What’s New in CPLEX Optimization Studio 12.6.1?
Agenda

- IBM Decision Optimization portfolio
- CPLEX Optimization Studio
  - OPL and the IDE
  - CPLEX CP Optimizer
  - CPLEX Mathematical Programming optimizers
IBM Decision Optimization

Engines and Tools

**Cplex Optimization**
High-performance mathematical/constraint programming solvers and modeling tools

Solution Platform

**Decision Optimization Center**
Build and deploy analytical decision support applications based on optimization technology

Integrated Analytics
- Decision support solutions for **Supply Chain Management**
- **SPSS** predictive analytics
- **Cognos** descriptive analytics
- **Maximo** asset management

Industry Solutions

**Optimization Assets**
Pre-built yet customizable cross-industry applications
Model Development Tools
CPLEX Studio IDE – OPL Modeling Language

ILOG Concert Technology (C++, .NET, Java)

Optimization Engines

Math Programming
CPLEX Optimizers
(Simplex, Barrier, Mixed Integer)

Constraint Programming
Constraint-based scheduling
CPLEX CP Optimizers

Connectors
MATLAB
Python
AMPL

Tools & APIs
CPLEX Interactive
C Callable Library

CPLEX Enterprise Server
Rapid development of optimization models
OPL = Optimization Programming Language
  – Algebraic modeling language
ILOG Script: a JavaScript implementation
  – Pre- and post-processing
  – Flow control
Data sources
  – Flat files
  – Databases
  – Excel
  – SPSS Modeler
Java, C++ and C# Concert APIs for OPL model deployment
Eclipse-based Interactive Development Environment
Model Development Tools

CPLEX Studio IDE – OPL Modeling Language

- Project navigator
- Model/script code editor
- Project outline
- Problem browser
- Logs, conflicts, errors, statistics
Model Development Tools
CPLEX Studio IDE – OPL Modeling Language

In-depth Features of the IBM CPLEX Optimization Studio IDE – Ferenc Katai
Wednesday Nov 12, 11:00am - 12:30pm Union Square 13B

Tips & Tricks: Write Scalable Models Using CPLEX Optimization Studio – Arnaud Schulz
Wednesday Nov 12, 4:30pm - 6:00pm Parc - Mason
- From IDE
  - Panel to submit/monitor/control jobs
  - Results are returned to IDE
- From Java APIs
- OPL .mod and .dat files are sent to server
  - Data read on server, not client
  - Data can be in .dat file or from a database, flat files, etc.

Tutorial: IBM Innovations that Simplify Application Development – John Chaves
Sunday, November 9, 8:45-9:30am
Hilton San Francisco, Green Room
GB Level

Deploying OPL Optimization Models on Client-Server Architectures – Vincent Beraudier
Tuesday Nov 11, 1:30 – 3:00pm
Powell B
ILOG Concert Technology (C++, .NET, Java)

- Common object-oriented technology in
  - OPL
  - Engines

New! Python 3 support

Best Practices Using the CPLEX Python API – Ryan Kersh
Wednesday Nov 12, 2:45pm - 4:15pm Union Square 13
Optimization Engines

**Math Programming**
- CPLEX Optimizers
  - (Simplex, Barrier, Mixed Integer)

**Constraint Programming**
- Constraint-based scheduling
  - CPLEX CP Optimizers
Model and run approach

Constraint programming
  - Discrete decision variables
  - Constraints are not restricted to linear and quadratic, e.g., AllDiff, Count
  - Algorithms associated with each constraint to characterize solutions
  - Alternative to IP when relaxations are hard or weak; best results by reformulating with CP constraints

Constraint-based scheduling
  - Specifically aimed at modeling and solving very complex scheduling problems with potentially millions of tasks/activities

Solve via a search tree with processing at each node
  - Primary technique is propagation: using values in the domains of the constraints to deduce smaller domains
  - Includes genetic algorithms, local search, machine learning, and many other techniques

Deterministic shared memory parallel
New features in CP Optimizer 12.6.1

- Annotated models
  - Associate a file and line number with variables, constraints and expressions
  - Used in error messages and conflicts
  - Used in problem export
- Failure directed search
  - How much time to spend in this search mode
  - Increasing it can be helpful on scheduling models
- Problem export and import using new .cpo file format
CP Optimizer file format

- Contains
  - Variables and constraints
  - Parameter settings
  - Search phases
- A single model can be contained in one file, or span several files
  - Allows separating parameters and model
  - #include directive
- Use cases
  - Send problems to support
  - Debugging
    - Examine the variables and constraints generated with APIs or OPL
    - Export during search with current domains
  - Maintain a model database for in-house testing
- Annotations included as comments
- Comprehensive documentation provided
CP Optimizer File Format

Example:

// Decision variables:
Belgium = intVar(1..4);
Denmark = intVar(1..4);
France = intVar(1..4);
Germany = intVar(1..4);
Luxembourg = intVar(1..4);
Netherlands = intVar(1..4);

parameters {
  SearchType = DepthFirst;
}

/* Constraints: */
Belgium != France;
Belgium != Germany;
Belgium != Netherlands;
Belgium != Luxembourg;
France != Germany;
France != Luxembourg;
Germany != Luxembourg;
Germany != Netherlands;
### CP Optimizer performance

<table>
<thead>
<tr>
<th>Version-to-version average solution time ratios with 4 threads</th>
<th>12.3/12.2</th>
<th>12.4/12.3</th>
<th>12.5/12.4</th>
<th>12.5.1/12.5</th>
<th>12.6/12.5.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scheduling</td>
<td>1.04</td>
<td>0.92</td>
<td>1.35</td>
<td>1.03</td>
<td>2.14</td>
</tr>
<tr>
<td>Integer</td>
<td>1.15</td>
<td>1.02</td>
<td>0.97</td>
<td>1.03</td>
<td>1.05</td>
</tr>
</tbody>
</table>

12.6.1 for integer problems
- 1 thread: +25%
- 4 threads: +15%

Evolutionary Multi Point Search in CPLEX Studio's Constraint Programming Solver Engine – Renaud Dumeur
Tuesday Nov 11, 4:30 - 6:00pm Sutter
- Linear and quadratic constraints; continuous and discrete variables
  - Convex quadratic constraints and SOCPs
  - Convex or non-convex quadratic objectives; global solution for non-convex
- Barrier and simplex optimizers
  - Continuous decision variables
  - Algorithms use linear algebra solvers specialized to unstructured super-sparse matrices
  - Deterministic and opportunistic shared memory parallel algorithms
Mixed integer (MIP) optimizer
- Continuous and discrete decision variables
- Relaxation: let discrete variables take continuous values. The relaxation gives a bound on the solution
- Solve with branch and cut: at search tree node, branch on discrete variables, solve the relaxation and tighten relaxation with cutting planes
- Many techniques used to generate and improve solutions and prune tree without full search
- Deterministic and opportunistic shared and distributed memory parallel algorithms

Tutorial: Identification, Assessment and Correction of Ill-Conditioning and Numerical Instability in Linear and Integer Programs – Ed Klotz
Tuesday Nov 11, 1:30 – 3:00pm
Continental 4
New features in the Math Programming Optimizers 12.6.1

- Distributed MIP improvements
- MIP improvements
- Parameters to control
  - Conflict refiner algorithm
  - Linearization of quadratic terms in the objective
- Query source of solution in a lazy constraint callback: node or heuristic
- Methods to control remote object transport, e.g.,
  - Time-outs
  - Query of process id, MPI rank
Distributed MIP

- A master distributes work to multiple workers
- Two phases
  - Racing ramp-up
    - Each machine uses different settings, and a winner is selected
      - Exploit performance variability
      - Only incumbent objective values and best bounds are communicated
      - Infinite ramp-up allowed (also called concurrent distributed MIP)
  - Distributed tree
    - Nodes of the tree created by the winner are distributed to workers
    - Workers process nodes they receive as supernodes: presolve, cutting planes, etc.
    - Rebalancing at sync points
  - Deterministic or opportunistic
- Available in all APIs and OPL

New! Specify settings in VMC file for ramp-up only
Distributed Concurrent MIP: Deterministic vs. Opportunistic

Compared solvers:
- **Default**: deterministic CPLEX 12.6.1 on one machine
- **Opportunistic**: opportunistic CPLEX 12.6.1 on one machine
- **DistDet4**: deterministic infinite horizon ramp-up, four workers and four machines
- **DistOpp4**: opportunistic infinite horizon ramp-up, four workers and four machines

Time limits: 45 / 40 / 17 / 12

Date: 5 November 2014
Testset: MILP: 2940 models
Machine: Intel X5650 @ 2.67GHz, 24 GB RAM, 12 threads,
Timelimit: 10,000 sec

© 2014 IBM Corporation
MIP improvements

- **Cuts**
  - Different separation strategies in parallel cut loop
  - Improvements in MIR cut aggregator
  - Local implied bound cuts

- **Presolve**
  - Constraint disaggregation
  - Propagation of quadratic objective function and constraints in node presolve

- **Branching**
  - Improvements in branching rule tie breaking
  - Improvements in reliability branching

- **General improvements including dynamic search**
Deterministic parallel MILP (12 threads)

<table>
<thead>
<tr>
<th>Time Limit</th>
<th>CPLEX 12.6.0</th>
<th>CPLEX 12.6.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;1s</td>
<td>1.00</td>
<td>0.94</td>
</tr>
<tr>
<td>&gt;10s</td>
<td>1.00</td>
<td>0.91</td>
</tr>
<tr>
<td>&gt;100s</td>
<td>1.00</td>
<td>0.86</td>
</tr>
</tbody>
</table>

Models:
- >1s: 1975 models
- >10s: 1188 models
- >100s: 619 models

Date: 5 November 2014
Testset: MILP: 4134 models
Machine: Intel X5650 @ 2.67GHz, 24 GB RAM, 12 threads, deterministic
Timelimit: 10,000 sec

Time limits: 40 / 28
CPLEX 12.6.0 vs. CPLEX 12.6.1: MIP Performance Improvement

Convex MIQP

- 1.28x
- 1.67x

251 models
123 >1s models

Convex MIQCP

- 1.23x
- 1.39x

172 models
115 >1s models

Time limits:
- CPLEX 12.6.0
- CPLEX 12.6.1

Time limits:
- 11 / 4
- 1 / 1

Date: 5 November 2014
Testset: Convex MIQP: 335 models, Convex MIQCP: 190 models
Machine: Intel X5650 @ 2.67GHz, 24 GB RAM, 12 threads, deterministic
Timelimit: 10,000 sec

© 2014 IBM Corporation
**CPLEX 12.6.0 vs. CPLEX 12.6.1: MIP Performance Improvement**

- The test set includes 308 QPs and 286 MIQPs
  - Same algorithmic framework (spatial branch-and-bound)
  - Very similar improvements on QPs and MIQPs

**Non Convex (MI)QP**

<table>
<thead>
<tr>
<th>Time limit</th>
<th>CPLEX 12.6.0</th>
<th>CPLEX 12.6.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;0s</td>
<td>1.00</td>
<td>0.90</td>
</tr>
<tr>
<td>&gt;1s</td>
<td>1.00</td>
<td>0.75</td>
</tr>
</tbody>
</table>

### Remarks:

- **Date:** 5 November 2014
- **Test set:** Non Convex QP: 308 models, Non Convex MIQP: 286 models
- **Machine:** Intel X5650 @ 2.67GHz, 24 GB RAM, 12 threads, deterministic
- **Timelimit:** 10,000 sec

© 2014 IBM Corporation
MILP Performance Evolution in CPLEX

Number of timeouts vs. total speedup:
- ≥ 10 sec
- ≥ 100 sec
- ≥ 1000 sec

Date: 5 November 2014
Testset: 3147 models (1792 in ≥ 10sec, 1554 in ≥ 100sec, 1384 in ≥ 1000sec)
Machine: Intel X5650 @ 2.67GHz, 24 GB RAM, 12 threads (deterministic since CPLEX 11.0)
Timelimit: 10,000 sec

© 2014 IBM Corporation
IBM Decision Optimization resources

- **CPLEX Optimization Studio on Academic Initiative** [https://ibm.biz/BdRqgq](https://ibm.biz/BdRqgq)
- **CPLEX Optimization Studio Preview Edition**
- **developerWorks forums** [https://ibm.biz/BdEH6Q](https://ibm.biz/BdEH6Q)
- **developerWorks community** [https://ibm.biz/BdEH6g](https://ibm.biz/BdEH6g)
- **Virtual User Group**
- **Client Success Essentials**
- **Fix Central**
- **Support portal**
- **Request for Enhancements (RFE)**
- **CPLEX Optimization Studio Knowledge Center**
- **Decision Optimization Center Knowledge Center**
Disclaimers

- The information contained in this publication is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this publication or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

- References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM’s sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

- Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.