Computing in the Cloud and High Performance Computing with GAMS

Franz Nelißen – FNelissen@gams.com
Agenda

Introduction

Computing in the Cloud - Solving MANY Scenarios

High Performance Computing - Solving HUGE Problems
Company

- *Roots at World Bank*, went commercial in 1987, pioneered Algebraic Modeling Languages

- GAMS Development Corp. (USA), GAMS Software GmbH (Germany)

- Software Tool Provider
GAMS at a Glance

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)

Used in more than 120 countries both for research and production in a broad range of applications

<table>
<thead>
<tr>
<th>Agricultural Economics</th>
<th>Applied General Equilibrium</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chemical Engineering</td>
<td>Economic Development</td>
</tr>
<tr>
<td>Econometrics</td>
<td>Energy</td>
</tr>
<tr>
<td>Environmental Economics</td>
<td>Engineering</td>
</tr>
<tr>
<td>Finance</td>
<td>Forestry</td>
</tr>
<tr>
<td>International Trade</td>
<td>Logistics</td>
</tr>
<tr>
<td>Macro Economics</td>
<td>Military</td>
</tr>
<tr>
<td>Management Science/OR</td>
<td>Mathematics</td>
</tr>
</tbody>
</table>
Algebraic Modeling Languages (AML)

➢ Specialized programming languages for mathematical optimization problems

➢ Similar to algebraic notation: Model is executable algebraic description of optimization problem

➢ Not a solver: Algebraic Modeling Languages interact with solver, but do not solve problem directly

➢ Increased productivity: Simplified model development & maintenance
Nowadays AML are Standard for . . .

- Broad range of application areas
- Diverse kinds of users
  - Anyone who took an “optimization” class
  - Newcomers to optimization
  - “Domain Experts” and anyone else with a technical background
- ... and trends favor this direction
  - Steadily faster and more powerful off-the-shelf solvers
  - Expanding options to incorporate models within hybrid schemes
Solving MANY Scenarios

What is time consuming?

- Model Generation / Update
  - GAMS Scenario Solver
- Solver (Optimization)
  - GAMS Grid Computing Facility
  - Solver on Demand
- Both
GAMS Scenario Solver - Basics

Model Generation ➢ Generates model once and updates the algebraic model keeping the model “hot” inside the solver

Solver Optimization ➢ Platform independent, works with all solvers

Model Update ➢ Performance close to native solver API
### GAMS Scenario Solver - Performance

**Stochastic model: 66,320 (linear) instances**

<table>
<thead>
<tr>
<th>Setting</th>
<th>Solve time (secs)</th>
<th>(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loop - communication through files</td>
<td>7,204</td>
<td>100%</td>
</tr>
<tr>
<td>Loop - in core communication with solver</td>
<td>2,481</td>
<td>34%</td>
</tr>
<tr>
<td><strong>GAMS Scenario Solver</strong></td>
<td>392</td>
<td>5%</td>
</tr>
<tr>
<td>CPLEX Concert Technology</td>
<td>210</td>
<td>2%</td>
</tr>
</tbody>
</table>
Grid Computing Facility

Runs GAMS jobs in a distributed environment

- Scalable, supports large grids (but also works on local machine)
- Solver and Platform independent
- Only minor changes to model required

1. Submission of jobs
2. “Grid Middleware”
   - Distribution of jobs
   - Job execution
3. Collection of solutions
4. Processing of results
Solver on Demand ("SaaS")

- **Network Enabled Optimization System**
  - Free “optimization on demand” hosted by University of Wisconsin, Madison
  - Access through Website, email, or modeling language (Kestrel)
  - Over 40 solvers, several optimization modeling languages
  - More than 537.00 total jobs submitted in 2017

- **Satalia (UK): (kind of) commercial NEOS**

- **Solver-Specific**
  - Gurobi Instant Cloud
  - IBM DoCloud
Application – Cloud Computing

Scenario Analysis at xyz Company

Challenge:
➢ Solve 1,000+ scenarios (MIPs, one hour) every week overnight

Issues:
➢ Automation
➢ Security
➢ Licensing (Costs)
Application – Cloud Computing

Scenario Analysis at xyz Company

Implementation:

➢ Amazon Cloud: 1,000+ parallel machines (instances), Python, GAMS + OO Python API
➢ Fully automated setup
➢ Only encrypted (obfuscated) files in the cloud
➢ Costs for virtual “Hardware” per run: $70! (1,000 instances/run * $0.07 instance / hour)
➢ Pricing for Licenses remains tricky
High Performance Computing - Solving Huge Problems

Implementation of acceleration strategies from mathematics and computational sciences for optimizing energy system models

A PROJECT BY

[Logos of the collaborating institutions]
Energy System Models (ESM)

- Technology Database
- Climate/Weather Data
- Energy Usage Statistics
- Population/Land Use
- Scenario Input

Input:
- Energy Data Analysis Tool REMix-EnDAT
  - Demand and RE Potentials
  - Renewable Energy Potentials
  - Installable capacities, hourly power output, resource limits, cost/full load hour potentials (PV, CSP, Wind, Hydro)

- Power Demand
- Heat Demand
- Hydrogen Demand
- Demand Flexibility

Output:
- Energy System Optimization Model REMix-OptiMo
  - Temporally and spatially resolved cost-minimized energy supply

Fluctuating Renewable Energy
- Wind, Photovoltaic, Run-of-river hydro

Concentrating Solar Power

Conventional Power Plants

Solar Thermal Heat

Reservoir Hydro Power

Conventional Boiler

DC Transmission

AC Transmission

H₂-Vehicles

Electrical Mobility

Result: Strategies for Generation, Transmission and Balancing
- Generation, storage and grid capacity expansion
- Hourly system operation
- Capacity utilization
- Supply system costs
- CO₂ emissions

Motivation

Energy system models (ESM) must grow in complexity to provide valuable quantitative insights for policy makers and industry:

➢ Uncertainty
➢ Large shares of renewable energies
➢ Complex underlying electricity systems

Challenge:

➢ Increasing complexity makes solving ESM more and more difficult: Need for new solution approaches
➢ ESM is just one potential field of application
Model Parameters that Drive Complexity

- **Time**
  - Planning Horizon
    - short term
    - long term

- **Discretization**
  - coarse
  - fine

---

Regional Aggregation

---

Technology Parameters

---

(Very-) Large-scale LP
- Scalable (resolution time, space, and technology)
- Block Structure
Solver: PIPS-IPM

- Open-Source parallel interior-point solver for LPs (and QPs), designed for high performance computing (HPC) platforms
- Originally for stochastic problems, extension to support linking constraints implemented by ZIB
- Already solved problems with more than $10^9$ variables
- Main developer: Cosmin Petra (Argonne National Lab.)
Model Annotation & Distributed Generation

Original problem with “random” matrix structure

\[
\begin{align*}
\text{min/max} & \quad c \\
A & \quad \leq x \\
\leq & \quad \star
\end{align*}
\]

Permutation reveals block structure required by PIPS API

\[
\begin{align*}
\text{min/max} & \quad \color{red}c & \quad \color{green}c \\
A & \quad \color{red}A & \quad \color{green}A & \quad \color{red}b \\
\leq & \quad \leq & \quad \leq & \quad \leq \star
\end{align*}
\]

To which block do variable \( x \) and equation \( E \) belong?
Also tested on other target platforms:

- Many-core platforms (JUQUEEN)
- Hazel Hen Supercomputer at the High-Performance Computing Center Stuttgart
- Intel Xeon Phi Processors
## Some Current Computational Results

<table>
<thead>
<tr>
<th>#regions</th>
<th>#time steps</th>
<th>#blocks</th>
<th>mode</th>
<th>#rows</th>
<th>#columns</th>
<th>#NZ</th>
<th>Time [hh:mm:ss]</th>
<th>Memory [GB]</th>
</tr>
</thead>
<tbody>
<tr>
<td>250</td>
<td>8760</td>
<td>366</td>
<td>mono</td>
<td>21,725,667</td>
<td>23,917,146</td>
<td>86,449,311</td>
<td>08:36</td>
<td>16.50</td>
</tr>
<tr>
<td>366</td>
<td>366 seq</td>
<td>21,725,667</td>
<td>23,917,146</td>
<td>86,449,311</td>
<td>06:37</td>
<td>6.50</td>
<td></td>
<td></td>
</tr>
<tr>
<td>366</td>
<td>mono</td>
<td>21,725,667</td>
<td>23,917,146</td>
<td>86,449,311</td>
<td>02:12</td>
<td>366 x 3.40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>366</td>
<td>seq</td>
<td>21,725,667</td>
<td>23,917,146</td>
<td>86,449,311</td>
<td>01:05</td>
<td>366 x 1.30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>366</td>
<td>mono</td>
<td>21,725,667</td>
<td>23,917,146</td>
<td>86,449,311</td>
<td>00:26</td>
<td>366 x 0.08</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1461</td>
<td>mono</td>
<td>654,329,207</td>
<td>720,310,215</td>
<td>2,603,662,965</td>
<td>03:04</td>
<td>1461 x 1.29</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Model Generation only!
Summary

➢ Various options to run “many” scenarios

➢ What are the bottlenecks?

➢ Cloud Computing
   ➢ Great potential for “burst computing”
   ➢ Automation, security, and licensing

➢ Very large problems remain challenging
Thank You

Questions?