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

Agricultural Economics	Applied General Equilibrium
Chemical Engineering	Economic Development
Econometrics	Energy
Environmental Economics	Engineering
Finance	Forestry
International Trade	Logistics
Macro Economics	Military
Management Science/OR	Mathematics

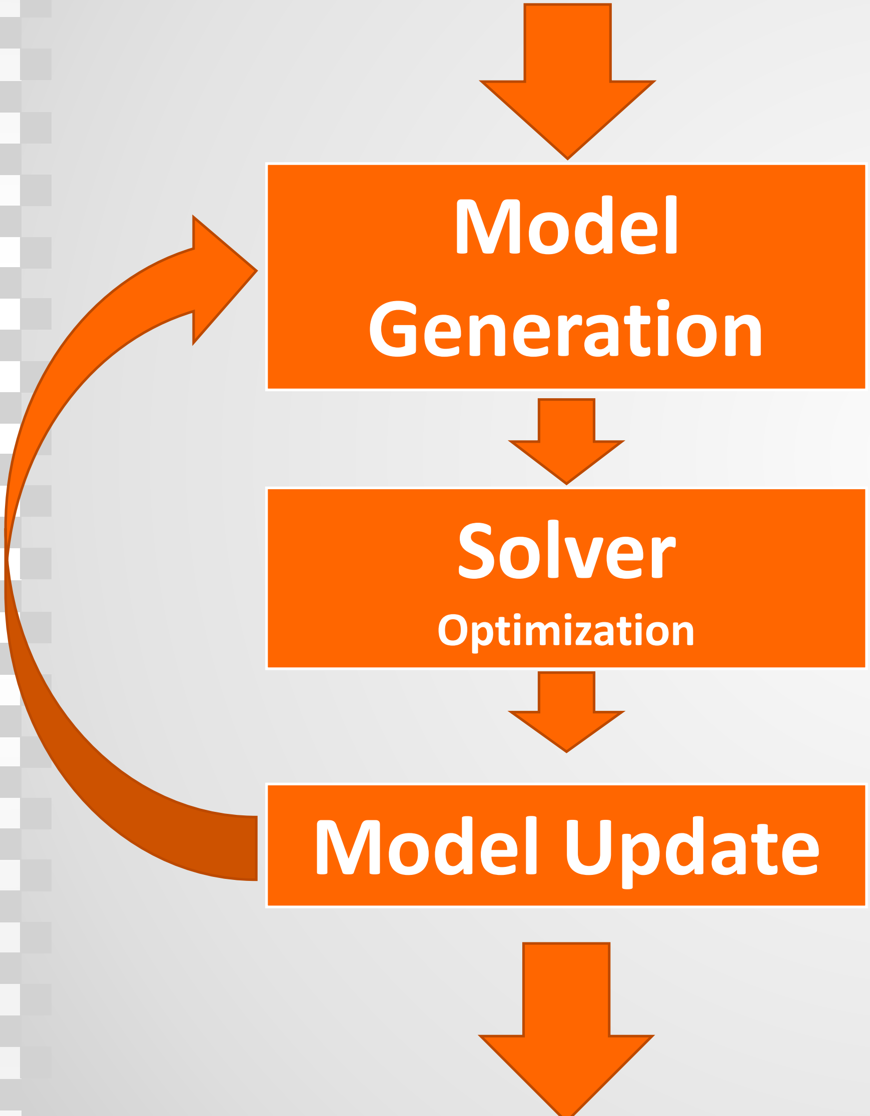
Algebraic Modeling Languages (AML)

- 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

Nowadys 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

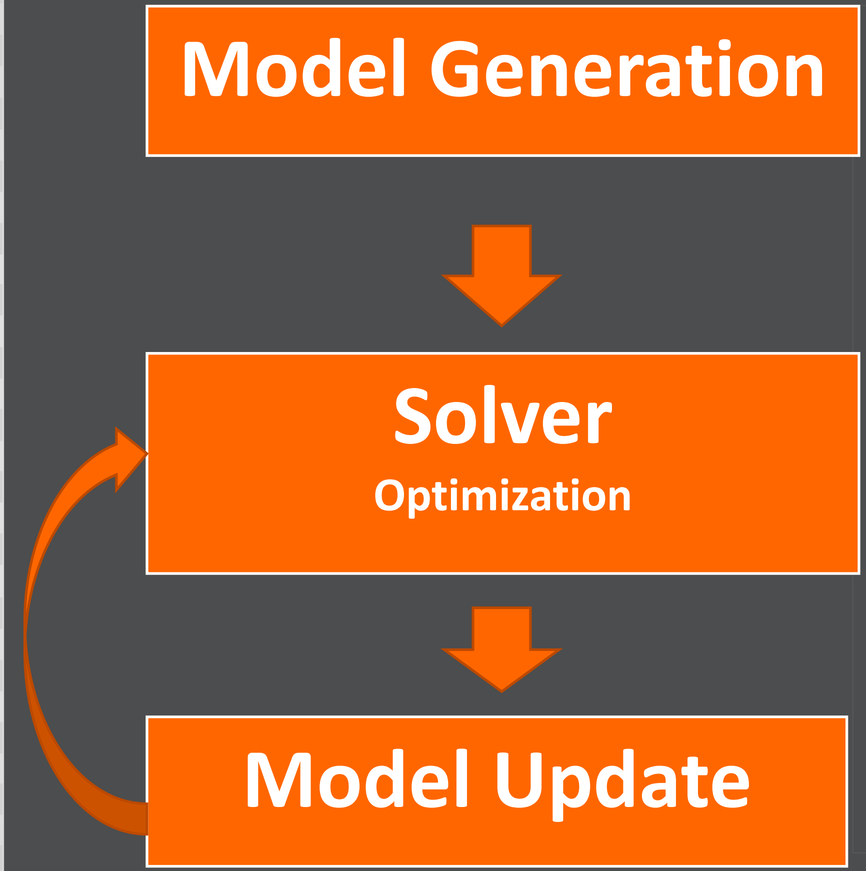
Solving MANY Scenarios



What is time consuming?

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

GAMS Scenario Solver - Basics



- Generates model once and updates the algebraic model keeping the model “hot” inside the solver
- Platform independent, works with all solvers
- Performance close to native solver API

GAMS Scenario Solver - Performance

Stochastic model: 66,320 (linear) instances

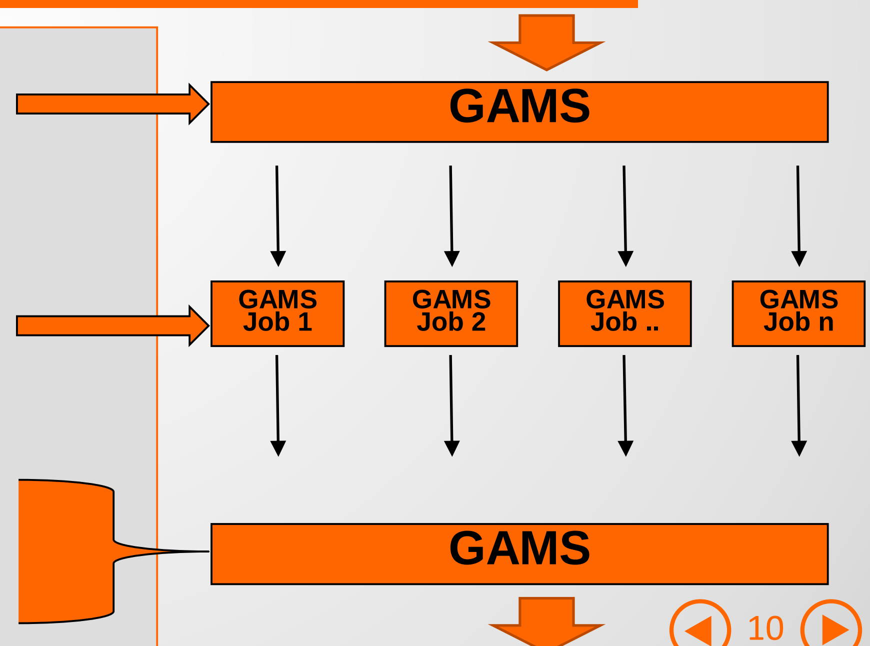
Setting	Solve time (secs)	(%)
Loop - communication through files	**7,204**	**100%**
Loop - in core communication with solver	2,481	34%
GAMS Scenario Solver	**392**	**5%**
CPLEX Concert Technology	210	2%

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")

- **N**etwork **E**nabled **O**ptimization **S**ystem
  - 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

BEAM-ME

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

A PROJECT BY

HLRS

JÜLICH FORSCHUNGSZENTRUM

Deutsches Zentrum für Luft- und Raumfahrt German Aerospace Center DLR

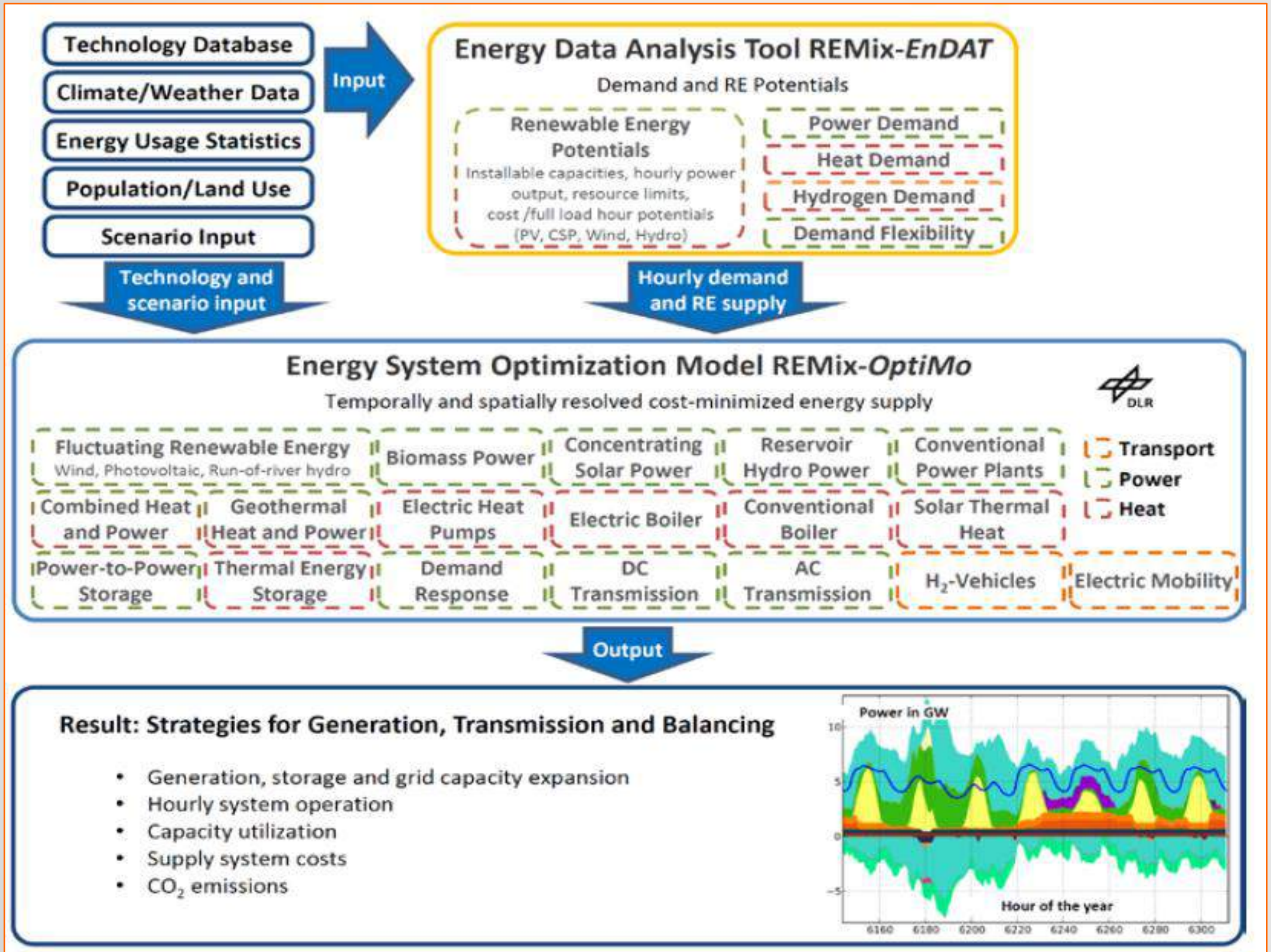
ZIB

Supported by:

Federal Ministry for Economic Affairs and Energy

on the basis of a decision by the German Bundestag

Energy System Models (ESM)



http://www.dlr.de/Portadata/41/Resources/dokumente/institut/system/Modellbeschreibungen/DLR_Energy_System_Model_REMix_short_description_2016.pdf

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

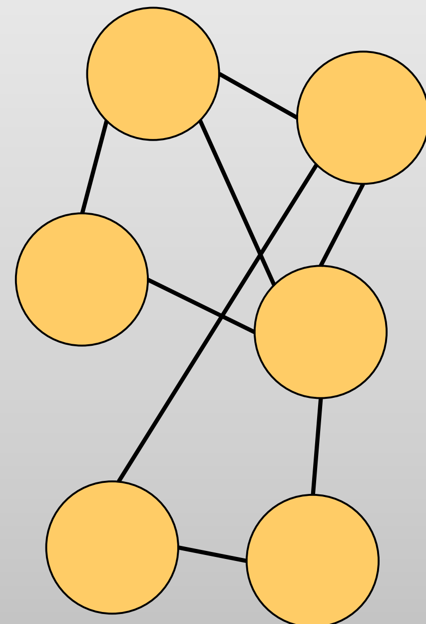
long term

Discretization

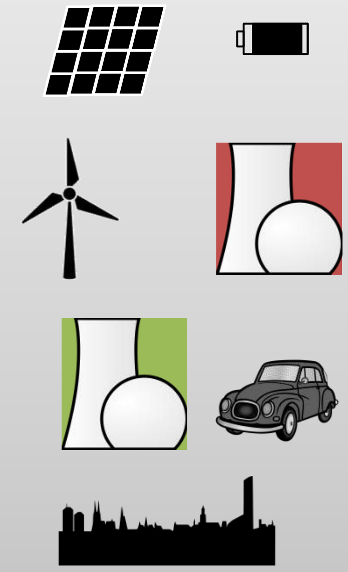
coarse

fine

Regional Aggregation



Technology Parameters



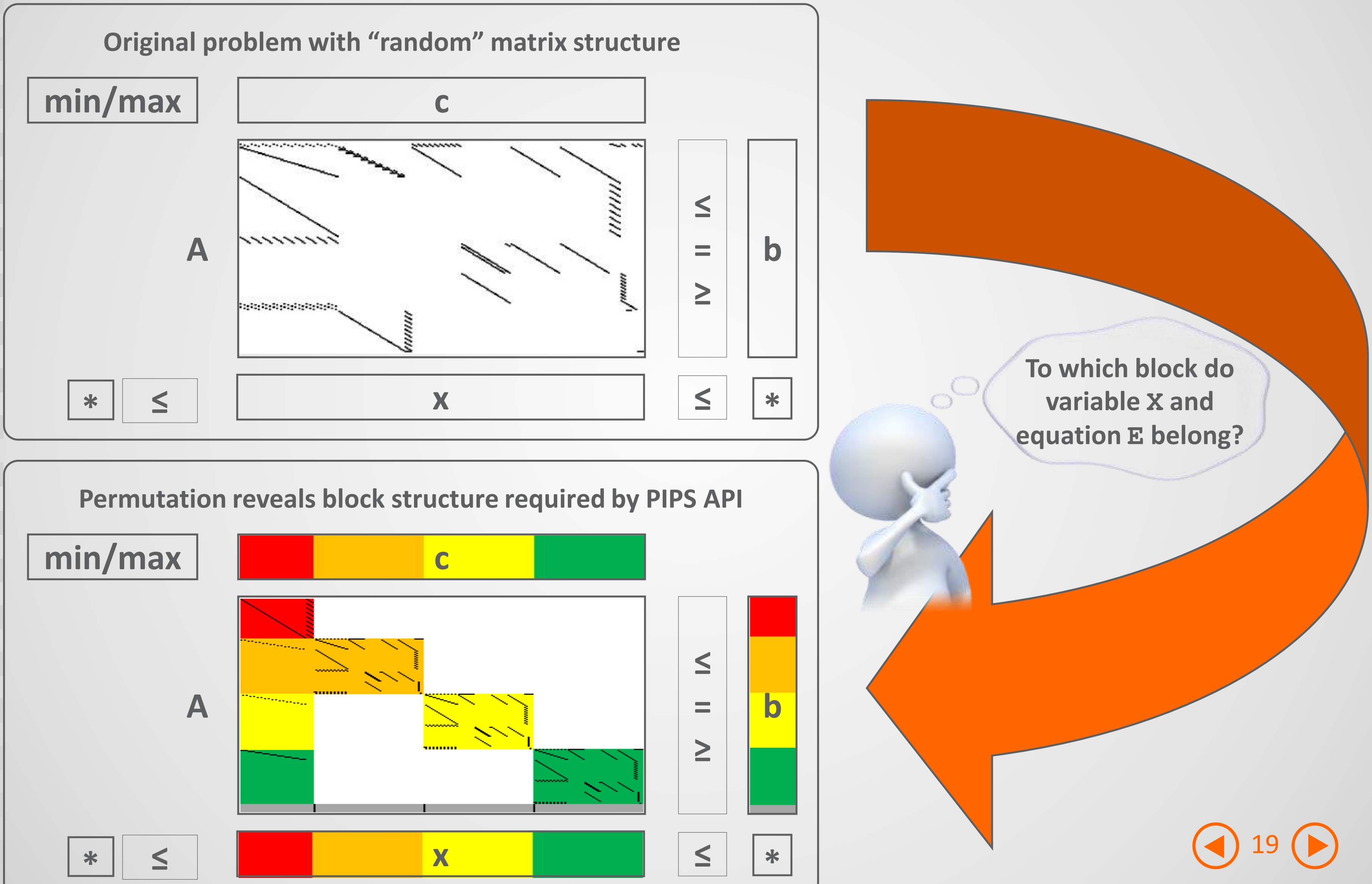
(Very-) Large-scale LP

- 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



JURECA at Jülich Supercomputing Centre

Hardware Characteristics of the Cluster Module

- 1872 compute nodes
  - Two Intel Xeon E5-2680 v3 Haswell CPUs per node
    - 2 x 12 cores, 2.5 GHz
    - Intel Hyperthreading Technology (Simultaneous Multithreading)
    - AVX 2.0 ISA extension
  - 75 compute nodes equipped with two NVIDIA K80 GPUs (four visible devices per node)
    - 2 x 4992 CUDA cores
    - 2 x 24 GiB GDDR5 memory
  - DDR4 memory technology (2133 MHz)
    - 1605 compute nodes with 128 GiB memory
    - 128 compute nodes with 256 GiB memory
    - 64 compute nodes with 512 GiB memory

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

#regions	#time steps	#blocks	mode	#rows	#columns	#NZ	Time [hh:mm:ss]	Memory [GB]
250	8760	366	mono	21,725,667	23,917,146	86,449,311	08:36	16.50
		366	seq	21,725,667	23,917,146	86,449,311	06:37	6.50
366 blocks in parallel (8 nodes @JURECA)			*p1*	21,725,667	23,917,146	86,449,311	*02:12*	*366 x 3.40*
366 blocks in parallel (8 nodes @JURECA)			*p2*	21,725,667	23,917,146	86,449,311	*01:05*	*366 x 1.30*
366 blocks in parallel (8 nodes @JURECA)			p3	21,725,667	23,917,146	86,449,311	00:26	366 x 0.08
2000	35040	1461	mono	654,329,207	720,310,215	2,603,662,965	07:46:41	~540.00
1461 blocks in parallel (31 nodes @JURECA)			p3	654,329,207	720,310,215	2,603,662,965	03:04	1461 x 1.29

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?