Systemy odporne na błędy sprawozdanie z realizacji projektu

Maciej Bandura, Marcin Ślusarczyk; 2ID21B

23 marca 2025

1. Wstęp

Wykonywany temat: System składający się z 8 serwerów połączonych w graf. Serwer nadzorujący wysyła 16-bitową informację zabezpieczoną kodem korekcyjnym Hamminga, z możliwością symulowania błędów podczas transmisji danych.

1.1 Założenia projektowe oraz funkcjonalności

- System składa się z 8 serwerów połączonych w graf, umożliwiający komunikację między nimi.
- Jeden z serwerów pełni funkcję serwera nadzorującego, który zarządza przesyłaniem danych i kontrolą poprawności transmisji.
- Topologia grafu może być dowolna, np. pełne połączenie, pierścień, drzewo, siatka częściowa - struktura będzie definiowana przez użytkownika za pomocą pliku XML
- Serwer nadzorujący przesyła 16-bitowe bloki danych, które są zabezpieczone kodem korekcyjnym Hamminga
- Kody Hamminga pozwalają na korekcję pojedynczych błędów i detekcję podwójnych błędów, co zwiększa niezawodność systemu.
- Każdy serwer odbierający musi być wyposażony w mechanizm dekodowania i sprawdzania syndromu błędu. Po otrzymaniu danych, wykonuje dekodowanie Hamminga, aby sprawdzić poprawność transmisji. W przypadku wykrycia pojedynczego błędu, serwer koryguje go samodzielnie.
- Symulacja błędów jest realizowana w sposób losowy.
- Serwer odbierający raportuje raportuje wystąpienie błędu oraz skuteczność jego korekcji.
- System umożliwia rozszerzenie liczby serwerów, zachowując poprawność kodowania i weryfikacji błędów.

1.2 Wstęp teoretyczny

Celem realizowanego projektu jest stworzenie systemu transmisji danych z ośmioma serwerami połączonymi w grafie, w którym komunikacja odbywa się za pomocą 16-bitowych informacji zabezpieczonych kodem korekcyjnym Hamminga. Aby zapewnić niezawodność transmisji danych, w systemie wykorzystany zostanie kod **Hamminga z dodatkowym bitem parzystości SECDED** (Single Error Correction, Double Error Detection), który zapewni zarówno wykrywanie, jak i korekcję pojedynczych błędów, a także detekcję podwójnych błędów.

Kod Hamminga jest jedną z najczęściej stosowanych metod **wykrywania i korekcji błędów** w systemach komunikacyjnych i komputerowych. Jego główną zaletą jest

zdolność do wykrywania i naprawiania pojedynczych błędów w danych bez konieczności ich retransmisji. Działa to na zasadzie dodania do danych dodatkowych bitów parzystości, które są rozmieszczane w odpowiednich miejscach w danych, umożliwiając w ten sposób identyfikację, który bit uległ zmianie w wyniku błędu transmisji. Dzięki temu, w przypadku wykrycia błędu w jednym z bitów, system jest w stanie go skorygować i przywrócić poprawność danych.

Jednak klasyczny kod Hamminga, choć skuteczny w przypadku wykrywania i korekcji pojedynczych błędów, nie radzi sobie z wykrywaniem podwójnych błędów. W sytuacjach, gdy w przesyłanych danych wystąpią **dwa błędy**, klasyczny kod Hamminga **nie jest w stanie ich wykryć**, co może prowadzić do nieprawidłowej korekcji i w konsekwencji do błędnych danych. Aby rozwiązać ten problem, wprowadzono rozszerzenie kodu Hamminga, zwane SECDED. Jest to modyfikacja klasycznego kodu Hamminga, która dodaje **dodatkowy bit parzystości**, umożliwiający detekcję dwóch błędów jednocześnie, a jednocześnie pozwala na korekcję pojedynczych błędów. Dzięki temu, kod SECDED zapewnia wyższą niezawodność transmisji, umożliwiając wykrywanie błędów, które mogą wystąpić w wyniku zakłóceń w sieci.

W omawianym projekcie, serwer nadzorujący będzie odpowiedzialny za wysyłanie 16-bitowych bloków danych, które będą zabezpieczone tym kodem, oraz za symulowanie błędów w transmisji w celu testowania skuteczności tego mechanizmu korekcji. Symulowanie błędów w transmisji pozwala na przeprowadzenie testów w różnych warunkach, sprawdzając, jak system radzi sobie z błędami pojedynczymi oraz podwójnymi, a także w jaki sposób reaguje na ewentualne zakłócenia w sieci.

1.3 Wybór technologii z stosownym uzasadnieniem

W projekcie zastosowane zostaną technologie **Docker** oraz **PHP** w celu zapewnienia wysokiej niezawodności, elastyczności oraz łatwego zarządzania serwerami i aplikacjami.

1.3.1 Docker

Docker to narzędzie do konteneryzacji, które pozwala na uruchamianie aplikacji w odizolowanych środowiskach zwanych kontenerami. Dzięki temu możliwe jest stworzenie spójnego środowiska pracy, niezależnie od infrastruktury serwera. Docker pozwala na łatwe tworzenie, wdrażanie i skalowanie aplikacji w różnych środowiskach.

Zalety użycia Dockera w tym projekcie:

- **Izolacja środowisk** – Każdy serwer w systemie może być uruchomiony w osobnym kontenerze, co zapewnia pełną izolację aplikacji i minimalizuje ryzyko konfliktów między zależnościami.
- **Łatwość skalowania** – Docker umożliwia szybkie tworzenie nowych instancji serwerów (kontenerów), co jest istotne, jeśli w przyszłości projekt będzie

musiał obsługiwać większą liczbę serwerów w grafie lub innych rozproszonych systemach.

- **Niezależność od systemu operacyjnego** – Docker zapewnia, że aplikacja będzie działać w ten sam sposób na różnych systemach operacyjnych, co znacząco upraszcza wdrożenia i zarządzanie infrastrukturą.

Wybór technologii **Docker** i **PHP** jest uzasadniony potrzebą stworzenia skalowalnego, niezawodnego i elastycznego systemu, który będzie łatwy do wdrożenia, zarządzania i testowania. PHP, dzięki swojej elastyczności i wsparciu dla różnych bibliotek, jest odpowiednim wyborem do budowy aplikacji backendowej obsługującej logikę systemu i interakcję pomiędzy serwerami. Te technologie zapewnią stabilność, skalowalność oraz łatwość zarządzania systemem.