JiST – Java in Simulation Time

Rimon Barr
**JiST – Java in Simulation Time**

- Transparent, efficient, scalable, optimistic execution of discrete event MANET simulations over a standard virtual machine
- discrete event simulations are useful and needed
- but, most published ad hoc network simulations
  - lack *scalability* ~250 nodes; or
  - compromise *detail* packet level; or
  - are short *duration* few minutes
JiST: existing alternatives

**ns2** is the *gold standard*
- Tcl-based, with C++ bindings
- used extensively within community
- written for detailed TCP simulation
- modified to support ad hoc networks
- processor and memory intensive
- sequential; max. ~250 nodes, O(n^3)

**PDNS** – parallel distributed ns2
- event loop uses RTI-KIT
- requires fast inter-connect
- helps with memory limits

**GloMoSim**
- Parsec-based, a custom C-like language
- entities are memory intensive
- requires “node aggregation,” which imposes conservative parallelism
- high event throughput per processor, but requires shared memory for scale
- shown ~10,000 nodes, but on NUMA (SPARCserver 1000, est. $300,000)

**custom-made** simulators
- fast, specialized computation
- lack sophisticated execution, parallelism
- lack *credibility*
The gist of JiST

- achieve **scalability** through
  - parallelism, optimism: maximize execution concurrency
  - state partitioning: simulation split into fine-grained entities
  - transparency: automated binary rewrite of serial programs
  - genericity: general-purpose, optimized systems language and runtime
  - COTS hardware: cluster of inexpensive PCs

Automatic simulation partitioning  Optimistic parallel execution
JiST: status

- the “hello world” of event simulations
  ```java
  class MySim implements JistAPI.Entity {
    public void myEvent() {
      JistAPI.sleep(1);
      myEvent();
      System.out.println("hello world, time=");
      System.out.println(JistAPI.getTime());
    }
  }
  ```
  
- currently building SWANS atop JiST
  - Scalable Wireless Ad hoc Network Simulator
  - Java application running in simulation time

<table>
<thead>
<tr>
<th># events</th>
<th>JIST</th>
<th>GloMoSim</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>10^5</td>
<td>0.044s</td>
<td>0.435s</td>
<td>10%</td>
</tr>
<tr>
<td>10^6</td>
<td>0.262s</td>
<td>2.938s</td>
<td>9%</td>
</tr>
<tr>
<td>10^7</td>
<td>2.301s</td>
<td>28.04s</td>
<td>8%</td>
</tr>
<tr>
<td>10^8</td>
<td>22.48s</td>
<td>278.4s</td>
<td>8%</td>
</tr>
</tbody>
</table>

Memory Limit

<table>
<thead>
<tr>
<th></th>
<th>Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>JiST</td>
<td>40 bytes (&gt;10^6) entities</td>
</tr>
<tr>
<td>Parsec</td>
<td>27800 bytes (~10^4) entities</td>
</tr>
</tbody>
</table>

JiST scales to more entities per process

serial throughput increase of 12x