**Nazwa przedmiotu:**

Wirtualne środowiska obliczeniowe

**Koordynator przedmiotu:**

Rajmund Kożuszek

**Status przedmiotu:**

Fakultatywny dowolnego wyboru

**Poziom kształcenia:**

Studia II stopnia

**Program:**

Informatyka

**Grupa przedmiotów:**

Przedmioty techniczne - zaawansowane

**Kod przedmiotu:**

WSO

**Semestr nominalny:**

3 / rok ak. 2021/2022

**Liczba punktów ECTS:**

4

**Liczba godzin pracy studenta związanych z osiągnięciem efektów uczenia się:**

1. liczba godzin kontaktowych – 45 godz., w tym
obecność na wykładach 30 godz.,
spotkania projektowe 15 godz.
2. praca własna studenta – 70 godz., w tym
przygotowanie do kolokwium 20 godz.,
realizacja projektu 50 godz.
Łączny nakład pracy studenta wynosi 115 godz., co odpowiada 4 pkt. ECTS.

**Liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich:**

1,56 pkt. ECTS, co odpowiada 45 godz. kontaktowym

**Język prowadzenia zajęć:**

polski

**Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym:**

1,74 pkt. ECTS, co odpowiada 50 godz. poświęconym na przygotowanie projektu

**Formy zajęć i ich wymiar w semestrze:**

|  |  |
| --- | --- |
| Wykład: | 30h |
| Ćwiczenia: | 0h |
| Laboratorium: | 0h |
| Projekt: | 15h |
| Lekcje komputerowe: | 0h |

**Wymagania wstępne:**

Nie określa się wymagań odnośnie zaliczenia przedmiotów poprzedzających, jednakże zalecane jest aby studenci dysponowali pewną wiedzą na temat systemów operacyjnych i sieci komputerowych czemu sprzyja wcześniejsze uczestnictwo w takich przedmiotach jak Systemy Operacyjne, Administrowanie Systemem Unix czy Sieci Komputerowe.

**Limit liczby studentów:**

30

**Cel przedmiotu:**

Celem przedmiotu jest wprowadzenie do projektowania wszystkich elementów nowoczesnych środowisk obliczeniowych w tym sieci łączącej maszyny i architektur przechowywania danych a także przybliżenie zagadnień wymiarowania zasobów czy alokacji i szeregowania zadań.

**Treści kształcenia:**

Wykład:
Na treść wykładów składają się następujące bloki tematyczne:
• Wprowadzenie do zagadnień związanych z wirtualizacją i konteneryzacją, przykłady typowych architektur klastrowych i chmurowych, systemy operacyjne, przykłady zastosowań.
• Wirtualizacja: rodzaje wirtualizacji, zasada działania, tworzenie obrazów systemów, standardy, typowe operacje: migracja, klonowanie, metody przydzielania zasobów: thin provisioning.
• Omówienie podstawowych środowisk wirtualizacyjnych: KVM, VMWare, Xen – porównanie architektur i możliwości, przykłady zastosowania.
• Zagadnienia związane z przygotowaniem pamięci dyskowej – technologie: RAID, JBOD, LVM, macierze dyskowe, rozproszone systemy plików – zasady wykorzystania i projektowania, technologie transmisji danych, przechowywanie danych blokowych i plikowych, przykłady wykorzystania.
• Organizacja sieci dla klastrów i chmur: przegląd technologii sieciowych (Ethernet 1/10/100Gb, InfiniBand, sieci dla pamięci masowych), topologie sieci w ramach klastra – metody zapewnienia wysokiej dostępności, wydajności i skalowalności. Wykorzystanie redundancji łącz i urządzeń sieciowych, przegląd typowych topologie sieci.
• Sieci wirtualne w ramach pojedynczego gospodarza wirtualizacji - zarządzanie w warstwie 2 ISO/OSI. Wykorzystanie protokołów: 802.1Q (sieci wirtualne VLAN), 802.1ad (Q-in-Q). Wykorzystanie możliwości warstwy 3 ISO/OSI: ruting statyczny i dynamiczny, polityki rutingowe, kształtowanie i filtrowanie ruchu - elementy bezpieczeństwa. Porównanie technologii dla KVM i VMWare.
• Budowa sieci dla klastra, wykorzystanie narzędzi dla uproszczenia zarządzania i automatyzacji provisioningu. Sieci definiowane programowo (SDN) - wykorzystanie w dużych klastrach – przegląd jakiś rozwiązań: OpenFlow, VMWare NSX.
• Zagnieżdżona wirtualizacja: przykłady, problemy: ograniczenia w budowie sieci wirtualnych.
• Systemy zarządzania klastrami i chmurami: OpenStack, oVirt.
• Konteneryzacja: zasada, przykłady rozwiązań: Docker, Kubernetes.
• Modelowanie środowisk klastrowych i chmurowych, wybrane metody optymalizacji przydziału zasobów i alokacji zadań. Równoważenie obciążeń i szeregowanie zadań jako metody bezpośredniego sterowania wydajnością.
• Symulacja środowisk klastrowych i chmurowych. Symulator Cloudsim.
• Zagadnienie związane z oszczędnością energii: sterowanie wydajnością na poziomie procesora, systemu operacyjnego i klastra. Przykłady definicji i metod rozwiązania zadań. Uwzględnienie zużycia energii przez sieć. Przegląd technologii i standardów związanych z energooszczędnością: ACPI, 802.3az, energooszczędne sterowniki procesora w Linuksie, energooszczędne topologie sieci.
Projekt:
Projekt polega na zaprojektowaniu i implementacji w postaci prototypu o ograniczonych możliwościach lub modułu symulatora wybranych fragmentów typowego systemu wykorzystującego architekturę klastra bądź chmury. W realizacji projektu mogą być wykorzystane elementy rozwiązań rozwijanych przez zespół autorski w ramach prowadzonych projektów badawczych takie jak np. energooszczędne algorytmy przydziału zasobów. Przykładowe tematy projektu:
Projekt 1: Zaprojektuj architekturę klastra dla systemu przetwarzającego dane z dużego zestawu czujników. Architektura powinna uwzględniać możliwość wydajnego zapisywania i udostępniania danych do dalszej obróbki, oraz wysoką skalowalność. Sprawdź podstawowe założenia architektury wdrażając kluczowe elementy w ograniczonym środowisku (pojedynczy komputer z wirtualiztorem).
Projekt 2: Zaprojektuj architekturę małego klastra umożliwiającego przetwarzanie zadań obliczeniowych w ramach pewnej organizacji. Zadbaj o wysoki poziom bezpieczeństwa i niskie koszty operacyjne. Sprawdź podstawowe założenia architektury wdrażając kluczowe elementy w ograniczonym środowisku (pojedynczy komputer z wirtualiztorem).
Projekt 3: Zaimplementuj wskazany algorytm szeregowania zadań w symulatorze CloudSim.
Projekt 4: Zaimplementuj wskazany algorytm alokacji zasobów w symulatorze CloudSim.

**Metody oceny:**

Uzupełnieniem wykładu jest projekt, w ramach którego studenci wykonują zadanie związane z projektowaniem i implementacją wybranych elementów prezentowanych architektur. Jest on realizowany w zespołach trzyosobowych i składa się z następujących etapów: Projekt wstępny (2tyg.), weryfikacja koncepcji (2. tyg), implementacja i testowanie (4 tygodnie). W pierwszym etapie studenci przygotowują wstępny projekt, który jest przedstawiany w formie pisemnej do akceptacji prowadzących. Pozwala to, w następnym etapie, wprowadzić poprawki i przygotować ostateczny projekt implementowany w trzeciej fazie. Wynikiem implementacji jest, oprócz działającego kodu, raport z testów zawierający wnioski dotyczące badanego rozwiązania.

**Egzamin:**

nie

**Literatura:**

Książki:
• Dac-Nhuong Le, Raghvendra Kumar, Gia Nhu Nguyen, Jyotir Moy Chatterjee, Cloud Computing and Virtualization, John Wiley & Sons, 2018
• Ken Gray, Thomas D. Nadeau, Cloud Computing and Virtualization, Morgan Kaufmann, 2016
• Edouard Bugnion, Jason Nieh, Dan Tsafrir, Hardware and Software Support for Virtualization (Synthesis Lectures on Computer Architecture), Morgan & Claypool Publishers, 2017
Dokumentacja pakietów i przewodniki internetowe:
• Xen: https://wiki.xenproject.org/wiki/Main\_Page
• VMWare: https://www.vmware.com/support/pubs/
• KVM: https://www.linux-kvm.org/page/Documents
• Docker: https://docs.docker.com/get-started/

**Witryna www przedmiotu:**

https://usosweb.usos.pw.edu.pl/kontroler.php?\_action=katalog2/przedmioty/pokazPrzedmiot&prz\_kod=103A-INISY-MSP-WSO

**Uwagi:**

(-)

## Charakterystyki przedmiotowe