OR 2016
HAMBURG

GAMS

Solving Scenarios in the Cloud

Franz Nelißen – FNelissen@gams.com

GAMS - History

- *Roots: World Bank, 1976*
- Alex Meeraus founded GAMS Development Corp. (USA) in 1987 and GAMS Software GmbH (Germany) in 1995

- New management team in 2016

GAMS - Product

General **A**lgebraic **M**odeling **S**ystem

- Pioneered Algebraic Modeling Languages: Model is executable algebraic description of optimization problem
  - ➢ Made mathematical optimization available to broader audience (domain experts)
  - ➢ Increased productivity tremendously
  - ➢ 2012 INFORMS Impact Prize
- Evolution through more than 25 years of R&D and user feedback
- Maturity through experience and rigorous testing

Agenda

GAMS – Basic Concepts

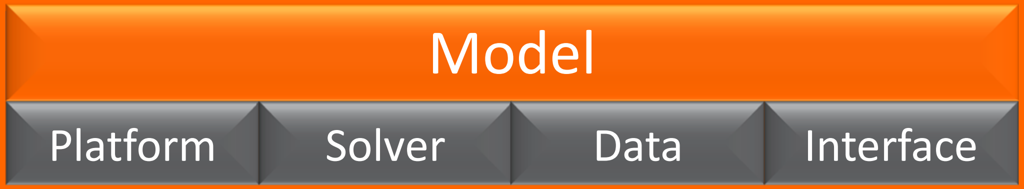
Application: Solving Scenarios in the Cloud

GAMS at a Glance

Powerful algebraic modeling language with open architecture and uniform interface to all major commercial and academic solvers (30+ integrated)

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

Different layers



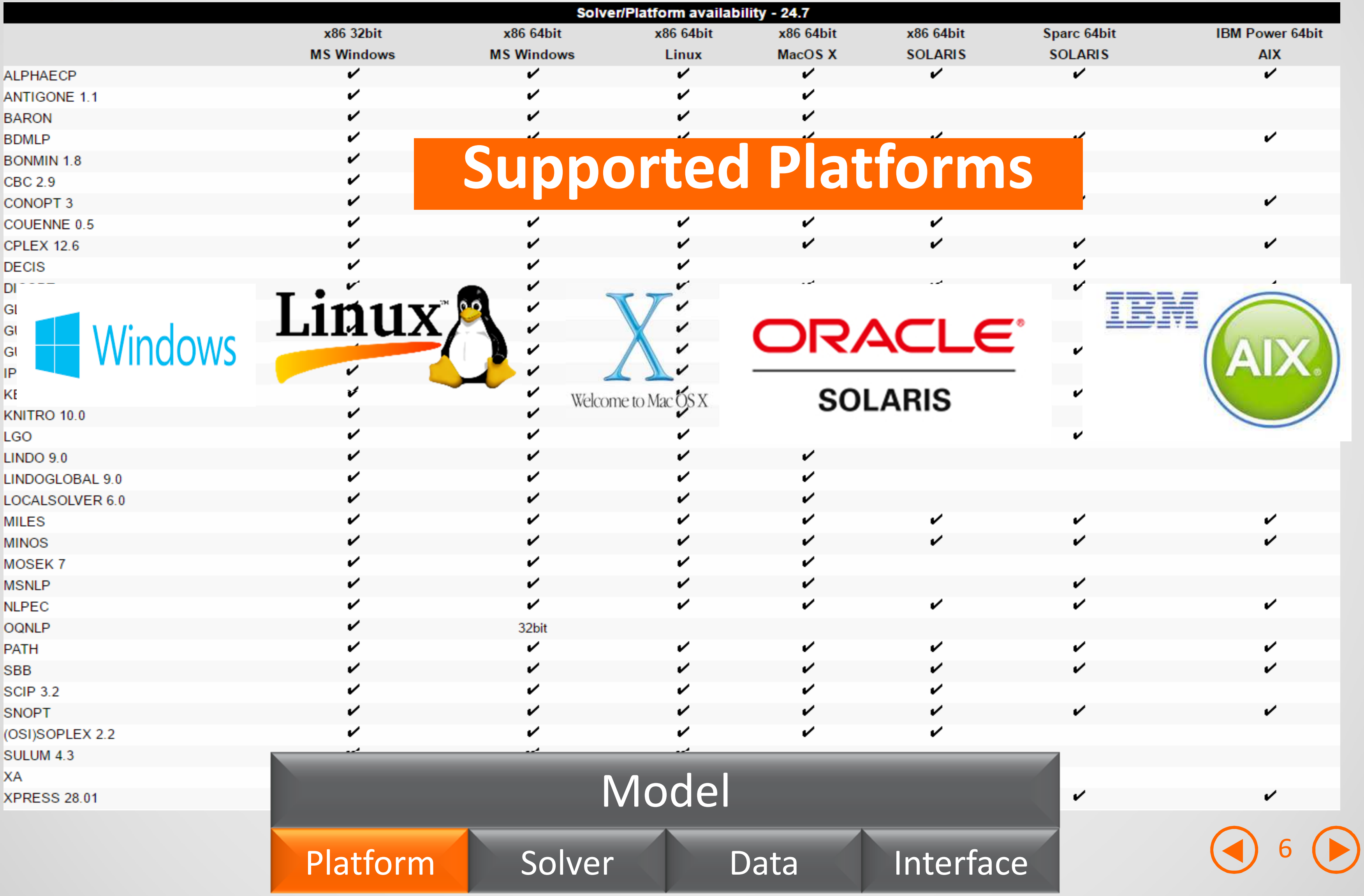
Model

Platform Solver Data Interface

GENERAL ALGEBRAIC MODELING SYSTEM

GAMS

Separation of Model and Platform



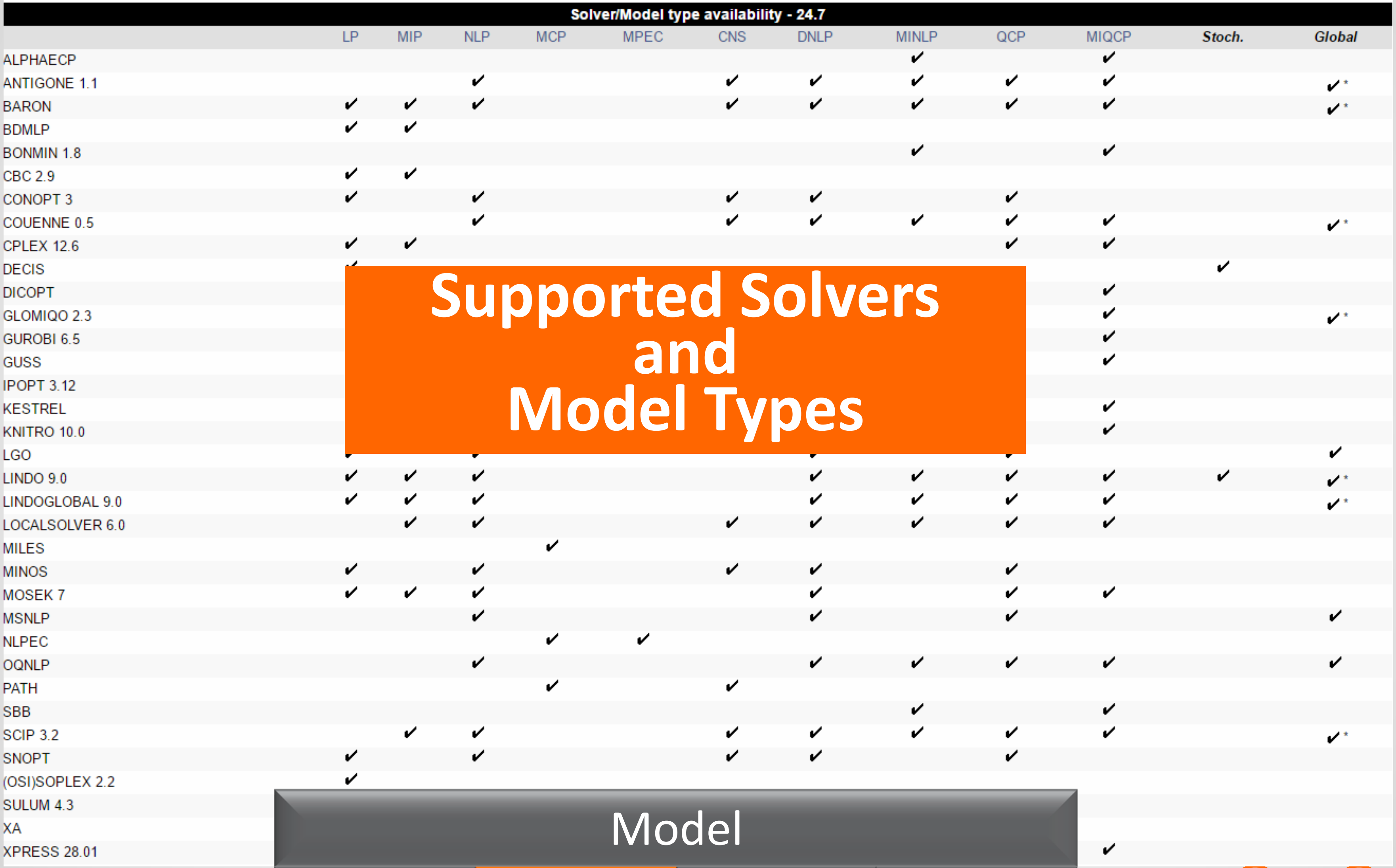
Solver/Platform availability - 24.7							
	x86 32bit	x86 64bit	x86 64bit	x86 64bit	x86 64bit	Sparc 64bit	IBM Power 64bit
	MS Windows	MS Windows	Linux	MacOS X	SOLARIS	SOLARIS	AIX
ALPHAECP	✔	✔	✔	✔	✔	✔	✔
ANTIGONE 1.1	✔	✔	✔	✔			
BARON	✔	✔	✔	✔			
BDMLP	✔	✔	✔	✔	✔	✔	✔
BONMIN 1.8	✔						
CBC 2.9	✔						
CONOPT 3	✔					✔	✔
COUENNE 0.5	✔	✔	✔	✔	✔		
CPLEX 12.6	✔	✔	✔	✔	✔	✔	✔
DECIS	✔	✔	✔			✔	
DI	✔	✔	✔	✔	✔	✔	✔
GI		✔	✔				
GI	✔	✔	✔				
GI	✔	✔	✔			✔	
IP	✔	✔	✔				
KE	✔	✔	✔			✔	
KNITRO 10.0	✔	✔	✔				
LGO	✔	✔	✔			✔	
LINDO 9.0	✔	✔	✔	✔			
LINDOGLOBAL 9.0	✔	✔	✔	✔			
LOCALSOLVER 6.0	✔	✔	✔	✔			
MILES	✔	✔	✔	✔	✔	✔	✔
MINOS	✔	✔	✔	✔	✔	✔	✔
MOSEK 7	✔	✔	✔	✔			
MSNLP	✔	✔	✔	✔		✔	
NLPEC	✔	✔	✔	✔	✔	✔	✔
OQNLP	✔	32bit					
PATH	✔	✔	✔	✔	✔	✔	✔
SBB	✔	✔	✔	✔	✔	✔	✔
SCIP 3.2	✔	✔	✔	✔	✔		
SNOPT	✔	✔	✔	✔	✔	✔	✔
(OSI)SOPLEX 2.2	✔	✔	✔	✔	✔		
SULUM 4.3	✔	✔	✔				
XA							
XPRESS 28.01						✔	✔

Supported Platforms

Model

Platform | Solver | Data | Interface

Separation of Model and Solver



Solver/Model type availability - 24.7

	LP	MIP	NLP	MCP	MPEC	CNS	DNLP	MINLP	QCP	MIQCP	*Stoch.*	*Global*
ALPHAECP								✓		✓		
ANTIGONE 1.1			✓			✓	✓	✓	✓	✓		✓*
BARON	✓	✓	✓			✓	✓	✓	✓	✓		✓*
BDMLP	✓	✓										
BONMIN 1.8								✓		✓		
CBC 2.9	✓	✓										
CONOPT 3	✓		✓			✓	✓		✓			
COUENNE 0.5			✓			✓	✓	✓	✓	✓		✓*
CPLEX 12.6	✓	✓							✓	✓		
DECIS	✓										✓	
DICOPT										✓		
GLOMIQO 2.3										✓		✓*
GUROBI 6.5										✓		
GUSS										✓		
IPOPT 3.12												
KESTREL										✓		
KNITRO 10.0										✓		
LGO	✓		✓				✓		✓			✓
LINDO 9.0	✓	✓	✓				✓	✓	✓	✓	✓	✓*
LINDOGLOBAL 9.0	✓	✓	✓				✓	✓	✓	✓		✓*
LOCALSOLVER 6.0		✓	✓			✓	✓	✓	✓	✓		
MILES				✓								
MINOS	✓		✓			✓	✓		✓			
MOSEK 7	✓	✓	✓				✓		✓	✓		
MSNLP			✓				✓		✓			✓
NLPEC				✓	✓							
OQNLP			✓				✓	✓	✓	✓		✓
PATH				✓		✓						
SBB								✓		✓		
SCIP 3.2		✓	✓			✓	✓	✓	✓	✓		✓*
SNOPT	✓		✓			✓	✓		✓			
(OSI)SOPLEX 2.2	✓											
SULUM 4.3												
XA												
XPRESS 28.01										✓		

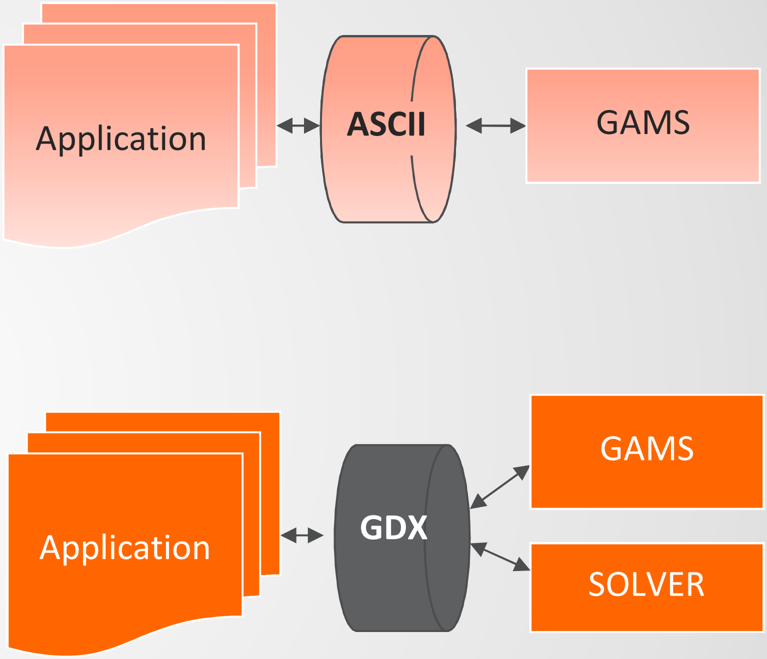
Supported Solvers and Model Types

Model

Platform | Solver | Data | Interface

Separation of Model and Data

- **Declarative Modeling**: *x(j), j∈{1,...}*
  - ASCII: Initial model development
  - GDX: Binary Data layer ("contract") between GAMS and applications
    - Platform independent
    - Direct GDX interfaces and general API

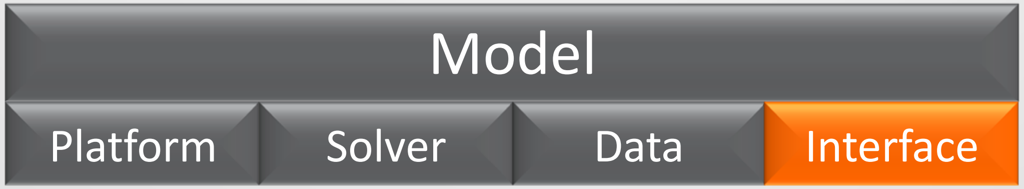


Separation of Model and User Interface

- Open architecture and interfaces to other systems
  → No preference for a particular user interface

- Smart Links to popular environments: Excel, MATLAB, R, ...

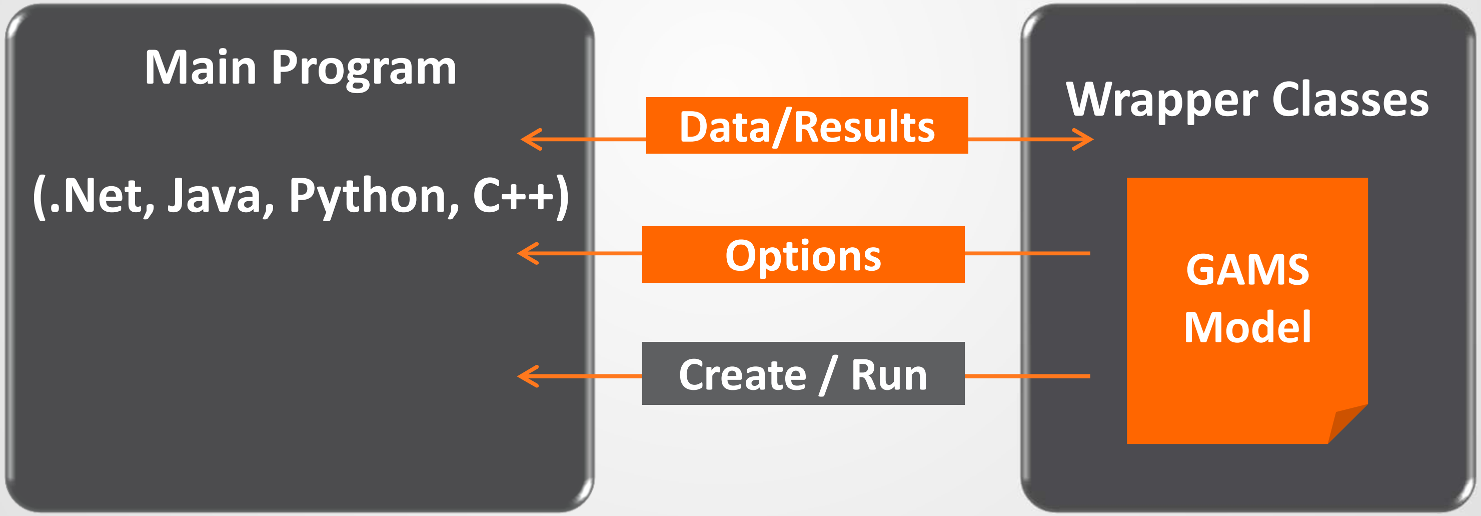
- **O**bject **O**riented **A**pplication **P**rogramming **I**nterfaces
  - .Net, Java, Python, C++,...
  - Wrapper classes that encapsulates a GAMS model
  - No modeling capability: Model is written in GAMS



Encapsulation of a 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 and run model instance(s)**



Agenda

GAMS – Basic Concepts

Application: Solving Scenarios in the Cloud

Solving “many” independent Scenarios

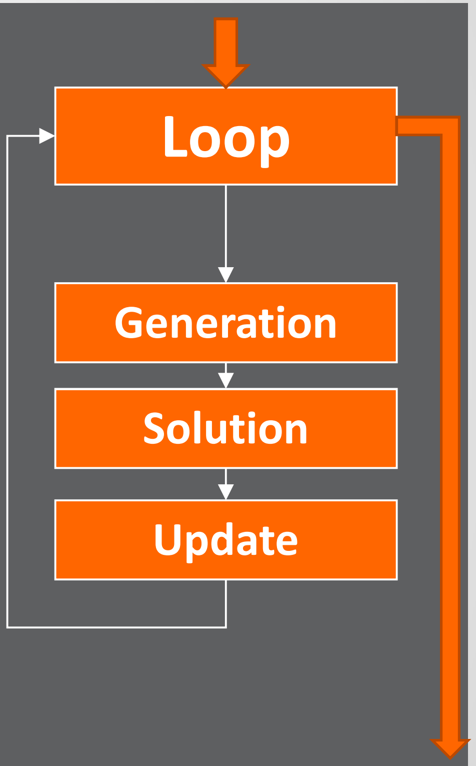
1. Small Ratio of MP solution time/GAMS overhead
   → GUSS/Scenario Solver

Simple Serial Solve Loop - Basics

Both declarative and imperative Elements

```
Set s /s*s10/;
Parameter
 A_s(s,i,j) “scenario data”,
 xlo_s(s,i,j) ”scenario lower bound for x.l”,
 em_s(s,i) “scenario solution for e.m;
Loop(s,
     A(i,j) = A_s(s,i,j);
     x.lo(j)=xlo_s(s,j);
     solver mymodel min z using lp;
     xl_s(s,j) = x.l(j);
     em_s(s,i) = e.m(i);

    );
```

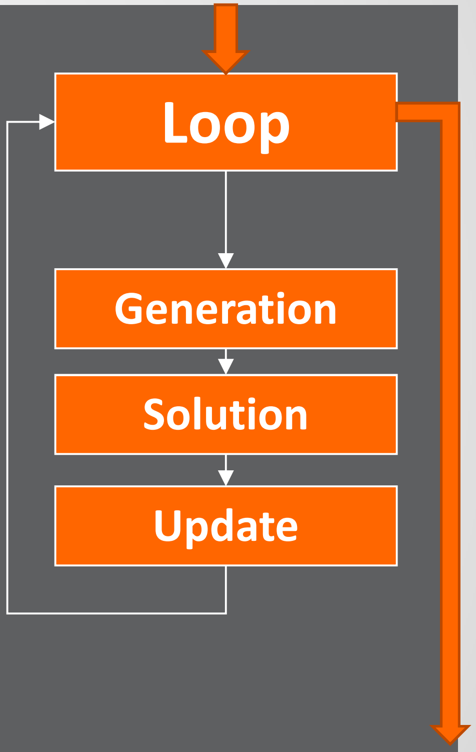


Simple Solve Loop - Performance

Different options to call the solver from GAMS

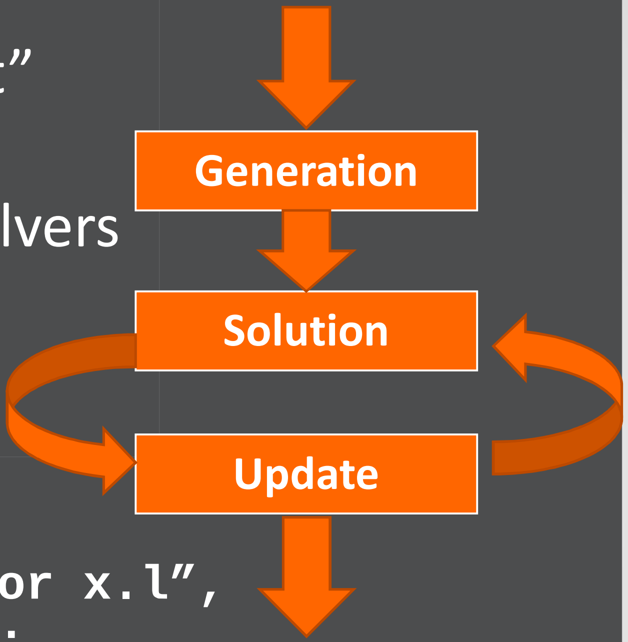
- trnsport.gms (LP) solved 500 times (CPLEX)

Setting	Solve time (secs)
Solvelink=%Solvelink.ChainScript%	52.221
Solvelink=%Solvelink.CallModule%	37.366
Solvelink=%Solvelink.LoadLibrary%	03.252



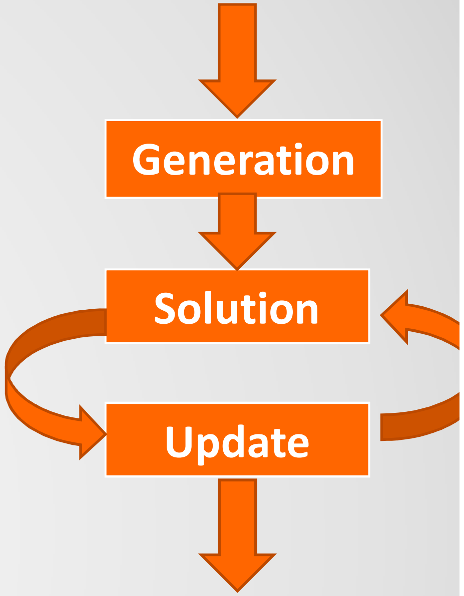
Scenario Solver - Basics

- 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



```
Set s /s*s10/;
Parameter
  A_s(s,i,j) “scenario data”,
  xlo_s(s,i,j) ”scenario lower bound for x.l”,
  em_s(s,i) “scenario solution for e.m;
Set dict / s. scenario. ´´,
           A. param.    A_s,
           x. lower.    xlo_s,
           x. level.    xl_s,
           e. marginal. em_s /;
solve mymodel min z using lp scenario dict;
```

Scenario Solver - Performance



Stochastic model: 66,320 (linear) instances

Setting	Solve time (secs)
Loop: Solvelink=%Solvelink.Chainscript (def.)	7,204
Loop: Solvelink=%Solvelink.LoadLibrary%	2,481
GAMS Scenario Solver	392
CPLEX Concert Technology	210

Solving “many” independent Scenarios

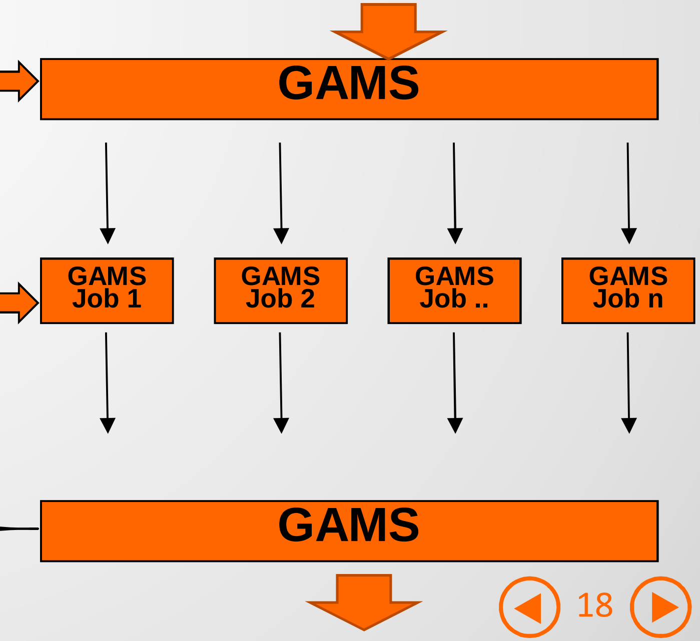
1. Small Ratio of MP solution time/GAMS overhead
   → GUSS/Scenario Solver
2. **Large ratio i.e. only MP time is relevant (pre/post processing not critical)**
   → Grid Facility, NEOS, Gurobi/Cplex Server

Grid Computing Facility

GAMS jobs in a **distributed** environment

- ➢ Scalable: supports large grids, but also works on local machine
- ➢ Platform independent, works with all solvers/model types
- ➢ 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



GAMS/Kestrel

- Remote Solver Execution on NEOS Servers
- Stay in your GAMS environment
- Results are being processed as with any local solver

Local Machine ↔ Remote Cluster (NEOS)

```
Model transport /all/;
Option lp=kestrel;
transport.optfile=1;

$onecho > kestrel.opt
kestrel_solver xpress
$offecho

Solve transport using lp minimizing z;
```

```
--- Executing KESTREL: elapsed 0:00:00.006
Connecting to: http://neos-server.org:3332
NEOS Solver: xpress
NEOS job#=3631352, pass=iJLdAkhP

Check the following URL for progress report :
http://neos-server.org/neos/cgi-bin/nph-neos-
solver.cgi?admin=results&jobnumber=3631352&pass=i
JLdAkhP
Job 3631352 dispatched
password: iJLdAkhP
---------- Begin Solver Output -----------
Job submitted to NEOS HTCondor pool.

FICO-Xpress      24.3.3 r48116 Released Sep 19,
2014 LEG x86 64bit/Linux
```

Solving “many” independent Scenarios

1. Small Ratio of MP solution time/GAMS overhead
   → GUSS/Scenario Solver
2. Large ratio i.e. only MP time is relevant (pre/post processing not critical)
   → Grid Facility, NEOS, Gurobi/Cplex Server
3. **If entire model run including pre processing/MP solve/post processing is costly**
   → Parallel/asynchronous execution of entire model in the cloud

Application – Cloud Computing

xyz Energy Company

Challenge

- Scenario Analysis: Solve 1,000+ scenarios (MIPs, one hour) every week overnight

Issues:

- Automation
- Security
- Licensing

Application – Cloud Computing

xyz – Energy Company

Implementation:

- Amazon Cloud: 1,000+ parallel machines (instances), Python, GAMS + OO Python API
- Automated setup, including
  - Starting instances
  - Prepare / Submit / Run GAMS jobs
  - Collect results
  - Stop instances

Application – Cloud Computing

Protecting IP and Sensitive Data

Options to hide sensitive information:

- Extrinsic function libraries / External Equations
- Encrypted source files / secure work files
- Obfuscated work files: change all the names and other documentation related to a specific model run

Application – Cloud Computing

Obfuscated Files – An Example

Normal Solver Log

```
Reduced LP has 5 rows, 6 columns, and 12 nonzeros.
Presolve time = 0.00 sec. (0.00 ticks)

Iteration      Dual Objective            In Variable           Out Variable
     1              73.125000    x(seattle.new-york) demand(new-york) slack
     2             119.025000     x(seattle.chicago)  demand(chicago) slack
     3             153.675000    x(san-diego.topeka)   demand(topeka) slack
     4             153.675000  x(san-diego.new-york)  supply(seattle) slack
LP status(1): optimal
```

Obfuscated Solver Log

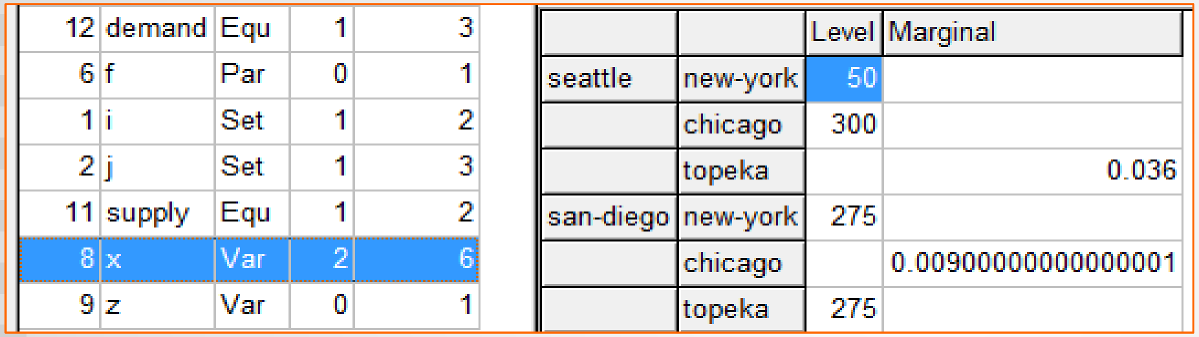
```
Reduced LP has 5 rows, 6 columns, and 12 nonzeros.
Presolve time = 0.00 sec. (0.00 ticks)

Iteration      Dual Objective            In Variable           Out Variable
     1              73.125000H('"!!!!!!'.'"!!!!!!!')A00002('"!!!!!!!' slack
     2             119.025000 H('"!!!!!!'.'#!!!!!!')A00002('#!!!!!!') slack
     3             153.675000H('"!!!!!!!!'.'"!!!!!') A00002('"!!!!!') slack
     4             153.675000H('"!!!!!!!!'.'"!!!!!!!A00001('"!!!!!!') slack
LP status(1): optimal
```

Application – Cloud Computing

Obfuscated Files – An Example

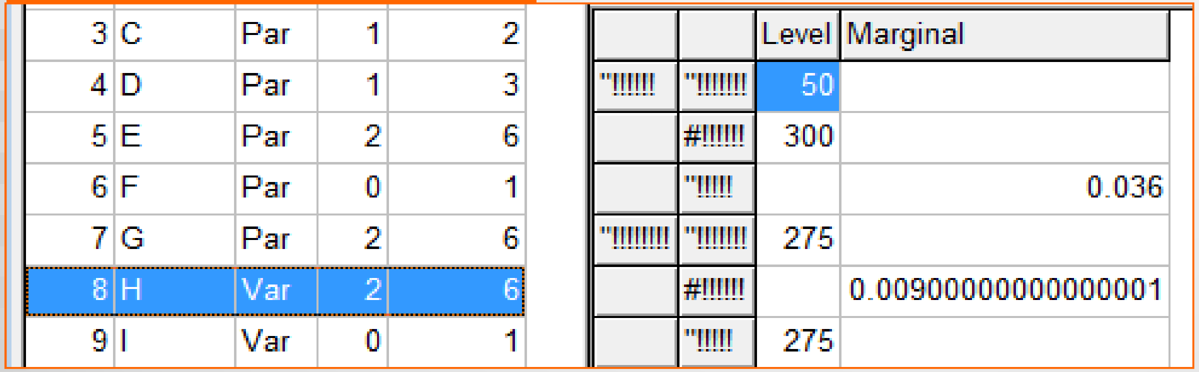
Normal gdx file



12	demand	Equ	1	3
6	f	Par	0	1
1	i	Set	1	2
2	j	Set	1	3
11	supply	Equ	1	2
8	x	Var	2	6
9	z	Var	0	1

		Level	Marginal
seattle	new-york	50	
	chicago	300	
	topeka		0.036
san-diego	new-york	275	
	chicago		0.00900000000000001
	topeka	275	

Obfuscated gdx file



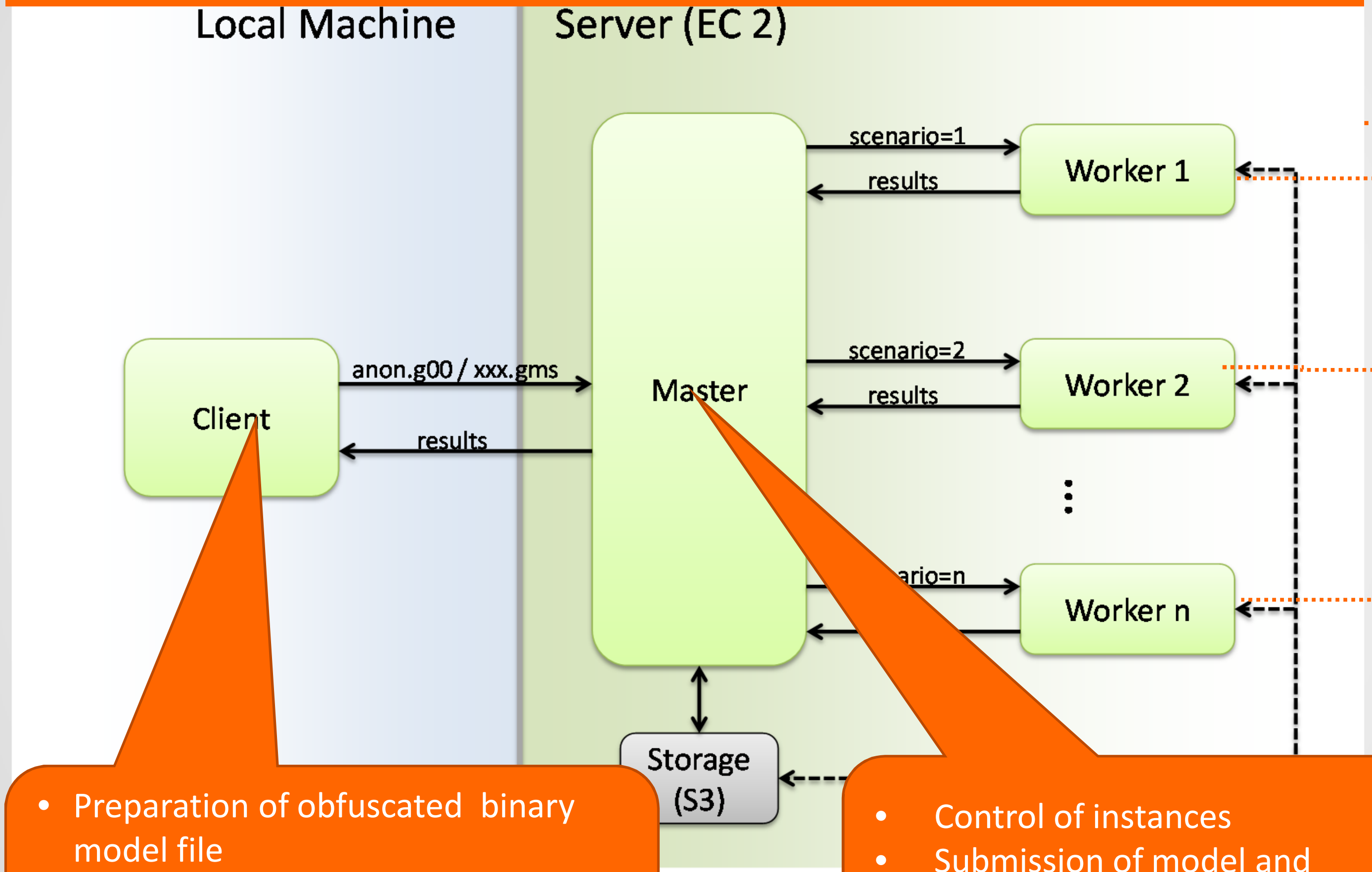
3	C	Par	1	2
4	D	Par	1	3
5	E	Par	2	6
6	F	Par	0	1
7	G	Par	2	6
8	H	Var	2	6
9	I	Var	0	1

		Level	Marginal
"!!!!!!	"!!!!!!!	50	
	#!!!!!!	300	
	"!!!!!		0.036
"!!!!!!!!	"!!!!!!!	275	
	#!!!!!!		0.00900000000000001
	"!!!!!	275	

Application – Cloud Computing

Setup

amazon web services™



Configured AWS Instances

- Python
- GAMS

- Preparation of obfuscated binary model file
- Preparation of data for every worker
- Mapping of results into original namespace

- Control of instances
- Submission of model and scenario data to workers
- Collection and merge of results

Application – Cloud Computing

Commercial Aspects

“Hardware” Amazon Cloud (1,000 instances) :

Hardware Costs / run: **$70!**

(1,000 instances/run * $0.07 instance / hour)

Software Licensing:

- Gurobi and IBM offer per-usage license
- Client with strong preference for annual license fee, not a per-usage license

Application – Cloud Computing

45 Provided Model Instances

- Statistics:
  - 163,608 – 1,959,550 rows
  - 84,930 – 983,587 var. (32,240-258,796 dis.)
  - 447,537 – 6,068,729 NZ
- CPLEX, SCIP, and CBC
- 60 minutes, gap max. 1%
- Manual option tuning for SCIP (thanks to Gerald Gamrath & Ambros Gleisner)

Application – Cloud Computing

Results

- CPLEX: All instances solved to optimality
- SCIP:
  - Could solve all 45 instances
  - But: After 60 min. 2 instances with gap > 20%
- CBC:
  - Did also well
  - But: After 60 min. no solution for some instances (< 10%)

Application – Cloud Computing

Proposed Strategy

- Run all instances simultanuesly with SCIP and CBC

  → *„hardware“ costs: $0,07 per instance hour*

- After 60 minutes take the best solution

- If necessary solve „difficult“ instances with CPLEX (outside the cloud)

GAMS

Thank You

Europe	USA
GAMS Software GmbH	GAMS Development Corp.
P.O. Box 40 59	1217 Potomac Street, NW
50216 Frechen, Germany	Washington, DC 20007, USA
Phone: +49 221 949 9170	Phone: +1 202 342 0180
Fax: +49 221 949 9171	Fax: +1 202 342 0181
info@gams.de	sales@gams.com