GAMS

Grid Computing

Michael Bussieck mbussieck@gams.com

Jan-Hendrik Jagla jhjagla@gams.com

Lutz Westermann lwestermann@gams.com

GAMS Software GmbH

www.gams.de

GAMS Development Corporation

www.gams.com

Solver Technology

➔ **Tremendous algorithmic and computational progress**

- **LP** in fact only restricted by available memory
- **MIP**
  - Some small (academic) problems still unsolvable
  - Commercial problems in most case docile
- **NLP/MINLP**
  - Predictions are problem and data specific, global vs. local solutions

Multiple Threads / SMP

- **CPLEX**
  - **parallel extension** for B&B and interior point solver
  - **concurrent optimizer**
  - academic license includes 4 threads
- **MOSEK**
  - **parallel extension** for the interior solver comes free of charge
  - **concurrent optimizer**
- **XPRESS**
  - **parallel extension** for B&B and interior point solver
  - academic license includes 4 threads
- **XA (XAPAR)**

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

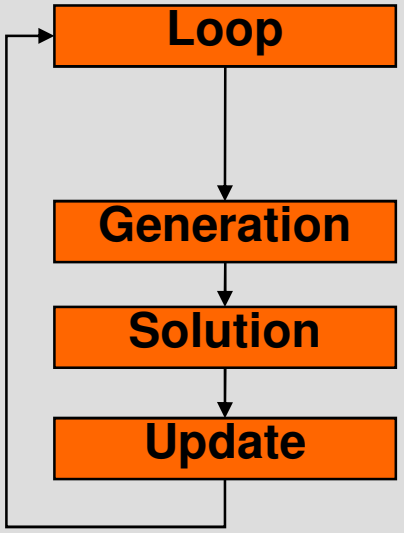
What is Grid Computing?

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

Simple Serial Solve Loop

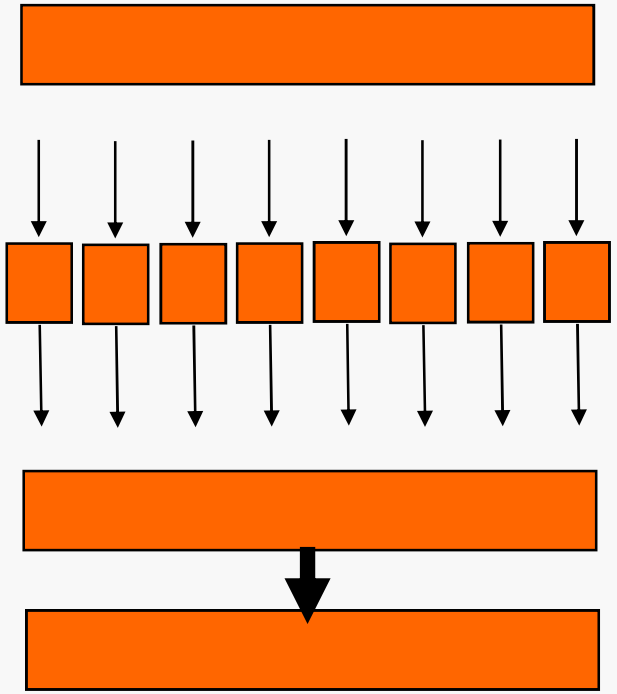
```
Loop(p(pp),
    v.fx = vmin + (vmax-vmin)
     /(card(pp)+1)*ord(pp) ;
    Solve var1 maximizing m using nlp ;
    xres(i,p)          = x.l(i);
    xres('mean',p)     = m.l;
    xres('var',p)      = v.l;
    xres('status',p)  = var1.modelstat;
);
```



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



Hands-on! 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

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	$x_i \geq 0$

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

Job Submission Loop

```
* turn on grid option
var1.solvelink=3;
Loop(p(pp),
  v.fx = vmin + (vmax-vmin)/(card(pp)+1)*ord(pp);
  Solve var1 maximizing m using nlp ;
* save instance handle
  handle(p) = var1.handle );
```

LOG

```
…
--- LOOPS pp = p1
---   3 rows  9 columns  23 non-zeroes
---   538 nl-code  7 nl-non-zeroes
--- meanvar_edited.gms(174) 3 Mb
--- Submitting model var1 with handle grid133000004
…
```

“Grid”- Middleware (PC)

```
@echo off
: 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 echo OK ^> %3finished >> %3runit.cmd
echo exit                  >> %3runit.cmd

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

exit
```

Solution Collection Loop

```
Repeat
    loop(p(pp)$handlecollect(handle(p)),
    xres(i,p)      = x.l(i); xres('mean',p)    = m.l;
    xres('var',p) = v.l;     xres('status',p) = var1.modelstat;
    display$handledelete(handle(p)) 'trouble deleting handles';
    handle(p) = 0 ) ;
    display$sleep(card(handle)*0.2) 'sleep some time';
until card(handle) = 0 or timeelapsed > 100;
xres(i,p(pp))$handle(p) = na;
```

LOG

```
…
--- meanvar_edited.gms(161) 3 Mb
--- GDXin=C:\...\225a\grid133000004\gmsgrid.gdx
--- meanvar_edited.gms(161) 3 Mb
--- Removed handle grid133000004
…
```

Hands-on! GAMS Grid Computing

gamside: C:\Documents and Settings\Jan\My Documents\presentation\2007-07-EURO-Prag\Workshop\meanvar_grid\meanvar.gpr

File Edit Search Windows Utilities Help

c:\documents and settings\jan\my documents\presentation\2007-07-euro-prag\wor...

meanvar.gms

```
   xres('var',p)      = v.l;
   xres('status',p)   = var1.modelstat;
   vmin = v.l; );

Loop(p(pp),
   v.fx = vmin + (vmax-vmin)/(card(pp)+1)*ord(pp) ;
   Solve var1 maximizing m using nlp ;
   xres(i,p)          = x.l(i);
   xres('mean',p)     = m.l;
   xres('var',p)      = v.l;
   xres('status',p)   = var1.modelstat; );

Display xres;
```

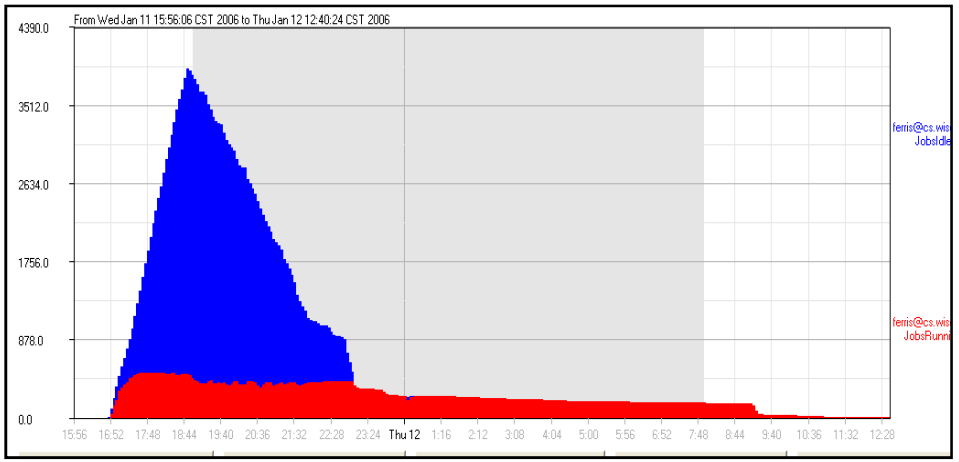
C:\Documents and Settings\Jan\My Documents\presentation\2007-07-EURO-Prag\Wo...

meanvar.gms | meanvar_edited.gms

```
   xres('var',p)      = v.l;
   xres('status',p)   = var1.modelstat;
   vmin = v.l; );

$if not set grid $set grid 0
parameter handle(p) Grid handle;

if(not %grid%,
   Loop(p(pp),
      v.fx = vmin + (vmax-vmin)/(card(pp)+1)*ord(pp)
      Solve var1 maximizing m using nlp ;
      xres(i,p)          = x.l(i);
      xres('mean',p)     = m.l;
      xres('var',p)      = v.l;
      xres('status',p)   = var1.modelstat; );
else
   var1.solvelink=3;
   Loop(p(pp),
      v.fx = vmin + (vmax-vmin)/(card(pp)+1)*ord(pp)
      Solve var1 maximizing m using nlp ;
      handle(p) = var1.handle );
   Repeat
      loop(p(pp)$handlecollect(handle(p)),
         xres(i,p)          = x.l(i);
         xres('mean',p)     = m.l;
         xres('var',p)      = v.l;
         xres('status',p)   = var1.modelstat;
         display$handledelete(handle(p)) 'trouble del
         handle(p) = 0 ) ;
      display$sleep(card(handle)*0.2) 'sleep some tim
   until card(handle) = 0 or timeelapsed > 100;
   xres(i,p(pp))$handle(p) = na;
);
Display xres;

execute_unload "portfolio.gdx" xres;
```

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**

Sources of GAMS Information

Download: http://download.gams-software.com/
Release Notes: http://www.gams.com/docs/release/release.htm
Contributed Documentation: http://www.gams.com/docs/contributed
Contributed Software: http://www.gams.com/contrib/contrib.htm
Presentations: http://www.gams.com/presentations
Workshops: http://www.gams.com/courses.htm

Bruce McCarl's Newsletter: http://www.gams.com/maillist/newsletter.htm
GAMS User Group: http://www.gams.com/maillist/gams_l.htm
GAMS Google Group: http://groups.google.de/group/gamsworld

Other relevant sites on the Web: http://www.gams.com/hotlinks.htm

Contacting GAMS

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

sales@gams.com
support@gams.com

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

info@gams.de
support@gams-software.com