

# Запуск задачи в Grid

А. Кирьянов  
ОИТА ПИЯФ



# Последовательность действий для регистрации среде LCG

Детально описана на странице <http://lcg-registrar.cern.ch/>

Основные шаги:

- 1) Ознакомиться с правилами использования ресурсов LCG;
- 2) Получить персональный цифровой сертификат;
- 3) Зарегистрироваться в соответствующей виртуальной организации.

# Работа с Grid – интерфейс ПОЛЬЗОВАТЕЛЯ

Два основных вида интерфейсов: командная строка (CLI) и графический (GUI, Web-GUI). Преимущества Web-интерфейса:

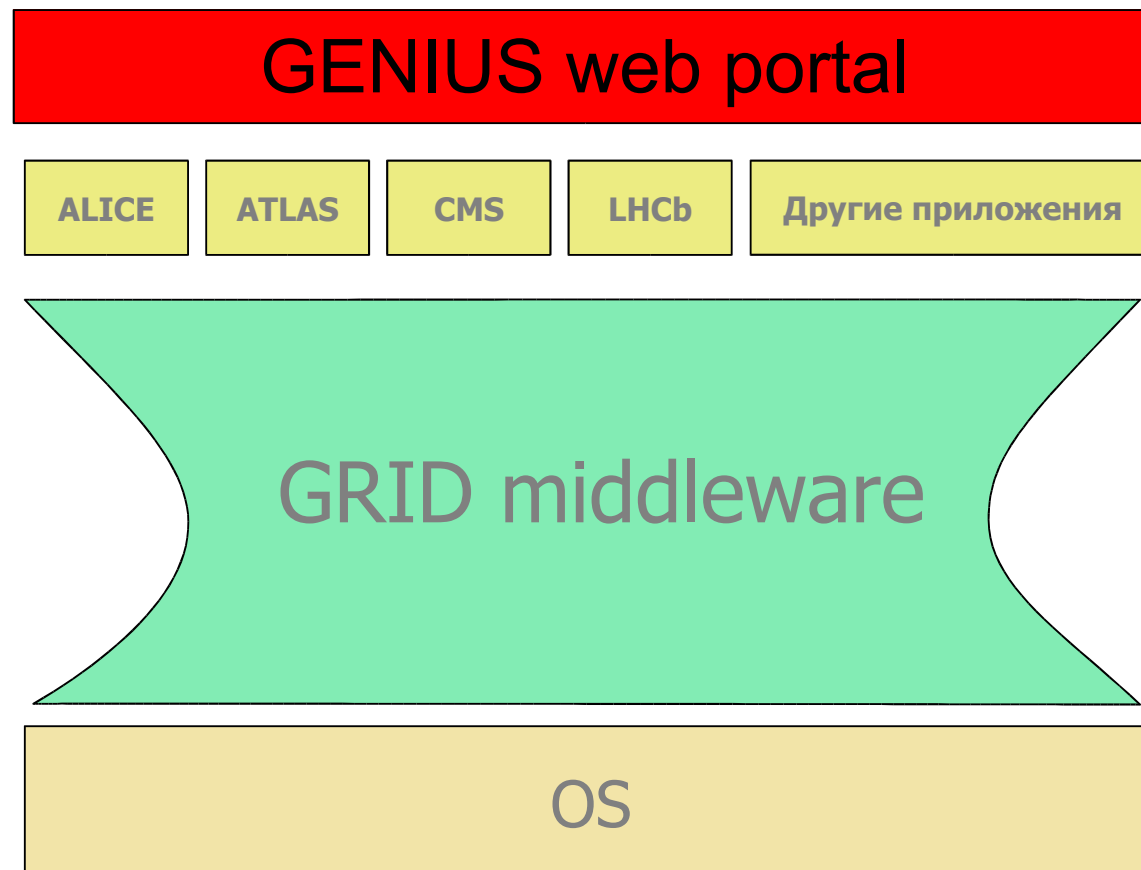
- Доступ с любого компьютера, не зависимо от платформы и ОС;
- Интерфейс не зависит от версии и типа ПО Grid (middleware);
- Безопасность на всех уровнях – использование только защищенных каналов передачи данных.
- Все доступные сервисы Grid удобно структурированы, доступ к ним возможен простым щелчком “мыши”;
- Интерфейс прост и доступен даже неопытному пользователю.

# Web-интерфейс GENIUS

**GENIUS**<sup>®</sup>

(**G**rid **E**nabled web **eN**vironment for  
site **I**ndependent **U**ser job **S**ubmission)

<https://genius.ct.infn.it>



# Получение персонального цифрового сертификата X.509

```
$ grid-cert-request -cn "Ivan Petrov"
```

```
A certificate request and private key is being created.  
You will be asked to enter a PEM pass phrase.  
This pass phrase is akin to your account password,  
and is used to protect your key file.  
If you forget your pass phrase, you will need to  
obtain a new certificate.
```

```
Using configuration from /etc/grid-security/globus-user-ssl.conf  
Generating a 1024 bit RSA private key  
.....++++++  
.....++++++  
writing new private key to '/root/.globus/userkey.pem'  
Enter PEM pass phrase: *****  
Verifying password - Enter PEM pass phrase: *****
```

# Получение персонального цифрового сертификата X.509

A private key and a certificate request has been generated with the subject:

```
/C=RU/O=DataGrid/OU=pnpi.nw.ru/CN=Ivan Petrov
```

После выполнения этой команды в домашнем каталоге создается новый подкаталог **.globus** с тремя файлами:

- **usercert\_request.pem** – запрос на получение цифрового сертификата;
- **usercert.pem** – цифровой сертификат пользователя (часть с открытым ключом), размер этого файла пока равен 0;
- **userkey.pem** – секретный ключ сертификата пользователя.

Запрос на получение сертификата следует направить ответственному по сертификации в Вашей организации. Далее произойдет регистрация Вашего сертификата в Российском центре сертификации (на данный момент он находится в НИИЯФ МГУ). По завершению процесса регистрации Вы получите по электронной почте свой цифровой сертификат, который следует сохранить в файле **usercert.pem**

# Начало работы: создание проху

Перед началом работы с LCG Вам необходимо создать так называемый проху-сертификат:

```
$ grid-proxy-init
Your identity: /C=RU/O=DataGrid/OU=pnpi.nw.ru/CN=Ivan Petrov
Enter GRID pass phrase for this identity: ****
Creating proxy ..... Done
Your proxy is valid until: Thu Sep  2 23:36:15 2004
```

По умолчанию время жизни такого сертификата равно 12 часам. Для получения информации о проху-сертификате можно воспользоваться следующей командой:

```
$ grid-proxy-info
subject   : /C=RU/O=DataGrid/OU=pnpi.nw.ru/CN=Ivan Petrov/CN=proxy
issuer    : /C=RU/O=DataGrid/OU=pnpi.nw.ru/CN=Ivan Petrov
identity  : /C=RU/O=DataGrid/OU=pnpi.nw.ru/CN=Ivan Petrov
type      : full legacy globus proxy
strength  : 512 bits
path      : /tmp/x509up_u500
timeleft  : 11:59:51
```

# Долгоживущие проху-сертификаты

Задача не может выполняться дольше, чем время жизни проху-сертификата. Если необходимо запустить задачу на длительное время, то проху-сертификат можно зарегистрировать на сервере Муроху для автоматического обновления.

Регистрации проху-сертификата на сервере Муроху:

```
$ муроху-init -s <сервер> -t <время>
```

Получение проху-сертификата с сервера Муроху:

```
$ муроху-get-delegation -s <server>
```

Получение информации о зарегистрированном проху-сертификате:

```
$ муроху-info -s <server>
```

Отмена регистрации:

```
$ муроху-destroy -s <server>
```



Элементарный файл с описанием задания (*test.jdl*):

```
Executable = "/bin/echo" ;  
Arguments  = "Hello World" ;  
StdOutput  = "hello.out" ;  
StdError   = "hello.err" ;  
OutputSandbox = {"hello.out" ,  
"hello.err"} ;
```

# Возможности пользователя по управлению заданиями

- Пользователь управляет заданиями через подсистему управления загрузкой (Workload Management System - WMS);
- Основная задача WMS - **планирование и управление распределенными ресурсами в системе Grid;**
- Что может пользователь?
  - Посылать задачи на выполнение;
  - Выполнять задачи на наиболее подходящих для этого ресурсах (WMS автоматически оптимизирует использование ресурсов);
  - Получать информацию о состоянии задач;
  - Получать результаты выполнения задач.

# Команды управления заданиями

Получение списка подходящих ресурсов:

```
$ edg-job-list-match hello.jdl
```

Посылка задачи на выполнение:

```
$ edg-job-submit hello.jdl
```

Получение текущего статуса задачи:

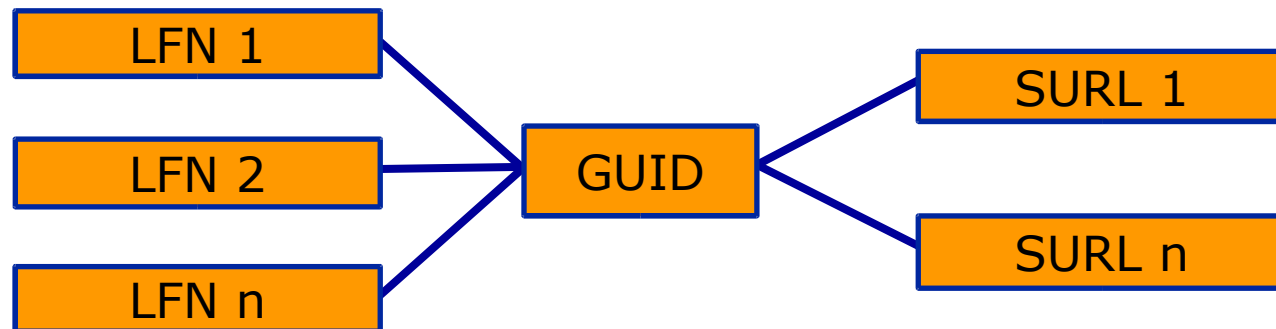
```
$ edg-job-status <JobID>
```

Получение результатов выполнения задачи:

```
$ edg-job-get-output <JobID>
```

# Управление данными: ОСНОВНЫЕ ПОНЯТИЯ

- Logical File Name (**LFN**)
  - Удобное для пользователя название конкретного файла с данными:  
“lfn:cms/20030203/run2/track1”
- Site URL (**SURL**) = Physical File Name (**PFN**)
  - Адрес физического расположения конкретного экземпляра данных:  
“srm://pcrd24.cern.ch/flatfiles/cms/output10\_1”
- Globally Unique Identifier (**GUID**)
  - Уникальный идентификатор конкретных данных: “guid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6”



# Команды управления данными

\$ **edg-rm** <команда> <параметры>

- **copyAndRegisterFile**
  - Скопировать файл на элемент хранения данных и зарегистрировать его в каталоге реплик.
- **replicateFile**
  - Реплицировать файл с одного элемента на другой с регистрацией в каталоге реплик.
- **deleteFile**
  - Удалить файл с элемента хранения данных и удалить соответствующую запись из каталога реплик.
- **registerFile**
  - Зарегистрировать файл в каталоге реплик.
- **unregisterFile**
  - Удалить запись из каталога реплик.

# Команды управления данными

- **registerGUID**

- Зарегистрировать SURL с известным GUID в каталоге реплик.

- **listGUID**

- Получить GUID для данного LFN или SURL.

- **listReplicas**

- Получить список всех реплик файла.

- **listBestFile**

- Получить список “лучших” реплик для данного LFN.

- **getBestFile**

- Реплицировать файл на ближайший элемент хранения данных.

- **copyFile**

- Скопировать файл.

- **listDirectory**

- Получить список файлов с данного элемента хранения данных.