School of Mathematics

How to solve QPs with 10^9 variables

Andreas Grothey, Jacek Gondzio

OOPS (Object Oriented Parallel Solver)

- Mantra: "Truly large scale problems are not only sparse but structured" (due to e.g. dynamics, uncertainty, spatial distribution etc.)
- Exploiting structure is key to building efficient IPMs for large problems:
  - Faster linear algebra
  - Reduced memory use
  - Possibility to exploit (massive) parallelism
  - **We assume that structure is known!** $\Rightarrow$ no automatic detection.
- OOPS currently solves LP/QP problems.
- Simple sequential-QP scheme solves nonlinear ALM models

Linear Algebra of IPMs

Main work: solve

$$\underbrace{\begin{bmatrix} -Q-\Theta & A^\top \\ A & 0 \end{bmatrix}}_{\Phi} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} r \\ h \end{bmatrix}$$

for several right-hand-sides at each iteration

$\Rightarrow$ Two stage solution procedure

- factorize $\Phi = LDL^\top$
- backsolve(s) to compute direction $(\Delta x, \Delta y)$ + corrections

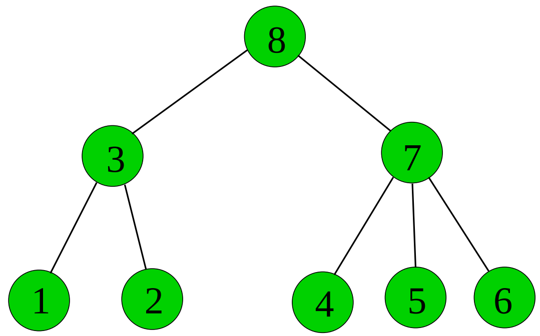
$\Rightarrow$ Φ changes numerically but not structurally at each iteration

Key to efficient implementation is exploiting structure of Φ in these two steps

OOPS: (Block) Elimination Trees:

Elimination tree orders rows/columns for elimination with minimum fill-in:

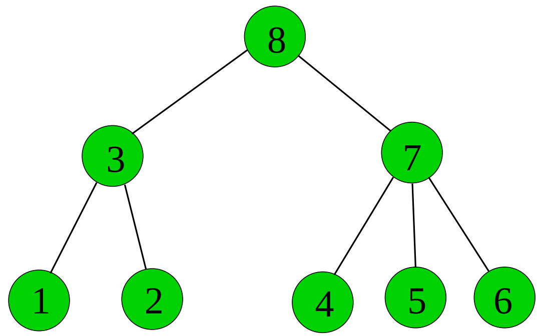
$$\begin{bmatrix} x & & x & & & & & x \\ & x & x & & & & & x \\ x & x & x & & & & & x \\ & & & x & & & x & x \\ & & & & x & & x & x \\ & & & & & x & x & x \\ & & & x & x & x & x & x \\ x & x & x & x & x & x & x & x \end{bmatrix}$$



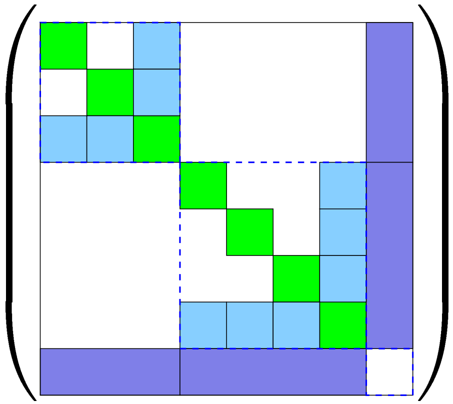
OOPS: (Block) Elimination Trees:

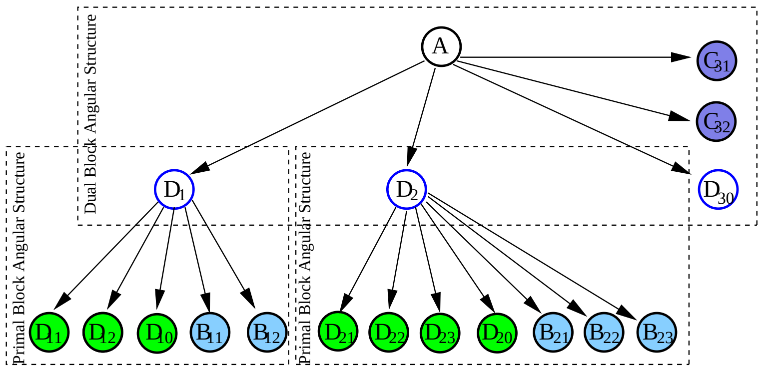
Elimination tree orders rows/columns for elimination with minimum fill-in:

$$\begin{bmatrix} x & & x & & & & & x \\ & x & x & & & & & x \\ x & x & x & & & & & x \\ & & & x & & & x & x \\ & & & & x & & x & x \\ & & & & & x & x & x \\ & & & x & x & x & x & x \\ x & x & x & x & x & x & x & x \end{bmatrix}$$



Elimination Tree can be extended to Block Elimination Tree

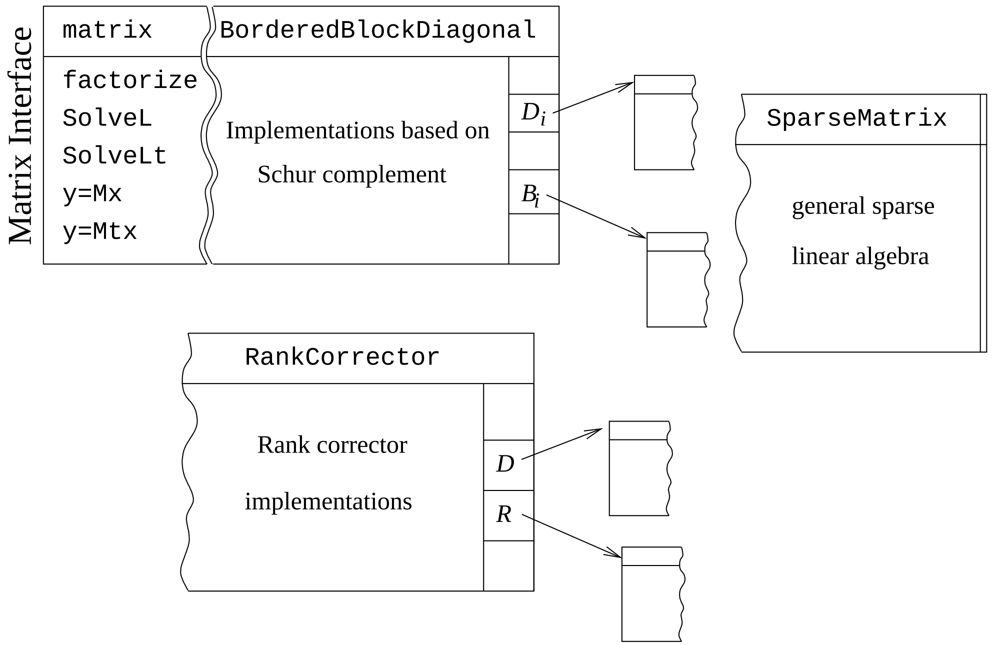




$\Rightarrow$ Organisation of linear algebra, Parallelism

OOPS: Object-oriented linear algebra implementation

- Every node in *block elimination tree* has own linear algebra implementation (depending on its type)
- Implementation is realisation of an abstract linear algebra interface.
- Different implementations for different structures are available.



⇒ Rebuild *block elimination tree* with matrix interface structures

Application: Asset and Liability Management Problem - ALM

- A set of assets $\mathcal{J} = \{1, \ldots, J\}$ is given (e.g. bonds, stock, real estate).
- At every stage $t = 0, \ldots, T-1$ we can buy or sell different assets.
- The return of asset j at stage t is *uncertain* (but distribution is known).

We have to make investment decisions: what to buy or sell, at which time stage

Objectives:

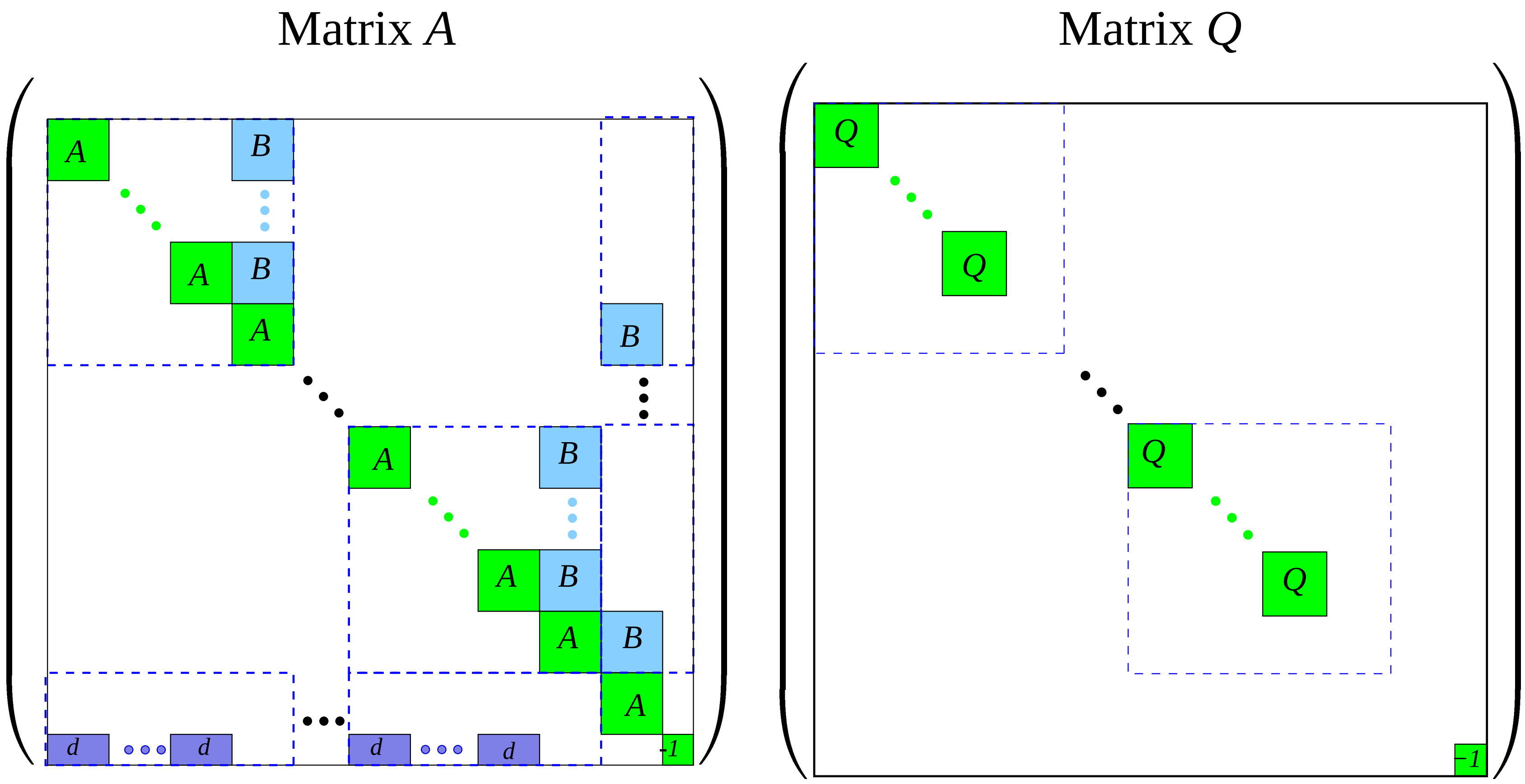
- maximize the final wealth
- minimize the associated risk

$\Rightarrow$ Mean Variance formulation:
$$\max \mathbb{E}(X) - \rho \mathrm{Var}(X)$$

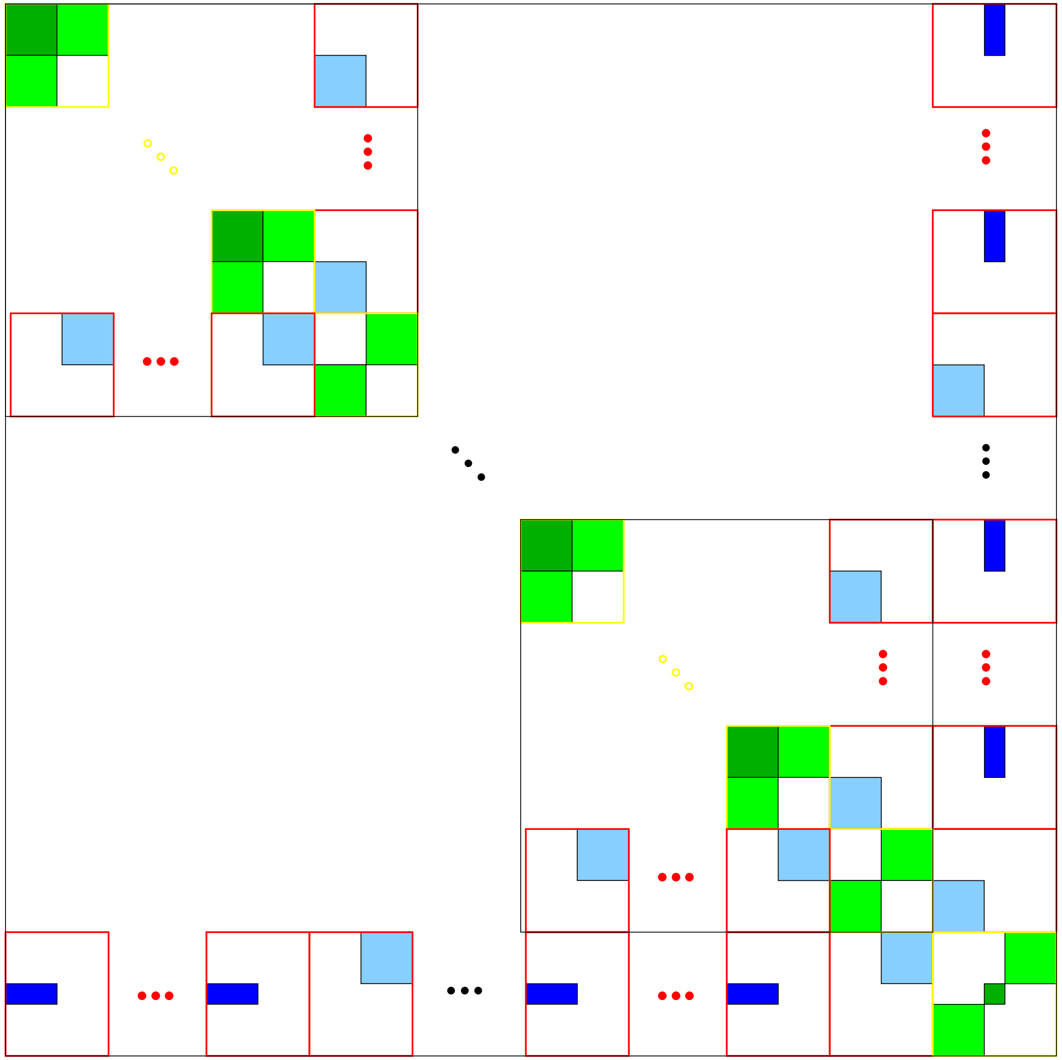
$\Rightarrow$ Stochastic Program:

- Can formulate deterministic equivalent problem
- standard QP, but huge

ALM: Structure of matrices A and Q:

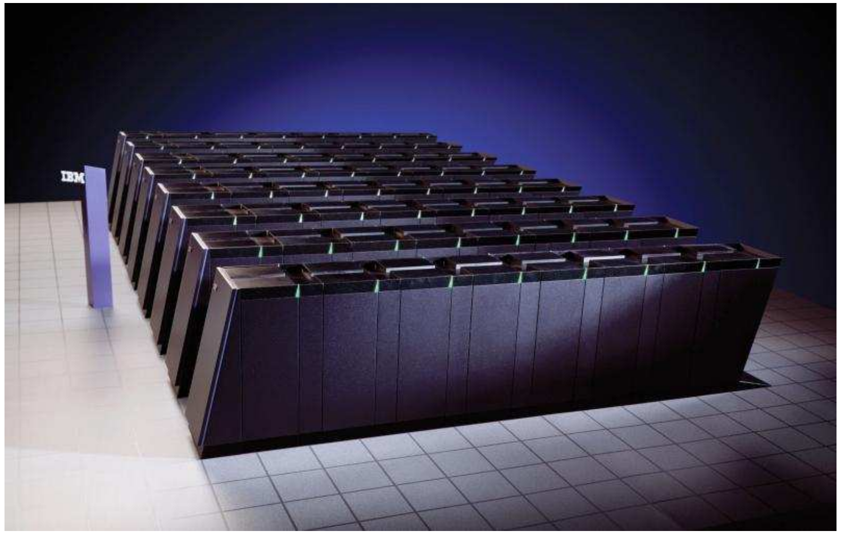


ALM: Structure of Augmented System matrix:



ALM: Largest Problem attempted

- Optimization of 21 assets (stock market indices) over 7 time stages.
- Using multistage stochastic programming
  Scenario tree geometry: 128-30-16-10-5-4 $\Rightarrow$ 16 million scenarios.
- Scenario Tree generated using geometric brownian motion.
- $\Rightarrow$ 1.01 billion variables, 353 million constraints



BlueGene (Edinburgh, Scotland)

- 2048 Processors
- 0.7GHz, 256Mb
- 4.7 TFlops
- #64 in `top500.org` list

HPCx (Daresbury, England)

- 1600 IBM Power-4 Processors
- 1.7GHz, 800Mb
- 6.2 TFlops
- #45 in `top500.org` list

Issues for Massive Parallelism

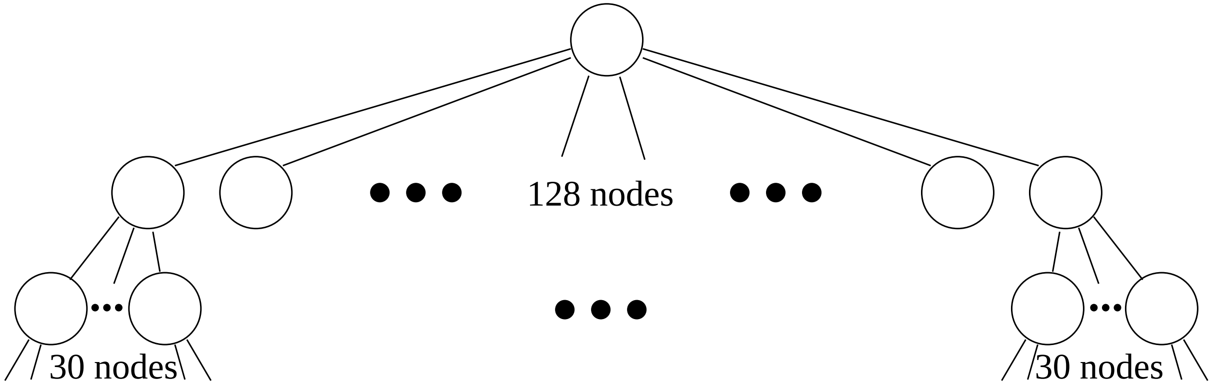
- Sparsity of multilevel linear Algebra
- Memory Management

- Sparsity of multilevel linear Algebra
- Memory Management

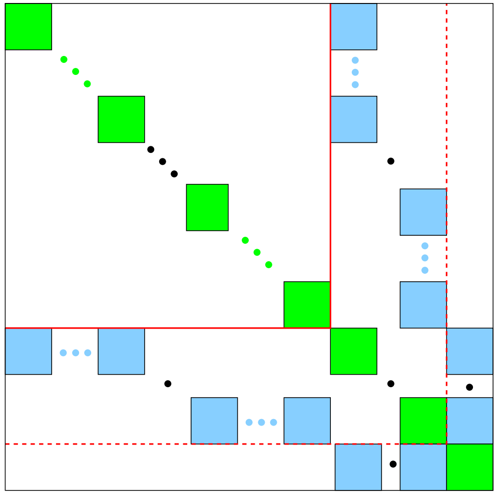
Sparsity of Linear Algebra I

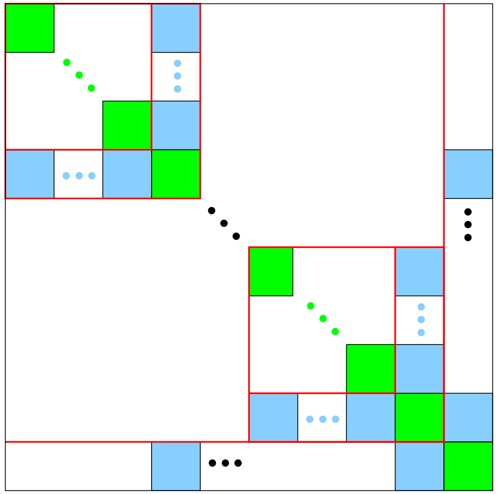
- In ALM problems matrices up to ≈ 500.000 – 1.000.000 variables can be treated as unstructured sparse matrices
- Problem has:
  - 128 first level nodes with 10.000.000 variables each.
  - 3840 second level nodes with 350.000 variables each.

⇒ need to decompose problem at second level
(with 1280 processors ⇒ 3 blocks per processor)



Sparsity of Linear Algebra II

-  ⇒
  - $63 + 128 \times 63 = 8127$ columns for Schur-complement
  - Prohibitively expensive

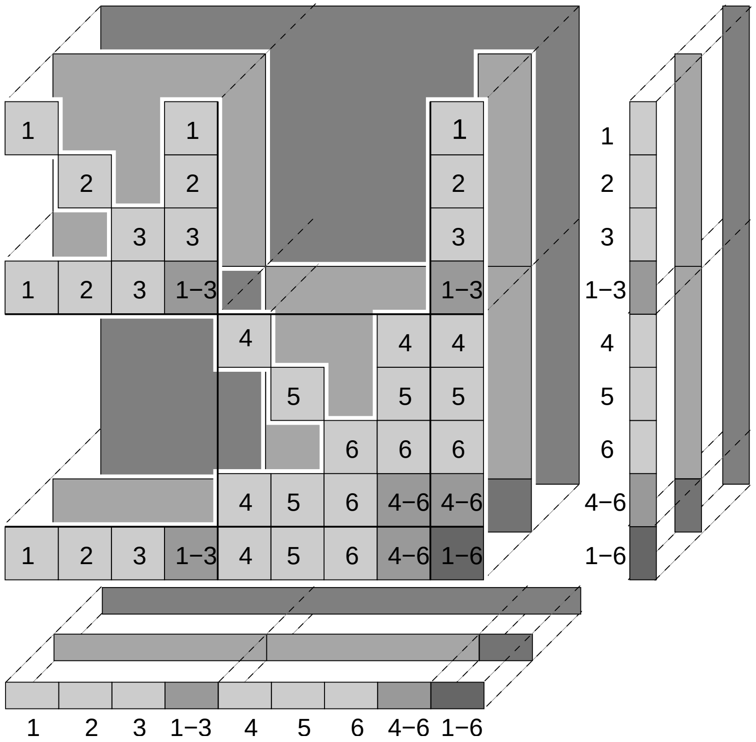
-  ⇒
  - Need facility to exploit nested structure
  - Need to be careful that Schur-complement calculations stay sparse on second level

Memory Management

- Data for problem requires $>$ 1GB of memory.
  $\Rightarrow$ need to split information between processors
- To each node in block-elimination tree a set of processors is assigned
- Linear Algebra is implemented so that processors communicate when needed

Distribution of leading matrix blocks among processors implies

- Distribution of subordinate blocks
- Distribution of row/column vector contributions



Results (ALM: Mean-Variance QP formulation):

Problem	Stages	Blk	Assets	Scenarios	Constraints	Variables	iter	time	procs	machine
ALM8	7	128	6	12.831.873	64.159.366	153.982.477	42	3923	512	BlueGene
ALM9	7	64	14	6.415.937	96.239.056	269.469.355	39	4692	512	BlueGene
ALM10	7	128	13	12.831.873	179.646.223	500.443.048	45	6089	1024	BlueGene
ALM11	7	128	21	16.039.809	352.875.799	1.010.507.968	53	3020	1280	HPCx

Future Work on OOPS:

- l_2-SQP scheme with warmstarting strategies
- Link to a structure conveying modeling language (SMPS, MPS/SET is supported)
- Implement other structures/other strategies (Iterative solver etc)

Object-Oriented Parallel Solver (OOPS):

`http://www.maths.ed.ac.uk/~gondzio/parallel/solver.html`

References:

- J. Gondzio and R. Sarkissian, *Parallel interior point solver for structured linear programs*, **Mathematical Programming** 96 (2003) pp 561–584.
- J. Gondzio and A. Grothey, *Parallel interior point solver for structured quadratic programs: application to financial planning problems*, Tech. Rep. MS-03-001, School of Maths, University of Edinburgh, April 2003 (to appear in EJOR).
- J. Gondzio and A. Grothey, *Solving nonlinear portfolio optimization problems with the primal-dual interior point method*, Tech. Rep. MS-04-001, School of Maths, University of Edinburgh, May 2004.
- J. Gondzio and A. Grothey, *Exploiting Structure in Parallel Implementation of Interior Point Methods for Optimization*, Tech. Rep. MS-04-004, School of Maths, University of Edinburgh, Dec 2004.

Papers available from:
`http://www.maths.ed.ac.uk/ERGO/group/`