XXV Seminarium

ZASTOSOWANIE KOMPUTERÓW W NAUCE I TECHNICE 2015

Oddział Gdański PTETiS

OPTYMALIZACJA PARAMETRÓW APLIKACJI W PROCESIE WYTWARZANIA OPROGRAMOWANIA DLA BIG DATA

Paweł KACZMAREK

Politechnika Gdańska, Wydział Elektroniki, Telekomunikacji i Informatyki
tel: 58 347 24 89 fax: 58 348 61 25 e-mail: pkacz@eti.pg.gda.pl

Streszczenie: Wytwarzanie oprogramowania wiąże się z szeregiem decyzji projektowych obejmujących architekturę aplikacji, wykorzystywane technologie implementacji, jak i zewnętrzne biblioteki. W pracy przedstawiono metodę wyboru technologii i bibliotek związanych z big data, której celem jest optymalizacja atrybutów aplikacji takich jak wydajność działającej aplikacji jak również optymalizacja procesu wytwarzania oprogramowania. Metoda wyboru obejmuje identyfikację parametrów bibliotek, określenie ograniczeń i celu optymalizacji. Na podstawie tych danych następuje ocena alternatywnych rozwiązań i wybór optymalnego wykorzystując metody optymalizacji wielokryterialnej. W kontekście zaproponowanej metody opisano wybrane systemy wspomagające.

Słowa kluczowe: optymalizacja wielokryterialna, big data, integracja oprogramowania.

1. WSTĘP

Rozwój systemów big data (pl. wielkie dane lub gigadane) był spowodowany między innymi pojawieniem się znacznie większych ilości danych niż wcześniej przetwarzane [1] [2]. Wytwarzanie takich systemów jest związane z wykorzystaniem wielu bibliotek i modułów pomocniczych, które realizują znaczną część zadań systemu. Takie podejście wpisuje się w koncepcję integracyjnego wytwarzania oprogramowania. W podejściu tym, część funkcjonalności jest realizowania przez zewnętrzne biblioteki, które różnią się między sobą pod względem zakresu funkcjonalnego jak i parametrów działania, takich jak wydajność, cena i sposób implementacji w formie dostępnego API.

Wybór odpowiednich bibliotek wymaga decyzji projektowych, które optymalizują zarówno wytwarzanie oprogramowania jak i finalne działanie. Dotychczas opracowano wiele technik wyboru bibliotek i modułów [3] [4], jednak istniejące metody mają głównie charakter ogólny i abstrahują od konkretnych zastosowań. Podejście takie powoduje, że niektóre aspekty praktyczne lub specyficzne dla konkretnych zastosowań nie zostają uwzględnione, przez co finalne wyniki mogą nie być optymalne. Istniejące rozwiązania w niewielkim stopniu uwzględniają między innymi: powiązania technologiczne między różnymi grupami alternatywnych modułów, warstwową budowę aplikacji, połączenia gotowych modułów i własnego kodu, oraz infrastrukturę wykonania.

Biorą pod uwagę powyższe problemy, w pracy przedstawiono metodę wyboru technologii i bibliotek związanych z big data, której celem jest optymalizacja parametrów działającej aplikacji jak również optymalizacja procesu wytwarzania oprogramowania.

Głównymi krokami metody są: identyfikacja parametrów bibliotek dla czasu wytwarzania i czasu działania, ocena alternatywnych bibliotek, określenie wymaganych ograniczeń parametrów, określenie wag optymalizacji dotyczących aplikacji, wybór bibliotek. Identyfikacja parametrów czasu wytwarzania i czasu działania aplikacji ma na celu zrównoważenie zużycia zasobów związanych z wytworzeniem oprogramowania względem korzyści w czasie działania aplikacji. Wybór alternatywnych bibliotek zależy od parametrów czasu wytwarzania i czasu wykonania zarówno dla tych bibliotek jak i spełnienia wymagań aplikacji.

Ukierunkowanie badań na technologii big data wynika z dynamicznego rozwoju tej dziedziny nauki odpowiadającego na zapotrzebowanie rynku. Szacuje się [2], że do roku 2019 rynek rozwiązań big data osiągnie wartość ponad trzech miliardów Euro. Odpowiada to wzrostowi rocznemu w tempie ponad 20% i może stanowić jeden z głównych nurtów rozwoju informatyki.

2. OPTYMALIZACJA PARAMETRÓW APLIKACJI

2.1. Parametry aplikacji

Optymalizacja parametrów aplikacji dotyczy zarówno etapu wytwarzania jak i etapu działania aplikacji. Można założyć w uproszczeniu, że etap wytwarzania wiąże się z niepożądanymi nakładami, zaś etap działania wiąże się z pożądanymi korzyściami. Proces optymalizacji dąży do maksymalizacji korzyści przy jednoczesnej minimalizacji kosztów i parametrów niepożądanych [3].

Wśród parametrów czasu wytwarzania można wyróżnić przede wszystkim koszt i czas. Każdy z parametrów może być szczegółowo analizowany zależnie od szczegółowych faz projektu: przygotowanie i nauka, projektowanie, kodowanie.

Parametry czasu działania aplikacji obejmują wiele ogólnych i szczegółowych metryk jakości takich jak wydajność, wiarygodność, prostota utrzymania i inne. Przykładowo, ogólny parametr wiarygodność [5] obejmuje szereg szczegółowych parametrów: niezawodność, dostępność, bezpieczeństwo zewnętrzne i wewnętrzne (safety, security).

2.2. Optymalizacja wielokryterialna

Optymalizacja wielokryterialna jest znaną metodą optymalizacji stosowaną w przypadku, gdy występuje wiele

kryteriów optymalizacji. W procesie optymalizacji dąży do maksymalizacji funkcji celu. Ponadto przewiduje się zdefiniowanie ograniczeń na zasoby jako maksymalnych lub minimalnych wartości. Optymalizacja przyjmuje następujące założenia [3] [4]:

- Zostają zdefiniowane atrybuty $(q^1,...,q^n)$
- Dla każdej klasy usług (S) istnieje jeden lub więcej alternatywnych wyborów (s).
- Każdy alternatywny wybór posiada określone wartości atrybutów $q_i = [q_i^1,...,q_i^n]$
- Zdefiniowana funkcja celu (F) końcowej aplikacji jako suma ważona wartości atrybutów
- Zostają zdefiniowane ograniczenia $Q_c = [Q_c^1,..,Q_c^m]$

W ogólnym przypadku problem jest modelowany jako całkowitoliczbowe programowe liniowe.

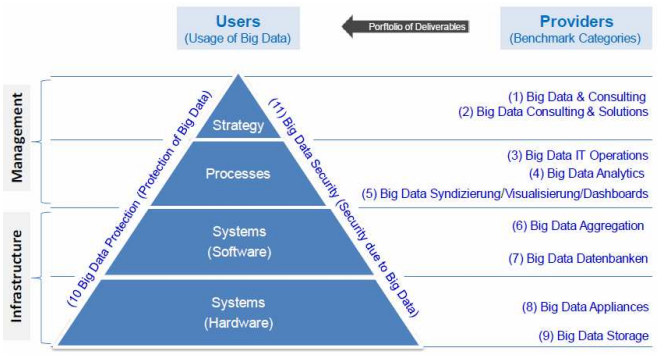
$$\begin{aligned} &Max \sum_{i=1}^{N} \sum_{j \in S_i} F_{ij} x_{ij} \\ &gdzie \sum_{i=1}^{N} \sum_{j \in S_i} q_{ij}^q * x_{ij} \le Q_c^a (a = 1,...,m) \\ &\sum_{j \in S_i} x_{ij} = 1, \quad x_{ij} \in \{0,1\}, \quad i = 1,...,N, \end{aligned} \tag{1}$$

gdzie: F – waga funkcji celu, x_{ij} – wybór alternatywny j dla usługi S_i, q – wartość atrybutu

3. ROZWIĄZANIA BIG DATA

Podejście big data jest stosunkowo nowym rozwiązaniem stosowanym do zarządzania wielkimi zasobami danych, zazwyczaj nie posiadającymi ścisłej struktury i pochodzącymi z różnych źródeł. Big data wprowadza funkcjonalność zarządzania i analizy wielkimi zasobami danych do głównego nurtu rozwoju systemów. Jako argument potwierdzający zasadność stosowania tego typu rozwiązań podaje się, że co dwa dni powstaje tyle informacji co od początku cywilizacji do roku 2003 [1].

Istnieje duży wybór modułów, systemów, bibliotek i innych alternatywnych rozwiązań wspierających big data. Ich zakres funkcjonalny różni się zasadniczo zależnie od ich przeznaczenia i dojrzałości. Na rysunku 1 przedstawiono typy rozwiązań według grupy Experton [2].



Rys. 1. Typy rozwiązań big data według grupy Experton [2]

W kontekście tej pracy wyróżniono następujące grupy systemów wspierających:

- podstawowe biblioteki funkcjonalne (systemy) - obejmują przetwarzanie danych i ich przechowywanie (np. Hadoop, NoSQL db) [6] [7]
- frameworki wytwarzania - wspierają wytwarzanie oprogramowania (np. Spring Framework) [8]
- środowiska zintegrowane (procesy) - wspierają rozwiązania takie jak analiza, wizualizacja, raportowanie (np. IBM Watson, SAP HANA)
- Infrastruktura cloud

4. METODA WYBORU ROZWIĄZAŃ

Zaproponowana metoda doboru rozwiązań big data ma na celu optymalizację parametrów aplikacji (wytwarzania i wykonania) przy założeniu, że mogą zostać wykorzystane gotowe biblioteki przy wytwarzaniu aplikacji. W tym rozwiązaniu uwzględniono zarówno dobór odpowiednich, istniejących rozwiązań jak również wytwarzanie wymaganych modułów od początku. Wybór właściwych bibliotek jest zależny od rozwiązywanego problemu, rozmiaru i charakterystyki danych.

W analizie założono, że integrowane biblioteki są ze sobą kompatybilne, co implikuje zastosowanie tych samych języków programowania lub zgodnych interfejsów usług zależnie od sposobu integracji. Analiza zgodności wykracza poza zakres tej pracy i była szczegółowo opisywana przez autora w [9]. W uproszczeniu można założyć, że integracja na zasadzie komponentowej obejmuje wbudowanie biblioteki w aplikację i zastosowanie tego samego języka programowania. Integracja na zasadzie usług, natomiast, obejmuje wywołanie funkcji przez zdefiniowany interfejs zewnętrzny (zazwyczaj usługi Web services), co nie wymaga zastosowania tego samego języka programowania.

4.1. Etapy wyboru

Metoda wyboru obejmuje kilka głównych kroków decyzyjnych:

1. Zastosowanie technologii big data
2. Wybór metody wykonania
    a. wykonanie systemu od podstaw albo
    b. wybór bibliotek wspierających:
        i. podstawowe biblioteki
        ii. frameworki wspierające
        iii. zintegrowane środowiska
3. Wybór cloud computing

W celu usystematyzowania procesu wyboru konieczne jest zdefiniowanie następujących elementów analogicznie do problemu optymalizacji wielokryterialnej:

- wymagane moduły funkcjonalne $(S^1,...,S^t)$
- parametry aplikacji $(q^1,...,q^n)$
- ograniczenia na parametry $Q_c = [Q_c^1,..,Q_c^m]$
- wagi dla funkcji celu

4.2. Warunki wyboru technologii big data

W wielu przypadkach wybór technologii big data nie jest zasadny. Zastosowanie tej technologii może być korzystne w przypadku, gdy system spełnia następujące wymagania [1] [2]:

- Dane są dostarczane w ilościach, które nie mogą być przetworzone przez tradycyjne systemy
- Analiza danych powinna wspierać decyzje w czasie rzeczywistym
- Decyzje powinny być optymalizowane na podstawie danych otrzymywanych na bieżąco
- Analiza danych powinna obejmować dane historyczne w różnych zakresach czasowych

- Występuje konieczność walidacji poprawności wielkiej liczby danych

Ocena istnienia tych przesłanek jest pierwszym etapem decydujących o zastosowaniu big data.

4.3. Ocena kosztów wykonania

Dla uproszczenia, w dalszej części analizy będzie używane określenie biblioteka w znaczeniu zewnętrznej usługi lub frameworku.

Dla każdej z alternatywnych bibliotek zostają określone:

- wymagane zasoby
- klasy usług (S) realizowane przez bibliotekę oraz atrybuty q dla każdej z realizowanych klas usług.
- poziom spełnienia ograniczeń

Powyższe dane stanowią jednocześnie minimalne parametry opisu biblioteki, które umożliwiają zastosowanie jej w proponowanej metodzie.

W tabeli 1 przedstawiono modelowe biblioteki wraz z zaznaczeniem realizowanych klas usług oraz spełnienia ograniczeń.

W przypadku kosztów wyróżniono: koszt programowania oraz koszt licencji biblioteki zewnętrznej. W przypadku zasobu czasu wyróżniono czas nauki, czas programowania oraz czas wdrożenia. Ponadto dla każdego modułu funkcjonalnego zostaje oszacowany koszt realizacji bez wykorzystania zewnętrznej biblioteki.

W przypadku ograniczeń przyjęto decyzję binarną, że rozwiązanie może spełniać lub nie spełniać ograniczenia.

Tablica 1. Wzór macierzy oceny efektywności zastosowania zewnętrznych bibliotek

Rozwiązanie	Koszt	Czas	Moduł funkcjonalny				Ograniczenia			
			S1	S2	S3	...	Q1	Q2	Q3	...
Biblioteka 1	c_1	t_1	1	1	1		1	1	1	
Biblioteka 2	c_2	t_2	0	0	1		1	1	1	
Biblioteka 3	c_3	t_3	0	0	0		0	1	1	
... ...										

4.4. Ocena korzyści i optymalizacja wyboru

Celem optymalizacji jest wybór zewnętrznych bibliotek, które będą realizowały wymagane moduły. Typowe podejście do wyboru bibliotek [3] zakłada, że dla każdej funkcjonalności istnieje wiele alternatywnych realizacji. Zakładając takie podejście, problem może być zamodelowany następująco:

- moduły odpowiadają klasom usług
- alternatywne biblioteki realizujące wymagania odpowiadają alternatywnym usługom
- inne elementy (atrybuty, ograniczenia) są modelowane jak w typowym podejściu.

Takie podejście jednak, może nie odpowiadać realiom wytwarzania, gdyż jedna biblioteka może być pakietem, który realizuje wiele, choć nie wszystkie, moduły. W związku z tym w proponowanej metodzie wykorzystuje się dodatkowe warunki ograniczające zakres wyboru. Biblioteki mogą zostać zaklasyfikowane do kategorii, gdzie z każdej kategorii wybierana jest jedna biblioteka. Biorąc pod uwagę typową architekturę aplikacji tego typu wyróżniono następujące kategorie: środowiska zintegrowane i frameworki, biblioteki interfejsu użytkownika, biblioteki dostępu do danych. Jeżeli biblioteka nie należy do żadnej z tych kategorii, przyjęto, że może być dowolnie integrowana z innymi. Po dokonaniu klasyfikacji i określeniu alternatywnych kombinacji, dla każdego rozwiązania obliczane są następujące parametry:

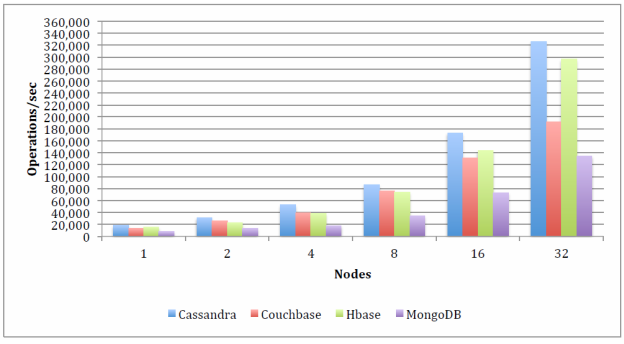
- koszt i czas wykorzystania bibliotek
- koszt i czas implementacji nie zrealizowanych wymagań
- wartość ważonej funkcji celu
- weryfikacja ograniczeń

Na podstawie dokonanych obliczeń dla alternatywnych kombinacji, zostają wybrane biblioteka, które zapewniają najwyższą wartość funkcji celu przy jednoczesnym spełnieniu wymagań na ograniczenia. Finalna aplikacja jest komponowana z wybranych bibliotek oraz dedykowanych modułów.

5. SYSTEMY WSPOMAGAJĄCE

5.1. Bazy danych noSQL

Rozwiązania big data są silnie powiązane z bazami danych noSQL, zwanymi też not-only-SQL [7] [10]. Zastosowanie nowego podejścia do przechowywania danych wynika z natury danych big data, gdzie struktura danych jest często nieokreślona lub też zmienna. Dzięki takiemu podejściu możliwe jest uproszczenie projektowania i uzyskanie większej skalowalności i wydajności. Wadą takiego rozwiązania jest podejście określane jako zasada CAP (Consistency, Availability, Partition Tolerance). Zasada CAP zakłada, że możliwe jest spełnienie dwóch z trzech właściwości.



Rys. 2. Porównanie wydajności baz danych noSQL [7]

W przypadku baz noSQL zasadnicze znaczenie ma skalowalność i wydajność. Wybór odpowiedniego rozwiązania w proponowanej metodzie optymalizacji może bazować na jednym z wielu dostępnych porównań wydajności baz danych noSQL, np. [7] [10], jak przedstawiono na rysunku 2.

5.2. Frameworki wytwarzania

Na rynku dostępnych jest wiele frameworków wytwarzania aplikacji, które różnią się między sobą poziomem zaawansowania i funkcjonalnością. Przykładem może być Hadoop [6], popularny framework zorientowany na

zastosowania big data udostępniający bazowe funkcjonalności. Wśród funkcji systemu dostępne są rozproszony system plików, szeregowanie zadań, algorytmy MapReduce. Hadoop nie udostępnia rozwiązań dających bezpośrednią korzyść biznesową, takich jak możliwości analityczne, dlatego zazwyczaj jest częścią bardziej złożonych systemów.

Spring IO [8] jest frameworkiem programowania ogólnego zastosowania, choć zawiera również moduły dedykowane do rozwiązań big data. Budowa framworka jest wysoce modułowa dzięki czemu możliwe jest instalowanie i uruchomienie wyłącznie potrzebnych funkcjonalności. Stanowi wygodne narzędzie umożliwiające integrację niskopoziomowych rozwiązań takich jak Hadoop czy bazy danych noSQL.

5.3. Porównywarki rozwiązań cloud computing

Wykorzystanie rozwiązań cloud computing jest jedną z decyzji projektowych. W przypadku zastosowania takiego podejścia konieczne jest dokonanie optymalnego wyboru spośród wielu dostępnych alternatyw. Obecnie istnieje około kilkudziesięciu dostawców rozwiązań cloud computing, każdy z nich udostępnia różne opcje współpracy i zasady rozliczania należności.

Podobnie jak w przypadku rozwiązań noSQL, na rynku pojawiły się systemy porównywania ofert pochodzących od różnych dostawców. Porównanie i wybór odpowiedniej oferty może stanowić uzupełnienie proponowanej metody optymalizacji wytwarzania aplikacji big data. Wśród dostępnych usług można wyróżnić systemy interaktywne takie jak: cloud-computing.softwareinsider.com oraz Cloudorado www.cloudorado.com. Ponadto dostępne są liczne porównania tekstowe, np. opracowania Wikipedii.

6. PODSUMOWANIE

Zaproponowana metoda wyboru bibliotek ma na celu uproszczenie procesu wytwarzania systemów big data przy jednoczesnej optymalizacji działania. Przedstawione w pracy rozwiązanie dotyczy konkretnej dziedziny zastosowania i uwzględnia uwarunkowania techniczne występujące podczas implementacji. W konsekwencji, zakres alternatywnych wyborów zostaje zawężony zależnie od technicznych przesłanek wykorzystania bibliotek.

Dalszy rozwój metody obejmuje implementację systemu wspomagającego, który będzie usprawniał proces opisu bibliotek, ich klasyfikacji oraz wybór finalnego rozwiązania. Wymagane jest również przebadanie wybranych bibliotek w celu opracowania ich rzeczywistych parametrów. Ponadto analiza może być rozszerzona o usunięcie ograniczenia na przypisanie biblioteki do jednej warstwy architektury aplikacji. Dzięki temu możliwe będzie odwzorowanie bibliotek, których zakresy funkcjonalne obejmują wiele warstw lub też pokrywają się częściowo z innymi bibliotekami.

7. BIBLIOGRAFIA

1. Hazra A., Jewell D. et. al.: Performance and Capacity Implications for Big Data, IBM - International Technical Support Organization, 2014
2. Landrock H., Schonschek O., Gadatsch A.: Big Data Vendor Benchmark, A Comparison of Big Data Solution Providers, Experton Group AG, Germany, 2015
3. Yu T., Zhang Y., Lin K.: Efficient Algorithms for Web Services Selection with End-to-End QoS Constraints, ACM Transactions on the Web, 2007
4. Cao H., Feng X., Sun Y., Zhang Z., Wu Q.: A Service Selection Model with Multiple QoS Constraints on the MMKP, IFIP International Conference on Network and Parallel Computing, 2007, DOI 10.1109/NPC.2007.35
5. Krawczyk H., Wiszniewski B.: Analysis and Testing of Distributed Software Applications, Research Studies Press Ltd., 1998
6. White T.: Hadoop The Definite Guide, O'Reilly Media, 2011, ISBN 978-1-449-38973-4
7. Benchmarking Top NoSQL Databases - Apache Cassandra, Couchbase, HBase, and MongoDB, End Point Corporation, http://www.endpoint.com/, 2015
8. Johnson R. i inni: Spring Framework Reference Documentation, spring.io, 2004-2015
9. Kaczmarek P.: Interoperability Constraints in Service Selection Algorithms, ENASE 2012 - 7th International Conference on Evaluation of Novel Approaches to Software Engineering, Wroclaw, Poland; 29-30 June, 2012
10. Abramova V., Bernardino J., Furtado P.: Which NoSQL Database? A Performance Overview Open Journal of Databases (OJDB), ISSN 2199-3459, 2014

OPTIMIZATION OF BIG DATA APPLICATION ATTRIBUTES CONSIDERING SOFTWARE DEVELOPMENT PROCESS

During software development, effective design decisions must be made considering application architecture, development technology and integration of external libraries. The paper presents a method of selection of big data technologies and libraries. The purpose of the method is optimization of application attributes such as performance as well as optimization of the software development process. The method covers identification of library parameters, specification of application constraints and definition of optimization purpose. Considering gathered information, alternative development options are rated and optimal solution is selected using multicriteria optimization methods. Selected big data supporting systems were described in the context of the proposed method.

Key-words: multicriteria optimization, big data, software integration.