.

Authors:
Ivan Rodero, BSC
Francesc Guim, BSC
Julita Corbalán, UPC
Jesús Labarta, UPC

09 February 2006

A Proposal for Extending the Job Submission Description Language (JSDL) Version 1.0 to Support the Parallelism

Abstract

This document specifies an extension of the JSDL to support the parallelism details of Grid jobs. This extension is proposed in general terms for supporting current multilevel parallel applications and incoming approaches in parallel programming models. The document includes the XML Schema for the JSDL extension, along with examples of JSDL documents based on this schema.

Contents

1 Introduction

In the specification of the JSDL there is a mechanism to determine some parallelism details of jobs but it is not enough for a global and generic solution. The IndividualCPUCount and TotalCPUCount elements could be taken into account to specify the parallelism of the application. For instance we could suppose in a multilevel application (a MPI+OpenMP application) that the IndividualCPUCount element is the number of OpenMP threads and the TotalCPUCount element is the number of MPI processes. It would be a partial solution only for homogeneous resources and it would require doing an assumption about the specification of the application because the original semantic of these attributes is not related to the definition of the application itself. Furthermore, it is not a consistent specification mechanism and it could cause mistakes in the execution of the applications.

In the same way, in the POSIX normative extension it is not available any information related to the parallelism details of the applications. With the Environment element we could specify by environment variables all the required information to describe deeply a parallel application, but this is not a robust and concrete solution.

2 JSDL-par

We are following the same format, normative, and notational conventions that the original specification of the JSDL. In fact, the JSDL provides two mechanisms for its extension: using new attributes or new elements, we have chosen the extension by new XML elements.

The namespace prefix used for this schema in the specification is “jsdl-par”. Since this is a proposal, the normative namespace for this schema is not given.

2.1 Enumeration Types

The following types are defined by this extension: ParallelApplTypeEnumeration, and ParallelApplTopologyEnumeration.

2.1.1 ParallelApplTypeEnumeration Type

The following are the allowed types of parallel applications. It can be extended with new models.

Table 2-1 Programming models (jsdl-par:ParallelApplTypeEnumeration)

Name	Definition
mpi	MPI, message passing
openmp	OpenMP, shared memory model
mpi_openmp	Hybrid programming model MPI+OpenMP
pvm	Parallel Virtual Machine
threads	Threaded application
caf	Co-Array Fortran
upc	Unified Parallel C
hpf	High Performance Fortran
other	Other programming models

2.1.2 ParallelApplTopologyEnumeration Type

The following are the possible kinds of application regarding its topology.

Table 2-2 Topology specification (jsdl-par:ParallelApplTopologyEnumeration)

Name	Definition
unrestricted	The LRMS can select any number of processes/threads for the application between the range specified in level description.
power2	The number of processes/threads for the application MUST be power of two (e.g. 3 power 2 = 9).
explicit	The number of processes/threads is explicitly specified.

2.2 ParallelApplication Element

2.2.1 Definition

This element describes the parallelism details of an application. It contains the description of the application type, the number of levels of the application, and the topology for each level. If it is present as a sub-element of the JSDL Application element it MUST appear only once. It MUST NOT appear for sequential applications.

2.2.1.1 Multiplicity

The multiplicity of this element is zero or one.

2.2.1.2 Type

This is a complex type. It MUST support the following elements:

- ApplType
- Levels
- LevelDescription

2.2.1.3 Attributes

The following attributes are defined:

- name—an optional name for the ParallelApplication element. Its type is xsd:NCName so that it can be reused and referred to from outside the containing document.

2.2.1.4 Pseudo Schema

```
<ParallelApplication name="xsd:NCName"?>
   <ApplType... />
   <Levels... />
   <LevelDescription... />+
   <xsd:any##other>*
</ParallelApplication>
```

2.2.2 ApplType Element

2.2.2.1 Definition

This element is an enumeration type specifying the type of application regarding the programming model used in the application. This type should be interpreted by the Local Resource Management System in order to use the most suitable tool or runtime to execute the application (e.g. POE, OpenMP runtime, etc…).

2.2.2.2 Multiplicity

The multiplicity of this element is one.

2.2.2.3 Type

The type of this element is jsdl-par:ParallelApplTypeEnumeration

2.2.2.4 Attributes

No attributes are defined.

2.2.2.5 Pseudo Schema

```
<ApplType>
   jsdl-par:ParallelApplTypeEnumeration
</ApplType>
```

2.2.2.6 Examples

MPI application:

```
<jsdl-par:ApplType> mpi </jsdl-par:ApplType>
```

2.2.3 Levels Element

2.2.3.1 Definition

This element is a positive integer that specifies the number of parallelism levels of the application. For instance, a MPI+OpenMP application has 2 levels of parallelism.

2.2.3.2 Multiplicity

The multiplicity of this element is one.

2.2.3.3 Type

The type of this element is xsd:positiveInteger .

2.2.3.4 Attributes

No attributes are defined.

2.2.3.5 Pseudo Schema

```
<Levels> xsd:positiveInteger </Levels >
```

2.2.3.6 Examples

Application with 2 levels of parallelism:

```
<jsdl-par:Levels> 2 </jsdl-par:Levels>
```

2.2.4 LevelDescription Element

2.2.4.1 Definition

This element is a complex type specifying the parallelism of the different levels of the application. It is required at least the description of one level since a parallel application has one or more levels of parallelism. This proposal was initially though for two levels of parallelism but can be applied for more levels of parallelism.

2.2.4.2 Multiplicity

The multiplicity of this element is one or more.

2.2.4.3 Type

This is a complex type. It MUST support the following elements:

- Parallelism
- Topology

2.2.4.4 Attributes

The following attributes are defined:

- level—it indicates the level of parallelism that the element describes. It is supposed that the first level of parallelism correspond to the most external layer. Its type is xsd:positiveInteger and the default value is 1 (it is supposed that there is only one level of parallelism like in the case of a MPI application).
- malleable—it indicates if the application is malleable or not. In fact, currently this attribute only has sense for OpenMP applications (the number of threads can be dynamically modified in run-time). Its type is xsd:boolean and the default is false.

2.2.4.5 Pseudo Schema

```
<LevelDescription level=”xsd:positiveInteger”
malleable=”xsd:boolean”?>
   <Parallelism... />?
   <Topology... />?
   <xsd:any##other>*
</LevelsDescription>+
```

2.2.4.6 Examples

Application with at least 6 processes and this number has to be power of 2:

```
<jsdl-par:LevelDescription level=”1”>
  <jsdl-par:Parallelism>
   <jsdl:LowerBoundedRange>6.0</jsdl:LowerBoundedRange>
  </jsdl-par:Parallelism>
  <jsdl-par:Topology>power2
  </jsdl-par:Topology>
</jsdl-par:LevelDescription >
```

2.2.5 Parallelism Element

2.2.5.1 Definition

This is a range value that describes the number of processes or threads that are required by the parallel application. The topology element only has sense if the value of this element is not an exact number. In this case it can be supposed that the application is moldable but the malleability is explicitly indicated by the malleable attribute of the LevelDescription element.

2.2.5.2 Multiplicity

The multiplicity of this element is zero or one.

2.2.5.3 Type

The type of this element is jsdl:RangeValue_Type.

2.2.5.4 Attributes

No attributes are defined.

2.2.5.5 Pseudo Schema

```
<Parallelism>
   jsdl:RangeValue_Type
</Parallelism>?
```

2.2.5.6 Examples

An application with at most 16 processes/threads:

```
<jsdl-par:Parallelism>
   <jsdl:UpperBoundedRange>16.0</jsdl:UpperBoundedRange>
</jsdl-par:Parallelism>
```

2.2.6 Topology Element

2.2.6.1 Definition

This element is an enumeration type specifying the topology of the application. This value can be taken into account by the LRMS when deciding the number of processes or threads that will be spawned for the application.

2.2.6.2 Multiplicity

The multiplicity of this element is zero or one.

2.2.6.3 Type

The type of this element is jsdl-par: ParallelApplTopologyEnumeration

2.2.6.4 Attributes

No attributes are defined.

2.2.6.5 Pseudo Schema

```
<Topology>
   jsdl-par:ParallelApplTopologyEnumeration
</Topology>
```

2.2.6.6 Examples

The number of processes/threads is fixed by the number specified in the parallelism element:

```
<jsdl-par:Topology> explicit </jsdl-par:Topology>
```

2.3 Document Example

The following example shows the description of the details of parallelism of a MPI+OpenMP application. It requires as maximum 16 MPI processes and the number of MPI processes have to be power of 2 (possible assignations are 1, 2, 4, 8 or 16). It requires at least 2 OpenMP threads but the number of threads can be decided by the LRMS because is unrestricted. The assignation of OpenMP threads can be modified dynamically since it is a malleable OpenMP application.

```
<ParallelApplication>
  <ApplType>mpi_openmp</ApplType>
  <Levels>2</Levels>
  <LevelDescription level="1">
   <Parallelism>
     <jsdl:UpperBoundedRange>16.0</jsdl:LowerBoundedRange>
   </Parallelism>
   <Topology>power2</Topology>
  </LevelDescription>
  <LevelDescription level="2" malleable="true">
   <Parallelism>
     <jsdl:LowerBoundedRange>2.0</jsdl:UpperBoundedRange>
   </Parallelism>
   <Topology>unrestricted</Topology>
  </LevelDescription>
</ParallelApplication>
```