Get started with IBM ILOG CPLEX Enterprise Server

Scalability, failover, and client interaction for your optimization solutions

Jun Zhang

February 21, 2012

The IBM ILOG CPLEX Enterprise Server, a new feature of IBM® ILOG® CPLEX® Optimization Studio Version 12.4, allows remote, concurrent, queued, multi-user solving for OPL applications. In this article, learn about the architecture of CPLEX Enterprise Server and implement its CPLEX optimization solution as a client application in Java™.

Introduction

IBM ILOG CPLEX Optimization Studio, which is available for trial download on developerWorks, is a consolidation of the OPL integrated development environment and the CPLEX and CP Optimizer solution engines in a single product. IBM ILOG CPLEX Enterprise Server, a new component, lets you deploy models and solutions within a client-server architecture.

Frequently used abbreviations

• ODM: Optimization Decision Manager
• OPL: Optimization Programming Language
• OR: Operation Research

CPLEX Enterprise Server is a scalable server environment that offers an optimization service based on OPL projects. It provides an enterprise-scale platform for developing and deploying highly effective optimization-based analytical planning and scheduling solutions across a variety of industries.

With the CPLEX Enterprise Server, you can now:

• Implement your CPLEX optimization solution as a client application in Java.
• Send an OPL application (which is currently composed of an OPL project).
• Ask to solve a specific run configuration.
• Get the status of the solved job.
• Retrieve the solution.
The CPLEX Enterprise Server supports: Windows® Server 2008 32/64bits, AIX® 5.3, and AIX 6.1 64 bits. It can be deployed on JavaEE application servers, such as WebSphere® Application Server V7.0 and WebSphere Application Server Community Edition V2.1.1.5.

CPLEX Enterprise Server is based on the IBM ILOG ODM Enterprise Optimization Server architecture. It provides:

- Scalability, by having several processor nodes; one processor node is a server in charge of running optimization jobs.
- Failover, by having jobs automatically restart in case of unexpected errors.
- Client interaction with the ongoing optimization process.

You can deploy the OPL project to CPLEX Enterprise Server, store it in a database, and extract it on the server to be used by the optimization jobs. For better code isolation, execution is in a subprocess for optimal thread and memory management. This also prevents unexpected errors from affecting the application server. Each processor node is able to run multiple optimization processes in parallel (multiple subprocesses).

CPLEX Enterprise Server has a web console to: view and undeploy applications; monitor and stop ongoing jobs (ODM Enterprise Dashboard); and see job status and completion.

CPLEX Enterprise Server provides a Java client API that’s wrapped as a web service. Communication between client and server is through HTTP protocols. Anyone with access privilege to the server URL can deploy/undelete/list applications, and submit/abort/delete jobs.

Compared to a local solve engine, CPLEX Enterprise Server has some limitations. When writing the OPL model, consider the following.

- Only inline data (.dat containing all the values) or database statements are supported. Excel and SPSS connection are not supported in this version.
- There is no automatic encryption.
  You can use the OPL compilation of OPL models if you want some level of confidentiality in your deployed application.
- All models and included submodels must be in the OPL project folder. Subfolders are also supported.

In this article, learn how CPLEX Enterprise Server lets you operate applications in client/server mode with CPLEX on the server and interact with the client that starts optimization.

**Architecture overview**

Figure 1 shows the high-level architecture of CPLEX Enterprise Server.
Install the CPLEX Enterprise Server

The new CPLEX installer is composed of Java EE Enterprise Archives (EARs) at: 
<Install Dir>/server/ears/cplexserver-processor-ear-{x86_win32, x86_win64, or power64_aix.5.3.9.0}.ear.

If you install the ODM Enterprise 3.6 with an embedded WebSphere Application Server instance, you have a ready-to-use CPLEX Enterprise Server on an embedded WebSphere Application Server instance. The default profile is AppSrvOptim. Or, you can use a different application server to install CPLEX Enterprise Server, such as WebSphere Application Server Community Edition, which has been supported since ODM Enterprise 3.6.

Deployment on WebSphere Application Server Community Edition

To install CPLEX Enterprise Server components on WebSphere Application Server Community Edition, execute the following script in a command line tool: 
<Install Dir>/server/configure-scripts/wasce-resources/installServer.bat

The default components option "All" will set up the required database and Java Message Service resources, and then deploy all EARs to WebSphere Application Server Community Edition.

If you want to install CPLEX Enterprise Server on WebSphere Application Server Community Edition separately, you can deploy the EAR manually (using the Administration console or command line tool). If deploying manually, make sure the basic ODME server components (Application and Job Management) are ready.

Develop the OPL project

Operation Research (OR) professionals develop the optimization project by: programming the model in OPL, handling the data access within OPL, creating run configuration in the OPL project, and testing it locally.
For the sake of simplicity, we’ll use the following example.

1. In CPLEX Optimization Studio, start up the IDE.
2. Select **File → New → Example...** → **IBM ILOG OPL Examples → nurses**, as in **Figure 2**.
3. Click **Finish**.

**Figure 2. Import OPL example nurses**

![Import OPL example nurses](image)

An OPL project called nurses is created, which includes the OR model and data file.

To create the Java project that will interact with the server, you need to include the following Java libraries into the project class path dependency, as in **Figure 3**.

- `<Install Dir>/CPLEX_Server/client/lib/axis2-client.jar`
- `<Install Dir>/CPLEX_Server/client/lib/cplexserver-client.jar`
- `<Install Dir>/CPLEX_Server/client/lib/icu4j.jar`
- `<Install Dir>/CPLEX_Server/client/lib/lpgruntime.jar`
- `<Install Dir>/CPLEX_Server/client/lib/odmeserver-client.jar`
Figure 3. Add Java libraries

Client API classes

Use classes from the com.ibm.ilog.optim.server.client package, as shown in Figure 4, to:

- Connect to the CPLEX Enterprise Server.
- Deploy the application.
- Submit an optimization job for an existing run configuration.
- Get job results and optimization model solutions, when available.

Figure 4. com.ibm.ilog.optim.server.client package
You can access OPL data, such as array, tuple, set, dvar, and dexpr, with classes in the com.ibm.ilog.optim.server.client.opl.element package shown in Figure 5.

**Figure 5. com.ibm.ilog.optim.server.client.element package**

At the time of this writing, the client API has some limitations. For example, there is no `find()` method for the `Set` element, you can't access a Map's elements through the indexer, and there's no control on conflicts/relaxation (but you can use the OPL settings to switch them on/off in your .mod file).

**Example using the client API**
After you complete the OPL project and OPL model, you can build the client application using the CPLEX Enterprise Server Java Client API.

Listing 1 shows how to deploy the OPL project to CPLEX Server and then undeploy it.

**Deploy OPL project code**
```java
/** Creates a new OPL application named "nurses".*/
OplApplicationPackager packager = CplexServerFactory.createApplicationPackager
("nursesCPLEX", new File("C:/Documents and Settings/Administrator
/workspace/nurses"));

/** Creates a connection with the CPLEX Enterprise Server.*/
CplexServer server = CplexServerFactory.createCplexServer
("http://localhost:8080/odme");
```
We will deploy our OPL application but if an OPL application with the same name is already deployed, we will use it.

```java
Application appId = null;
try {
    appId = server.deploy(packager);
} catch (ApplicationAlreadyDeployedException e) {
    // If an application with the same name is already deployed, we will use it.
    appId = server.getDeployedApplication("nursesCPLEX");
}
/** Undeploy the OPL application. */
server.undeploy(appId);
```

Listing 2 shows how to submit an optimization job to CPLEX Enterprise Server and monitor the job status.

**Listing 2. Submit optimization job code**

```java
/** We ask for the execution of the default OPL run configuration. */
Job oplJob = appId.submit("Basic Configuration");

/** We will wait for the solve to be done.
* A better way would be to use a thread, but we are only demonstrating the concepts here.
* */
int nbRetry = 0;
// With the Job object, can get more detail and execute action such as abort.
while (Job.Status.PROCESSED != oplJob.getStatus() && nbRetry < 5) {
    try {
        Thread.sleep(5000);
    } catch (InterruptedException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    nbRetry += 1;
}
```

The OPL model has decision variables (`dvar`), as in Listing 3. You need to get the value of those `dvars` after optimization is done.

**Listing 3. OPL decision variables**

```java
dvar int NurseAssignments[Nurses][Shifts] in 0..1;
dvar float+ NurseWorkTime[Nurses];
dvar float+ NurseAvgHours;
dvar float+ NurseMoreThanAvgHours[Nurses];
dvar float+ NurseLessThanAvgHours[Nurses];
dvar float+ Fairness;
dvar int CostByDepartments[Departments];
dvar int AllocationByDepartments[Departments];
```

Listing 4 shows how to retrieve the optimization result.

**Listing 4. Retrieve optimization solution data**

```java
/** Get the result and solution after solve job completed */
Result res = oplJob.getResult();
Solution solution = res.getSolution();
```
//Solution status: we only deal with relaxed, feasible, and best solution.
System.out.println("Solution type is :"+solution.getType());

/** Objective */
try {
    System.out.println("The objective is " + solution.getObjValue());
} catch (ObjectiveValueNotAvailableException e) {
    System.out.println("Objective Value not available!");
}

/** dvar int and dvar float will also be integer and double elements. */
double dvar1 = solution.getElement("NurseAvgHours").asNum();
double dvar2 = solution.getElement("Fairness").asNum();

/** Set of strings: you iterate and get strings */
Set deps = solution.getElement("Departments").asSet();
System.out.println("Departments is composed of ");
/** Iterate on this set the Java way */
Iterator<Element> nurseItr = deps.iterator();
while (nurseItr.hasNext()) {
    String elt = nurseItr.next().asString();
    System.out.print(elt + " ");
}

/** Query Preprocessing objects: you can query the OPL sets, OPL arrays. */
Map nurseAssignments = solution.getElement("NurseAssignments").asMap();
System.out.println("dvar NurseAssignments is composed of :");
/** Iterate on this set the Java way */
Iterator<Element> itr = nurseAssignments.iterator();
while (itr.hasNext()) {
    Map nas2 = itr.next().asMap();
    Iterator<Element> itr2 = nas2.iterator();
    while(itr2.hasNext()){
        int elt = itr2.next().asInt();
        System.out.print(elt + " ");
    }
    System.out.println();
}

After running this class, you should get the result shown in Listing 5 in your console.

Listing 5. Console output

Solution type is : RELAXED_SOLUTION
Objective Value not available!
dvar NurseAvgHours is : 35.438
dvar Fairness is : 35.438
Departments is composed of : Emergency Consultation
dvar NurseAssignments is composed of :
1 1 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 1 0 1 0
0 0 1 0 0 0 0 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0

ODM Enterprise Dashboard

The CPLEX Enterprise Server has a web console to view or undeploy your deployed applications and to view or stop solve jobs. You can access the console in a web browser at a URL similar to http://localhost:8080/odme/console. To specify the ODM Enterprise type application, verify that the Type column contains OPL (indicating that it’s a CPLEX Server application). Figure 6 shows an example. You can then get optimization job detail, such as status, started time, duration, and so on.

Figure 6. Web console

Conclusion

With the CPLEX Enterprise Server, you can operate your applications in client-server mode, with CPLEX running on the server, and interact with the client who starts the optimization. The IBM ILOG CPLEX Enterprise Server introduces a flexible and simplified application reference architecture for industries optimization solutions.

See Related topics to learn more about the CPLEX Enterprise Server. Get more details about this server-based, scalable execution environment for deploying OPL applications that will solve the model, store all the objects of the model, and let you query them on a local client thanks to a Java client API.
Related topics

- Go to the IBM ILOG CPLEX Optimization Studio V12.4 Information Center to learn more about the product line and features.
- Follow developerWorks on Twitter.
- Find other evaluation software.

© Copyright IBM Corporation 2012
Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)