

Politechnika Warszawska

W Y D Z I A Ł M A T E M A T Y K I
I N A U K I N F O R M A C Y J N Y C H



Praca dyplomowa inżynierska

na kierunku Informatyka i Systemy Informacyjne

System do zdalnej pracy w środowisku graficznym wykorzystujący
maszyny wirtualne QEMU z akceleracją sprzętową

Krzysztof Smogór

Numer albumu 298906

Piotr Widomski

Numer albumu 298919

promotor

dr inż. Marek Kozłowski

WARSZAWA 2022

.....

podpis promotora

.....

podpis autora

Streszczenie

System do zdalnej pracy w środowisku graficznym wykorzystujący maszyny wirtualne
QEMU z akceleracją sprzętową

Streszczam.

Lorem ipsum dolor sit amet, consetetur sadipscing elit, sed diam nonumyeirmod
tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos
et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata
sanctus est Lorem ipsum dolor sit amet.

Słowa kluczowe: slowo1, slowo2, ...

Abstract

Environment for remote work with Graphical User Interface using QEMU virtual machines with hardware acceleration

Konieczne jest załączenie wypełnionego oświadczenia o autorstwie pracy. By tego dokonać, skan (w formacie PDF) należy umieścić w folderze *scans* i nazwać go, np. `oswiadczenie_o_autorstwie_pracy.pdf` (w przypadku innej nazwy lub umieszczenia w innym folderze, konieczne jest adekwatne zmodyfikowanie ścieżki w komendzie je załączającej — patrz fragment kodu OŚWIADCZENIA).

Keywords: keyword1, keyword2, ...

Załącznik nr 1 do zarządzenia nr 109 /2021

Rektora PW z dnia 9 listopada 2021 r.

załącznik nr 5 do zarządzenia nr 42 /2020 Rektora PW



Politechnika Warszawska

.....
miejscowość i data

.....
imię i nazwisko studenta

.....
numer albumu

.....
kierunek studiów

OŚWIADCZENIE

Świadomy/-a odpowiedzialności karnej za składanie fałszywych zeznań oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie, pod opieką kierującego pracą dyplomową.

Jednocześnie oświadczam, że:

- niniejsza praca dyplomowa nie narusza praw autorskich w rozumieniu ustawy z dnia 4 lutego 1994 roku o prawie autorskim i prawach pokrewnych (Dz.U. z 2021 r., poz. 1062) oraz dóbr osobistych chronionych prawem cywilnym,
- niniejsza praca dyplomowa nie zawiera danych i informacji, które uzyskałem/-am w sposób niedozwolony,
- niniejsza praca dyplomowa nie była wcześniej podstawą żadnej innej urzędowej procedury związanej z nadawaniem dyplomów lub tytułów zawodowych,
- wszystkie informacje umieszczone w niniejszej pracy, uzyskane ze źródeł pisanych i elektronicznych, zostały udokumentowane w wykazie literatury odpowiednimi odnośnikami,
- znam regulacje prawne Politechniki Warszawskiej w sprawie zarządzania prawami autorskimi i prawami pokrewnymi, prawami własności przemysłowej oraz zasadami komercjalizacji.

.....
czytelny podpis studenta

Załącznik nr 3 do zarządzenia nr 109 /2021

Rektora PW z dnia 9 listopada 2021 r.

załącznik nr 9 do zarządzenia nr 42 /2020 Rektora PW



Politechnika Warszawska

.....
miejsowość i data

.....
imię i nazwisko studenta

.....
numer albumu

.....
Wydział i kierunek studiów

Oświadczenie studenta w przedmiocie udzielenia licencji
Politechnice Warszawskiej

Oświadczam, że jako autor/współautor* pracy dyplomowej pt.
..... udzielam/nie udzielam* Politechnice Warszawskiej
nieodpłatnej licencji na niewyłączne, nieograniczone w czasie, umieszczenie pracy dyplomowej w elek-
tronicznych bazach danych oraz udostępnianie pracy dyplomowej w zamkniętym systemie bibliotecznym
Politechniki Warszawskiej osobom zainteresowanym.

Licencja na udostępnienie pracy dyplomowej nie obejmuje wyrażenia zgody na wykorzystywanie pracy
dyplomowej na żadnym innym polu eksploatacji, w szczególności kopiowania pracy dyplomowej w całości
lub w części, utrwalania w innej formie czy zwielokrotniania.

.....
czytelny podpis studenta

* niepotrzebne skreślić

Spis treści

1. Wstęp	11
1.1. Opis problemu	11
1.2. Podobne rozwiązania	11
1.3. Wizja systemu	11
1.4. Istotne pojęcia	12
1.5. Wymaganie funkcjonalne	14
1.5.1. Nadzorca	14
1.5.2. Serwer wirtualizacji	16
1.5.3. Panel administratora	18
1.6. Wymaganie niefunkcjonalne	20
1.7. Analiza ryzyka	21
1.7.1. Omówienie zagrożeń	21
1.8. Podział pracy	23
2. Opis rozwiązania	24
2.1. Architektura systemu	24
2.1.1. Nadzorca	25
2.1.2. Serwer wirtualizacji	26
2.1.3. Aplikacja kliencka	26
2.1.4. Panel administratora	26
2.1.5. Broker wiadomości	27
2.2. Zewnętrzne narzędzia	27
2.2.1. Ansible	27
2.2.2. Vagrant	28
2.2.3. Libvirt z QEMU	28
2.3. Technologie	28
2.4. Stany biznesowe	30
2.5. Procesy biznesowe	30

2.6.	Komunikacja	30
2.6.1.	Komunikacja wewnętrzna	30
2.6.2.	Komunikacja zewnętrzna	30
2.6.3.	Sekwencje komunikacji	30
3.	Analiza rozwiązania	31
4.	Podsumowanie	32

1. Wstęp

1.1. Opis problemu

Można aktualnie zaobserwować dużą zmianę w rynku pracy. Z powodu globalnej epidemii wiele firm zdecydowało się na zmianę pracy stacjonarnej na zdalną. Nawet po złagodzeniu obostrzeń, znaczna część miejsc pracy pozostała przy takim trybie, lub przyjęło hybrydową formę pracy. Taka forma pracy prowadzi jednak do pewnych utrudnień. Pracownicy mogą musieć łączyć się za pomocą funkcji zdalnego pulpitu z komputerami znajdującymi się w biurze. Może to wynikać z niewystarczającej wydajności sprzętu pracownika, lub dostępu do specyficznych programów lub zasobów. W takim wypadku komputer, z którym łączy się pracownik, musi być uruchomiony, a w przypadku awarii - zrestartowany. Dodatkowo taki dostęp może być wymagany przez ograniczony czas, co powoduje, że dużą część czasu spędza włączony, ale nieużywany.

Możliwym sposobem na złagodzenie tego problemu jest użycie zmniejszonej liczby komputerów, które mogą być używane przez większą liczbę pracowników jednocześnie, za pośrednictwem maszyn wirtualnych. Tym zmniejszamy liczbę działających maszyn, a zarządzanie może być rozwiązane za pomocą zdalnego operowania komputerem, na którym działają.

System stworzony w ramach tej pracy adresuje opisany problem. Rozwiązanie opiera się na tym wcześniej opisanym, jednocześnie rozbudowując je w sposób ułatwiający użytkowanie oraz zarządzanie.

1.2. Podobne rozwiązania

1.3. Wizja systemu

Tworzony system ma za zadanie umożliwiać zdalną pracę za pomocą protokołu zdalnego pulpitu. System skierowany jest w stronę firm zatrudniających wielu pracowników, które chcą scentralizować sprzęt używany do pracy zdalnej.

Użytkownikami końcowym są pracownicy, którzy za pomocą okienkowej aplikacji klienckiej

mogą uzyskać sesję do pracy zdalnej. Użytkownik podczas łączy się za pomocą protokołu zdalnego pulpitu z maszyną wirtualną uruchamiającą obraz systemu GNU/Linux. Uruchamianie i zarządzanie maszynami jest zadaniem aplikacji działającej na rzeczywistej maszynie, która udostępnia swoje zasoby maszynom wirtualnym. Aplikacja ta, oraz rzeczywista maszyna uruchamiająca ją, nazywana jest dalej serwerem wirtualizacji. Aplikacje te działają niezależnie od siebie i nie ma teoretycznego ograniczenia na ich liczbę w systemie. Komunikacją z użytkownikami oraz zarządzaniem systemem zajmuje się aplikacja nadzorcza. Ilość jej instancji również jest teoretycznie nieograniczona, co umożliwia balansowanie obciążeniem.

Wyróżniamy dwa typy maszyn wirtualnych: maszyny wykorzystujące jedynie procesor maszyny, na której pracuje, oraz takie, które mają bezpośredni dostęp do karty graficznej maszyny. Do używania systemu użytkownik musi posiadać konto w systemie katalogowym, który umożliwia użytkownikom dostęp do własnego folderu domowego na każdej maszynie. System katalogowy nie jest ujęty w obrębie systemu, ale jego poprawna konfiguracja jest wymagana do użytkowania systemu.

System udostępnia panel administracyjny w postaci strony WWW umożliwiający podgląd obciążenia i stanu systemu przez upoważnione osoby. Komunikacja aplikacji klienckiej z aplikacją nadzorczą oraz panel administratora wykorzystują komunikację za pomocą protokołu HTTP. Możliwe jest użycie szyfrowanego protokołu HTTPS, pod warunkiem użycia poprawnych certyfikatów SSL/TSL.

1.4. Istotne pojęcia

- Aplikacja kliencka - aplikacja okienkowa uruchamiana na komputerze użytkownika, która umożliwi komunikację z systemem oraz uruchomienie zewnętrznego programu implementującego protokół RDP.
- Aplikacja nadzorcza (Nadzorca) - aplikacja, która przetwarza zapytania od aplikacji klienckiej oraz komunikuje się ze wszystkimi serwerami wirtualizacji. Na podstawie tych informacji buduje model zajętości każdego z serwerów wirtualizacji oraz decyduje kiedy, i na którym serwerze, trzeba uruchomić nowe maszyny wirtualne. Decyduje również, do której wirtualnej maszyny ma podłączyć się użytkownik proszący o utworzenie sesji.
- Serwer wirtualizacji - komputer, który udostępnia swoje zasoby (rdzenie procesora, karty graficzne, pamięć RAM oraz przestrzeń dyskową) w postaci uruchamianych na nim maszyn wirtualnych. Komputer ten uruchamia aplikację, która odpowiada na zapytania aplikacji

nadzorczej oraz wykonuje operacje na maszynach wirtualnych (uruchamianie i wyłączanie). Komputer może uruchamiać co najwyżej jedną aplikację, dlatego zarówno komputer, jak i aplikację, nazywamy serwerem wirtualizacji.

- Maszyna wirtualna CPU - maszyna systemowa emulująca, lub para-emulująca, sprzęt i służąca do uruchamiania systemu operacyjnego. Udostępnia użytkownikowi podstawowe zasoby (procesor, pamięć RAM i przestrzeń dyskowa). Uruchamiana jest na serwerze wirtualizacji z liczbą zasobów określoną w konfiguracji. Maszyna wirtualna uruchamia system operacyjny GNU/Linux (ArchLinux).
- Maszyna wirtualna GPU - maszyna analogiczna do maszyny wirtualnej CPU. Wyróżnia się przekazaną na wyłączność, za pośrednictwem mechanizmu GPU Passthrough, kartą graficzną podłączoną do serwera wirtualizacji.
- RDP - protokół zdalnego dostępu do pulpitu od firmy Microsoft¹. Maszyny wirtualne uruchamiają serwer RDP(XRDP²), który umożliwia zdalną pracę za pośrednictwem protokołu RDP.
- Sesja - jednorazowy dostęp użytkownika do systemu oraz maszyny wirtualnej. Utworzenie sesji wiąże się z przypisaniem do użytkownika konkretnej maszyny wirtualnej, na której będzie pracować. Sesja kończy się w przypadku, gdy użytkownik poinformuje system o zakończeniu pracy lub gdy minie czas oczekiwania na odzyskanie połączenia jego utracie.
- Vagrant-box³ - przygotowany wcześniej obraz maszyny wirtualnej, który umożliwia zmianę dostępnych zasoby. Uruchamiają się bardzo powtarzalnie w środowisku programu Vagrant. Obrazy te używane są do tworzenia maszyn wirtualnych.
- Ansible playbook⁴ - skrypt konfiguracyjny dla systemu operacyjnego, który umożliwia parametryzację oraz wykonywanie podczas uruchamiania Vagrant-boxa.
- Panel administratora - aplikacja przeglądarkowa, która umożliwia administratorowi systemu podgląd listy serwerów wirtualizacji znajdujących się w systemie oraz zajętości zasobów.
- Konto użytkownika - profil użytkownika w systemie, do którego ma dostęp na każdej maszynie wirtualnej. Używając przygotowanych wcześniej danych logowania może za ich pomocą

¹Dokumentacja protokołu RDP od Microsoft

²Strona projektu XRDP

³Dokumentacja i opis na stronie Vagranta

⁴Dokumentacja i opis na stronie Ansible'a

logować się do maszyn wirtualnych. Przechowywane są w zewnętrznym (poza opisanym systemem) systemie katalogowym.

- Katalog użytkownika - prywatny folder dostępny dla użytkownika na każdej maszynie wirtualnej. Przechowywany na zewnętrznym (poza opisanym systemem) dysku sieciowym.
- Konfiguracja stała - konfiguracja maszyny wirtualnej, która nie zmienia się w zależności od miejsca uruchomienia. Docelowo ta konfiguracja ma być zapisana w Vagrant-boxie. W razie potrzeby można ją także zdefiniować w odpowiednim Ansible playbooku.
- Konfiguracja zmienna - konfiguracja maszyny wirtualnej, która zmienia się w zależności od miejsca uruchomienia. Jest definiowana w odpowiednim Ansible playbooku uruchamianym przy każdym włączeniu maszyny.

1.5. Wymaganie funkcjonalne

1.5.1. Nadzorca

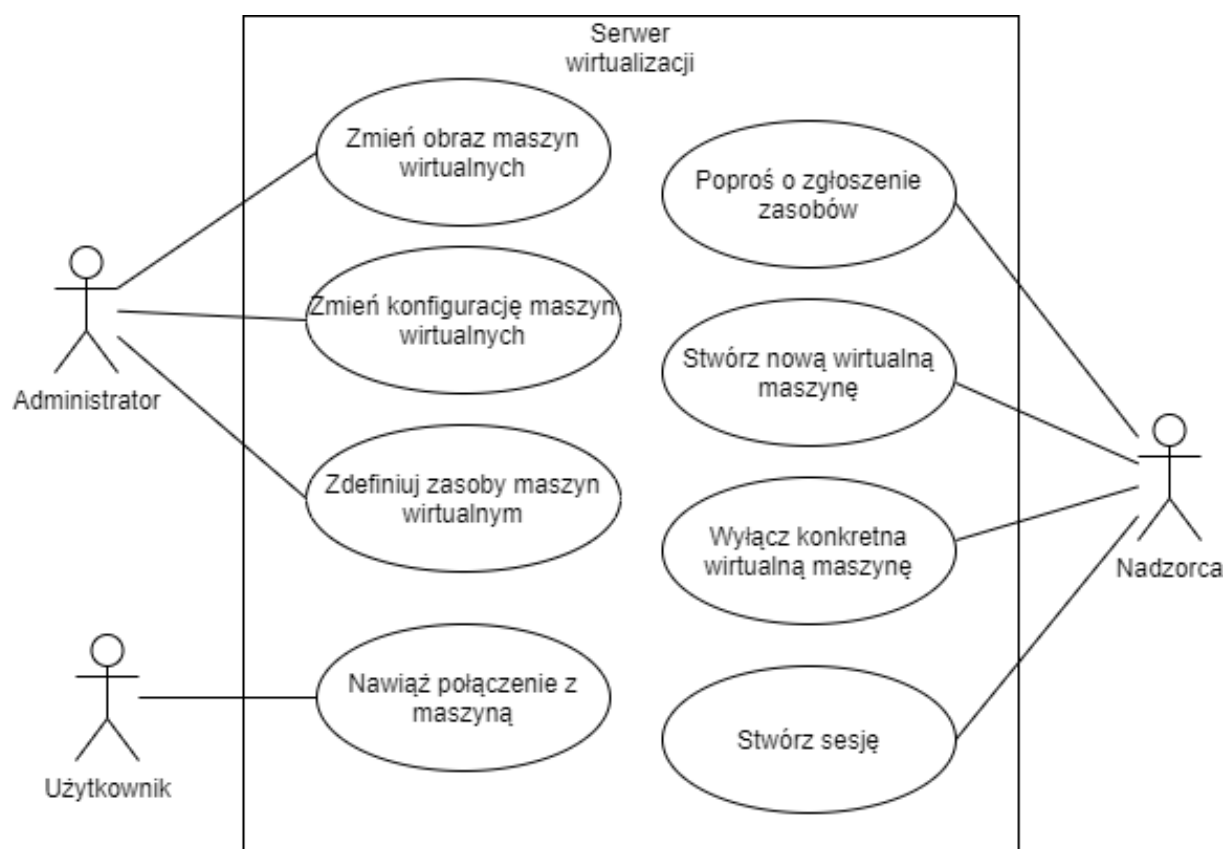


Rysunek 1.1: Przypadki użycia aplikacji nadzorczej

Tablica 1.1: Przypadki użycia aplikacji nadzorczej

Aktor	Nazwa	Opis	Odpowiedź systemu
Użytkownik	Uzyskanie sesji do pracy	Uzyskanie sesji do pracy na maszynie wirtualnej CPU lub GPU	Do użytkownika zostaje przydzielona maszyna wirtualna oraz zestawione połączenie RDP. W przypadku, gdy utracił on połączenie, to przydzielana jest do niego poprzednio używana maszyna, jeżeli jego sesja nie została jeszcze umorzona.
	Poznanie ilości dostępnych maszyn	Wyświetlanie szacowanej ilości dostępnych maszyn każdego typu	Użytkownikowi zostaje wyświetlona szacowana liczba dostępnych maszyn obliczona na podstawie informacji o dostępnych zasobach każdego z serwerów wirtualizacji
Server wirtualizacji	Zgłoszenie dostępnych zasobów	Serwer zgłasza nadzorcy dostępne zasoby	Nadzorca wykorzystuje zgłoszone zasoby do wyliczania szacowanej liczby dostępnych maszyn oraz do balansowania obciążenia serwerów wirtualizacji
Panel administratora	Podgląd stanu modelu	Nadzorca udostępnia panelowi administratora stan zasobów systemu.	Panel administratora wykorzystuje uzyskane dane do wygenerowania raportu o stanie systemu dla administratora wirtualizacji

1.5.2. Serwer wirtualizacji

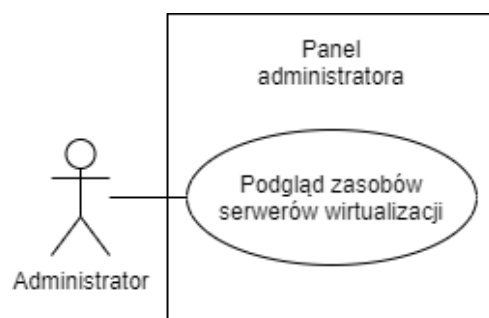


Rysunek 1.2: Przypadki użycia aplikacji nadzorczej

Tablica 1.2: Przypadki użycia serwera wirtualizacji

Aktor	Nazwa	Opis	Odpowiedź systemu
Użytkownik	Nawiązanie połączenia z maszyną	Użytkownik nawiązuje połączenie z maszyną wirtualną	Maszyna wirtualna zostaje zajęta przez użytkownika; serwer wirtualizacji rozpoczyna monitorowanie, czy sesja wciąż trwa
Nadzorca	Poproś o zgłoszenie zasobów	Nadzorca wysyła do wszystkich serwerów wirtualizacji prośbę o zgłoszenie swoich używanych i wolnych zasobów	Serwer wirtualizacji informuje nadzorcę o stanie swoich zasobów
	Stwórz nową wirtualną maszynę	Nadzorca prosi serwer wirtualizacji o stworzenie nowej wirtualnej maszyny dla danego użytkownika na wybranym typie maszyny	Serwer wirtualizacji tworzy wirtualną maszynę i udostępnia możliwość połączenia się z nią
	Wyłącz konkretną wirtualną maszynę	Nadzorca prosi serwer wirtualizacji aby wyłączył konkretną wirtualną maszynę.	Serwer wirtualizacji wyłącza konkretną wirtualną maszynę oraz pilnuje aby na pewno się wyłączyła.
Administrator	Zmień obraz maszyn wirtualnych	Zmiana obrazu źródłowego maszyn wirtualnych	Zdefiniowany przez administratora vagrant-box jest używany przez serwery wirtualizacji
	Zmień konfigurację maszyn wirtualnych	Zmiana zmiennej konfiguracji maszyn wirtualnych	Zmodyfikowany ansible playbook jest używany przez serwery wirtualizacji
	Zdefiniuj zasoby maszyn wirtualnych	Zmiana ilości zasobów przydzielanych na każdy z typów maszyn wirtualnych oraz łączną ilość zasobów przeznaczonych na maszyny	Zmodyfikowana konfiguracja zasobów będzie wykorzystywana przez serwer wirtualizacji przy kolejnym uruchomieniu

1.5.3. Panel administratora



Rysunek 1.3: Przypadki użycia aplikacji nadzorczej

Tablica 1.3: Przypadki użycia panelu administratora

Aktor	Nazwa	Opis	Odpowiedź systemu
Administrator	Podgląd zasobów serwerów wirtualizacji	Wyświetlanie wolnych oraz zajętych zasobów serwerów wirtualizacji	Wyświetlenie zasobów poszczególnych serwerów wirtualizacji, liczby zajętych maszyn oraz szacowanej liczby wolnych maszyn

1.5. WYMAGANIE FUNKCJONALNE

1.6. Wymaganie niefunkcjonalne

Tablica 1.4: Wymagania niefunkcjonalne

Grupa wymagań	Nr wymagania	Opis
Użytkowanie (Usability)	1	Aplikacja kliencka ma działać na systemach operacyjnych MS Windows (Windows 10) oraz GNU/Linux (ArchLinux). Aplikacja na systemach GNU/Linux wymaga zainstalowanego klienta RDP zgodnego z XRDP ⁵ .
	2	Aplikacja kliencka musi udostępniać możliwość użycia własnego klienta RDP do nawiązania połączenia z maszyną wirtualną
	3	Maszyny wirtualne muszą mieć dostęp do systemu przechowującego konta użytkowników wraz z ich katalogami domowymi
Niezwadność (Reliability)	4	System musi być odporny na awarie poszczególnych serwerów wirtualizacji i kontynuować działanie w sposób niezauważalny dla użytkowników nie używających danego serwera.
	5	Awaria nadzorcy może spowodować uniemożliwienie rozpoczęcia nowych sesji, ale nie może przerwać istniejących sesji
Wydażność (Performance)	6	Łącznie zużywane zasoby przez maszyny wirtualne na poszczególnym serwerze wirtualizacji nie mogą przekroczyć wcześniej zdefiniowanych limitów
	7	Nadzorca musi balansować obciążenie serwerów wirtualizacji
	8	W systemie zawsze musi istnieć jedna działająca maszyna wirtualna nie połączona z żadną sesją, aby można było ją szybko przydzielić użytkownikowi
	9	Zwolnione maszyny wirtualne, które nie są wykorzystywane jako zapas, muszą być wyłączane
Utrzymanie (Supportability)	10	Możliwe jest działanie więcej niż jednego nadzorcy w systemie, w celu zwiększenia dostępności lub przeprowadzenia prac utrzymaniowych

1.7. Analiza ryzyka

Tablica 1.5: Analiza ryzyka

<p>Mocne strony</p> <ul style="list-style-type: none"> • Łatwa skalowalność pod względem liczby sesji w systemie • Wiele rozwiązań Open Source • Elastyczność pod względem konfiguracji • Tańsze rozwiązanie niż kupno stacji roboczych 	<p>Słabości</p> <ul style="list-style-type: none"> • System trudny w konfiguracji • Potrzeba wymiany sprzętu komputerowego • Krótki czas rozwoju systemu • Ograniczenie doświadczenie twórców systemu • Małe prawdopodobieństwo wsparcia projektu po zakończeniu prac
<p>Okazje</p> <ul style="list-style-type: none"> • Grupa docelowa to firmy z dużą ilością stacji roboczych • Zwiększenie zapotrzebowania na prace zdalną na rynku pracy 	<p>Zagrożenia</p> <ul style="list-style-type: none"> • Istnienie konkurencji ugruntowanej na rynku • System w dużej mierze oparty o oprogramowanie rozwijane przez inne organizacje

1.7.1. Omówienie zagrożeń

- System trudny w konfiguracji - wysoko prawdopodobne

Można temu zaradzić poprzez udostępnienie dokładnej dokumentacji lub ścisłą współpracę z klientem przy wdrażaniu systemu.

Wartość: duża

- Potrzeba wymiany sprzętu komputerowego - średnio prawdopodobne

Klient może potrzebować wymienić aktualne stacje robocze na terminale oraz zainwestować w sprzęt serwerowy. Jednak gdy klientami będą firmy, które mają dużo pracowników pracujących spoza biura, lub dopiero tych pracowników pozyskują, to kupno terminali i

serwerów powinno być bardziej zachęcające niż kupno stacji roboczych.

Wartość: średnia.

- Krótki czas rozwoju systemu - wysoko prawdopodobne

Czas rozwoju systemu jest bardzo ograniczony. Aby pomimo tego ograniczenia działał on w sposób akceptowalny powinniśmy skupić się na dobrym przedyskutowaniu i opisanu kluczowych modułów systemu. W czasie projektu należy pilnować aby nie dodawać nadmiarowych funkcjonalności do systemu. W czasie implementacji krytyczne będzie dokładne zaplanowanie aplikacji pod kątem testowania automatycznego. Ułatwi to wyłapywanie prostych błędów jeszcze we wczesnej fazie projektu.

Wartość: wysoka

1.8. PODZIAŁ PRACY

- Ograniczone doświadczenie twórców systemu - pewne

Jedynym sposobem na ograniczenie ryzyka jest rozważna implementacja.

Wartość: średnia

- Małe prawdopodobieństwo wsparcia projektu po zakończeniu prac - wysoko prawdopodobne

Trudno teraz przewidzieć co się stanie z projektem po zakończeniu prac. Jednak prawdopodobnie twórcy systemu zajmą się innymi projektami. Można jedynie dokładnie komentować kod i pokrywać jak najwięcej jego części testami. Wtedy inne osoby będą w stanie szukać błędów albo próbować w taki sposób uzupełnić brakującą wiedzę o systemie.

Wartość: niska

- Istnienie konkurencji ugruntowanej na rynku - bardzo prawdopodobne

Konkurencyjne systemy oferujące podobne rozwiązania są już dobrze ugruntowane na rynku i przetestowane. Nasz system może spróbować konkurować jedynie z nimi ceną implementacji oraz elastycznością.

Wartość: średnia

- System w dużej mierze oparty o oprogramowanie rozwijane przez inne organizacje - nisko prawdopodobne

W czasie życia systemu mogą pojawić się błędy w oprogramowaniu nie rozwijanym w ramach naszego systemu. naprawa takich błędów może trwać bardzo długo. Pewnym sposobem wsparcia takiego systemu jest własnoręczne poprawianie błędów w zewnętrznym oprogramowaniu i zgłaszanie ich do odpowiedniej organizacji. Do czasu zastosowania poprawki jest możliwość korzystania z wersji, na którą nanieśliśmy własną poprawkę.

Wartość: wysoka

1.8. Podział pracy

2. Opis rozwiązania

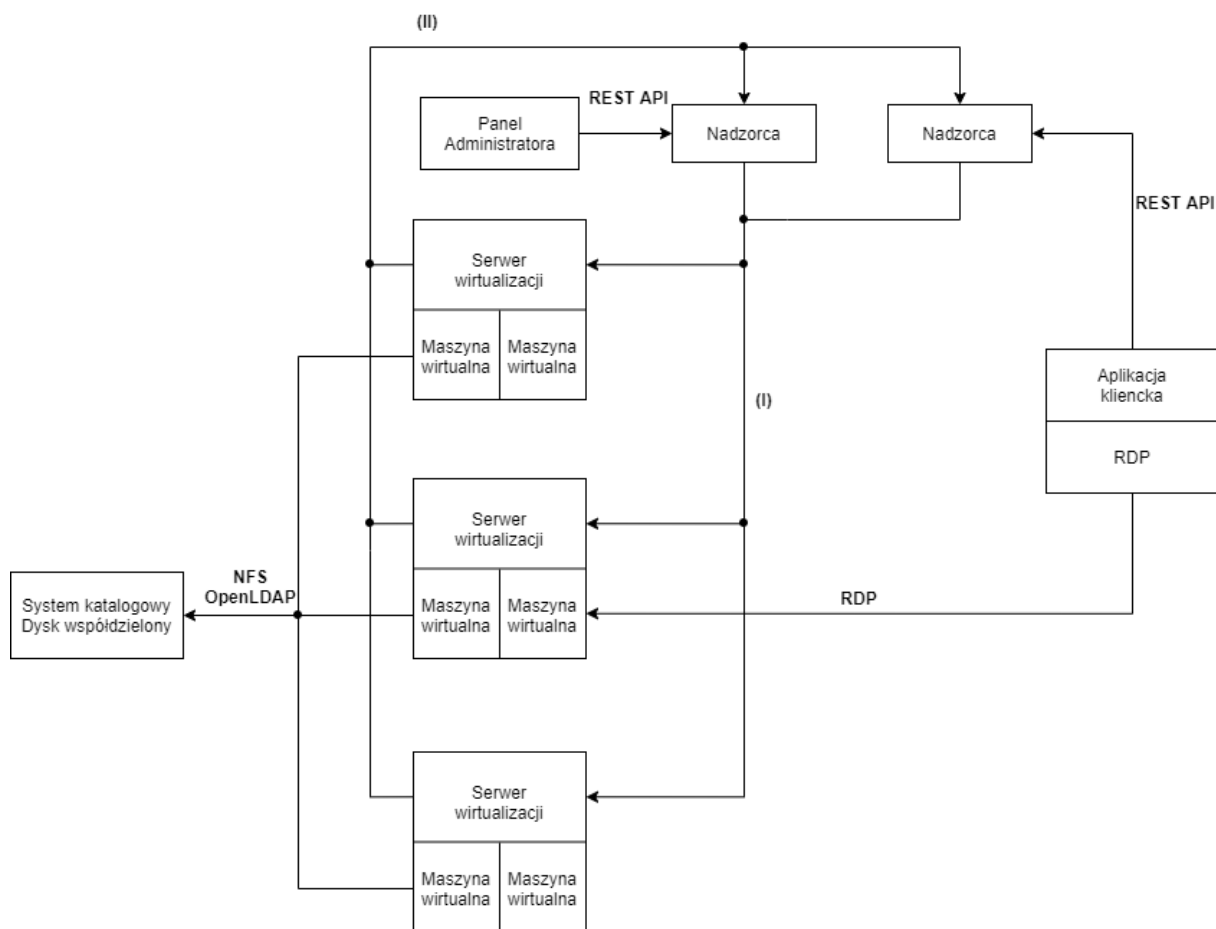
2.1. Architektura systemu

Opracowywany system składa się z następujących modułów:

- nadzorcy,
- serwera wirtualizacji,
- aplikacji klienckiej,
- panelu administratora,
- brokera wiadomości,
- systemu katalogowego,
- dysku współdzielonego.

Schematyczny obraz systemu przedstawia poniższy rysunek.

2.1. ARCHITEKTURA SYSTEMU



Rysunek 2.1: Schematyczna architektura systemu

Połączenia oznaczone liczbami rzymskimi oznaczają kolejki komunikacji za pośrednictwem brokera wiadomości, które opisane zostały w punkcie jemu poświęconym. Z założenia system powinien móc skalować się w dwóch wymiarach, to znaczy:

1. Zwiększanie liczby serwerów wirtualnych - zwiększenie liczby istniejących jednocześnie sesji.
2. Zwiększenie liczby nadzorców - zwiększenie liczby obsługiwanych klientów jednocześnie oraz niezawodności systemu.

2.1.1. Nadzorca

Aplikacja mająca za zadanie obsługiwać komunikację z aplikacjami klienckimi oraz wysyłać polecenia do serwerów wirtualizacji. Udostępnia REST API służące do komunikacji z aplikacjami klienckimi. Do komunikacji z serwerami wirtualizacji wykorzystuje kolejki.

Nadzorca przechowuje wewnętrznie model systemu zawierający informację o działających serwerach wirtualizacji i stanie ich maszyn. Na podstawie tego modelu moduł stwierdza, do której maszyny przypisać nowo utworzoną sesję. Wewnętrzne procesy skupione są wokół zmian modelu.

Jeżeli proces wysłał do serwera wirtualizacji prośbę o zmianę stanu, to dalsze procesowanie odbywa się, gdy stan modelu został zaktualizowany, i na podstawie jego stanu podejmowane są decyzje.

Dzięki zastosowaniu kolejek oraz zasad komunikacji w systemie może istnieć więcej niż jeden nadzorca. Instancje nadzorców działają niezależnie od siebie i przechowują identyczny model systemu. Dzięki temu uzyskujemy retencję i możemy zmniejszyć obciążenie poszczególnych nadzorców.

2.1.2. Serwer wirtualizacji

Zadaniem serwera wirtualizacji jest uruchamianie i zarządzanie maszynami wirtualnymi, z którymi łączy się użytkownik systemu. Komunikuje się on z nadzorcami i wykonuje operacje na maszynach wirtualnych zgodnie z żądaniami.

Moduł ten nie jest w stanie funkcjonować samodzielnie. Z tego powodu aplikacja nie uruchomi się, jeżeli nie jest w stanie nawiązać połączenia z aplikacją nadzorczą, a w przypadku ostatni nadzorca w systemie zakończy działanie, aplikacja również je zakończy, pod warunkiem że nie ma żadnych działających sesji.

Serwer wirtualizacji jest częścią systemu, która przechowuje realne zasoby udostępniane użytkownikom. System zaprojektowany jest w taki sposób aby teoretycznie nie było ograniczenia na liczbę serwerów wirtualizacji działających jednocześnie.

2.1.3. Aplikacja kliencka

Aplikacja okienkowa umożliwiająca użytkownikowi autoryzację, uzyskanie sesji oraz automatyczne rozpoczęcie połączenia. Komunikuje się z nadzorcą za pomocą REST API.

Proces uzyskania sesji z perspektywy aplikacji klienckiej zawiera:

1. Uzyskanie informacji o dostępnych typach i liczbie maszyn
2. Wybór typu maszyny
3. Oczekiwanie na utworzenie sesji
4. Nawiązanie połączenia RDP
5. Utrzymanie i monitorowanie stanu połączenia.

2.1.4. Panel administratora

Prosta aplikacja internetowa umożliwiająca administratorowi systemu podgląd stanu zużycia zasobów serwerów wirtualizacji.

2.1.5. Broker wiadomości

Komunikacje wewnątrz systemu, czyli pomiędzy serwerami wirtualizacji oraz nadzorcami, będzie realizowali poprzez kolejki wiadomości. W tym celu użyty został system RabbitMQ, który zajmuje się transportem wiadomości wewnątrz systemu oraz niezawodnością komunikacji między modułami.

Zdefiniowane zostały następujące kolejki wiadomości:

- (I) Kolejka kończąca się na każdym z serwerów wirtualizacji powielająca wiadomości między nich. Służy ona do wysyłania nie spersonalizowanych próśb od nadzorców do serwerów wirtualizacji.
- (II) Kolejka kończąca się na każdym z nadzorców powielająca wiadomości między nich. Służy ona do przesyłania informacji do nadzorców o zmianach wewnątrz serwera wirtualizacji.
- (III) Kolejka kończąca się wyłącznie na pojedynczym serwerze wirtualizacji. Liczba kolejek zgadza się z liczbą serwerów wirtualizacji aktywnych w systemie. Służą one do przesyłania spersonalizowanych wiadomości oraz sprawdzania, czy serwer wirtualizacji nadal pracuje po drugiej stronie. Skorzystamy z funkcjonalności kolejek na wyłączność (Exclusive Queue¹).
- (IV) Kolejka kończąca się na aktualnie podłączonym do maszyny wirtualnej kliencie. Podobnie jak powyżej kolejek istnieje tyle ile aktywnych użytkowników. Celem kolejki jest sprawdzenie, czy aplikacja kliencka nadal jest podłączona do wirtualnej maszyny (mechanizm Exclusive Queue). W celach bezpieczeństwa będą one definiowane na oddzielnym procesie brokera, który będzie można w razie potrzeby udostępnić poza sieć lokalną.

Powyższe 4 grupy kolejek umożliwią prawidłowe działanie systemu. Każdy z modułów tworzy w trakcie uruchamiania kolejki, z których odbiera wiadomości. Jedynym wymogiem prawidłowego uruchomienia komunikacji jest dostępny dla wszystkich serwerów wirtualizacji oraz nadzorców proces brokera.

2.2. Zewnętrzne narzędzia

2.2.1. Ansible

Ansible został wykorzystany w systemie do zaaplikowania zmiennej konfiguracji do każdej uruchamianej wirtualnej maszyny. Podstawowo playbook będzie zawierać informacje o:

¹Opis zachowania kolejek na wyłączność

1. danych dostępowych do dysku sieciowego oraz wykorzystanym protokole,
2. danych dostępowych do usługi katalogowej.

Playbook można rozszerzać o potrzebne dane zależne od użycia.

2.2.2. Vagrant

Vagrant został wykorzystany w celu łatwej parametryzacji oraz powtarzalnego tworzenia maszyn wirtualnych z przygotowanego wcześniej obrazu systemu.

Wykorzystywany jest głównie mechanizm Vagrant-boxów, które są obrazami wcześniej przygotowanego systemu operacyjnego. Aby system działał prawidłowo obraz systemu zamknięty w Vagrantboxie musi spełniać następujące warunki:

1. Użytkownicy muszą być pobierani z usługi katalogowej.
2. Katalogi domowe użytkowników muszą być na dysku sieciowym.
3. Musi istnieć serwer RDP

2.2.3. Libvirt z QEMU

Libvirt połączony z QEMU jest wykorzystany do zarządzania maszynami wirtualnymi uruchamianymi na serwerze wirtualizacji. Umożliwia on:

1. Tworzenie maszyn wirtualnych.
2. Uruchamianie maszyn wirtualnych.
3. Przyporządkowanie zasobów maszynom wirtualnym (w tym krat graficznych).
4. Wyłączanie maszyn wirtualnych.
5. Sprawdzanie, czy maszyna o danej nazwie już działa.

2.3. Technologie

- Aplikacja kliencka

– Typescript² /Javascript³

²Strona projektu Typescript

³Obecny standard języka Javascript

- Node.js⁴ - środowisko uruchomieniowe używane do integracji z systemem użytkownika
- Angular⁵ - renderowanie widoków
- Electron⁶ - platforma programistyczna
- Jest⁷ - testy jednostkowe
- Cypress⁸ - testy integracyjne
- Panel administratora
 - Typescript/Javascript
 - Angular - platforma aplikacji WWW
 - Jest - testy jednostkowe
 - Cypress - testy integracyjne
- Nadzorca i serwer wirtualizacji
 - C#⁹
 - RabbitMQ¹⁰ - broker asynchronicznych wiadomości
 - Ansible¹¹ - konfigurowanie maszyn wirtualnych
 - Vagrant¹² - tworzenie obrazów maszyn wirtualnych oraz ich uruchamianie
 - libvirt¹³ - zarządzanie maszynami wirtualnymi
 - OpenLDAP¹⁴ - dostęp do systemu katalogowego
 - NFS¹⁵ - dostęp do katalogów domowych z maszyny wirtualnej
 - Arch Linux¹⁶ - system operacyjny uruchamiany przez maszyny wirtualne
 - GNU/Linux - wspierany system operacyjny

- Różne

⁴Strona projektu Node.js

⁵Strona projektu Angular

⁶Strona projektu Electron

⁷Strona projektu Jest

⁸Strona projektu Cypress

⁹Dokumentacja języka C#

¹⁰Strona projektu RabbitMQ

¹¹Strona projektu Ansible

¹²Strona projektu Vagrant

¹³Strona projektu libvirt

¹⁴Strona projektu OpenLDAP

¹⁵Opis na stronie firmy Microsoft

¹⁶Strona systemu operacyjnego Arch Linux

- Swagger Codegen¹⁷ - automatyczna generacja API na podstawie specyfikacji
- RDP¹⁸ - łączenie ze zdalnymi sesjami

2.4. Stany biznesowe

2.5. Procesy biznesowe

2.6. Komunikacja

2.6.1. Komunikacja wewnętrzna

2.6.2. Komunikacja zewnętrzna

2.6.3. Sekwencje komunikacji

¹⁷Opis narzędzia na stronie firmy Swagger

¹⁸Dokumentacja protokołu RDP od Microsoft

3. Analiza rozwiązania

4. Podsumowanie

Bibliografia

- [1] A. Author, *Title of a book*, Publisher, year, page–page.
- [2] J. Bobkowski, S. Dobkowski, Jak stworzyć bibliografię w BibTeX-u, *Czasopismo nr*, rok, strona–strona.
- [3] C. Brink, Power structures, *Algebra Universalis* 30(2), 1993, 177–216.
- [4] F. Burris, H. P. Sankappanavar, *A Course of Universal Algebra*, Springer-Verlag, Nowy Jork, 1981.

Wykaz symboli i skrótów

nzw. nadzwyczajny

* operator gwiazdka

~ tyllda

Jak nie występują, usunąć.

Spis rysunków

1.1	Przypadki użycia aplikacji nadzorczej	14
1.2	Przypadki użycia aplikacji nadzorczej	16
1.3	Przypadki użycia aplikacji nadzorczej	18
2.1	Schematyczna architektura systemu	25

Spis tabel

1.1	Opis skrócony	15
1.2	Opis skrócony	17
1.3	Opis skrócony	18
1.4	Opis skrócony	20
1.5	Opis skrócony	21

Spis załączników

1. Załącznik 1
2. Załącznik 2
3. Jak nie występują, usunąć rozdział.