

# AliEn Job Splitting



Pablo Saiz

CAF and Grid User Forum

# Job Splitting

---

- Simplify the handling of multiple jobs
- The user submits one single JDL (masterjob)
- AliEn will split it in several subjobs
- Each subjob can be executed independently
- Possible to merge the output of the subjobs
- Once all the subjobs and the merging finish, the masterjob will be marked as DONE
- 'masterjob <jobid>'

# Splitting methods

---

## □ Possible splitting methods:

- Production

- File

- SE

- Directory

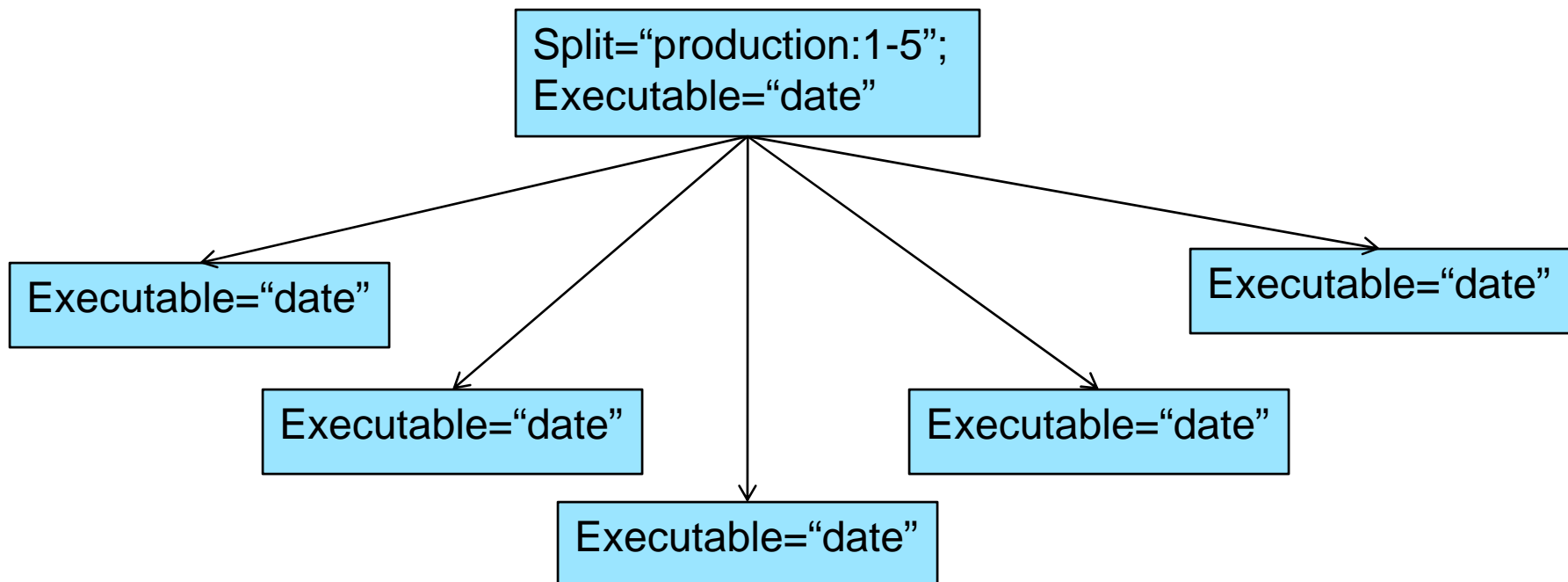
- Event

} According to InputData and/or  
InputDataCollection

- Splitting Arguments

# Production splitting

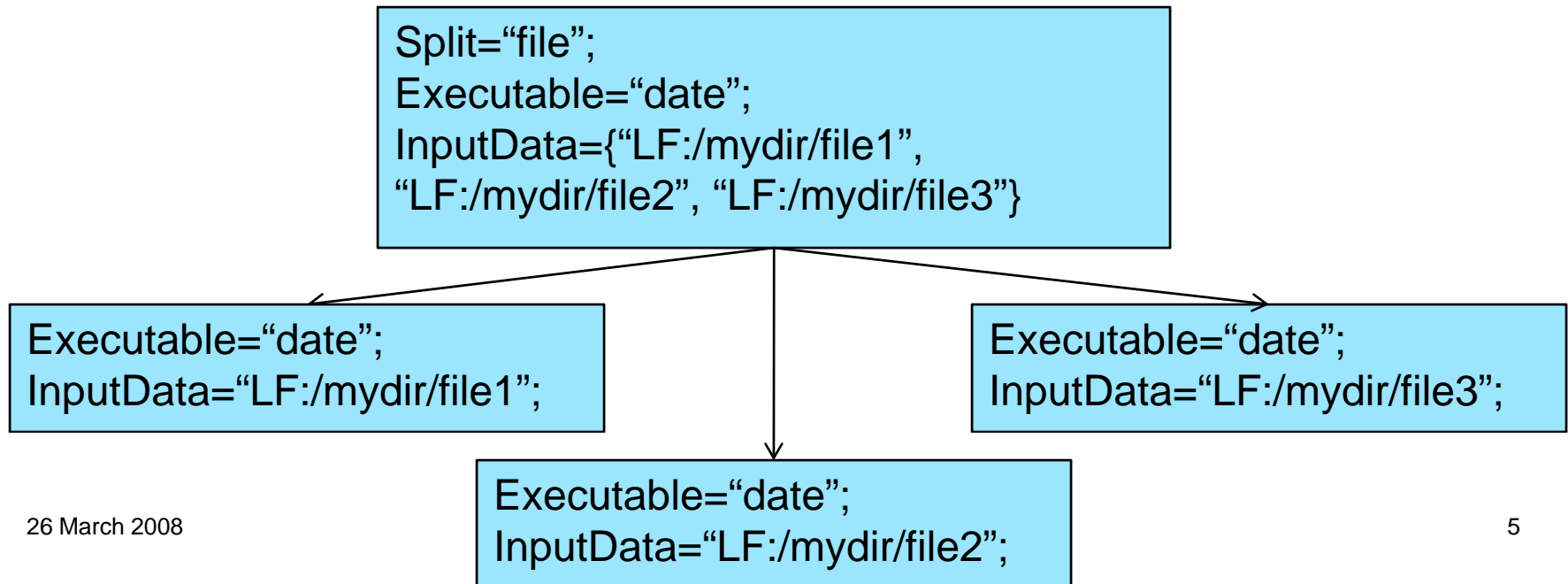
- The JDL of the masterjob will be the same for each subjob
- Example:



# File splitting

---

- Each subjob has one file as InputData
- It can be used with InputData or InputDataCollection
- Example:



# SE splitting

- Each subjob has all the files on the same Ses
  - Possible to restrict with “SplitMaxInputFileSize” and “SplitMaxInputFileNumber”

- Example:

Let's assume:  
File1 → CERN, CNAF  
File2 → CERN, CNAF  
File3 → CERN  
File4 → CNAF

```
Split="SE";  
Executable="date";  
InputDataCollection="LF:/mydir/mycollection1"
```

```
Executable="date";  
InputData= {"LF:/mydir/file1",  
"LF:/mydir/file2"};  
Req=(execute at CERN or  
CNAF)
```

```
Executable="date";  
InputData="LF:/mydir/file3";  
Req=(execute at CERN)
```

```
Executable="date";  
InputData="LF:/mydir/file4";  
Req= (execute at CNAF)
```

# Directory splitting

---

- All the files in the same directory will be analyzed by the same job
  - Possible to restrict with “SplitMaxInputFileSize” and “SplitMaxInputFileNumber”

- Example:

```
Split="directory";  
Executable="date";  
InputData={"LF:/mydir1/file1",  
"LF:/mydir1/file2", "LF:/mydir2/file3"}
```

```
Executable="date";  
InputData={"LF:/mydir/file1",  
"LF:/mydir/file2"};
```

```
Executable="date";  
InputData="LF:/mydir2/file3";
```

# Event splitting

---

- All the subjobs under the same directory name will be analyzed together
  - Not the full directory name!!
- Example:

```
Split="event";  
Executable="date";  
InputData={"LF:/mydir1/event1/file1",  
"LF:/mydir2/event1/file2", "LF:/mydir2/event2/file3"}
```

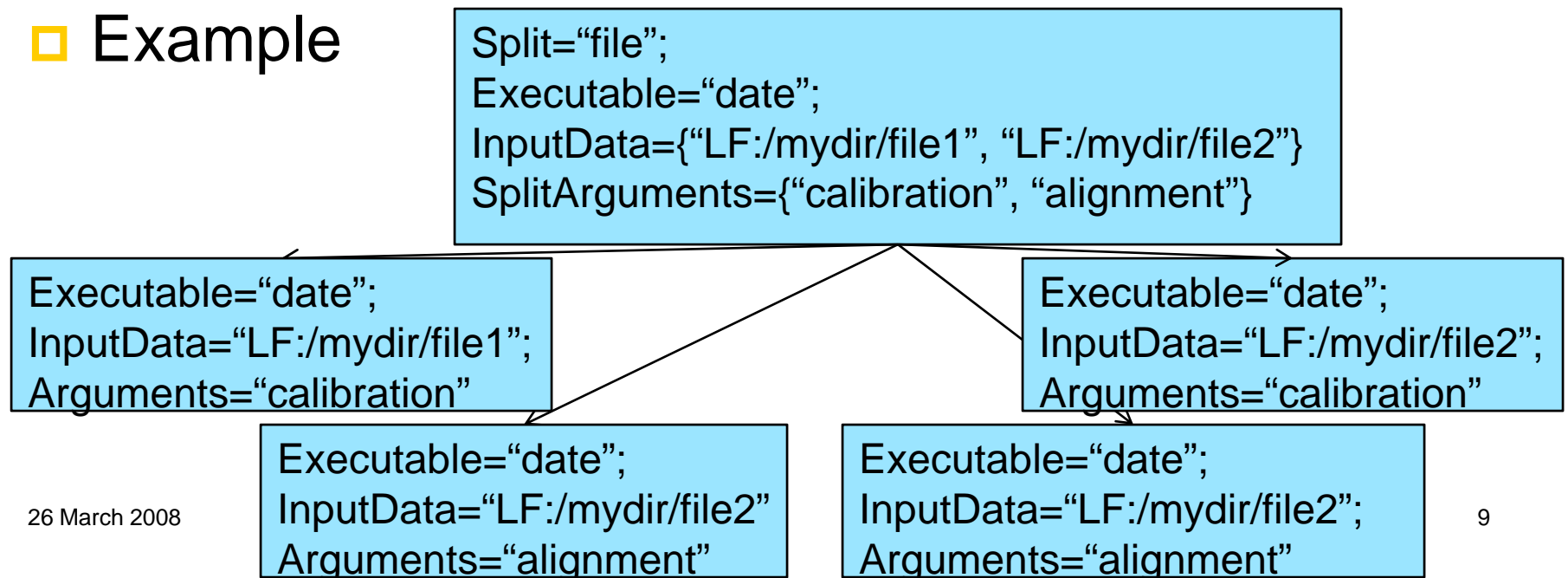
```
Executable="date";  
InputData={"LF:/mydir1/event1/file1",  
"LF:/mydir2/event1/file1"}
```

```
Executable="date";  
InputData="LF:/mydir2/event2/file3"
```



# SplitArguments

- Can be combined with any of the previous methods.
- Every job will be submitted as many times as entries in the SplitArguments
- Example



# Splitting Patterns

---

- Used to have different values on each subjob
- Can be used for: OutputDir, OutputFile, Arguments
- #alien(pattern)#
- Example

```
Split="file";  
Executable="date";  
InputData={"LF:/mydir/file1", "LF:/mydir/file2"}  
Arguments="analyze #alienfilename#"
```

```
Executable="date";  
InputData="LF:/mydir/file1";  
Arguments="analyze file1"
```

```
Executable="date";  
InputData="LF:/mydir/file2";  
Arguments="analyze file2"
```

# All possible patterns

---

- filename

- fulldir

- (dir)+

} Based on the InputData. If there is more than one inputdata, the pattern can be prepended by: first, last, all

- `_counter`

- Category of the splitting ( be careful!!)

- `_split`

- Number of the subjob: 1, 2, 3 ...

`_counter` and `_split` can specify format (i.e. `_split%03i`)

# AliEn pattern examples

---

```
Split="file";  
Executable="date";  
InputData={"LF:/mydir1/dir1/file1", "LF:/mydir2/dir2/file2"}  
Arguments="#alienfilename# #alienfulldir# #aliendir# #aliendirdir#  
#alien_counter# #alien_split#"
```

↓

```
Executable="date";  
InputData="LF:/mydir/file1";  
Arguments="file1 /mydir1/dir1/file1 dir1 mydir1 /mydir1/dir1/file1 1"
```

↓

```
Executable="date";  
InputData="LF:/mydir/file2";  
Arguments="file2 /mydir2/dir2/file2 dir2 mydir2 /mydir2/dir2/file2 2"
```

# Merging

---

- Once all the subjobs finish, AliEn can submit jobs to merge the output
- Merge={"<input>:<jdl>:<output>"}
- Example:

```
Split="file";  
Executable="alroot";  
InputData={"LF:/mydir/file1", "LF:/mydir/file2"};  
OutputFile={"histogram.root", "histogram.log"}  
Merge={"histogram.root:/alice/jdl/mergeroot.jdl:Newhistogram.root",  
       "histogram.log:/alice/jdl/mergetxt.jdl:Newhistogram.log"}
```

# InputDataList

---

- Get the list of files that will be analyzed by the job
- AliEn will create a file in the working directory before the job runs
- InputDataListFormat:
  - Xml-single
  - Xml-group
  - Merge:<lfn>:
    - <lfn> is an xml file with one entry per file
    - The output will be another xml file with the files that the subjob has as inputdata

# Conclusions

---

- Splitting jobs simplifies handling multiple jobs
  - 'masterjob <id>' to follow up status
- Several splitting mechanism:
  - Production, file, SE, directory, event
  - SplitArguments
- Split patterns can be used to configure subjobs
- Easy merging of all the subjobs

[http://alien.cern.ch/twiki/bin/view/AliEn/HowToUseAliEn#Splitting\\_jobs](http://alien.cern.ch/twiki/bin/view/AliEn/HowToUseAliEn#Splitting_jobs)