A distributed Optimization Bot/Agent Application Framework for GAMS Models

Franz Nelißen – FNelissen@gams.com
Implementation of Prototype by Girish Garg
Agenda

Introduction

Model Building and Application Building

A Prototype

Summary and Outlook
Company

- Went commercial in 1987
- GAMS Development Corp. (USA), GAMS Software GmbH (Germany)
- 2016: New management team
- Software Tool Provider
GAMS at a Glance

- Pioneered Algebraic Modeling Languages

- Robust, scalable state-of-the-art algebraic modeling technology for complex, large-scale optimization

- Open architecture and uniform interface to all major commercial and academic solvers (30+ integrated)

- Evolution through more than 25 years of R&D and user feedback, maturity through experience and rigorous testing
Change in Focus: "Ages Ago"

Computation

- **Constraints:**
  - Algorithms
  - Hardware
  - Modeling Technology

- **User:** Left out
Progress in LP: 1988—2004


- **Algorithms** (*machine independent*):
  - Primal *versus* best of Primal/Dual/Barrier: 3,300x
- **Machines** (*workstations* → *PCs*): 1,600x
- **NET**: Algorithm × Machine: 5,300,000x
  
  (2 months/5300000 ~ = 1 second)
Change in Focus: Past $\rightarrow$ Now (2)

**Modeling Technology**

- **Algebraic Modeling Languages (AML)**
- **Simplified model development & maintenance**
- **Increased productivity tremendously**
- **Made mathematical optimization available to a broader audience (domain experts)**
- **2012 INFORMS Impact Prize**
- **Constraint: Modeling Skills**

- **User: Involved**
Change in Focus: **Now → Future**

**Application Building**
- Models (small) Part of Applications
- New Constraint: Integration Skills
- User: Hardly aware of the model
Modeler’s Perspective

- Problem class
- Algorithm / Algebra
- Data
- Solver
- Solution
Application Developer’s Perspective

- IT (Software) Driven
- Software Architecture, OO-Design, Agile Development, Tiers, Components, Encapsulation, Classes, Data Access Layer,

- Mathematical Optimization: (Maybe) Limited Knowledge / Interest
Different Communities

Analytic Professionals

- INFORMS 2017: 12,500+ members (GOR: 1,350)
- Niche Market: Mathematical Programming
  - Few standards, slow progress (in certain areas)
  - Active Academic Research Area
- Certification: Certified Analytics Professional (CAP) introduced in 2013
## Different Communities

### Software Developers

- 11 Million+ Software Developers worldwide (2014, IDC)
- Many and rapidly changing IT environments (Web, Cloud, Mobile, ...)
- Certifications for IT professionals:
  - Essential
  - Plenty

---

**The 7 Highest-Paying IT Certifications of 2017**

*BY JUAN MARTINEZ  7 APR 2017, 8:05 P.M.*

Show your mettle by getting one of these top-paying IT certifications.

1.2K shares
Possible Issues

- Different Lifecycles for Applications and Models
- "Optimization takes longer than one is willing to wait, and will eventually fail"
- Quality of Data
- Generalist (domain expert and modeling expert and IT expert) versus Team of Specialists
- “New” Application or Integration
Application Building for AML

Some Approaches

Integration with Analytical Software

- “Top-Down” - Add AML to existing analytical software systems with “large” user base, e.g. MATLAB or SAS

- “Bottom-Up” - Add GUI-builder / Application Framework to AML with “small” user base, e.g.: AIMMS (Pro) or FICO Xpress-Insight
Application Building for AML

Some Approaches

Integration with Programming Language

- “Top-Down” - Extend existing programming language with declarative AML, e.g.: Pyomo (Python), JuMP (Julia), MS Solver Foundation (discontinued)

- “Bottom-Up” - Make it easy to embed GAMS into different (programming) environments
Seamless Integration

Separation of Tasks

- Use GAMS for modeling and optimization tasks
- Object oriented GAMS API connects GAMS to other environments to build Applications
  - Programming languages
  - Smart Links to Databases, Spreadsheets, Matlab, R,...
- .Net, Java, Python, C++ (open source)
- Communication through Memory or Files
OO-API: Encapsulation of GAMS Model

Simple Interface to interact with GAMS

- Classes to communicate input data and results
- Classes to change options like the solver to use
- Classes to create, run, and control model instance(s)

Main Application (.Net, Java, Python, C++)

Wrapper Classes

GAMS Model

Data / Results
Options
Create / Run
Task At Hand

Develop Prototype of a distributed Multi-Tier Application with a Multi-User Web Interface

- Application connects GAMS Model to Databases and Web User Interface
- Bot/Agents run Model instances
- User Interface allows
  - Setup and submission of (multiple) GAMS jobs
  - Visualization of results
- Communication with GAMS through OO-API only
- .Net Application
- Application Developer has no knowledge about GAMS
- Tight Budget for Application Development
Architectural Overview

- Corporate Database - Tier 2 -
- Web UI - Tier 1 -
- Task Bot - Tier 3 -
- Task Queue - Tier 1 -
Technical Overview

- Corporate Database (MS SQL Server)
- Task Bot (Win Form)
- Web Services (Async.asmx)
- Application Database (MS SQL Server)
- Web Form (New Task.aspx)
- Web Form (Result View.aspx)
- Web Form (New Task.aspx)
- Task Queue.asmx
Some Features

- Asynchronous architecture: Task submission and execution are decoupled through the Task Queue
- Data Contract between Bots and Application: Common Data Structures for clear interface
- Distributed system (multiple tiers): WebUI, Databases and Bots can run on separate machines
- Scalable with multiple bot instances
Example: Roll Cutting (cutstock.gms)

Cut paper rolls of fixed width ("raw") into smaller portions ("finals")

- **Input:**
  - Width of the raw
  - Demand: Widths and number of finals

- **Objective:** Minimize the required number of raws

- **Output:**
  - Combination and number of cuts ("patterns")
  - Number of required raws
Web User Interface
Create **New Cutstock Task**

### New Cutstock Task

<table>
<thead>
<tr>
<th>ID</th>
<th>Width</th>
<th>Demand</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>i1</td>
<td>47</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>i2</td>
<td>36</td>
<td>610</td>
<td></td>
</tr>
<tr>
<td>i3</td>
<td>31</td>
<td>395</td>
<td></td>
</tr>
<tr>
<td>i4</td>
<td>14</td>
<td>211</td>
<td></td>
</tr>
</tbody>
</table>

**Raw Width:**

- ID: 100
- Max Pattern: 35

[Generate Task]
## Task List (= Queue) and Status

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Status</th>
<th>Issued On</th>
<th>Allotted On</th>
<th>Completed On</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>cutstock</td>
<td>task7</td>
<td>available</td>
<td>19-Jul-17 07:59:55</td>
<td>NA</td>
<td>NA</td>
<td>Status Checker</td>
</tr>
<tr>
<td>cutstock</td>
<td>task3</td>
<td>completed</td>
<td>18-Jul-17 02:14:14</td>
<td>18-Jul-17 02:14:22</td>
<td>18-Jul-17 02:14:24</td>
<td>View Result</td>
</tr>
<tr>
<td>cutstock</td>
<td>task2</td>
<td>completed</td>
<td>18-Jul-17 02:13:58</td>
<td>18-Jul-17 02:14:02</td>
<td>18-Jul-17 02:14:03</td>
<td>View Result</td>
</tr>
<tr>
<td>cutstock</td>
<td>task1</td>
<td>completed</td>
<td>18-Jul-17 02:13:45</td>
<td>18-Jul-17 02:13:51</td>
<td>18-Jul-17 02:13:52</td>
<td>View Result</td>
</tr>
<tr>
<td>cutstock</td>
<td>task4</td>
<td>completed</td>
<td>18-Jul-17 02:12:04</td>
<td>18-Jul-17 02:12:10</td>
<td>18-Jul-17 02:12:12</td>
<td>View Result</td>
</tr>
<tr>
<td>cutstock</td>
<td>task3</td>
<td>completed</td>
<td>18-Jul-17 02:09:41</td>
<td>18-Jul-17 02:10:19</td>
<td>18-Jul-17 02:10:21</td>
<td>View Result</td>
</tr>
<tr>
<td>cutstock</td>
<td>task2</td>
<td>completed</td>
<td>18-Jul-17 02:09:28</td>
<td>18-Jul-17 02:10:08</td>
<td>18-Jul-17 02:10:10</td>
<td>View Result</td>
</tr>
<tr>
<td>cutstock</td>
<td>task1</td>
<td>completed</td>
<td>18-Jul-17 02:09:16</td>
<td>18-Jul-17 02:10:07</td>
<td>18-Jul-17 02:10:09</td>
<td>View Result</td>
</tr>
</tbody>
</table>
## Task Results Page (1)

### Task Results (task3)

<table>
<thead>
<tr>
<th>Name</th>
<th>task3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>cutstock</td>
</tr>
<tr>
<td>Status</td>
<td>completed</td>
</tr>
<tr>
<td>Done By</td>
<td>CutstockBot01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IssuedOn</th>
<th>18-Jul-17 02:14:14</th>
</tr>
</thead>
<tbody>
<tr>
<td>AllottedOn</td>
<td>18-Jul-17 02:14:22</td>
</tr>
<tr>
<td>CompletedOn</td>
<td>18-Jul-17 02:14:24</td>
</tr>
</tbody>
</table>

### Input

<table>
<thead>
<tr>
<th>RawWidth</th>
<th>MaxPattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>125</td>
<td>35</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Piece Name</th>
<th>Width</th>
<th>Demand</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>i1</td>
<td>47</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>i2</td>
<td>36</td>
<td>610</td>
<td></td>
</tr>
<tr>
<td>i3</td>
<td>31</td>
<td>395</td>
<td></td>
</tr>
<tr>
<td>i4</td>
<td>14</td>
<td>211</td>
<td></td>
</tr>
</tbody>
</table>
### Output

- **pattern 5**: 87 times
  - 13, 13, 13, 13, 13, 13, E

- **pattern 6**: 203 times
  - 12, 12, 12, 12, 14, E

- **pattern 9**: 47 times
  - 11, 11, 13, E

- **pattern 11**: 3 times
  - 11, 12, 14, 14, 14, E

### Log

- New pattern Value: -3
- New pattern Value: -2.333333333333333
- New pattern Value: 1.666666666666667
- New pattern Value: -1.25
- New pattern Value: -0.25
- New pattern Value: -0.041666666666667
- New pattern Value: -0.0220588235294117
- Optimal Solution: 340
- pattern 5: 87 times: 13: 4
- pattern 6: 203 times: 12: 3 14: 1
- pattern 9: 47 times: 11: 2 13: 1
- pattern 11: 3 times: 11: 1 12: 1 14: 3
Cutstock Bots Log

Log Text:
- 08:35:53: Processing Complete
- 08:35:52: Task Received: task3, Probing Stopped.
- 08:35:42: Bot Started
- 08:35:15: Bot Stopped
- 08:35:11: Processing Complete
- 08:35:09: Task Received: task1, Probing Stopped.
- 08:34:59: No Tasks Available, Probing will continue.
- 08:34:49: No Tasks Available, Probing will continue.
- 08:34:39: No Tasks Available, Probing will continue.
- 08:34:20: No Tasks Available, Probing will continue.
Summary and Outlook

- Building Optimization Applications may be challenging
- GAMS has no preference for a specific User Interface
- OO-API makes it easy to embed GAMS models

- Optimization Bot/Agent Application Framework
  - Integrates Web UI, GAMS, task bots and queues, and databases
  - Distributed system / multiple bot instances
  - Prototype done in .Net using the .Net OO-API
  - Source code for prototype will become open source
Thank You