

---

# **Guide d'utilisation de DIRAC**

*Version 2.2*

**Jérôme Pansanel**

**sept. 07, 2023**



---

## Table des matières

---

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>La grille de calcul</b>	<b>3</b>
<b>3</b>	<b>Gestion d'un calcul avec DIRAC</b>	<b>5</b>
3.1	Pré-requis à l'utilisation de DIRAC . . . . .	5
3.2	Soumission d'un calcul . . . . .	6
3.3	Suivi de l'état d'un calcul . . . . .	7
3.4	Récupération des résultats d'un calcul . . . . .	7
<b>4</b>	<b>Utilisation avancée de DIRAC</b>	<b>9</b>
4.1	Soumission d'un ensemble paramétrique de calculs . . . . .	9
<b>5</b>	<b>Utilisation des commandes COMDIRAC</b>	<b>11</b>
<b>6</b>	<b>Documentation complémentaire</b>	<b>13</b>



# CHAPITRE 1

---

## Introduction

---

Le service de calcul à haut-débit (*HTC*) de la [plateforme SCIGNE](#) repose sur un cluster de calcul optimisé pour le traitement de tâches parallélisées par la donnée. Ce cluster est connecté à une infrastructure plus large, la [grille européenne de calcul EGI](#).

La gestion des calculs est réalisée avec [DIRAC](#). Cet outil permet de gérer vos calculs sur le cluster de la plateforme SCIGNE, et également sur l'infrastructure de grille EGI.

Après une introduction sur le fonctionnement d'une grille de calcul, cette documentation décrit la soumission, le suivi et la récupération des résultats de vos calculs avec DIRAC.



### La grille de calcul

Cette section présente les différents services utilisés lors de la soumission d'une tâche (*job*) sur la grille de calcul. L'interaction entre ces services lors du *workflow* d'un job est illustrée sur la figure suivante. Les acronymes sont détaillés dans le tableau *Les principaux services de la grille de calcul*.

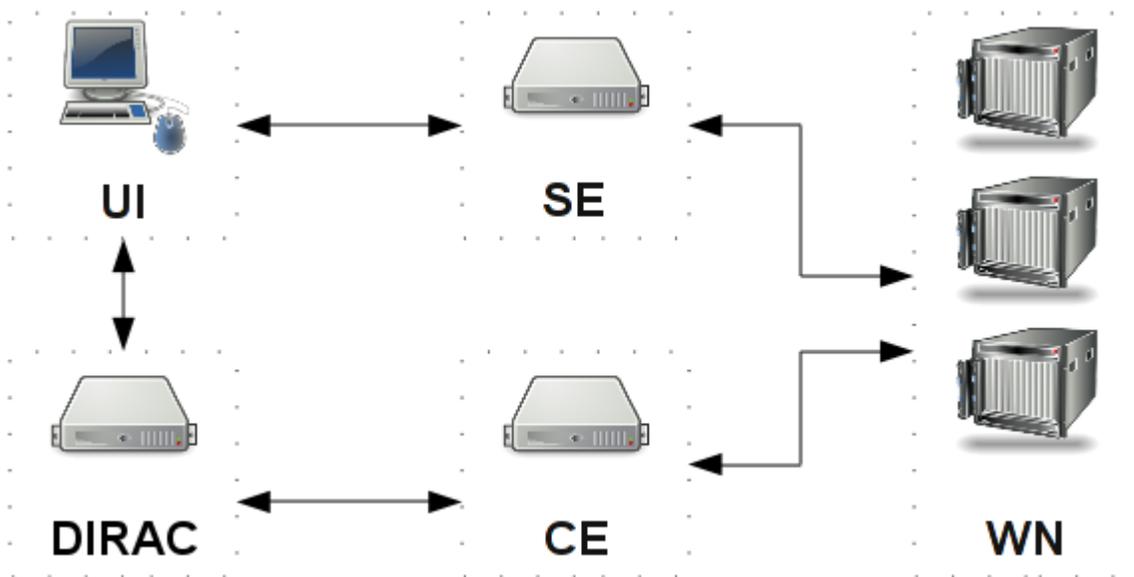


Fig. 2.1 – Le déroulement d'un calcul sur l'infrastructure de grille

L'utilisateur gère les calculs avec DIRAC. Avant de soumettre le calcul, les données volumineuses (> 10 Mo) doivent être copiées sur le *Storage Element* (SE). Le fonctionnement du SE est détaillé dans la [documentation dédiée au stockage](#). Ces données seront accessibles depuis les nœuds de calcul. Ensuite, le calcul est soumis au service DIRAC, qui sélectionne le (ou les) site(s) devant effectuer les calculs. Pour effectuer ce choix, DIRAC se base sur la disponibilité du site et le respect des pré-requis du calcul explicités par l'utilisateur. Une fois cette sélection réalisée, DIRAC soumet le calcul au *Computing Element* (CE). Cet élément distribue les jobs sur les *Worker Nodes* (WN). Une fois les calculs

terminés et les résultats copiés sur un SE, le CE collecte les informations d'exécution qui sont ensuite transférées à DIRAC.

Durant le temps des calculs, l'utilisateur peut interroger DIRAC pour connaître l'état du déroulement de son calcul. À la fin du workflow, les résultats peuvent être récupérés par l'utilisateur sur le SE et les journaux d'exécution sur DIRAC.

Tableau 2.1 – Les principaux services de la grille de calcul

Élément	Rôle
UI	L'UI ( <i>User Interface</i> ) est la station de travail équipée du client DIRAC. Elle permet de : <ul style="list-style-type: none"> <li>— gérer un calcul (soumission, suivi de l'état, annulation);</li> <li>— récupérer les résultats d'un calcul;</li> <li>— copier, répliquer ou supprimer des données sur la grille.</li> </ul>
DIRAC	DIRAC est le serveur avec lequel interagit l'utilisateur lors de la soumission, le suivi et la récupération des résultats d'un calcul. Ce serveur choisit les sites de grille correspondant au pré-requis du calcul. Il interagit principalement avec le CE pour la soumission des calculs, le transfert des <i>sandbox</i> et le suivi de l'état d'un calcul.
SE	Le SE ( <i>Storage Element</i> ) est l'élément d'une grille gérant le stockage. Il permet de récupérer les résultats des calculs ou bien de fournir au calcul des fichiers de données volumineux. Il est accessible via différents protocoles.
CE	Le CE ( <i>Computing Element</i> ) est le serveur interagissant directement avec le gestionnaire de queues. Il permet à DIRAC de soumettre des calculs sur un cluster local.
WN	Le WN ( <i>Worker Node</i> ) est le serveur effectuant le calcul. Il se connecte au SE pour récupérer les données nécessaires au calcul. Il peut également copier les résultats du calcul sur le SE.

---

## Gestion d'un calcul avec DIRAC

---

Le logiciel DIRAC peut être utilisé avec un client en ligne de commande ou à travers l'interface Web DIRAC. Cette section détaille l'utilisation de la ligne de commande.

### 3.1 Pré-requis à l'utilisation de DIRAC

Afin de pouvoir soumettre un calcul, les deux pré-requis suivants sont nécessaires :

- disposer d'une station de travail équipée du client DIRAC ;
- disposer d'un certificat valide enregistré dans une organisation virtuelle (VO). Le document [Gestion d'un certificat](#) détaille la demande de certificat et l'inscription aux VOs locales, régionales ou autres.

#### 3.1.1 Client DIRAC

L'installation d'un client DIRAC est détaillée dans la [documentation de DIRAC](#).

Pour les utilisateurs disposant d'un compte à l'IPHC, le client DIRAC est pré-installé sur les serveurs *UI*. Pour mettre en place l'environnement DIRAC, il suffit de réaliser :

```
$ source /cvmfs/dirac.egi.eu/dirac/bashrc_egi
$ export X509_CERT_DIR=/etc/grid-security/certificates
$ export X509_VOMS_DIR=/etc/grid-security/vomsdir
```

#### 3.1.2 Certificat

La partie privée et la partie publique du certificat doivent être placées dans le répertoire `${HOME}/.globus` sur le serveur à partir duquel les calculs seront soumis. Ces fichiers ne doivent être lisibles que par leur propriétaire :

```
$ ls -l $HOME/.globus
-r----- 1 user group 1935 Feb 16 2010 usercert.pem
-r----- 1 user group 1920 Feb 16 2010 userkey.pem
```

Dans cette documentation, nous utilisons la VO `vo.grand-est.fr`, qui est la VO régionale. Vous devez remplacer le nom de cette VO par celle que vous utilisez (par ex. `biomed` ou `vo.sbg.in2p3.fr`).

Avant toute opération sur la grille régionale, il est nécessaire d'avoir un *proxy* valide. Il peut être généré avec la commande suivante :

```
$ dirac-proxy-init -U -g grandest_user -M
```

L'option `-U` permet d'indiquer que nous souhaitons un *proxy* longue durée, l'option `-g` indique le groupe auquel nous appartenons et l'option `-M` informe que l'extension *VOMS* doit être ajoutée au *proxy*.

La commande suivante permet de vérifier la durée de validité de son *proxy* :

```
$ dirac-proxy-info
subject      : /O=GRID-FR/C=FR/O=CNRS/OU=IPHC/CN=Prenom Nom/CN=proxy/CN=proxy
issuer       : /O=GRID-FR/C=FR/O=CNRS/OU=IPHC/CN=Prenom Nom/CN=proxy
identity     : /O=GRID-FR/C=FR/O=CNRS/OU=IPHC/CN=Prenom Nom
timeleft     : 23:53:43
DIRAC group  : grandest_user
rfc          : False
path         : /tmp/x509up_u1000
username     : pnom
properties   : NormalUser
VOMS         : True
VOMS fqan    : ['/vo.grand-est.fr']
```

## 3.2 Soumission d'un calcul

La soumission d'un calcul avec DIRAC nécessite un fichier texte contenant des instructions au format JDL (*Job Description Language*). Ce type de fichier détaille les caractéristiques du calcul ainsi que ses besoins. Le fichier JDL présenté dans l'exemple ci-dessous est fonctionnel et utilisable pour la réalisation d'un calcul simple. Ce fichier sera nommé `myjob.jdl` dans la suite de cette documentation.

```
JobName = "mysimplejob";
Executable = "/bin/bash";
Arguments = "myscript.sh";
StdOutput = "stdout.out";
StdError = "stderr.err";
InputSandbox = { "myscript.sh" };
OutputSandbox = { "stdout.out", "stderr.err" };
VirtualOrganisation = "vo.grand-est.fr";
```

Chaque attribut du fichier JDL exemple a un rôle bien précis :

- *JobName* décrit le nom de la tâche réalisé.
- *Executable* définit la commande à exécuter sur les nœuds de calcul.
- *Arguments* spécifie les arguments à passer au programme définit par l'attribut *Executable*; le contenu du script `myscript.sh` contient ce que vous auriez tapé en interactif pour effectuer votre calcul.
- *StdOutput* indique le nom du fichier vers lequel est redirigée la sortie standard.
- *StdError* indique le nom du fichier vers lequel sont redirigés les messages d'erreur.
- *InputSandbox* indique les fichiers qui seront envoyés avec le fichier JDL au serveur DIRAC. Ces fichiers seront utilisables par le serveur (WN) exécutant le calcul. Il est important de noter que la taille totale de l'*InputSandbox* est limitée. Ainsi, si la taille de l'ensemble des fichiers à envoyer devait dépasser 10 Mo, il est conseillé d'utiliser un SE, ou de préinstaller les logiciels nécessaires au calcul.
- *OutputSandbox* indique les fichiers que nous souhaitons récupérer après l'exécution du calcul. Par défaut, il est conseillé de récupérer les fichiers `stdout.out` et `stderr.err` qui contiennent les sorties standard et

erreur. Ces fichiers seront téléchargés depuis DIRAC. De même que pour l'*InputSandbox*, le volume retourné ne doit pas dépasser 10 Mo. Si la taille totale des fichiers de l'*OutputSanbox* dépasse cette valeur, il est nécessaire de copier les fichiers de sortie (résultats, journaux du calcul) sur un SE, puis de les récupérer par la suite leur troncation.

- *VirtualOrganization* renseigne sur la VO dans laquelle va être effectué le calcul. Dans le cas de la grille régionale, il s'agit de `vo.grand-est.fr`.

Un exemple de script `myscript.sh` est détaillé ci-dessous :

```
#!/bin/sh

echo "==== Begin ====="
date
echo "The program is running on $HOSTNAME"
date
dd if=/dev/urandom of=fichier.txt bs=1G count=1
gfal-copy file://`pwd`/fichier.txt https://sbgsel.in2p3.fr/dpm/in2p3.fr/home/vo.grand-
↪est.fr/lab/name/fichier.txt
echo "==== End ====="
```

Les exécutables utilisés dans le fichier `myscript.sh` doivent être installés sur la grille. L'installation de ces logiciels est assurée par l'équipe support de la plateforme. N'hésitez pas à [la contacter](#) si vous avez besoin de logiciels spécifiques.

La commande `gfal-copy` permet de copier le fichier créé sur l'élément de stockage `sbgsel.in2p3.fr`. Le fonctionnement du service de stockage est détaillé dans la [documentation dédiée](#). Une fois que les fichiers `myjob.jdl` et `myscript.sh` sont créés, le calcul peut être soumis sur la grille avec la commande suivante :

```
$ dirac-wms-job-submit myjob.jdl
JobID = 79342906
```

Il est important de conserver l'identifiant de la tâche 79342906. En effet, cet identifiant est utilisé pour suivre les différentes étapes d'un calcul, ainsi que pour récupérer l'*OutputSandbox*.

### 3.3 Suivi de l'état d'un calcul

Pour connaître l'état d'un calcul, la commande suivante est utilisée avec l'identifiant du calcul :

```
$ dirac-wms-job-status 79342906
JobID=79342906 Status=Waiting; MinorStatus=Pilot Agent Submission; Site=ANY;
```

Une fois que le calcul est terminé, la commande précédente indique :

```
$ dirac-wms-job-status 79342906
JobID=79342906 Status=Done; MinorStatus=Execution Complete; Site=LCG.SBG.fr;
```

Les différents états que peuvent prendre le status d'un job sont détaillés sur la figure 3.1.

### 3.4 Récupération des résultats d'un calcul

Pour récupérer le contenu décrit par le paramètre *OutputSandbox* :

```
$ dirac-wms-job-get-output 79342906
```

Le rapatriement des fichiers de sortie peut être vérifié avec :

```
$ ls -rtl /home/user/79342906
total 12
-rw-r--r-- 1 user group 73 juil. 25 15:12 StdOut
```

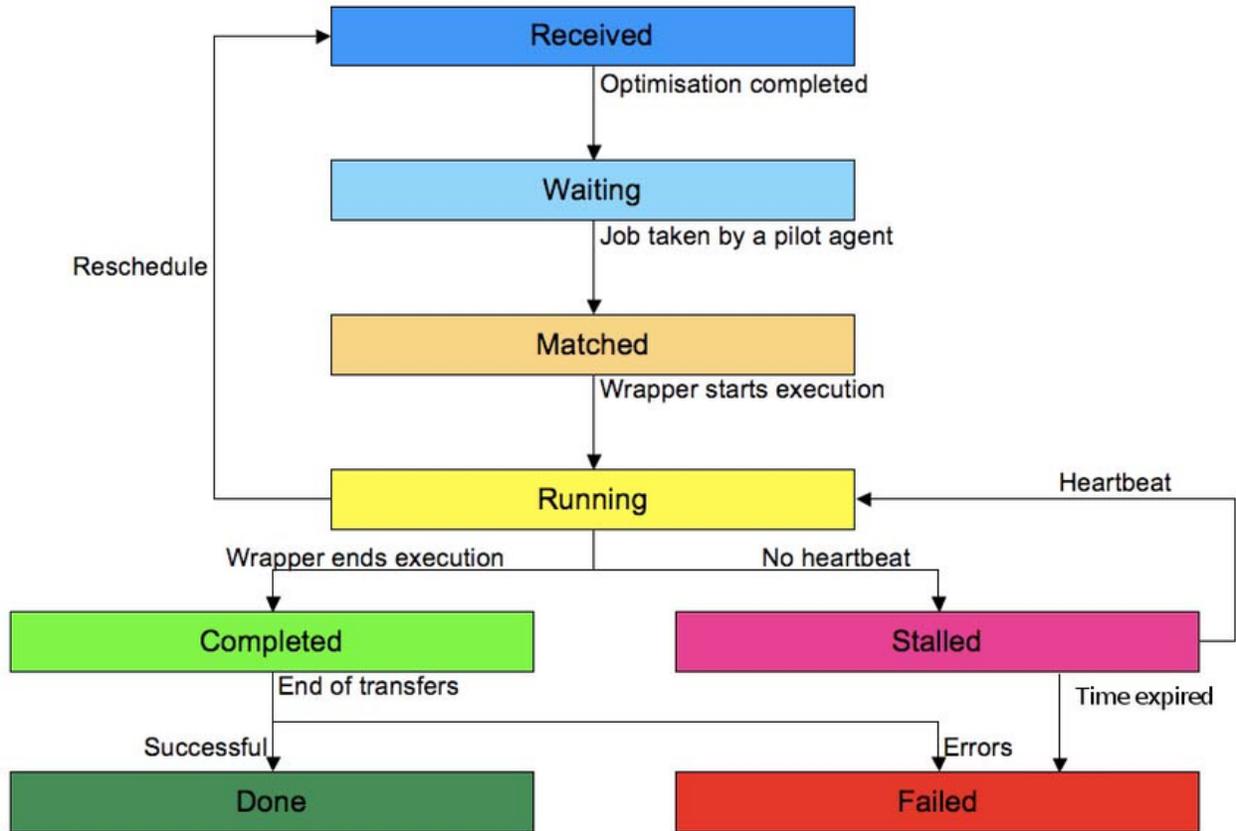


Fig. 3.1 – Les différents états d'un job DIRAC

---

## Utilisation avancée de DIRAC

---

Cette section détaille la soumission de tâches paramétriques.

### 4.1 Soumission d'un ensemble paramétrique de calculs

La soumission d'une tâche de type *Parametric* est la soumission d'un ensemble de calculs dont les fichiers JDL sont identiques, mais qui pourront accéder à une valeur paramétrable unique à chaque calcul. Ce type de tâche est très utile lorsque la soumission d'une série de calculs ne diffèrent que par la valeur d'un paramètre (par exemple, un paramètre numérique utilisé dans le nom des fichiers d'entrée et de sortie).

```
JobName = "Test_param_%n";
JobGroup = "Param_Test_1";
Executable = "/bin/sh";
Arguments = "myparamscript.sh %s";
Parameters = 10;
ParameterStart = 0;
ParameterStep = 1;
StdOutput = "stdout.out";
StdError = "stderr.err";
InputSandbox = { "myscript.sh" };
OutputSandbox = { "stdout.out", "stderr.err" };
VirtualOrganisation = "vo.grand-est.fr";
```

Ce document JDL va générer 10 jobs. Les paramètres suivants peuvent être utilisés dans le fichier JDL :

- %n : numéro de l'itération
- %s : valeur du paramètre
- %j : identifiant des jobs

```
$ dirac-wms-job-submit params.jdl
JobID = [79343433, 79343434, 79343435, 79343436, 79343437, 79343438, 79343439, ↵
↵79343440, 79343441, 79343442]
```



---

## Utilisation des commandes COMDIRAC

---

Les commandes **COMDIRAC** offre un accès simplifié et efficace au système DIRAC. Elles permettent de gérer plus simplement les jobs et les transferts, sous de multiples VOs.



## CHAPITRE 6

---

### Documentation complémentaire

---

Les liens ci-dessous permettent d'accéder à de la documentation complémentaire pour approfondir l'utilisation de DIRAC et de la grille de calcul :

- [support de cours pour la formation utilisateur DIRAC](#) ;
- [documentation officielle DIRAC](#) ;
- [documentation DIRAC sur le site France Grilles](#).