GAMS, Condor and the Grid

Michael C. Ferris
University of Wisconsin

Michael Bussieck, Alex Meeraus
GAMS Corp.
Use of Grid Computation in Optimization

- Aid search for global solutions (typically in non-convex or discrete setting)
  - Pattern search, evolutionary algorithms
  - Branch and bound/cut
- Treat uncertainty (sampling)
- Enhance speed of computation
  - Decomposition approaches
    - Splitting, Benders, Dantzig-Wolfe, Lagrangian
  - Linear algebra
Assumptions for talk

- Hard Optimization Problem to solve
- Access to modeling system
  - GAMS
  - AMPL
- Access to Grid computing
  - Condor
  - Sun N6 Grid Engine, Globus
- How to use Grid effectively to solve problem already modeled
- Extend existing solution methodologies...
Can we use it effectively?

- High throughput not high performance computing (modify perspective)
- New modeling features of GAMS facilitate use of grid computation and sophisticated solvers
- Optimization expertise shared with computational engines
Transportation model

\[ \begin{align*}
\min_{x} \quad & z = \sum_{(i,j)} c_{i,j} x_{i,j} \\
\text{st} \quad & \sum_{j} x_{i,j} \leq a_i, \quad \forall i \\
& \sum_{i} x_{i,j} \geq b_j, \quad \forall j
\end{align*} \]
b(j) = dem(j);
solve transport min z using lp;
report = z.l;
Understand “solve” statement

\[
b(j) = \text{dem}(j);
\]

solve transport min z using lp;

report = z.l;

Model

create

Problem Instance

eexecute solver

Solution File

load

Model

Foo.gms

Foo.mod

225a/*

Foo.nl

225a/gamssol.scr

Foo.sol
Typical Application for GAMS

loop(s,
    b(j) = dem(s,j)
    solve transport min z using lp;
    report(s) = z.l;
)
;
Typical Application for GAMS

\[
\begin{align*}
\text{loop}(s, & \\
& b(j) = \text{dem}(s,j) \\
& \text{solve transport min } z \text{ using lp;} \\
& \text{report}(s) = z.l; \\
) ;
\end{align*}
\]

Need notion of a handle
Typical Application for GAMS & Grid

```plaintext
transport.solvelink = 3;       // turn on grid option

loop(s,
   b(j) = dem(s,j)
   solve transport min z using lp;
   h(s) = transport.handle ;    // save instance handle
)

repeat
   loop(s$handlecollect(h(s)),
      report(s) = z.l;
      h(s) = 0 ) ;    // indicate that we have loaded the solution
      display$sleep(card(h)*0.2) 'was sleeping for some time';
   until card(h) = 0 or timeelapsed > 10;
```
Demonstration (source setit)

- cp -r /scratch/COPTA/gams/* .
- cd gams
- gams trnsgrid
- setenv USECONDOR Inx
- gamskeep trnsgrid
- condor_q
- mkdir gdir
- gams trnsgrid gdir=gdir
Demonstration (2)

- `gams trnsspawn gdir=gdir s=T`
- `condor_q`
- `condor_q`
- `...`
- `gams trnscollect gdir=gdir r=T`
Demonstration (3)

- setenv USECONDOR mw
- setenv MWWORKERS 2
- gams trnsspawn gdir=gdir s=T
- condor_q
- condor_q
- ...
- gams trnscollect gdir=gdir r=T
Exercises

- Change number of scenarios
- Change solver from default
- Run qmeanvarg example
  - use --maxtime=1000 option
  - use more points --points=5
  - detach spawn from collect
- Run the danwolfe example
  - use mw and the same gdir
- Run jacobi example
Solution hints

- option lp=xpress;
- gams qmeanvarg --maxtime=1000 gdir=grid -points=5 s=T
- gams danwolfe gdir=gdir
- gams jacobi gdir=gdir
  --vars=400 --parts=8
Multiple Solvers/Platforms

- Can use all supported solvers including:
  - CPLEX, XPRESS, PATH, SNOPT, MOSEK
- Runs on multiple platforms using heterogeneous machines for solvers
- Can interleave solutions on host and worker
- Available right now!
Clean up mw!

- Either `condor_rm` your mw “server” job
- Or “`mv gdir foo; sleep 20; mv foo gdir`