

Przetwarzanie dużych zbiorów danych

Krótki opis usługi

Usługa umożliwia realizację zadań obliczeniowych dla bardzo dużych zbiorach danych lub też wielu bardzo małych zadań w jak najkrótszym czasie. Budowane środowisko umożliwia złożone analizy i obliczenia w takich problemach jak:

- analiza wydźwięku dla danych tekstowych,
- przetwarzanie języka naturalnego,
- analityka marketingowa i analiza danych konsumenckich,
- przetwarzanie sieci złożonych, w tym np. sieci społeczne, sieci ekonomiczne, sieci transportowe,
- uczenie maszynowe i wspomaganie decyzji.

W ramach usługi można także dokonywać wizualizacji otrzymanych wyników badań. Przykładem takich wizualizacji mogą być grafiki prezentujące słowa nacechowane pozytywnie czy negatywnie (rysunki poniżej, im większy rozmiar czcionki danego słowa tym bardziej pozytywne/negatywne jest dane słowo). Na rysunkach przedstawiony jest wynik budowania słownika wykorzystywanego podczas analizy wydźwięku dla języka naturalnego.

Chmura słów - pozytywne zwroty



Chmura słów - negatywne zwroty



Ograniczenia w korzystaniu

Proszę sprawdzić czy aplikowano o wszystkie niezbędne usługi w Portalu PLGRID. Więcej wymagań Complex Networks zamieszczono w [osobnym rozdziale](#).

Pierwsze kroki

Zlecenie zadań obliczeniowych z użyciem QCG

Należy wybrać jedną z opcji: **Zleć zadanie QCG (formularz do zlecenia)** albo **Edytor QCG (możliwość stworzenia własnego skryptu Python/bash do wykonania obliczeń)**

Przykładowy skrypt QCG dostępny tutaj: [qcg-test-bash.qcg](#)

Dodatkowy plik tekstowy, dla którego będzie wykonane zliczenie liczby różnych słów w nim występujących: [bible-kjv.txt](#)



PL-GRID

Polska Infrastruktura Gridowa została zbudowana w ramach projektu PL-Grid, w celu dostarczenia polskiej społeczności naukowej platformy informatycznej opartej na klastrach komputerów, służących e-Science w różnych dziedzinach. Infrastruktura wspiera badania naukowe poprzez integrację danych doświadczalnych i wyników zaawansowanych symulacji komputerowych prowadzonych przez geograficznie rozproszone zespoły. Infrastruktura PL-Grid umożliwia polskim naukowcom prowadzenie badań naukowych w oparciu o symulacje i obliczenia dużej skali z wykorzystaniem klastrów komputerów oraz zapewnia wygodny dostęp do rozproszonych zasobów komputerowych.



Zleć zadanie QCG

Complex Networks umożliwia zlecanie zadań obliczeniowych z wykorzystaniem poniżej przedstawionego formularza. Wszystkie parametry oraz ich znaczenie można znaleźć na stronie [Format opisu zadań](#).

Po wskazaniu parametrów zadania możemy je uruchomić klikając na przycisk **Zleć zadanie**.

Każde z uruchomionych przez Portal Complex Networks zadań można podejrzeć w zakładce **Zadania** -> **Moje zadania**.

Ważnym punktem jest możliwość zapisywania specyfikacji zadania w postaci Szablону - **Zapisz jako szablon**. Dzięki temu możemy zlecać podobne albo wręcz takie same zadania szybciej i prościej.

Complex NetworksZadaniaGridFTPRepozytoriumAnotatorŁukasz Augustyniak

Zleć zadanieZapisz jako szablon

PodstawoweZasobyPlikiŚrodowiskoPowiadomieniaInne

Aplikacja

Plik głównyWybierz

Plik wykonywalnyWybierz

Skrypt

Argumenty

Opis

Grant

Zleć zadanieResetujUkryj zaawansowane

Przykładowe zadanie zlecane przez formularz - zliczanie słów dla zadanego pliku

Należy przygotować dwa pliki - skrypt bash oraz tekstowy plik do zliczenia słów (nazwa pliku jest bardzo ważna w tym przypadku, ponieważ w skrypcie bash jest zapisana nazwa pliku przykładowego)

[qcg-test-bash.qcg](#)

[bible-kjv.txt](#)

Najpierw musimy dodać pliki na jedną z naszych maszyn dostępowych - możemy połączyć się do nich poprzez SSH [Dostęp do UI \(User Interface\)](#) albo o wykorzystanie GridFTP w Portalu Complex Networks.

Complex NetworksZadaniaGridFTPRepozytoriumAnotatorŁukasz Augustyniak

Complex Networks

Zbiór usług dziedzinowych z zakresu analiz sieciowych i przetwarzania języka naturalnego.

PL-GRID

Polska Infrastruktura Gridowa została zbudowana w ramach projektu PL-Grid, w celu dostarczenia polskiej społeczności naukowej platformy informatycznej opartej na klastrach komputerów, służących e-Science w różnych dziedzinach. Infrastruktura wspiera badania naukowe poprzez integrację danych doświadczalnych i wyników zaawansowanych symulacji komputerowych prowadzonych przez geograficznie rozproszone zespoły. Infrastruktura PL-Grid umożliwia polskim naukowcom prowadzenie badań naukowych w oparciu o symulacje i obliczenia dużej skali z wykorzystaniem klastrów komputerów oraz zapewnia wygodny dostęp do rozproszonych zasobów komputerowych.



Następnie tworzymy folder `QCG-test-script` i przechodzimy do niego.

Complex Networks

Zadania ▾

GridFTP

Repozytorium

Anotator

Łukasz Augustyniak ▾

ui.plgrid.wcss.wroc.pl

/ > home > grid > users > plglaugust

↺

★

Pokaż ukryte

☐	Nazwa	Rozmiar	Data
☐	 complex-networks	2,0 KB	25 luty 2015, 18:35
☐	 nltk_data	2,0 KB	24 list 2014, 12:24
☐	 old-PBS-scripts	2,0 KB	10 lip 2015, 11:06
☐	 QCG-test-script	2,0 KB	5 paź 2015, 20:46
☐	 qcg-testing	6,0 KB	1 paź 2015, 09:28
☐	 qsub-qstat	4,0 KB	8 gru 2014, 12:43
☐	 sentiment-test	2,0 KB	10 paź 2014, 12:27
☐	 cn-2.txt	0 bajtów	17 wrze 2015, 06:43
☐	 cn.txt	18,2 MB	17 wrze 2015, 06:40
☐	 plglaugust@zeus.cyfronet.pl	2,0 KB	21 wrze 2015, 23:52

Do niego dodajemy pobrane wcześniej pliki - klikamy w przycisk **Wgraj**.

Complex Networks

Zadania ▾

GridFTP

Repozytorium

Anotator

Łukasz Augustyniak ▾

ui.plgrid.wcss.wroc.pl

/ > home > grid > users > plglaugust > QCG-test-script

Pokaż ukryte

	Nazwa	Rozmiar	Data
🔍 Brak elementów			

Wgraj plik

twórz katalog

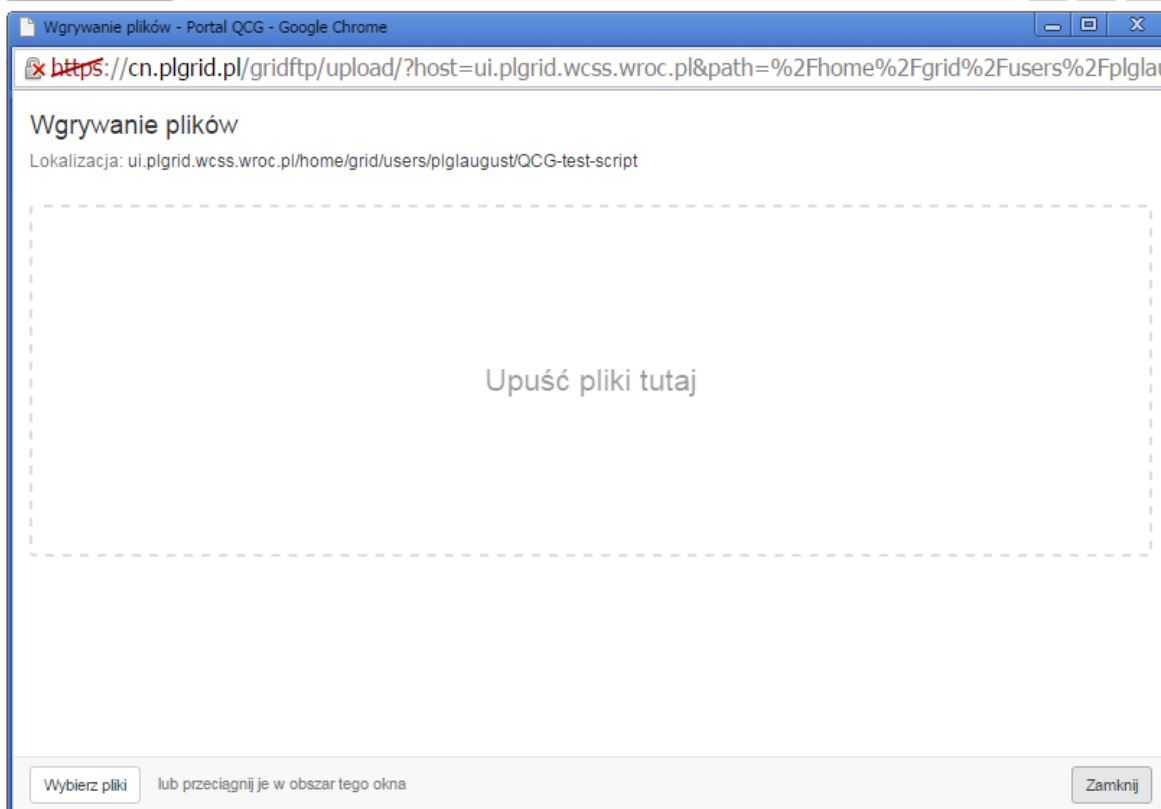
Zmień nazwę

Usuń

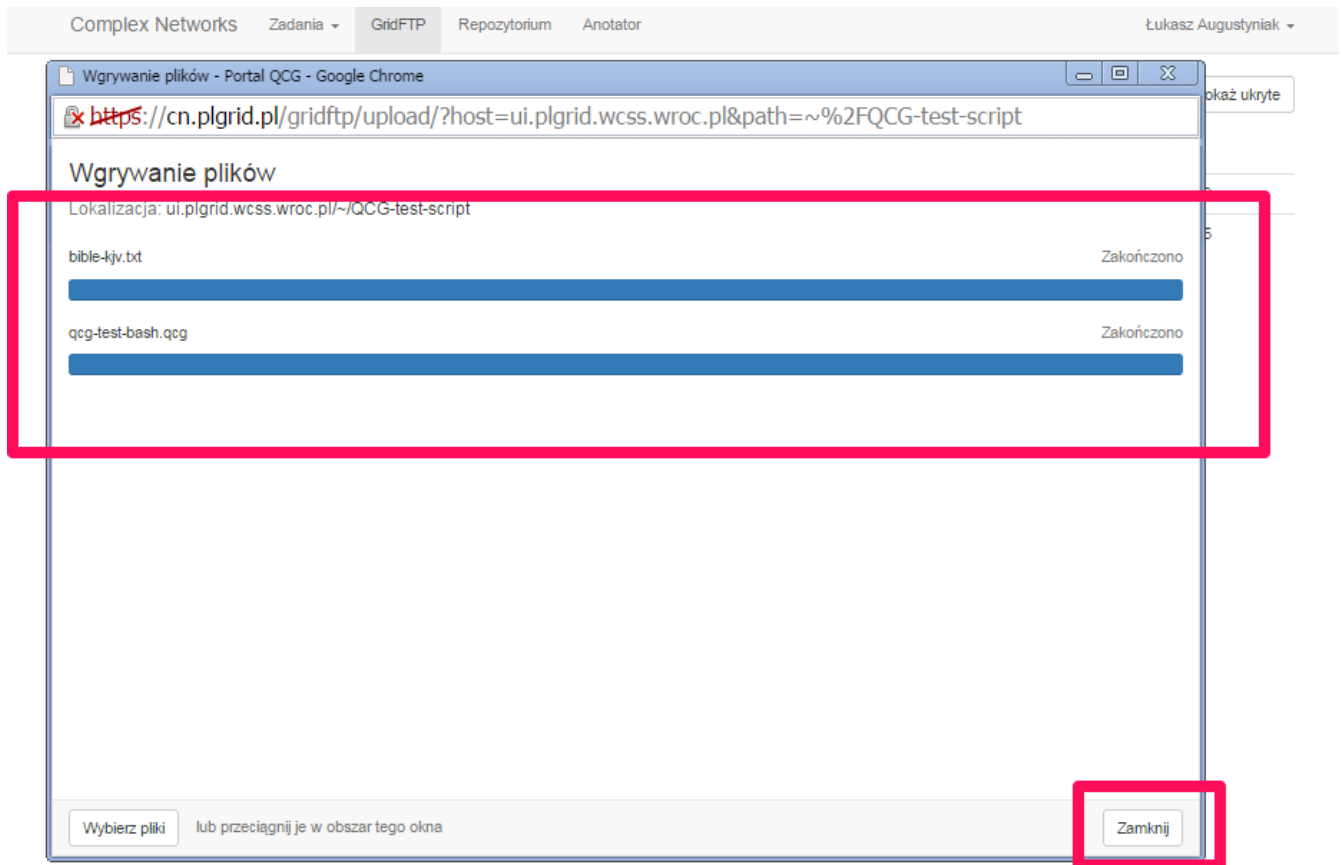
Spakuj

Rozpakuj

Pojawi się nowe okienko, do którego metodą przeciągnij/upuść albo poprzez kliknięcie w przycisk **Wybierz plik** dodajemy pliki potrzebne do uruchomienia zadania.



Poniżej widzimy poprawnie wgrane dwa pliki.



Następnie przechodzimy do formularza **zlecania zadań QCG**.



Wybieramy **Nowe zadanie**

Zlecanie zadań

Utwórz nowe zadanie

 lub

wybierz z listy zapisanych szablonów poniżej.

Zapisane szablony zadania

#	Nazwa	Data utworzenia	Data modyfikacji	
1	User's home	9 lipca 2015 12:12:49	9 lipca 2015 12:32:06	
2	Inula-Python-file	8 lipca 2015 10:50:43	1 października 2015 12:58:56	

Wybieramy aplikację **BASH** (skrypt bash zostanie użyty do uruchomienia zadania), a w polu **Plik główny** wskazujemy plik ze skryptem.

Complex Networks

Zadania ▾

GridFTP

Repozytorium

Anotator

Łukasz Augustyniak ▾

Zleć zadanie

Zapisz jako szablon

Podstawowe

Zasoby

Pliki

Środowisko

Powiadomienia

Inne

Aplikacja

BASH

Plik główny

Wybierz

Argumenty

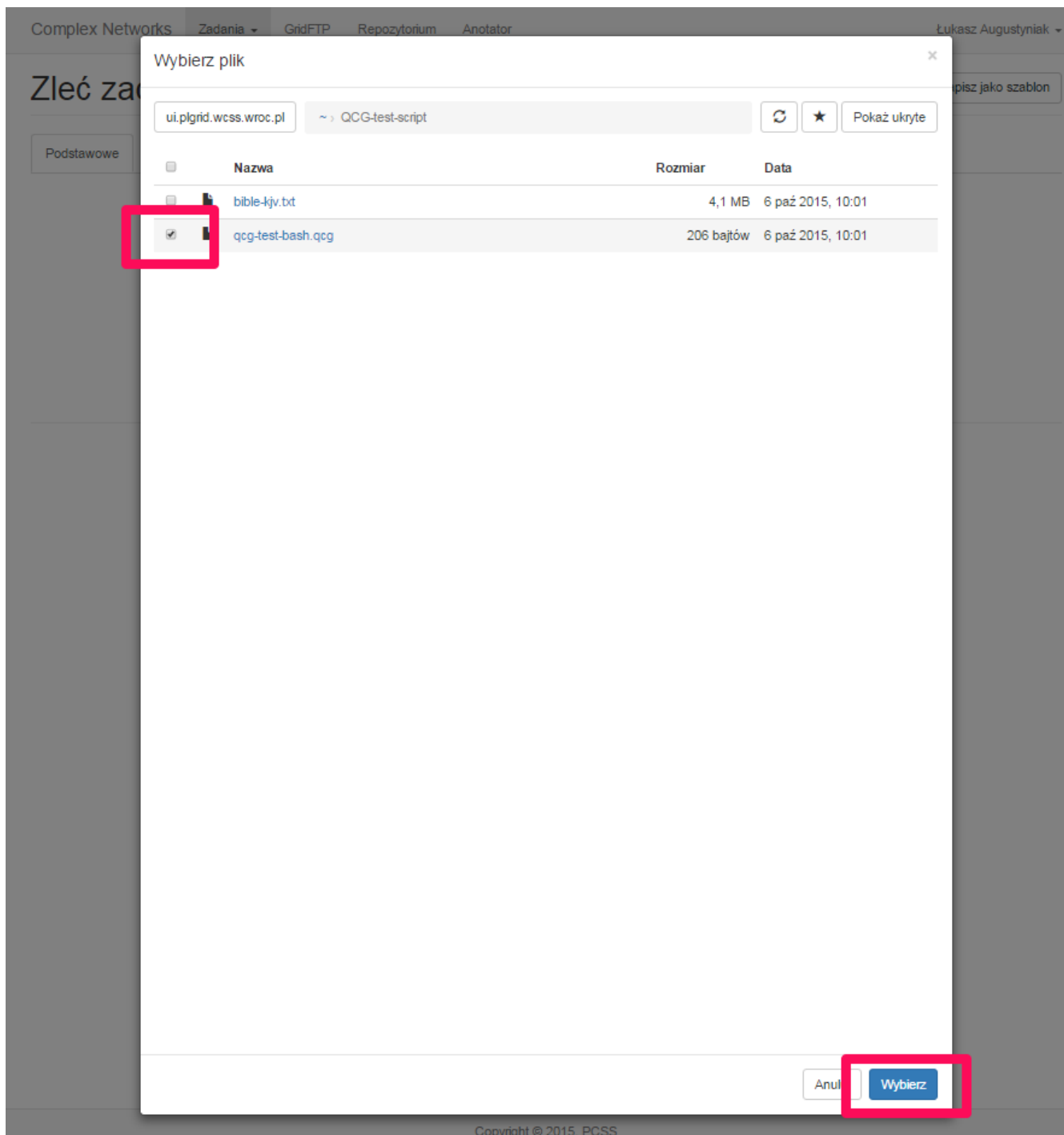
Opis

Zleć zadanie

Resetuj

Pokaż zaawansowane

Zaznaczamy wybór przy nazwie pliku i potwierdzamy przyciskiem **Wybierz**.



Następnie dodamy do zadania plik tekstowych, dla którego policzymy liczbę słów w nim występujących - zgodnie ze skryptem BASH. W tym celu klikamy na zakładkę **Plik** oraz wybieramy atrybut **Stage in**.

Complex NetworksZadaniaGridFTPRepozytoriumAnotatorŁukasz Augustyniak

Zleć zadanieZapisz jako szablon

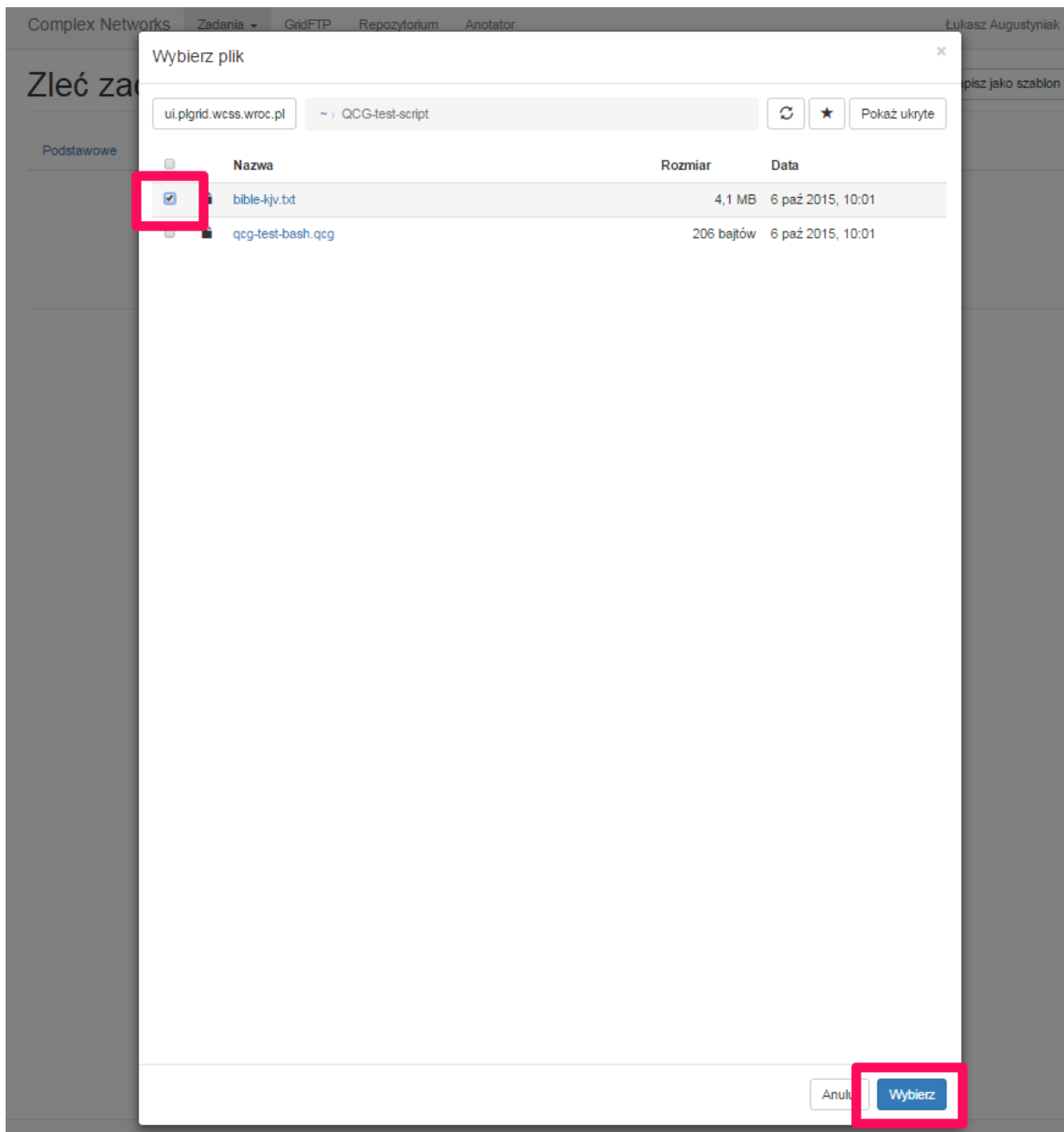
PodstawoweZasobyPlikiSkładowiskoPowiadomieniaInne

Standardowe wejścieWybierz

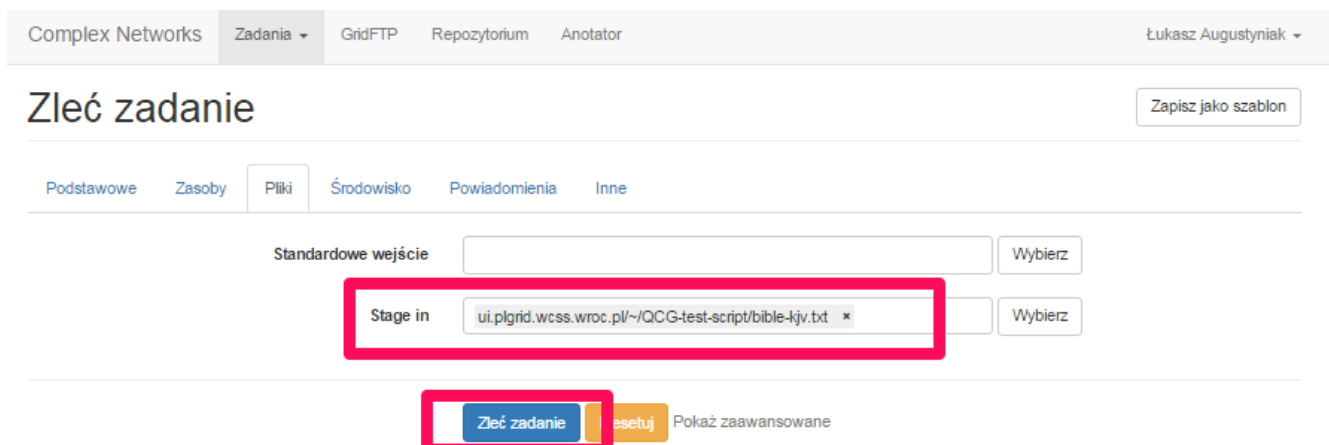
Stage inWybierz

Zleć zadanieResetujPokaż zaawansowane

Wskazujemy plik podobnie jak przy skrypcie bash.



W tym momencie przykładowe zadanie jest ustawione i można je zlecić przyciskiem **Zleć zadanie**.



W konsekwencji otrzymujemy wynik jak poniżej (podgląd pliku wyjściowego) - plik można znaleźć wybierając **Zadania -> Moje zadania** i wchodząc w **szczegóły** zleconego zadania oraz wybierając **Przeglądaj katalog roboczy**.

```
1 Test starts here
2 /home/grid/users/plglaugust
3 821133 bible-kjv.txt
4
```

Zlecanie zadań z wykorzystaniem Edytora QCG

Widok składa się co do zasady z 3 części:

1. Edytor kodu BASH/Python (lewa część interfejsu)
2. Sekcja wyboru przykładowych kodów (prawa górna część)
3. Sekcja wyboru zbioru danych (prawa dolna część)

Część pól z przedstawionego powyżej formularza pozostaje także i w tym widoku. Umożliwiają one dostosowanie zadania do naszych potrzeb. Bardzo ważnym jest wybór Aplikacji, tj. BASH albo Python, gdyż w ten sposób określamy jakiego rodzaju kod może zostać użyty do uruchomienia zadania.

W przykładzie poniżej widać wywołanie kodu BASH **wc -w AFINN-111.txt**. Należy zwrócić uwagę na wskazanie pliku **AFINN-111.txt**, plik ten został wybrany z repozytorium (po zleceniu zadania zostanie pobrany z repozytorium i przeniesiony na maszyny obliczeniowe - nie musimy nic dodatkowo kopiować!) Jeśli chcemy przenieść więcej plików należy je odnaleźć w repozytorium, a następnie pojedynczym kliknięciem myszy dodać do zadania. W celu dodania pliku w kodzie należy (po dodaniu go do zadania) ustawić kursor w odpowiednim miejscu kodu oraz kliknąć dwukrotnie na nazwę pliku (w poniższym przypadku można wybrać AFINN-111.txt albo license.txt w sekcji repozytorium). Usuwanie plików z zadania - kliknięcie krzyżyka po prawej stronie nazwy pliku.

Complex Networks

Zadania

GridFTP

Repozytorium

Anotator

Łukasz Augustyniak

Edytor kodu dla QCG

Podstawowe

Zaawansowane

Aplikacja

BASH

Styl

Eclipse

1

wc -w AFINN-111.txt

Przykłady kodu

+

Przetwarzanie tekstu

+

Testy pod klastry

GridFTP

Repozytorium

AFINN-111.txt

×

license.txt

×

Repozytorium

+

WT-tests

+

Python

+

Social Networks

+

Sentiment analysis AFINN-111 lexicon

5

AFINN-111.txt

license.txt

+

Syndication

Zleć zadanie

Resetuj

W celu wykonania zadania należy opakować kod Python w następującą strukturę:

python - <<END

kod python

END

Patrz przykład z przetwarzaniem danych sieciowych poniżej

Zlecenie zadania z wykorzystaniem przykładowych kodów - Python

Chcąc zlecić zadanie w języku Python należy zmienić typ Aplikacji z **BASH** na **Python**.

Następnie wyszukać interesujący nas fragment kodu, w przykładzie kod obliczający liczbę wystąpień poszczególnych słów w tekście - **word_count.py**

W celu podglądu zawartości kodu należy kliknąć na jego nazwę, w ten sposób dostaniemy dodatkowe okienko z podglądem.

The screenshot shows a web-based code editor interface. At the top, there are tabs for 'Complex Networks', 'Zadania', 'GridFTP', 'Repozytorium', and 'Anotator'. The main area is titled 'Edytor kodu dla C'. On the left, there's a sidebar with 'Podstawowe' and 'Zaawansowane' tabs, and a section for 'Aplikacja' with 'Python' selected. The main editor area shows a code snippet. A modal dialog box titled 'Word Count for Whole File' is open in the center, displaying the following Python code:

```
1 from collections import Counter
2 from re import split
3
4 BANNER = "-" * 35
5
6 def format_print(counter, is_reverse=False):
7     lst = counter.items()
8     lst.sort(key=lambda (a, b): (b, a), reverse=is_reverse)
9     print ("Unique Words: %d" % len(lst)).center(35, "=")
10    print ("%16s | %16s" % ("Word", "Count"))
11    print BANNER
12    for word, count in lst:
13        print ("%16s | %16d" % (word, count))
14
15 def count_words(filename):
16     counter = Counter()
17     with open(filename, "rU") as f:
18         for line in f:
19             line = line.strip().lower()
20             if not line:
21                 continue
22             counter.update(x for x in split("[a-zA-Z]+", line) if x)
23     return counter
24
25 format_print(count_words("filename.txt"), is_reverse=False)
```

At the bottom of the dialog, there are 'Zamknij' and 'Użyj' buttons. The background interface also shows a right sidebar with 'Styl' set to 'Eclipse', a list of code examples including 'word_count.py', and a file explorer showing 'AFINN-111.txt' and 'license.txt'.

Wybieramy przycisk **Użyj**, a kod zostanie skopiowany do Edytora.

Następnie musimy do zadania dodać z repozytorium plik, dla którego chcemy wykonać analizę oraz zmodyfikować kod Python w taki sposób, żeby wskazać tenże plik.

W tym celu wyszukujemy w repozytorium interesujący nas plik, dodajemy go do zadania (tak samo jak wybieraliśmy podgląd kodu przykładowego), a następnie wskazujemy miejsce w kodzie gdzie chcemy dodać plik i klikamy dwukrotnie na jego nazwę (prawie zaznaczenie na poniższym zrzucie ekranu).

W efekcie widzimy zmodyfikowany kod w Edytorze.

Edytor kodu dla QCG

Podstawowe

Zaawansowane

Aplikacja

Python ▾

Styl

Eclipse ▾

```
1 from collections import Counter
2 from re import split
3
4 BANNER = "-" * 35
5
6 def format_print(counter, is_reverse=False):
7     lst = counter.items()
8     lst.sort(key=lambda (a, b): (b, a), reverse=is_reverse)
9     print ("Unique Words: %d" % len(lst)).center(35, "=")
10    print "%-16s | %-16s" % ("Word", "Count")
11    print BANNER
12    for word, count in lst:
13        print "%-16s | %-16d" % (word, count)
14
15 def count_words(filename):
16     counter = Counter()
17     with open(filename, "rU") as f:
18         for line in f:
19             line = line.strip().lower()
20             if not line:
21                 continue
22             counter.update(x for x in split("[^a-zA-Z]+", line) if x)
23
24
25 format_print(count_words("AFINN-111.txt"), is_reverse=False)
```

Przykłady kodu

Przetwarzanie tekstu

Tokenizacja.py

word_count.py

+ Testy pod klastry

GridFTP

Repozytorium

AFINN-111.txt *

Repozytorium

+ WT-tests

+ Python

+ Social Networks

- Sentiment analysis AFINN-111

lexicon

- 5

AFINN-111.txt

license.txt

+ Syndication

Złóż zadanie

Resetuj

Warto zwrócić uwagę na zakładkę **Zaawansowane**, gdzie to możemy określić parametry wykonania zadania zgodnie z [Format opisu zadań](#). W przypadku zlecania zadań Python jest to bardzo ważne, ponieważ musimy wybrać odpowiedni moduł (z preinstalowanymi bibliotekami Python dla gridu Complex Networks).

W przedstawionym poniżej przypadku zlecamy zadanie obliczeniowe dla klastra **Zeus**, a to pociąga wybór modułu Python odpowiedniego właśnie dla tego klastra. W przypadku wyboru innego/dodatkowego klastra należy uwzględnić to w parametrze **Moduły**.

Edytor kodu dla QCG

Podstawowe

Zaawansowane

Podstawowe

Zasoby

Powiadomienia

Inne

Host

zeus.cyfronet.pl ×

Kolejka

plgrid-testing ▾

Właściwości węzłów

Moduły

tools/python/2.7.5 ×

Liczba procesów

Topologia węzłów

Wall time

Sekund ▾

Pamięć (MB)

Pamięci per proces (MB)

Rezerwacja

Złóż zadanie

Resetuj

Odpowiednie biblioteki Python są już pre-instalowane w następujących modułach:

Python-Task - plgrid/tools/python/2.7.9

Python-Zeus - tools/python/2.7.5

Python-Bem - plgrid/tools/python/2.7.8

Python-Inula - plgrid/tools/python/2.7.3

UWAGA!! Proszę nie usuwać zaznaczenia **TRWAŁE** z formularza specyfikacji parametrów zadania. W przeciwnym razie zadania zostaną wykonane na klastrze, ale nie zostaną zachowane i skopiowane z powrotem na maszyny lokalne, przez co nie będzie można zobaczyć wyników analiz!

Edytor kodu dla QCG

Podstawowe

Zaawansowane

Podstawowe

Zasoby

Powiadomienia

Inne

Preprocessing

Brak

Polecenie

Skrypt

Postprocessing

Brak

Polecenie

Skrypt

Opcje systemu kolejkowego

Trwałe ☒

Złóż zadanie

Resetuj

Zaawansowane użycie

Zlecanie zadania Python - wczytywanie danych sieciowych

W **Przykładach kodu** wybieramy **Przetwarzanie sieci**, a następnie **wczytywanie sieci**.

Edytor kodu dla QCG

Podstawowe

Zaawansowane

Aplikacja

BASH

Styl

Eclipse

Przykłady kodu

+ Przetwarzanie tekstu

+ Testy i Hosty

- Przetwarzanie sieci

Wczytywanie sieci

Generowanie sieci

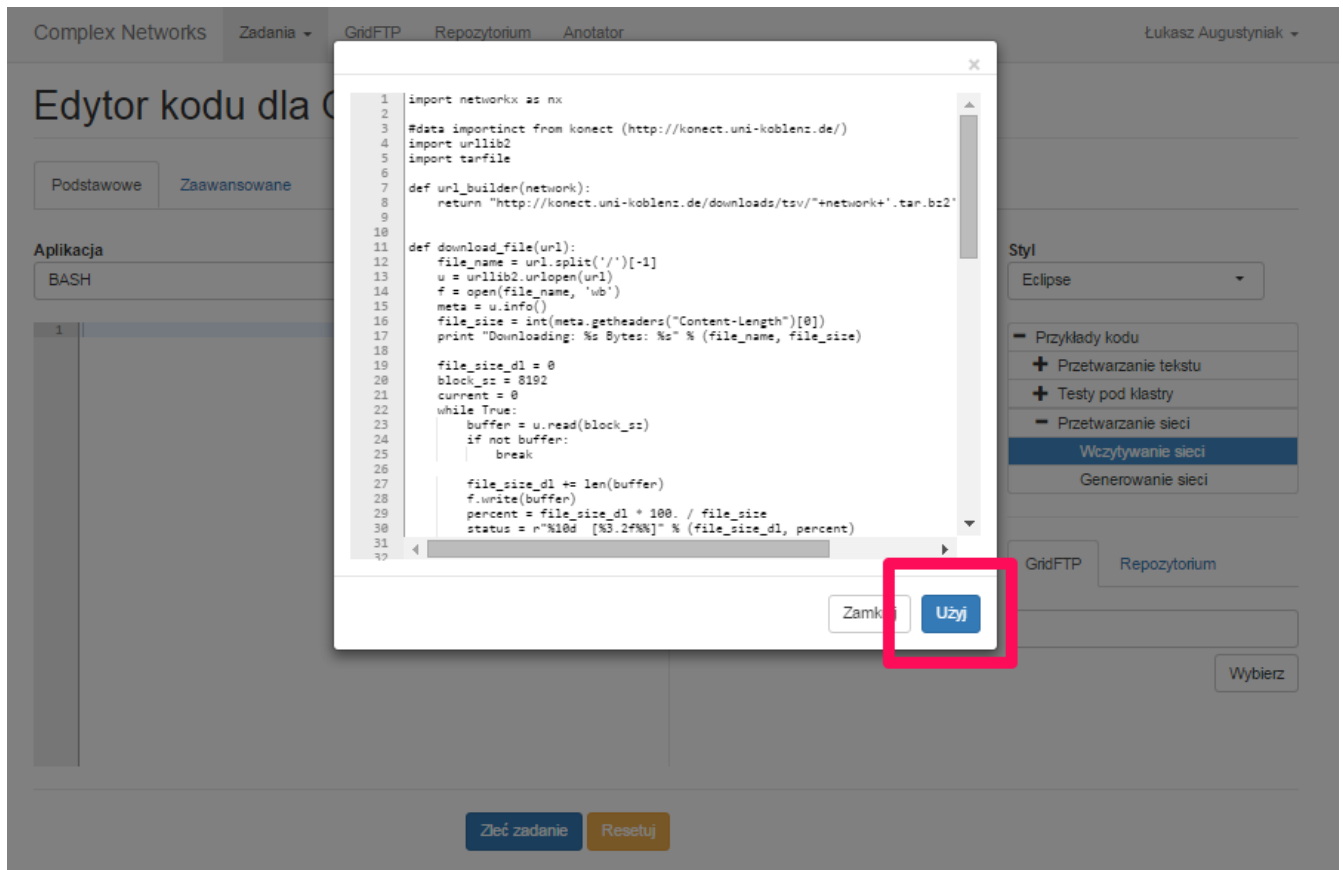
GridFTP

Repozytorium

Wybierz

Złóż zadanie

Resetuj



Dla klastrow innych niż Bem proszę dopisać kod **python - <<END** na początku skryptu, tak jak na poniższym przykładzie.

Edytor kodu dla QCG

Podstawowe

Zaawansowane

Aplikacja

BASH

Styl

Eclipse

```
1 python - <<END
3 import networkx as nx
4
5 #data importinct from konekt (http://konekt.uni-koblenz.de/)
6 import urllib2
7 import tarfile
8
9 def un1_builder(network):
10     return "http://konekt.uni-koblenz.de/downloads/tsv/"+network+".tar.bz2"
11
12
13 def download_file(url):
14     file_name = url.split('/')[-1]
15     u = urllib2.urlopen(url)
16     f = open(file_name, 'wb')
17     meta = u.info()
18     file_size = int(meta.getheaders("Content-Length")[0])
19     print "Downloading: %s Bytes: %s" % (file_name, file_size)
20
21     file_size_dl = 0
22     block_sz = 8192
23     current = 0
24     while True:
25         buffer = u.read(block_sz)
26         if not buffer:
27             break
28
29         file_size_dl += len(buffer)
30         f.write(buffer)
31         percent = file_size_dl * 100. / file_size
32         status = "%10d [%3.2f%%]" % (file_size_dl, percent)
33         status = status + chr(8)*(len(status)-1)
34         if ((percent)>current):
35             print status+"\n"
36             current=percent
37     print(f.name)
38     f.close()
39
40 def untar_file(infile):
41     tar = tarfile.open(infile,"r:bz2")
42     tar.extractall()
43     tar.close()
44
45 def remove_garbage_from_file(file_name):
46     f = open(file_name,"r")
47     lines = f.readlines()
48     f.close()
49
50     f = open(file_name,"w")
```

Przykłady kodu

+ Przetwarzanie tekstu

+ Testy pod klasty

- Przetwarzanie sieci

Wczytywanie sieci

Generowanie sieci

GridFTP

Repozytorium

Wybierz

Złć zadanie

Resetuj

oraz **END** na końcu kodu.

Complex Networks

Zadania ▾

GridFTP

Repozytorium

Anotator

Łukasz Augustyniak ▾

Edytor kodu dla QCG

Podstawowe

Zaawansowane

Aplikacja

BASH ▾

Styl

Eclipse ▾

```
38     f.close()
39
40 def untar_file(infile):
41     tar = tarfile.open(infile,"r:bz2")
42     tar.extractall()
43     tar.close()
44
45 def remove_garbage_from_file(file_name):
46     f = open(file_name,"r")
47     lines = f.readlines()
48     f.close()
49
50     f = open(file_name,"w")
51
52     for line in lines:
53         if not line.startswith( '%' ):
54             f.write(line)
55
56     f.close()
57
58 #building network from file
59 def build_network_from_file(dataset, classes_filename, network_type, network_name):
60
61     remove_garbage_from_file("./"+dataset+"/out."+dataset)
62
63     G=nx.Graph();
64     if(network_type == 'directed'):
65         G=nx.read_edgelist("./"+dataset+"/out."+dataset, create_using=nx.DiGraph(), data=data)
66     else:
67         G=nx.read_edgelist("./"+dataset+"/out."+dataset, create_using=nx.Graph())
68
69     G.name=network_name
70     return G
71
72
73 #use case
74
75 dataset = 'movielens-10m_ui'
76 network_type = 'directed'
77 url = "http://konect.uni-koblenz.de/downloads/tsv/"+dataset+".tar.bz2"
78 data= (('weight',float),('time',long))
79
80
81 download_file(url)
82 untar_file(dataset+".tar.bz2")
83 remove_garbage_from_file(dataset, '', network_type, dataset)
84 (G.order(),G.size())
85
86 END
87
```

GridFTP

Repozytorium

Wybierz

Zleć zadanie

Resetuj

Następnie określamy opis zlecanego zadania w zakładce **Zaawansowane** -> **Opis**.

Complex Networks

Zadania ▾

GridFTP

Repozytorium

Anotator

Łukasz Augustyniak ▾

Edytor kodu dla QCG

Podstawowe

Zaawansowane

Podstawowe

Zasoby

Powiadomienia

Inne

Opis

Testowe wczytanie sieci

Grant

Zleć zadanie

Resetuj

Wybieramy **Host (klastera)** oraz odpowiedni moduł Python dla klastra.

Odpowiednie biblioteki Python są już pre-instalowane w następujących modułach:

Python-Task - plgrid/tools/python/2.7.9
Python-Zeus - tools/python/2.7.5
Python-Bem - plgrid/tools/python/2.7.8
Python-Inula - plgrid/tools/python/2.7.3

Edytor kodu dla QCG

Podstawowe

Zaawansowane

Podstawowe

Zasoby

Powiadomienia

Inne

Host

zeus.cyfronet.pl ×

Kolejka

plgrid-testing ▾

Właściwości węzłów

Moduły

tools/python/2.7.5 ×

Liczba procesów

Topologia węzłów

Wall time

Sekund ▾

Pamięć (MB)

Pamięci per proces (MB)

Rezerwacja

Zleć zadanie

Resetuj

Wszystko gotowe i można zlecić zadanie.

Edytor kodu dla QCG

Podstawowe

Zaawansowane

Aplikacja

BASH ▾

Styl

Eclipse ▾

```
38     f.close()
39
40 def untar_file(infile):
41     tar = tarfile.open(infile,"r:bz2")
42     tar.extractall()
43     tar.close()
44
45 def remove_garbage_from_file(file_name):
46     f = open(file_name,"r")
47     lines = f.readlines()
48     f.close()
49
50     f = open(file_name,"w")
51
52     for line in lines:
53         if not line.startswith( '%' ):
54             f.write(line)
55
56     f.close()
57
58 #building network from file
59 def build_network_from_file(dataset, classes_filename, network_type, network_name):
60
61     remove_garbage_from_file("./"+dataset+"/out."+dataset)
62
63     G=nx.Graph();
64     if(network_type == 'directed'):
65         G= nx.read_edgelist("./"+dataset+"/out."+dataset, create_using=nx.DiGraph(), data=data)
66     else:
67         G= nx.read_edgelist("./"+dataset+"/out."+dataset, create_using=nx.Graph())
68
69     G.name=network_name
70     return G
71
72
73 #use case
74
75 dataset = 'movielens-10m_ui'
76 network_type = 'directed'
77 url = "http://konect.uni-koblenz.de/downloads/tsv/"+dataset+".tar.bz2"
78 data= (('weight',float),('time',long))
79
80
81 download_file(url)
82 untar_file(dataset+".tar.bz2")
83 G=build_network_from_file(dataset,'',network_type,dataset)
84 (G.order(),G.size())
85
86 END
87
```

- Przykłady kodu
- + Przetwarzanie tekstu
- + Testy pod klastry
- Przetwarzanie sieci
 - Wczytywanie sieci
 - Generowanie sieci

GridFTP

Repozytorium

Wybierz

Złóż zadanie

Resetuj

Zostajemy poinformowani o nazwie zlecanego zadania oraz przeniesieni do podglądu naszych wszystkich zlecanych zadań.

🔔 Zlecono zadanie J1444120338371__3023.

Lista zadań



< 1 > z 8

Opis	Miejsce	Start	Koniec	Status	Host	Typ	
Testowe wczytanie sieci	6 paź 2015, 10:32	6 paź 2015, 10:32		PREPROCESSING	nova	grid	szczegóły >
	6 paź 2015, 10:07	6 paź 2015, 10:07	6 paź 2015, 10:08	FINISHED	nova	grid	szczegóły >
	6 paź 2015, 10:06	6 paź 2015, 10:06		PENDING	inula	grid	szczegóły >
Test Networks	6 paź 2015, 09:45	6 paź 2015, 09:45	6 paź 2015, 09:46	FINISHED	nova	grid	szczegóły >
Test Networks	6 paź 2015, 09:42	6 paź 2015, 09:42	6 paź 2015, 09:43	FAILED	nova	grid	szczegóły >
	5 paź 2015, 20:22	5 paź 2015, 20:22	5 paź 2015, 20:23	FAILED	zeus	grid	szczegóły >
	5 paź 2015, 19:16	5 paź 2015, 19:16	5 paź 2015, 19:40	FINISHED	zeus	grid	szczegóły >
	5 paź 2015, 19:13	5 paź 2015, 19:13	5 paź 2015, 19:14	FAILED	zeus	grid	szczegóły >
	5 paź 2015, 19:12	5 paź 2015, 19:12	5 paź 2015, 19:35	FAILED	inula	grid	szczegóły >
Test	2 paź 2015, 12:26	2 paź 2015, 12:26	2 paź 2015, 12:26	FAILED	nova	grid	szczegóły >

1 2 3 4 5 z 8 następna >

Zlecanie zadania SPARK

Wchodzimy do zakładki **Zadania/Edytor QCG**

W polu Aplikacja wybieramy **bash**

Następnie wybieramy Przykład kodu **Spark distributed**

Complex NetworksZadaniaGridFTPRepozytoriumAnotatorŁukasz Augustyniak

Edytor kodu dla QCG

PodstawoweZaawansowane

AplikacjaBASH

StylEclipse

Przykłady kodu

+

Przetwarzanie tekstu

-

Testy pod klastry

scikit-learn+numpy.py

NLTK+pandas.py

Spark local

Spark distributed

+

Przetwarzanie sieci

GridFTP

Repozytorium

Wybierz

Złć zadanie

Resetuj

Complex NetworksZadaniaGridFTPRepozytoriumAnotatorŁukasz Augustyniak

Edytor kodu dla QCG

PodstawoweZaawansowane

AplikacjaBASH

StylEclipse

Przykłady kodu

+

Przetwarzanie tekstu

-

Testy pod klastry

scikit-learn+numpy.py

NLTK+pandas.py

Spark local

Spark distributed

+

Przetwarzanie sieci

GridFTP

Repozytorium

Wybierz

Złć zadanie

Resetuj

1#!/bin/env bash

2#module load test/spark

3

4start-multinode-spark-cluster.sh

5

6\$SPARK_HOME/bin/spark-submit --master spark://"hostname":7077 \

7--class org.apache.spark.examples.JavaWordCount \$SPARK_HOME/lib/spark-exam

8

9stop-multinode-spark-cluster.sh

Zamknij

Użyj

Na poniższym zrzucie widać przykładowy kod wywołujący zadanie z użyciem framework Spark - przykład zliczania słów w sposób rozproszony

Edytor kodu dla QCG

Podstawowe **Zaawansowane**

Aplikacja

BASH ▾

```
1 #!/bin/env bash
2 #module load test/spark
3
4 start-multinode-spark-cluster.sh
5
6 $SPARK_HOME/bin/spark-submit --master spark://"hostname":7077 \
7 --class org.apache.spark.examples.JavaWordCount $SPARK_HOME/lib/spark-examples*.jar /etc/passwd
8
9 stop-multinode-spark-cluster.sh
```

Styl

Eclipse ▾

Przykłady kodu

+ Przetwarzanie tekstu

- Testy pod klastry

scikit-learn+numpy.py

NLTK+pandas.py

Spark local

Spak distributed

+ Przetwarzanie sieci

GridFTP

Repozytorium

Wybierz

Złć zadanie

Resetuj

Następnie musimy wybrać odpowiedni klaster obliczeniowy (niestety nie wszystkie wspierają Spark). Przechodzimy do zakładki **Zaawansowane**

Complex Networks

Zadania ▾

GridFTP

Repozytorium

Anotator

Łukasz Augustyniak ▾

Edytor kodu dla QCG

Podstawowe **Zaawansowane**

Aplikacja

BASH ▾

```
1 #!/bin/env bash
2 #module load test/spark
3
4 start-multinode-spark-cluster.sh
5
6 $SPARK_HOME/bin/spark-submit --master spark://"hostname":7077 \
7 --class org.apache.spark.examples.JavaWordCount $SPARK_HOME/lib/spark-examples*.jar /etc/passwd
8
9 stop-multinode-spark-cluster.sh
```

Styl

Eclipse ▾

Przykłady kodu

+ Przetwarzanie tekstu

- Testy pod klastry

scikit-learn+numpy.py

NLTK+pandas.py

Spark local

Spak distributed

+ Przetwarzanie sieci

GridFTP

Repozytorium

Wybierz

Złć zadanie

Resetuj

Dalej wybieramy klaster **Zeus** oraz moduł **Spark**

Edytor kodu dla QCG

Podstawowe

Zaawansowane

Podstawowe

Zasoby

Powiadomienia

Inne

Host

zeus.cyfronet.pl ×

Kolejka

plgrid-testing ▾

Właściwości węzłów

Moduły

plgrid/apps/spark/1.5.0 ×

Liczba procesów

Topologia węzłów

Wall time

Sekund ▾

Pamięć (MB)

Pamięci per proces (MB)

Rezerwacja

W tym momencie mamy ustawione wszystkie niezbędne parametry do wywołania zadania. Możemy zlecić zadania klikając na przycisk **Zleć zadanie**

Edytor kodu dla QCG

Podstawowe

Zaawansowane

Podstawowe

Zasoby

Powiadomienia

Inne

Host	Zeus x
Kolejka	plgrid-testing
Właściwości węzłów	
Moduły	test/spark x
Liczba procesów	
Topologia węzłów	
Wall time	Sekund
Pamięć (MB)	
Pamięci per proces (MB)	
Rezerwacja	
	Zleć zadanie Resetuj

Status zadania możemy podejrzeć w zakładce **Zadania/Moje zadania**. Widzimy tutaj nazwę, numer, czas zlecenie oraz inne informacje związane z naszym zadaniem. Dodatkowo wybierając **szczegóły** możemy zobaczyć szczegółowe informacje dotyczące naszego zadania oraz wynik jego wykonania.

ⓘ Zlecono zadanie J1444069350161__9718.

Lista zadań



1



z 7

	Opis	Wystała	Start	Koniec	Status	Host	Typ	
		5 paź 2015, 20:22	5 paź 2015, 20:22		PREPROCESSING	zeus	grid	szczegóły >
		5 paź 2015, 19:18	5 paź 2015, 19:18	5 paź 2015, 19:40	FINISHED	zeus	grid	szczegóły >
		5 paź 2015, 19:13	5 paź 2015, 19:13	5 paź 2015, 19:14	FAILED	zeus	grid	szczegóły >
		5 paź 2015, 19:12	5 paź 2015, 19:12	5 paź 2015, 19:35	FAILED	inula	grid	szczegóły >
	Test	2 paź 2015, 12:26	2 paź 2015, 12:26	2 paź 2015, 12:26	FAILED	nova	grid	szczegóły >
	Test	2 paź 2015, 12:18	2 paź 2015, 12:18	2 paź 2015, 12:18	FAILED	nova	grid	szczegóły >
	For	2 paź 2015, 12:10	2 paź 2015, 12:10	3 paź 2015, 18:10	FAILED	inula	grid	szczegóły >
	For	2 paź 2015, 12:03	2 paź 2015, 12:03	2 paź 2015, 12:04	FINISHED	zeus	grid	szczegóły >
	Test	2 paź 2015, 12:01	2 paź 2015, 12:01	2 paź 2015, 12:02	FAILED	zeus	grid	szczegóły >
	Test	2 paź 2015, 12:01	2 paź 2015, 12:01	2 paź 2015, 12:01	FAILED	nova	grid	szczegóły >

1

2

3

4

5

z

7

następna >

🔔 Zlecono zadanie J1444069350161__9718.

Lista zadań



< 1 > z 7

Opis	Wysłane	Start	Koniec	Status	Host	Typ
	5 paź 2015, 20:22	5 paź 2015, 20:22		PREPROCESSING	zeus	grid szczegóły >
	5 paź 2015, 19:16	5 paź 2015, 19:16	5 paź 2015, 19:40	FINISHED	zeus	grid szczegóły >
	5 paź 2015, 19:13	5 paź 2015, 19:13	5 paź 2015, 19:14	FAILED	zeus	grid szczegóły >
	5 paź 2015, 19:12	5 paź 2015, 19:12	5 paź 2015, 19:35	FAILED	inula	grid szczegóły >
	Test 2 paź 2015, 12:26	2 paź 2015, 12:26	2 paź 2015, 12:26	FAILED	nova	grid szczegóły >
	Test 2 paź 2015, 12:18	2 paź 2015, 12:18	2 paź 2015, 12:18	FAILED	nova	grid szczegóły >
	For 2 paź 2015, 12:10	2 paź 2015, 12:10	3 paź 2015, 18:10	FAILED	inula	grid szczegóły >
	For 2 paź 2015, 12:03	2 paź 2015, 12:03	2 paź 2015, 12:04	FINISHED	zeus	grid szczegóły >
	Test 2 paź 2015, 12:01	2 paź 2015, 12:01	2 paź 2015, 12:02	FAILED	zeus	grid szczegóły >
	Test 2 paź 2015, 12:01	2 paź 2015, 12:01	2 paź 2015, 12:01	FAILED	nova	grid szczegóły >

1 2 3 4 5 z 7 następna >

Gdzie szukać dalszych informacji?

Więcej informacji można znaleźć na oficjalnej stronie usługi <https://cn.plgrid.pl>. Pytania proszę zadawać poprzez: <https://helpdesk.plgrid.pl>

Dodatkowo polecamy stronę [Helpdesk PLGrid](#)