



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Systemy wysokiej niezawodności [S2Inf1-SRC>SWN]

Przedmiot

Kierunek studiów
Informatyka

Rok/Semestr
1/2

Studia w zakresie (specjalność)
Systemy rozproszone i chmurowe

Profil studiów
ogólnoakademicki

Poziom studiów
drugiego stopnia

Język oferowanego przedmiotu
polski

Forma studiów
stacjonarne

Wymagalność
obligatoryjny

Liczba godzin

Wykład
30

Laboratorium
30

Inne
0

Ćwiczenia
0

Projekty/seminaria
0

Liczba punktów ECTS

5,00

Koordynatorzy

dr inż. Michał Szychowiak prof. PP
michal.szychowiak@put.poznan.pl

Wykładowcy

Wymagania wstępne

Student rozpoczynający ten przedmiot powinien posiadać podstawową wiedzę z systemów operacyjnych, sieci komputerowych, baz danych oraz bezpieczeństwa systemów informatycznych. Student powinien posiadać umiejętność pozyskiwania informacji ze wskazanych źródeł. Powinien również rozumieć konieczność poszerzania swoich kompetencji. W zakresie kompetencji społecznych student musi prezentować takie postawy jak uczciwość, odpowiedzialność, wytrwałość, ciekawość poznawcza, kreatywność, kultura osobista, szacunek dla innych ludzi.

Cel przedmiotu

1. Przekazanie studentom podstawowej wiedzy z dziedziny niezawodności przetwarzania rozproszonego, rozproszonej detekcji uszkodzeń, tolerowania awarii, wysokiej dostępności, replikacji procesów i odtwarzania stanu przetwarzania rozproszonego. 2. Rozwijanie u studentów umiejętności rozwiązywania problemów niezawodnego przetwarzania w zawodnym środowisku rozproszonym. 3. Zapoznanie studentów z aktualnie dostępnymi technologiami wysokiej dostępności systemów klastrowych i przetwarzania w chmurze.

Przedmiotowe efekty uczenia się

Wiedza:

1. ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie algorytmów i złożoności, w szczególności algorytmów rozproszonych o podwyższonej niezawodności - [k2st_w2]
2. ma zaawansowaną szczegółową wiedzę związaną z wybranymi zagadnieniami z zakresu informatyki, takimi jak: niezawodność przetwarzania, wysoka dostępność, rozproszone detektory uszkodzeń, tolerowanie uszkodzeń, odtwarzanie stanu przetwarzania, replikacja procesów, uzgadnianie, consensus, komunikacja grupowa, samostabilizacja itp. - [k2st_w3]
3. ma podstawową wiedzę o cyklu życia systemów informatycznych, z uwzględnieniem aspektów niezawodności - [k2st_w5]
4. zna podstawowe metody, techniki i narzędzia stosowane przy rozwiązywaniu złożonych problemów przetwarzania rozproszonego o podwyższonej niezawodności - [k2st_w6]

Umiejętności:

1. potrafi pozyskiwać informacje z literatury, baz danych oraz innych źródeł (w języku ojczystym i angielskim), integrować je, dokonywać ich interpretacji i krytycznej oceny, wyciągać wnioski oraz formułować i wyczerpująco uzasadniać opinie, w kontekście wiarygodności przetwarzania - [k2st_u1]
2. potrafi wykorzystać do formułowania i rozwiązywania problemów przetwarzania rozproszonego o podwyższonej niezawodności metody analityczne, symulacyjne oraz eksperymentalne - [k2st_u4]
3. potrafi – przy formułowaniu i rozwiązywaniu problemów przetwarzania rozproszonego o podwyższonej niezawodności – integrować wiedzę z różnych obszarów informatyki (m.in. programowanie obiektowe, przetwarzanie rozproszone, systemy operacyjne, systemy baz danych, złożoność obliczeniowa i komunikacyjna) - [k2st_u5]
4. potrafi zaproponować ulepszenia (usprawnienia) istniejących rozwiązań technicznych w kontekście wiarygodności przetwarzania - [k2st_u8]
5. potrafi ocenić przydatność metod i narzędzi służących do rozwiązania zadania inżynierskiego, polegającego na budowie lub ocenie systemu informatycznego lub jego składowych, w tym dostrzec ograniczenia tych metod i narzędzi - [k2st_u9]
6. potrafi – stosując m.in. koncepcyjnie nowe metody – rozwiązywać wybrane problemy niezawodności systemu informatycznego - [k2st_u10]

Kompetencje społeczne:

1. rozumie znaczenie wykorzystywania najnowszej wiedzy z zakresu informatyki w rozwiązywaniu problemów niezawodności systemu informatycznego - [k2st_k2]
2. rozumie znaczenie działalności popularyzatorskiej dotyczącej najnowszych osiągnięć z zakresu informatyki - [k2st_k3]

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Efekty kształcenia przedstawione wyżej weryfikowane są w następujący sposób:

Ocena formująca:

a) w zakresie wykładów:

- na podstawie odpowiedzi na pytania dotyczące materiału omówionego na poprzednich wykładach,

b) w zakresie laboratoriów / ćwiczeń:

- na podstawie oceny bieżącego postępu realizacji zadań,

Ocena podsumowująca:

a) w zakresie wykładów weryfikowanie założonych efektów kształcenia realizowane jest przez:

- ocenę wiedzy i umiejętności wykazanych na egzaminie w postaci testu o charakterze problemowym (4-5 pytań otwartych, do zdobycia 12 pkt. zaliczenie wymaga uzyskania min. 6 pkt.),
- omówienie wyników egzaminu,

b) w zakresie laboratoriów weryfikowanie założonych efektów kształcenia realizowane jest przez:

- ocenę przygotowania studenta do poszczególnych sesji zajęć laboratoryjnych (sprawdzian "wejściowy") oraz ocenę umiejętności związanych z realizacją ćwiczeń laboratoryjnych,

Uzyskiwanie punktów dodatkowych za aktywność podczas zajęć, a szczególnie za:

- przygotowanie i przedstawienie na forum grupy analizy i projektu algorytmicznego rozwiązania zadanego problemu,
- omówienia dodatkowych aspektów zagadnienia,

- efektywność zastosowania zdobytej na wykładzie wiedzy podczas rozwiązywania zadanego problemu,
- wskazywanie trudności percepcyjnych studentów umożliwiające bieżące doskonalenia procesu dydaktycznego.

Treści programowe

Program przedmiotu obejmuje następujące zagadnienia:

- Podstawowe problemy niezawodności i dostępności.
- Odtwarzanie stanu lokalnego i rozproszonego przetwarzania.
- Niezawodne zatwierdzanie transakcji.
- Techniki replikacji danych i procesów w środowiskach zawodnych.
- Podstawowe problemy uzgadniania rozproszonego.
- Samo-stabilizacja.

Tematyka zajęć

Program wykładu obejmuje następujące zagadnienia szczegółowe:

Podstawowa klasyfikacja problemów i środków osiągnięcia podwyższonej niezawodności i dostępności. Modele awarii, miary niezawodności. Systemy HAC (ang. High-Availability Clusters). Metody odtwarzania postępowego i wstecznego. Problematyka spójności stanów kanałów komunikacyjnych. Odtwarzanie stanu lokalnego i rozproszonego przetwarzania. Skoordynowane (synchroniczne) tworzenie punktów kontrolnych: metodyka i przykładowe algorytmy. Niezależne (asynchroniczne) tworzenie punktów kontrolnych: metodyka i przykładowe algorytmy. Rejestrowanie wiadomości optymistyczne i pesymistyczne. Hybrydowe algorytmy odtwarzania stanu, odtwarzanie adaptacyjne i quasi-synchroniczne. Niezawodne zatwierdzanie transakcji (2PC i 3PC) oraz jego ograniczenia w środowisku rozproszonym. Techniki replikacji danych i procesów w środowiskach zawodnych, podejścia pesymistyczne i optymistyczne. Zastosowanie replikacji aktywnej i pasywnej procesów w tolerowaniu uszkodzeń. Mechanizmy komunikacji grupowej wymagane do realizacji replikacji (RBCast, TOcast, UBCast, VScast). Podstawowe problemy uzgadniania rozproszonego, ich wzajemne relacje i ograniczenia rozwiązywalności. Algorytmy rozwiązania rozproszonego konsensusu dla różnych modeli błędów. Rozwiązania problemów uzgadniania z błędami bizantyjskimi. Zastosowanie mechanizmów kryptograficznych w bizantyjskim uzgadnianiu. Rozwiązywanie problemów uzgadniania przy pomocy detektorów uszkodzeń. Algorytmy samo-stabilizujące się.

Program laboratorium obejmuje następujące zagadnienia szczegółowe:

Problematyka awarii procesów i kanałów komunikacyjnych w wybranych problemach przetwarzania rozproszonego. Konfiguracja i weryfikacja eksperymentalna pracy wybranych reprezentantów technologii HAC (ang. High-Availability Clusters): ClusterIP, Linux Virtual Server, Pacemaker, DRBD. Konstrukcja wybranego algorytmu tolerującego awarie komunikacyjne dla wybranej realizacji rozproszonego przetwarzania (wzajemne wykluczanie) i dla różnych typów kanałów komunikacyjnych (kanały FIFO i nonFIFO). Zastosowanie podstawowych technik odtwarzania stanu przetwarzania rozproszonego w praktyce (odtworzenie synchroniczne i asynchroniczne). Zastosowanie mechanizmów rejestrujących wiadomości w odtwarzaniu stanu przetwarzania i optymalizacja ich efektywności. Zastosowanie samo-stabilizujących się rozwiązań wybranych problemów w sieciach komputerowych (problemy trasowania i orientacji grafu topologii).

Metody dydaktyczne

1. wykład: prezentacja multimedialna, pokaz multimedialny, demonstracja.
2. ćwiczenia laboratoryjne: demonstracja, dyskusja, warsztaty, ćwiczenia praktyczne, praca w zespole.

Literatura

Podstawowa

1. Klaus Schmidt, "High Availability and Disaster Recovery: Concepts, Design, Implementation", Springer, 2006
2. Kenneth P. Birman, "Guide to Reliable Distributed Systems", Springer, 2012
3. Floyd Piedad, Michael Hawkins, "High availability: design, techniques, and processes", Prentice Hall, 2001
4. Michel Raynal, "Distributed algorithms for message-passing systems", Springer, 2013

Uzupełniająca

1. Ajay D. Kshemkalyani, Mukesh Singhal, "Distributed Computing: Principles, Algorithms, and Systems", Cambridge University Press, 2008

2. Sander van Vugt, "Pro Linux High Availability Clustering", Apress, 2014
3. Daniel J. Sorin, "Fault Tolerant Computer Architecture", Synthesis Lectures on Computer Architecture No.5, 2009
4. Jerzy Brzeziński, Michał Szychowiak, "Self-Stabilization in Distributed Systems – a Short Survey", Foundations of Computing and Decision Sciences, vol. 25, no. 1, 2000
5. A.A. Helal, A.A. Heddaya, B.B. Bhargava, "Replication Techniques in Distributed Systems", Kluwer Academic Publishers, 1996

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	125	5,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	60	2,50
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwίων/egzaminu, wykonanie projektu)	65	2,50