Computational Challenges of Implementing the Atlanta Regional Commission Activity-Based Modeling System

May 11, 2010
Presentation Outline

- ARC Activity-Based Model Quick Introduction
- Implementation Design Goals
- Hardware and Software Setup
- Non-CT-RAMP Distribution and Threading
- CT-RAMP Distribution and Threading
- Overall System Setup
- Future Year Runs
- Next Steps
- Conclusions
ARC Activity-Based Modeling System

• Based on the CT-RAMP\(^1\) family of ABMs developed in New York, NY, Columbus OH (MORPC) and others
  - Explicit intra-household interactions
  - Continuous temporal dimension (Hourly time periods)
  - Integration of location, time-of-day, and mode choice models
  - Java-based package for AB model implementation

• Implemented with the existing Cube-based networks, GUI and ancillary models (external model, truck model, assignments, etc)

• Households: 1.7 million in 2005, 2.7 million in 2030

• Model development parallel effort with MTC

\(^1\)Coordinated Travel-Regional Activity-Based Modeling Platform
Project History

- 2003 → 2006
  - Models estimated, population synthesizer developed (as presented @ ITM 2006 in Austin TX)
- 2007 → 2008
  - Model implementation, calibration started
- 2009 → April 2010
  - Calibration/validation completed, documentation, deployment at ARC, and sensitivity testing
- Remainder 2010
  - Enhanced data reporting and visualization of outputs
Treatment of Space

- 2027 TAZs
- TAZs subdivided into transit accessibility:
  - Short walk (1/3 mi)
  - Long walk (2/3 mi)
  - No walk (> 2/3 mi)
- All origins and destinations identified by TAZ and sub-zone
- 6081 total alternatives in destination choice
Implementation Design Goals

• Overnight run time → Model Relevance
  • Around 16 hours
  • Requires distribution and threading

• Commodity hardware → Minimize total lifetime cost
  • Hardware available today from common vendors; reasonably priced

• Easy to Setup and Use → Staff acceptance
  • Not too complicated to setup, run, debug, etc
Hardware and Software Setup

- Three Windows Server 2003 64bit Machines:
  - Dual Quad Core Intel Xeon X5570 2.93 GHz with Hyper-Threading → 16 threads
  - 32 GB of RAM
  - Cube Voyager + 8 seat Cube Cluster license

- Total cost ~ $30,000 in 2009
Hardware and Software Setup

- 64 bit OS for large memory addresses
- 64 bit Java for CT-RAMP
- 32 bit Java to integrate with Cube’s native matrix I/O DLL
- Cube Base for the GUI
- Cube Voyager + Cluster for running the model, assignment, etc
- Java CT-RAMP software
- 64 bit R for reporting/visualization
User Interface
### CT-RAMP Model UEC Example

A row for each utility term

A column for each alternative (0, 1, 2, and 3+ autos)

<table>
<thead>
<tr>
<th>No</th>
<th>Token</th>
<th>Description</th>
<th>Filler</th>
<th>Formula for variable</th>
<th>Index</th>
<th>Alt1</th>
<th>Alt2</th>
<th>Alt3</th>
<th>Alt4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>Alternative-specific constant</td>
<td>1</td>
<td></td>
<td></td>
<td>-5.352</td>
<td>-2.132</td>
<td>0</td>
<td>-0.788</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>Household Size 1</td>
<td>i(@size==1,1,0)</td>
<td></td>
<td></td>
<td>2.613</td>
<td>2.172</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>Household Size 2</td>
<td>i(@size==2,1,0)</td>
<td></td>
<td></td>
<td>0.000</td>
<td>0.400</td>
<td>0</td>
<td>-0.673</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>Income Group 1</td>
<td>i(@income==1,1,0)</td>
<td></td>
<td></td>
<td>2.878</td>
<td>2.186</td>
<td>0</td>
<td>-1.265</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>Income Group 2</td>
<td>i(@income==2,1,0)</td>
<td></td>
<td></td>
<td>1.734</td>
<td>1.731</td>
<td>0</td>
<td>-1.061</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>Income Group 3</td>
<td>i(@income==3,1,0)</td>
<td></td>
<td></td>
<td>0.000</td>
<td>1.152</td>
<td>0</td>
<td>-1.025</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>Income Group 4</td>
<td>i(@income==4,1,0)</td>
<td></td>
<td></td>
<td>0.000</td>
<td>0.669</td>
<td>0</td>
<td>-0.535</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>Worker 0</td>
<td>i(@workers==0,1,0)</td>
<td></td>
<td></td>
<td>1.015</td>
<td>0.000</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>Worker 1</td>
<td>i(@workers==1,1,0)</td>
<td></td>
<td></td>
<td>0.000</td>
<td>0.000</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>Worker 2</td>
<td>i(@workers==2,1,0)</td>
<td></td>
<td></td>
<td>0.000</td>
<td>-0.534</td>
<td>0</td>
<td>0.648</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>Worker 3+</td>
<td>i(@workers==3,1,0)</td>
<td></td>
<td></td>
<td>2.195</td>
<td>0.000</td>
<td>0</td>
<td>2.257</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>GVSAD retirement zone</td>
<td>i(GV_SAD_IND==1,1,0)</td>
<td></td>
<td>z</td>
<td>0.000</td>
<td>1.200</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>HIRET retirement zone</td>
<td>i(HI RET_IND==1,1,0)</td>
<td></td>
<td>z</td>
<td>0.000</td>
<td>0.916</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>Tot emp w/ 20 min by transit, normalized</td>
<td>i(trn20m_emp)</td>
<td></td>
<td>z</td>
<td>0.014</td>
<td>0.000</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>Percent of TAZ w/ 1/3 mile of transit stop</td>
<td>shortWalk</td>
<td></td>
<td>z</td>
<td>0.021</td>
<td>0.010</td>
<td>0</td>
<td>0.000</td>
</tr>
</tbody>
</table>

A description for the term

A formula field for computing data items

Coefficients for each term and alternative

---

[Sources: PB, PBSJ, ATLANTA REGIONAL COMMISSION]
Distributing and Threading Non-CT-RAMP

• ARC ABM started with the ARC trip-based model
• Replaced Internal-Internal (II) with CT-RAMP
• Other models in Cube Voyager
  • Network processing
  • Commercial vehicle model
  • Airport model
  • External model
  • Non-II time-of-day model
  • Transit network building
  • Highway and transit assignment
  • Skimming
• Needed to distribute and thread Cube components
**Distributing and Threading in Cube Cluster**

- **DistributeINTRASTep**
  - Multithreading by origin zone
  - Flexible configuration
  - Highway assignment → AM (x4), MD, PM, NT
  - Matrix processing → Creating time-of-day matrices, etc
  - SPEED UP: ~4X

- **DistributeMULTIStep**
  - Distribute programs across processors and wait for completion
  - Requires explicit assignment of tasks to processes
  - Highway assignment by Time-Of-Day
  - Creating assignment matrices by Time-Of-Day
  - Transit assignments by Time-Of-Day
  - SPEED UP: ~3X
Distributing and Threading CT-RAMP

- Decompose computations by Households
  - Distribute by groups of households (2000 at a time)
- Implement Java Parallel Processing Framework (JPPF)
  - Open source library to run and manage the distribution of parallel tasks
  - Most computation done on the worker nodes
Distributing and Threading CT-RAMP

- Main Cube model script calls the JPPF client to start CT-RAMP
- ~1.76 million households split into 880 tasks of 2000 HHs
- CT-RAMP data managed through:
  - Household Manager – manages all HH and person data into RAM for quick I/O
  - Matrix Manager – reads all the matrix data into RAM for quick I/O
- Run a sample of HHs to save time: 33% → 50% → 100%
- HHs and Persons store a random number seed to avoid random number sequence order of processing problems
- SPEED UP: 9X
Overall System Setup

- Cube runs the show and calls all Java processes
- User starts the remote processes on the 2nd and 3rd machine (for now)
- Everything talks to one mapped network folder location
Future Year Runs

• 2.6 million households in 2030
• Significant increase in congestion
• Initial memory “leak” in task → distribution found and fixed that wasn’t a problem in the base year
• Approximately 20 - 21 hour run times depending on the scenario
Next Steps

• Improved run times testing
  • Benefit of adding an additional computer
  • Reconfiguring the system setup (moving the matrix manager; adding a 3rd worker node, etc)
  • Additional optimization of CT-RAMP code

• System Setup
  • Automatically start and stop remote processes (with JPPF or WMI)

• Model reporting and visualization tools
  • Some preliminary visualization done
  • Develop dynamic visualization tools (animation, etc) with Adobe Flex/Flash and other tools

• Transit on-board survey analysis and mode choice re-calibration
• Utilize NAVTEQ-conflate highway networks and ARC’s new 2010 20-county 6,000+ TAZ system
• Longer term: Potential integration with PECAS and TRANSIMS
Tracing of Activities/Tours

Person id= 1018897 type= Full-time worker
Conclusions

• Significant run time improvements through distribution and threading (16 hours versus 146 hours)
• Leverage commodity hardware and software (and open source software)
• Scalable architecture – can add more hardware and will get even faster!
• Economies of scale for ARC & MTC in model co-development, a +/- 20% cost saving for ARC & MTC
• Model relevance with reasonable run times
Questions and Discussion

Ben Stabler, PB, Stabler@pbworld.com

Jim Hicks, PB, HicksJi@pbworld.com

Joel Freedman, PB, Freedman@pbworkld.com

Guy Rousseau, ARC, GRousseau@atlantaregional.com

Kyeil Kim, ARC, kykim@atlantaregional.com

Jonathan Nicholson, PBS&J, JLNicholson@pbsj.com

Chris Simons, PBS&J, CNSimons@pbsj.com

Chuck Purvis, MTC, CPurvis@mtc.ca.gov

David Ory, MTC, DOry@mtc.ca.gov