Cyberinfrastructure and Optimization

Robert Fourer

Industrial Engineering & Management Sciences
Northwestern University, Evanston, IL, USA

AMPL Optimization LLC

4er@northwestern.edu — 4er@ampl.com

OR2010 — International Conference on Operations Research

Semi-Plenary Session WC-18

Munich, Germany, 1 September 2010

Topics

Concepts

- Infrastructure & cyberinfrastructure
- Software as a service

Optimization services

- COIN-OR open-source software
- The NEOS Server
- The OS project
- Intelligent optimization systems

Cyberinfrastructure for optimization

- Advanced computing resources
- Cloud computing

What is an infrastructure?

Familiar examples

- Transportation: road, rail, air
- Energy: power grids, pipelines
- Communications: mail, packages, phone

Common characteristics

- Doesn't *do* things; rather, *makes things possible*
- Facilitates activities *not initially foreseen*
- Relies on *standards*
- Requires broad *accessibility*

What is a cyberinfrastructure?

Infrastructure for computing

- Internet
- Web
- Supercomputer centers

Infrastructure for computing in some field

- Nanotechnology
- Seismology
- Optimization . . .

What is cyberinfrastructure research?

Ask the (U.S.) NSF . . .

- National Science Foundation
  Blue-Ribbon Advisory Panel on Cyberinfrastructure
- *Revolutionizing Science and Engineering Through Cyberinfrastructure*
  - `www.nsf.gov/od/oci/reports/toc.jsp`

Organization

- Office of Cyberinfrastructure
- Independent of programs in traditional areas

Workshops

- Over 30 reports
  - `www.nsf.gov/od/oci/reports.jsp`
- *An Operations Cyberinfrastructure: Using Cyberinfrastructure and Operations Research to Improve Productivity in American Enterprises*
  - `http://www.optimization-online.org/OCI/OCI.pdf`

Implications for Optimization Research

CI Focus at government agencies

- Big science
- Custom-built, specialized software
- Focused application

CI Focus in the optimization community

- Small science
- General-purpose software
- Broad distribution

Special circumstances of optimization

- Independence of modeling software, solver software, data
- Sensitivity of solver choice to mathematics of problem
- Huge variation in solver performance

Software as a Service ("SaaS")

as a Product

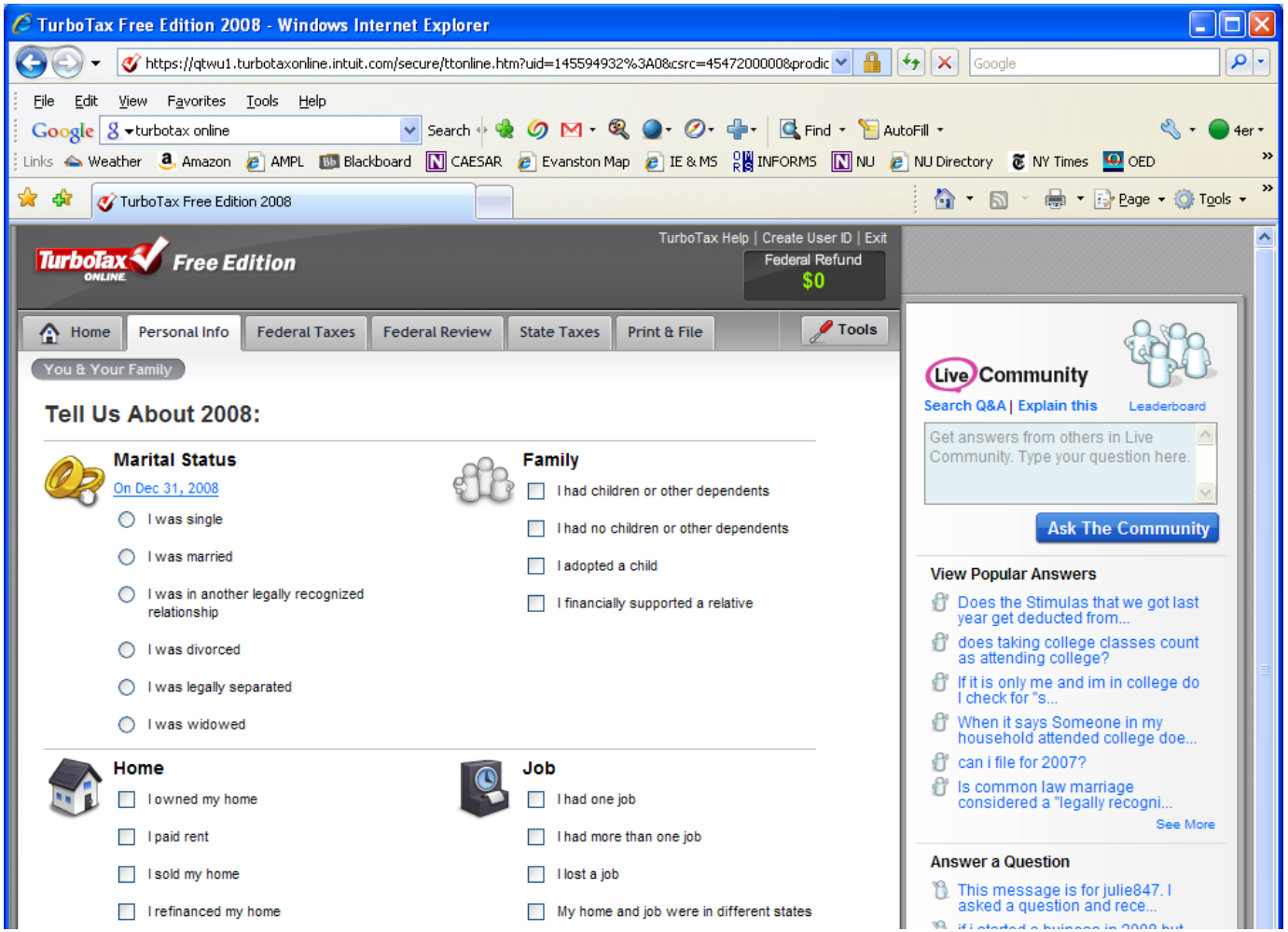
- You install & run on local computers
- You maintain & upgrade

as a Service

- They install on their servers
- You run on their servers
- They maintain & upgrade

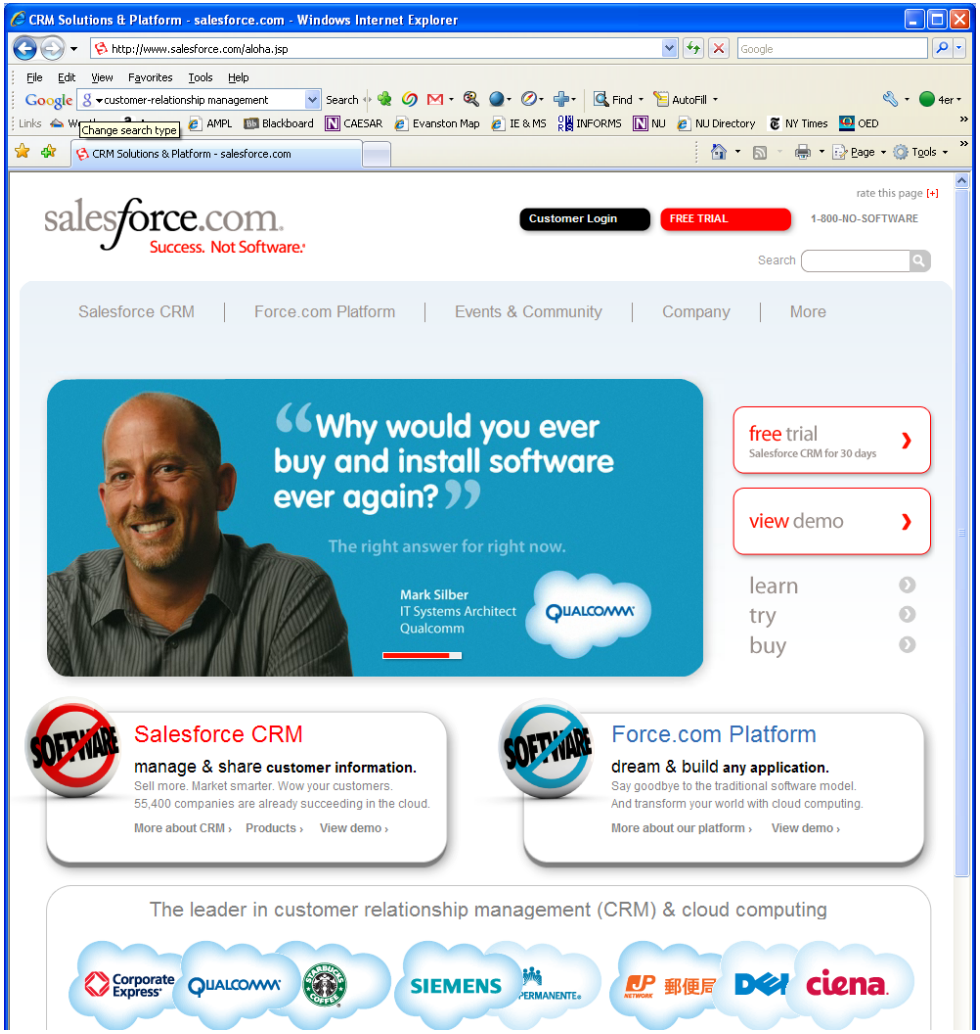
Software as a Service *(examples)*

Tax preparation



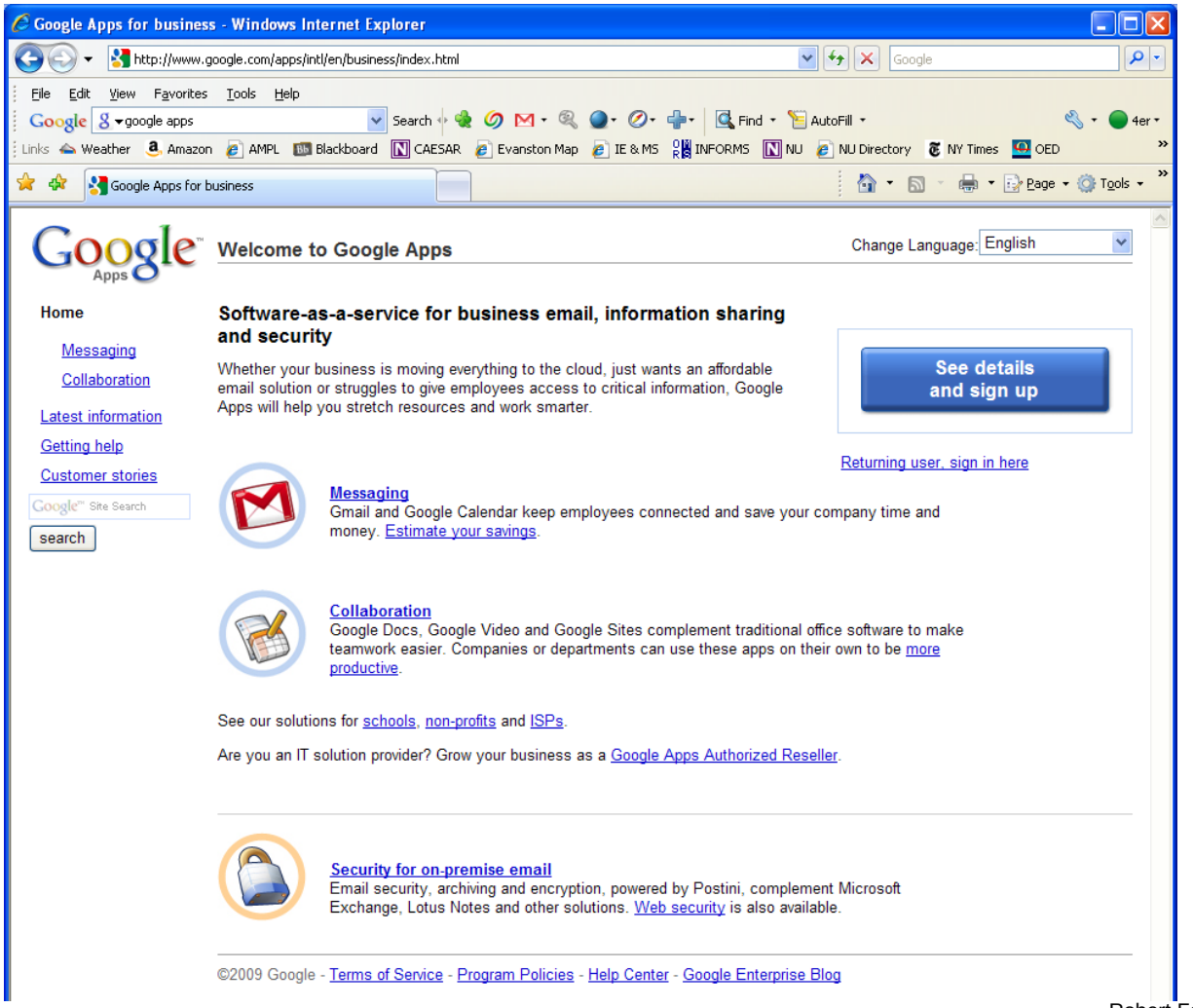
Software as a Service *(examples)*

Customer-relationship management



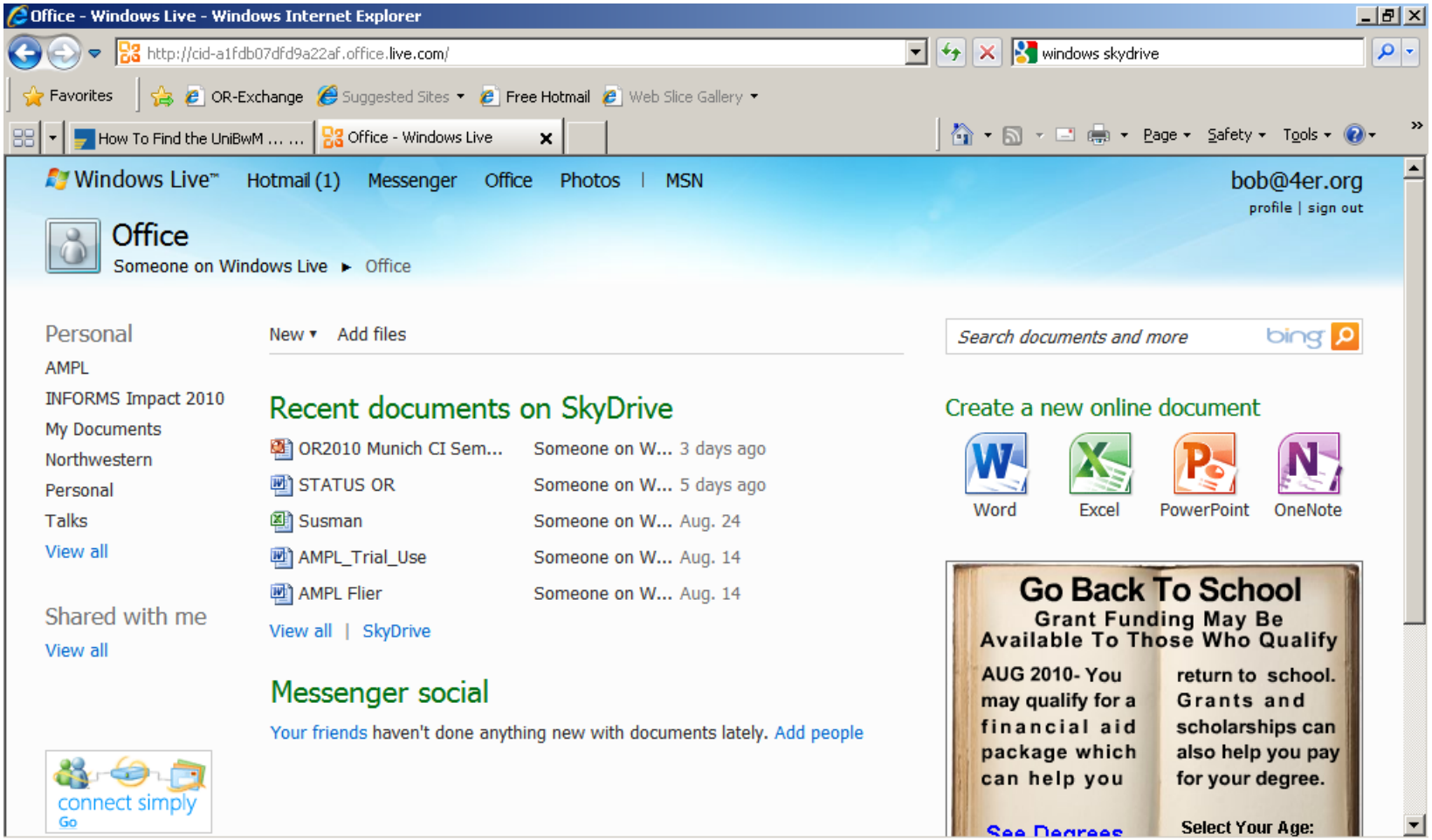
Software as a Service *(examples)*

Communication



Software as a Service *(examples)*

Office



Implications for Optimization

Menu of products

- Solvers
- Modeling systems

. . . independent and interchangeable

Potential impact

- Making products more readily available
- Facilitating comparison of products
- *Providing access to advanced computing resources?*

Progress in Optimization Services

COIN-OR open-source software

The NEOS Server

The OS Project

Intelligent optimization systems

COIN-OR www.coin-or.org

Computational Infrastructure for Operations Research

- Repository for open-source software for optimization
- Operated by nonprofit COIN-OR Foundation
  - Origins at IBM (2000)
  - Currently hosted by INFORMS
- Mission:
  - Develop, manage & distribute
  - OR software, models, and data so that
  - OR professionals can benefit from
  - peer-reviewed, archived, openly-disseminated software
- Strongest in optimization

Relevance to cyberinfrastructure

- Tools are open for any use
- Each tool has many potential applications
- Presentation, access, maintenance are standardized

Open-Source Software

Things to know

- Free, but subject to licensing restrictions
- Licenses vary considerably
- Equally available to all user classes
- Possibly owned (in part) by
  - Co-authors
  - Employer
  - Granting Agencies
  - Owner of the machines it was written on

Examples

- GNU Public License, GNU Library Public License
- Eclipse Public License / Common Public License
- Mozilla Public License
- Apache License
- BSD Licenses

Open-Source Software at COIN-OR

Solvers

- Linear continuous & discrete
- Nonlinear continuous & discrete
- Semidefinite
- Stochastic

. . . source & binary

Infrastructures

- Developer tools
- Optimization utilities
- Interfaces
- Modeling systems and environments

. . . more later on Optimization Services

Getting Published on COIN-OR

Scope

- Useful to Operations Research professionals
- Not clearly better suited to another repository

Requirements

- Working Code
- Documentation
- Legal

Operational Policies

COIN-OR Requirements

Working code

- The project has some type of "tests"
- The tests are in a form that can be easily supplemented
- No other specific requirements on form or magnitude

. . . tests ensure the code remains working as it evolves

COIN-OR Requirements *(cont'd)*

Documentation

- Authors
- Install
  - How to build, install, and run on at least one platform
  - Any platform, any computing language
- Readme (FAQs)
  - What is the project name?
  - What is the project for?
  - Who is the maintainer?
  - How do I contact the maintainer?
  - How do I run the project once it's built or installed?
  - How do I report a bug?
- Copying
  - Licensing information

COIN-OR Requirements *(cont'd)*

Legal (new format)

- License
  - Any certified open-source license (see `www.opensource.org`)
  - Common/Eclipse Public License recommended *(IBM origins)*
- Required paperwork
  - *List of authors*
  - *Committers' Statements of Respect for Ownership*
  - *Contributors' Statement of Ownership and Licensing*
- Recommended paperwork
  - *Owner's Confirmation of Licensing*
  - *Contributors' Statements of Respect for Ownership*
- Optional paperwork
  - *List of authors & their contributions*
  - *Committers' permissions from employers*
  - *Detailed contributor & contribution logs*

NEOS neos.mcs.anl.gov

Network Enabled Optimization System

- Server
  - free Internet access to over 60 solvers
- Guide
  - tutorials, case studies, test problems, FAQs

Relevance to cyberinfrastructure

- Facilitates many uses
  - but does not anticipate particular applications
- Makes all solvers open to anyone
- Supports standard problem representations
- Uses Web standards & protocols
  - XML files
  - callable via XML-RPC

NEOS Server

A general-purpose optimization server

- Over 45 solvers in all
  - Linear, linear network, linear integer
  - Nonlinear, nonlinear integer, nondifferentiable & global
  - Stochastic, semidefinite, semi-infinite, complementarity
- Commercial as well as experimental solvers
- Central scheduler with distributed solver sites
- 20,000 submissions in a typical month

. . . has handled over 100,000

A research project

- Currently free of charge
- Supported at Argonne National Laboratory since 1996
- ***Planned move*** to Wisconsin Institutes of Discovery in late 2010

Design

Flexible architecture

- Central controller and scheduler machine
- Distributed solver sites

Standard formats

- Low-level formats: MPS, SIF, SDPA
- Programming languages: C/ADOL-C, Fortran/ADIFOR
- High-level modeling languages: AMPL, GAMS

Varied submission options

- E-mail
- Web form
- Direct call via XML-RPC
  - from AMPL or GAMS client
  - from user's client program using NEOS's API

. . . server processes submissions of new solvers, too

Range of Solvers

For familiar problem types

- Linear programming
- Linear network optimization
- Linear integer programming
- Nonlinear programming
- Stochastic linear programming
- Complementarity problems

For emerging problem types

- Nondifferentiable optimization
- Semi-infinite optimization
- Global optimization
- Nonlinear integer programming
- Semidefinite & 2nd-order cone programming

. . . virtually every published semidefinite programming code

Contributors

Open-source solvers

- GLPK, CBC, Bonmin (mixed integer)

Research solvers

- BonsaiG (mixed integer)
- FILTER, LANCELOT, LOQO, MINOS, SNOPT (nonlinear)

Commercial solvers

- Xpress, FortMP (mixed integer)
- CONOPT, KNITRO, MOSEK (nonlinear)

Commercial modeling languages

- AMPL
- GAMS

Hosts

Varied workstations at

- Aachen University of Technology, Germany
- Argonne National Laboratory
- Arizona State University
- Lehigh University
- National Taiwan University
- Universidade do Minho, Portugal
- University of Wisconsin at Madison

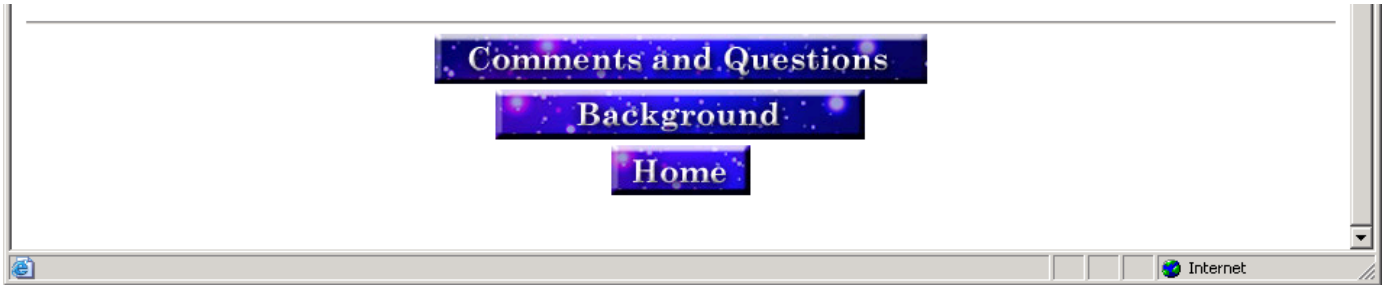
. . . new hosts are readily added anywhere on the Internet

Support

Large mailing list for questions

- NEOS developers
- Solver developers

Support request buttons on every page



Observations

Donations

- Processor cycles
- Many people's time

Status

- Limited support
- No performance guarantees

. . . watch for changes with move to WID

OS www.optimizationservices.org

Optimization Services

- Distributed optimization environment
- Unified framework of services & utilities for
  - modeling languages, servers, registries, agents, interfaces, analyzers, solvers, simulation engines
- Open-source project on COIN-OR

Relevance to cyberinfrastructure

- A "next generation" NEOS
- *Defines* standards for all activities necessary to support decentralized optimization on the Internet
- Embraces distributed computing standards
  - Extensible Markup Language (XML)
  - Service Oriented Architecture
  - Web Services

OS Goals

Conceive optimization as a service

- Software as a product
  - in multiple copies on users' machines
- ***Software as a service***
  - residing on a server
  - accessed by numerous client machines over a network

. . . adapted to special challenges

- Profusion of weak standards for problem description
- Lack of standards for results, queries, instructions
- Variety & complexity of information and behavior
- Disconnect between modeling and solving
  - mathematical problem types that characterize solvers
  - model types familiar to customers

OS Framework

Functions

- Representation of instances, results, and solver options
- Communication between clients and solvers
- Registration & discovery of optimization-related services
  - using the concept of Web Services.

Standards & protocols

- Representations of model instances
  - in text files
  - in memory
- Interfaces to these representations
  - **`get( )`**, **`set( )`**, **`calculate( )`** methods
- Protocols for solver registration & discovery
- Protocols for client-server communication

OS on the Internet

Home site: `www.optimizationservices.org`

- Overview, standards, publications, presentations, FAQs
- Contact information, downloads, licenses

Developer site: `www.coin-or.org/projects/OS.xml`

- Login, register, wiki, source repository, timeline, search

Newsgroup:

`groups.google.com/group/optimizationservices`

COIN mailing list:

`list.coin-or.org/mailman/listinfo/os`

OS Integration

Core (OSCommon library)

Modeler side

- AMPL
- LINGO
- MATLAB

Solver side

- COIN OSI
- AMPL-solver library
- Linear: CLP, CBC, CPLEX, Impact
- Nonlinear: IPOPT, LINDO, KNITRO, Bonmin
- CppAD (automatic differentiation)

OS Repository

Linear (netlib basic, infeasible, Kennington)

- Individual XML (**OSiL** format) files available now
- Zip files to come

Mixed integer (mainly from miplib 2003)

Nonlinear

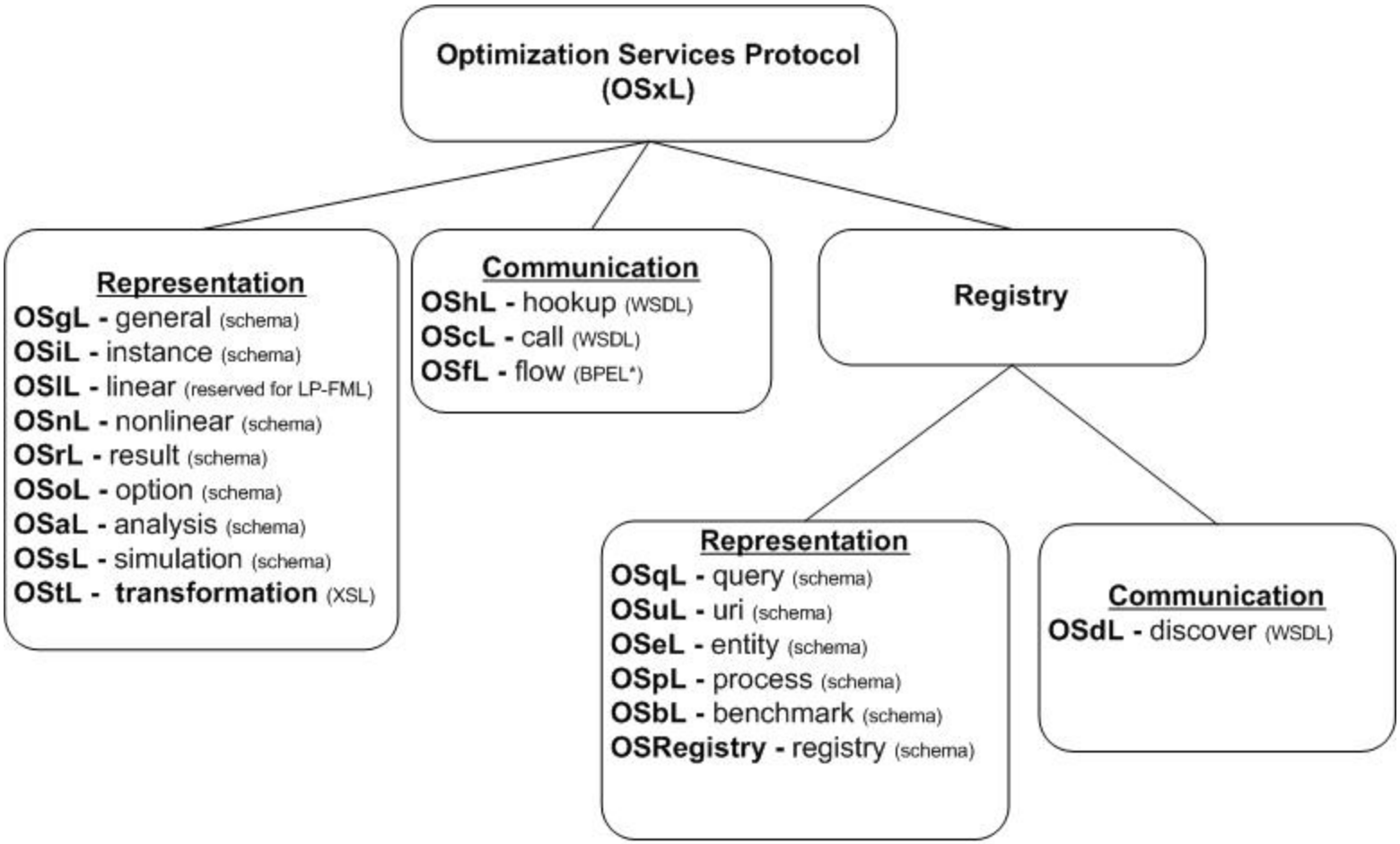
- CUTE now, more to come

Stochastic

- Thanks to Gus Gassmann

. . . all known documentation
(source, solution, description, type, etc.)

OS Standards



***OSmL:** a modeling language and NOT an Optimization Services Protocol
***Letters not currently used:** w, z
***BPEL:** Business Process Execution Language for flow orchestration.

OSiL: Optimization Problem Instances

Design goals

- Simple, clean, extensible, object-oriented

Standard problem types supported

- Linear
- Quadratic
- General nonlinear
- Mixed integer for any of above
- Multiple objective for any of above
- Complementarity

OSiL *(cont'd)*

Extensions (stable or near-stable)

- User-defined functions
- XML data (within the OSiL or remotely located)
- Data lookup (via XPath)
- Logical/combinatorial expressions and constraints
- Simulations (black-box functions)

OSiL *(cont'd)*

Prototypes

- Cone & semidefinite programming
- Stochastic
  - recourse, penalty-based, scenario (implicit or explicit)
  - risk measure/chance constrained
  - major univariate, multivariate, user-defined distributions
  - general linear transformation and ARMA processes
  - R. Fourer, H.I. Gassmann, J. Ma, and R.K. Martin, "An XML-Based Schema for Stochastic Programs." Forthcoming in *Annals of Operations Research*.

OSrL: Optimization Problem Results

Counterpart to OSiL for solver output

- General results such as serviceURI, serviceName, instanceName, jobID, time
- Results related to the solution such as status (unbounded, globallyOptimal, etc.), substatus, message
- Results related to variables (activities), objectives (optimal levels), constraints (dual values)
- Service statistics such as currentState, availableDiskspace, availableMemory, currentJobCount, totalJobsSoFar, timeLastJobEnded, etc.
- Results related to individual jobs including state (waiting, running, killed, finished), userName, submitTime, startTime, endTime, duration, dependencies, scheduledStartTime, requiredDirectoriesAndFiles.

OSrL *(cont'd)*

Additional solution support

- Support for non-numeric solutions
  such as those returned from
  combinatorial or constraint programming solvers
- Support for multiple objectives
- Support for multiple solutions
- Integration of analysis results
  collected by the solver

OSoL: Optimization Options

Counterpart to OSiL for solver instructions

- General options including serviceURI, serviceName, instanceName, instanceLocation, jobID, license, userName, password, contact
- System options including minDiskSpace, minMemorySize, minCPUSpeed
- Service options including service type
- Job options including scheduledStartTime, dependencies. requiredDirectoriesAndFiles, directoriesToMake, directoriesToDelete, filesToCreate, filesToDelete, processesToKill, inputFilesToCopyFrom, inputFilesToCopyTo, etc.

Limited standardization of algorithmic options

- Currently only initial values

OSoL *(cont'd)*

Including support for:

- Various networking communication mechanisms
- Asynchronous communication (such as specifying an email address for notification at completion)
- Stateful communication (achieved mainly through the built-in mechanism of associating a network request with a unique jobID)
- Security such as authentication and licensing
- Retrieving separately uploaded information (when passing a large file as a string argument is inefficient)
- Extended or customized solver-specific or algorithm-specific options

Other XML Schema-Based Standards

Kept by the OS registry

- OSeL (entity, experimental): static information on optimization services (such as type, developer)
- OSpL (process, near stable): dynamic information on optimization services (such as jobs being solved)
- OSbL (benchmark, experimental): benchmark information on optimization services

For use by the discovery process

- OSqL (query, experimental): specification of the query format used to discover the optimization services in the OS registry
- OSuL (uri/url, experimental): specification of the discovery result (in uri or url) sent back by the OS registry

Other Schema-Based Standards *(cont'd)*

Formats and definitions

- OSsL (simulation, stable): format for input and output used by simulation services invoked via the Optimization Services to obtain function values
- OSgL (general, near stable): definitions of general elements and data types used by other OSxL schemas.
  Usually included in the beginning of another OSxL schema through the statement:
  `<xs:include schemaLocation="OSgL.xsd"/>`
- OSnL (nonlinear, stable): definitions (operators, operands, etc.) of the nonlinear, combinatorial, and other nodes used in other OSxL's, mainly OSiL

Other WSDL-Based Standards

WSDL

- Web Service Definition Language

WSDLs for OS (stable)

- OShL (hook): for invoking solver/analyzer services
- OSdL (discover): for invoking optimization registry services to register and discover services
- OScL (call) for invoking simulation services, usually to obtain function values.

Observations

Contrast to NEOS

- Lightweight repository only
- Emphasis on standards

Status

- Still in formative stages
- Release 2.2 in open source from COIN-OR

. . . binary distributions also available

Intelligent Optimization Systems

Beyond repositories and registries

- Interactive aids to modeling
- Automatic connection of modelers to solvers

Relevance to cyberinfrastructure

- Missing link in optimization as a service

Interactive Aids to Modeling

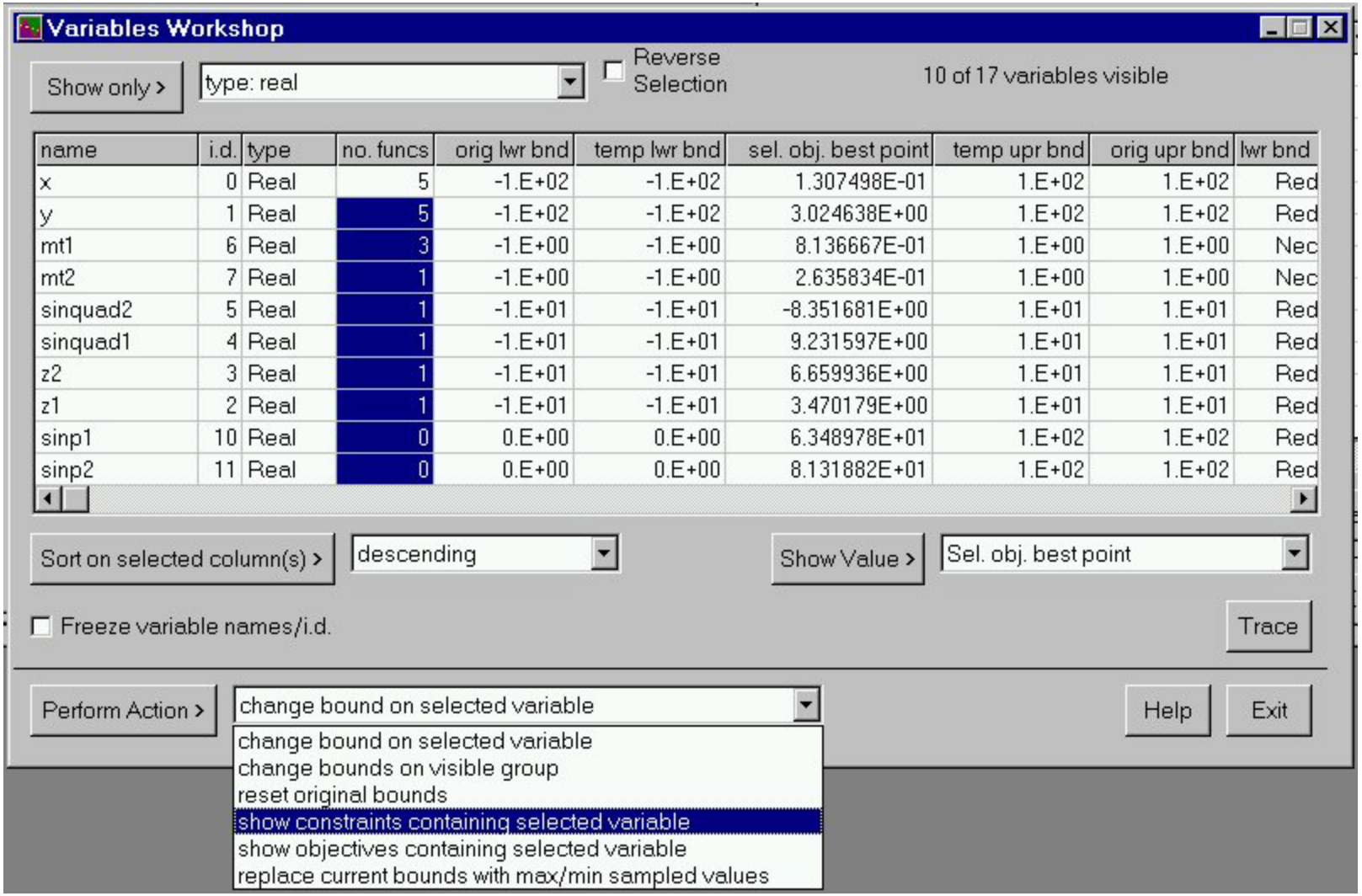
ANALYZE (Greenberg)

- Alternative views
- Simplification
- Rule-based reporting
- Extended sensitivity analysis
- Infeasibility diagnosis

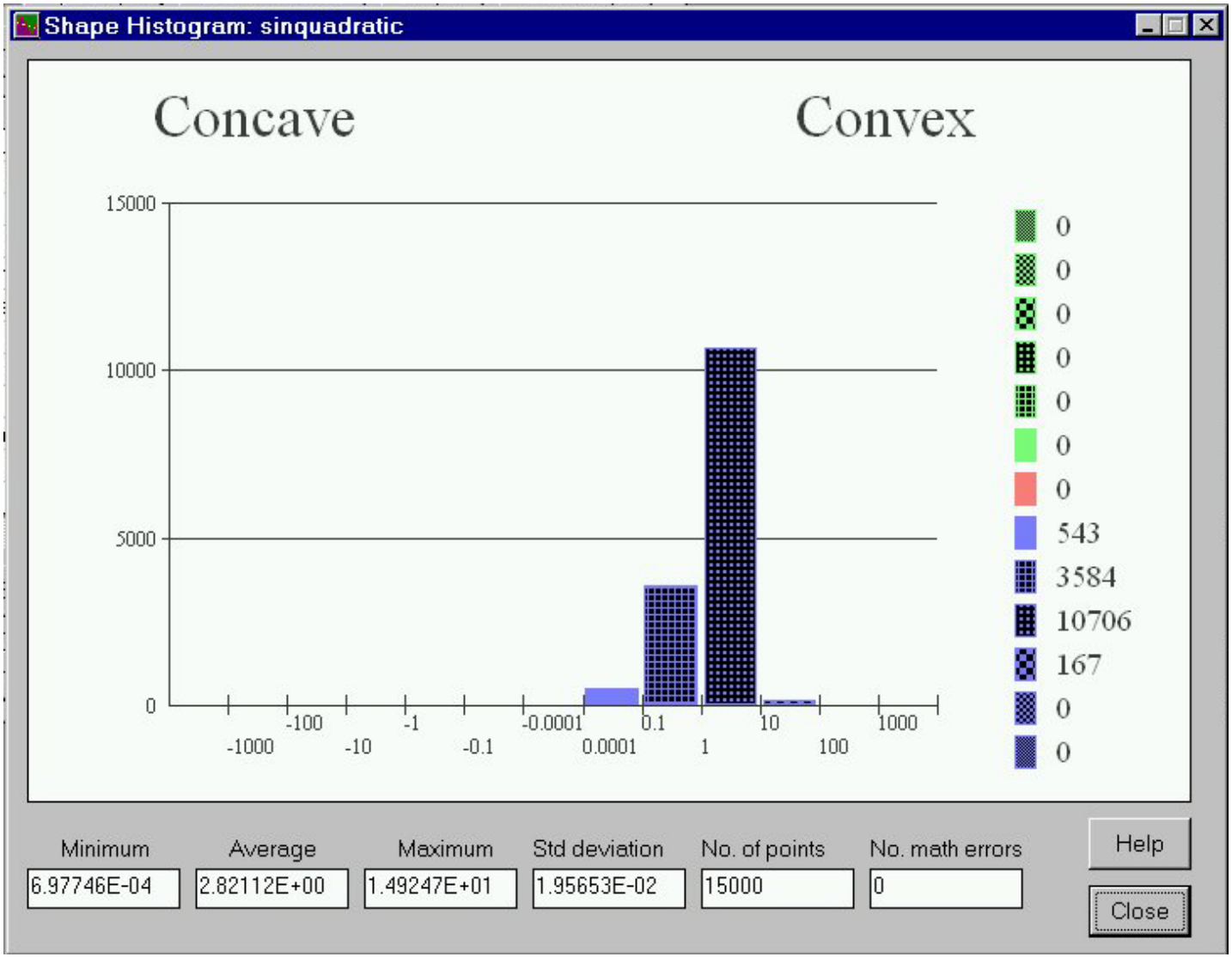
MProbe (Chinneck)

- Analyze feasibility, redundancy, constraint effectiveness
- Tighten bounds, find near-feasible points
- Plot functions
- Analyze function shape (convex? concave? almost linear?)
- Estimate shape of constrained region (convex?)
- Estimate function range and slope
- Estimate objective effect (global? local?)

MProbe Example (1)



MProbe Example (2)



Automatic Connection to Solvers

DrAMPL (Orban & Fourer)

- Process model written in AMPL language
- Determine type of problem instance
  - Linear, quadratic, nonlinear
  - Continuous, differentiable, integer
  - Unconstrained, bounded, constrained
  - Convex, concave
- Compare against database of solver abilities
- Recommend choice of solver

. . . some obvious, some difficult

Example ("Dr. AMPL")

Analysis

```
Problem type
-------------
 -Problem has bounded variables
 -Problem has no constraints

Analyzing problem using only objective
---------------------------------------
 -This objective is quadratic
 -Problem is a QP with bounds

 -0.833013 <= objective <= 0.8359

Problem convexity
------------------
 Nonlinear objective looks convex on its domain.

 Detected 0/0 nonlinear convex  constraints,
          0/0 nonlinear concave constraints.
```

Example (“Dr. AMPL”)

Solver recommendations

```
### Specialized solvers, based on all properties ###

        MOSEK
        OOQP

### Specialized solvers, excluding "hard" properties ###

        BLMVM
        FortMP
        L-BFGS-B
        MINLP
        MOSEK
        OOQP
        PathNLP
        SBB
        TRON

 ### General-purpose solvers ###

       KNITRO
       LANCELOT
       LOQO
```

Challenges

Detection & transformation

- Identify tractable problem types
- Convert to forms that solvers can handle
- Convert the results back

Examples

- Generalized variable domains
  - arbitrary unions of points & intervals
- Separable piecewise-linear functions
- Complementarity constraints
- Logical and combinatorial constraints
  - <c1> or <c2>, <c1> implies <c2>
  - counting, all-different
- Second-order cone constraints

Cyberinfrastructure for Optimization

Advanced computing resources

Cloud computing

Advanced Computing Resources

Platforms offering special power

- Multiple processors
- Huge storage devices
- Fast networks

. . . and algorithms to take advantage of them

Relevance to cyberinfrastructure

- Facilitates many uses — sometimes unanticipated
- Infrastructure already in existence
  - supercomputer centers
  - computational grids

Types of Advanced Computing

High-performance computing

- Large numbers of specialized processors
- Specialized interconnections

Distributed computing (MW)

- Standard computers
- Working together through Internet connections

High-throughput computing (Condor)

- Otherwise idle computers
- Networked by special software

Sample Research

From the metaNEOS project (1997–2001)

- 1010-variable deterministic equivalent of a 107-scenario stochastic program on a computational grid of about 800 workstations, in about 32 hours of wall-clock time
- a previously intractable quadratic assignment problem using an average of 650 worker machines over a one-week period, providing the equivalent of almost 7 years of computation on a single workstation
- a mixed-integer nonlinear programming problem with parallel efficiency of up to 80% on 600 million search-tree nodes

Resulting Applications

Essentially none

- Impractical to write new software for each application
- No access to software from previous studies

Needs

Advanced computing + software services

- Put optimization software on high-performance platforms
- Make it generally available through NEOS or OS

Advanced computing for optimization

- Solving on demand
- Flexible resource allocation

. . . a new paradigm

Cloud Computing

Current significance

- Software-as-a-service by another name
- Cyberinfrastructure for building software services

What it means for optimization providers

- Coordination
  - NEOS scheduler
  - OS registry
- Resources
  - Windows Azure
  - IBM's various options
  - Amazon EC2

Cloud Coordination

NEOS

- Central ***registry***
- Central ***scheduler***
- You supply solvers and hardware

OS

- Central ***registry*** only
- You supply scheduler, solvers and hardware

. . . lighter-weight solution

Cloud Resources

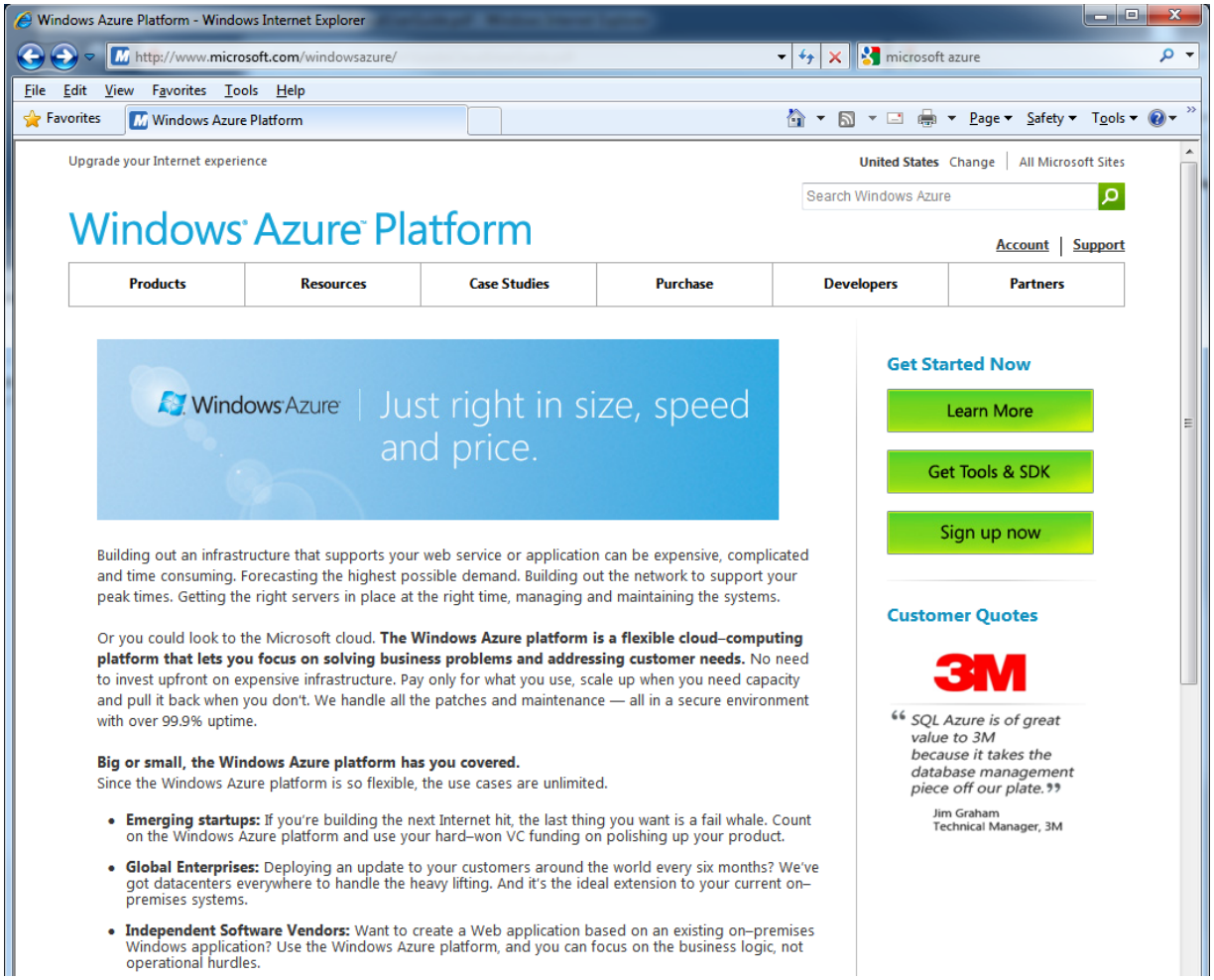
Several commercial offerings

- Central ***hardware***
- Central ***development software***
- You supply solvers

Current examples . . .

Cloud Resources (1)

Microsoft Windows Azure



Excel Solver
for the Cloud
. . . coming soon?

Cloud Resources (2)

IBM solutions, technologies, consulting, design

Many offerings . . .

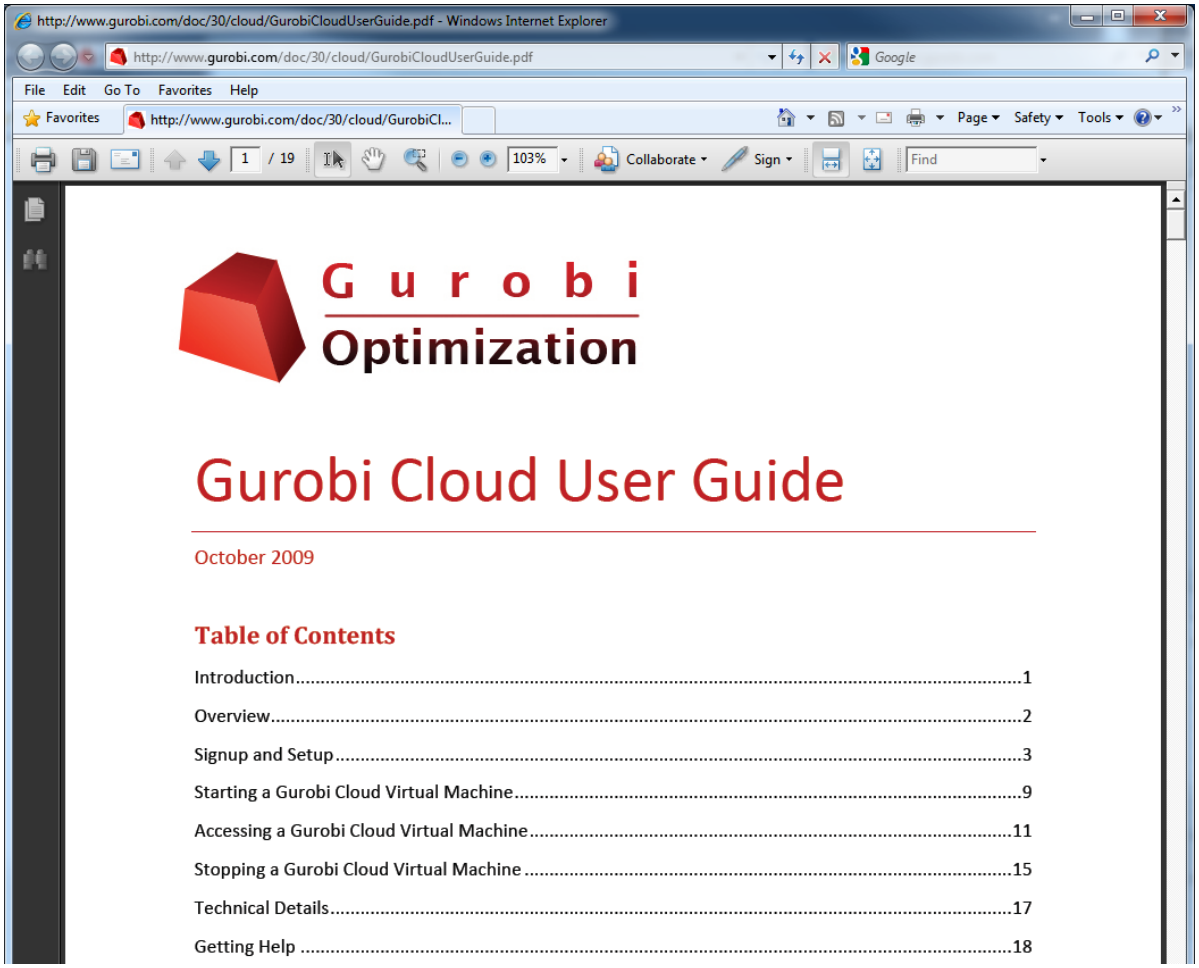
CPLEX
not there yet

Cloud Resources (3)

Amazon EC2

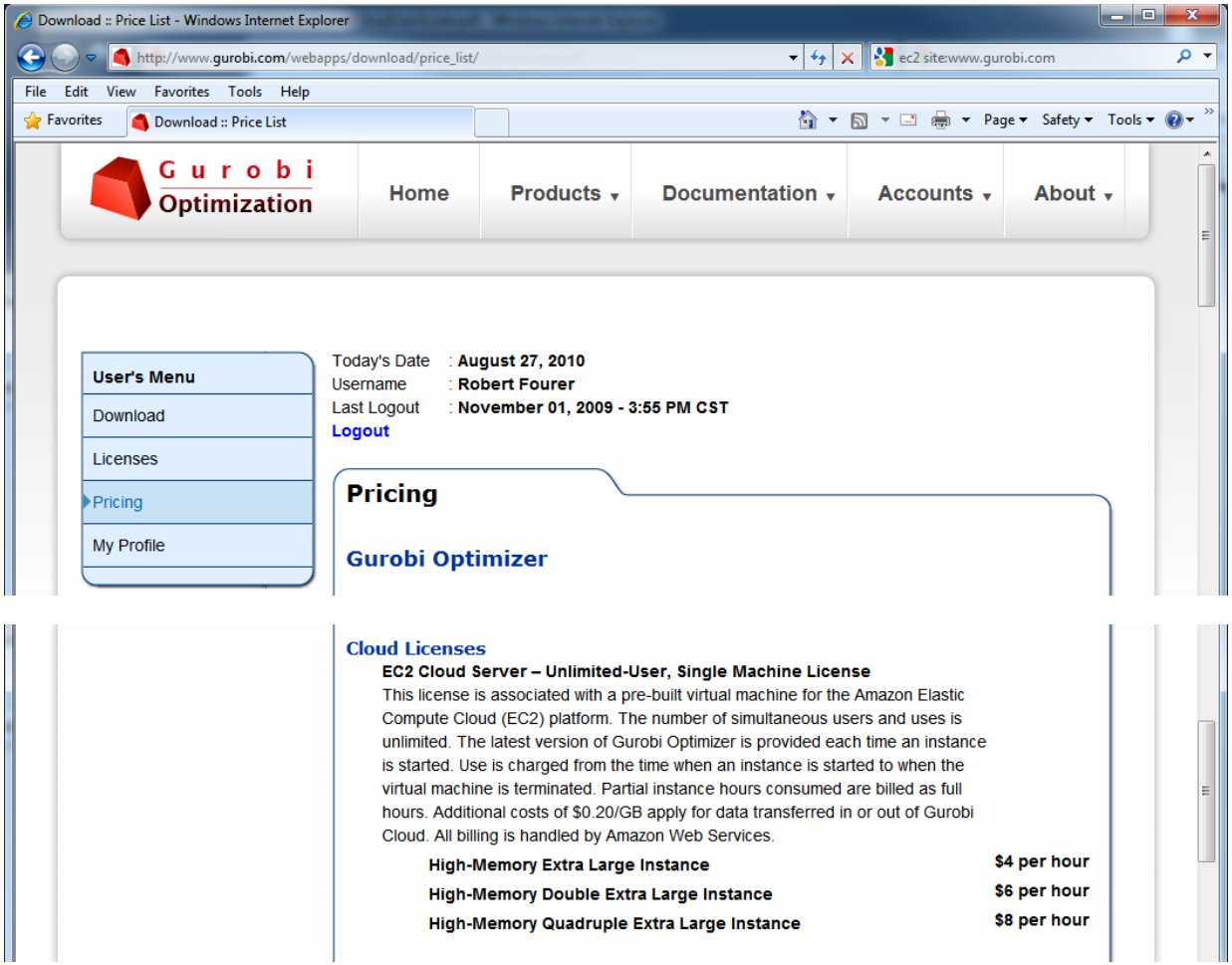
Cloud Resources (3)

Gurobi solver on Amazon EC2



Cloud Resources (3)

Gurobi solver on Amazon EC2 (prices)



Cloud Optimization Prices

Gurobi prices

- High-memory extra-large instance **$4/hour**
- High-memory double extra-large instance **$6/hour**
- High-memory quadruple extra-large instance **$8/hour**
- Data transfer in or out **$0.20/gigabyte**

General principles

- Rent machine time, not CPU time
- Adaptable to mixed modeling and solving

Practical issues

- Will there be day, week, month discounts?
- How can both a modeling system and solver be provided?

. . . need to divide the revenues