Deploying OPL Optimization Models on Client-server Architectures
Taking advantage of all your remote computing resources to solve optimization problems, respecting IT standards
Why do we need remote computing resources?

- Server architecture enables separating the computationally intensive algorithms of the Optimizers onto dedicated hardware.
  - Solving difficult problems can be limited by shared memory of a single machine, thus the need of distribution.
- This separation facilitates the scalability and throughput needed in large-scale deployments and can support applications shared among multiple decision makers.
- Easier deployment of the optimization application on end-user machine (only a light client layer instead of the full optimization DLLs)
CPLEX bundles to help benefit from remote computing resources

- Client/Server architecture with Websphere application server: CPLEX Server
- CPLEX distributed MIP from OPL (new)
- Decision Optimization on the Cloud (new)

Demos right here, right now:
- Create / Test and Debug a client server application running optimization thanks to:
  - OPL bundles and the OPL Studio from IBM ILOG Optimization products
  - Less than 10 lines of code.
- Take advantage of distributed parallel MIP optimization without code.
- Use/Discuss the new IBM Cloud offer
What are OPL bundles?

- IBM CPLEX model development toolkit for mathematical and constraint programming
  - OPL Language/Studio:
    - Natural compact mathematical description of optimization models.
    - Development studio with dedicated views for optimization/data/profiling…
    - Connect to relational databases and Excel spreadsheets.
    - Deployment APIs
    - Support for data files, excel, databases (oracle, access, …), SPSS.
  
  - CPLEX Server:
    - Capability to deploy decision optimization applications in enterprise environments using a client-server architecture.
    - Debug throw the standard development studio
    - Flexible and simplified application architecture for industries optimization solutions.

- Decision Optimization on Cloud:
  - Capability to solve OPL projects on the IBM cloud offering
Looking for more on OPL?

- **Wednesday Nov 12, 11:00 - 12:30** In-depth features of the IBM CPLEX Optimization Studio IDE
- **Wednesday Nov 12, 16:30 - 18:00** Tips and Tricks to Write Scalable Models using Cplex Optimization Studio

- Anytime at the exhibitor stands
OPTIMIZATION ON A CLIENT / SERVER ARCHITECTURE
Typical Enterprise Deployment

**Developer**
- Deploy App (jar)

**Clients**
- Submit concurrent jobs
- Monitor progress, abort
- Job interaction

**Jobs**
- WAS
- Manage job queue
- Broadcast deployment/undeployment, submission
- Relay abort
- Relay interactions
- Clean and restart jobs

**Data Server**
- WAS
- Encapsulate access to scenarios
- Security

**Job Processor Node**
- WAS
- Poll/Pick new jobs from queue
- Execute jobs in external processes
- Limit of concurrent jobs
- Send progress messages
- Nodes can be added dynamically

**Management Node**
- WAS
- Poll/Pick new jobs from queue
- Execute jobs in external processes
- Limit of concurrent jobs
- Send progress messages
- Nodes can be added dynamically

**Web Services/SOAP**
- HTTP/HTTPS

**Process Pool**
- Active Processes

**Web Services/SOAP**
- HTTP/HTTPS

**Active Processes**

**JMS**

**SQL**

© 2014 IBM Corporation
Seems pretty complex, right?

- OR modelers are not Code developers:
  - How do you build a CPLEX Server?
  - How to code/debug/deploy a CPLEX server application?
  - APIs means lack of debug support, poor maintenance/evolution level, need for data connectors, mix of data/model (unreadable).
    - Data connectors may also be different if you run locally or server side.
  - APIs are different depending on the language (Java, C++, .NET, ...)
  - How do you code/debug client side if we are given a server running optimization?
- OR modelers are not IT people.
  - How do we deploy such application?
  - Who is responsible for what?
  - IT compliance is mandatory.

- One nice answer is CPLEX Enterprise Server
Development / Debug / Deployment of CPLEX Server models

- You can develop/test/debug your optimization model as a standard model
  - The model will run on the server
  - OPL IDE will populate as a local solve
  - Running locally or on the server is only 1 click.
  - Fine grain persistence selection on Server side if needed for performance, thanks to OPL scripting.

- Deployment is done thanks to a light full Java API
  - Only small jars to redistribute for the client application
  - Simple light API (deployment can be done with less than 10 lines of codes for the simplest ones)
  - Real Java collection API to iterate on the solutions.
  - Access to any OPL type (collection, variables, tuples, cumuls, …)
  - Report of the optim solution can be done on the server or client side as you wish
    - OPL post processing to create create the report (remove null values for examples)
    - Or create the report client side by iterating on all variables and data.
Development / Debug of CES models

Run on a server
Can quit the IDE

Monitor servers/applications/jobs
Development / Debug of CES models

Import / display results

same as local solve,
all info (dataviews, logs) is available
Deployment of CES models (in less than 10 lines of code)

```java
private static final String CPLEXSERVER = "http://localhost:8080/odme";
private static final String PROJECT_PATH = "C:/Users/yberaudi/Application Data/IBM/ILOG/CPLEX_Studio126/workspace/";
private static final String appName = "distMIP";

static public void main(String[] args) throws Exception {
    /** Action 1: create the app to deploy **/
    OplApplicationPackager packager = CplexServerFactory.createApplicationPackager(appName, new File(PROJECT_PATH + appName));

    /** Action 2: get the server instance **/
    CplexServer server = CplexServerFactory.createCplexServer(CPLEXSERVER);

    /** Action 3: deploy the app **/
    Application appId = server.deploy(packager);

    /** Action 4: trigger the execution **/
    Job oplJob = appId.submit("standard");
    int nbRetry = 0;
    while (Job.Status.PROCESSED != oplJob.getStatus() && nbRetry < 20) {
        Thread.sleep(5000);
        nbRetry += 1;
    }

    /** Action 5: access the solution **/
    Set report = oplJob.getResult().getSolution().getElement("report").asSet();
    display(report);

    /** Action 6: Undeploy the OPL application. **/
    server.undeploy(appId);
}
```
CPLEX DISTRIBUTED MIP
Reminder about CPLEX distributed MIP

- The algorithm takes advantage of multiple computers (clusters/grid) to solve a single model.
- The architecture uses notions of:
  - ‘Master’ which controls/distributes work to multiple ‘workers’
  - ‘worker’ which works on a given # of nodes of the search tree and reports back the end/partial results
- There are 2 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**

- Distributed || MIP is available directly from OPL Language, Script, APIs
How to use it?

- Create a VMC file which describes the master/worker nodes

```xml
<?xml version="1.0" encoding="US-ASCII"?>
<vmc>
  <machine name="machine1">
    <transport type="process">
      <cmdline>
        <item value="$(CPLex_STUDIO_DIR)/cplex/bin/x64_win64/cplex.exe"/>
        <item value="-worker-process"/>
        <item value="-libpath=$(CPLex_STUDIO_DIR)/cplex/bin/x64_win64"/>
      </cmdline>
    </transport>
  </machine>
  
  <machine name="machine2">
    <transport type="process">
      <cmdline>
        <item value="$(CPLex_STUDIO_DIR)/cplex/bin/x64_win64/cplex.exe"/>
        <item value="-worker-process"/>
        <item value="-libpath=$(CPLex_STUDIO_DIR)/cplex/bin/x64_win64"/>
      </cmdline>
    </transport>
  </machine>
</vmc>
```

- Add settings to the OPL model (.ops file to OPL)
Running with distMIP either locally or on CES

- Distributed MIP runs transparently in OPL, OPL IDE and CES.

```
Tried aggregator 2 times.
MIP Presolve eliminated 1028 rows and 29 columns.
MIP Presolve modified 32 coefficients.
Aggregator did 1 substitutions.
Reduced MIP has 1118 rows, 36 columns, and 9404 nonzeros.
Reduced MIP has 36 binaries, 0 generals, 0 SOSs, and 0 indicators.
Presolve time = 0.00 sec. (11.93 ticks)
Running distributed MIP on 2 solvers.
Setting up 2 distributed solvers.
Setup time = 0.05 sec. (0.00 ticks)
Starting ramp-up.
```

- Deployment thanks to OPL C++/Java/.NET APIs.
DECISION OPTIMIZATION ON CLOUD
New coming beta

Decision Optimization on Cloud

Solve your problems online

Just drop your problem files to start solving automatically and benefit from exceptional optimization results.

On the beta site, you can view a Demo of the new DropSolve service.

Register to start your Free Trial
Handling your OPL projects with status

DropSolve

Drop your problem file(s) and download results when complete. All relevant files must be dropped in together (lp, mps, sav, prn, mod, dat, ops).

Learn more or get sample

It's time to download Results. To drop again, simply remove a job.

Free trial is limited to 3 jobs at a time.

- eil33-2.lp 559K
  - Completed 2700014TEY14132225153300

- sched_optional.dat 3K (1 more file)
  - Completed 2700014TEY1413216032580

- factoryPlanning1.dat 3K (1 more file)
  - Completed 2700014TEY1413222975521
Solutions are available on web or download
Cloud computing resources.

Benefit from this free trial during the beta version to start solving your own problems.

**Free trial**
- Submit up to 3 problems at a time
- Solve process of up to 60 minutes per problem

Start now

**Buy solving time or Pay monthly**
- Range of hardware options with custom configuration

Coming soon

Tweet #IBMDOcloud  Follow us @IBMOptimization  Contact us
Deployment thanks to IBM cloud

And that's just the beginning!
More services to come.

Developers, you will soon be able to use our Web API to integrate powerful online solvers with your application. We handle the connection so that you can jump into coding faster.
Legal Disclaimer

- © IBM Corporation 2014. All Rights Reserved.
- 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.