**Chmura Obliczeniowa i Technologie Big Data**

*Moduł „PyPork” realizujący mechanizmy logiki rozmytej dla platformy Apache Pig*

|  |  |
| --- | --- |
| Kierownik | dr hab. inż. Dariusz Mrozek, Prof. PŚ |
| Autorzy | Jan Skowronek  Zuzanna Lempa  Paweł Kudzia  Wojciech Wątroba |
| Kierunek | Informatyka SSM |
| Specjalizacja | IDSI |
| Technologia | Apache Pig |

Spis treści

[1. Wstęp 3](#_Toc76203442)

[2.1. Cel projektu 3](#_Toc76203443)

[2.2. Założenia projektu 3](#_Toc76203444)

[2. Analiza tematu 4](#_Toc76203445)

[2.1. Platforma Hadoop 4](#_Toc76203446)

[2.2. Platforma Apache Pig 5](#_Toc76203447)

[2.3. Logika rozmyta 6](#_Toc76203448)

[3. Specyfikacja zewnętrzna 8](#_Toc76203449)

[3.1. Przygotowanie środowiska 8](#_Toc76203450)

[3.2. Przykład użycia biblioteki „PyPork” 10](#_Toc76203451)

[4. Specyfikacja wewnętrzna 11](#_Toc76203452)

[5. Podsumowanie 12](#_Toc76203453)

# 1. Wstęp

Big Data to termin odnoszący się do dużych i różnorodnych zbiorów danych. Przetwarzanie tego typu danych jest zadaniem wyjątkowo trudnym i wymagającym, ale z drugiej strony analiza ogromnych zbiorów okazuje się być wartościowa, ponieważ pozwala zdobyć nowy rodzaj wiedzy. Z tego też powodu na przestrzeni lat powstały narzędzia i rozbudowane platformy programistyczne umożliwiające m.in. efektywne gromadzenie, przechowywanie i przetwarzanie dużych ilości informacji. Warto zaznaczyć, że dane mogą pochodzić z wielu różnych źródeł, co niesie za sobą konieczność użycia specjalistycznego oprogramowania, za pomocą którego można usprawnić proces przetwarzania. Korzyści z używania narzędzi Big Data są niezaprzeczalne, dlatego istotne jest, aby istniała możliwość rozszerzania o nowe funkcjonalności i tak już zaawansowanych platform.

W ramach kursu *„Chmura Obliczeniowa i Technologie Big Data”* należy zrealizować własny moduł rozszerzający dla jednej z przykładowych platform programistycznych stosowanych w dziedzinie Big Data.

## 2.1. Cel projektu

Celem projektu jest wzbogacenie przykładowej platformy programistycznej o moduł, który umożliwi realizację rozmytego przetwarzania danych. Moduł ten należy opracować dla technologii Apache Pig będącej częścią ekosystemu Hadoop.

## 2.2. Założenia projektu

Nowy moduł dla Apache Pig powinien dostarczać podstawowe funkcjonalności rozmytego przetwarzania danych.

Główne założenia, które powinny zostać spełnione to:

1. Rozmyte filtrowanie w klauzuli WHERE,
2. Projekcja z rozmytą transformacją,
3. Tworzenie zmiennych lingwistycznych,
4. Grupowanie rozmyte względem zmiennej lingwistycznej,
5. Rozmyte złączenia poprzez:
   1. Zmienną lingwistyczną,
   2. Rozmywanie liczb.

Bibliotekę należy zaimplementować wykorzystując do tego UDF (*ang. User-defined function*), czyli funkcje rozszerzające dany program lub środowisko.

Ogromną zaletą ekosystemu Hadoop jest to, iż funkcje mogą zostać zaimplementowane w wybranych językach programowania, wspieranych przez aplikację Apache Pig. Jednym z nich jest język Python i to właśnie on posłużył do wykonania projektu.

# 2. Analiza tematu

W niniejszym rozdziale przedstawiono opis najważniejszych zagadnień teoretycznych związanych z projektem *„PyPork”*. Czytelnikowi przybliżone zostaną podstawowe informacje na temat dwóch bardzo popularnych rozwiązań Big Data, którymi są Hadoop i Apache Pig. Dodatkowo zaprezentowane zostaną kluczowe pojęcia z logiki rozmytej, których znajomość okazuje się być całkiem przydatna w wielu zastosowaniach.

## 2.1. Platforma Hadoop

Apache Hadoop jest otwartą platformą programistyczną napisaną w języku Java. Technologia ta służy do rozproszonego składowania i przetwarzania wielkich zbiorów danych. Z reguły tego typu przetwarzanie odbywa się przy pomocy większej liczby współpracujących ze sobą komputerów – grupa urządzeń to tzw. klaster. Zwiększenie liczby dostępnych węzłów obliczeniowych powoduje poprawę szybkości analizowania danych, co oczywiście ma znaczenie w przypadku, gdy danych jest bardzo dużo i dodatkowo pochodzą z wielu różnych źródeł. Istotną kwestią jest również to, iż platforma Hadoop wchodzi w skład grupy projektów rozwijanych przez fundację Apache.

Platforma Hadoop składa się z następujących modułów:

* Hadoop Common – typowe narzędzia, które używane są przez pozostałe moduły,
* Hadoop Distributed File System (HDFS) – rozproszony system plików, zapewnia dostęp do danych wykorzystywanych przez aplikację,
* Hadoop YARN – pozwala na planowanie zadań i zarządzanie zasobami klastra,
* Hadoop MapReduce – system do równoległego przetwarzania dużych ilości danych, implementacja paradygmatu MapReduce,
* Hadoop Ozone – magazyn obiektowy.

Warto zaznaczyć, iż samo określenie Hadoop utożsamiane jest z pewnym ekosystemem w skład którego wchodzi wiele dodatkowych aplikacji. Można do nich zaliczyć m.in.:

* Apache Pig,
* Apache Spark,
* Apache Storm,
* Apache Hive,
* Apache HBase,
* Apache Phoenix,
* Apache ZooKeeper,
* Apache Flume,
* Apache Sqoop,
* Apache Oozie,
* Cloudera Impala.

Jak wspomniano wcześniej do wykonania modułu *„PyPork”* wykorzystano Apache Pig, który jak się okazuje w stosunkowo prosty sposób pozwala zwiększyć dużą liczbę funkcjonalności technologii Hadoop.

## 2.2. Platforma Apache Pig

Apache Pig to platforma do analizy dużych zbiorów danych. Udostępnia język wysokiego poziomu przeznaczony do konstruowania programów operujących na danych, które mogą pochodzić z wielu źródeł. Dzięki opisywanej technologii można jednocześnie przetwarzać informacje, które pochodzą z relacyjnych bazy danych, plików tekstowych, czy plików CSV (*ang. comma-separated values*), niemniej często zdarza się, iż dane nie posiadają ściśle zdefiniowanej struktury. Obszar zainteresowań Big Data dotyczy zarówno informacji w pełni ustrukturyzowanych, jak i nieustrukturyzowanych. Pierwszy rodzaj danych pochodzi przede wszystkim z klasycznych systemów zarządzania relacyjnymi bazami danych, zaś dane nieustrukturyzowane to wszelkiego rodzaju wiadomości e-mail, wpisy ze stron internetowych (np. artykuły z blogów), jak i również treść generowana w mediach społecznościowych. Ponadto warto wspomnieć o istnieniu częściowo ustrukturyzowanych danych, do których zaliczają się wspomniane wcześniej pliki tekstowe, czy chociażby dzienniki systemowe.

Istotną zaletą programów napisanych w oparciu o technologię Apache Pig jest to, że wykonywane zadanie można w prosty sposób zrównoleglić, ponieważ całość odbywa się dzięki zastosowaniu systemu Hadoop. Podstawowym elementem infrastruktury Apache Pig jest kompilator, którego zadanie sprowadza się do tworzenie sekwencji programów Map-Reduce. Ponadto platforma Apache Pig dostarcza proceduralny język programowania Pig Latin, do którego zalet można zaliczyć m.in.:

* Łatwość tworzenia kodu. Wynika to z faktu, iż Pig Latin zaprojektowano w taki sposób, aby mógł być używany przez osoby, które na co dzień nie pracują z językiem Java lub posiadają niewielkie doświadczenie programistyczne,
* Duże możliwości optymalizacji, dzięki czemu zadania mogą być automatycznie optymalizowane przez moduły niższego poziomu, co pozwala użytkownikowi w pełni skupić się na określeniu oczekiwanego rozwiązanie zamiast poświęcać czas na próbie napisania bardzo wydajnego kodu,
* Możliwość tworzenia własnych rozwiązań i modułów poprzez zdefiniowanie funkcji specjalnego przeznaczenia.

Funkcje rozszerzające pozwalają wzbogacić standardowy zestaw funkcjonalności języka Pig Latin. W tym celu Apache Pig wspiera aż sześć języków programowania:

* Java,
* Jython,
* Python,
* JavaScript,
* Ruby,
* Groovy.

W chwili pisania pracy największe wsparcie dostępne jest tylko i wyłącznie dla języka Java. Dzięki własnym funkcjom możliwe jest dostosowanie etapów potoku przetwarzania danych tj. ładowaniem, przechowywanie, czy transformację danych, jak i również ich agregację.

## 2.3. Logika rozmyta

Logika rozmyta stanowi rozszerzenie tradycyjnej logiki binarnej i pozwala na opisywanie zjawisk fizycznych w bardziej naturalny sposób dla człowieka. Zazwyczaj w życiu codziennym opis różnych zjawisk nie opiera się na ścisłym definiowaniu parametrów. Oczywiście dokładna wiedza na temat np. wzrostu pacjenta, czy aktualnej temperatury jest ważna, jednak w praktyce bardzo często rezygnuje się z tej precyzji na rzecz zdecydowanie mniej precyzyjnych określeń jak np. *„ten mężczyzna jest wysoki”* lub *„dzisiaj jest bardzo ciepło”*.

Logika rozmyta okazała się przydatna w różnych zastosowaniach inżynierskich, a przede wszystkim w takich, gdzie klasyczna logika dokonująca klasyfikacji na podstawie dwóch wartości logicznych (tzn. prawda i fałsz) nie potrafi skutecznie poradzić sobie z wieloma sprzecznościami i niejednoznacznymi stwierdzeniami. Przykłady użycia logiki rozmytej to np. zastosowanie elementów teorii zbiorów rozmytych w elektronicznych systemach sterowania, czy w zadaniach związanych z eksploracją danych. Ponadto elementy logiki rozmytej można wykorzystać przy opracowywaniu zaawansowanych systemów ekspertowych, które na podstawie pewnej bazy wiedzy, modeli oraz reguł wnioskowania mogą rozwiązywać określone problemy.

Zbiór rozmyty jest uogólnieniem pojęcia zbioru ostrego, który polega na dopuszczeniu, by funkcja przynależności zbioru mogła przyjmować wartości z zakresu . W sposób bardziej formalny zbiór rozmyty należy rozumieć jako zbiór par w pewnej numerycznej przestrzeni rozważań

dla każdego

gdzie jest funkcją przynależności zbioru rozmytego . Każdemu elementowi zbioru przypisywany jest stopień przynależności do zbioru , przy czym istotne jest, aby wartości funkcji przynależności mieściły się w dopuszczalnym przedziale .

Kolejnym równie istotnym zagadnieniem będącym częścią logiki rozmytej jest pojęcie zmiennej lingwistycznej. Zmienna lingwistyczna to po prostu zmienna, której wartościami są pewne słowa lub zdania wyrażone w języku naturalnym. Przykładowe zmienne lingwistyczne zostały zaprezentowane w tabeli 2.3.1.

Tabela 2.3.1. Przykłady zmiennych lingwistycznych

|  |  |
| --- | --- |
| **Kategoria** | **Wartość** |
| Prędkość | mała, średnia, duża |
| Wzrost | niski, średni, wysoki |
| Widoczność | bardzo zła, zła, średnia, dobra, bardzo dobra |
| Ciśnienie | niskie, wysokie |

Spełnienie przesłanki w logice rozmytej może być reprezentowane przez wspomniane wcześniej pojęcie funkcji przynależności . Podczas projektowania systemów, w których zastosowane mają zostać elementy logiki rozmytej, bardzo ważną kwestią jest dobór rodzaju funkcji przynależności, która jednoznacznie będzie określać poszczególne zbiory rozmyte zmiennej lingwistycznej.

Do najpopularniejszych funkcji przynależności można zaliczyć funkcję trójkątną oraz funkcję trapezową, aczkolwiek wyróżnia się jeszcze inne funkcje, jak np. funkcję klasy , czy funkcję klasy .

Trójkątna funkcja przynależności jest opisywana przy pomocy trzech parametrów: .

Trapezowa funkcja przynależności jest opisywana za pomocą czterech parametrów: .

Lewa zewnętrzna funkcja przynależności klasy jest opisywana poprzez użycie dwóch parametrów: .

Prawa zewnętrzna funkcja przynależności klasy jest opisywana z wykorzystaniem dwóch parametrów: .

Dodatkowo istnieje również Gaussowska funkcja przynależności, która definiowana jest przy pomocy dwóch parametrów: .

gdzie: jest środkiem, a to wariancja.

Autorzy modułu *„PyPork”* zdecydowali się zaimplementować ww. funkcje przynależności, jednak należy mieć na uwadze, że funkcji tego typu jest zdecydowanie więcej.

Rozwiązanie (w postaci kodu) przedstawiono w rozdziale 4. niniejszego raportu.

# 3. Specyfikacja zewnętrzna

Rozdział 3. poświęcony jest sprawom technicznym dotyczącym zastosowanych narzędzi. To właśnie tutaj opisano kwestie związane z przygotowaniem lokalnego środowiska testowego, które posłużyło do wykonania własnego modułu poszerzającego możliwości Apache Pig.

## 3.1. Przygotowanie środowiska

W celu pracy z platformą Apache Pig należy przygotować odpowiednie środowisko, które zazwyczaj przyjmuje postać jakiegoś systemu operacyjnego z rodziny Linux. Z reguły wiele narzędzi programistycznych jest tworzonych właśnie pod ten system. Nie inaczej jest w przypadku technologii Hadoop i Apache Pig, w związku z czym pierwszym krokiem do rozpoczęcia pracy było skonfigurowanie przykładowej maszyny wirtualnej. Autorzy projektu zdecydowali, że użyją jedną z popularnych dystrybucji Linuksa, którą niewątpliwie jest Ubuntu. Instalacja systemu jest prosta i tak właściwie sprowadza się do postępowania według wskazówek wyświetlanych na ekranie. W przypadku, gdy zdecydowano się wybrać wersję desktop Ubuntu, podczas instalacji systemu warto skorzystać z opcji *„minimal”*, dzięki której finalnie zainstalowane zostaną tylko niezbędne pakiety i narzędzia.

Po prawidłowym wdrożeniu systemu na maszynie wirtualnej można przejść do kolejnego kroku, który dotyczy instalacji Hadoop i Apache Pig. Aby zminimalizować mogące się pojawić problemy i przy okazji uprościć ręczną konfigurację, autorzy projektu utworzyli przykładowy skrypt powłoki Bash. Skrypt ten realizuje następujące zadania:

* Instalacja środowiska uruchomieniowego Java,
* Pobranie i rozpakowanie paczek Hadoop i Apache Pig,
* Przeniesienie plików Hadoop i Apache Pig do odpowiednich katalogów,
* Utworzenie zmiennych środowiskowych,
* Sprawdzenie poprawności wykonanej instalacji za pomocą polecenia wypisującego informacje o zainstalowanej wersji danego oprogramowania.

Na listingu 3.1.1. zaprezentowany został opisywany skrypt instalacyjny. Należy pamiętać o tym, aby uruchomić ten skrypt z uprawnieniami administratora systemu.

#!/bin/bash

sudo apt install openjdk-8-jdk

sudo apt install openjdk-8-jre

java -version

# main directory where Hadoop and Pig will be installed

MAIN\_DIRECTORY=/usr/local

cd $MAIN\_DIRECTORY

# download hadoop

sudo wget http://apache.claz.org/hadoop/common/hadoop-2.10.1/hadoop-2.10.1.tar.gz

sudo tar xzf hadoop-2.10.1.tar.gz

sudo mkdir hadoop

sudo mv hadoop-2.10.1/\* hadoop

sudo rm -r hadoop-2.10.1

# download pig

sudo wget http://apache.claz.org/pig/pig-0.17.0/pig-0.17.0.tar.gz

sudo tar xzf pig-0.17.0.tar.gz

sudo mkdir pig

sudo mv pig-0.17.0/\* pig

sudo rm -r pig-0.17.0

# home directories for Hadoop and Pig

MY\_HADOOP\_HOME=$MAIN\_DIRECTORY/hadoop

MY\_PIG\_HOME=$MAIN\_DIRECTORY/pig

# create environmental variables

echo export HADOOP\_HOME=$MY\_HADOOP\_HOME >> ~/.bashrc

echo export HADOOP\_INSTALL=$MY\_HADOOP\_HOME >> ~/.bashrc

echo export HADOOP\_MAPRED\_HOME=$MY\_HADOOP\_HOME >> ~/.bashrc

echo export HADOOP\_COMMON\_HOME=$MY\_HADOOP\_HOME >> ~/.bashrc

echo export HADOOP\_HDFS\_HOME=$MY\_HADOOP\_HOME >> ~/.bashrc

echo export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$MY\_HADOOP\_HOME/lib/native >> ~/.bashrc

echo export HADOOP\_OPTS="-Djava.library.path=$MY\_HADOOP\_HOME/lib/nativ" >> ~/.bashrc

echo export YARN\_HOME=$MY\_HADOOP\_HOME >> ~/.bashrc

# if you installed java 7 change name here

echo export JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64 >> ~/.bashrc

# add Hadoop and Pig to PATH

HADOOP\_PATH=$MY\_HADOOP\_HOME/sbin:$MY\_HADOOP\_HOME/bin

PIG\_PATH=$MY\_PIG\_HOME/bin

echo export PATH=$PATH:$HADOOP\_PATH:$PIG\_PATH >> ~/.bashrc

source ~/.bashrc

hadoop version

pig -version

echo "Script completed."

Listing 3.1.1. Skrypt instalacyjny

Następnie można zweryfikować poprawności instalacji Apache Pig. Najprostszy sposób polega na uruchomieniu aplikacji w trybie lokalnym korzystając z komendy przedstawionej na listingu 3.1.2.

pig -x local

Listing 3.1.2. Uruchomienie Apache Pig w trybie lokalnym

Jeśli wszystko zostało prawidłowo skonfigurowane, aplikacja powinna uruchomić się w trybie interaktywnym, w którym możliwe jest wpisywanie poleceń języka Pig Latin. Aby zapoznać się z podstawowymi funkcjami można np. wczytać plik CSV i wyświetlić jego zawartość.

## 3.2. Przykład użycia biblioteki „PyPork”

W celu sprawdzenia poprawności opisywanej biblioteki należy skorzystać z dostarczonego pliku skryptowego w języku Pig Latin, który umieszczony został w katalogu pig\_examples. W przykładzie pokazano użycie operacji rozmytych na danych pochodzących z pliku CSV.

Ponadto należy pamiętać również o tym, aby na początku własnych skryptów zarejestrować główny plik biblioteki main\_fuzzy.py. Plik ten musi zostać umieszczony w miejscu, do którego jest łatwy dostęp, jak np. główny katalog użytkownika lub jego pulpit.

Rejestracja pliku biblioteki odbywa się w pliku o rozszerzeniu .pig za pomocą operatora REGISTER w podobny sposób, jak prezentuje to listing 3.2.1. Ponadto należy pamiętać o tym, aby w miejscu, w którym znajduje się skrypt Pig Latin umieszczony został również plik konfiguracyjny pattern.json oraz odpowiedni plik CSV przechowujące dane używane w skrypcie użytkownika.

Listing 3.2.1. Wskazanie lokalizacji skryptu main\_fuzzy.py

REGISTER '/home/piggy/PyPork/main\_fuzzy.py' using jython as fuzzy

Uruchomienie skryptu dla języka Pig Latin odbywa się z wykorzystaniem komendy z listingu 3.2.2. W oknie konsoli systemowej powinny zostać wypisane zarówno logi generowane przez narzędzie Hadoop oraz wyniki poszczególnych operacji zdefiniowanych w skrypcie języka Pig Latin.

Listing 3.2.2. Uruchomienie skryptu linguistic.pig

pig -x local linguistic.pig

# 4. Specyfikacja wewnętrzna

W niniejszym rozdziale opisana została architektura biblioteki *„PyPork”*.

Cały projekt składa się z kilku plików skryptowych języka Python, w ramach których zaimplementowane zostały funkcje odpowiedzialne za realizację określonych zadań.

W katalogu pig\_examples umieszczono przykłady użycia rozszerzenia *„PyPork”*. Można tutaj znaleźć:

* linguistic.pig – zawiera kilka przykładów pokazujących działanie autorskich funkcji,
* user\_data.csv – przechowuje analizowane dane,
* pattern.json – plik konfiguracyjny, w którym zdefiniowane są zmienne lingwistyczne.

W głównym katalogu znajduje się skrypt main\_fuzzy.py zawierający wszystkie potrzebne funkcje do prawidłowego działania biblioteki. W pliku tym dostępne są m.in. funkcje operujące na wartościach lingwistycznych, definicje funkcji przynależności (np. trójkątna, trapezowa), czy funkcje realizujące działanie rozmytych operatorów AND, OR i NOT. Każda z funkcji została udokumentowana bezpośrednio w kodzie źródłowym, dzięki czemu podczas pracy możliwe jest uzyskanie informacji pomocniczych bezpośrednio w edytorze kodu. Skrypt ten jest wymagany do pracy z Apache Pig, co zresztą zaprezentowano w poprzednim rozdziale.

Ponadto główny katalog repozytorium zawiera kilka skryptów, które weryfikują poprawność działania poszczególnych funkcjonalności:

* fequal\_test.py – sprawdza działanie funkcji fequal,
* linguistic\_test.py – zawiera testy sprawdzające obsługę wartości rozmytych,
* membership\_test.py – przechowuje przykładowe testy, które weryfikują poprawność zaimplementowanych funkcji przynależności,
* operators\_test.py – sprawdza działanie rozmytych operatorów AND, OR i NOT.

Powyższe skrypty należy uruchomić z poziomu konsoli systemowej w podobnym sposób, jak pokazano na listingu 4.1.

Listing 4.1. Uruchamianie skryptu Python z poziomu konsoli systemowej

python operators\_test.py

Wynik uruchomienia pliku operators\_test.py zaprezentowany został na listingu 4.2.

Listing 4.2. Wynik działania testów ze skryptu operators\_test.py

Tests: F\_OR, F\_AND, F\_NOT operators

========== test 1 ==========

values[0]: 1

values[1]: 0.5

result: {'F\_OR': 1, 'F\_AND': 0.5, 'F\_NOT': {'values[0]': 0, 'values[1]': 0.5}}

========== test 2 ==========

values[0]: 0.75

values[1]: 0.1

result: {'F\_OR': 0.75, 'F\_AND': 0.1, 'F\_NOT': {'values[0]': 0.25, 'values[1]': 0.9}}

========== test 3 ==========

values[0]: 0.1

values[1]: 0.4

result: {'F\_OR': 0.4, 'F\_AND': 0.1, 'F\_NOT': {'values[0]': 0.9, 'values[1]': 0.6}}

========== test 4 ==========

values[0]: 0.6

values[1]: 1

result: {'F\_OR': 1, 'F\_AND': 0.6, 'F\_NOT': {'values[0]': 0.4, 'values[1]': 0}}

========== test 5 ==========

values[0]: 0.2

values[1]: 0.3

result: {'F\_OR': 0.3, 'F\_AND': 0.2, 'F\_NOT': {'values[0]': 0.8, 'values[1]': 0.7}}

# 5. Podsumowanie

Wykonanie projektu pt. *„PyPork”* pozwoliło autorom zapoznać się z platformą Apache Pig, która znajduje zastosowanie w analizie i przetwarzaniu dużych zbiorów danych. Platforma ta udostępnia proceduralny język Pig Latin, dzięki któremu możliwe jest pisanie zarówno bardzo prostych, jak i bardziej zaawansowanych skryptów umożliwiających przetwarzanie danych. Pig Latin oprócz udostępniania typowych mechanizmów charakterystycznych dla języków programowania tj. różne typy danych (tuple, bag, map), czy funkcje (np. LOAD, STORE), daje również możliwość definiowania własnych funkcji, co pozwala wprowadzić dodatkowe operacje w potoku przetwarzania. Opcja rozbudowy istniejących rozwiązań niewątpliwie jest jedną z kluczowych zalet platform Hadoop i Apache Pig, ponieważ dzięki temu standardowy zestaw narzędzi (który swoją drogą i tak jest obszerny) może być rozbudowany. W kontekście projektu *„PyPork”* Apache Pig wzbogacono o wybrane elementy logiki rozmytej, która jak się okazuje może być niezwykle przydatna w ciekawej dziedzinie informatyki jaką jest Big Data.

Zaletą technologii Apache Pig jest przede wszystkim to, iż do definiowanie UDF możliwe jest z wykorzystaniem kilku języków programowania. Implementacja biblioteki w języku Python okazała się bardzo wygodna.

Autorzy modułu są zadowoleni z uzyskanych wyników i osiągniętego celu, choć nie da się ukryć, że aktualne rozwiązanie mogłoby zostać bardziej usprawnione. Jeśli dodatkowym celem byłoby udostępnienie biblioteki *„PyPork”* szerzej publiczności, wtedy koniecznym krokiem jest usprawnienie repozytorium w serwisie GitHub. Ponadto przydałaby się rozbudowana i jednocześnie czytelna dokumentacja biblioteki, aby każda zainteresowana osoba mogła z tej biblioteki skorzystać. Tego typu zmiana z pewnością podniosłaby ogólną jakość projektu, a sama popularyzacja tym samym mogłaby przyczynić się do powstania społeczności, której członkowie rozwijaliby istniejące rozwiązanie.

Podsumowując, realizacja projektu *„PyPork”* poruszającego dosyć niestandardową tematykę w ramach kursu *„Chmura Obliczeniowa i Technologie Big Data”*, zdecydowanie poszerzyła wiedzę dotyczącą technik pracy z językiem Python, zbiorów rozmytych i co najważniejsze specjalistycznych narzędzi służących do analizy, gromadzenia i przetwarzania dużych zbiorów danych. Nie da się ukryć, że wiedza o istnieniu narzędzi tj. Hadoop, czy Apache Pig jest niezwykle przydatna podczas projektowania komercyjnych produktów informatycznych, które główny nacisk kładą m.in. na sprawne przekazywanie i analizowanie dużych zbiorów informacji.