

Chemia: QCG-SimpleClient (Gaussian)

Krótki opis usługi

Poniższy przykład opisuje niezbędne kroki do uruchomienia obliczeń kwantowo-chemicznych za pomocą programu Gaussian. Obliczenia będą wykonywane na maszynach udostępnionych w ramach infrastruktury [PL-Grid](#).

Aktywowanie usługi

Uruchamianie zadań możliwe jest w ramach różnych interfejsów graficznych do obsługi zadań. W poniższym przykładzie do uruchomienia zadania wykorzystano usługę [QCG](#). Poniżej opisane zostały najważniejsze kroki wymagane do rozpoczęcia obliczeń:

- **Rejestracja(założenie konta)**
Rejestracja dotyczy tylko osób które nie mają jeszcze konta w [portalu PL-Grid](#). [Opis rejestracji konta w Infrastrukturze PLGrid](#).
- **Wystąpienie o certyfikat**
Certyfikat jest niezbędny do uruchamiania zadań w PL-Grid. Do pracy wystarczy certyfikat SimpleCA. Sposób jego uzyskiwania opisany jest w [osobnym rozdziale](#) podręcznika.
- **Wystąpienie o grant obliczeniowy (opcjonalnie)**
Każdy użytkownik po założeniu konta otrzymuje automatycznie grant pilotażowy, który pozwala na szybkie rozpoczęcie pracy. W miarę prowadzenia obliczeń wskazane jest [wystąpienie o grant](#) tzw. "właściwy".
- **Aktywowanie usługi**
W celu wykonania poniższych ćwiczeń konieczne jest uzyskanie dostępu do usługi QCG (procedura aplikowania o dostęp do QCG jest opisana [tutaj](#)).

Pierwsze kroki

Cel ćwiczenia

Zasadniczym celem poniższego samouczka jest prezentacja różnych metod pozwalających na uruchomienie zadania obliczeniowego w programie Gaussian za pośrednictwem klienta konsolowego Simple usługi QCG. **Przykład 1** prezentuje w jaki sposób zlecić zadanie z wykorzystaniem mechanizmu obsługi aplikacji Gaussian w systemie QCG. **Przykład 2** pokazuje jak uruchamiać to samo zadanie za pośrednictwem ręcznego ładowania modułów. Główna różnica pomiędzy obiema metodami polega na dodatkowych akcjach jakie są wykonywane przez mechanizm obsługi aplikacji Gaussian w systemie QCG (**Przykład 1**). Wykonują one:

- automatyczne dostosowanie parametrów pamięciowych opisu zadania Gaussian do wymagań zdefiniowanych w zadaniu,
- konwersja pliku z opisem zadania do formatu UNIX,
- automatyczne wykonanie polecenia *formchk* na pliku wyjściowym checkpoint.

We wszystkich przykładach będziemy korzystać z przykładowego pliku wsadowego do programu Gaussian [sample.com](#) (celem obliczeń zdefiniowanych w tym pliku jest optymalizacja struktury i analiza częstotliwości drgań dla cząsteczki etanolu na poziomie B3LYP/3-21g).

Przykład 1 - uruchamianie zadania z pomocą interfejsu QCG Simple do aplikacji Gaussian

Do uruchamiania zadań wykorzystamy klienta konsolowego QCG Simple. W celu skorzystania z klienta należy zalogować się na maszynę dostępową z zainstalowanym oprogramowaniem klienckim. Aktualnie istnieje kilka interfejsów użytkownika (User Interface, UI) do infrastruktury. Dane dostępowe tych maszyn można znaleźć [tutaj](#). Z jedną z maszyn dostępowych należy połączyć się wykorzystując protokół ssh. W celu użytkownik o loginie `plguser` ponien w terminalu wydać polecenie:

```
$ ssh plguser@ui.plgrid.wcss.wroc.pl
```

Hasło na wszystkich UI jest takie jak w portalu PLGrid.

Następnie należy utworzyć w bieżącym katalogu plik [sample.com](#) będący plikiem wsadowym do programu Gaussian. Oprócz tego potrzebny jest plik specyfikujący uruchomienie zadania za pomocą klienta QCG. Przykładowy plik służący do uruchomienia zadania ([sample.qcg](#)) wygląda tak:

```
#QCG queue=plgrid-testing
#QCG name=etanol
#QCG note=etanol Gaussian
#QCG output=${JOB_ID}.output
#QCG error=${JOB_ID}.error
#QCG stage-in-file=sample.com -> sample.com
#QCG stage-out-file=wynik.tar -> ${JOB_ID}.tar
#QCG nodes=1:1:1
#QCG walltime=PT1H
#QCG memory=1800

#QCG application=g09
#QCG argument=sample.com
#QCG postprocess=tar cvf wynik.tar *
```

Pierwsze 10 linii w pliku służy do ustawienia nazwy zadania, wyboru kolejki i ilości pamięci i procesorów, które zadanie ma użyć oraz deklaruje czas potrzebny na wykonanie obliczeń. Linie ze słowami kluczowymi stage-in-file i stage-out-file służą do wstazania pliku, który ma zostać skopiowany na węzeł obliczeniowy (sample.com) i pliku, który ma być skopiowany na UI po zakończeniu obliczeń (plik wynik.tar zostanie skopiowany pod nazwą \${JOB_ID}.tar do naszej bieżącej lokalizacji, gdzie \${JOB_ID} jest identyfikatorem zadania QCG. Linia 11 w przykładowym pliku konfiguracyjnym jest pusta. Natomiast w 12 linii zdefiniowano, że ma być wykonane zadanie obliczeniowe wykorzystujące aplikację Gaussian09. Następną linią definiuje argument, z jakim aplikacja g09 ma być uruchomiona. W tym przypadku będzie to plik wsadowy do obliczeń Gaussian. Ostatnia linia pliku tworzy z katalogu roboczego na węźle obliczeniowym, na którym zadanie wykonano, archiwum o nazwie wynik.tar, zawierające wszystkie pliki. W celu uruchomienia obliczeń wystarczy teraz wydać polecenie:

```
wcss [plguser@ui ~]$ qcg-sub sample.qcg
```

Zadanie zostanie zlecone przez klienta QCG. Ponieważ nie wskazano konkretnego klastra obliczeniowego, na którym zadanie ma się wykonać, broker QCG sam wybierze klastę z dostępnymi zasobami i tam uruchomi zadanie. Do sprawdzenia statusu zadania możemy wykorzystać polecenie qcg-list. Po zakończeniu obliczeń w naszym katalogu domowym na UI pojawi się archiwum z zawartością katalogu roboczego z węzła obliczeniowego po zakończeniu obliczeń.

Przykład 2 - uruchomienie zadania z wykorzystaniem mechanizmu modułów

W poprzednim przykładzie nie wskazano klastra na którym chcemy wykonać obliczenia i nie podano wersji oprogramowania Gaussian, z której chcemy skorzystać. W rezultacie broker QCG wybrał klastę z dostępnymi zasobami i tam wysłał zadanie. Zostało ono uruchomione z pomocą wersji programu Gaussian09 ustawionej na tym klastrze jako domyślna. Jeżeli istnieje potrzeba kontrolowania wersji oprogramowania i klastra, na którym zadanie ma się wykonać możemy skorzystać z mechanizmu modułów. (Ten przykład może okazać się przydatny również gdy chcemy uruchomić obliczenia w aplikacji nie wspieranej przez klienta QCG.)

Plik opisujący zadanie stosujące technologię modułów może wyglądać tak ([sample.module.qcg](#)):

```
#QCG queue=plgrid-testing
#QCG name=et_modul
#QCG note=etanol Gaussian
#QCG output=${JOB_ID}.output
#QCG error=${JOB_ID}.error
#QCG stage-in-file=sample.com -> sample.com
#QCG stage-out-file=wynik.tar -> ${JOB_ID}.tar
#QCG nodes=1:1:1
#QCG host=bem
#QCG persistent
#QCG walltime=PT1H
#QCG memory=1900

#QCG postprocess=tar cvf wynik.tar *

module load gaussian/g09.B.01
cat sample.com | g09 >& sample.log
```

W powyższym przykładzie należy zwrócić uwagę na dwie ostatnie linie. Są to komendy, które zostaną uruchomione na węźle obliczeniowym po uzyskaniu wolnych zasobów. Przedostatnia linia pliku służy do załadowania modułu gaussian/09.B.01. Dzięki temu można mieć pewność, że uruchomiony zostanie Gaussian09 w wersji B.01. Ostatnia linia służy do uruchomienia zadania i przekierowuje jego wyników do pliku sample.log.

Należy również zwrócić uwagę na linię ze słowem kluczowym QCG host. Pozwala ona na kontrolowanie, na którym klastrze zadanie się wykona. W tym wypadku obliczenia zostaną uruchomione na klastrze bem.wcss.wroc.pl (alias bem).

O czym warto pamiętać

Narzędzia i usługi QCG oferują dodatkową funkcjonalność jak na przykład monitorowanie stanu zadania ([usługa QCG-Monitoring](#)).

Pełny zestaw narzędzi i ich zakres funkcjonalności opisany jest na stronie ([Uruchamianie zadań przez QosCosGrid](#)).

Dalsze informacje

Pełna informacja dotycząca narzędzia QCG-SimpleClient dostępna jest w osobnym rozdziale - [QCG-SimpleClient](#).