results

Results

Goal: Synthetic Traffic Results \(\sim\) Original Workload Results
gem5 Contributions (WIP)

Perform Once-Few Times

- Workload
- Heterogeneous System
- Trace

Perform Many Times

- gem5 Traffic Generator
- Other Heterogeneous System(s)
- Statistical Profile

Results

Results
A Heterogeneous System

- Central Processing Unit (CPU)
- Graphics Processing Unit (GPU)
- Memory Controllers
- Video Processing Unit (VPU)
- Display Processing Unit (DPU)
- Other Components
- Cache Coherent Interconnect
Why Synthetic Traffic?

1. Distribute models, not traces
2. Modify the model for custom behaviour
3. Converge quickly to final result
1. What kind of message should be sent?

2. When should a message be sent?

3. How big is the message?

4. Where is the message going?

5. How does it change over time?
<table>
<thead>
<tr>
<th>Question</th>
<th>Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>What kind of message should be sent?</td>
<td>Read Percentage</td>
</tr>
<tr>
<td>When should a message be sent?</td>
<td>Average Delta Time</td>
</tr>
<tr>
<td>How big is the message?</td>
<td>Average Data Size</td>
</tr>
<tr>
<td>Where is the message going?</td>
<td>Address Synthesis</td>
</tr>
<tr>
<td>How does it change over time?</td>
<td>State Transitions</td>
</tr>
</tbody>
</table>

A Potential Statistical Profile
nextTick = nextPacketTick()
packet = getNextPacket()
Generating Time-Varying Traffic

- **Update Time**
  - **Next State?**
    - Yes: state = nextState
    - No: nextTick = state.nextPacketTick()
  - **Next Packet?**
    - Yes: packet = state.getNextPacket()
    - No: nextTick = state.nextPacketTick()
From Trace to Profile

Divide
Discretize trace into intervals

Group
Group similar intervals into a state

Model
Model each state

Transition
Model transitions between states
Dividing a Trace into Intervals

Total Messages Sent

Time (interval is 100,000 cycles)
Dividing a Trace into Intervals

GPU

- Reads: 100%
- Delta Time: 38 cycles
- Data Sent: 165,000 bytes

- Reads: 72%
- Delta Time: 27 cycles
- Data Sent: 233,000 bytes
Interval Replay Results

Write Bandwidth

Read Bandwidth

Interval Replay Results

Bandwidth
Grouping Similar Intervals

Total Messages Sent

Time (interval is 100,000 cycles)

3 Groups
The Representative Intervals

Medoids

Reads: 52%
Delta Time: 19 cycles
Data Sent: 330,000 bytes

Reads: 98%
Delta Time: 36 cycles
Data Sent: 170,000 bytes

Reads: 39%
Delta Time: 17 cycles
Data Sent: 375,000 bytes

Total Messages Sent

Time (interval is 100,000 cycles)
Replacing Observed Data with Medoid

Original Observed Data

Representative Replay
Grouped Replay Results

**Write Bandwidth**

<table>
<thead>
<tr>
<th></th>
<th>CPU</th>
<th>DPU</th>
<th>GPU</th>
<th>VPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>trace</td>
<td></td>
<td></td>
<td>1600</td>
<td></td>
</tr>
<tr>
<td>N intervals</td>
<td>500</td>
<td>1000</td>
<td>1500</td>
<td>2000</td>
</tr>
<tr>
<td>grouped replay</td>
<td>0</td>
<td>500</td>
<td>1500</td>
<td>2000</td>
</tr>
</tbody>
</table>

**Read Bandwidth**

<table>
<thead>
<tr>
<th></th>
<th>CPU</th>
<th>DPU</th>
<th>GPU</th>
<th>VPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>trace</td>
<td></td>
<td></td>
<td>2000</td>
<td></td>
</tr>
<tr>
<td>N intervals</td>
<td>1000</td>
<td>1500</td>
<td>2000</td>
<td>2500</td>
</tr>
<tr>
<td>grouped replay</td>
<td>0</td>
<td>1000</td>
<td>1500</td>
<td>2000</td>
</tr>
</tbody>
</table>
Observed Sequence of Medoids

Markov Chain

Steady State Markov Chain

```
\begin{bmatrix}
0.6 & 0.07 & 0.33 \\
0.6 & 0.2 & 0.2 \\
0.5 & 0.5 & 0.0 \\
\end{bmatrix}^{512} = \begin{bmatrix}
0.58 & 0.19 & 0.23 \\
0.58 & 0.19 & 0.23 \\
0.58 & 0.19 & 0.23 \\
\end{bmatrix}
```

Converging Early

Compare Simulated Distribution to Steady State

\[
\begin{bmatrix}
0.60 \\
0.16 \\
0.24 \\
\end{bmatrix} \approx \begin{bmatrix}
0.58 \\
0.19 \\
0.23 \\
\end{bmatrix}
\]
Time to Converge
Converging Results

Bandwidth

Total Bandwidth

MBytes Per Second

AFBC-linear-0  |  AFBC-tiled-0  |  coastguard  |  HEVC  |  opencl-1  |  trex-f17

trace  |  grouped replay  |  grouped markov
• What should the interval size be?
• How many groups should there be?
• How should addresses be synthesized?