Rapid Application Development & Grid Computing Using GAMS

Software Demonstration
INFORMS San Francisco 2005
Demo Objectives

- Basics / Algebraic Modeling
- Data Exchange / Charting
- Tying things together (procedural elements)
- Grid Computing
- Model Deployment
Company Profile

- **General Algebraic Modeling System**
- Started as a *research project* at the World Bank 1976
- Went *commercial* in 1987
- **Offices** in Washington, D.C and Cologne
- Professional *software tool provider*
- Operating in a *segmented niche market*
- Broad *academic* and *commercial* user base
Modeling Systems

- Describe problems to a computer system in the same way that people describe problems to each other.
- Simplify the model building and solution process
- Create maintainable models
- Adapt models quickly to new situations
Basic Technical Principles

- Separation of model and data
- Separation of model and solution methods
- Computing platform independence
- Multiple model types, solvers, platforms
- Balanced mix of declarative and procedural approaches
- Model is a data base operator and/or object
Basic Technical Principles II

- Open architecture and interfaces to other systems:
  - GUI
  - Excel, Databases
  - Programming Languages etc.

- Maintainable models and protection of investments
Multiple Model Types

- LP - Linear Programming
- MIP - Mixed Integer Programming
- QCP - Quadratically Constrained Programming
- NLP - Nonlinear Programming
- CNS – Constrained Nonlinear Systems
- MINLP - Mixed Integer Nonlinear Programming
- MPEC - NLP with Complementarity Constraints
- MPSGE - General Equilibrium Models
- Stochastic Optimization
<table>
<thead>
<tr>
<th>Solver/Platform availability - 22.0 August 1, 2005</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel MS Windows</td>
</tr>
<tr>
<td>-------------------</td>
</tr>
<tr>
<td>BARON 7.4</td>
</tr>
<tr>
<td>BDMLP</td>
</tr>
<tr>
<td>COIN</td>
</tr>
<tr>
<td>CONOPT 3</td>
</tr>
<tr>
<td>CPLEX 9.1</td>
</tr>
<tr>
<td>DECIS</td>
</tr>
<tr>
<td>DICOPT</td>
</tr>
<tr>
<td>KNITRO 4.0</td>
</tr>
<tr>
<td>LGO</td>
</tr>
<tr>
<td>MILES</td>
</tr>
<tr>
<td>MINOS</td>
</tr>
<tr>
<td>MOSEK 3.2</td>
</tr>
<tr>
<td>MPSGE</td>
</tr>
<tr>
<td>MSNLP</td>
</tr>
<tr>
<td>OQNLP</td>
</tr>
<tr>
<td>OSL V3</td>
</tr>
<tr>
<td>OSLSE</td>
</tr>
<tr>
<td>PATH</td>
</tr>
<tr>
<td>SBB</td>
</tr>
<tr>
<td>SNOPT</td>
</tr>
<tr>
<td>XA</td>
</tr>
<tr>
<td>XPRESS 15.30</td>
</tr>
</tbody>
</table>

Multiple Solvers & Platforms
Basic Portfolio Selection

- Markowitz
- Closing Values of Stocks over Time
- Calculate Return, Deviation, Covariance using Closing Values
- Select Portfolio to
  - Minimize Variance
  - s.t. Some Return Goal
  - Minimum Portfolio Size per Stock
Basic Portfolio Selection Model QP1
- Data from Excel QP2
Gams Data eXchange

- Separation of Responsibility for Data and Model (Data contract)
- Gams Data eXchange (GDX):
  - A priori validation of data contract
  - Complements the ASCII text data input
GDX Tools

GDX

IDE

gdxsql

gdxsplit

gdxmerge

gdxdump

gdxdiff

gdxxrw (MS Office)

gams

GDX API
Reporting

- Integrated Standard Reporting Facilities
  - text files
  - GDX: MS Office, DB, Charting, …

- Specialized Reporting
  - Scenario Management/Data Cube: VEDA
  - Geographical Information System: MapInfo
  - Visualization with MATLAB
Contrast pure declarative approach:
- e.g. AMPL, MPL, OPL

Scripting added outside language
- AMPL Script, OPL script
- Programming Languages API (OptiMax)

Experience shows importance of procedural paradigm
- Part of the GAMS Language
- Procedural Elements QP3
What is Grid Computing?

- A pool of connected computers managed and available as a common computing resource
  - Allows parallel task execution
  - Allows effective sharing of CPU power
  - For us, not necessarily distributed
  - Scheduler handles management tasks
  - Can be rented or owned in common
  - E.g. Condor, Sun Grid Engine, Globus
Economics of Grid Computing

- Yearly cost, 2-CPU workstation: $5200
  - Hardware - $1200
  - Software - $4000
- Hourly cost on the grid: $2
  - $1/hour for CPU time (to grid operator)
  - $1/hour for software (GAMS, model owner)
- 1 workstation == 50 hrs/week grid time
- Up-front vs. deferred, as-needed costs
Energy planning problems

- Given expected energy demands, costs, and supplies, one can quickly form an optimal plan.
- Uncertain data → many possible scenarios
- Planning time increases dramatically
  - 30 sec/scen * 4K scen = 33 hours
- Limits the number of scenarios allowed
Solution A: Optimize for Speed

- Modify model to **batch** scenarios
- Eliminate generation time
  - GAMS generates each model from scratch
  - Fraction of total time used is non-trivial
  - Requires significant programming
  - Increased chance of error
  - Not maintainable/flexible/layered
- Speedup: 1.5 (at most!), or 22 hours
Solution B: Use a Grid

- Solve the scenarios in parallel, e.g.
  - 100 CPUs: 20 minutes
- Marginal cost is $66
- No programming required
- Model stays maintainable
- Separation of model and solution maintained
Grid Specifics

```bash
#!/bin/bash
mkdir finished
chmod 750 runit.sh
${3}runit.sh > /dev/null &
```

Parallel Submission QP4
Massively Parallel MIP

- MIP/B&C Algorithm ideal to parallelize
  - Seymour problem solved that way
  - Software: FATCOP/Condor, BCP/PVM
  - A priori subdivision in $n$ independent problems

- Open Pit Mining Problem (openpit in GAMS Model library)
  - Used integer variables to subdivide Model into 4096 sub-problems
  - Experiments (Ferris) at UW using Condor Pool
Results

- Submission start Nov. 9 at 6 pm
- All job submitted by Nov. 9 at 7:30 pm
- All jobs returned by Nov 10, 10:15 am
  - 16 hours wall time, 1812 CPU hours
  - Peak number of CPUs: 270
- Different Instance:
  - 24 hours wall time, 3000 CPU hours
Condor Statistics
Other Examples

- Student-centric scheduling at USMA (West Point)
  - Models used intensively 4 times/year
  - 4000 independent MIP models

- Hill & Associates (energy/coal experts)
  - National Power Model (coal & electricity)
  - Multi-period (20 yr) model: runs 10 hours
  - Tens of runs & quick turnaround required
Model Deployment

- All options are available
  - Spreadsheets, databases
  - Web services
  - GUI (IDE, custom built)
  - On the Grid
- Independent of model or data sources
- Basic philosophy: layers of separation
Fully Embedded Application

WEB CLIENT

Web Front End

Communication via HTML/JAVA Servlets

Graph Solution as HTML (JAVA Servlets)

SERVER

Server

JAVA Servlet Spawns GAMS

Return Solution

GAMS

BACK END APPLICATION

Access / Excel Database
User may specify:

- Expected Return
- Investment Period (Days)
- Number of Stocks
- Solver
- Database (Excel or Access)

Portfolio Optimization Using GAMS

This application illustrates embedding GAMS into a real-world web-based financial application. The model used is the financial optimization model `portfolio.gms`, which minimizes the risk given a user-specified expected return.

The submitted data is passed on to the server, which solves the model using GAMS and returns the results in real-time back to the client browser.

Minimum Return (in %): 25
Period (in days): 30 Days
Number of Stocks (1-170): 10
Solver: MINOS
Back-end Database: Excel

Background

A standard formulation for the optimal portfolio problem looks like:

\[
\begin{align*}
\text{min} & \quad x'Qx \\
\text{subject to} & \quad r'x \geq R \\
& \quad Ax = b \\
& \quad x \geq 0
\end{align*}
\]

The Q matrix is often a variance-covariance matrix. \( r(i) \) is the return on investment instrument \( i \) and \( R \) is the required return on the portfolio.
Output:

- User input parameters
- Real-time GAMS output of model solve
Output:

- GAMS Return Codes
- Investment distribution
- Graphical output

--- Reading solution for model qpl
--- portfolio_excel.gms (114) 4 Mb
--- Putfile fers
C:\gamspart\conferences\informs2004\portfolio\models\results.txt
*** Status: Normal completion

<table>
<thead>
<tr>
<th>Model Status</th>
<th>Solver Code</th>
<th>Stock</th>
<th>Investment Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.00</td>
<td>1.00</td>
<td>GAB</td>
<td>40.54%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GAP</td>
<td>4.27%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GDW</td>
<td>4.45%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GE</td>
<td>0.00%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GEN</td>
<td>19.87%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GENZ</td>
<td>0.00%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GGUY</td>
<td>0.13%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GIS</td>
<td>0.00%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GM</td>
<td>0.75%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GME</td>
<td>2.27%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GNE</td>
<td>15.48%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GNTX</td>
<td>6.99%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GPS</td>
<td>3.83%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GPTA</td>
<td>1.43%</td>
</tr>
</tbody>
</table>