# POLITECHNIKA WARSZAWSKA

WYDZIAŁ ELEKTRYCZNY

# Rozprawa doktorska

mgr inż. Artur Sławomir Krupa

Metody obliczeń rozproszonych w stochastycznym modelowaniu tkanek

> Promotor dr hab. inż. Bartosz Sawicki

> > Promotor pomocniczy dr inż. Tomasz Leś

WARSZAWA 2019

"Etiam si omnes, ego non" / "Choćby wszyscy, ja nie" Ewangelia według św. Mateusza (26,33)

### STRESZCZENIE

### Politechnika Warszawska, Wydział Elektryczny

### Rozprawa doktorska: "Obliczenia rozproszone w stochastycznym modelowaniu tkanek"

Autor: mgr inż. Artur Sławomir Krupa

W niniejszej pracy przedstawione zostały metody pozwalające na rozwiązywanie problemów symulacji tkanek żywych przy wykorzystaniu platformy obliczeń rozproszonych. Metoda obejmuje proces projektowania obiektu wraz ze zdefiniowaniem podstawowych parametrów materiałowych uwzględniających ich zmienność. Stochastyczne podejście do zagadnienia pozwala uniknąć ograniczeń narzuconych przez tradycyjne metody deterministyczne. Wykorzystanie chmury obliczeniowej pozwala na zwielokrotnienie liczby analizowanych przypadków symulacji, zwiększając tym samym dokładność rozwiązania.

Zagadnienie statystyczne oraz jego wykorzystanie w modelowaniu tkanek jest nowatorskim podejściem, które do tej pory nie było szeroko omawiane w literaturze. Praca obejmuje analizę teoretyczną zagadnienia stochastycznego. Modelowanie tkanek zostało w pracy omówione i zweryfikowane na przykładach homogenicznych oraz heterogenicznych obiektów. Wyznaczone zostały także różnice zależne od parametrów materiałowych.

Praca porusza także zagadnienie związane z platformą obliczeń rozproszonych opartą o chmurę obliczeniową. Badania skupiają się na przedstawieniu dostępnych rozwiązań chmurowych, wyjaśnieniu ich funkcjonowania oraz pokazują kompletną konfigurację usługi od uruchomienia do działającego systemu obliczeniowego. Systemy obliczeń rozproszonych zostały także przebadane zarówno w infrastrukturze wykorzystującej rozwiązania Apache Hadoop oraz HTCondor.

Przygotowane zostały trzy modele, w praktycznych scenariuszach obliczeniowych, które pozwoliły sprawdzić skuteczność oraz wydajność rozwiązania. Pierwszym analizowanym przypadkiem było zagadnienie grzania rezystancyjnego na modelu dwuwymiarowym. Drugim zadaniem obliczeniowym był również przypadek dwuwymiarowy, jednak bardziej złożony,

którego tematyka obejmowała zagadnienie stymulacji nerwowo-mięśniowej. Duża liczba parametrów oraz zmiennych zweryfikowała możliwości chmury obliczeniowej. Ostatnim scenariuszem była symulacja zabiegu terapii elektrowstrząsowej na modelu trójwymiarowym głowy pacjenta.

Zaprezentowane rozwiązania oraz opracowane przykładowe scenariusze modelowe pokazały możliwości wykorzystania platform obliczeń rozproszonych w badaniach bioinżynieryjnych, w których wpływ rozwiązań technicznych na tkanki biologiczne może być znaczący.

### ABSTRACT

Warsaw University of Technology, Faculty of Electrical Engineering

### Doctoral Thesis: "Distributed computing in stochastic tissue modelling"

Author: M. Sc. Artur Sławomir Krupa

The PhD Thesis presents the methods of solving problems in simulations of living tissues with usage of distributed systems. The method includes design of an object with description of material parameters with its variability. Stochastic approach helps to avoid limitations of typical deterministic solutions. At the same time, usage of cloud computing gives an opportunity to multiply number of parallel simulations, increasing accuracy.

Stochastic issue and its usage in tissues modelling is innovative solution, not that much described in literature. Thesis discuses theoretic analysis of stochastic approach with popular probability density functions as well as random variability algebra. Tissue modeling was discussed and verified based on homogenous and heterogenous objects. Differences based on materials parameters were also determined.

Thesis topics are also related to distributed computing platform, based on cloud computing. An area of analysis discuses available solutions of cloud computing introduces its functionality and shows complete configuration of services from early beginning to working computing system. Distributed systems verified and tested were based on Apache Hadoop and HTCondor solutions.

There were designed three models of computational problems, related to realistic scenarios, that they let us to verify an efficiency and performance of solutions as well as they confirmed an assumptions of modelling schema. First scenario was an ohmic heating problem on twodimensional model. Second one was also two-dimensional model, but more complex, related to EMS (electric muscle simulation). Massive number of parameters and variables verified cloud platform capabilities. Last scenario was a simulation of ECT (electro-convulsive therapy) treatment on a patient head three-dimensional-model. Discussed solutions and designed example scenarios provided performance of cloud computing in bioengineering research, where technical solutions in biological tissues problems can be significant and is rational.

### PODZIĘKOWANIA

Pragnę podziękować wszystkim, bez których niniejsza praca nie mogłaby powstać.

Przede wszystkim serdecznie dziękuję Promotorowi mojej pracy **dr. hab. inż. Bartoszowi Sawickiemu** za cierpliwość, cenne rady i nieocenioną pomoc na każdym etapie pracy.

Dziękuję również Promotorowi pomocniczemu **dr. Tomaszowi Lesiowi** za pomoc w organizacji pracy oraz trafne sugestie.

Dziękuję pracownikom Zakładu Elektrotechniki Teoretycznej i Informatyki Stosowanej za zapewnienie cichego i spokojnego miejsca do pracy, życzliwość oraz dobre pomysły.

Podziękowania wędrują do Macieja Piaska, pracownika naszego Wydziału, za przygotowanie planów i zbudowanie naszego urządzenia pomiarowego.

Dziękuję Maciejowi Sobiankowi i Rafałowi Czupryńskiemu z oddziału Microsoft Polska za ogrom pomocy w początkach przygody z Azure oraz zapewnienie dostępu do platformy.

Składam podziękowania Michałowi Furmankiewiczowi z Microsoft-u, ale także Emilowi Wasilewskiemu, Łukaszowi Kałużnemu, Tomaszowi Wiśniewskiemu oraz wszystkim kolegom i koleżankom z grupy Azurowej. Dziękuję za Waszą pomoc w zgromadzeniu pokaźnej wiedzy o Microsoft Azure, cenne porady na każdym etapie wdrażania infrastruktury oraz wyjaśnienia zagadnień z zakresu chmury obliczeniowej, z jakimi się spotkałem podczas pisania tej pracy.

Na sam koniec pragnę podziękować trzem wyjątkowym osobom – moim Rodzicom, za to, że dzięki Nim zaszedłem tak daleko i nie przestałem wierzyć w siebie oraz mojej Narzeczonej, która dodawała i nieustannie dodaje mi sił oraz motywacji każdego dnia.

Artur

# SPIS TREŚCI

ROZDZ	IAŁ	1. WPROWADZENIE
1.1.	Sto	chastyczne modele bioelektromagnetyczne2
1.2.	Cel	i zakres pracy4
ROZDZ	IAŁ	2. PARAMETRY MODELI TKANEK7
2.1.	Zło	żoność tkanek7
2.2.	Roz	zkłady gęstości prawdopodobieństwa12
2.3.	Nar	zędzia statystyczne14
2.3.	1.	Założenia stochastyczne modelowania tkanek15
2.3.	2.	Metoda Monte Carlo (MC)17
2.3.	3.	Metoda Chaosu Wielomianowego (PCE)18
2.3.	4.	Porównanie metody MC oraz PCE19
2.4.	Two	orzenie modelu statystycznego tkanki23
2.4.	1.	Akwizycja danych24
2.4.	2.	Pomiary materiałów jednorodnych
2.4.	3.	Materiały niejednorodne
2.5.	Opr	acowanie modeli tkanek
ROZDZ	IAŁ	3. PLATFORMY OBLICZEŃ ROZPROSZONYCH
3.1.	Obl	iczenia rozproszone40
3.1.	1.	Grid40

3.1.2.	BOINC	42
3.2. Chr	nura obliczeniowa	43
3.2.1.	Wirtualizacja	45
3.2.2.	Dostępne rozwiązania chmurowe	46
3.3. Wy	korzystane technologie	49
3.3.1.	Apache Hadoop	50
3.3.2.	HTCondor	52
3.4. Tes	ty technologiczne systemów dystrybucji zadań	54
ROZDZIAŁ	4. PRZYKŁADY OBLICZENIOWE	61
4.1. Grz	zanie rezystancyjne	61
4.1.1.	Symulowany obiekt	63
4.1.2.	Metodyka badań	65
4.1.3.	Wyniki i wnioski	67
4.2. Ele	ktrostymulacja nerwowo-mięśniowa	68
4.2.1.	Symulowany obiekt	69
4.2.2.	Metodyka badań	70
4.2.3.	Wyniki i wnioski	70
4.3. Ter	apia elektrowstrząsowa	72
4.3.1.	Symulowany obiekt	73
4.3.2.	Metodyka badań	73
4.3.3.	Wyniki i wnioski	74
ROZDZIAŁ	5. PODSUMOWANIE	79
BIBLIOGRA	AFIA	83
ZAŁĄCZNI	K 1 ARCHITEKTURA CHMURY MICROSOFT AZURE	91
ZAŁĄCZNI	K 2 WDROŻENIE W MICROSOFT AZURE	101

### PODSTAWOWE OZNACZENIA I TERMINY

AWS (ang. Amazon Web Services)

Chmura obliczeniowa (ang. Cloud computing)

**BOINC** (ang. Berkeley Open Infrastructure for Network Computing)

CLI (ang. Command Line Interface)

FEM (ang. Finite Element Method)

GCP (ang. Google Cloud Platform)

GPGPU (ang. General-purpose Processing on Graphics Processing Unit)

Grupa zasobów (ang. Resource Group)

GUI (ang. Graphical User Interface) Rozwiązanie chmury obliczeniowej dostarczane przez Amazon.com Inc.

Rozwiązanie informatyczne obejmujące połączone ze sobą jednostki obliczeniowe w ramach jednej serwerowni, które udostępnia zasoby w formie usług.

Wielosystemowa platforma open-source przeznaczona do obliczeń rozproszonych wspierająca rozwiązania Grid.

Interfejs komunikacji z komputerem oparty o okno tekstowe do wpisywania poleceń.

Metoda Elementów Skończonych (MES), sposób rozwiązywania układów równań różniczkowych opierający się o dyskretyzacji obszaru na mniejsze elementy.

Rozwiązanie chmury obliczeniowej dostarczane przez Google Inc.

Jednostka graficzna oferująca funkcjonalności obliczeń numerycznych, funkcjonująca podobnie do procesora.

Zbiór usług powiązanych ze sobą, uruchomionych na platformie chmury obliczeniowej.

Interfejs komunikacji z komputerem oparty o graficzny interfejs użytkownika. GUID (ang. Globally Unique Identifier)

HDFS (ang. Hadoop Distributed File System)

HPC (ang. High Performance Computing)

HPCC (ang. High Performance Computing Cluster

HTC (ang. High-Throughput Computing)

Hypervisor / VMM (ang. Virtual Machine Manager)

IaaS (ang. Infrastructure as a Service) Identyfikator globalnie unikalny przypisywany do obiektu wymagającego indywidualnego oznaczenia.

System plików stosowany w rozwiązaniu Apache Hadoop do przechowywania danych.

Rozwiązanie oferujące dużą wydajność obliczeniową przez zastosowanie rozbudowanej jednostki obliczeniowej lub połączonych ze sobą mniejszych jednostek, pracujących razem.

Platforma obliczeniowa open-source dedykowana dla systemów klastrowych i przyspieszania obliczeń rozproszonych.

Pojęcie określające rozwiązanie obliczeń rozproszonych przeznaczone do wykonywania zadania wspólnymi jednostkami obliczeniowymi.

Nadrzędny system lub proces monitorujący dostęp do podzespołów komputerowych dla będącego powyżej systemu operacyjnego.

Model funkcjonowania chmury obliczeniowej, w której zarządzane przez użytkownika są oprogramowanie i system oraz zwirtualizowana infrastruktura. Fizyczna obsługa sprzętowa zapewniona jest przez dostawcę rozwiązania.

Jednostka obliczeniowa (ang. Computing unit)

Klaster (obliczeniowy) (ang. Computing cluster)

MapReduce

Komputer wykonujący obliczenia.

Zbiór połączonych ze sobą jednostek obliczeniowych.

Technologia dystrybucji zadań w systemach obliczeń rozproszonych dostępna w rozwiązaniu Apache Hadoop.

#### Mapper

MC (ang. Monte Carlo method)

NSF (ang. National Science Foundation)

**On-Premise** 

PaaS (ang. Platform as a Service)

PCE (ang. Polynomial Chaos Expansion)

PDE (ang. Partial Differential Equation)

PDF (ang. Probability Density Function)

**RBAC** (ang. Role-Based Access Control) Część rozwiązania stosowanego na platformie opartej o rozwiązanie MapReduce, które rozdziela duże zadania na mniejsze.

Algorytm polegający na losowym generowaniu wartości obliczeniowych służący do deterministycznego rozwiązywania zagadnień stochastycznych.

> (US) Narodowa Fundacja Nauki, odpowiednik polskiej Fundacji Badań i Rozwoju Nauki.

Model funkcjonowania rozwiązania informatycznego, w którym infrastruktura fizyczna, obsługa oraz oprogramowanie są realizowane przez użytkownika.

Model funkcjonowania chmury obliczeniowej, w której zarządzane przez użytkownika są oprogramowanie i system – infrastruktura fizyczna pozostaje w obsłudze przez dostawcę rozwiązania.

Chaos Wielomianowy, metoda niedeterministyczna służąca do opisywania procesów parametrycznie zmiennych.

> Rozwiązanie różniczkowe cząstkowe składające się z równań funkcji wielowymiarowych oraz pochodnych. cząstkowych używane w modelowaniu obiektów numerycznych.

Funkcja gęstości prawdopodobieństwa, rozkład opisywany w postaci funkcji probabilistycznej, najczęściej gaussoidy.

Kontrola dostępu oparta o przydzielane role.

### Reducer

SaaS (ang. Software as a Service)

VMSS (ang. Virtual Machine Scale Set)

Węzeł roboczy (ang. Worker node)

Węzeł zarządzający (ang. Master node/Manager)

YARN (ang. Yet Another Resource Negotiator) Część rozwiązania stosowanego na platformie opartej o rozwiązanie MapReduce, które łączy wynikowe zadania cząstkowe w całość.

Model funkcjonowania chmury obliczeniowej, w której odpowiedzialność za oprogramowanie oraz infrastrukturę fizyczną zapewnia dostawca rozwiązania.

Skalowalny zestaw połączonych ze sobą maszyn wirtualnych, posiadających takie same parametry technologiczne w ramach platformy Microsoft Azure.

> Jednostka/komputer w klastrze obliczeniowym, która zajmuje się obliczeniami.

Jednostka/komputer w klastrze obliczeniowym, która zajmuje się dystrybucją zadań oraz odbieraniem wyników.

Technologia dystrybucji zadań w systemach obliczeń rozproszonych dostępna w rozwiązaniu Apache Hadoop. Zwana także MapReduce 2.0.

# ROZDZIAŁ 1. WPROWADZENIE

Dynamiczny rozwój systemów informatycznych i metod cyfrowej komunikacji ma silny wpływ na dzisiejsze społeczeństwo. Dzieje się to nie tylko na płaszczyźnie używanych codziennie i masowo urządzeń elektronicznych, ale również w środowiskach naukowych. Powstają całe nowe obszary nauki związane z wykorzystywaniem pojawiających się możliwości. Systemy komputerowe zbierają ogromne ilości danych o otaczającym go świecie (Big Data), umożliwiają analizowanie ich i odkrywanie niedostępnych wcześniej prawidłowości (Data Mining), aż w końcu samodzielnie zaczynają formułować wnioski i podejmować działania (Machine Learning). Wszystko to bazuje na komercyjnych centrach komputerowych o niespotykanych do tej pory rozmiarach,<sup>1</sup> które oferują usługi obliczeniowe w atrakcyjnych cenach (Cloud Computing).

Pojęcie chmury obliczeniowej (ang. Cloud Computing) wiąże się również z możliwościami elastycznego współdzielenia zasobów sprzętowych poprzez wykorzystanie technologii wirtualizacyjnych. Pozwala to na łączenie dużych grup pojedynczych jednostek obliczeniowych w jeden system, niezależnie od ich fizycznej lokalizacji. Takie rozwiązanie pozwoliło na optymalizację kosztów i wynikające z tego obniżenie cen wykorzystania infrastruktury obliczeniowej. Technologia chmury obliczeniowej pojawiła się jako hasło marketingowe. Jednak jej możliwości oraz koszty, które z miesiąca na miesiąc maleją, sprawia, że wykorzystanie rozwiązania w innych celach niż komercyjne, staje się atrakcyjne.

Przy badaniach prezentowanych w niniejszej pracy intensywnie wykorzystywana była infrastruktura usług chmurowych jako platforma do obliczeń rozproszonych. Tylko w ten sposób możliwe było przeprowadzanie ogromnych liczb symulacji numerycznych potrzebnych do statystycznej analizy wyników. Tego typu spojrzenie jest szczególnie istotne w przypadku

<sup>&</sup>lt;sup>1</sup> w 2016 roku zasoby obliczeniowe firmy Google szacowane były na 2,5 miliona serwerów (źródło: https://www.datacenterknowledge.com/archives/2017/03/16/google-data-center-faq) (dostęp: 2019.01.15)

modelowania obiektów pochodzenia biologicznego, które charakteryzują się wysoką i nieprzewidywalną zmiennością opisujących je parametrów.

### 1.1. Stochastyczne modele bioelektromagnetyczne

Bioelektromagnetyzm jest specyficzną dziedziną wiedzy, która zajmuje się badaniami nad wpływem pola elektromagnetycznego na organizmy ludzkie [1]. Dynamika rozwoju wiąże się z nieustannie ulepszaną technologią, która wymaga opracowywania bezpiecznych urządzeń. Szczególnie ważne jest to w przypadku wpływu ich na organizm człowieka.

Każde urządzenie elektryczne oraz elektroniczne jest potencjalnym generatorem pola elektromagnetycznego na organizm człowieka. Jego wpływ zależy od wielu parametrów, jak miejsce oddziaływania, częstotliwość, czy moc emitowaną do otoczenia. W przypadku człowieka znajdującego w pobliżu – wpływ na jego ciało i narządy.

W ciągu ostatnich lat rozwinęła się analiza zagadnień poruszających problemy związane z odpowiednim badaniem obszarów biologicznych w zakresie norm bezpieczeństwa. Przede wszystkim jednak widoczny jest znaczny wzrost wpływu pola elektromagnetycznego na organizmy żywe. Bezpieczeństwo człowieka jest priorytetem, ale badania prowadzone w zakresie oddziaływania pola elektromagnetycznego nie mogą zagrażać życiu ludzkiemu. Tak narodziła się idea symulowania wpływu pól elektromagnetycznych na obiekty biologiczne przy wykorzystaniu komputera.

Modelowanie numeryczne wykorzystuje parametry dielektryczne obiektów biologicznych, których baza została stworzona w oparciu o wiele lat badań [2]. Modelowanie obiektów biologicznych początkowo dotyczyło tkanek niekoniecznie pochodzenia ludzkiego [3]. Proces modelowania na potrzeby bioinżynieryjne wykorzystywał proste rozwiązania oparte o MES [4], które stosowane są do dzisiaj. Uproszczenia w metodzie związane są z błędami pomiarowymi, które należy uwzględnić, a ich uniknięcie nie jest łatwe [5]. Wskazują one, że w przypadku wielkoskalowych obliczeń, skomplikowania zadań i modeli – mogą narastać niejednostajnie. Rozpoczęcie symulacji należy rozpocząć od zagadnienia wnikania pola elektromagnetycznego w głąb tkanki [6]. Bez względu na to, czy generator pola elektromagnetycznego jest na zewnątrz obiektu, czy w środku – na przykład w głowie pod czaszką, styka się on z tkanką, a nie jest jej częścią. Przyjąć można mniejsze modele tkanek, ponieważ organizmy złożone są z pojedynczych komórek oraz drobnych organów [7], lecz

wymaga to odpowiedniego zamodelowania także i tej wielkości obiektów. Proces jest złożony i wymaga określenia złożoności takich modeli.

Rozwój technologii medycznej pozwolił także na powstanie tomografii komputerowej, która pozwoliła dokonać zapisu wirtualnego skanu człowieka warstwa po warstwie. Zbudowana tym samym została pierwsza baza danych w tak dokładnie przygotowanej formie – Visible Human Project (VHP)<sup>2</sup>. Zawiera zbiór medycznych skanów z których powstały przez lata siatki dwu- i trójwymiarowe osób, które podarowały lub zgodziły się na zwirtualizowanie przekroju swoich ciał po śmierci. Te właśnie modele od wielu lat są podstawą do dalszych badań nad wpływem pola elektromagnetycznego na organizmy biologiczne [8] [9] [10]. Wspomniane modele siatki w trójwymiarze dostępne są jako modele AustinMan & AustinWoman<sup>3</sup>.

Literatura obejmuje jednak również badania, które dotyczą nie samego tworzenia modelu oraz opisu parametrycznego, ale przede wszystkim metodyki obliczeniowej. Wpływ pola elektromagnetycznego na ciało ludzkie [11] jest dotychczas szeroko omówiony, podobnie jak same metody obliczeniowe. Pomiary niskich częstotliwości są istotne z punktu widzenia urządzeń, stąd badania w tym zakresie są prowadzone w obszarach medycznych [12], a nawet w obszarach motoryzacji [13].

Spośród wielu artykułów oraz badań dość interesujące okazały się analiza wpływu niskich częstotliwości na ciało ludzkie z opisem parametrycznym metody impedancyjnej modelowania tkanek [14].

Podstawowym problemem w przypadku metod modelowania numerycznego jest dobre odwzorowanie obiektu, na który będzie wpływać symulowane pole, czy jakie zmiany mają zostać w nim zobrazowane. Procesy bioelektromagnetyczne są jednak najbardziej nieobliczalnymi z nich, ponieważ łączą ze sobą kwestię zmienności w szerokim zakresie, ale także i nieznajomość samych tkanek pod względem ich struktury [15] [16].

Rozpoczynając od procesu tworzenia, trzeba uwzględnić każdy parametr modelowanego obiektu, geometrię, strukturę, czy parametry – to wszystko utrudnia przygotowanie dobrego obiektu do badań symulacyjnych. Rozwiązaniem upraszczającym proces modelowania staje się wspomniany VHP lub ogólnodostępne bazy danych [17].

<sup>&</sup>lt;sup>2</sup> Strona projektu: https://www.nlm.nih.gov/research/visible/visible\_human.html (dostęp: 2017.11.17)

<sup>&</sup>lt;sup>3</sup> Strona projektu: https://sites.utexas.edu/austinmanaustinwomanmodels/ (dostęp: 2017.11.17)

### 1.2. Cel i zakres pracy

Tematyka rozprawy dotyczy rozwoju narzędzi informatycznych wykorzystywanych podczas badań nad modelowaniem rozkładów pola elektromagnetycznego w organizmach żywych. Znane metody modelowania komputerowego wymagają ciągłej weryfikacji założeń w związku z dynamicznym rozwojem technologii informatycznych. Bezpośrednią motywacją było podjęcie próby wykorzystania możliwości oferowanych przez dzisiejsze technologie przetwarzania rozproszonego oferowane w formie usług chmury obliczeniowej. Systemy informatyczne budowane w tej architekturze charakteryzują się elastycznością i łatwą dostępnością do dużych mocy obliczeniowych, połączone z przystępnymi cenami.

Klasyczne metody modelowania rozkładu pola bazują na opisie deterministycznym, w którym budowa obiektu, a także parametry występujących w nim materiałów są dobrze znane. Oba te założenia należy podważyć w przypadku modelowania obiektów pochodzenia biologicznego, które charakteryzują się dużą zmiennością parametrów. Odpowiedzią na te wyzwania są stochastyczne metody modelowania bazujące na zmiennych opisywanych przy pomocy gęstości prawdopodobieństwa. Pełna analiza tego typu problemów jest poza zasięgiem dostępnych dotychczas mocy obliczeniowych. Jednak ostatnie pojawienie się wspomnianej już chmury obliczeniowej może stanowić przełom w tym zakresie.

W oparciu o powyższe obserwacje został sformułowany następujący cel pracy:

Opracowanie metod i narzędzi do analizy numerycznej wykorzystujących chmurę obliczeniową do modelowania problemów bioelektromagnetycznych uwzględniając stochastyczny charakter parametrów materiałowych tkanek.

Zakres badań został ograniczony do obiektów pochodzenia biologicznego, głównie tkanek, o losowo dobranych parametrach materiałowych – bez uwzględniania jednorodnych materiałów inżynieryjnych w zastosowaniach technicznych. Symulowane obiekty były modelowane w oparciu o parametr konduktywności ze względu na stacjonarne problemy elektromagnetyzmu. Nie dotyczyły one zagadnień zmiennych w czasie. Ostatecznie zdefiniowana i skonfigurowana platforma obliczeń rozproszonych z wykorzystaniem chmury obliczeniowej została przygotowana bez zagłębiania się w wewnętrzną strukturę funkcjonalną, czy zastosowane technologie informatyczne.

Praca została podzielona na rozdziały, omawiające następujące treści i zagadnienia:

- *Rozdział 1* Rozdział ten zawiera wprowadzenie do tematyki bioelektromagnetyzmu, przedstawia aktualny stan wiedzy oraz zawiera cel i zakres niniejszej pracy.
- *Rozdział 2* W rozdziale tym zawarte zostało wprowadzenie do analizy stochastycznej z wykorzystaniem wybranych metod obliczeniowych, a także przygotowane zostało opracowanie modeli tkanek na podstawie założeń statystycznych.
- Rozdział 3 Rozdział zawiera przegląd platform obliczeń rozproszonych, technologii dystrybucji zadań przeznaczonych na te platformy oraz przedstawia wyniki oraz wnioski z zastosowania tych rozwiązań w obliczeniach numerycznych.
- *Rozdział 4* W tym rozdziale omówione i zbadane zostały trzy zaimplementowane przypadki metod stochastycznych w modelowaniu tkanek z wykorzystaniem rozwiązania rozproszonego.
- *Rozdział 5* Zawiera on podsumowanie rezultatów oraz wnioski końcowe badań.
- Załącznik 1 Zawiera opis technologiczny architektury rozwiązania chmurowego oparty o platformę Microsoft Azure.
- Załącznik 2 Zawiera szczegółowy opis wdrożenia systemu obliczeń rozproszonych, wykorzystywanego do badań, na platformie Microsoft Azure.

# ROZDZIAŁ 2. PARAMETRY MODELI TKANEK

Modelowanie numeryczne to jedno z najważniejszych obecnie stosowanych narzędzi w zagadnieniach bioelektromagnetyzmu. Problem modelowania pojawia się jednak już samym początku. W większości obecnie stosowanych modeli numerycznych przyjmuje się, że są one deterministyczne – złożone w strukturze, ale o stałych wartościach parametrów materiałowych. Powoduje to wprowadzenie niepewności pomiarowej uzyskiwanych wyników, która nie odzwierciedla faktycznego stanu rzeczy. Spowodowane jest to tym, że organizmy żywe nie są homogeniczne. Tym samym należałoby nie tylko uwzględnić zmienność budowy tkanek, ale i zmienność parametrów materiałowych, które mogą przybierać wartości w szerokim zakresie.

### 2.1. Złożoność tkanek

Jedną z pierwszych kwestii, którą należy poruszyć przystępując do modelowania numerycznego, jest opracowanie obiektu badawczego. Powinien on posiadać zdefiniowane cechy, które będą wykorzystywane na etapie obliczeniowym. Są to np. kształt, struktura wewnętrzna, parametry materiałowe.

Zagadnienie opisu matematycznego obiektów biologicznych od dawna stanowi obiekt zainteresowań naukowców i badaczy zajmujących się teorią podstaw elektromagnetyzmu oraz wpływem na organizmy żywe. Obecny stan wiedzy pozwala wyróżnić setki różnych tkanek wewnątrz tylko samego ciała ludzkiego. W rozumieniu złożoności obiektów i korelacji wspomnianych wcześniej cech, stanowi to początek procesu projektowania.

Organ, jako modelowany obiekt numeryczny, jest trójwymiarowym modelem stanowiącym zbiór czworościanów w przestrzeni. Jednym z przykładów, które odzwierciedlają poruszaną kwestię jest przekrój dowolnej tkanki mięsno-tłuszczowej (Rys. 1). Jednocześnie zauważyć można, że rozmiar siatki, dla której wyznaczony obszar będzie miał przypisaną wartość parametru podczas obliczeń wskazuje na dokładność obliczeniową modelu.



Rys. 1 – Jednorodność przekroju tkanki mięsno-tłuszczowej definiowana przez siatkę modelu

Jak widać na rysunku, zamodelowanie złożonego obiektu z określoną dokładnością stanowi złożony problem w szczególności, jeżeli obiekt ma stanowić uniwersalny wzorzec obliczeniowy. Tkanki jako materiały są na tyle skomplikowane, że opisanie ich przy pomocy stałych wartości numerycznych, nie zapewnia wiarygodnego wyniku symulacyjnego.

Opisanie zaprezentowanej powyżej tkanki w formie modelu matematycznego obiektu w przestrzeni jest możliwe. Uwzględnienie dodatkowych parametrów utrudnia to zadanie. Każdy obiekt w przestrzeni dwu- lub trójwymiarowej posiada stałe parametry, które używane są podczas obliczeń. Obiekt taki nie może posiadać parametru, który opisany jest statystycznie, z określoną zmiennością.

Tworząc model przestrzenny określany jest przede wszystkim parametr wielkości siatki. Jest to zdefiniowany rozmiar najmniejszych elementów, które odpowiednio powiązane ze sobą, będą tworzyć budowany obiekt. Te elementy w takim modelu traktowane jako jednorodne. Oznacza to, że są niepodzielne parametrycznie – w całej powierzchni (2D) lub objętości (3D) zdefiniowana wartość jest niezmienna.

Nawiązując do wspomnianej siatki (Rys. 1), zamodelowana tkanka przy wykorzystaniu rozmiaru siatki o mniejszej gęstości (z lewej) oraz większej (z prawej) obrazuje kwestię jednorodności modelu obiektu rzeczywistego. Znając bowiem parametry materiałowe tkanki mięsnej oraz parametry materiałowe tkanki tłuszczowej pojawia się wątpliwość, jaką wartość numeryczną przypisać do czworoboku, w którego obszarze (lub objętości, jeśli to model 3D) znajduje się fragment dwóch lub więcej rodzajów tkanek.

Rozwiązaniem tej kwestii jest zdefiniowanie skali siatki modelu, która w przypadku odpowiednio dobranych wartości, może ograniczyć to zjawisko. Tym samym poruszana jest definicja granicy szczegółowości modelowanego obiektu, ale i celu jej stosowania dla konkretnych obliczeń.

Jak zostało wspomniane w badaniach [18], występuje zjawisko tak zwanej niepewności. Niepewność definiowana jest przez rozbieżność między stanem faktycznym (rzeczywistym), a stanem modelowym (obliczeniowym) analizowanego obiektu. Jak wspomniał autor (Tabela 1), istnieją cztery podstawowe źródła niepewności oraz związane z nimi typy ograniczeń:

- model matematyczny,
- błędy numeryczne,
- geometria kształtu,
- właściwości tkanek.

Źródło niepewności	Poziom, typ ograniczenia
Własności tkanek	1000-3000%, aleatoryczny
Geometria kształtu	10-50%, epistemiczny
Błędy numeryczne	1-5%, aleatoryczny
Model matematyczny i fizyczny	nieokreślony, epistemiczny

Tabela 1 - Źródła niepewności pomiarowych względem ich poziomu oraz typu [18]

Wymienione źródła posiadające wspomnianą niepewność, związane są danym etapem projektowania modelu. Poziomy niepewności oraz typy ograniczenia to zdefiniowane w literaturze [19] cechy danego obiektu. Niektóre z nich mogą być zredukowane lub ograniczone.

Niepewność aleatoryczna, to typ niepewności nawiązującej do zmienności parametrycznej, elementów statystyki i rachunku prawdopodobieństwa. Jest to niepewność nieredukowalna, możliwa tylko do oszacowania lub przybliżenia w określonych warunkach metodycznych.

Niepewność epistemiczna z kolei, to niepewność wynikająca z limitu poznania, aktualnego stanu wiedzy lub np. dokładności pomiarowych urządzeń. Jej ograniczenie możliwe jest przez zastosowanie innych technik pomiarowych, czy sposobu modelowania.

Tkanki, jako obiekty, są złożone w swojej strukturze. Uwzględniane parametry materiałowe fizyczne, elektryczne, czy biologiczne zwykle są dla potrzeb badawczych uzyskiwane z ogólnodostępnych baz danych (publicznych lub naukowo-specjalistycznych) - np. IT'IS [20]. Zwykle takie parametry podawane są jako jedna, stała wartość numeryczna lub przedział zmienności. Dane te są często uzyskane doświadczalnie podczas pomiarów tkanek, zwykle invitro. Pomiary takie jednak nie są ujednolicone – wykonywane w różnych odstępach czasu, na przestrzeni wielu lat, w różnych warunkach laboratoryjnych, z zastosowaniem różnej klasy sprzętu pomiarowego. Wprowadzają one niepewności na poziomie nawet kilku tysięcy procent. W przypadku specyficznych wartości pomiarowych niektórych parametrów (np. konduktywności), rozbieżność na tym poziomie jest numerycznie niedopuszczalna. Zastosowanie przedziału zmienności takiej niepewności powodować może otrzymanie wartości ujemnych, co numerycznie jest niepoprawne. Jednocześnie warto zauważyć, że sprawdzone i zweryfikowane ogólnodostępne bazy danych zawierają pomiary oparte o kilka kilkanaście próbek. Tym samym nie stanowią wiarygodnej podstawy pomiarowej i parametrycznej, którą można wykorzystać do stworzenia rzeczywistego modelu obiektu tkanki żywej.

Geometria kształtu stanowi drugie wspomniane źródło niepewności. Modelowanie geometrii danej tkanki związane jest z jej wymiarami, szczegółowością, ale i po raz kolejny – zmiennością. Tworząc geometrię, która odzwierciedla rzeczywisty obiekt, nie można przyjmować jednej struktury dla wszystkich przypadków. Niezbędne staje się dokonanie pewnych przybliżeń, statystycznych uśrednień. Określenie geometrii modelu serca nie jest niemożliwe. Znane są bowiem parametry, budowa fizyczna, a nawet cykl oraz zakres otwierania i zamykania komór przez zastawki. Kwestią istoty modelowania jest jednak zaprojektowanie obiektu tak, by uwzględniał naturalne i środowiskowe cechy zmienności. Dotyczy to np. płci człowieka, wzrostu, niedoskonałości samej budowy, objętości fizycznej, czy zmian genetycznych. Takie podejście do modelowania, z uwzględnieniem wspomnianych cech sprawia, że uzasadnione staje się uśrednienie tych parametrów do budowy geometrii. Powoduje jednak powstanie niepewności w pewnym zakresie – wahającym się od 10 do 50%.

Błędy numeryczne stanowią o kolejnej niepewności, którą należy uwzględnić podczas tworzenia modelu złożonego obiektu, jakim jest wspomniana tkanka. Niepewności w tym obszarze związane są z zastosowaną metodą numeryczną. Błędy danych wejściowych, zaokrągleń, obcięcia oraz metody stanowią o poziomie ograniczenia. Są one możliwe do zredukowania w niewielkim stopniu, przez zastosowanie adaptacyjnych metod obliczeń, czy współczynników korekcji. Niepewność numeryczna osiąga poziom 1-5%.

Ostatnim uwzględnionym źródłem niepewności jest sam model. Obliczenia numeryczne obejmują tylko zdefiniowane matematycznie wybrane własności i parametry. Wykorzystane są do wyliczenia zwykle jednego wybranego zagadnienia. Zwiększenie liczby jednoczesnych zjawisk fizycznych wpływających na obiekt jest istotna. Uwzględnienie np. wpływu pola magnetycznego wraz z przepływającym prądem w przewodniku sprawia, że niepewność wyniku symulacji opartej o model zaczyna rosnąć. Poziom niepewności jest tym samym nieokreślony. Każda ze składowych posiada bowiem wartość niepewności. Korelacja takich parametrów jest możliwa do zasymulowania. Obliczenia są tym samym skomplikowane, ale wyniki stanowią o dokładności.

Jak można zauważyć, niepewności pojawiają się na każdym etapie projektowania modelu numerycznego. Jedne wynikać mogą z drugich lub być ze sobą powiązane. Istotną kwestią modelowania staje się zdefiniowanie złożoności modelu. Określenie siatki geometrii, czy zmienności parametrów oraz przyjęcie pewnych przybliżeń pozwala na zamodelowanie obiektu. Jednak nawet takie działania nie dają pewności czy dany model będzie na tyle dokładny, aby przeprowadzane na nim obliczenia numeryczne dały wiarygodne wyniki z oczekiwaną dokładnością.

W przypadku obiektów, których struktura jest homogeniczna, zaprojektowanie modelu numerycznego jest łatwiejsze. Tym samym redukuje się nie tylko ilość źródeł niepewności, ale także ich skala. Mniej złożona struktura oznacza mniej równań opisujących zależności względem tkanek. Jednak rozpatrywany bioelektromagnetyzm związany jest z organizmami żywymi, biologicznymi, które mają bardziej złożoną formę. Oznacza to, że liczba zmiennych jest większa, a tym samym konieczne staje się uwzględnienie ich podczas projektowania obiektu.

11

### 2.2. Rozkłady gęstości prawdopodobieństwa

Parametry materiałowe nie są jednak stałe i zależą od wielu czynników – uwarunkowań genetycznych, wieku, czy środowiska. Wspomniane modele wirtualne zbudowane na podstawie pobranych danych są zwykle opisane znanymi parametrami o stałych wartościach. W przypadku obiektów technicznych, takie modelowanie jest uzasadnione. Dla obiektów żywych nie ma zastosowania z powodu zmienności parametrycznej.

W badaniach w literaturze [18] [21] zostały rozpatrzone możliwe modele zmienności, jako krzywe rozkładu prawdopodobieństwa parametrów biologicznych. Teoria prawdopodobieństwa o którą rozkłady są oparte jest dziedziną matematyki zajmującą się zjawiskami nie mającymi cech deterministycznych. W przypadku parametru zmiennego wewnątrz populacji, bądź w czasie, pojawia się potrzeba zdefiniowania jak opisać taki parametr.

Zwykle przyjmuje się parametry opisywane są z danym prawdopodobieństwem ich wystąpienia oraz możliwą zmiennością, określaną jako odchylenie standardowe. Oznacza to, że w zbiorze danych wartość średnia ( $\mu$ ) występuje najczęściej, natomiast odchylenie standardowe ( $\sigma$ ) jest zakresem zmienności. Zwykle symetrycznym względem średniej. Zależności opisuje równanie:

$$f_{\mu,\sigma}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\left(\frac{-(x-\mu)^2}{2\sigma^2}\right)}$$
(1)

Funkcja jest nieskończona w obie strony przedziału, jednak dla wartości funkcji będącej średnią określane jest też prawdopodobieństwo, jak zmienna X jest bliska lub równa wartości średniej. Określa to reguła dystrybuanty, zwana także "regułą trzech sigm".

$$P(X \le x) = \int_{-\infty}^{x} \left( \frac{1}{\sigma\sqrt{2\pi}} e^{\left(\frac{-(x-\mu)^2}{2\sigma^2}\right)} \right) dx$$
(2)

Regułę (2) opisuje także wykres (Rys. 2):



Rys. 2 - Ilustracja "reguły trzech sigm"<sup>4</sup>

Tym samym prawdopodobieństwo dające pewność 99,6% wystąpienia wartości X mniejszej lub równej x zawiera się w przedziale  $\pm 3\sigma$ . Jednak rozkład taki jest dość optymistyczną wersją rozwiązania. Zakłada, że parametry większe są tak samo prawdopodobne jak parametry mniejsze w przedziale trzykrotności wartości  $\sigma$ . Nie koniecznie musi być to prawdą, ponieważ zakłada on symetrię względem wartości  $\mu$ .

W przypadku małych wartości, jakie mogą przyjąć parametry materiałów biologicznych, wykorzystanie takiego rozkładu prawdopodobieństwa może sprawić, że wartości będą znajdować się drugiej ćwiartce układu kartezjańskiego, czyli będą ujemne. Dla materiałów biologicznych wartości ujemne nie mają zastosowania i są nieprawidłowe.

Rozwiązaniem takiego przypadku jest zastosowanie rozkładu funkcji gęstości prawdopodobieństwa, który uwzględnia jako wartość średnią dodatnią zmienną losową. Rozkład Logarytmiczny-normalny (ang. Log-normal) uwzględnia dodatnie parametry zmiennych losowych. W przeciwieństwie do wspomnianego wcześniej rozkładu normalnego, niezbędne jest wyliczenie parametrów średniej (m) oraz wariancji (v) z (3), (4):

$$m = e^{\left(\mu + \frac{\sigma^2}{2}\right)} \tag{3}$$

$$v = e^{(2\mu + \sigma^2)} \cdot (e^{\sigma^2} - 1) \tag{4}$$

<sup>&</sup>lt;sup>4</sup> Źródło: https://en.wikipedia.org/wiki/Standard\_deviation (dostęp: 2017.11.17)

Tym samym wartości funkcji gęstości prawdopodobieństwa dla rozkładu opisane są jako:

$$f(x) = \begin{cases} \frac{1}{\sigma x \sqrt{2\pi}} e^{\left(\frac{-(\ln x - \mu)^2}{2\sigma^2}\right)}, & x > 0\\ 0, & x \le 0 \end{cases}$$
(5)

### 2.3. Narzędzia statystyczne

Wspomniane w poprzednim podrozdziale zagadnienie modelowania obiektu wiąże się z określoną niepewnością tego obiektu. Wśród wskazanych źródeł niepewności, niektóre można zredukować, w przypadku innych – nie jest to możliwe. Źródła te są jednak wyzwaniem, które stanowi podstawę rozważań i działań prowadzonych w kierunku rozwijającej się medycyny komputerowej oraz pochodnych dziedzin. Opisane badania nad zmiennością tkanek [18] [22] pokazują, z jakim problemem spotykają się osoby zajmujące tą tematyką. Zagadnienie zmienności materiałów dotyczy wielu dziedzin także w przemyśle – nie tylko organicznym, jak wspomniana medycyna, czy przetwórstwo żywności, ale także obróbki drewna, czy górniczym.

Każda z tych dziedzin wymaga precyzyjnego zaprojektowania modelu obiektu, będącego wzorcem stanowiącym podstawę do badań, czy obliczeń symulacyjnych. Kluczowym, z punktu dzisiejszego środowiska naukowego, zagadnieniem związanym z obliczeniami numerycznymi staje się temat modelowania stochastycznego. Standardowa forma obliczeń stochastycznych opiera się o pewne zmienne losowe, które wykorzystywane są do rozwiązania problemu.



Rys. 3 – Przykładowa siatka zdyskretyzowanego modelu tkanki w 2D

Metody stochastyczne można podzielić na dwa podstawowe rodzaje – bezpośrednią oraz pośrednią. Bezpośrednia jest przykładem wprost realizującym zagadnienia matematyczne, np. stochastyczna Metoda Elementów Skończonych (ang. FEM) [23]. Jest ona łatwa w implementacji oraz wydajna dla prostych obliczeniowo problemów. Zastosowanie dyskretyzacji modelu pozwala na zredukowanie złożonego zagadnienia do mniejszej liczby równań cząstkowych. Tym samym obliczenia są wykonywane dla uproszczonych równań. Dla przykładowego modelu dwuwymiarowej tkanki, dyskretyzacja (Rys. 3) pozwala na wyliczenie wartości numerycznych danego zagadnienia jako sumy wartości zagadnień mniejszych obszarów.

W przypadku wielkoskalowych symulacji, w których parametry obiektów są określanie z zadaną niepewnością, użycie metody analitycznej zwiększałoby liczbę przypadków obliczeniowych wydłużając jednocześnie czas obliczeń. Zmniejszenie z kolei liczby przypadków mogłoby nie uwzględnić niektórych zależności lub złożoności wewnątrz modelu. W takiej sytuacji rozwiązaniem jest użycie metod stochastycznych w celu pośredniego wyznaczenia rozwiązania. Do takich metod należą między innymi Metodę Chaosu Wielomianowego (ang. PCE) [24] [25] czy klasyczna metoda stochastyczna Monte Carlo (ang. MC).

Jak wspomniano wcześniej, obiekty pochodzenia biologicznego są bardziej skomplikowane niż inżynieryjne (techniczne). Ilość parametrów opisujących taki obiekt jest większa, ale również parametry zdefiniowane są pewną zmiennością. Dlatego też klasyczne podejście (deterministyczne) nie jest rozwiązaniem, które uwzględnia tę zależność.

Tradycyjne rozwiązanie wykorzystujące rozkład gęstości prawdopodobieństwa (PDF) wymaga przygotowania dużej liczby niezależnych przypadków. Stanowią one rozwiązania cząstkowe. Zebrane razem w formie histogramu, stanowią rozkład prawdopodobieństwa danego rozwiązania w stosunku do innych.

#### 2.3.1. Założenia stochastyczne modelowania tkanek

W badaniach opisanych w [26] założono zamodelowanie rozkładu gęstości mocy powstałej przez przepływ prądu. Model został opisany w sposób deterministyczny przy pomocy równania Laplace'a z warunkami brzegowymi Dirichleta definiującymi źródło napięcia na elektrodach (6), a następnie z równania (7).

$$\nabla \cdot \sigma \nabla \varphi = 0 \tag{6}$$

gdzie:  $\varphi$  – potencjał elektryczny,  $\sigma$  – konduktywność.

$$p = \vec{E} \cdot \vec{J} = \nabla \varphi \cdot \sigma \nabla \varphi \tag{7}$$

gdzie:  $\vec{E}$  – wektor natężenia pola elektrycznego,  $\vec{J}$  – wektor gęstości prądu.

Rozwiązanie zagadnienia zostało obliczone w formie klasycznej, wykorzystując FEM. Model geometryczny wraz z parametrami ( $\sigma$ ) został opisany funkcją (f). Model stochastyczny jest formą rozszerzoną modelu deterministycznego, która uwzględnia zmienność parametrów modelu. W opisywanym przypadku wszystkie parametry zostały opisane zmienną losową (Q), a są to:

- konduktywność,
- geometria próbek,
- źródło napięcia.

Badania pokazały, że przeprowadzone na tkance biologicznej pomiary parametru elektrycznego konduktywności dla 175 próbek umożliwiły na stworzenie osobistego zbioru danych. Zebranych zostało 8 gatunków bulw ziemniaczanych dla których uzyskane wartości konduktywności utworzyły histogram rozkładu wartości parametru (Rys. 4, po lewej).



Rys. 4 - Histogram wartości konduktywności badanej tkanki (po lewej) oraz przybliżony wykres PDF (po prawej) w zależności od liczby próbek [S/m]

Dla wspomnianego zbioru wartości konduktywności zbadanych próbek, został wyznaczony parametr σ. Dla tego parametru wygenerowany został wykres funkcji gęstości

prawdopodobieństwa, będący funkcją logarytmiczną normalną. Przybliża on rozkład funkcji bazując na zebranych pomiarach próbek (Rys. 4, po prawej).

Model matematyczny, oparty o badania wykonane na wspomnianych bulwach, zakłada wyliczenie wartości konduktywności dla próbek wielkości 3x4 cm. Próbka podłączona jest do źródła napięcia o wartości 30 V przy pomocy dwóch elektrod umieszczonych po obu stronach modelu. Parametry modelowanego przypadku, jak geometria, czy wartość napięcia, zostały opisane przy pomocy zadanej niepewności, z zastosowaniem rozkładu Gaussowskiego. W celu zwiększenia precyzji pomiarowej oraz symulacji rzeczywistych parametrów modelu, zmienność geometrii przyjęto  $\pm 10\%$ , wartość napięcia na operowanym generatorze  $\pm 1\%$ . Gęstość siatki modelu to 1 mm, przyjęty rozkład losowości parametru konduktywności, zgodnie z wynikami badań (Rys. 4), to logarytmiczny normalny (log-normal).

Symulacja omawianego modelu została wyliczona z wykorzystaniem dwóch wspomnianych wcześniej metod. Zastosowanie dwóch jednocześnie miało na celu porównanie ich skuteczności w zagadnieniu stochastycznego modelowania tkanek w praktycznym przypadku inżynieryjnym.

#### 2.3.2. Metoda Monte Carlo (MC)

Monte Carlo jest pierwszą stosowaną metodą w omawianym przypadku modelowym. Pozwala zastosować deterministyczne rozwiązania w stochastycznych założeniach. Opiera się na idei generacji parametrów, których wartość jest określana przez rozkład funkcji gęstości prawdopodobieństwa. Skuteczność tej metody definiowana jest przez ilość rozpatrywanych przypadków deterministycznych.

W przypadku metody MC dobór wartości parametrów odbywał się w losowy sposób. Wygenerowane wartości parametrów wejściowych (*Q*) zgodnych z założonych rozkładem posłużyły w dalszych obliczeniach już przeprowadzanych jako rozwiązanie klasyczne. Metoda zachowuje założoną informację statystyczną, jednak wymaga znacznej liczby próbek, aby osiągnąć stabilne rozwiązanie problemu.



Rys. 5 – Histogramy wartości mocy całkowitej dla symulacji przy użyciu metody MC w zależności od liczby próbek [W]

### 2.3.3. Metoda Chaosu Wielomianowego (PCE)

Drugą metodą stosowaną w omawianym przypadku modelowym jest Metoda Chaosu Wielomianowego (PCE) (Rys. 6). Polega na stworzeniu modelu przybliżonego funkcji f(Q) będącej sumą wielomianów cząstkowych w postaci:

$$\hat{f}(Q) = \sum_{i=0}^{m} c_i \Phi_i(Q) \tag{8}$$

gdzie:  $c_i$  – współczynnik wielomianu cząstkowego,  $\Phi_i$  – wielomian ortogonalny.



Rys. 6 - Schemat ideowy metody PCE w oparciu o [24]

Jak opisuje literatura [27] [28], założeniem metody PCE było wykorzystanie podstaw teorii wielomianów aproksymujących ortonormalnych w celu przybliżenia rozwiązania wyjściowego. Dla szeroko stosowanego w statystyce rozkładu Gaussowskiego, zastosowanie tej metody zostało zbudowane o Wielomiany Hermite'a [29]. PCE opiera się zatem o obliczenia wielomianowe, przybliżające funkcję ciągłą sumą wielomianów cząstkowych.

#### 2.3.4. Porównanie metody MC oraz PCE

Przeprowadzone symulacje wykonane zostały w dwóch niezależnych środowiskach obliczeniowych. Rozwiązanie wprost zrealizowane zostało w środowisku FEniCS<sup>5</sup>, które jest oprogramowaniem wyspecjalizowanym w obliczeniach numerycznych FEM, opierającym się o język programowania Python. Rozwiązania stochastyczne – zarówno MC, jak i PCE wykonane zostało w oprogramowaniu Chaospy<sup>6</sup> [30]. Stanowi zestaw narzędzi do obliczeń opartych o teorię chaosu, w tym wyżej wymienione metody.

Jak widać (Rys. 5), metoda Monte Carlo jest wolno zbieżna. Dla próbek rzędu n = 10, histogram wyników reprezentujących rozwiązanie zagadnienia jest trudny do zdefiniowania. Zwiększenie liczby próbek stanowiących wygenerowane wartości parametrów wejściowych pozwoliło na otrzymanie wyniku zgodnego z oczekiwaniami. Wartości rozkładu gęstości mocy dla liczby próbek n = 200 pokazuje, że wyniki spełniają rozkład normalny (Gaussowski).

<sup>&</sup>lt;sup>5</sup> Strona projektu: https://fenicsproject.org (dostęp: 2018.07.25)

<sup>&</sup>lt;sup>6</sup> Strona projektu: http://chaospy.readthedocs.io/ (dostęp: 2018.11.05)

	średnia	odch. std.	skośność	kurtoza	czas [s]
Monte Carlo (n = 10)	265,42	14,72	0,333	-0,268	30
Monte Carlo (n = 30)	265,96	18,62	-0,363	-0,695	93
Monte Carlo (n = 100)	264,17	20,01	0,11	0,303	290
Monte Carlo (n = 1000)	264,50	19,40	0,238	0,242	2943
PCE $(n = 10, m = 0)$	269,44	0	0	-3	31
PCE (n = 10, m = 1)	263,06	20,38	0,002	-0,022	31
PCE $(n = 10, m = 2)$	196,64	266,87	-1,024	1,559	31
PCE $(n = 10, m = 3)$	331,89	231,33	3,507	26,15	31

Tabela 2 - Statystyczne podsumowanie metod numerycznych MC/PCE względem czasu obliczeń

Istotną kwestią w obliczeniach jest wydajność. Określa czas wykonywania obliczeń względem liczby przypadków oraz skomplikowania modelu. Statystyczne dane (Tabela 2) pokazują, że złożoność obliczeniowa dla metody PCE nie wpływa na czas obliczeń. W przeciwieństwie do metody MC, który wzrasta proporcjonalnie do liczby wyliczanych równań. Czas obliczeń jest równy czasowi pojedynczego wyliczenia numerycznego wymnożony przez ilość równań.

W tabeli powyżej uwzględnione są jeszcze dwa dodatkowe parametry, które precyzują wyliczony rozkład stochastyczny. Są to skośność oraz kurtoza.

Skośność (ang. skewness) to współczynnik asymetrii rozkładu względem wartości średniej. Wyliczany z (9) przyjmuje wartości ujemne, dla rozkładu rozciągniętego po lewej stronie osi symetrii, wartości dodatnie – dla rozkładu rozciągniętego po prawej stronie osi symetrii (Rys. 7).

$$S = \frac{\mu - mode}{\sigma} \tag{9}$$

gdzie:  $\mu$  – średnia,  $\sigma$  – odchylenie standardowe, *mode* – dominanta.


*Rys.* 7 - Skośność ujemna oraz dodatnia rozkładu prawdopodobieństwa<sup>7</sup>

Kurtoza [31] jest współczynnikiem wykorzystywanym w teorii prawdopodobieństwa do określenia spłaszczenia układu (Rys. 8) określanym wzorem (10):

$$K = \frac{\mu_4}{\sigma^4} \tag{10}$$

gdzie:  $\mu_4$  – czwarty moment,  $\sigma$  – odchylenie standardowe.

Zdefiniowany powyżej czwarty moment  $\mu_4$  opisywany jest wzorem (11):

$$\mu_4 = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^4 \tag{11}$$

gdzie:  $X_i$  – kolejna wartość zbioru danych,  $\overline{X}$  – średnia zbioru danych.



Rys. 8 – Rozkład PDF dla dystrybucji normalnej w zależności od współczynnika kurtozy

Dla wyliczonych przypadków dwoma metodami, zauważyć można interesujący fakt w przypadku zastosowania PCE współczynnikami n = 10 oraz m = 1. Przy niewielkiej liczbie próbek (n) oraz dla wielomianu stopnia pierwszego (m) pokrycie wyników tej metody

<sup>&</sup>lt;sup>7</sup> Źródło rysunku: https://en.wikipedia.org/wiki/Skewness (dostęp: 2018.11.17)

z rozwiązaniem wyliczonym przy użyciu MC jest najbardziej zbliżone. Jednocześnie PCE jest metodą szybką (czas obliczeń to 31 sekund), dającą akceptowalne rezultaty (przy odpowiednim dobraniu parametrów i stopni wielomianów). Wadą PCE jest wynikowa wartość współczynnika skośności oraz kurtozy, które znacząco odstają od założeń rozkładu wyliczonego dla MC (Rys. 9).





Podsumowując, złożone metody stochastyczne rozwiązywania układów równań, niebędące tradycyjnymi rozwiązaniami stacjonarnymi, ze względu na zmienność parametrów, stanowią poważne wyzwanie obliczeniowe. Dostępne metody są skuteczne, jeżeli budowane są w oparciu o odpowiednio dużo danych wejściowych. Taką metodą jest MC, które jest optimum dla nawet dość złożonych modeli, posiadających znaczną ilość parametrów. Wprawdzie opiera się w ostatecznych obliczeniach o jedno rozwiązanie stacjonarne, to ilość przypadków uwzględniana w tej metodzie pozwala na przybliżenie wyników względem założeń stochastycznych. Wadą MC jest zapotrzebowanie na moc obliczeniową.

Czas ten można skrócić wykorzystując nowocześniejsze metody, jak PCE, jednak potencjalna różnica w stosunku do oczekiwanego rozwiązania oraz skali błędów jest duża, co wymaga odpowiedniego dopracowania warunków rozwiązania równań.

W przypadku wspomnianych w poprzednim podrozdziale obiektów złożonych, metoda PCE może mieć znaczący udział w sprawności obliczeń numerycznych. Jednak rosnąca ilość parametrów, złożoność samego modelu oraz konieczność odpowiedniej parametryzacji wielomianowej może wpłynąć także negatywnie na uzyskiwane wyniki.

W obecnej sytuacji, która zakłada wykorzystanie obliczeń równoległych w przypadkach numerycznych, to metoda MC jest racjonalnym wyjściem. Wymagania związane z mocą obliczeniową nie zmniejszą się, ale możliwość rozłożenia obciążenia przez zrównoleglenie daje pewne możliwości [32] [33] [34].

## 2.4. Tworzenie modelu statystycznego tkanki

Mając zdefiniowane warunki dotyczące parametryzacji tkanek oraz teorię zmienności, z wykorzystaniem metod statystycznych, można rozpocząć proces budowania modelu obiektu. Wspomniana na początku tego rozdziału, baza danych materiałów organicznych, zwykle mimo dostępności, jest ograniczona. Różnice w parametrach elektrycznych tkanek zostały już wspomniane w literaturze [35]. Brak jednolitej standaryzacji pomiarowej, ciągły rozwój technik pomiarowych, zmieniająca się technologia sprawiają, że wyniki oraz ich skala zmienić mogą się na przestrzeni lat.

Dlatego celem badań stało stworzenie metodologii badawczej pozwalającej na poznanie struktur materiałów, sparametryzowanie ich pomiarowo oraz określenie wymagań niezbędnych do poprawnej akwizycji danych z tkanek. Takie parametry pozwolą na stworzenie osobistej bazy danych, ale także na zweryfikowanie uzyskanych wyników z ogólnodostępnymi.

Istotną kwestią, oprócz samych pomiarów, jest zmienność tkanek niezależna całkowicie od badań. Zmiany pór roku, wiek, choroby, zmiany fizjologiczne czy wpływ środowiska sprawiają, że dynamika parametrów materiałów biologicznych jest nieprzewidywalna. Nawet w przypadku prostej tkanki mięśnia, zmienność osiąga nawet 50% [36].

Struktura ciała jest zmienna, a każdy organ, tkankę można opisać pod względem zadanej niepewności. Wyjaśnienie zagadnienia zmienności pod względem wielkości tkanek, zależy od unikalnej mikrostruktury oraz powiązanej z nimi różnicy na poziomie makroskopowym [37]. To sprawia, że algebra zmiennych losowych staje się istotnym zagadnieniem [38] dla tego typu problemów.

Wspomnianą w poprzednim podrozdziale odpowiednią metodą do opisu zmienności tkanek jest algebra zmiennych losowych. Aby móc potwierdzić dany rozkład prawdopodobieństwa dla wybranej tkanki, niezbędne staje się dokonanie znacznej liczby niezależnych od siebie pomiarów. Obecny stan wiedzy w literaturze oraz źródłach pochodnych nie zapewnia pokrycia

takiej liczby zapotrzebowania na parametry materiałowe wszystkich tkanek. Nawet jedna z popularnych baz danych parametrów tkanek ludzkich, IT'IS [20], ma w swoich zasobach mniej niż 10 pomiarów dla każdej z tkanek. Podawane wartości są w dodatku uśrednione z parametrów źródłowych. Stanowią tym samym wątpliwą wiarygodność w badaniach symulacyjnych.

Powyższe problemy stały się inspiracją do własnych badań w zakresie pomiarowym oraz opracowania technologii pozwalającej na rozwiązanie kwestii liczby sparametryzowanych próbek tkanek biologicznych.

#### 2.4.1. Akwizycja danych

Głównym parametrem materiałowym w obliczeniach symulacyjnych badań bioelektromagnetycznych jest konduktywność elektryczna (σ), która w dla badanego materiału jednorodnego może zostać wyliczona z równania:

$$\sigma = \frac{l}{R \cdot s} \tag{12}$$

gdzie: l – odległość pomiędzy elektrodami, s – powierzchnia elektrod, R – zmierzona rezystancja badanej próbki [39].

Rozważania te stały się ideą do zbudowania własnego urządzenia, pozwalającego na badanie materiałów zarówno pochodzenia biologicznego, jak i obiektów o strukturze jednorodnej



*Rys.* 10 - Model CAD 3D (po lewej) oraz zbudowane i podłączone urządzenie pomiarowe (po prawej)

Urządzenie (Rys. 10) zostało skonstruowane na Politechnice Warszawskiej w warsztacie technicznym Wydziału Elektrycznego. Zbudowane jest z dwóch elektrod ze stali nierdzewnej 316L umieszczonych horyzontalnie na dwóch niezależnych od siebie częściach całego statywu. Dolna część jest stała i zintegrowana z całym urządzeniem. Górna elektroda

umieszczona jest na ruchomej aluminiowej płycie, połączona ze śrubą, która pozwala na przemieszczenie się elektrody w pozycji góra-dół. Oprócz śruby sterującej pracą płyty, obecne są dwie prowadnice, które zapewniają stabilność podczas ruchu. Dla zapewnienia wysokiej precyzji położenia obu płyt względem siebie, został zamontowany liniał wraz z cyfrowym wyświetlaczem o dokładności 0.01 [mm]. Sam wyświetlacz posiada funkcję kalibracji oraz co ważne – pamięć ostatniego położenia, co daje możliwość powtarzalności pomiarowej.

Wyprowadzone z tyłu urządzenia zaciski podłączone zostały do cyfrowego mostka rezystancyjnego (model: GW Instek LCR-8110G). Mierzy on rezystancję dzieląc wartość wmuszonego napięcia (U<sub>m</sub>) przez wartość prądu (I<sub>m</sub>). Podczas prowadzonych pomiarów użyte było napięcie o małej wartości (2V) przy częstotliwości 50 Hz. Wartości te nie zostały dobrane przypadkowo, ale w celu zredukowania wpływu prądu na materiał biologiczny.

Również wspomniana baza danych IT'IS zapewnia dane dla parametrów tkanek dla niskich częstotliwości. Obszar zainteresowań wpływu niskich częstotliwości na tkanki nie został tym samym zdefiniowany przypadkowo [40].



*Rys.* 11 - Pomiar rezystancji w połączeniu dwuprzewodowym (a) i czteroprzewodowym (b)<sup>8</sup> [41]

<sup>&</sup>lt;sup>8</sup> Według dokumentacji pomiarowej producenta miernika rezystancji, firmy Keithley

Do pomiarów wartości konduktywności używane są dwa podstawowe typy połączeń: dwuprzewodowe oraz czteroprzewodowe (Rys. 11) [41]. Pierwszy z nich to połączenie wykorzystujące dwie elektrody, które są jednocześnie łącznikami dla prądu i napięcia. Obie końcówki są podłączone do źródła prądu, a mierzony jest spadek napięcia na elektrodach.

Drugi typ połączenia wykorzystuje do pomiaru cztery niezależne przewody – dwa przepływu wymuszonego prądu, dwa napięcia. Równocześnie takie rozwiązanie sprawia, że efekty związane z polaryzacją elektrody są zmniejszone [36], dzięki temu takie rozwiązanie staje się wartościowym dla pomiarów parametrycznych tkanek. Chociaż w przypadku pomiarów małych wartości rezystancji (poniżej 1 k $\Omega$ ) jest to zalecana metoda, dla układu wspomnianego urządzenia użyte zostanie połączenie o zredukowanej liczbie przewodów. Głównym powodem, dla którego takie połączenie zostanie wykorzystane jest badanie tkanek niejednorodnych.



Rys. 12 - Przykład połączenia dwu- oraz czteroprzewodowego dla tkanki mięsno-tłuszczowej

W przypadku pomiarów wykonywanych dwu- oraz cztero-przewodowo zakłada się, że obiekt badany jest jednorodny w swojej strukturze. W rzeczywistości obiekt ten (Rys. 12) może nie spełniać tego założenia.



*Rys.* 13 - Schemat pomiarowy urządzenia ( $R_c$  – rezystancja styku między elektrodą, a próbką  $R_s$ ) [42]

Błąd pomiarowy wzmacniany jest w metodzie czteroprzewodowej, której cztery elektrody mogą stykać się z różnymi tkankami, a tym samym pomiar rezystancji nie będzie prawidłowy. W przypadku dwuprzewodowego połączenia, wartość błędu związana jest przede wszystkim z uśrednieniem dla badanego obiektu w jego objętości. Zakłada ona, że obiekt nie jest idealnie jednorodny, jednak metodyka przybliża obiekt do takich wymagań.

Pomiar konduktywności w oparciu o wzór (12) oraz budowę wspomnianego urządzenia realizowany jest przez diagram pomiarowy (Rys. 13). Istotnym elementem, który jest uwzględniany w tym układzie pomiarowym jest wartość  $R_c$ . Wartość ta określa rezystancję styku między elektrodą pomiarową, a badaną próbką. Rezystancja mierzona jest połączeniem szeregowym trzech składowych wartości rezystancji: rezystancji próbki ( $R_s$ ) oraz wspomnianej rezystancji styku ( $R_c$ ). Zatem:

$$R_m = 2R_c + R \tag{13}$$

Jeśli dodatkowo założymy, że rezystancja powierzchni styku określona jest przez  $\rho_c \ [\Omega \cdot m^2]$ , wówczas korzystając z równania (12), otrzymamy:

$$R_c = \frac{\rho_c}{s}, R = \frac{l}{\sigma s} \tag{14}$$

Tym samym podsumowując i łącząc oba równania dla mierzonej rezystancji (*R<sub>m</sub>*) otrzymamy:

$$R_m = \frac{2\rho_c}{s} + \frac{l}{\sigma s} \tag{15}$$

Powyższe równanie zawiera w sobie znane nam już wartości, a są to: powierzchnia styku (s) oraz wymiar odległości między elektrodami, czyli wysokość próbki (*l*). Wiedząc, że każda

próbka będzie w formie prostopadłościanu, wymiar *l* będzie jednocześnie wysokością próbki, jak i po jej obróceniu – podstawą *s*. Możemy zatem potroić każdy pomiar uzyskując trzy razy więcej wyników pomiarowych ze wszystkich próbek.

W wyniku doświadczeń pomiarowych wyznaczone błędy pomiarowe, metodą wielokrotnych pomiarów, zostały oszacowane na poziomie 0.5 mm dla wymiaru fizycznego próbki oraz 5  $\Omega$  dla rezystancji. W przypadku próbki o wymiarach każdej z krawędzi 1 cm (V = 1 cm<sup>3</sup>), wartość błędu konduktywności to w przybliżeniu 10%.

#### 2.4.2. Pomiary materiałów jednorodnych

Posiadając zbudowane urządzenie pomiarowe, opisane w poprzednim podrozdziale, kolejnym krokiem w badaniu parametrów tkanek było przygotowanie metodyki pomiarowej. Celem jest ustandaryzowanie procedur pomiarowych w taki sposób, który zapewni ich powtarzalność. Drugą kwestią jest opracowanie schematu postępowania i kolejności wykonywania operacji, do poprawnego wykonania pomiaru parametrów obiektów.

Zanim zostaną przebadane tkanki heterogeniczne, należy najpierw zweryfikować założenia na tkankach homogenicznych. Materiały jednorodne są bardziej przewidywalne. Tym samym pozwalają na pewne uproszczenia, w celu wypracowania wspomnianej metodyki pomiarowej.

Do badań nad tkankami homogenicznymi został wybrany żółty ser, jako obiekt pochodzenia biologicznego. Wybrany gatunek jest wolny od pęcherzyków, które podczas cięcia mogłyby zafałszować pomiary. Oprócz błędu związanego z objętością, potencjalne dziury na krawędzi zmieniłyby także powierzchnię styku z elektrodami pomiarowymi. Badania prowadzone na jednorodnym materiale badawczym, pozwoliły zaobserwować zjawiska, których pominięcie mogłoby uniemożliwić powtarzalność pomiarów.

**Temperatura**, jako czynnik znaczący w badaniach wykazał, że odpowiednie przechowywanie tkanek laboratoryjnych w celu późniejszego wykonywania badań jest istotny. Podobnie jak opisano w literaturze [43], gdzie temperatura może służyć do podgrzewania tkanek, tak w przypadku zapewnienia ich stabilności parametrycznej, zachowanie niskiej temperatury jest kluczowe. Jednym z głównych badań nad tkanką homogeniczną było zweryfikowanie, jak próbka ochłodzona do temperatury 4 °C będzie zmieniała swoje parametry w pomieszczeniu, którego temperatura jest stała i jest równa 23 °C. By zweryfikować te zmiany, badanie obejmowało regularne pomiary wartości rezystancji w jednakowych odstępach czasu.



Rys. 14 – Wykres wartości rezystancji w czasie dla próbki wyjętej z chłodni (pomarańczowa linia) oraz próbki przechowywanej od początku w temperaturze pokojowej (niebieska linia)

Jak widać na podstawie wyników (Rys. 14), próbka wyjęta z chłodni uzyskała stabilną wartość rezystancji 50  $\Omega$  po około 30 minutach. Tym samym wartość rezystancji była zbliżona do próbki przechowywanej cały czas w temperaturze pokojowej.

Warto zauważyć także, że próbka przechowywana w temperaturze pokojowej również wymagała oczekiwania na ustabilizowanie się wyniku pomiarowego. Czas ten jednak był krótszy i wyniósł około 10 minut. Wnioski z tego eksperymentu wskazują, że próbka powinna być przechowywana w chłodni dla zachowania jej struktury i parametrów. W przypadku konieczności wykonywania pomiarów, konieczne jest zachowanie okresu przynajmniej 30 minut przed przystąpieniem do badania, od wyjęcia materiału badawczego z chłodni.

**Wilgotność**, jest kolejnym parametrem wpływającym na zachowanie się tkanki podczas pomiarów. Parametr konduktywności jest związany z zawartością wody – im mniej wody, tym mniejsza przewodność materiału.



Rys. 15 - Wykres obrazujący proces wysychania próbki jako wartość rezystancji w czasie oraz w chwilę po nałożeniu elektro-konduktywnego żelu w 70. minucie

Badania wykazały, że wilgotność jest istotną kwestią w procedurze pomiarowej. Wpływa to na próbki znajdujące się w laboratorium, oczekujące na badania. Efekt wysychania nie jest bowiem redukowany nawet przykryciem materiału badawczego.

W wyniku eksperymentów wypracowane zostało jednak rozwiązanie, które zapewnia wyeliminowanie wspomnianego efektu praktycznie całkowicie. Realizuje to zastosowanie żelu elektroprzewodzącego. Już cienka warstwa na badanej próbce zapewnia przywrócenie pełni przewodności (Rys. 15).

Próby badawcze pokazały, że wartość rezystancji materiału pozostawionego bez przykrycia w warunkach laboratoryjnych rośnie w czasie nawet trzykrotnie. Badana próbka homogeniczna ciągu 70 minut ciągłego mierzenia parametru rezystancji w stałych odstępach czasowych wykazała zmianę wartość parametru rezystancji z początkowej 63  $\Omega$  do końcowej 250  $\Omega$ . Nałożenie cienkiej warstwy żelu elektroprzewodzącego w 70. minucie, praktycznie natychmiast przywróciło początkową wartość rezystancji próbki.



*Rys.* 16 – Histogram wartości konduktywności 81 próbek dla materiału homogenicznego [S/m]

**Wielkość materiału** jest również istotnym zagadnieniem, który realizuje kwestie związane z przygotowaniem próbek do badań. Przeprowadzone badania pokazały, że wielkość próbki w przypadku homogenicznej struktury jest niezależna pod względem parametru rezystywności (Rys. 17).



Rys. 17 - Wartość konduktywności materiału jednorodnego w relacji do wielkości próbki

Parametr konduktywności zmienia się w przedziale  $\pm 9\%$ . Potwierdza to, że próbkę taką można traktować jako jednorodną o dowolnej objętości i wymiarach geometrycznych. Każda próbka została zbadana w trzech osiach. Tym samym z jednego materiału możliwe było zebranie trzy razy więcej pomiarów parametrów materiałowych.

Próbki zostały zmierzone geometrycznie, a wartości wysokości (*l*) oraz powierzchni styku (*s*) podstawione do wzoru (12). W taki sposób zostały wyliczone wartości konduktywności. Na podstawie histogramu konduktywności próbek, wyznaczonego na podstawie 81 zebranych pomiarów, można podać dane statystyczne rozkładu funkcji gęstości prawdopodobieństwa.

Wartość średniej 0,530 [S/m] oraz odchylenie standardowe 0,059 [S/m] wskazuje na zmienność parametryczną na poziomie 11%, co potwierdza założenia z poprzedniego podrozdziału dotyczącej akwizycji danych i ustalonego błędu pomiarowego na poziomie 10%

### 2.4.3. Materiały niejednorodne

Głównym celem opisywanego etapu badań opracowanie parametrów tkanek żywych organizmów oraz wykorzystania tych danych do dalszych obliczeń symulacyjnych. Pomiar tkanek żywych jest jednak kwestią problematyczną. Z jednej strony pomiar "in-vivo" jest badaniem wprost na żywym organizmie, a tym samym realizowanie takich pomiarów podczas zabiegu jest utrudnione, ze względu na czas, czy temperaturę. Z drugiej każdy eksperyment, czy zabieg "in vivo" wymaga zgody komisji etyki<sup>9</sup> (Art.45.1).



Rys. 18 - Widok badanej tkanki mięsno-tłuszczowej

Do badań nad obiektami heterogenicznymi wykorzystana została tkanka mięsno-tłuszczowa (Rys. 18). Pomiary materiału heterogenicznego zostały opisane w poprzednim podrozdziale. Opierają się o taką samą metodologię, jak w przypadku tkanek homogenicznych.

<sup>&</sup>lt;sup>9</sup> Źródło: https://www.nil.org.pl/\_\_data/assets/pdf\_file/0003/4764/Kodeks-Etyki-Lekarskiej.pdf (dostęp: 2018.11.17)



Rys. 19 – Histogram wartości konduktywności dla próbek heterogenicznych o boku długości 2 cm (po lewej) i 1 cm (po prawej) [S/m]

Dla materiału jednorodnego, histogram wartości konduktywności kształtował się w sposób zbliżony, niezależnie od wielkości badanej próbki. Dla materiałów niejednorodnych rozmiar próbki pozwala zauważyć różnicę w tej kwestii. Widać wyraźnie (Rys. 19), że dla próbek większych jednorodność materiałowa jest zaburzona (po lewej). Próbki takie, wydzielone z materiału badawczego, nie zawierają tylko jednego typu tkanki. Tym samym zmierzone parametry elektryczne takich próbek nie są wartościami jednej danej tkanki, ale kilku tkanek w różnych proporcjach (po prawej).

Zmniejszając wielkość badanej próbki do boku o rozmiarze 1 cm, otrzymujemy materiały badawcze w większym stopniu jednorodne pod względem struktury tkankowej. Zawartość jednego typu tkanki w próbce tej wielkości jest bliska 100%. Tym samym próbka może zostać potraktowana jako homogeniczna.



Rys. 20 - Histogram wartości konduktywności próbki heterogenicznej dla dwóch tkanek (kolor czerwony i niebieski) oraz rozkład uśredniony [S/m]

W przypadku tej wielkości próbek, histogram konduktywności zaczyna się zmieniać (Rys. 20). Rozkład oparty o próbki bez uwzględniania ich jednorodności był rozkładem normalnym (Gaussowskim). Uwzględnienie rozmiaru próbek oraz zależności zawartości tkanek, przybliża rozkład wszystkich badanych próbek do mieszanego (Mixture).

Na podstawie wyników badań wyników badań wywnioskować można, że kolejne zmniejszenie wymiarów geometrycznych próbek dałoby bardziej widoczny efekt. Pozwoliłoby to na zobrazowanie rzeczywistego, proporcjonalnego rozkładu tkanek homogenicznych w złożonej próbce.



*Rys.* 21 - Wartość μ i σ konduktywności dla różnej wielkości próbek heterogenicznych [S/m]

Podsumowując – badania te pokazały, że różnica średniej i odchylenia standardowego dla konduktywności dla większych próbek, a próbek mniejszych wynosi nawet 23% (Rys. 21). Tym samym odpowiednie dobranie parametrów próbek pozwala na uniknięcie zafałszowania faktycznych parametrów tkanek. Jednocześnie pozwala to na otrzymanie rzeczywistych wyników pomiarowych i stworzenia bazy danych, która odzwierciedla rzeczywiste parametry tkanek.

## 2.5. Opracowanie modeli tkanek

Na podstawie przeprowadzonych badań, udało się ustalić różnorodność parametrów materiałów biologicznych oraz jak zmienne potrafią być w swojej zmienności. Dobrze opisują to przeprowadzone badania nad tkankami podczas operacji chirurgicznej na mózgu [44]. Zmienność tkanek jest nie tylko w samej populacji – wśród dziewięciu pacjentów, ale także i w samych tkankach tych pacjentów. Co ciekawe – zmierzone parametry (Tabela 3) potrafią być opisywane rozbieżnością rzędu połowy wartości badanej (średnia: 3,61  $\Omega$ m, odchylenie: 1,25).

Tym samym trudno jest przygotować modele tkanek, w których nawet jeden typ tkanki różni się od drugiej znacząco. Najczęściej parametrycznie takie tkanki są opisywane rozkładem Gaussowskim. Uśrednienie z wszystkich pomiarów, każdego z 9 pacjentów. Takie podejście z pewnością jest łatwiejsze, ale czy rzeczywiście odzwierciedla poprawność wyników. Już wspomniane w poprzednim rozdziale badania [18] pokazują, że wyniki podawane w ten sposób (zgodnie z rozkładem normalnym) potrafią przyjąć wartości ujemne, co jest nieprawidłowe.

Pojawia się kwestia, jak zatem zamodelować tkankę, której rozkład jest znacząco zmienny. Z pomocą przychodzi wspomniany rozkład logarytmiczny normalny (Log-normal), który jest przede wszystkim rozkładem dodatnim. Zastosowanie tego rozwiązania pozwala na dopasowanie także innych rozkładów statystycznych tkanek, dla których dane źródłowe nie są dostępne do danych pomiarowych spełniających założenia.

	<b>P-1</b>	<b>P-2</b>	P-3	<b>P-4</b>	P-5	<b>P-6</b>	<b>P-7</b>	<b>P-8</b>	P-9
Tkanka szara [S/m]	0,515	0,185	0,469	0,246	0,240	-	0,200	0,372	0,379
Tkanka biała [S/m]	0,277	0,216	_	0,307	_	0,210	_	0,302	_

Tabela 3 - Uśrednione wartości konduktywności tkanki szarej i białej dla 9 pacjentów [44]

Stosując rozkład normalny wartość średnia dla powyższych parametrów wraz z odchyleniem standardowym byłaby równa:

- tkanka szara:
  - średnia 0,325 [S/m],
  - $\circ$  odchylenie standardowe 0,126.
- tkanka biała:
  - średnia 0,262 [S/m],
  - $\circ$  odchylenie standardowe 0,047.



Rys. 22 – Rozkład gęstości prawdopodobieństwa wartości konduktywności dla tkanki szarej (po lewej) oraz tkanki białej (po prawej) [S/m]

Na podstawie przedstawionych histogramów (Rys. 22) widać jak wygląda rozkład parametrów tkanek dwóch typów na podstawie zebranych danych pomiarowych (Tabela 3). Warto zauważyć tutaj kwestię poruszaną wcześniej – dla tkanki szarej (po lewej) wartości rozkładu trafiają w wartości ujemne. Jest to sytuacja niedopuszczalna w przypadku modelowania tkanek, ponieważ wartości ujemnie konduktywności nie mają biologicznego uzasadnienia.

Dla takich wartości zastosować można zupełnie inny rozkład – logarytmiczny normalny. Należy dla niego wyliczyć dwie wartości –  $\mu$  oraz  $\sigma$ , z:

$$\mu = \log \frac{m^2}{\sqrt{\nu + m^2}} \tag{16}$$

$$\sigma = \sqrt{\log \frac{\nu}{m^2} + 1} \tag{17}$$

gdzie: v oraz m – wartości średniej oraz odchylenia standardowego dla rozkładu normalnego.

W przypadku takiego przełożenia rozkładu jednego w drugi, mogą nastąpić pewne rozbieżności, dlatego skuteczniejszym jest wyliczenie tych wartości ze zbioru wartości parametrów, które są dostępne (Tabela 3).

Takie wyliczenie wymaga zastosowania operacji estymacji parametrów dla rozkładu logarytmicznego normalnego.

W środowisku MATLAB polecenie realizujące to założenie nazywa się *lognfit*. Pozwala ono dla zbioru danych wartości wyliczyć parametry realizujące rozkład Log-normal w określonym przedziale z określonym prawdopodobieństwem oraz pewnością. Wyniki są następujące:

- tkanka szara:  $\mu = -1,189, \sigma = 0,393,$
- tkanka biała:  $\mu = -1,351, \sigma = 0,183$ .

Dla tych parametrów zostały wygenerowane histogramy w rozkładzie logarytmicznym normalnym, które obrazują różnicę w rozwiązaniu tego samego zagadnienia parametryzacji.



Rys. 23 - Porównanie rozkładów normalnego i logarytmicznego normalnego wartości konduktywności dla tkanki szarej i tkanki białej (a – normalny, tkanka szara, b – logarytmiczny normalny, tkanka szara, c – normalny, tkanka biała, d – logarytmiczny normalny, tkanka biała) [S/m]

Na podstawie porównania (Rys. 23), zarówno rozkład normalny, jak i logarytmiczny normalny wpasowują się w wartości parametrów tkanek względem średniej oraz odchylenia standardowego. Różnicą podstawową między nimi, jest brak wartości ujemnych w przypadku rozkładu logarytmicznego normalnego, a te wartości stanowią poważny problem w przypadku zamodelowania tkanek heterogenicznych.

W podobny sposób zostały przygotowane rozkłady siedmiu tkanek, które mogą posłużyć do symulacji podczas obliczeń na platformie rozproszonej (Tabela 4).

Tkanka	Rozkład I	Normalny	Rozkład I	ognormal
	μ	σ	μ	σ
Skóra	0,369	0,033	-0,999	0,088
Kość	0,181	0,095	-1,851	0,642
Mięsień	0,422	0,325	-1,115	0,751
Tkanka szara	0,326	0,126	-1,189	0,393
Tkanka biała	0,262	0,047	-1,351	0,183
CSF	1,535	0,060	0,428	0,040
Krew	0,587	0,093	-0,545	0,163

Tabela 4 - Wartości μ oraz σ rozkładów normalnego oraz logarytmicznego normalnego parametru konduktywności dla siemiu wybranych tkanek [S/m]

# ROZDZIAŁ 3. PLATFORMY OBLICZEŃ ROZPROSZONYCH

Pojęcie obliczeń rozproszonych nie jest nowe i jego początki sięgają jeszcze sieci ARPANET<sup>10</sup>. Złożona struktura połączonych ze sobą urządzeń w logicznej sieci stanowiła pierwowzór dzisiejszych platform przeznaczonych do obliczeń rozproszonych. Wprawdzie było to fizyczne połączenie urządzeń oddalonych od siebie o setki tysięcy kilometrów na terenie Stanów Zjednoczonych, jednak idea współdzielenia między sobą zasobów i działającej równolegle aplikacji poczty email zapoczątkowała równoległe obliczenia.

Przez lata określenie to, z języka angielskiego zwane "distributed systems" rozwinęło się także w "rozproszone oprogramowanie", czyli aplikacje działające równocześnie na wielu zasobach współpracujących ze sobą. Tak narodziły się systemy takie, jak: Grid czy BOINC.

Dość ważną kwestią jest różnica w nazewnictwie między systemami rozproszonymi, a systemami równoległymi. Mimo iż określenie to jest stosowane często zamiennie, wskazać należy pewne różnice. O ile na przykład systemy rozproszone mogą pracować w zrównolegleniu, o tyle jednostki obliczeniowe (CPU) w takich systemach pracują tylko równolegle. Inaczej mówiąc systemy rozproszone to luźno powiązane ze sobą systemy równoległe, a systemy równoległe są dość precyzyjnie związanymi systemami rozproszonymi. Aby móc powiedzieć, który system jest rozproszony, a który równoległy, przyjęło się stosować dwie ważne definicje<sup>11</sup>:

- **systemy równoległe** korzystają ze współdzielonej pamięci, z której korzysta każdy CPU, a zadania są rozdzielane wewnątrz jednego systemu operacyjnego,
- **systemy rozproszone** korzystają z własnej pamięci, wymieniają się komunikatami o zasobach i zadaniach, które są rozdzielane przez system centralny.

<sup>&</sup>lt;sup>10</sup> Źródło: http://scihi.org/arpanet-became-internet/ (dostęp: 2018.10.15)

<sup>&</sup>lt;sup>11</sup> Źródło: https://www.distributed-systems.net/index.php/books/ (dostęp: 2018.10.15)

Systemy rozproszone nadal są dynamicznie rozwijane, ponieważ z roku na rok rośnie ilość wymagań stawiana takim systemom. Wzrost wydajności obliczeniowej komputerów sprawił, że rozwinęły się technologie wirtualizacyjne<sup>12</sup>, zwiększające możliwości tych systemów.

Obecnie najpopularniejszą formą systemów rozproszonych jest chmura obliczeniowa. Od momentu wzrostu zainteresowania tą tematyką od początku 2008 roku<sup>13</sup> do kulminacyjnego momentu popularności na początku 2011, wartość tego zagadnienia nieustannie utrzymuje się w czołówce technologicznych zagadnień. Ujednolicenie systemów rozproszonych pozwoliło na połączenie ich w jeden centralnie zarządzany system. Jednocześnie ułatwiło skalowanie rosnących zasobów, udostępnianie ich, ale także na skomercjalizowanie. Oferowane jako usługi z przeznaczeniem do różnych zastosowań.

W kolejnych podrozdziałach zostaną zaprezentowane poszczególne typy architektury systemów rozproszonych wraz z technologicznym wyjaśnieniem ich funkcjonowania.

## 3.1. Obliczenia rozproszone

Obliczenia rozproszone (ang. distributed computing), to określenie, które związane jest z infrastrukturami komputerowymi. Złożone często z wielu niejednolitych systemów połączonych ze sobą przez sieć teleinformatyczną. Celem ich jest dzielenie się zbiorem danych i zadań obliczeniowych w celu skrócenia czasu wykonywania obliczeń. Zdarza się, że są one realizowane we wspólnym celu, lecz nie jest to wymagane, by nazwać systemy rozproszonymi.

Wśród rozwiązań, które można wydzielić jako typy połączeń, ale także i platformy zastosowań, należą Grid (w tym World Community Grid<sup>14</sup> wspierane przez IBM), BOINC, ale także Parabon NanoLabs<sup>15</sup>.

#### 3.1.1. Grid

Podstawowym pojęciem związanym z obliczeniami rozproszonymi jest Grid. Polskie określenie tego słowa to "siatka" i najtrafniej określa, czym taki system jest oraz w jaki sposób realizowane są w nim zadania. Historycznie pojęcie Grid pojawiło się w 1998 roku [45] wraz z pierwszą książką poruszającą tematykę infrastruktury komputerowej [46] [47].

<sup>&</sup>lt;sup>12</sup> Źródło: http://www.kernelthread.com/publications/virtualization/ (dostęp: 2018.10.15)

<sup>&</sup>lt;sup>13</sup> Źródło: https://trends.google.com/trends/explore?date=all&q=cloud computing,distributed computing (dostęp: 2019.02.25)

<sup>&</sup>lt;sup>14</sup> Źródło: https://www.worldcommunitygrid.org/discover.action (dostęp: 2018.11.17)

<sup>&</sup>lt;sup>15</sup> Źródło: https://parabon-nanolabs.com/ (dostęp: 2018.11.17)

Z pojęciem tym wiąże się także inne, bardziej rozległe – grid computing. Są to obliczeniowe systemy rozproszone, które wymieniają się danymi we wspólnym celu obliczeniowym. Jest to element uszczegółowiający spośród typowej sieci rozproszonej.

Głównym założeniem sieci Grid, jest zapewnienie niezależności od platformy, lokalizacji, czy metody łączności między jednostkami obliczeniowymi. Tym samym nie jest to typowy klaster obliczeniowy [48]. Sieć Grid jest bardziej heterogeniczna pod względem struktury w stosunku do rozwiązań typowego HPCC<sup>16</sup>. Co ważne, Grid jest systemem skupiającym w swojej strukturze systemy nie tylko o sporej wydajności, ale o dowolnych możliwościach obliczeniowych. Warunkiem koniecznym jest poprawna komunikacja z siecią Grid. Przewagą systemu są niewielkie wymagania, a aprobata związana z dużym rozproszeniem pozwala na stworzenie sieci określanej mianem wirtualnego superkomputera (Rys. 24).



Rys. 24 - Systemy rozproszone w skali i zastosowaniach [48]

Grid jest odporny na awaryjność. Awaryjność może być spowodowana problemami sieciowymi lub sprzętowymi. Rozwiązanie jest odporne także na odłączenie się systemów strukturalnych od sieci. Jednostka obliczeniowa (węzeł) może w każdej chwili zostać odłączona, co nie wpłynie na dalsze obliczenia, stabilność, czy funkcjonowanie sieci. Również spójność danych nie zostanie zakłócona.

Ponadto warto wskazać, że Grid jest swoistą infrastrukturą pozwalającą na dołączenie się do sieci urządzeń, które nie są w pełni przeznaczone do obliczeń. Węzły mogą być współdzielone lub wykorzystywane do innych zadań. Przykładem jest wykorzystanie mocy obliczeniowej

<sup>&</sup>lt;sup>16</sup> Źródło: https://hpccsystems.com/ (dostęp: 2018.11.17)

procesora, gdy węzeł komputera pracującego w sieci Grid nie jest urządzeniem dedykowanym. Komputer prywatnego użytkownika pracujący w nieregularnych odstępach czasu lub używany równocześnie w innych zadaniach nie wpływa na funkcjonowanie sieci Grid. Podobne możliwości są w kwestii wykorzystania części mocy procesora. Sieć korzysta z zasobów obliczeniowych w przypadku wykrytej nieaktywności podczas pozostałego czasu pracy.

Jednocześnie dzisiejsze komputery to nie tylko procesor, czy pamięć operacyjna, ale także jednostka graficzna z (ko)procesorami pozwalającymi na przyspieszenie obliczeń. Tym samym oprogramowanie zapewniające komunikację z siecią Grid jest w stanie wykorzystać dostępne podzespoły komputera według potrzeb, monitorując ich użycie i zapewniając ciągłość pracy.

#### 3.1.2. BOINC

Wśród sieci opartych o rozwiązanie Grid, jednym z najpopularniejszych i najbardziej wydajnych<sup>17</sup> wirtualnych superkomputerów jest projekt BOINC<sup>18</sup>. Projekt ten jest obecnie najbardziej rozbudowanym i największym przykładem wykorzystania założeń sieci Grid w celach obliczeniowych (Rys. 25).

Źródłem projektu jest oprogramowanie pozwalające na łączność wielu jednostek obliczeniowych będących częścią innych podprojektów, które są częścią sieci BOINC. Infrastruktura jest na tyle rozproszona, że użytkownik samemu określa się, jaką częścią sieci będzie jego komputer.

Oprogramowanie służy do łączności z serwerem zasobów obliczeniowych (listą projektów), do monitorowania każdej jednostki obliczeniowej, w tym czasu poświęcanego dziennie na obliczenia, faktycznej możliwości obliczeniowej oraz przydzielania danych obliczeniowych zgodnie z wybranymi projektami, w których jednostka przesyła żądanie wzięcia udziału.

Projekt oraz uczestnictwo są dostępne i aktywne w ramach licencji wolnego i otwartego oprogramowania GNU/LGPL, które w dodatku jest aktywnie wspierane przez NSF<sup>19</sup>.

Projekt BOINC jest z góry nastawiony na wolontariat w celach naukowych, ponieważ nie wymaga inwestycji w ogromne zasoby obliczeniowe, ale dzięki uczestnictwu każdej z osób

<sup>&</sup>lt;sup>17</sup> Źródło: https://boincstats.com/en/stats/-1/project/detail (dostęp: 2018.08.04)

<sup>&</sup>lt;sup>18</sup> Źródło: https://boinc.berkeley.edu/ (dostęp: 2018.08.04)

<sup>&</sup>lt;sup>19</sup> Źródło: https://www.nsf.gov/ (dostęp: 2018.08.04)

posiadającej zainstalowane oprogramowanie projektu, pozwala na przeniesienie kosztów obliczeń oraz czasu na poszczególne, pojedyncze jednostki obliczeniowe.



Rys. 25 - Zarys struktury sieci BOINC [49]

Obliczenia również są realizowane przy pomocy intuicyjnego systemu przydzielania rozproszonego podprojektu. Oprogramowanie zapewnia jednolitość środowiska obliczeniowego, zapewniając np. aplikację wykonującą dane obliczenia dla wskazanej porcji danych. Ponadto projekt BOINC przesyła dane obliczeniowe do wielu różnych jednostek obliczeniowych, co zapewnia w pewnym sensie redundancję danych, ponieważ wyniki są uzyskiwane z kilku źródeł i wzajemnie porównywane, co ogranicza możliwy błąd, bądź braki w obliczeniach pewnych zbiorów danych.

Istotną zaletą jest wykorzystanie używanych zasobów w zgodzie z technologią, to znaczy jeśli komputer jest wielowątkowy, można wykonywać wiele obliczeń jednocześnie na każdym z rdzeni CPU. Równocześnie obliczenia wykorzystują zasoby z poszanowaniem systemu Grid, czyli korzystają z wolnych zasobów, nie ograniczając użycia jednostki dobrowolnie podłączonej pod projekt w ramach wolontariatu.

Korzyścią dla uczestnika projektu BOINC są punkty, które są przyznawane proporcjonalnie do liczby czasu obliczeń poświęconego na dane projekty. Punkty zbierane są swoistą nagrodą pozwalającą na rywalizację z innymi uczestnikami i branie udział w lokalnych i globalnych rankingach.

## 3.2. Chmura obliczeniowa

Obliczenia rozproszone, jak wspomniano w poprzednim rozdziale pokazują, czym tak naprawdę jest idea rozdzielenia zadań na wielu jednostkach obliczeniowych. Nie tylko w celu

skrócenia czasu obliczeń, ale także w celu wydajniejszej komunikacji. W przypadku systemów rozproszonych, często znajdujących się w różnych miejscach, czy regionach świata jest kluczowe.

Ważnym momentem stało się zdefiniowanie pojęcia, jako "Cloud computing", po polsku chmury obliczeniowej. Dość istotne jest tutaj ujednolicenie pojęcia chmury prywatnej oraz publicznej<sup>20</sup>. Definicje te są często traktowane wymiennie, co nie jest prawidłowe. Chmura prywatna, to rozproszony system obliczeniowy wewnątrz osobistej infrastruktury fizycznej. Chmura publiczna to system dzierżawiony, którego właścicielem w żaden sposób nie jest użytkownik.

Niejednokrotnie zasoby zapewniane przez dostawcę usług, jako gwarantowane (CPU, RAM, HDD), definiowane są jako chmura prywatna. Taka definicja jest również nieprawidłowa. Rozwiązanie oparte o wirtualizację z udostępnieniem wybranych zasobów nie zapewnia pełnego zarządzania nimi. To jest podstawowa różnica, którą otwarcie definiuje każdy z trzech kluczowych dostawców rozwiązań chmury na świecie – Microsoft<sup>21</sup>, Amazon<sup>22</sup> oraz IBM<sup>23</sup>.

Rozwiązanie dziś zwane chmurą obliczeniową zaczęło się od wprowadzenia wielowątkowości systemów informatycznych i współdzielenia zasobów sprzętowych na lokalnych urządzeniach. Możliwość wykonywania wielu zadań jednocześnie pozwoliły na rozwinięcie modelu tradycyjnego sieci Grid. Rozwój ogromnych firm IT, takich jak Microsoft, Google czy Amazon, dały szansę na stworzenie ogromnych w skali rozwiązań złożonych z setek, tysięcy, czy milionów połączonych ze sobą komputerów.

W dodatku często serwerownie dostawców chmury obliczeniowej złożone są z jednolitych klastrów, czy całych sieci, co pozwala na wyeliminowanie problemu różnorodności wydajności. Nakłady finansowe takich firm pozwoliły na rozwinięcie infrastruktury oraz zbudowanie centrów danych, które dziś przynoszą także ogromne zyski. Dzięki wspomnianej jednolitej infrastrukturze, zasobom dostępnym dla każdego oraz konkurencyjności (zapewniającej niskie koszty zakupu usług), rozwiązanie stało się ważnym i rozwojowym przedsięwzięciem informatycznym.

<sup>&</sup>lt;sup>20</sup> Źródło: https://www.expedient.com/blog/private-vs-public-cloud-whats-difference/ (dostęp: 2018.11.17)

<sup>&</sup>lt;sup>21</sup> Źródło: https://azure.microsoft.com/pl-pl/overview/what-are-private-public-hybrid-clouds/ (dostęp: 2018.11.17)

<sup>&</sup>lt;sup>22</sup> Źródło: https://aws.amazon.com/types-of-cloud-computing/ (dostęp: 2018.11.17)

<sup>&</sup>lt;sup>23</sup> Źródło: https://www.ibm.com/cloud/learn/what-is-private-cloud (dostęp: 2018.11.17)

#### 3.2.1. Wirtualizacja

Wprowadzenie wielowątkowych procesorów pozwoliło nie tylko na przyspieszenie wykonywania obliczeń, ale także na zrównoleglenie zadań poprzez wykorzystanie ich zasobów w tym samym czasie, niezależnie od siebie .



Rys. 26 - Podział metod wirtualizacji w zależności od infrastruktury [50]

Wyróżnić można trzy podstawowe typy wirtualizacji (Rys. 26):

Typ-1 (tzw. nadzorczy), który polega na uruchomieniu tzw. Hypervisora/VMM, który rozdziela wszystkie zasoby sprzętowe dla uruchamianych maszyn (gości). Przykładem jest Vmware ESXi<sup>24</sup>, Hyper-V Server<sup>25</sup>.

Typ-2 (tzw. parawirtualizacja), czyli VMM uruchomiony na fizycznie zainstalowanym systemie-gospodarzu. Typ-2 jest wirtualizacją programową, ponieważ VMM nie ma bezpośredniego dostępu do sprzętu, zasoby są przydzielane przez system nadrzędny (Host OS).
Przykładem jest np. Docker<sup>26</sup>, Sandboxie<sup>27</sup>.

Hybrydowy, czyli połączenie obu powyższych systemów. Takie rozwiązanie jest wtedy, gdy
VMM instaluje w systemie własne sterowniki sprzętowe i korzysta z własnych warunków
przypisywania procesora, pamięci, czy kart rozszerzeń, niezależnie od systemu-gospodarza.
Przykładem jest MS Virtual PC, jako aplikacja instalowana w systemie operacyjnym.

Znaną wśród użytkowników domowych wirtualizacją jest Typ-1 lub Hybrydowy (przy wykorzystaniu narzędzi dedykowanych, jak np. VMware<sup>28</sup> Workstation). W systemie

<sup>&</sup>lt;sup>24</sup> Źródło: https://www.vmware.com/products/esxi-and-esx.html (dostęp: 2018.11.17)

<sup>&</sup>lt;sup>25</sup> Źródło: https://docs.microsoft.com/en-us/windows-server/virtualization/hyper-v/hyper-v-server-2016 (dostęp: 2018.11.17)

<sup>&</sup>lt;sup>26</sup> Źródło: https://www.docker.com/ (dostęp: 2018.11.17)

<sup>&</sup>lt;sup>27</sup> Źródło: https://www.sandboxie.com/ (dostęp: 2018.11.17)

<sup>&</sup>lt;sup>28</sup> Źródło: https://www.vmware.com/ (dostęp: 2018.11.17)

zainstalowanym na fizycznym sprzęcie uruchamiane jest oprogramowanie pozwalające na zainstalowanie jednego lub więcej takich lub innych systemów operacyjnych, z czego każdy działa w osobnych zasobach, odizolowanych od siebie z wydzielonymi zasobami sprzętowymi.

W przypadku rozwiązań w rozbudowanych systemach rozproszonych, takie rozwiązanie nie zdaje efektu, ponieważ konieczne jest zapanowanie nad wszystkimi urządzeniami, zasobami i ich sprawne przypisywanie w zależności od wymagań i potrzeb.

Dla takich celów powstało rozwiązanie określone mianem wirtualizacji Typ-2, która całkowicie odchodzi od systemu gospodarza. Oprogramowanie VMM jest nadrzędnym systemem, który sprawuje kontrolę nad każdym elementem sprzętowym i z całej puli zasobów pozwala uruchamianym nad nim maszynom na ich wykorzystywanie w pełnym zakresie adresowym.

## 3.2.2. Dostępne rozwiązania chmurowe

Na rynku dostępnych jest wiele rozwiązań od wielu dostawców. Swoje usługi oferuje Amazon z AWS (ang. Amazon Web Services)<sup>29</sup>, Google z GCP (ang. Google Cloud Platform)<sup>30</sup>, IBM z IBM Cloud<sup>31</sup>, Alibaba z Alibaba Cloud<sup>32</sup>, czy Oracle z Oracle Cloud<sup>33</sup>.

W rzeczywistości, pomijając kwestie konkretnej firmy oferującej dostęp do zasobów swojej serwerowni, każda z usług jest dostępna z mniejszą lub większą funkcjonalnością u każdego dostawcy. Różnice są tak naprawdę tylko w zakresie architektury samej chmury, co dla użytkownika nie ma większego znaczenia oraz tego jaki dostęp (interfejs) dostępu do usług jest zapewniony.

Każdy z dostawców oferuje różne poziomy usług, różny podział i hierarchię struktur, według których pewne zasoby są dostępne wraz z innymi, a które wymagają większych lub mniejszych konfiguracji. W przypadku Microsoft Azure, rozwiązania używanego w dalszej części pracy, podział usług jest następujący [51]:

- Usługi infrastrukturalne (ang. Infrastructure Services)
- Obliczeniowe (ang. Compute)

<sup>&</sup>lt;sup>29</sup> Źródło: https://aws.amazon.com/ (dostęp: 2018.11.17)

<sup>&</sup>lt;sup>30</sup> Źródło: https://cloud.google.com/ (dostęp: 2018.11.17)

<sup>&</sup>lt;sup>31</sup> Źródło: https://www.ibm.com/cloud/ (dostęp: 2018.11.17)

<sup>&</sup>lt;sup>32</sup> Źródło: https://www.alibabacloud.com/ (dostęp: 2018.11.17)

<sup>&</sup>lt;sup>33</sup> Źródło: https://cloud.oracle.com/home (dostęp: 2018.11.17)

- Przechowywanie danych (ang. Storage)
- Infrastruktura sieciowa (ang. Networking)
- Usługi platformowe (ang. Platform Services)
  - Multimedia oraz CDN (ang. Media and Content Delivery Network)
  - Platforma aplikacyjna (ang. Application Platform)
- Elementy bazodanowe (ang. Databases)
- Elementy integracyjne (ang. Integration)
- Usługi obliczeniowe (ang. Compute Services)
- Usługi projektowe (ang. Developer Services)
  - Elementy informacyjne (ang. Intelligence)
  - Elementy analityczne oraz IoT (ang. Analytics and Internet of Things)
- Usługi związane z bezpieczeństwem oraz zarządzaniem (ang. Security and Management)

PRODUCT	aws	Microsoft Azure	Google Cloud Platform
Virtual Servers	Instances	VMs	VM Instances
Platform-as-a-Service	Elastic Beanstalk	Cloud Services	App Engine
Serverless Computing	Lambda	Azure Functions	Cloud Functions
Docker Management	ECS	Container Service	Container Engine
Kubernetes Management	EKS	Kubernetes Service	Kubernetes Engine
Object Storage	S3	Block Blob	Cloud Storage
Archive Storage	Glacier	Archive Storage	Coldline
File Storage	EFS	Azure Files	ZFS / Avere
Global Content Delivery	CloudFront	Delivery Network	Cloud CDN
Managed Data Warehouse	Redshift	SQL Warehouse	Big Query

• Usługi związane z chmurą hybrydową (ang. Hybrid Cloud)

Rys. 27 - Porównanie nazw usług w AWS, Azure oraz GCP [52]



Rys. 28 - Globalny podział infrastrukturalny według Amazon [53], [54]

Powyższy podział jest w przypadku jednego dostawcy. Analogiczne rozwiązania u wspomnianych wyżej firm konkurencyjnych, są także dostępne, pod innymi nazwami (Rys. 27) lub inaczej zgrupowane (Rys. 28).

Z punktu technologicznego, każde z rozwiązań ma swoje plusy i minusy. Rozwiązania nie są ze sobą całkowicie kompatybilne, w przypadku konieczności przeniesienia realizowanego projektu. Również skalowalność i dostępność są różnie definiowane i zapewniane na różnym poziomie. Nie ma uniwersalnego rozwiązania, ale jednocześnie dzięki konkurencyjności, każde z rozwiązań staje się w regularnych odstępach czasu coraz tańsze.

W przypadku prowadzonych badań naukowych, wymagania technologiczne, czy możliwości skorzystania z infrastruktury chmurowej do wykonywania obliczeń oraz eksperymentów naukowych według własnych schematów są tutaj istotne. Kluczowe staje się tutaj umożliwienie optymalnego wykorzystania zasobów infrastrukturalnych w celu przyspieszenia wykonywania zadań związanych z nauką bez konieczności ponoszenia sporych kosztów własnych, czy zakupu ogromnej infrastruktury, która w dodatku musi być łatwo modyfikowana pod nowe potrzeby, skalowana w zależności od intensyfikacji obliczeń, ale i uniwersalna.

Podkreślić tutaj należy, że do badań została wykorzystana infrastruktura Microsoft Azure, której zasoby są dostępne dzięki umowie Uczelni z firmą Microsoft i w ramach grantu zostały przyznane wirtualne środki przeznaczone na dowolne usługi świadczone przez firmę wewnątrz infrastruktury Azure.

Szczegółowy opis architektury Microsoft Azure został przedstawiony w Załączniku 1.

# 3.3. Wykorzystane technologie

Celem niniejszej pracy było wykorzystanie obliczeń rozproszonych do modelowania tkanek. O ile modelowanie tkanek zostało już wyjaśnione w poprzednim rozdziale, wraz z opisem i metodologią, o tyle nadal zostaje kwestia tego, w jaki sposób obliczenia symulacyjne rozdystrybuować na wiele komputerów.

Niniejszy rozdział opisuje w jaki sposób realizowane jest wykonywanie obliczeń rozproszonych oraz działanie usług w tych systemach. Przechodząc historycznie od rozwiązań prototypowych (sieć ARPANET), przez rozwiązania sieci Grid, aż do rozwiązania jakim jest chmura obliczeniowa.

To właśnie chmura stała się dla nas tym z najbardziej dostępnych rozwiązań i to na niej postanowione zostało zbudowanie skalowalnej platformy obliczeniowej.

Modele symulacyjne tkanek wyliczane są dla określonych danych wejściowych zwanych parametrami symulacyjnymi. Cała symulacja przeprowadzana jest z użyciem oprogramowania FEniCS [55], jako oprogramowanie dostępne na bezpłatnej licencji Open Source do wyliczania równań różniczkowych cząstkowych (PDE).

Celem jest jednak wykonywanie wielu obliczeń jednocześnie na wielu węzłach (jednostkach obliczeniowych). Konieczne jest przygotowanie nie tylko rozwiązania w postaci pliku z obliczeniami do wykonania, ale także listy parametrów obliczeniowych, jako danych wejściowych. Ważnym etapem jest wdrożenie systemu rozproszonego pozwalającego na sprawną dystrybucję zadania symulacyjnego.

Spośród rozwiązań do dystrybucji zadań pomiędzy komputerami w sieci rozproszonej, swoją funkcjonalność podczas testów wykazały Apache Hadoop<sup>34</sup> oraz HTCondor<sup>35</sup>. Oba rozwiązania różnią się jednak schematem działania i dedykowanym przeznaczeniem. Hadoop jest pakietem narzędzi skierowanym na przetwarzanie danych BigData, czyli zbioru nieustrukturyzowanych danych. HTCondor jest rozwiązaniem rozwijanym przede wszystkim akademicko, którego zadaniem jest przesłanie instrukcji do węzła i odebrania wyniku.

<sup>34</sup> Źródło: https://hadoop.apache.com/ (dostęp: 2018.07.25)

<sup>&</sup>lt;sup>35</sup> Źródło: http://research.cs.wisc.edu/htcondor/ (dostęp: 2018.07.25)

#### 3.3.1. Apache Hadoop

Apache Hadoop jako rozwiązanie samo w sobie nie jest młode, gdyż pierwsze wzmianki pojawiły się już w 2003 roku [56] w dokumentacji Google, której celem było stworzenie, cytując: "skalowalnego systemu plików rozproszonych dla wielkoskalowych aplikacji intensywnie przetwarzających dane". Sam algorytm przetwarzania narodził się blisko rok później w 2004 roku [57], gdy został zaprezentowany ogólny zarys rozwiązania zwanego MapReduce.

W obecnej postaci (Rys. 29) w uproszczeniu graficznym przedstawia się następująco: dane wejściowe jako nieuporządkowany zbiór (Input) są dzielone najpierw na *n* części mniejszych rozmiarów (Split), łatwiejszych nie tyle do przetworzenia, co szybkiej dystrybucji i przesłania na węzły obliczeniowe. Później są one przekazywane dostępnym węzłom (Map) gdzie odpowiednie algorytmy wykonawcze dokonują ich przetworzenia (posortowania, wyszukania konkretnych danych, uporządkowania, itd.). Każde przetworzene dane są wykonywane na podzielonych "paczkach danych" (InputData\_X), a przetworzenie ich nie wymusza na algorytmie wykonywania ich po kolei, a tym samym nie ma problemu tzw. "wąskiego gardła" (ang. bottle neck), w oczekiwaniu na dane. Później te przetworzone według schematu paczki (Shuffle&Sort) są zbierane przez wolne węzły redukujące (Reducer), których zadaniem jest ostateczne połączenie danych i przesłanie do zbioru danych wyjściowych (Output) na etapie łączenia (Merge).



Rys. 29 - Schemat algorytmu MapReduce [58]

Ideą rozwiązania jest automatyczne przyspieszenie zadania poprzez wykonywanie go w wielu wystąpieniach równocześnie, a tym samym dane nie są wykonywane po kolei, a rozdzielenie

pozwala na uzyskanie dużej wydajności w porównaniu do przetwarzania jednego ogromnego zbioru danych.



Rys. 30 - Algorytm MapReduce według architektury YARN [59]

Ten algorytm wydał się rozwiązaniem problemu rozdzielenia znacznej liczby przypadków danych wejściowych, które mogłyby być wykonywane równolegle, na węzły obliczeniowe. Przygotowanie środowiska obliczeniowego wymagało skonfigurowania Apache Hadoop w strukturze architektury (Rys. 30) zwanej YARN.

Powodem wykorzystania tego rozwiązania architekturalnego był fakt, że jest ono obecnie najbardziej rozwinięte pod względem wykorzystania, pozwala na interaktywne przetwarzanie danych, strumieniowanie oraz obsługę i przetwarzanie z użyciem HDFS.

Sprawdzenie skuteczności rozwiązania dla badań nad modelowaniem tkanek wymagało przeprowadzenia eksperymentu numerycznego, który pokazałby skuteczność założeń z wykorzystaniem platformy Hadoop. Przetestowanie wydajności wymagało nie tylko przygotowania środowiska obliczeniowego (z wykorzystaniem oprogramowania FEniCS), ale także samego klastra obliczeniowego. Każda z maszyn obliczeniowych wymagała także zainstalowania oprogramowania FEniCS na węzłach roboczych, ponieważ wykonywane obliczenia numeryczne uruchamiały oprogramowanie niezależnie, dla każdych obliczeń (Rys. 31).



Rys. 31 - Schemat obliczeniowy z wykorzystaniem Apache Hadoop [60]

W przypadku naszych obliczeń eksperymentalnych jedynym skutecznym rozwiązaniem do dystrybucji zbioru danych wejściowych (Input) było wykorzystanie wprost Mappera do przydzielania zadań i uruchamianie na węzłach obliczeniowych oprogramowania symulacyjnego, wykonując bezpośrednie przerzucenie wyników symulacyjnych przy węzłach pracujących jako Reducer, przekazując wyniki na wyjście (Output).

### 3.3.2. HTCondor

W poszukiwaniach odpowiedniego rozwiązania udało się znaleźć platformę HTCondor. Od początku wydawała się spełniać wszystkie wymagania niezbędne do uruchomienia pełnoprawnego środowiska. Z jednej strony łatwego do skalowania, z drugiej pozwalającego na większą możliwość kontroli uruchamianego oprogramowania podczas jego dystrybucji na węzłach.

Rozwiązanie powstało w 1988 roku jako Condor (w 2012 zmieniono nazwę na HTCondor)<sup>36</sup> w celu dystrybucji zadań w oparciu o środowiska HTC. Ideą było zrównoleglenie zadań na wiele jednostek obliczeniowych. Kluczowe było optymalne wykorzystanie dostępnych zasobów obliczeniowych. Podstawową różnicą w stosunku do Apache Hadoop jest ukierunkowanie na komunikowanie się z węzłami roboczymi (Workerami) i przesyłanie zadań do wykonania. W uproszczeniu idea opiera się o przekazanie poleceń oraz powiązanych z nimi danych wejściowych oraz odebranie danych wyjściowych.

<sup>&</sup>lt;sup>36</sup> Źródło: https://lists.cs.wisc.edu/archive/condor-users/2012-October/msg00110.shtml (dostęp:2018.07.25)

esource Group	H	Condor
	► Negotiator	StartD
Central Manager	Collector	Schedd
VMSS		
	StartD	
VM 1 (Worker)	Schedd	
11.	StartD	
VM 2 (Worker)	Schedd	

*Rys.* 32 - Architektura systemu HTCondor uruchomionej na platformie Microsoft Azure [61]

Z racji możliwości personalizacji środowiska oraz struktury działania, zdecydowaliśmy się na wykorzystanie rozwiązania opartego o chmurę obliczeniową, Microsoft Azure (Rys. 32). Podstawowe założenia struktury to grupa zasobów, wewnątrz której znajdować się będą dwie usługi – pierwsza to Manager, jako maszyna wirtualna, która będzie dystrybuować zadania, ale także będzie zarządzać węzłami roboczymi. Manager to maszyna pozwalająca na zarządzanie innymi. Konieczne było tym samym przygotowanie maszyn wirtualnych, które uruchomione w dowolnej liczbie, o dowolnych parametrach zawsze będą komunikować się poprawnie z Menadżerem oraz będą jednorodne środowiskowo.

Worker to również maszyna wirtualna, która w przeciwieństwie do Managera, powinna być skonfigurowana tak, aby miała niezbędne oprogramowanie do przeprowadzania symulacji, poprawną konfigurację komunikacyjną w sieci oraz była responsywna na zadania przekazywane przez Managera.

Dla poprawnej konfiguracji został przygotowany w sieci Manager, dla którego przypisano stały adres IP. Uruchomione zostały na nim usługi: weryfikująca dostępne węzły robocze (ang. Negotiator), odbierająca oraz wysyłająca zadania do/z kolejki (ang. Collector), uruchamiająca zadania oraz pozwalająca na ich uruchomienie i wstrzymanie (ang. StartD) oraz wysyłająca zadania według harmonogramu i kolejkowania (ang. Schedd). Dodatkowo na potrzeby weryfikacji działania każdego scenariusza, na maszynie zostało zainstalowane środowisko FEniCS pozwalające sprawdzić poprawność działania symulacji przekazywanej dalej za pośrednictwem HTCondor.

Konfigurowanie wszystkich maszyn od nowa, gdy zajdzie potrzeba dodania kolejnych mija się z celem skalowalności, którą zapewnia środowisko chmury. Jednocześnie utrzymywanie gotowych do obliczeń kilkuset, czy kilku tysięcy maszyn również jest sprzeczne z założeniami uproszczenia.

Przygotowane zostało dlatego środowisko na jednej maszynie, która została wdrożona jako węzeł roboczy (Worker) z pracującym na niej systemem GNU/Linux z dystrybucji Ubuntu. Zapewniało to jednorodność obu środowisk oraz poprawność działania skryptów pisanych w środowisku linuksowym.

### 3.4. Testy technologiczne systemów dystrybucji zadań

Wspomniane w poprzednim podrozdziale rozwiązania zostały przetestowane pod kątem poprawności działania, ale także funkcjonalności w systemach rozproszonych. Każdy system otrzymał przygotowane zadanie modelowe, które pozwoliło sprawdzić wspomnianą wydajność oraz skalowalność rozwiązań.

Platforma Apache Hadoop została skonfigurowana na klastrze złożonym z czterech fizycznych jednostek, z których każda była wyposażona w dwa czterordzeniowe procesory Intel Xeon E6520 taktowane 1.6 GHz połączone z dyskami SSD o pojemności 256GB. Cała pamięć dostępna dla Apache Hadoop wynosiła 24GB. Węzły były połączone przez internet połączeniem o szybkości 1 Gb/s, na których pracował GNU/Linux w wersji 64-bitowej. Każdy węzeł miał zainstalowany oprogramowanie Hadoop w wersji 2.6.0.

Zadanie modelowe na standardowym komputerze nowej generacji obliczeniowo trwało około 4 sekund. Łączna liczba przypadków do wyliczenia to 10 201. Szacunkowy czas obliczeń dla wspomnianego komputera to 40 804 sekund, czyli 11 godzin 20 minut.



Rys. 33 - Optymalizacja czasu obliczeń w zależności od liczby rdzeni [60]

Jeden z węzłów stanowił administrację klastrami, zatem do dyspozycji obliczeniowej dostępne były 22 rdzenie CPU. Hadoop został skonfigurowany tak, aby korzystać ze wszystkich dostępnych rdzeni. Wyniki eksperymentu pokazują, że najbardziej optymalne rozwiązanie następuje w przypadku wykorzystania 21 rdzeni procesora. Tym samym wywnioskować można, że dwa rdzenie są niezbędne, aby zapewnić stabilność platformy, możliwość zarządzania środowiskiem Hadoop oraz procesami dystrybuującymi dane wejściowe na węzły obliczeniowe.



Rys. 34 - Czas obliczeń pojedynczego PC oraz klastra Hadoop względem od liczby przypadków [60]

Wydajność platformy jest istotna, ponieważ bez względu na ilość przypadków (wielkość parametru Input) powinna ona zapewniać jednakowy przyrost zysku z czasu obliczeń w stosunku do pojedynczego rozwiązania wykonywanego na jednej jednostce obliczeniowej.

Z wykresu porównawczego ilość przypadków do czasu obliczeń dla pojedynczego komputera PC oraz klastra Hadoop wykorzystującego jeden albo wszystkie (optymalnie) 21 rdzeni (Rys. 33) widać, że ta korzyść jest znacząca. Dla największej liczby przypadków (1066) czas pojedynczej symulacji dla PC wynosi blisko 4300 sekund (Rys. 34). Ten sam przypadek dla klastra Hadoop wyniósł zaledwie 275 sekund [60]. Jest to przyrost rzędu 15 razy.

$$narzut = \frac{czas_{oblicze\acute{n}} * liczba_{węzłów}}{czas_{1zadania} * liczba_{przypadkow}} - 1$$
(18)

gdzie: *czas<sub>obliczeń</sub>* – czas całkowitych obliczeń dla platformy obliczeniowej, *liczba<sub>węzłów</sub>* – liczba węzłów obliczeniowych platformy, *czas<sub>1zadania</sub>* – czas obliczeń pojedynczego przypadku, *liczba<sub>przypadków</sub>* – liczba wykonywanych niezależnych symulacji obliczeń. W teorii w przypadku użycia wszystkich dostępnych 22 rdzeni, spodziewany przyrost maksymalny to 22 razy. Jednak przyrost zmierzony i eksperymentalnie udowodniony jest mniejszy. Spowodowane jest to wieloma czynnikami. Dla każdej platformy obliczeniowej jest to tzw. współczynnik narzutu (18). Jest to zbiór wszystkich czynników, które są wykonywane przed i poza faktycznymi obliczeniami – dodanie zadania do kolejki, odebranie go przez węzeł obliczeniowy, przygotowanie skryptu uruchamiającego oprogramowanie symulacyjne, czas uruchomienia samego programu, a po zakończeniu obliczeń, zapisanie wyniku, przesłanie go do Menadżera zadań i odebranie potwierdzenia. Nie są to znaczne ilości czasu, ale wpływają na wartość narzutu.



Rys. 35 - Narzut procentowy obliczeń w stosunku do rozmiaru problemu [60]

Jak widać po wykresie (Rys. 35) ukazującym wielkość współczynnika narzutu względem liczby przypadków obliczeń (aż do maksymalnych wspomnianych wyżej: 10 201) widać, że korzyść z wykorzystania platformy Hadoop rośnie wraz ze wzrostem liczby przypadków, czyli tak naprawdę – w czasie. Wartość procentowa narzutu stabilizuje się na poziomie około 20% już przy 4 000 przypadków, co potwierdza przyjęte założenia.

Eksperyment został zrealizowany na ponad 10 000 przypadkach, jednak już teraz można przewidzieć, że nawet w sytuacji 1 000 000 scenariuszy obliczeniowych, wartość narzutu będzie większa niż 10%. Jest pewien poziom współczynnika, którego nie da się zminimalizować w prosty sposób, ponieważ wpływa na niego wiele czynników wspomnianych wyżej.

Rozwiązanie to jednak mimo iż dość atrakcyjne i popularne, nie stanowi dobrego rozwiązania na potrzeby obliczeń i symulacji w zastosowaniu modelowania tkanek. Platforma nie
wykorzystuje pełni możliwości, które posiada (pominięcie etapu Reducer), jest przeznaczona do przetwarzania danych, a nie zarządzania węzłami obliczeniowymi, co w przypadku badań byłoby kluczowe.

Platforma HTCondor również została zweryfikowana testem technologicznym, także w celu zweryfikowania wartości narzutu (Rys. 36).



Rys. 36 - Narzut obliczeniowy w stosunku do liczby przeprowadzonych symulacji

Początkowe sprawdzenie wydajności platformy (dla symulacji identycznej, jaka była wykonywana dla środowiska Apache Hadoop) okazało się, że wartość narzutu utrzymuje się między 20-30%. Zgodnie z wynikami poprzednich testów wydajności dla Apache Hadoop, również i dla HTCondor okazało się, że w przypadku małej liczby symulacji, wartość narzutu będzie znacznie większa niż dla większej ich liczby.

Eksperyment obejmował symulację w problemie optymalizacyjnym dla przypadku magnetostatycznego. Cewka złożona z pięciu toroidalnych pierścieni (cylindryczny układ współrzędnych) dla których położenie mogło być zmieniane tylko w jednej płaszczyźnie. Celem było uzyskanie jak największej wartości energii wpływającej na obiekt.

Na podstawie przeprowadzonego eksperymentu udało się sprawdzić nie tylko skuteczność rozwiązania chmurowego, czyli obliczeń rozproszonych w wielkoskalowym środowisku, skuteczność działania środowiska HTCondor w eksperymentach modelowania z wykorzystaniem oprogramowania FEniCS, które posłuży także kolejnym eksperymentom oraz badaniom, ale przede wszystkim dowieść przyspieszenia obliczeń oraz skalowalności rozwiązania w zależności od potrzeb.

Klaster VMSS zbudowany w Microsoft Azure został przygotowany z 40 maszyn, każda z 8 rdzeniami CPU, co łącznie dało 320 węzłów roboczych (Workerów). Teoretyczne przyspieszenie w takiej sytuacji powinno wynieść 320 razy. Pojedyncza symulacja zajmuje w przybliżeniu 5 sekund. Dla 100 000 przypadków zająć to powinno 500 000 sekund (czyli 139 godzin). HTCondor w powyższej konfiguracji rozdystrybuował zadanie i wykonał je w czasie około 2 godzin, co pozwala wywnioskować, że narzut sięgał w przybliżeniu 150%.

Szczegółowa instrukcja przygotowania i konfiguracji środowiska obliczeniowej chmury publicznej opartej o Microsoft Azure została zawarta w Załączniku 2.

Eksperyment sprawdził także jak wielkość narzutu zmienia się w przypadku długości trwania zadań (Rys. 37). Zauważyć można, że dla krótkich zadań wartość narzutu jest duża i sięga nawet ponad 600% – spowodowane jest to krótkim czasem wykonywania samej symulacji, natomiast czas narzutu jest relatywnie zbliżony bez względu na czas wykonywania obliczeń. Dlatego też w przypadku bardziej złożonych obliczeń, wartość narzutu maleje do poziomu ok. 30%, co jednocześnie pokrywa się z wartością narzutu ogólnego. Mimo to nie osiągnie poziomu niższego niż 20-30%.



Rys. 37 - Wartość narzutu w funkcji czasu wykonywania pojedynczego zadania

Podsumowując kwestie zastosowanego środowiska obliczeniowego – chmura obliczeniowa jest wydajnym, skalowalnym i istotnie funkcjonalnym rozwiązaniem. Przygotowanie środowiska nie stanowi większego problemu ze względu na wiele poradników i scenariuszy, które prowadzą krok po kroku w zależności od potrzebnego zastosowania, a wszystko dostępne jest w sieci internet. Skalowalność zapewniona przez dostawcę usług chmurowych pozwala na uruchomienie praktycznie dowolnej liczby maszyn w kilka-kilkanaście minut.

Najważniejszym zyskiem i korzyścią ze stosowania chmury obliczeniowej jest koszt, który stawia rozwiązanie na pierwszym miejscu w porównaniu do rozwiązań "on-premise", czyli we własnych zasobach czy w miejscu pracy. Koszt konfiguracji środowiska wraz z przeprowadzeniem symulacji 100 000 przypadków dla 40 instancji wyniósł 40 \* 0,049 EUR/h \* 4 godziny pracy czyli 7,84 EUR.

## ROZDZIAŁ 4. PRZYKŁADY OBLICZENIOWE

Poznanie teorii dotyczącej modelowania stochastycznego tkanek oraz poznanie zagadnienia obliczeń rozproszonych z wykorzystaniem technologii chmury obliczeniowej pozwoliło przygotować działające, skalowalne i przenośne środowisko do badań. Pozwala ono na szybkie uruchomienie dużej liczby maszyn wirtualnych, które współpracują ze sobą, w celu zrównoleglenia obliczeń rozproszonych.

Rozdział ten zawiera podsumowanie teorii analizy stochastycznej modelowania tkanek opisanej w rozdziale drugim, a także przygotowania praktycznego rozwiązania w postaci działającej platformy obliczeń rozproszonych opisanej w rozdziale trzecim. Szczegółowy opis chmury obliczeniowej znajduje się w Załączniku 1. Szczegółowa konfiguracja platformy obliczeń rozproszonych w oparciu o Microsoft Azure znajduje się w Załączniku 2.

W niniejszym rozdziale zostaną poruszone trzy zagadnienia, które pokazują w jaki sposób można wykorzystać możliwości chmury obliczeniowej do rozwiązywania problemów inżynieryjno-technicznych, ale także i medycznych. Wszystkie te zagadnienia dotyczą problemów wymagających analizy statystycznej. Dla przedstawienia możliwości oraz wydajności przygotowanej platformy obliczeniowej, opracowane zostały trzy tematy. Każdy z nich jest przede wszystkim praktycznym przypadkiem inżynieryjnym.

## 4.1. Grzanie rezystancyjne

Jednym z przypadków związanych z przepływem ciepła, a tym samym wartości energii w polu przepływowym, jest grzanie rezystancyjne, opisane Prawem Joule'a. Jest to z jednej strony dobry przykład złożonego modelowania i symulacji, a z drugiej stanowi podstawę do praktycznego zweryfikowania skuteczności rozwiązania opartego o obliczenia rozproszone. Przepływ prądu przez przewodnik powoduje wytworzenie się (indukowanie) ciepła w tym przewodniku. Równanie opisujące ten proces jest zwane Prawem Joule'a-Lenz'a, bądź Pierwszym Prawem Joule'a i opisywane jest wzorem:

$$\frac{Q}{t} = RI^2 \tag{19}$$

gdzie: Q – ilość wydzielonego ciepła w Joule'ach  $[J = \frac{kg * m^2}{s^2}]$ , t – czas przepływu prądu [s], R – opór elektryczny [ $\Omega$ ], I – prąd elektryczny [A]

Z prawa tego wynika wprost zasada zachowania energii, która mówi, że straty energii przepływającego prądu zamienia się w energię wewnętrzną.

Prawo, a właściwie zjawisko związane z opisywanym przez nie prawem jest wykorzystywane w przemyśle żywieniowym, w przetwarzaniu pożywienia (ang. food processing), polegającym na wykorzystaniu procesu grzania przewodnika z prądem jako źródła energii stanowiącej o podgrzewaniu żywności [43].

	Zalety		Wady
1.	Wymagana temperatura osiągana	1.	Mała ilość informacji ogólnych
	szybko		o systemie
2.	Równomierne i szybkie ogrzewanie	2.	Wymagana kalibracja systemu
	substancji płynnych		w zależności od zastosowanego
3.	Zmniejszenie liczby zanieczyszczeń		przewodnika oraz płynu grzewczego
	powierzchniowych	3.	Wąskie pasmo częstotliwości
4.	Brak efektu przenoszenia ciepła po	4.	Trudne monitorowanie procesu oraz
	wyłączeniu źródła prądu		kontrola grzania
5.	Niskie koszty utrzymania rozwiązania	5.	Skomplikowane powiązania zależności
	(mała ilość części ulegających zużyciu		temperatury i pola elektrycznego
	i ruchomych) z zachowaniem		w dystrybucji energii
	wysokiego poziomu energii		
6.	Natychmiastowe wyłączanie systemu		
7.	Cichy i przyjazny środowisku system		

Tabela 5 - Podsumowanie zalet i wad grzania rezystancyjnego w przemyśle żywieniowym [62]

Wykorzystanie zjawiska podgrzewania przewodzonego jest zagadnieniem złożonym z powodu konieczności powiązania temperatury i dystrybucji gęstości mocy w obiekcie grzanym. Ze względu na jednorodną strukturę jest to znacznie łatwiejsze w cieczach niż w przypadku struktury materiału będącego mieszaniną, która wymaga bardziej skomplikowanego modelu

geometrycznego. Spowodowane jest to nie tylko złożoną strukturą przewodności cieplnej obiektów, ale i zmiennymi parametrami materiałowymi (Tabela 5).

Postępując wobec Prawa Ohma:

$$U = RI \tag{20}$$

gdzie: U – napięcie doprowadzone do układu [V], R – rezystancja przewodnika [Ω], I – wartość prądu płynąca przez przewodnik [A]

oraz wzoru na moc:

$$P = UI \tag{21}$$

gdzie: P – moc prądu elektrycznego [W]

otrzymamy równanie opisujące moc jako kwadrat prądu oraz napięcia przewodnika:

$$P = RI^2 \tag{22}$$

Uwzględniając zmianę w czasie, a dokładniej przepływ mocy P w danym okresie czasu t, otrzymamy równanie (19), które pokazuje nam zależność ilości wydzielonego ciepła jako pracy wykonanej przez prąd zależną od współczynnika t.

Tym samym powiedzieć można, że oczekujemy uzyskania przy stałej wartości napięcia źródłowego jak największej wartości energii oraz tym samym mocy w układzie przy jak najmniejszej wartości prądu (efektywność energetyczna układu). Im większa wartość energii, tym większa praca wykonana, a w efekcie – więcej ciepła dystrybuowanego jest do obiektu, a w analizowanym przypadku, do żywności.

Problemem staje się poprawne zamodelowanie przypadku, który uwzględniałby dystrybucję ciepła, był uniwersalny w implementacji w przypadku bardziej lub mniej złożonych obiektów, które są podgrzewane oraz który zapewnia uwzględnienie zakresów zmienności wielu parametrów w strukturze.

#### 4.1.1. Symulowany obiekt

Założenia symulacyjne przeniosły się już na samym początku na model prostego jednorodnego (homogenicznego) obiektu, który może być podgrzewany, a którego przetwarzanie ma zastosowanie praktyczne.



Rys. 38 - Ziemniaki w kostkach dla których modelowany został przypadek podgrzewania rezystancyjnego

Prowadzone w poprzednich latach [21] [38] [42] [63] badania oraz artykuły nad obiektami biologicznymi i ich strukturą pokazały, że początkowe badania nad tkankami powinny być wykonywane na modelu homogenicznym. Ostatnio prowadzone badania nad ziemniakami są dobrym przykładem obiektu, który w dodatku jest popularny i szeroko stosowany w gastronomii.

Średnia [S/m]	Lilly	Red Lady	Irga	Sunny	Daisy	Madeira	Gala	Lord	
0,0222	0,0227	0,0260	0,0232	0,0185	0,0189	0,0223	0,0235	0,0225	
0,0043	0,0049	0,0054	0,0032	0,0019	0,0043	0,0055	0,0044	0,0046	
Odchylenie od średniej w procentach									
19,14%	22%	21%	14%	10%	23%	24%	19%	21%	

Tabela 6 - Porównanie wartości średniej oraz odchylenia standardowego konduktywności kostek gatunków ziemniaków (źródło własne)<sup>37</sup>

W prowadzonych badaniach nad parametrami elektrycznymi bulw ziemniaczanych, wyniki jednoznacznie wskazują, że różnice konduktywności wprawdzie zróżnicowane, nadal bez względu na gatunek są do siebie zbliżone, a zbadane próbki kostek (V =  $1 \text{ cm}^3$ ) (Rys. 38) wykazały dość sporą jednorodność obiektów w ramach badanych próbek homogenicznych (Tabela 6).

<sup>&</sup>lt;sup>37</sup> Opis pomiarów w rozdziale 2.3.2

### 4.1.2. Metodyka badań

Dla modelowanego przypadku oraz procesu grzania rezystancyjnego, założeniem było przygotowanie prostego przykładu, który pozwoliłby na zbadanie skuteczności algorytmu oraz zweryfikowanie poziomu złożoności przez zakres zmienności parametrów wewnątrz modelu.

Przygotowanym modelem symulacyjnym stał się model dwuwymiarowy pojemnika o przekroju kwadratu, w którym znajdują się nagrzewane obiekty biologiczne. Były to pokrojone bulwy ziemniaczane. Całość wypełniona jest cieczą będącą przewodnikiem. Do obu okładek pojemnika przykładane jest napięcie, a przepływający prąd propaguje się wewnątrz układu powodując emisję energii w ośrodku (Rys. 39).



Rys. 39 - Schemat przepływu obiektów w polu grzejnym

Przygotowanie modelu symulacyjnego wymagało określenia pewnych założeń projektowych. Ustawione jako zmienne zostały następujące parametry (Tabela 7):

- parametry cieczy w ośrodku konduktywność,
- parametry materiału biologicznego wielkość kostki, kąt obrotu, położenie, konduktywność,
- parametry ogólne w tym ilość kostek, czy rozdzielczość obrazu siatki.



Rys. 40 - Rozkład wylosowanych "kostek" (po lewej) oraz rozkład gęstości energii w polu przepływowym badanego pojemnika (po prawej)

Na podstawie powyższych informacji zauważyć można, że tak przygotowany model symulacyjny posiada wiele elementów zmienności, a tym samym przygotowanie jednego modelu symulacji nie pozwoli na sprawdzenie skuteczności działania grzania rezystancyjnego. Jednocześnie zmienianie wielu parametrów w tym samym czasie w określonym zakresie daje obraz na proces wynikowy.

Parametr	Zakres zmienności
Konduktywność środowiska [S/m]	10
Wielkość kostki materiału [m]	$0,005 \pm 0,001$
Kąt obrotu [°]	0 – 90
Położenie kostki materiału [(x,y), m]	0,005 - 0,095
Konduktywność kostki materiału (Tabela 6)	$0,022 \pm 0,004$
Liczba kostek	Do 100
Rozdzielczość obrazu siatki	30
Napięcie zasilania [V]	10

Tabela 7 - Parametry oraz zakres zmienności symulowanego modelu

Celem badań podczas symulacji wspomnianego modelu było wygenerowanie jak najbardziej jednorodnego rozkładu energii w obszarze, zapewniając jednocześnie jak najmniejszą wartość napięcia i prądu dostarczoną do obiektu.

Jednocześnie warto dodać, że generowanie obiektów jest losowe, ale uwzględniony został parametr kolizji, który sprawdza, czy wygenerowany obiekt nie zawiera się w żadnym z istniejących obiektów.

Zaimplementowany został algorytm sprawdzający kolizję obiektów [64]. Opisana reguła, po polsku zwana "parzysty-nieparzysty", sprawdza otoczenie każdego nowo wygenerowanego punktu w przestrzeni. Punkt jest środkiem czworokąta foremnego, dla którego generowane są cztery punkty będące jego wierzchołkami. Algorytm w całym badanym obszarze sprawdza współrzędne środka oraz każdego wierzchołka, czy nie znajdują się w obszarze zamkniętym już istniejących czworokątów. Jeżeli żaden z punktów nie spełnia tego warunku, utworzony obiekt zostaje uznany za bezkolizyjny.

W badanym przypadku jeżeli nastąpiła kolizja, określony był warunek powtórzeń generowania nowego punktu, aby ilość obiektów w badanej przestrzeni była znacząca. Limit prób kolizji w przypadku niepowodzenia został ustalony na 500, a maksymalna liczba kostek na 100. Zapewnia to, że generowanie obiektów kończy się, gdy jeden z dwóch warunków zostanie spełniony jako pierwszy. Takie rozwiązanie zapewnia pokrycie przestrzeni obiektami, a jednocześnie pozwala uniknąć sytuacji, w której proces samego generowania mógłby trwać w nieskończoność.

### 4.1.3. Wyniki i wnioski

Wynik jednej z wielu wykonanych symulacji, którą opisano powyżej, to siatka obrazu pola (Rys. 40), która ilustruje liczbę obiektów jakie wygenerowano wraz z ich położeniem oraz widocznym obróceniem losowym (po lewej), ale i rozkład gęstości mocy w polu przepływowym (po prawej).

Jak widać, siatka reprezentuje niejednorodne pole w obszarze z nielicznymi, relatywnie niewielkimi punktowymi obszarami zmian wartości pola elektrycznego. Związane jest to z lokalnym, losowym układem kostek.

Oczywistym jest, że zadanie stworzenia modelu statystycznego wymaga przygotowania wielu przypadków symulacyjnych, zbadania ich wartości mocy oraz rezystancji. W tym właśnie celu zostanie wykorzystane środowisko obliczeń rozproszonych opisane w Rozdziale 3. Przygotowane środowisko obliczeniowe zostało wykorzystane do wykonania 10 000 symulacji.

W badanym modelu trzy parametry to zmienne, które pozwalają na przygotowanie dużej liczby niezależnych modeli symulacyjnych.



Rys. 41 - Histogram wartości mocy [W] (po lewej) i rezystancji [S/m](po prawej) w obiekcie dla 10 000 symulacji

Na powyższych histogramach (Rys. 41) wykres maksymalnej wartości mocy jest uśrednionym modelem całego obiektu w jego strukturze oraz maksymalnej wartości rezystancji dla zasymulowanego przypadku modelu.

Na podstawie przeprowadzonych symulacji udało się pokazać, że modelowanie problemu grzania rezystancyjnego obiektów biologicznych jednorodnych, jest skomplikowane obliczeniowo. Uwzględnienie zmienności statystycznej parametrów materiałowych także komplikuje obliczenia.

Wygenerowanie dużej liczby zmiennych dla zadanego rozkładu oraz przeprowadzenie symulacji jest zadaniem wymagającym mocy obliczeniowej. Można jednak skompensować moc obliczeniową i rozłożyć na wiele maszyn o mniejszej wydajności, lecz w większej liczbie.

Badanie to pokazało, jak w przystępny sposób można zmaksymalizować wykorzystanie zasobów chmury obliczeniowej, redukując czas obliczeń. Czas pojedynczej symulacji to około 30 sekund. Dla 10 000 symulacji to ponad 5 000 minut, czyli 83 godziny. W przypadku wykorzystania chmury obliczeniowej dla 100 węzłów, czas obliczeń to 60 minut, narzut to około 20%.Koszt obliczeń dla 100 maszyn w użytej konfiguracji to \$0.09/h, co daje kwotę w wysokości \$9.

## 4.2. Elektrostymulacja nerwowo-mięśniowa

Kolejnym przykładem praktycznego zastosowania obliczeń rozproszonych jest przypadek elektrostymulacji nerwowo-mięśniowej, czyli EMS (ang. Electrical Muscle Stimulation) [65].

Sam przypadek jest dobrze znany od wielu lat i stosowany przede wszystkim w sporcie oraz w przypadku osób, u których reakcja mięśni jest ograniczona lub mało wydajna. Stymulacja mięśni przez krótką chwilę powoduje skurcze, które pobudzają odpowiednie nerwy w zależności od konkretnego wysiłku, jaki jest wykonywany.

Takie zagadnienie jest bardziej skomplikowane niż analizowane wcześniej grzanie rezystancyjne, ponieważ obejmuje kilka typów tkanek, a tym samym pokazuje modelowanie przypadków bardziej natury medycznej, a dokładniej bioelektromagnetycznej.

#### 4.2.1. Symulowany obiekt

Złożoność tego zagadnienia jest zależna tylko od skomplikowania modelu, ale zrozumienie istoty samego przypadku jest ważne. Rozpatrywany model składa się z pięciu tkanek – skóry zewnętrznej, kości ramienia, obszaru żylnego znajdującego się tuż przy kości oraz mięśnia, który stanowi całą pozostałą objętość modelu (Rys. 42).



Rys. 42 – Przekrój [66] (po lewej) oraz model przekroju ramienia użyty w symulacji (po prawej)

Model został skomplikowany przez zmienną geometrię parametrów fizycznych. Zmiana dotyczy proporcji kształtu przekroju ramienia, średnicy kości, czy grubości skóry. Współczynnik zmienności tych parametrów to ±10%. Parametry tkanek również zostały określone jako zmienne. Taka modyfikacja pozwala na uwzględnienie różnic natury biologicznej – wieku pacjenta, środowiska, czy wzrostu.

### 4.2.2. Metodyka badań

Na podstawie źródeł danych WHO<sup>38</sup>, ale także z literatury [67] [68] [69] udało się uzyskać statystyczne dane pomiarowe parametrów struktury ramienia, które posłużyły za parametry wejściowe modelu. Przyjęte wartości wraz ze zmiennością:

Parametr	Zakres zmienności
Promień ramienia [cm]	4 – 5
Zmienność rozmiaru promienia ramienia [%]	10
Grubość skóry ramienia [cm]	0,2-0,4
Promień obszaru żył w osi X [cm]	0,6 - 0,8
Promień obszaru żył w osi Y [cm]	$0,\!4-0,\!5$
Rozmiar elektrody [cm]	1
Napięcie na elektrodzie [V]	150

Tabela 8 - Parametry oraz zakres zmienności symulowanego modelu

Dla tak zbudowanego modelu symulacyjnego zostały określone i wygenerowane parametry tkanek (Tabela 8) w dwóch znanych rozkładach stochastycznych – normalnym oraz logarytmicznym normalnym. Oba rozkłady dla każdej tkanki zostały wygenerowane 100 000, a każdy tak sparametryzowany przypadek został obliczony na platformie badawczej.

## 4.2.3. Wyniki i wnioski

Po przeprowadzeniu symulacji dla każdego z przypadków zauważyć można, że histogramy wartości parametrów tkanek kształtują się prawidłowo według założeń stochastycznych, czyli wyliczone wcześniej wartości  $\mu$  oraz  $\sigma$  dla obu typu rozkładów są prawidłowe (Rys. 43 oraz Rys. 44).

<sup>&</sup>lt;sup>38</sup> Źródło: https://www.who.int/childgrowth/standards/ac\_for\_age/en/ (dostęp: 2019.02.10)



Rys. 43 - Histogramy wartości konduktywności zamodelowanych tkanek w rozkładzie normalnym dla 100 000 symulacji [S/m]



Rys. 44 - Histogramy wartości konduktywności zamodelowanych tkanek w rozkładzie log-normal dla 100 000 symulacji [S/m]

Kluczowym staje się jednak histogram wyliczonej wartości mocy oraz średniej wartości pola elektrycznego w mięśniu, jako tkance (Rys. 45 oraz Rys. 46).



Rys. 45 - Histogramy wartości mocy [W] oraz pola elektrycznego [V/m] rozkładu normalnego dla 100 000 symulacji



Rys. 46 - Histogramy mocy [W] oraz pola elektrycznego [V/m] rozkładu log-normal dla 100 000 symulacji

Zarówno w przypadku zastosowania parametryzacji tkanek w rozkładzie normalnym, jak i logarytmicznym normalnym, wykresy energii pola elektrycznego oraz średniej wartości mocy w modelu bliższe są swoim kształtem rozkładowi logarytmicznemu normalnemu. Oznacza to, że dobranie parametrów materiałowych ze zmiennością określoną dowolnym z rozkładów statystycznych, w wyniku symulacji wartości wyliczone mogą rozkładać się w zupełnie innej dziedzinie parametrycznej. Wyliczone dwa parametry elektryczne pokazały, że tym rozkładem zbliżonym do założonego jako naturalny jest logarytmiczny normalny.

Obliczenia były wykonywane na platformie obliczeniowej z wykorzystaniem 200 maszyn obliczeniowych. Przy czasie symulacji na poziomie 4 sekund, obliczenia w przypadku pojedynczego komputera klasy PC powinny zająć ok. 800 000 sekund, czyli 222 godziny. Czas obliczeń z wykorzystaniem opracowanej platformy obliczeniowej to 4413 sekund, czyli 74 minuty. Wyliczony narzut w przypadku tego zadania to 10,31%. Całkowity koszt obliczeń wyniósł \$0.14/h<sup>39</sup> \* 200, czyli ok. \$35.

## 4.3. Terapia elektrowstrząsowa

Trzecim analizowanym zagadnieniem jest modelowanie terapii elektrowstrząsowej (ang. ECT – electroconvulsive therapy) w ludzkiej głowie. Ideą tego zagadnienia jest pobudzenie obszarów mózgu przy pomocy pola elektromagnetycznego wytworzonego przez elektrody umieszczone na głowie pacjenta.

Terapia ma szerokie zastosowanie w psychologii [70] [71]. Metoda polega na przepływie krótkotrwałego impulsu prądu elektrycznego w głowie pacjenta wytworzonego przez elektrody umieszczone na zdefiniowanych miejscach czaszki. Wpływ mocy wytwarzanej w odpowiednich obszarach mózgu w dalszej perspektywie ma wpłynąć na wyleczenie

<sup>&</sup>lt;sup>39</sup> Źródło: https://azure.microsoft.com/en-us/pricing/ (dostęp: 2017.06.13)

pacjenta. Terapia stosowana jest w zaburzeniach psychicznych, stanach lękowych oraz depresjach połączonych z myślami samobójczymi.

#### 4.3.1. Symulowany obiekt

Wykorzystany został trójwymiarowy, realistyczny model głowy dostępny w ramach pakietu BrainSuite<sup>40</sup>, który jest zestawem narzędzi przeznaczonych do analizy w oparciu o obrazy rezonansu magnetycznego (MRI). Gotowy model głowy został w niezmienionej formie wykorzystany w analizowanym przypadku – posiada on 5 zaznaczonych obszarów tkanek, które można wyróżnić (Rys. 47).



Rys. 47 - Model trójwymiarowy (po lewej) oraz przekrój (po prawej)z zaznaczonymi tkankami (1 – skóra, 2 – kość czaszki, 3 – CSF, 4 – tkanka szara, 5 – tkanka biała)

#### 4.3.2. Metodyka badań

Zamodelowanie zabiegu ECT, wymaga zdefiniowania przylegających do skroni elektrod (Rys. 48). Rozmiar elektrod został ustalony na 2 cm. Wartość napięcia źródła to 100 V [72].

<sup>&</sup>lt;sup>40</sup> Źródło: http://brainsuite.org/ (dostęp: 2018.11.19)



*Rys.* 48 - *Symboliczne zobrazowanie umiejscowienia elektrod na badanym modelu (po lewej) oraz miejsc pomiaru wartości energii pola elektrycznego w trzech regionach (po prawej)* 

Model, podobnie jak w podrozdziale 2.3.1, został opisany przy pomocy równania Laplace'a z warunkami brzegowymi Dirichleta (6), a następnie (7). Parametry materiałowe tkanek zostały określone na podstawie zagadnień z poprzednich podrozdziałów (Tabela 4).

Na podstawie poprzednio uzyskanych wyników, model tego zagadnienia został opisany tkankami sparametryzowanymi w rozkładzie logarytmicznym normalnym.

#### 4.3.3. Wyniki i wnioski

Model został zaprojektowany tak, aby uwzględnić parametry stałe dla ECT oraz parametry tkanek, według zmienności w przedziale (Rys. 49). Symulacja została przeprowadzona na platformie obliczeniowej z wykorzystaniem 200 maszyn wirtualnych liczących numerycznie 10 000 jednocześnie wygenerowanych przypadków.

W wyniku przeprowadzonych symulacji zostały zebrane parametry wejściowe modelu oraz wyjściowe dla każdego przypadku z osobna. Dodatkowo został przygotowany histogram wartości mocy całkowitej w modelu.



Rys. 49 – Rozkłady gęstości prawdopodobieństwa wartości konduktywności wygenerowanych parametrów tkanek [S/m]

Jak widać, model jest dość złożony, chociaż nadal uproszczony w swojej budowie. Zmienność parametrów kilku materiałów oraz struktura posiadająca rozbudowaną geometrię kształtu sprawiły, że wyznaczenie rozkładu pola jest dość czasochłonne. Założony rozkład prawdopodobieństwa tkanek określony jako logarytmiczny normalny potwierdził, że odpowiednie dobranie parametrów wejściowych ma duże znaczenie w badaniu ECT. Ich odpowiednie określenie pozwoliło na określenie skuteczności metody ECT w przypadku pacjenta, bez narażania fizycznej osoby na działanie terapii.



*Rys.* 50 - Rozkład gęstości prawdopodobieństwa dla średniej mocy [W] modelu oraz wartości natężenia pola elektrycznego [V/m] w trzech punktach wewnątrz głowy pacjenta (Rys. 48)

Rozkład mocy (Rys. 50) w przypadku zdefiniowanych tkanek o zadanym rozkładzie pokazał, że bliższy jest rozkładowi logarytmicznemu normalnemu. Podobnie w przypadku wartości natężenia pola elektrycznego w trzech obszarach głowy modelu pacjenta.



Rys. 51 - Wyznaczone dwie ścieżki pomiarowe przez przekrój głowy pacjenta

Dla badanego rozkładu gęstości pola elektrycznego  $E_m$  dodatkowo zostały wykreślone linie pomiarowe w przekroju między czaszkowym (Rys. 51). Zaznaczone kolorem żółtym linie zostały podzielone na 100 jednakowo odległych od siebie punktów w przestrzeni znajdującej się wewnątrz modelu. Dla każdego z tych punktów wyznaczone zostały wartości pola elektrycznego. Przedstawiony na (Rys. 52) zbiór wartości pokazuje rozkład  $E_m$  w przekroju.



Rys. 52 - Wykresy rozkładu pola elektrycznego w osi poprzecznej (linia A-A') oraz w osi wzdłużnej (linia B-B') według linii z Rys. 51 [V/m]

Wykresy przedstawione na wizualizują przestrzeń możliwych rozwiązań. Szczegółowa merytoryczna analiza tych wyników wymaga wiedzy medycznej, co przekracza zakres tej pracy. Można jednak sformułować dwa podstawowe wnioski:

- największej zmienności natężenia pola elektrycznego należy się spodziewać w warstwach zewnętrznych, takich jak skóra i czaszka,
- zmienność pola elektrycznego w wybranych fragmentach mózgu sięga 50% (300-650 V/m)

Dodatkowym wyzwaniem był sposób prezentacji zmienności rozwiązania. Histogramy są dobrym narzędziem dla pojedynczych wartości liczbowych (takich jak całkowita moc w mózgu lub moduł pola elektrycznego w punkcie). Jednak zilustrowanie zmienności pola trójwymiarowego jest poważnym wyzwaniem, które nie ma dobrego rozwiązania. Pewnym uproszczeniem są jednowymiarowe analizy przy pomocy natężenia kolorów (Rys. 52). Pozwalają one zobrazować charakter zmienności pola na wybranej linii referencyjnej.

Na podstawie przeprowadzonych badań pokazany także został potencjał samej chmury obliczeniowej w zakresie możliwości symulacyjnych na dużą skalę. Obliczenia wykonane zostały dla 10 000 przypadków, z których każdy obliczany był w przybliżeniu 5 minut. Łączny czas symulacji tak złożonego zagadnienia to prawie 50 000 minut, czyli 833 godziny. Przy jednoczesnym uruchomieniu zadania na 200 maszynach wirtualnych, czas ten skrócił się do 280 minut, czyli 4,66 godziny. Wyliczony narzut, według wzoru (18), to około 12%.

Całkowity koszt takich maszyn pracujących nad tym zadaniem przy analogicznej, jak wcześniej wspomnianej stawce to \$0.14/h \* 200 \* 4,5 czyli koszt \$126.

Końcowe zestawienie wyników z testów opracowanej platformy obliczeniowej jest zaprezentowane w Tabela 9. Potwierdza ono przydatność systemu, a także ekonomiczne uzasadnienie dla korzystania z chmury obliczeniowej.

	Liczba elementów modelu	Liczba symulacji	Czas rozwiązania	Narzut platformy	Koszt obliczeń
Grzanie rezystancyjne	4 128	10 000	50 [min.]	20 [%]	\$9
Elektrostymulacja nerwowo mięśniowa	9 530	100 000	74 [min.]	10.31 [%]	\$34
Terapia elektrowstrzasowa	1 734 728	10 000	280 [min.]	12 [%]	\$126

Tabela 9 - Zestawienie kosztów obliczeń i narzutu platformy

## ROZDZIAŁ 5. PODSUMOWANIE

W niniejszej pracy przedstawiona została metoda pozwalająca na rozwiązywanie problemów związanych z zagadnieniami symulacji tkanek żywych dzięki wykorzystaniu platformy obliczeń rozproszonych. Metoda obejmuje proces definiowania modelu obiektu wraz ze zdefiniowaniem podstawowych parametrów materiałowych uwzględniając ich zmienność. Stochastyczne podejście do zagadnienia pozwala uniknąć ograniczeń narzuconych przez tradycyjne metody deterministyczne. Wykorzystanie chmury obliczeniowej pozwala na zwiększenie liczby analizowanych przypadków symulacji, a tym samym dokładność rozwiązania.

Zagadnienie statystyczne oraz jego wykorzystanie w modelowaniu tkanek jest nowatorskim podejściem, które do tej pory nie było szeroko omawiane w literaturze. Praca obejmuje analizę teoretyczną zagadnienia stochastycznego, w którym uwzględnione są popularne rozkłady gęstości prawdopodobieństwa dla zmiennych losowych. Modelowanie tkanek z opisem parametrycznym dla rozkładów statystycznych wymaga uwzględnienia obszernej liczby przypadków. Jednocześnie stosowane obecnie sposoby zapisu zmienności tkanek okazują się być ograniczone poprzez wykorzystanie normalnych rozkładów prawdopodobieństwa. Badania przeprowadzone w ramach tej pracy pokazały, że zaprezentowane podejście stochastyczne prowadzi do dokładniejszych wyników, niż wykorzystanie klasycznych modeli.

Zagadnienie modelowania tkanek zostało w pracy omówione i zweryfikowane na przykładach homogenicznych oraz heterogenicznych obiektów. Zostało zbudowane urządzenie, które posłużyło do stworzenia własnej bazy danych parametrów wybranych tkanek biologicznych. Urządzenie pozwoliło także na wypracowanie pewnych reguł postępowania przy pomiarach tkanek, zrozumienia występujących zjawisk wpływających na pomiary oraz sposobu minimalizacji ich wpływu lub wyeliminowania.

Praca porusza także zagadnienie związane z platformą obliczeń rozproszonych opartą o chmurę obliczeniową. Badania skupiają się na przedstawieniu dostępnych rozwiązań chmurowych, wyjaśnieniu ich funkcjonowania oraz pokazują kompletną konfigurację usługi od

uruchomienia do działającego systemu obliczeniowego. Systemy obliczeń rozproszonych zostały także przebadane zarówno w infrastrukturze wykorzystującej rozwiązania Apache Hadoop oraz HTCondor. Zaobserwowano różnice między nimi oraz wyciągnięto wnioski na temat wad i zalet obu rozwiązań. Systemy zostały wdrożone i przetestowane w wielkoskalowej infrastrukturze chmury obliczeniowej Microsoft Azure.

Kluczowym elementem pracy było zaimplementowanie rozważań teoretycznych związanych z modelowaniem tkanek w oparciu o zagadnienia statystyczne w postaci modelu biologicznego. Przygotowane zostały trzy modele, w realnych scenariuszach obliczeniowych, które pozwoliły sprawdzić skuteczność oraz wydajność rozwiązania. Pierwszym przypadkiem było zagadnienie grzania rezystancyjnego na modelu dwuwymiarowym. Drugim zadaniem obliczeniowym był również przypadek dwuwymiarowy, jednak bardziej złożony, którego tematyka obejmowała zagadnienie stymulacji nerwowo-mięśniowej. Duża liczba parametrów oraz zmiennych zweryfikowała możliwości chmury obliczeniowej. Ostatnim scenariuszem była symulacja zabiegu terapii elektrowstrząsowej na trójwymiarowym, realistycznym modelu głowy ludzkiej.

Zaprezentowane rozwiązania oraz opracowane przykładowe scenariusze modelowe pokazały możliwości wykorzystania platform obliczeń rozproszonych w badaniach bioinżynieryjnych, w których wpływ rozwiązań technicznych na tkanki biologiczne może być znaczący.

Główną wadą zastosowanej techniki jest konieczność dobrego poznania złożonej technologii obliczeń chmurowych, jej konfiguracji oraz możliwości przeniesienia problemów na tę platformę. Początkowy nakład pracy szybko zwraca się poprzez dostęp do wydajnego systemu obliczeniowego.

Także brak dostępności parametrów żywych tkanek w literaturze nie pozwala na precyzyjne zamodelowanie obiektów biologicznych. Statystyczne narzędzia wprawdzie pozwalają na wygenerowanie rozkładów gęstości prawdopodobieństwa nawet w przypadku dwóch próbek, ale wiarygodność takich wyników jest mocno ograniczona.

Wyniki pracy pokazują, że uwzględnienie możliwości ze stosowania chmury obliczeniowej daje możliwości, które nie były do tej pory dostępne. Niniejsza praca zawiera szczegółowe instrukcje pozwalające na łatwe wdrożenie opracowanego rozwiązania.

Za główne osiągnięcia pracy uważam:

- przegląd dostępnych narzędzi statystycznych oraz metod wykorzystywanych w modelowaniu tkanek, uwzględniając zmienność parametryczną,
- zaprojektowanie metodyki pomiarowej tkanek oraz opracowanie modeli statystycznych parametrów materiałowych obiektów biologicznych,
- zbadanie dostępnych rozwiązań obliczeń rozproszonych oraz przetestowanie dostępnych usług w chmurze obliczeniowej,
- zaproponowanie oraz wdrożenie konfiguracji chmury obliczeniowej pozwalającej na przygotowanie środowiska obliczeniowego opartej o Microsoft Azure,
- przeprowadzenie trzech analiz praktycznych problemów bioelektromagnetyzmu z wykorzystaniem opracowanego systemu, które potwierdziły uniwersalność i skalowalność środowiska.

Powyższe osiągnięcia są potwierdzeniem, że cel postawiony w rozdziale pierwszym pracy został osiągnięty.

Dalsze prace badawcze prowadzone są w kierunku wykorzystania możliwości oferowanych przez innych dostawców chmur obliczeniowych oraz poprawienia wydajności systemu.

## BIBLIOGRAFIA

- [1] J. Malmivuo and R. Plonsey, Bioelectromagnetism Principles and Applications of Bioelectric and Biomagnetic Fields, New York: Oxford University Press, 1995.
- S. Gabriel, R. W. Lau and C. Gabriel, "The dielectric properties of biological tissues: III. Parametric models for the dielectric spectrum of tissues.," *Physics in Medicine & Biology*, vol. 41, no. 11, p. 2271, 1996.
- [3] G. Hartsgrove, A. Kraszewski and A. Surowiec, "Simulated biological materials for electromagnetic radiation absorption studies," *Bioelectromagnetics: Journal of the Bioelectromagnetics Society, The Society for Physical Regulation in Biology and Medicine, The European Bioelectromagnetics Association*, vol. 8, no. 1, pp. 29-36, 1987.
- [4] K. D. Paulsen, "Finite element modeling in therapeutic and diagnostic bioelectromagnetics applications," *Proceedings of the 19th Annual International Conference of the IEEE Engineering in Medicine and Biology Society 'agnificent Milestones and Emerging Opportunities in Medical Engineering'*, vol. 6, 1997.
- [5] F. Wei and et al., "Error measures for comparing bioelectromagnetic simulators," *Proceedings of the 2012 IEEE International Symposium on Antennas and Propagation*, 2012.
- [6] V. Pleskachev and et al., "On-body surface electromagnetic wave propagation: Modeling and measurements," *10th European Conference on Antennas and Propagation (EuCAP)*, 2016.
- [7] E. Sharifi, A. H. Bouchali and M. Saviz, "Creating 3D Geometric models of Cells and Organelles for Bioelectromagnetic Simulations.," *Modares Journal of Biotechnology*, vol. 9, no. 2, pp. 187-192, 2018.
- [8] S. N. Makarov and et al., "Virtual human models for electromagnetic studies and their applications.," *IEEE reviews in biomedical engineering*, vol. 10, pp. 95-121, 2017.
- [9] G. M. Noetscher and et al., "The visible human project male CAD based computational phantom and its use in bioelectromagnetic simulations.," *39th Annual*

International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2017.

- [10] J. W. Massey and et al., "Analyzing UHF-band antennas near anatomical human models with a fast integral-equation method.," *10th European Conference on Antennas and Propagation (EuCAP)*, 2016.
- [11] C. M. Furse, "Bioelectromagnetic Dosimetry: Simulating Electromagnetic Fields in the Human Body," in *The World of Applied Electromagnetics*, Cham, Springer, 2018, pp. 351-368.
- [12] V. De Santis and et al., "Human exposure from pulsed magnetic field therapy mats: a numerical case study with three commercial products," *Bioelectromagnetics*, vol. 36, no. 2, pp. 149-161, 2015.
- [13] L. Congsheng and et al., "Dosimetry Assessment for Human Exposure to Extremely Low Frequency Magnetic Fields in the Electric Vehicles," *12th International Symposium on Antennas, Propagation and EM Theory (ISAPE)*, 2018.
- [14] O. P. Gandhi, "Electromagnetic fields: human safety issues," *Annual review of biomedical engineering*, vol. 4, no. 1, pp. 211-234, 2002.
- [15] J. Haueisen, D. Tuch, C. Ramon, P. Schimpf and V. J. Wedeen, "The Influence of Brain Tissue Anisotropy on Human EEG and MEG," *Haueisen*, vol. 15, no. 1, pp. 159-166, 2002.
- [16] J. Starzyński, B. Sawicki, S. Wincenciak, A. Krawczyk and T. Zyss, "Simulation of Magnetic Stimulation of the Brain," *IEEE TRANSACTIONS ON MAGNETICS*, vol. 38, no. 2, pp. 1237-1240, 2002.
- [17] E. Conil, A. Hadjem, F. Lacroux, M. F. Wong and J. Wiart, "Variability analysis of SAR from 20 MHz to 2.4 GHz for different adult and child models using finitedifference time-domain," *Physics in Medicine & Biology*, vol. 53, no. 6, pp. 1511-1525, 2008.
- [18] B. Sawicki, "Uncertainty of numerical simulations in bioelectromagnetic problems," *Przegląd Elektrotechniczny*, vol. 91, no. 7, pp. 49-51, 2015.
- [19] S. Ferson and L. R. Ginzburg, "Different methods are needed to propagate ignorance and variability," *Reliability Engineering and System Safety*, vol. 54, no. 2-3, pp. 133-144, 1996.
- [20] P. Hasgall, E. Neufeld, M.-C. Gosselin and A. Klingenböck, "IT'IS Database for thermal and electromagnetic parameters of biological tissues," 2015. [Online]. Adres: https://itis.swiss/virtual-population/tissue-properties/overview/.
- [21] B. Sawicki and A. Krupa, "Experiments with models of variability for biological tissues," *Przegląd Elektrotechniczny*, pp. 83-86, 7 2016.

- [22] P. Hauseux, J. Hale, S. Cotin and S. Bordas, "Accelerating Monte Carlo estimation with derivatives of high-level finite element models," *Computer Methods in Applied Mechnics and Engineering*, vol. 318, pp. 917-936, 2017.
- [23] G. Stefanou, "The stochastic finite element method: past, present and future," *Computer methods in applied mechanics and engineering*, Vols. 198(9-12), pp. 1031-1051, 2009.
- [24] A. Kaintura, T. Dhaene and D. Spina, "Review of Polynomial Chaos-Based Methods for Uncertainty Quantification in Modern Integrated Circuits," *Electronics*, vol. 7, no. 3, p. 30, 2018.
- [25] S. Yang, X. Fenfen and W. Fenggang, "Polynomial Chaos Expansion for Probabilistic Uncertainty Propagation," Uncertainty Quantification and Model Calibration. InTech, 2017.
- [26] B. Sawicki and A. Krupa, "Stochastic Modelling of Electrical Field in the Plants Using Polynomial Chaos," *ITM Web of Conferences (SAGA)*, p. 8, 2019.
- [27] T. Crestaux, O. Le Mai<sup>tre</sup> and J.-M. Martinez, "Polynomial chaos expansion for sensitivity analysis," *Reliability Engineering & System Safety*, vol. 94, no. 7, pp. 1161-1172, 2009.
- [28] R. M. Claudio and E. Zio, "Global Sensitivity Analysis of Multi-State Markov Reliability Models of Power Equipment Approximated by Polynomial Chaos," *Journal of KONBiN*, vol. 23, no. 1, pp. 59-70, 2012.
- [29] N. Wiener, "The Homogenous Haos," *American Journal of Mathematics*, vol. 60, no. 4, pp. 897-936, 1938.
- [30] J. Feinberg and H. P. Langtangen, "Chaospy: An open source tool for designing methods of undertainty quantification," *Journal of Computational Science*, vol. 11, pp. 46-57, 2015.
- [31] "Measures of Skewness and Kurtosis," [Online]. Adres: : https://www.itl.nist.gov/div898/handbook/eda/section3/eda35b.htm. [Dostęp: 17 11 2018].
- [32] R. Nasser, K. Breitman and H. Lopes, "Monte Carlo Service in Windows Azure," Microsoft Research, Berkeley, California, 2012.
- [33] A. Cunha Jr, R. Nasser, R. Sampaio, H. Lopes and K. Breitman, "Uncertainty quantification through the Monte Carlo method in a cloud computing setting," *Computer Physics Communications*, vol. 185, no. 5, pp. 1355-1363, 2014.
- [34] H. Miras, R. Jiménez, C. Miras and C. Gomà, "CloudMC: a cloud computing application for Monte Carlo simulation," *Physics in Medicine & Biology*, vol. 58, no. 8, pp. 125-133, 2013.

- [35] D. Miklavcic, N. Pavselj and H. X. Francis, Electric properties of tissues, Wiley Encyclopedia of Biomedical Engineering, 2006.
- [36] F. L. H. Gielen, W. Wallinga-de Jonge and K. L. Boon, "Electrical conductivity of skeletal muscle tissue: Experimental results from different muscles in-vivo," *Medical and Biological Engineering and Computing*, vol. 22, no. 6, p. 569–577, 1984.
- [37] S. Torquato, Random heterogeneous materials: microstructure and macroscopic properties, Springer Science & Business Media, 2013.
- [38] A. Krupa and B. Sawicki, "Preliminary study of random variable algebra for biological tissue description," in 17th International Conference Computational Problems of Electrical Engineering, CPEE 2016, Sandomierz, Poland, 2016.
- [39] B. M. Heaney, "Electrical Conductivity and Resistivity," *Electrical measurement, signal processing, and displays,* p. 14, 2003.
- [40] C. Gabriel, A. Peyman and E. H. Grant, "Electrical conductivity of tissue at frequencies below 1 MHz," *Physics in medicine & biology*, vol. 54, no. 16, pp. 4863-4878, 2009.
- [41] J. Janesch, "Two-Wire vs. Four-Wire Resistance Measurements: Which Configuration Makes Sense for Your Application?," Keithley Instruments, Inc., 2013.
- [42] A. Krupa and B. Sawicki, "Measurement-based stochastic models of biological materials," in 18th International Conference on Computational Problems of Electrical Engineering, CPEE 2017, Hutna Hora, Czech Republic, 2017.
- [43] S. Sarang, S. K. Sastry and L. Knipe, "Electrical conductivity of fruits and meats during ohmic heating," *Journal of Food Engineering*, vol. 87, pp. 351-356, 2008.
- [44] L. Juha, T. Kuurne and H. Eskola, "Conductivity of living intracranial tissues," *Physics in Medicine & Biology*, vol. 46, no. 6, pp. 1611-1616, 2001.
- [45] I. Foster and C. Kesselman, The grid: blueprint for a new computing infrastructure, San Francisco, CA: Morgan Kaufmann Publishers, 1998.
- [46] H. Stockinger, "Defining the Grid: A Snapshot on the Current View," *The Journal of Supercomputing*, vol. 42, no. 1, pp. 3-17, 2007.
- [47] I. Foster, "What is the grid?-a three point checklist," *GRIDtoday*, vol. 1, no. 6, pp. 1-4, 2002.
- [48] I. Foster, Y. Zhao, I. Raicu and S. Lu, "Cloud Computing and Grid Computing 360-Degree Compared," *arXiv preprint*, vol. 0901.0131, pp. 1-10, 2008.

- [49] F. Chávez, M. Rubio Del Solar, J. L. Guisado and F. . F. de Vega, "A Genetic Programming infrastructure profiting from public computation resources," pp. 1-8, 2007.
- [50] "An Introduction to Hyper-V Virtualization in Windows Server 2008," [Online]. Adres: https://technet.microsoft.com/en-us/library/2008.10.hyperv.aspx. [Dostęp: 25 07 2018].
- [51] "Easy Auth Overview Tokyo Azure Meetup Feb 2018," [Online]. Adres: : https://www.slideshare.net/ChrisGillum/easy-auth-overview-tokyo-azure-meetupfeb-2018. [Dostęp: 25 07 2018].
- [52] "A Comparison of AWS vs Azure vs Google," [Online]. Adres: https://www.cloudhealthtech.com/blog/aws-vs-azure-vs-google. [Dostęp: 17 11 2018].
- [53] "Destiny The Cloud: Amazon Web Services Part 1: Do you know all of these icons?," [Online]. Adres: http://cloudn1n3.blogspot.com/2014/11/22-amazon-webservices-part-1-do-you.html. [Dostęp: 25 07 2018].
- [54] "Why to choose AWS Cloud for your Web Application?," [Online]. Adres: https://www.clickittech.com/aws/why-aws-cloud/. [Dostęp: 25 07 2018].
- [55] "FEniCS Project," [Online]. Adres: https://fenicsproject.org/. [Dostęp: 25 07 2018].
- [56] "The Google File System," [Online]. Adres: http://research.google.com/archive/gfs.html. [Dostęp: 25 07 2018].
- [57] "MapReduce: Simplified Data Processing on Large Clusters," [Online]. Adres: https://ai.google/research/pubs/pub62. [Dostęp: 25 07 2018].
- [58] A. Krupa and B. Sawicki, "Massive Simulations Using MapReduce Model," Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska, pp. 45-47, 2015.
- [59] "Apache Hadoop YARN," [Online]. Adres: https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarnsite/YARN.html. [Dostęp: 17 11 2018].
- [60] A. Krupa and B. Sawicki, "High-resolution scatter analyse using cloud computing," *Przegląd Elektrotechniczny*, pp. 140-142, 12 2015.
- [61] B. Sawicki and A. Krupa, "Monte Carlo optimization of coil geometry using Microsoft Azure cloud platform," in *Computational Problems of Electrical Engineering*, Slovak Republic, 2018.
- [62] M. Sark and S. Liu, "A comprehensive review on applications of ohmic heating (OH)," *Renewable and Sustainable Energy Reviews*, vol. 39, pp. 262-269, 2014.

- [63] B. Sawicki and A. Krupa, "Study of biological tissues variability modeling," in 16th International Conference on Computational Problems of Electrical Engineering, CPEE 2015, Lviv, Ukraine, 2015.
- [64] W. R. Franklin, "Even-odd rule," [Online]. Adres: https://wrf.ecse.rpi.edu/Research/Short\_Notes/pnpoly.html. [Dostęp: 17 11 2018].
- [65] N. A. Maffiuletti, M. A. Minetto, D. Farina and R. Bottinelli, "Electrical stimulation for neuromuscular testing and training: state-of-the art and unresolved issues," *European Journal of Applied Physiology*, vol. 111, pp. 2391-2397, 2011.
- [66] M. Schuenke, E. Schulte and U. Schumacher, "3.7 Cross sections of the Arm and Forearm," in *General Anatomy and Musculoskeletal System*, THIEME, 2014, p. 624.
- [67] "Medial Upper-Arm Circumfence," [Online]. Adres: http://www.nephjc.com/news/2017/10/9/median-upper-arm-circumferencewhatsup-with-that. [Dostęp: 10 02 2018].
- [68] S. M. Jain, K. Pandey, A. Lahoti and P. Kiran Rao, "Evaluation of skin and subcutaneous tissue thickness at insulin injection sites in Indian, insulin naïve, type-2 diabetic adult population," *PubMed*, vol. 17, no. 5, pp. 864-870, 2013.
- [69] "Median upper arm circumference...what's up with that. | NephJC," US NHANES, [Online]. Adres: http://www.nephjc.com/news/2017/10/9/median-upper-armcircumferencewhats-up-with-that. [Dostęp: 17 11 2018].
- [70] H. W. Folkerts, "Elektrokrampftherapie," *Der Nervenarzt,* vol. 82, no. 1, pp. 93-103, 2011.
- [71] H. W. Folkerts, "Terapia elektrowstrząsowa. Wskazania, procedury i wyniki leczenia," *Psychiatria po dyplomie*, vol. 10, no. 3, pp. 5-13, 2013.
- [72] "Voltage of Electroshock Therapy The Physics Factbook," [Online]. Adres: https://hypertextbook.com/facts/2005/GinaCastellano.shtml. [Dostęp: 04 01 2019].
- [73] "Microsoft Build Live 2018," [Online]. Adres: https://developer.microsoft.com/enus/events/build/content/. [Dostęp: 25 07 2018].
- [74] "Inside Microsoft Azure datacenter hardware and software architecture with Mark Russinovich," [Online]. Adres: https://channel9.msdn.com/Events/Ignite/Microsoft-Ignite-Orlando-2017/BRK3203. [Dostęp: 25 07 2018].
- [75] "Microsoft Azure and Open Source TechNet Articles," [Online]. Adres: https://social.technet.microsoft.com/wiki/contents/articles/36473.microsoft-azureand-open-source.aspx. [Dostęp: 25 07 2018].

- [76] "What Is Azure Application Insights? | Microsoft Docs," [Online]. Adres: https://docs.microsoft.com/en-us/azure/application-insights/app-insights-overview. [Dostęp: 25 07 2018].
- [77] "It's Me, and Here's My Proof: Why Identity and Authentication Must Remain Distinct," [Online]. Adres: https://technet.microsoft.com/enus/library/cc512578.aspx. [Dostęp: 25 07 2018].
- [78] "Deploying Applications to Azure Virtual Machine Scale Sets," [Online]. Adres: https://blogs.msdn.microsoft.com/devops/2017/05/15/deploying-applications-to-azure-vm-scale-sets/. [Dostęp: 14 11 2017].
- [79] "Virtual Machine Scale Set czyli o tym, jak "uPaaSowić" maszyny wirtualne w Azure. cz. 1," [Online]. Adres: http://cloudarchitects.pl/2017/10/virtual-machinescale-set-czyli-o-tym-jak-upaasowic-maszyny-wirtualne-w-azure/. [Dostęp: 22 07 2018].
- [80] "Check VM Sizes Available Region Using Azure CLI 2.0," [Online]. Adres: https://www.itprotoday.com/iaaspaas/check-vm-sizes-available-region-usingazure-cli-20. [Dostęp: 27 11 2018].
- [81] "AZ VM | Microsoft Docs," [Online]. Adres: https://docs.microsoft.com/enus/cli/azure/vm?view=azure-cli-latest. [Dostęp: 17 11 2018].
- [82] "Azure VM Scale Sets public preview," [Online]. Adres: https://azure.microsoft.com/en-us/blog/azure-vm-scale-sets-public-preview/. [Dostęp: 25 07 2018].
- [83] "About Disks and VHDs," [Online]. Adres: https://docs.microsoft.com/enus/azure/virtual-machines/windows/about-disks-and-vhds. [Dostęp: 17 11 2018].
- [84] "Convert a Windows virtual machine from unmanaged disks to managed disks," [Online]. Adres: https://github.com/MicrosoftDocs/azuredocs/blob/master/articles/virtual-machines/windows/convert-unmanaged-tomanaged-disks.md. [Dostęp: 17 11 2018].
- [85] "Azure CLI Documentation az image command," [Online]. Adres: https://docs.microsoft.com/en-us/cli/azure/image?view=azure-cli-latest. [Dostęp: 17 11 2018].
- [86] "Create a managed image of a generalized VM in Azure," [Online]. Adres: https://docs.microsoft.com/en-us/azure/virtual-machines/windows/capture-imageresource. [Dostęp: 17 11 2018].
- [87] "Azure Subscription Service Limits," [Online]. Adres: https://docs.microsoft.com/en-us/azure/azure-subscription-service-limits. [Dostęp: 17 11 2018].

# ZAŁĄCZNIK 1 ARCHITEKTURA CHMURY MICROSOFT AZURE

Chmura obliczeniowa jest zbiorem połączonych ze sobą centrów danych, których zasoby są zrównoleglone, a dostęp do nich zapewnia odpowiednio przygotowana infrastruktura połączeń sieciowych, systemów operacyjnych oraz interfejsów programowych, które pozwalają na korzystanie z zasobów w wydajny i efektywny sposób.

Z racji grantu przyznanego w ramach umowy Uczelni z firmą Microsoft na wykorzystanie zasobów Microsoft Azure, na tej infrastrukturze omówiona zostanie architektura chmury obliczeniowej. Wiedza zawarta w tym załączniku oparta jest o oficjalnie dostępne źródła internetowe, w tym materiały z ostatniej konferencji Microsoft Build [73], która odbyła się w Seattle w stanie Waszyngton w dniach 7-9 maja 2018.

Należy podkreślić, że chmura obliczeniowa nie stanowi jednolitej struktury. Jest to architektura często rozproszona, złożona z wielu elementów. Przykładem takiej chmury obliczeniowej jest właśnie Microsoft Azure.

Użytkownik, posiadający subskrypcję, jest przypisywany do wybranego regionu, a nie do określonego centrum danych (Rys. 53). Co ważne, każdy region automatycznie dokonuje wzajemnego powielenia wszystkich zasobów i grup zasobów między trzema pobliskimi centrami danych. Dzięki temu każda usługa uruchomiona przez użytkownika jest automatycznie zwielokrotniona w celu zapewnienia szybkości działania, ale i bezpieczeństwa danych. Każdy region złożony z centrów danych ma gwarantowany czas odpowiedzi wewnątrz regionu poniżej 600 us.

Według informacji Microsoft, czas odpowiedzi z centrum danych do punktu w regionie to mniej niż 2 ms. Jednocześnie każde centrum danych ma swoje własne systemy ochrony przed awariami, czy przepięciami realizowane przez dodatkowe doprowadzone linie zasilania, czy punkty dostępowe. Odległości między regionami to setki kilometrów.



Rys. 53 - Strefy dostępności (po lewej) oraz architektura (po prawej) Regionu Microsoft Azure [73]

Region jest dość złożoną strukturą. Oprócz obecności i poprawnego funkcjonowania centrów danych, wymagane jest zapewnienie ich komunikacji z siecią zewnętrzną chmury (określaną mianem Microsoft Backbone). Taką strukturę komunikacji ilustruje grafika (Rys. 53). Dla zapewnienia tej komunikacji z siecią zewnętrzną, każde centrum danych łączy się dodatkowo z centrami danych zwanymi RNG (ang. Regional Network Gateway), czyli sieciowymi centrami łączności. Centra danych zawierają w sobie tylko infrastrukturę sieciową. Każde centrum danych łączy się z dwoma, wzajemnie zduplikowanymi, centrami RNG wewnątrz regionu, a dopiero RNG łączy się z Backbone.
Sama sieć RNG jest, z uwagi na wielkość swojej struktury, określana przez firmę Microsoft jako "T-Shirt Size". W zależności od liczby centrów danych w regionie, sieć skalowana jest do rozmiarów S (ang. Small), M (ang. Medium) lub L (ang. Large), nazewnictwem nawiązując do rozmiarów ubrań. Sieć optyczna RNG sięga przepustowości 100 Gbps.

Sieć Microsoft Azure to nie tylko część komunikacyjna centrów danych, ale przede wszystkim samo centrum danych. Również jest to dość złożona struktura, opisywana jako SDN (ang. Software Defined Network). Łączy w sobie funkcjonalność planowania w Zarządzaniu, Kontroli oraz zakresie obsługi Danych. Każda funkcjonalność realizuje pewne scenariusze. Funkcja związana z Zarządzaniem otrzymuje zadanie które brzmi np. "stwórz użytkownika". W odpowiedzi kolejna funkcja związana z Kontrolą otrzymuje zadanie, które może brzmieć przykładowo: "określ konieczne zasoby do wykonania przypisania użytkownika". Ostatecznie funkcja związana z Danymi może brzmieć jak: "wykorzystaj konieczne zasoby do stworzenia użytkownika". Ideą takiego rozwiązania jest hierarchiczne postępowanie bez konieczności znajomości struktury poniżej. Pozwala to na sprawniejsze wykonanie zleconego zadania wdrożeniowego.

Przedstawiona architektura chmury obliczeniowej (Rys. 54) obrazuje wspomniany wcześniej hierarchiczny stos usług i elementów, które stanowią o tym, jak wydajne i poprawne będzie działanie architektury.



Rys. 54 - Architektura Microsoft Azure [73]

Schemat obejmuje zarówno część sprzętową, jak i punkty końcowe (ang. endpoints) po drugiej stronie chmury obliczeniowej. Znajdują się tam użytkownicy, interfejsy dla rozwiązań onpremise, a także urządzenia i rozwiązania firm trzecich.

Na samym dole struktury widoczna jest architektura wszystkich serwerowni, których każda stanowi część regionu. Region za pośrednictwem wspomnianego Microsoft Backbone stanowi całość określaną jako "Azure Architecture". Nad regionem znajduje się Menadżer sprzętowy, czyli element zarządzający typami dostępnych zasobów, na których można będzie uruchomić wymagane usługi (np. serwery bazodanowe, serwery wysokiej wydajności HPC, czy maszyny ze wsparciem dla przetwarzania GPGPU).

Według oficjalnych informacji podanych przez Marka Russinovich'a (CTO of Microsoft Azure) podczas konferencji Microsoft Ignite w 2017 roku w Orlando na Florydzie [74], Menadżerem w chmurze Microsoft jest Azure OS. Jest oparty o system Windows Server Core 2016 na którym uruchomiony jest Hyper-V, czyli wspomniany wcześniej VMM.

W celu minimalizacji zużycia zasobów, system wspiera tylko natywny 64-bitowy kod, bez dodatkowych narzędzi pozwalających na zarządzanie kodem na hoście. System ponadto pozbawiony jest dodatkowych języków i ich obsługi, poza angielskim. Wyłączone i zablokowane są wszystkie zbędne usługi systemowe, jak np. bufor wydruku. Sam system uruchomiony jest z minimalną liczbą ról na serwerze, aby odciążyć wszystkie zasoby, mogące być niepotrzebnie zużywane.

Kolejnym poziomem w hierarchii, znajdującym się zaraz nad systemem nadzorującym pracę całej chmury jest Azure Fabric Controller. To obecnie jeden z najważniejszych elementów strukturalnych platformy Microsoft Azure oraz integralna część ASP (ang. Azure Service Platform). Jest to cała funkcjonalność komunikacyjna między zasobami w najniższej warstwie architektury, a tym co zostało przekazane i przesłane w formie zadań, zgłoszeń lub komunikatów do przetworzenia w odpowiedniej postaci przez infrastrukturę.

Do zadań Azure Fabric Controller należą między innymi:

- HA, czyli zapewnienie wysokiej wydajności (ang. High Availablity)
- DP, czyli Określenie partycjonowania danych (ang. Data Partitioning)
- LB, czyli obsługa rozkład obciążenia sieciowego (ang. Load Balancer)
- ARF, czyli replikacja danych i zapewnienie bezawaryjności (ang. Auto Replication & Failover)
- HM, czyli monitorowanie życia serwisów (ang. Health Monitoring)
- FSS, czyli obsługa uruchamiania i zamykania/dealokacji zasobów (ang. Fast Startup & Shutdown)

Service Fabric obsługuje powyższe zadania sprawdzając dostępne i wymagane zasoby oraz przygotowuje zadania, które będą realizowane na węzłach klastrów (ang. Cluster Nodes). Jednocześnie wykonane zadania, występujące błędy, czy komunikaty zwracane są w hierarchii o poziom wyżej, dzięki czemu Menadżer Zasobów (ang. Azure Resource Manager – ARM) jest w stanie przygotować odpowiednie komunikaty dla użytkownika korzystającego z Portalu, linii poleceń (CLI), czy narzędzi komunikacyjnych firm trzecich.

Ważną kwestią w działaniu Azure Fabric Controller to funkcja kontrolera, czyli zbieranie wszystkich informacji z warstw poniżej i powyżej oraz tłumaczenia ich na odpowiednie polecenia lub zadania.

Każdy element struktury chmury obliczeniowej, w tym węzły partycji, repozytoria obrazów maszyn, maszyny wirtualne, mają pod sobą uruchomiony monitor o nadrzędnej funkcji kontroli. Jest to tzw. FCA (ang. Fabric Controller Agent), czyli Agent pracujący z najwyższymi uprawnieniami, mogący zainicjować aktualizację systemu, jego wstrzymanie lub zamknięcie, zmianę rozmiaru partycji. Praktycznie każdą operację z najwyższymi uprawnieniami.

Co ważne, Fabric Controller korzysta ze swojego Agenta tylko w przypadku rozwiązania PaaS (Rys. 55), natomiast z racji zupełnie innej struktury dostępności dla rozwiązań typu IaaS, funkcje kontrolne przejmuje bezpośrednio Hypervisor Azure. Nie oznacza to jednak, że AFC nie przekazuje poleceń w dół hierarchii. Jego uprawnienia i przywileje są jednak mniejsze.

Związane jest to z tym, że w przypadku rozwiązania IaaS, użytkownik ma dostęp do fizycznego systemu operacyjnego i do usług jakie są instalowane na dedykowanym dla niego sprzęcie. Tym samym instalacja np. spersonalizowanego systemu operacyjnego będzie pozbawiona

Agenta FC. Jednak nad całym systemem czuwa AFC, który ma dostęp do niżej położonego w hierarchii Hypervisora. Polecenie przesłane do Azure, które ma np. włączyć lub uruchomić ponownie zasoby, bądź maszynę wirtualną, nadal będzie możliwe. Chociaż sam AF nie ma konektora do monitorowania stanu maszyn, czy usług, jak w przypadku rozwiązania PaaS.



*Rys.* 55 - Porównanie możliwości zarządzania warstw infrastruktury w zależności od typu usług [75]

Nad Azure Fabric Controller znajduje się szereg elementów, stanowiących o strukturze chmury firmy Microsoft. Są to m.in. Moduł Telemetryczny, Autentykacyjny, RBAC (ang. Role-Based Access Control), SF wraz AKS (ang. Azure Kubernetes Service), a także dedykowane RP (ang. Resource Provider).

Moduł Telemetryczny, zwany "Telemetry & Insights", to rozszerzenie APM (ang. Application Performance Management), które pozwala na zbieranie informacji z usług, serwisów, procesów i modułów uruchomionych w ramach Azure (Rys. 56) i poprzez odpowiedni moduł zwany AI (ang. Azure Insights).



Rys. 56 – Schemat działania i przekazywania danych z usług Azure do wybranych interfejsów/API [76]

Jak widać na schemacie, każdy element w chmurze Microsoft Azure dzięki ASF jest w stanie zebrać informacje niezbędne do monitorowania życia aplikacji, zużywanych zasobów sprzętowych, czy na komunikowanie się ze skorelowanymi zasobami, korzystając z funkcji AI. Wystawione połączenia (ang. connectors) pozwalają na wykorzystanie dowolnego środowiska, czy rozwiązania.

Dla przykładu Alerts to komunikaty otrzymywane w środowisku Portalu Azure lub przez linię poleceń. PowerBI pozwala na tworzenie rozbudowanych raportów w czasie niemal rzeczywistym odświeżając dane przez wystawione AI. Opcja Continuous Export pozwala na ciągłe kolekcjonowanie danych zbieranych ze wszystkich lub wybranych usług i przekazywanie ich dalej do bazy danych, która może być wykorzystana przy przetwarzaniu typu BigData.

Kolejnym elementem z bezpośrednim dostępem od strony użytkownika końcowego jest moduł Autentykacji. Jest to tak naprawdę usługa AAD (ang. Azure Active Directory) określana jako ADFS (ang. Active Directory Federation Services) dla platformy Microsoft Azure. Jest to usługa, która pozwala na autentykację z zasobami, uprawnieniami oraz dostępem subskrypcji, do wykonywania i realizowania określonych ról, według których działa Azure.

Łącząc się z Microsoft Azure, zawsze jesteśmy proszeni o poświadczenia, aby móc określić nasze uprawnienia, a tym samym role i dostęp do zasobów. Ale poświadczenia są także wykorzystywane w celu uzyskania dziedziczonych uprawnień, jeśli takie istnieją.

0	Provided by	Answers	Attributes
Identity	principal	"Who are you?"	public assertion
Authentication	principal	"OK, how can you prove it?"	secret response
Authorization	system	"What can I do?" —	token or ticket
			access control

Rys. 57 - Porównanie procedury określania tożsamości, uwierzytelniania i autoryzacji [77]

Jak dobrze widać z powyższego rysunku (Rys. 57), autentykacja, to przedstawienie informacji, która nadaje odpowiednie uprawnienia dla konkretnego zasobu, obiektu, zbioru danych. Jest to zaszyfrowane hasło, bądź token (który w dodatku może zawierać zapisane informacje

o uprawnieniach, dacie ważności, profilu uprawnień, itd.).Z autoryzacją, nadawaną po podaniu żądanej tożsamości związany jest kolejny element strukturalny, czyli RBAC.

Najpopularniejszą obecnie strukturą dostępu jest przyznawanie ról i dostęp dziedziczony (w formie drzewiastej). Jeśli użytkownik zostanie autoryzowany z uprawnieniami dostępu do zasobów, czy usług na danym poziomie (np. w grupie zasobów), wówczas automatycznie otrzymuje dostęp do wszystkiego co w takiej grupie jest. Określenie roli także pozwala na dostęp do modułów i zasobów chmury obliczeniowej, które są z danym kontem o odpowiednich przywilejach (uprawnieniach) powiązane. Użytkownik grupy odpowiadającej za obsługę maszyn wirtualnych może mieć dostęp do wszystkich grup zasobów w ramach subskrypcji, ale będzie mógł zobaczyć tylko usługi spełniające kryteria maszyny wirtualnej. Tym samym nie jest ograniczany do konkretnych grup zasobów, ale i nie jest autoryzowany do przeglądania wszystkiego, co w grupie zasobów się znajduje.

Kolejnym elementem w strukturze nad AFC jest Service Fabric oraz AKS (Azure Kubernetes Services). Pierwsza usługa to wspomniana wcześniej usługa Service Fabric, dedykowana do obsługi mikroserwisów oparta o ekosystem Windows oraz całego Microsoft Azure. AKS to system pozwalający na zarządzanie (ang. orchestrate) rozwiązaniami kontenerowymi głównie opartymi o rozwiązania linuksowe.

Oba rozwiązania pozwalają na obsługę rozwiązań aplikacyjnych wielu środowisk do jednego, zwartego – nazywanego kontenerem. Kontener to kolejna hierarchia w strukturze serwerowej pozwalająca na współdzielenie pewnych zasobów bez konieczności uruchamiania całego systemu dla takiego rozwiązania. Dla przykładu uruchomiona aplikacja potrzebuje tylko kilku bibliotek systemowych, zasobów do uruchomienia, pamięci do alokacji swoich plików. Nie ma konieczności uruchamiania całego środowiska systemowego, które będzie w dodatku zużywało więcej zasobów. W takiej sytuacji przygotowuje się rozwiązanie w postaci paczki (kontenera), który zawiera najpotrzebniejsze pliki, niezbędne do poprawnego działania usługi i rozwiązania.

Podobnie jest w przypadku rozwoju rozwiązania, czy wykorzystywania wielu bibliotek i danych z różnych rozwiązań. Porównując do kontenerów na statkach transportowych, mogą one być stawiane jeden na drugim. Tak samo aplikacja ze swoimi plikami, wymagająca innego modułu lub usługi (np. bazy danych SQL) może odwołać się do kontenera z przygotowanym

rozwiązaniem bazodanowym bez konieczności instalowania fizycznie na zasobach sprzętowych rozwiązania SQL.

System nadrzędny nie wymaga instalacji aplikacji ze wszystkimi zasobami, ale uruchamia w swoim środowisku tylko odizolowaną aplikację, korzystając ze wszystkiego, co w kontenerze się znajduje. Aplikacja jest dostarczana w całości oraz jest przenośna. Uruchomienie tego samego kontenera na innym systemie z zainstalowanym orchestratorem pozwala na zapewnienie takiej samej funkcjonalności aplikacji na zupełnie innym węźle. Przenośność to zaleta rozwiązania kontenerowego.Rozwiązań zarządzających zasobami, czy usługami jest wiele, a są to np. Docker Swarm, Mesos DC/OS czy wspomniany wcześniej Kubernetes.

Powyżej w strukturze jest ARM, czyli globalna część systemu, która z jednej strony zapewnia dostęp do części UI (ang. User Interface) typu Portal, czy API, z drugiej zapewnia przekazywanie poleceń dla konkretnych usług w strukturze hierarchicznej poniżej. Tłumaczy także odpowiednie zadania, jak ładowanie szablonów konfiguracyjnych, sprawdzanie dostępności usług dla subskrypcji.

Tak przedstawiony model jest tylko przybliżonym schematem infrastruktury chmury obliczeniowej. Architektura (jak widoczna na Rys. 54) nie jest jednak stała. Nieustannie zmienia się zapewniając nie tylko dostępność nowych usług, ale i rozwój fizyczny przez ciągle rozrastające się środowiska, systemy, urządzenia w każdej skali.

## ZAŁĄCZNIK 2 WDROŻENIE W MICROSOFT AZURE

Posiadanie zasobów w chmurze w postaci subskrypcji jest dopiero pierwszym krokiem do działającego rozwiązania. Aby móc wykorzystać zasoby w sposób efektywny, należy poprawnie skonfigurować usługi. Proces przygotowania chmury obliczeniowej do gotowości do obliczeń to etap wdrożenia. W przypadku prowadzonych badań, całość środowiska opierała się o platformę Microsoft Azure. Zastosowane rozwiązanie w Microsoft Azure zostało opracowane w modelu PaaS. Zapewnia optymalny poziom możliwości zarządzania, bez ograniczania w zakresie konfiguracji usług.

Konfiguracja środowiska do pracy jest dość złożonym procesem. W przypadku obliczeń rozproszonych, dla których wykorzystana będzie znaczna ilość jednostek obliczeniowych, wszystkie usługi i procesy muszą być ze sobą ściśle powiązane. Ułatwienia w zakresie jednolitości środowiska zapewnia chmura obliczeniowa, lecz funkcjonalność uruchomionych usług zależy od użytkownika.

Wprowadzenie do infrastruktury chmury zostało zaprezentowane w Załączniku 1. Oddzielne wyjaśnienie niezbędne jest w przypadku procedury konfiguracji maszyn wirtualnych oraz przygotowania obrazu VMSS.

Celem zastosowanego rozwiązania, była możliwość wdrożenia odpowiednio skonfigurowanej aplikacji na wszystkich dostępnych maszynach pracujących w ramach jednej sieci. Każda z jednostek obliczeniowych jest identyczna pod względem zainstalowanego systemu operacyjnego, oprogramowania, uruchomionych usług, konfiguracji do pracy w sieci.

Kwestia konfiguracji własnego rozwiązania środowiskowego została już poruszona [78], jednak opisane możliwości były zbyt rozbudowane w stosunku do realnych potrzeb (Rys. 58).



Rys. 58 - Schemat wdrożenia własnego obrazu z wykorzystaniem repozytorium Azure [78]

Środowisko zostało uproszczone, a konfiguracja oparta była na rozwiązaniach przygotowanych przez osoby z polskiej grupy Azure, którą opublikowano w formie poradnika [79]. Scenariusz ten dobrze wpasowywał się w oczekiwane rozwiązanie.

Microsoft Azure posiada dwa zasadnicze tryby pracy dla użytkownika. Jest to wspomniany wcześniej interfejs graficzny (GUI) oraz interfejs linii poleceń (CLI). Ten drugi jest skuteczniejszy i posiada znacznie więcej opcji, niż wersja graficzna. Dlatego też wszystkie polecenia jakie są wykonywane, realizowane są w trybie tekstowym.

Pierwszą rzeczą, jaką należało zrealizować, to zalogowanie się do portalu Microsoft Azure<sup>41</sup> i wywołanie linii poleceń wprost na stronie internetowej. To przydatne rozwiązanie, ponieważ posiadacze innych systemów operacyjnych, niż Microsoft Windows, chcący skorzystać z możliwości konfiguracji platformy Azure w formie tekstowej, nie są zmuszani to instalowania dodatkowego oprogramowania. Taką alternatywą jest wieloplatformowe PowerShell Core<sup>42</sup>.

Skorzystanie z Azure Cloud Shell, bo tak nazywa się interfejs tekstowy w nomenklaturze Microsoft, wywoływany jest przyciskiem w górnej części witryny Portalu. Co ważne, dla kompatybilności ze starszą wersją interfejsu, możliwe jest skorzystanie także z wersji Bash (określanym jako CLI 1.0), które opiera się o rozwiązanie linuksowe. Kompatybilność wsteczna jest dostępna, lecz nie w 100%. Polecenia z CLI 1.0 w znacznej większości działają

<sup>&</sup>lt;sup>41</sup> Źródło: https://portal.azure.com/ (dostęp: 2017.11.25)

<sup>&</sup>lt;sup>42</sup> Źródło: https://github.com/PowerShell/PowerShell (dostęp: 2017.11.17)

na nowszym interfejsie. Dla potrzeb naszych rozwiązań będzie wykorzystane jednak CLI 2.0 (Rys. 59).



Rys. 59 - Widok interfejsu PowerShell na stronie Portalu Azure

Po zalogowaniu do Azure Cloud Shell należy wykonać polecenie, które określi nam, na jakiej subskrypcji chcemy wykonywać polecenia (Kod 1).

```
1. az account set --subscription <ID>
```

W miejscu *<ID>* należy podać identyfikator subskrypcji w formacie GUID. Jest to istotny krok, ponieważ brak wykonania tego polecenia na samym początku, może spowodować błędy o niezrozumiałej treści. Nie są jednak związane z nieprawidłową składnią polecenia, ale z tym, że powłoka Azure nie jest w stanie określić na jakich zasobach polecenie powinno zostać wykonane.

Węzeł nadrzędny (Master Node)

Kolejnym krokiem jest utworzenie jednostki obliczeniowej, która zostanie skonfigurowana jako węzeł dystrybucyjny, oznaczony w poprzednim podrozdziale jako Master Node. Zadanie to jest realizowane poleceniem *az vm create* z odpowiednimi przełącznikami (Kod 2).

2. az vm create -n <NAZWA> -l <REGION> -g <GRUPA> --size <ROZMIAR> -use-unmanaged-disk --admin-username <UŻYTKOWNIK> --admin-password
<HASŁO> --image UbuntuLTS --storage-sku Standard\_LRS

W poleceniu pojawia się kilka zmiennych, które oznaczają odpowiednio:  $\langle NAZWA \rangle$  – nazwa maszyny wirtualnej,  $\langle REGION \rangle$  – oznacza wspomniany w Załączniku 1 region geograficzny w którym grupa zasobów zostanie umieszczona,  $\langle GRUPA \rangle$  – nazwa grupy zasobów w której zostanie umieszczona maszyna wirtualna,  $\langle SIZE \rangle$  – typ maszyny wirtualnej w nomenklaturze Microsoft Azure,  $\langle UZYTKOWNIK \rangle$  – nazwa użytkownika w systemie operacyjnym na maszynie wirtualnej,  $\langle HASLO \rangle$  – hasło dla wspomnianego użytkownika maszyny wirtualnej.

Jak widać powyżej istotne są dwie zmienne, *<REGION>* oraz *<SIZE>*. Obie można określić znając strukturę platformy Azure, wymaga to jednak doświadczenia.

W przypadku osób mniej obeznanych, listę dostępnych lokalizacji można pobrać wprost z centralnej listy zasobów ARM. Polecenie wyświetlające lokalizacje to (Kod 3)

3. azure location list

W odpowiedzi na wykonane polecenie, uzyskamy listę lokalizacji z wyświetlaną przyjazną użytkownikowi nazwą. Dodatkowo polecenie pobiera i wyświetla także listę zasobów, które w ramach danej lokalizacji są dostępne dla subskrypcji użytkownika. Co ważne, polecenie wyświetla tylko listę lokalizacji powiązanej z daną subskrypcją. Jeśli subskrypcja jest przypisana tylko do jednego lub kilku regionów, lista będzie niepełna. Tym samym informacje przekazywane użytkownikowi są zmaksymalizowane do uprawnień, jakie posiada.

```
4. data:
         Location : southcentralus
5. data:
         DisplayName : South Central US
6. data:
         Providers : Microsoft.Advisor, Microsoft.Compute,
  Microsoft.Network, Microsoft.Portal...
7. data:
8. data:
         Location : northeurope
9. data: DisplayName : North Europe
10. data:
           Providers : Microsoft.Advisor, Microsoft.Compute,
  Microsoft.Network, Microsoft.Portal...
11. data:
12. data: Location : westeurope
13. data:
           DisplayName : West Europe
14. data: Providers : Microsoft.Advisor, Microsoft.Compute,
  Microsoft.Network, Microsoft.Portal...
```

Wyświetlona lista (Kod 4-14) to tylko fragment odpowiedzi, jaka wyświetla się po wykonaniu polecenia. Dla naszych potrzeb istotna jest nazwa opisana jako *Location*, czyli wartość np. "westeurope". Dość ważne w wyborze lokalizacji jest dobranie jej w taki sposób, aby była ona rozmieszczona jak najbliżej miejsca, z którego będzie do niej najczęściej uzyskiwany dostęp. Jeżeli jesteśmy w Polsce, najbardziej adekwatnym wyborem jest jedna z lokalizacji w Europie. W tym przypadku dostępne są dwie pozycje: North Europe oraz East Europe.

Drugim ważnym elementem w konfiguracji jest określenie rozmiaru maszyny wirtualnej. W praktyce wiąże się to nie tylko z parametrami technicznymi (jak CPU, pamięć RAM, czy wielkość dysku), ale przede wszystkim ze stawką za godzinę pracy takiej maszyny. Lista dostępnych dla subskrypcji maszyn jest dostępna ponownie albo z poziomu Portalu, albo przez polecenie wykonywane za pomocą CLI [80] [81] (Kod 15).

15. az vm list -l <REGION>

Polecenie to powoduje wyświetlenie listy dostępnych maszyn wirtualnych wraz z podstawowymi informacjami o nich. Jedna z opcji, "name" (Kod 19) to wspomniany <*ROZMIAR*>.

```
16.
    {
17.
         "maxDataDiskCount": 16,
18.
         "memoryInMb": 28672,
19.
         "name": "Standard DS12 v2 Promo",
20.
         "numberOfCores": 4,
         "osDiskSizeInMb": 1047552,
21.
         "resourceDiskSizeInMb": 57344
22.
23.
       },
```

Po utworzeniu maszyny wirtualnej, będącej węzłem głównym, można przystąpić do konfiguracji systemu operacyjnego, instalacji niezbędnych aplikacji, określenia reguł systemowych oraz sieciowych. Pozwoli to na poprawne funkcjonowanie maszyny wirtualnej, jako usługi.

Istotne jest to, że obraz, który został wykorzystany do stworzenia maszyny, jest oznaczony jako "unmanaged disk". W przypadku takiego rozwiązania dysk będzie w pełni zarządzany przez użytkownika. Jest on przechowywany jako plik, którego parametry wydajności IOPS (ang. Input/Output operations Per Second), czy rozmiaru nie są w żaden sposób zarządzane przez ARM. Użytkownik kontroluje parametry niezbędne do poprawnej pracy i odpowiada za limity, jeśli takie zostaną dla obiektu dyskowego przekroczone.

## Węzeł roboczy (Worker Node)

Przygotowanie maszyny głównej do pracy w sieci rozproszonej jest jednym z etapów do przygotowania środowiska rozproszonego złożonego z wielu maszyn wirtualnych. Aby zapewnić współpracę maszyn ze sobą, niezbędne jest posiadanie maszyn roboczych, które będą realizować przesyłane zadania i zwracać wyniki w odpowiedniej postaci. Oprogramowanie, które realizuje takie zadania to wspomniany w poprzednim rozdziale Apache Hadoop lub HTCondor. Konfiguracja tych rozwiązań jest dostępna w dokumentacji na stronie projektów. Rozdział ten nie skupia się na zainstalowanym oprogramowaniu na każdej z maszyn wirtualnych, ale na konfiguracji do pracy bez względu na użyte oprogramowanie.

Aby móc przygotować wspomniany wcześniej VMSS, należy stworzyć maszynę wirtualną, która będzie realizowała zadania jako węzeł roboczy. Tym samym będzie to maszyna, która odbiera zadania z węzła głównego, utworzonego wcześniej, oraz przesyła wyniki z powrotem do niego.

Przygotowanie takiej maszyny jest wykonywane przez polecenie tworzące standardową maszynę wirtualną (Kod 2). Różnicą jest konfiguracja oprogramowania przeznaczonego do wykonywania obliczeń, które w takim systemie operacyjnym powinno zostać uprzednio zainstalowane.

Aby maszyny mogły ze sobą poprawnie współpracować (relacja jeden do wielu), należy bezwzględnie skopiować klucz SSH z maszyny nadrzędnej na roboczy węzeł. Celem jest umożliwienie przesyłania zadań i poprawna komunikacja maszyny dystrybuującej zadania obliczeniowe do maszyn obliczeniowych.

Aby tego dokonać, należy wygenerować klucz współdzielony SSH na węźle nadrzędnym (Kod 24), a następnie skopiować go na węzeł roboczy. Nastąpi dopisanie węzła nadrzędnego do listy hostów zaufanych, czyli urządzeń dozwolonych do łączenia z maszyną, bez konieczności podawania każdorazowo loginu i hasła (Kod 25).

```
24. ssh-keygen -t rsa25. ssh-copy-id UŻYTKOWNIK@ADRES_IP
```

Po skonfigurowaniu systemu operacyjnego należy przygotować obraz systemu do pracy w środowisku skalowalnych maszyn wirtualnych. Rozwiązanie to jest przede wszystkim skuteczne ze względu na automatyzację środowiska. Nie trzeba kontrolować ręcznie nazwy systemu operacyjnego, adresu sieciowego, czy tworzyć i uruchamiać wielu maszyn wirtualnych jednocześnie.

Idea usługi VMSS (Rys. 60) polega na zdefiniowaniu obrazu systemu, którego zwielokrotnienie i uruchomienie nie zakłóci wzajemnej komunikacji. Usługi i procesy są w pełni automatyczne i komunikują się z węzłem nadrzędnym, a ustawienia lokalne maszyny są niezależne od funkcjonowania samego środowiska.



Rys. 60 - Schemat architektury VMSS w Microsoft Azure [82]

Aby móc przygotować obraz do pracy, należy skorzystać z procedury deprowizjonowania (ang. deprovisioning), czyli procedury pozbawienia maszyny wirtualnej informacji na temat konta użytkownika, danych sprzętowych, zasobów komputera. Obraz systemu tym samym będzie niezależny od środowiska na którym zostanie uruchomiony ponownie. Celem VMSS nie jest bowiem ograniczanie się co do wielkości maszyny, która może być zmieniana w zależności od potrzeb w dowolnym momencie. Tym samym obraz będzie neutralny środowiskowo.

W przypadku prowadzonych badań oraz wspomnianej powyżej konfiguracji, system operacyjny używany w środowisku obliczeniowym to Ubuntu Linux. Polecenie realizujące operację usuwania danych sprzętowych i lokalnych użytkownika to (Kod 26). Należy wykonać je wprost w linii poleceń systemu operacyjnego, a nie CLI w Microsoft Azure.

26. sudo waagent -deprovision+user -force

Należy następnie wylogować się z systemu i go wyłączyć całkowicie (dealokacja) poleceniem (Kod 27) wykonywanym ponownie w CLI Portalu Azure.

27. az vm deallocate --resource-group <NAZWA\_RG> --name <NAZWA\_VM>

Podane *<NAZWA\_RG>* to nazwa grupy zasobów, a *<NAZWA\_VM>* to nazwa konkretnej maszyny wirtualnej, której dotyczy wykonywane polecenie.

W przypadku systemu operacyjnego Windows uruchomionego na platformie Azure, takim narzędziem byłoby Sysprep w ramach usługi OOBE (ang. Out-Of-Box Experience) i automatycznym wyłączeniem maszyny (Kod 28)

```
28. cd %windir%\system32\sysprep
   sysprep /generalize /shutdown /oobe
```

Kolejnym krokiem jest przygotowanie obrazu systemu, gotowego do zainstalowania na dowolnej usłudze maszyn wirtualnych środowiska Microsoft Azure. Jest to proces ujednolicenia (ang. generalize), czyli uniezależnienia od platformy chmury obliczeniowej. Realizowane jest to przez polecenie (Kod 29).

29. az vm generalize -g <NAZWA RG> -n <NAZWA VM>

Aby stworzony i przygotowany obraz maszyny wirtualnej wykorzystać jako źródło obrazu systemu dla VMSS, należy przede wszystkim uzyskać lokalizację obrazu nośnika systemu w strukturze zasobów chmury obliczeniowej. Jest to plik VHD (ang. Virtual Hard Drive), który będzie wskazany jako źródło VMSS.

Informacje takie można otrzymać we właściwościach nośnika w Portalu (Rys. 61). Uzyskanie takiej informacji z CLI jest w praktyce niemożliwe, ponieważ URL jest generowany na podstawie GUID z kilku elementów generowanych losowo, w tym z miejsca przechowywania obrazu.



Rys. 61 - Adres źródłowy nośnika VHD z systemem operacyjnym maszyny wirtualnej

Do stworzenia VMSS, które będzie wykorzystywać uzyskany powyżej odnośnik do obrazu systemu należy wykonać polecenie (Kod 30)

```
30. az vmss create -n <NAZWA_VMSS> -l <REGION> -g <GRUPA> --instance-
count <LICZBA> --use-unmanaged-disk --image
"https://vhdstorage123456abcd.blob.core.windows.net/vhds/osdisk_123abc45
6def.vhd" --os-type linux --authentication-type ssh --admin-username
<UŻYTKOWNIK> --ssh-key-value "<KLUCZ SSH>"
```

Polecenie to również zawiera zmienne, które należy określić informacjami dla środowiska, w którym jest ono wykonywane. Są to:  $\langle NAZWA\_VMSS \rangle$  – nazwa dla zestawu maszyn VMSS,  $\langle REGION \rangle$  – oznacza wspomniany wcześniej region geograficzny,  $\langle GRUPA \rangle$  – nazwa grupy zasobów,  $\langle LICZBA \rangle$  – oznacza początkową wartość maszyn uruchomionych w ramach VMSS,  $\langle KLUCZ SSH \rangle$  – uzyskany wcześniej klucz SSH służący do komunikacji między maszynami z węzła nadrzędnego do węzłów roboczych.

Dość istotna jest tutaj zmienna *GRUPA*, którą należy dodatkowo wyjaśnić. Grupa zasobów to parametr polecenia, który musi być identyczny jak Grupa zasobów, w której będzie uruchomiony węzeł nadrzędny. Nie ma bowiem możliwości komunikacji między dwoma Grupami zasobów w infrastrukturze Microsoft Azure.

## Ograniczenia i ich rozwiązanie

Skonfigurowane w powyższy sposób maszyny wirtualne są gotowe do pracy. W przypadku prawdziwie wielkoskalowych rozwiązań konieczne jest skorzystanie z możliwości, jakie posiada chmura obliczeniowa, czyli skalowania. Rozwiązanie oparte o dyski niezarządzane uniemożliwia uruchomienie znacznej ilości maszyn wirtualnych. Limitowane są także inne zasoby, które w przypadku klastrów z liczbą przynajmniej setek maszyn mogą być ograniczeniem [83]. Nie chodzi tutaj o limity operacji na sekundę, czyli ogólnej wydajności systemu. Podobnie jest w przypadku możliwości konfiguracji parametrów maszyn, ponieważ te można zmienić. Ograniczeniem jest ilość maszyn jednocześnie uruchamianych w ramach jednego VMSS.

Chcąc mieć większą kontrolę nad zasobami i możliwość wykorzystania większych zasobów chmury obliczeniowej, niezbędne staje się skonfigurowanie usług do pełnego zarządzania przez ARM. Wykonane wcześniej polecenia utworzą działające maszyny wirtualne, jednak skalowanie ich, czy zarządzanie statusami wymuszają obowiązki obsługi ich przez użytkownika.

W celu uniknięcia tego ograniczenia należy skonfigurować utworzone w powyższy sposób maszyny wirtualne na takie, których dyski będą w pełni zarządzane przez infrastrukturę Azure. Konieczne jest tym samym skorzystanie z rozwiązań obecnie stosowanych, unikając określanych wśród usług jako wycofywane (ang. legacy).

W przypadku skonfigurowanych i działających już maszyn wirtualnych, dość łatwo można przekonwertować dyski niezarządzane na zarządzane (ang. managed) wprost z poziomu Portalu, ale także i CLI [84]. Jest to wygodne rozwiązanie, ponieważ dostępne jest dla osób mniej doświadczonych, ale także daje możliwość przejścia na nowoczesne rozwiązania. Co ciekawe, sugestia konwersji dysków utworzonych w wycofywanym formacie na nowszy, będzie wyświetlać się po wejściu we właściwości danej maszyny wirtualnej. Po wykonanej operacji, maszyna zostanie uruchomiona ponownie. Sama konwersja nie zmienia jednak

funkcjonowania systemu operacyjnego maszyny, ani jej dostępności, za wyjątkiem krótkiej migawki.

Create image	□ ×
* Name	
VM_Image	✓
* Subscription	
Microsoft Azure Sponsorship	~
* Resource group	
	~
Create new	
* Location	
West Europe	~
Zone resiliency ① On Off	
OS disk	
* OS type 🚯	
Windows Linux	
* Storage blob	
https:// .blob.core.windows.net/vhds/ .vhd 🗸	Browse
* Account type 🚯	
Standard HDD	~
* Host caching 🚯	
Read/write	$\checkmark$

Rys. 62 - Okno tworzenia obrazu maszyny wirtualnej z poziomu Portalu Azure

Aby rozwiązać wspomniany problem należy przygotowany wcześniej dysk skonwertować na zarządzany, a następnie stworzyć obraz maszyny wirtualnej z przekonwertowanego obrazu dysku. Nowy obraz maszyny wirtualnej będzie wówczas dostępny jako szablon (ang. template) na koncie Microsoft Azure użytkownika w Portalu (Rys. 62) oraz z poziomu linii komend.

Aby stworzyć maszynę z już zdealokowanej maszyny wirtualnej, która została także zdeprowizjonowana, należy wykonać polecenie (Kod 31) [85] [86].

31. az image create -g <GRUPA> -n <NAZWA\_OBRAZU> --source <ŹRÓDŁOWA\_VM>

Dla powyższego polecenia ważne są zmienne  $\langle NAZWA\_OBRAZU \rangle$  – oznaczające jak będzie nazywać się szablon dostępny z poziomu platformy Azure oraz  $\langle ZRODLOWA\_VM \rangle$  – oznaczająca nazwę maszyny wirtualnej z której będzie stworzony szablon.

Tym samym obraz maszyny wirtualnej jako szablon jest utworzony. Jednak stworzenie licznej grupy maszyn wewnątrz VMSS wymaga poprawnej komunikacji między nimi. Aby również to ograniczenie zostało zniesione, konieczne jest przygotowanie tak zwanego Load Balancera (LB). Jest to usługa, która rozdziela ruch sieciowy między maszyny wirtualne i kontroluje przepływ danych w sieci, zapewniając także dostępność maszyn w sieci.

Na samym początku należy stworzyć nowy LB, który będzie wykorzystany jako łącznik sieciowy w VMSS, zamiast domyślnego, tworzonego wraz z maszyną wirtualną. Brak zdefiniowania LB nakłada bowiem ograniczenia np. w liczbie adresów i maszyn obsługiwanych jednocześnie. Stworzenie LB realizuje polecenie (Kod 32).

```
32. az network lb create --resource-group <GRUPA> --name <NAZWA_LB> --
sku Standard
```

Oprócz nazwy LB, która jest dowolną nazwą przyjazną dla użytkownika, istotne jest określenie poprawnej Grupy zasobów. Podobnie, jak poprzednio, musi to być Grupa zasobów taka sama, jak Grupa zasobów wszystkich maszyn pracujących ze sobą w danej sieci (Kod 33).

```
33. az network lb inbound-nat-pool create --resource-group <GRUPA> --
    lb-name <NAZWA_LB> --name <NAZWA_PULI> --protocol Tcp --frontend-
    port-range-start <PORT_START> --frontend-port-range-end <PORT_KONIEC>
    --backend-port 80 --frontend-ip-name <NAZWA_IP >
```

Wykonanie polecenia tworzy zakres adresacji, który ma zostać wykorzystany jako liczba połączeń wchodzących i wychodzących dla wszystkich maszyn w sieci. Polecenie ponownie wykorzystuje zmienne, które należy poprawnie określić. Są to: *<NAZWA\_PULI>* – jako nazwa dla tej puli adresów sieciowych dostępnych dla LB, *<PORT\_START>* oraz odpowiednio *<PORT\_KONIEC>* – to numery portu początkowego oraz końcowego dla którego maszyny będą się między sobą komunikować. Ostatnią zmienną jest *<NAZWA\_IP>*, jako nazwa portu komunikacji z Portalem.

Numery portów są dowolną liczbą (zalecane 50 000 i wyżej), której zakres (start – koniec) jest przynajmniej dwukrotnie większy niż liczba maszyn, jaka ma zostać uruchomiona wewnątrz VMSS. W przypadku 150 maszyn, pula adresów powinna być większa lub równa 300.

Ograniczenie liczby maszyn wirtualnych dla dysków niezarządzanych to 20. Dla maszyn opartych o dyski zarządzane limit ten wynosi obecnie do 600 (do niedawna 100) maszyn wirtualnych [87]. Samych VMSS może być maksymalnie 2000 w danym regionie, co łącznie oznacza liczbę nawet 12000 maszyn wirtualnych w całej subskrypcji użytkownika na dany region. Skalowanie jest tym samym możliwe znacznie większe, uwzględniając pozostałe parametry maszyn wirtualnych.

Ostatnim poleceniem, jakie należy wykonać jest stworzenie nowego VMSS, który nie jest już ograniczony liczbą 20 maszyn uruchomionych jednocześnie, przez polecenie (Kod 34).

```
34. az vmss create --name <NAZWA_VMSS> --location <REGION> --resource-
group <GRUPA> --load-balancer <NAZWA_LB> --instance-count <LICZBA_VM>
--image <NAZWA_OBRAZU> --authentication-type ssh --admin-username
<UŻYTKOWNIK> --ssh-key-value "<KLUCZ SSH>"
```

Polecenie to korzysta ze wszystkich wcześniej utworzonych obiektów – LB, szablonu obrazu maszyny wirtualnej, czy klucza SSH. Jedna zmienna w poleceniu jest ważna, aby VMSS stworzył się bez kolejnego graniczenia, które limituje liczbę maszyn wirtualnych do 100. Takie ograniczenie może pojawić w przypadku automatyzacji ARM. Należy wówczas po stronie Azure wpisać odpowiednią liczbę w zmiennej *<LICZBA\_VM>*. Jeżeli tworzenie VMSS będzie połączone z wartością mniejszą niż 100, wówczas maksymalną wartością uruchomionych maszyn będzie liczba 100. Wpisanie wartości większej, nawet 101 automatycznie zdejmuje ograniczenie i pozwala na ominięcie wspomnianego limitu. W takim przypadku zostanie utworzony VMSS, którego wartość dostępna do skalowania będzie równa 1000.

Dość rozbudowany sposób postępowania, jednak pozwalający obejść domyślne limity, jakie nakłada na użytkownika platforma Microsoft Azure. Limity te nie są spowodowane celowym ograniczaniem możliwości, czy funkcjonalności, ale bezpieczeństwem. Bez limitów, uruchomienie kilkudziesięciu czy nawet kilkuset maszyn wirtualnych mogłoby narazić posiadacza subskrypcji na niekontrolowany wydatek. Z tego też powodu jakiekolwiek zmiany w limitach usług są przydzielane indywidualnie na żądanie użytkownika przez wsparcie techniczne. Dotyczy to każdej usługi, jaka jest dostępna w ramach Microsoft Azure.

Jak widać możliwości chmury obliczeniowej pokazują, że pojęcie wielkoskalowości nie jest tutaj wymyślone, ale całkowicie realne, a w dodatku dostępne za pomocą kilku poleceń lub kliknięć.