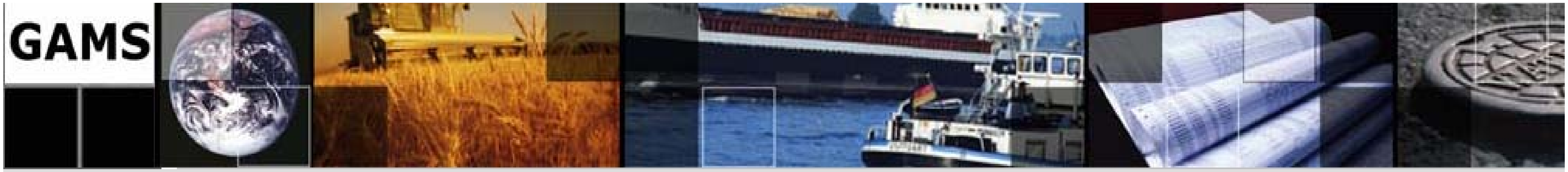


Rapid Application Prototyping using GAMS

Franz Nelißen
FNelissen@gams.com

GAMS Software GmbH
www.gams.de

APMOD 2006
Madrid, Spain, June 19 - 21, 2006

Agenda

- GAMS Development / GAMS Software
- GAMS at a Glance
- An illustrative Example: The Mean Variance Model
- Grid Computing

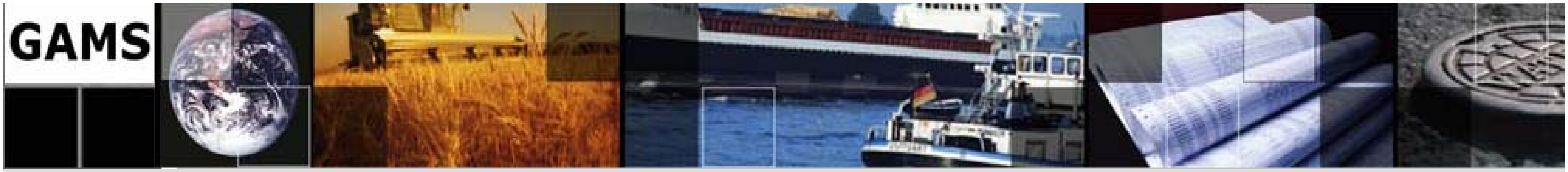
Agenda

GAMS Development / GAMS Software

GAMS at a Glance

An illustrative Example: The Mean Variance Model

Grid Computing



GAMS Development / GAMS Software

- Roots: **Research project** World Bank 1976
- Pioneer in **Algebraic Modeling Systems** used for economic modeling
- Went **commercial** in 1987
- **Offices** in Washington, D.C and Cologne

- Professional **software tool provider**
- Operating in a **segmented niche market**
- Broad **academic** & **commercial** user base and network

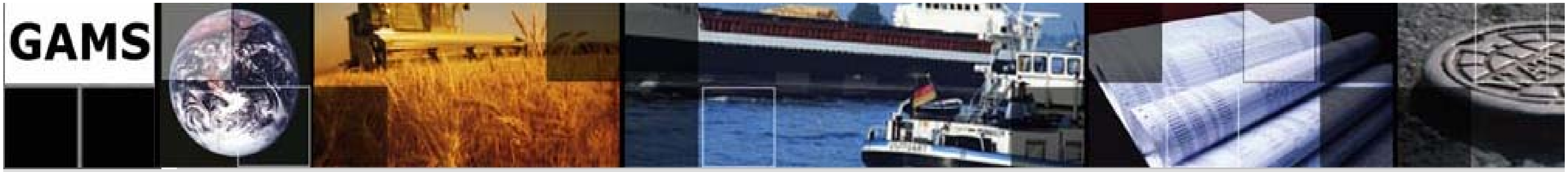
Typical Application* Areas:

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

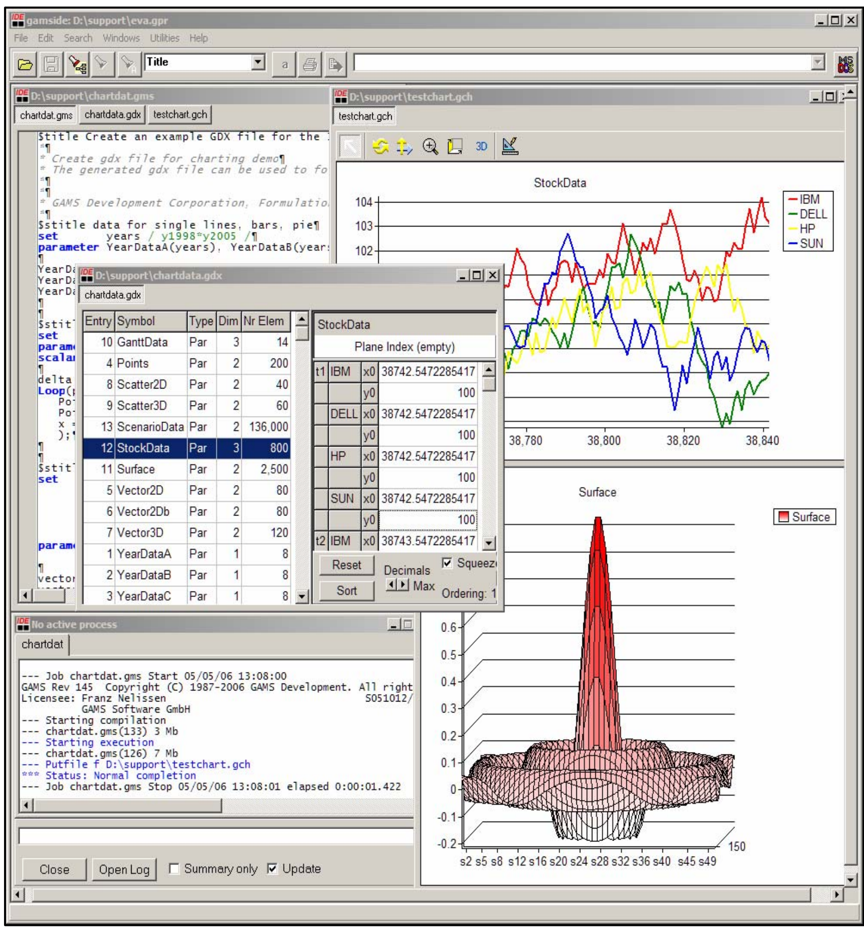
* Illustrative examples in the GAMS Model Library

Agenda

- GAMS Development/GAMS Software
- **GAMS at a Glance**
- An illustrative Example: The Mean Variance Model
- Grid Computing



GAMS at a Glance

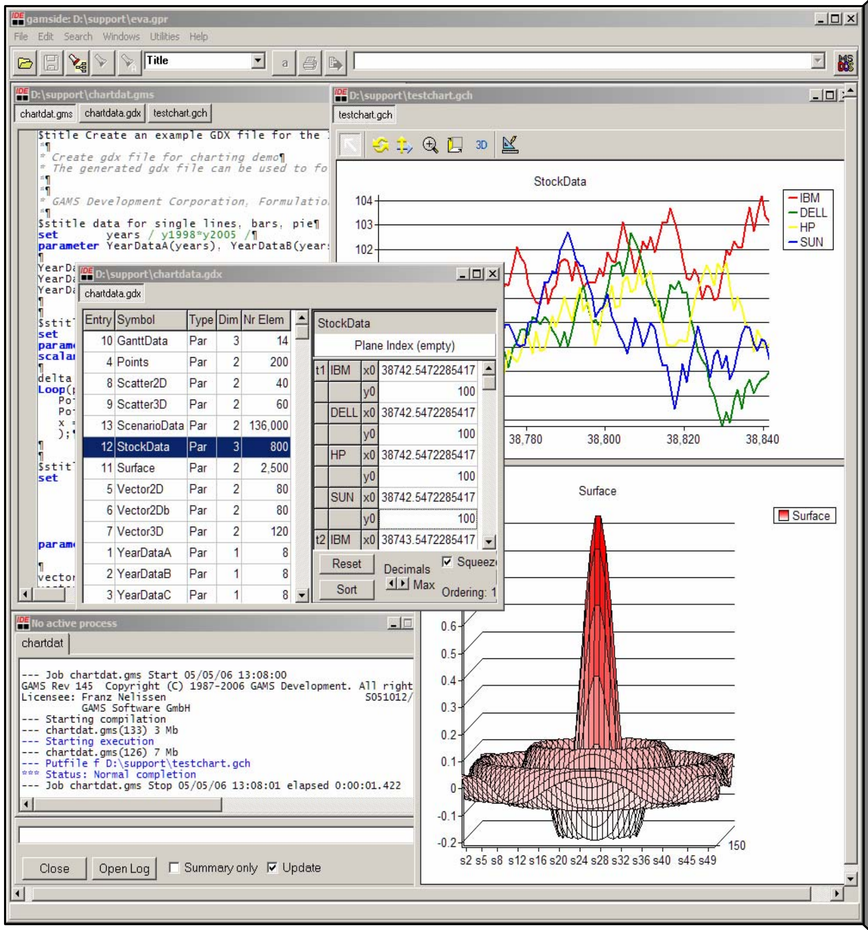


General **A**lgebraic **M**odeling **S**ystem: Algebraic Modeling Language, Integrated Solver, Model Libraries, Connectivity- & Productivity Tools

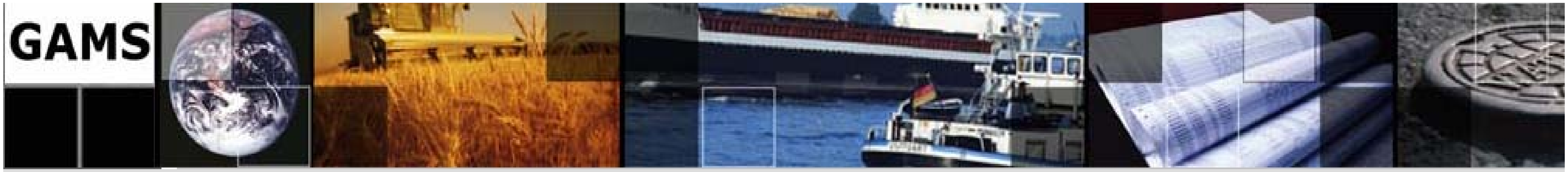
Design Principles:

- Balanced mix of declarative and procedural elements
- Open architecture and interfaces to other systems
- Different layers with separation of:
  - model and data
  - model and solution methods
  - model and operating system
  - model and interface

Benefits for Clients



- State of art professional modeling technology
- Increased productivity
- Robust and scalable
- Rapid development
- Broad Network
- Large model libraries with templates
- Multiple Model Types
- Platform / Solver independence:
  - Maintainable models
  - Protection of investments



System Overview

Interactive | **Offline/Batch**

Connectivity Tools

- Uniform Data Exchange:
  - ASCII
  - GDX (ODBC, SQL, XLS, XML)
- GDX Tools
- Data API
- Ext. programs
  - EXCEL
  - MATLAB
  - GNUPLOT, ...
  - C, Delphi, ...

User Interfaces

GAMS Language Compiler and Execution System

Solvers

LP-MIP-QCP-MIQCP-NLP-MINLP-CNS-MCP-MPEC
MPSGE, global, and stochastic optimization

Productivity Tools

- **I**ntegrated **D**evelopment **E**nvironment
- Model Debugger and Profiler
- Model Libraries
- Data Browser
- Charting Engine
- Benchmarking
- Deployment System
- Quality Assurance and Testing

BARON, COIN, CONOPT, CPLEX, DECIS, DICOPT, KNITRO, LGO,MINOS, MOSEK, OQNLP, PATH, SNOPT, XA, XPRESS, …

GAMS

Agenda

GAMS Development/GAMS Software

GAMS at a Glance

An illustrative Example: The Mean Variance Model

Grid Computing

The Mean-Variance Model

Markowitz (1952), Nobel prize 1990

Given

Some investments x_i with historical data

- **Rewards = Expected returns** of investments: μ_i (**Mean** of historical returns)
- Risk: **Variance** of investments $Q_{i,j}$

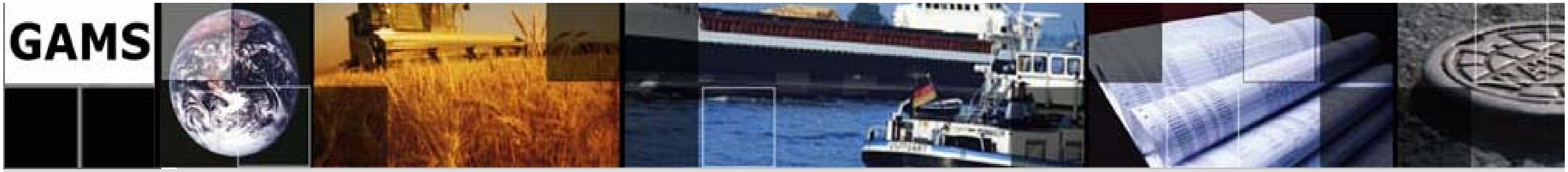
Goal

Balance risk *r* of portfolio against expected **returns** of portfolio

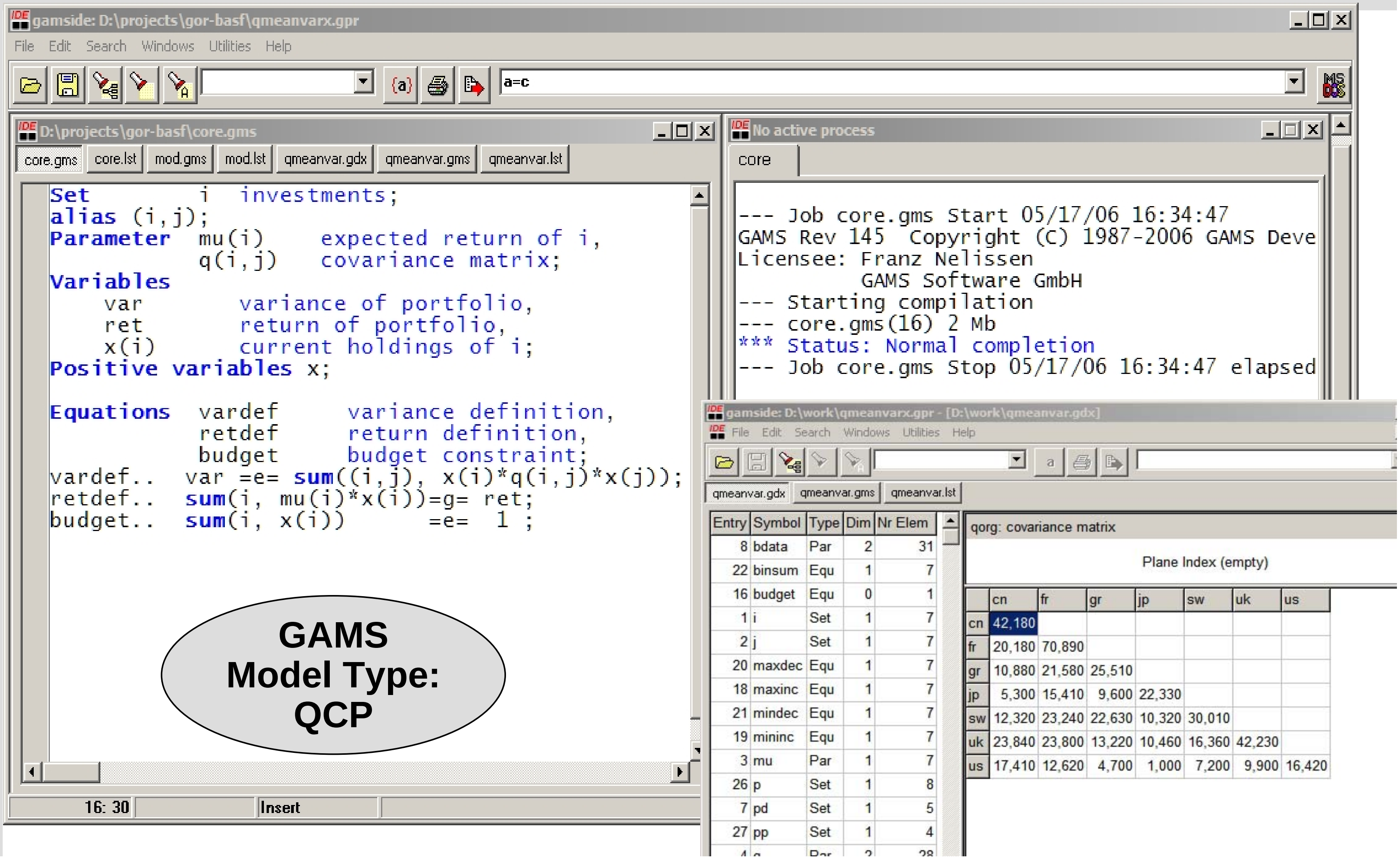
Minimize variance *v* of portfolio for a given target return *r*

MV Model Algebra

Variance of Portfolio	$Min \sum_{i=1}^{I} \sum_{j=1}^{J} x_i \, Q_{i,j} \, x_j$
Target return	$s.t. \quad \sum_{i=1}^{I} \mu_i x_i \geq r$
Budget constraint	$\sum_{i=1}^{I} x_i = 1$
No short sales	$x_i \geq 0$



Declarative Model and some Data



Trading Restrictions

"Zero or Range"-Constraint

- Revision of existing (not optimized) portfolio
- "Zero or Range"-Constraint: **Either** no trade **or** the trade must stay between pre-defined ranges both for purchase and selling
- Portfolio turnover: The total purchase of investments x_i may not exceed some threshold τ

D:\projects\gor-basf\qmeanvar.gdx

core.gms | core.lst | mod.gms | mod.lst | qmeanvar.gdx | qmeanvar.gms | qmeanvar.lst

Entry	Symbol	Type	Dim	Nr Elem
7	bdata	Par	2	31
21	binsum	Equ	1	7
15	budget	Equ	0	1
1	i	Set	1	7
2	j	Set	1	7
19	maxdec	Equ	1	7
17	maxinc	Equ	1	7
20	mindec	Equ	1	7
18	mininc	Equ	1	7
3	mu	Par	1	7
25	p	Set	1	8

bdata: portfolio data and trading restrictions

Plane Index (empty)

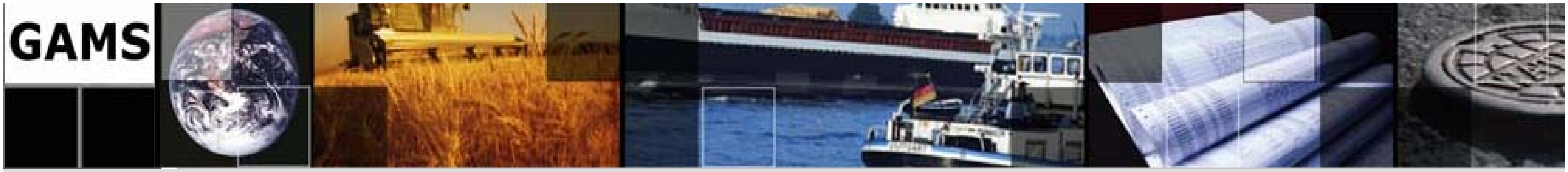
	old	umin	umax	lmin	lmax
cn	0.20	0.03	0.11	0.02	0.20
fr	0.20	0.04	0.10	0.02	0.15
gr		0.04	0.07	0.04	
jp		0.03	0.11	0.04	
sw	0.20	0.03	0.20	0.04	0.10
uk	0.20	0.03	0.10	0.04	0.15
us	0.20	0.03	0.10	0.04	0.20

e.g. cn: either no trade (20%) or new share between 23-31% (u) or between 0-18% (l)

GAMS Formulation

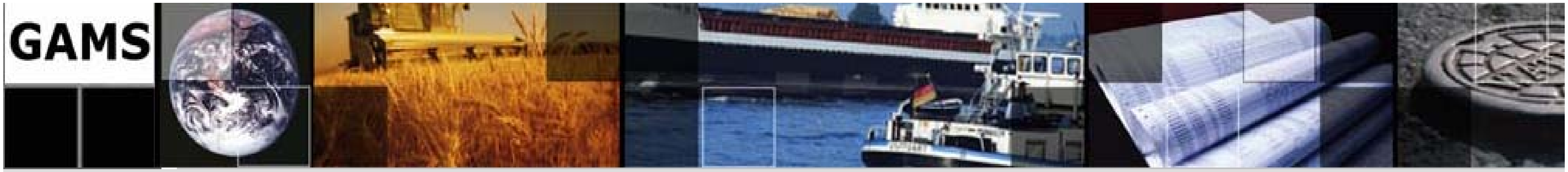
```
Variables
 xi(i)   fraction of portfolio increase,
 xd(i)   fraction of portfolio decrease,
 y(i)    binary switch for increasing current holdings of i,
 z(i)    binary switch for decreasing current holdings of i;
Binary Variables  y, z; Positive Variables xi, xd;
Equations
 xdef(i)    final portfolio definition,
 maxinc(i)  bound of maximum lot increase of fraction of i,
 mininc(i)  bound of minimum lot increase of fraction of i,
 maxdec(i)  bound of maximum lot decrease of fraction of i,
 mindec(i)  bound of minimum lot decrease of fraction of i,
 binsum(i)  restricts use of binary variables,
 turnover   restricts maximum turnover of portfolio;

xdef(i)..   x(i)   =e=  bdata(i,'old') - xd(i) + xi(i);
maxinc(i).. xi(i)  =l=  bdata(i,'umax')* y(i);
mininc(i).. xi(i)  =g=  bdata(i,'umin')* y(i);
maxdec(i).. xd(i)  =l=  bdata(i,'lmax')* z(i);
mindec(i).. xd(i)  =g=  bdata(i,'lmin')* z(i);
binsum(i).. y(i) + z(i)        =l= 1;
turnover.. sum(i, xi(i))       =l= tau;
```

GAMS Model Type: MIQCP

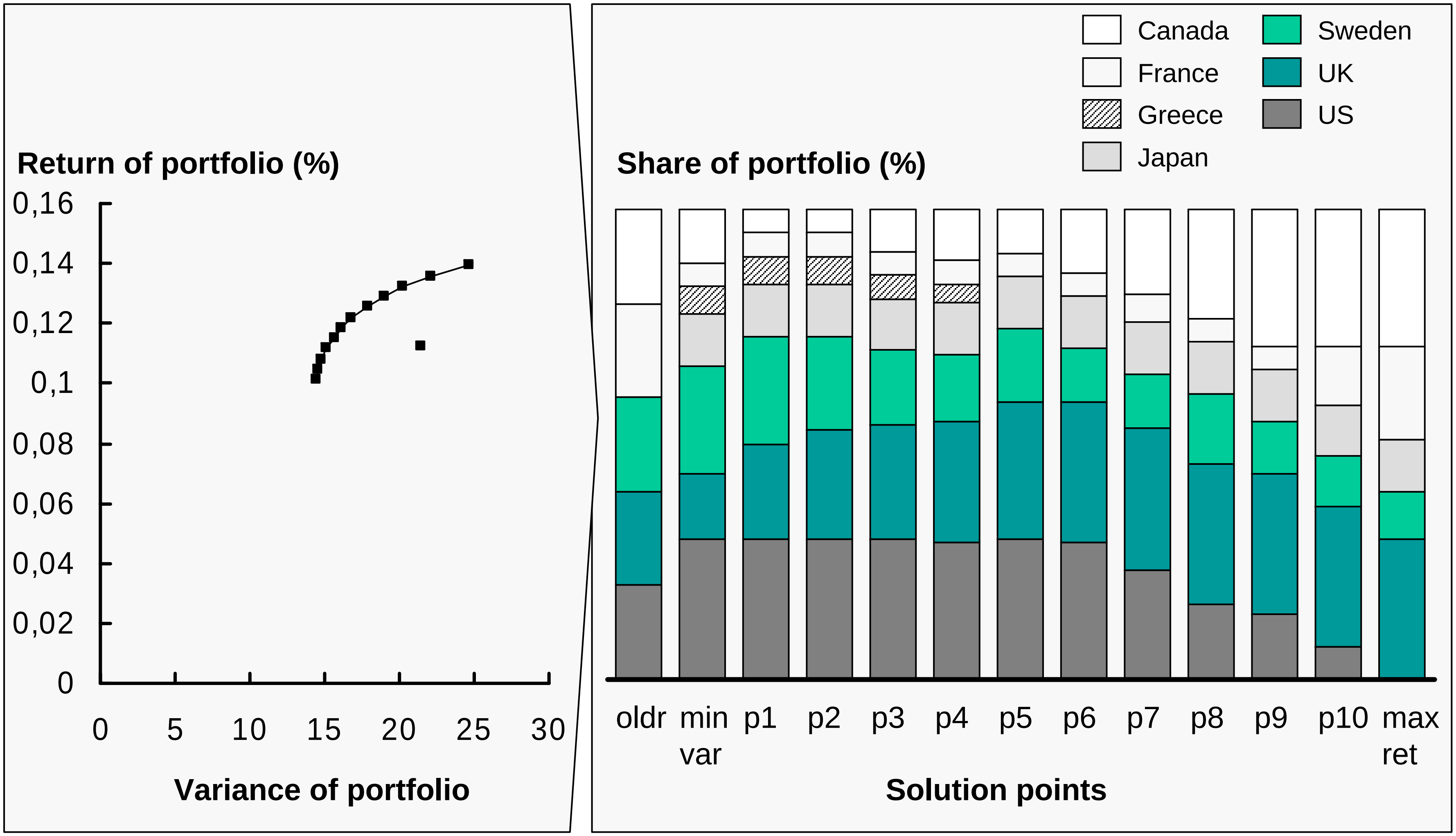


Procedural Elements

```
$gdxin data                                          # get data & setup model
$load i mu q
q(i,j) = 2*q(j,i) ;  q(i,i) = q(i,i)/2;
Model var / all / ;
set p      points for efficient frontier /minv, p1*p8, maxr/,
    pp(p) points used for loop           /      p1*p8      /;
parameter minr, maxr,rep(p,*),  repx(p,i);

# get bounds for efficient frontier
solve var minimizing v using miqcp;        #find portfolio with minimal variance
minr = r.l; rep('minv','ret') = r.l;
rep('minv','var') = v.l; repx('minv',i) = x.l(i);

solve var maximizing r using miqcp;        #find portfolio with maximal return
maxr = r.l; rep('maxr','ret')= r.l;
rep('maxr','var')=v.l;repx('maxr',i)= x.l(i);

loop(pp,                                   #calculate efficient frontier
         r.fx = minr + (maxr-minr)/(card(pp)+1)*ord(pp);
         solve var minimizing v using miqcp;
         rep(pp,'ret') =r.l;rep(pp,'var') = v.l;repx(pp,i)= x.l(i);
     );

Execute_Unload 'results.gdx',rep, repx;    # export results to GDX & Excel
Execute 'GDXXRW.EXE results.gdx par=repx rng=Portfolio!a1 Rdim=1';
Execute 'GDXXRW.EXE results.gdx par=rep  rng=Frontier!a1  Rdim=1';
```



Efficient Frontier and Portfolios (τ = 0.3)



More Theory and Templates

Theory	• **Practical Financial Optimization** (forthcoming) by S. Zenios • **A Library of Financial Optimization Models** (forthcoming) by A. Consiglio, S. Nielsen, H. Vladimirou and S. Zenios • **Financial Optimization** by S. Zenios (ed.)
Templates available online	• **GAMS Model Library:** http://www.gams.com/modlib/libhtml/subindx.htm • **Course Notes „Financial Optimization“:** http://www.gams.com/docs/contributed/financial/

GAMS

Agenda

- GAMS Development/GAMS Software
- GAMS at a Glance
- An illustrative Example: The Mean Variance Model
- **Grid Computing**

Grid Computing

Imagine…

.. you have to solve 1.000's of independent scenarios..

.. and you can do this very rapidly for little additional money…

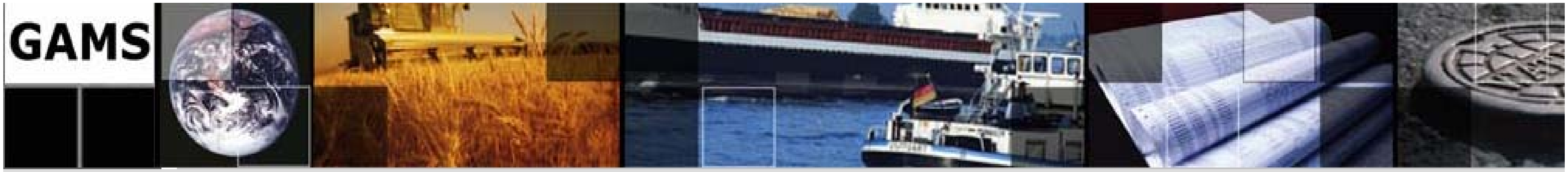
.. without having to do lots of cumbersome programming work..

Grid Computing

What is Grid Computing?

A pool of connected computers managed and available as a common computing resource

- Effective sharing of CPU power
- Massive parallel task execution
- Scheduler handles management tasks
- E.g. Condor, Sun Grid Engine, Globus
- Can be rented or owned in common
- Licensing & security issues



Advantages of Grid Computing

- Solve a certain number of scenarios faster, e.g:
  - sequential: 50 hours
  - parallel (200 CPUs): ~15 minutes

    →Cost is $100 (2$ CPU/h)
- Get better results by running more scenarios*:

#SIM	VaR error	CVaR error
1000	5.42%	6.74%
20,000	1.21%	1.49%

* http://www.tc.cornell.edu/NR/shared/Presentations/24Feb04.Garp.pdf

GAMS

Economics of Grid Computing

- Yearly cost, 2-CPU workstation: $5200
  - Hardware: $1200
  - Software: $4000
- Hourly cost on the grid: $2/cpu
  - $1/hour for CPU time (to grid operator)
  - $1/hour for software (GAMS, model owner)
- 1 workstation:
  - ~ 2600 hrs grid time or
  - ~ 50 hrs/week grid time
- Up-front vs. deferred, as-needed costs

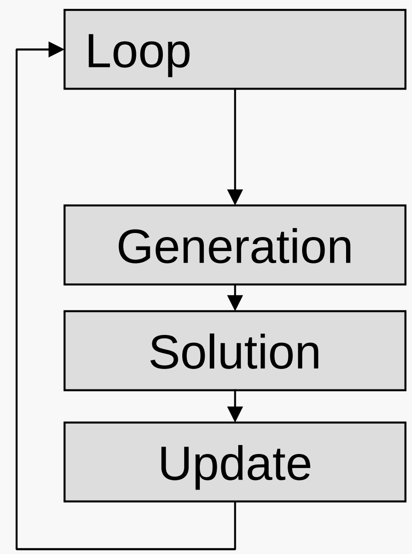
GAMS & Grid Computing

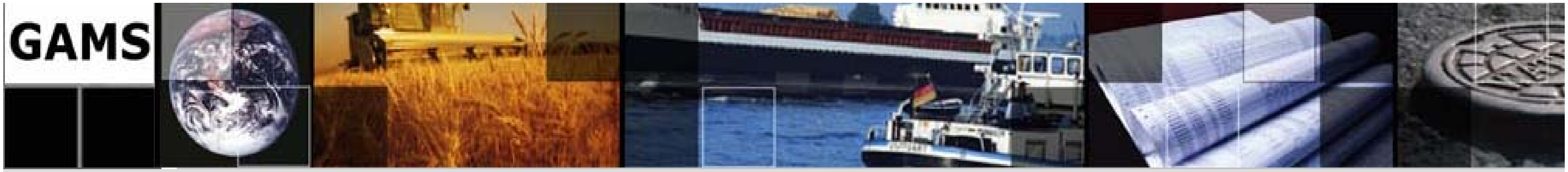
- **Scalable**:
  - support of massive grids, **but also**
  - multi-cpu / multiple cores desktop machines
  - “1 CPU - Grid”
- Platform **independent**
- Only **minor changes** to model required
- **Separation** of model and solution method
  → Model stays **maintainable**

Simple Serial Solve Loop

```
Loop(p(pp),
   ret.fx = rmin +(rmax-rmin)
     /(card(pp)+1)*ord(pp) ;
   Solve minvar min var using miqcp;
   xres(i,p)          = x.l(i);
   report(p,i,'inc') = xi.l(i);
   report(p,i,'dec') = xd.l(i)
  );
```

How do we get to parallel and distributed computing?





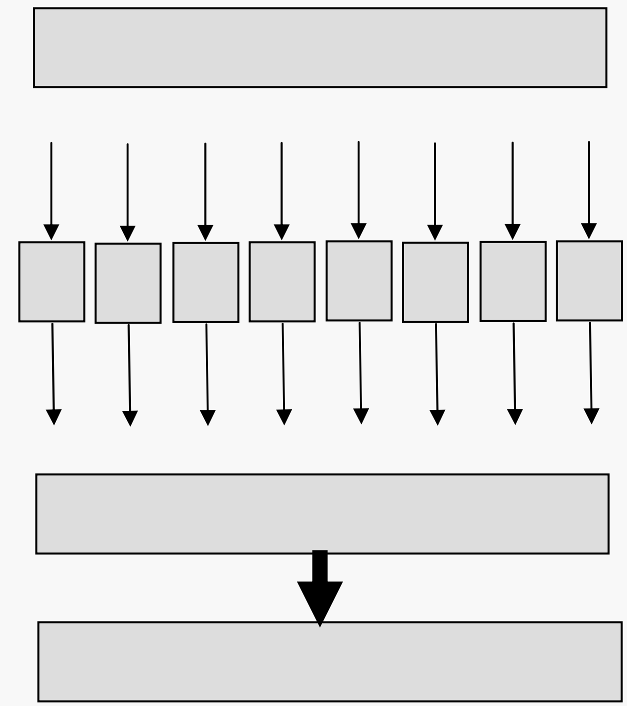
GRID specific enhancements

1. Submission of Jobs

2. "Grid Middleware"
- *Distribution of Jobs*
- *Job execution*

3. Collection of Solutions

4. Processing of results

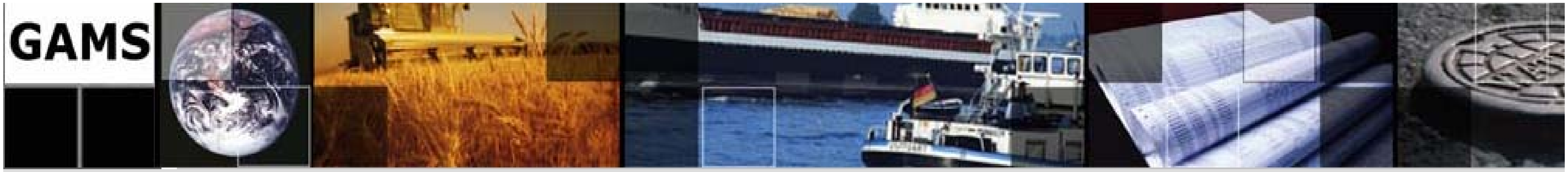


Job Submission Loop

```
Parameter h(p) store the instance handle;
minvar.solvelink = 3;        # turn on grid option
Loop(p(pp),
   ret.fx = rmin + (rmax-rmin)
   /(card(pp)+1)*ord(pp) ;
   Solve minvar min var using miqcp ;
   h(pp) = minvar.handle ); # save instance handle
```

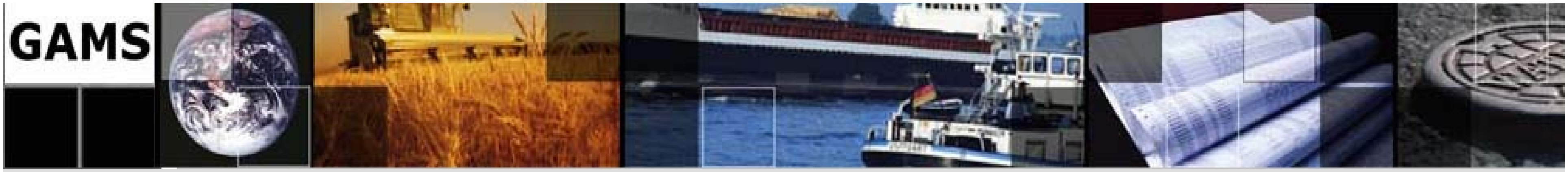
LOG:

```
...
--- LOOPS pp = p1
---   46 rows  37 columns  119 non-zeroes
---   311 nl-code  7 nl-non-zeroes
---   14 discrete-columns
--- grid_qmeanvar.gms(150) 3 Mb
--- Submitting model minvar with handle grid137000002
--- grid_qmeanvar.gms(148) 3 Mb
--- Generating MIQCP model minvar...
```



“Grid”- Middleware (PC)

```
: gams grid submission script
: arg1 solver executable
:    2 control file
:    3 scratch directory
: gmscr_nx.exe processes the solution and produces 'gmsgrid.gdx'
: note: %3 will be the short name, this is neeeded because
:       the START command cannot handle spaces or "...'
:       before we use %~3  will strip surrounding "..."
:       makes the name short
: gmsrerun.cmd will resubmit runit.cmd

echo @echo off             > %3runit.cmd
echo %1 %2                >> %3runit.cmd
echo gmscr_nx.exe %2      >> %3runit.cmd
echo mkdir %3finished     >> %3runit.cmd
echo exit                 >> %3runit.cmd

echo @start /b /BELOWNORMAL %3runit.cmd ^> nul  > %3gmsrerun.cmd
start /b /BELOWNORMAL %3runit.cmd > nul
exit
```



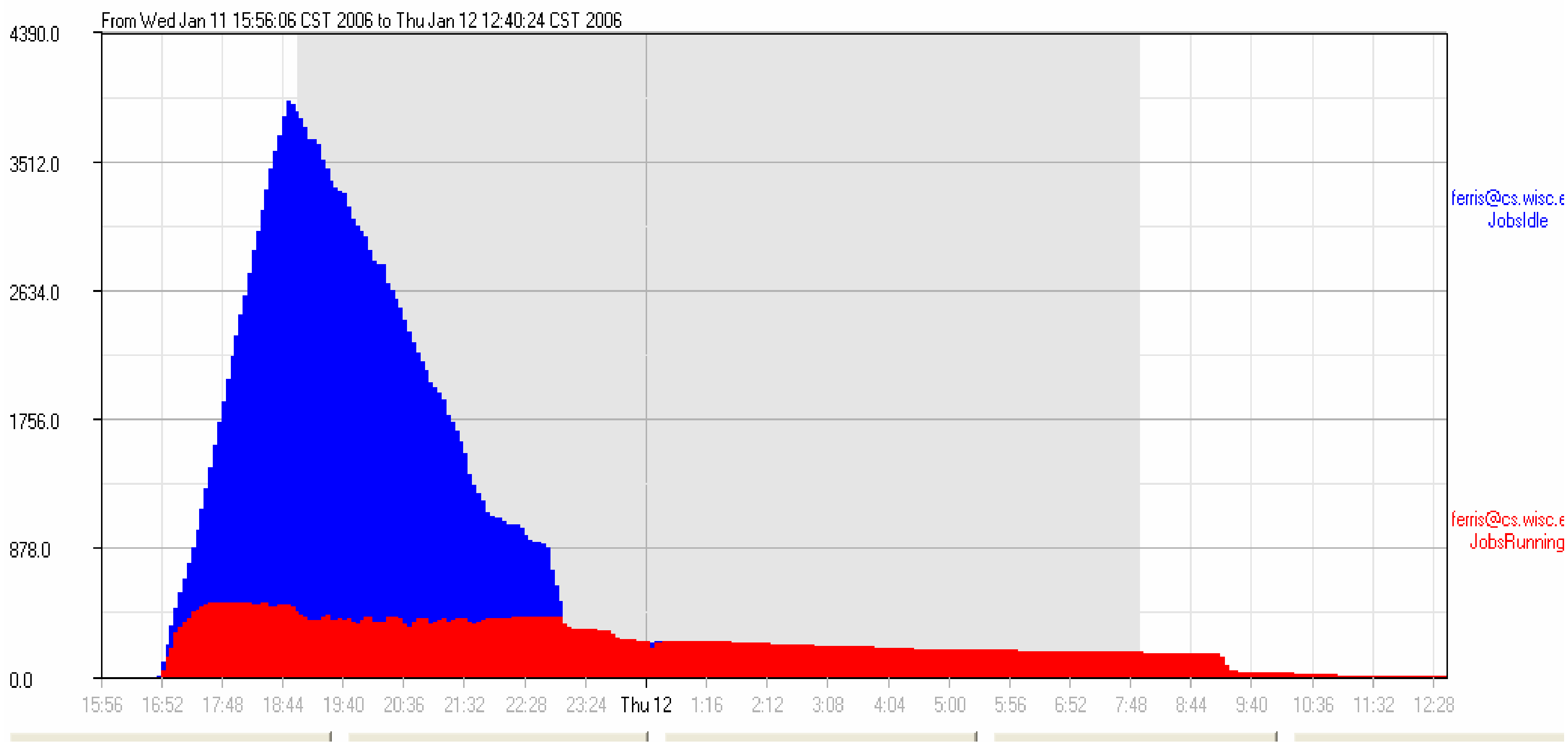
Solution Collection Loop

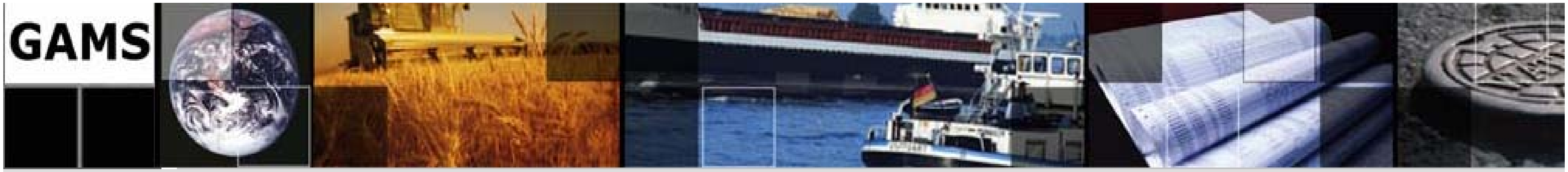
```
Repeat
  loop(p(pp)$h(p),
    if(handlestatus(h(p))=2,
      minvar.handle = h(p); execute_loadhandle minvar;
      xres(i,p)=x.l(i); report(p,i,'inc')=xi.l(i);
      report(p,i,'dec')= xd.l(i)
      display$handledelete(h(p))'Could not remove handle';
      h(p) = 0)
   );                            # indicate solution is loaded
   if(card(h), execute 'sleep 1');
until card(h) = 0 or timeelapsed > 100;
```

```
LOG: …
      --- GDXin=c:\work\mod\225b\grid137000002\gmsgrid.gdx
      --- grid_qmeanvar.gms(154) 3 Mb
      --- Removing handle grid137000002
      --- GDXin=c:\work\mod\225b\grid137000003\gmsgrid.gdx
      --- Removing handle grid137000003
      --- GDXin=c:\work\mod\225b\grid137000007\gmsgrid.gdx
      …
```

Results for 4096 MIPS on Condor Grid

- Submission started Jan 11,16:00
- All jobs submitted by Jan 11, 23:00
- All jobs returned by Jan 12, 12:40
  - 20 hours wall time, 5000 CPU hours
  - Peak number of CPU's: 500





More Developments

→ http://www.gams.com/docs/release/release.htm

Release Notes

Each new release incorporates numerous fixes and improvements to the core GAMS system and its many components. A selected list of improvements and new components is given below.

If you are interested in receiving the latest information about new GAMS releases and trying out beta releases, please subscribe to our release email list.

Distribution 22.2 Apr 21, 2006

Distribution 22.2 is a maintenance release to correct some performance issues in the GAMS system and include newly available solver libraries.

Acknowledgements

We would like to thank all of our users who have reported problems and made suggestions for improving this release. In particular, we thank Wolfgang Britz (Bonn University), Paritosh Desai (DemandTec), Michael Ferris (UW-Madison), Edgar Ramirez (at hotmail.com), and Rich Roberts (SRS Technologies).

GAMS System

- The limit on nonlinear instructions in a single block has been raised from 16 million to 64 million instructions.
- Performance improvements for very large and complicated loop structures.
- International characters in file and path names are now handled correctly.
- GAMS IDE:
  - GDX data browser is faster and can sort indices by name vs. entry order
  - A symbol shown in the GDX data browser can be written to an Excel file

Solvers

- CONOPT: New libraries are included which address minor fixes.
- CPLEX: New libraries (version 10.0.1, a maintenance release)
- LGO: New libraries
  - The built-in stochastic searches have been improved.
  - Some internal limits were increased to allow larger models to be solved.

Distribution 22.1 Mar 15, 2006

GAMS System

- *Relaxation of discrete variables (.prior=Inf):*

  The priority attribute of a discrete variable can be used to relax a specific variable instance. The priority attribute `.prior` establishes in what order variables are to be fixed to integral values while searching for a solution. Variables with a specific `.prior` value will remain relaxed until all variables with a lower .prior values have been fixed. Setting the `.prior` value to +inf will relax this variable

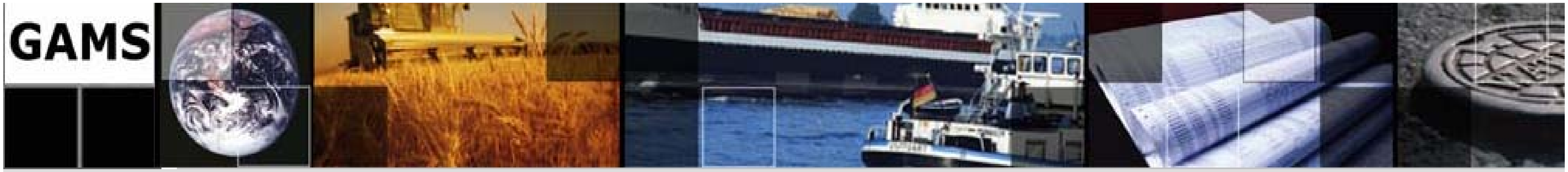
GAMS

Conclusions

- **Hardware:** Parallel computing environments are becoming available at **low cost**. *(SUN just introduced a 5.000 node network in the US giving 100 hours away for free for experiments)*
- **Software:**
  - Simple language extensions provide easy and scalable access
  - Today's modeling languages are well suited to experiment with parallel approaches for solving difficult problems

The End

Thank you!
... Questions?



Contacting GAMS

Europe:

GAMS Software GmbH
Eupener Str. 135-137
50933 Cologne
Germany
Phone: +49 221 949 9170
Fax: +49 221 949 9171
Http://www.gams.de

USA:

GAMS Development Corp.
1217 Potomac Street, NW
Washington, DC 20007
USA
Phone: +1 202 342 0180
Fax: +1 202 342 0181
Http://www.gams.com