

# Wirtualizacja sieci bezprzewodowych

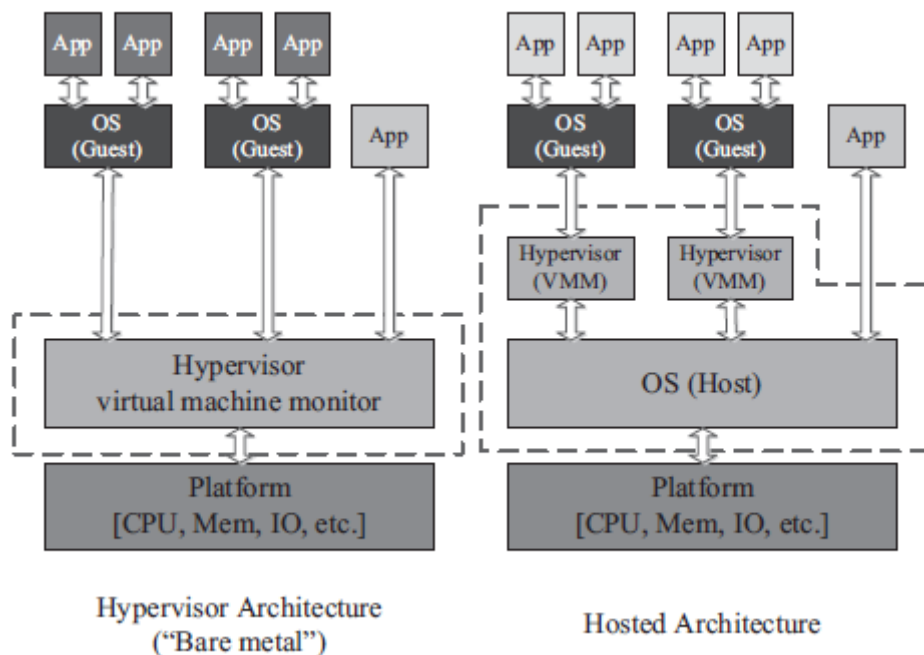
## 10.1 Wprowadzenie

Dzięki imponującemu rozwojowi inteligentnych terminali i kart danych, mobilnego Internetu w ciągu ostatnich kilku lat nastąpił gwałtowny wzrost usługi. Jednak ludzie nie widzą, że operatorzy telefonii komórkowej zawsze ponoszą wiele wysiłku, aby odpowiedzieć na stale rosnący ruch. W rzeczywistości, podczas gdy ruch ma wzrost wykładniczy, tempo wzrostu dochodów i zysków większości operatorów jest nie więcej niż 10%, a nawet ujemne. Powodem jest to, że szybko rozwijający się mobilny Internet wywiera ogromną presję na sieci komórkowe. Aby poradzić sobie z codziennym rosnącym ruchem, operatorzy muszą płacić więcej zasobów za wdrożenie, przebudowę lub modernizację sieci, co prowadzi do ogromnych nakładów inwestycyjnych i kosztów operacyjnych (OPEX). Łatwo można sobie wyobrazić, że jeśli w nadchodzących latach mobilny Internet będzie się nadal zwiększał, sieci komórkowe będą coraz bardziej naciskać. Operatorzy mobilni stoją przed niespotykanymi dotąd wyzwaniami. W jaki sposób operatorzy skutecznie redukują OPEX, zwiększają wykorzystanie sieci, a nawet przyspieszają aktywację rozwoju usług, aby sprostać wyzwaniom związanym z erą mobilnego Internetu? Aby rozwiązać ten problem, zacznijmy od wiedzy na temat tradycyjnego sprzętu telekomunikacyjnego, jego formy, trybu rozwoju i debugowania, konserwacji sieci i aktualizacji. Sprzęt sieci telekomunikacyjnej ma dwie główne cechy, mianowicie "zastrzeżoną platformę" i "dedykowane chipsety". "Zastrzeżona platforma" oznacza, że platforma, na której opracowywane jest urządzenie, jest zastrzeżona, różni się od dostawcy do dostawcy. "Dedykowane chipsety" oznaczają, że chipsety, takie jak cyfrowe przetwarzanie sygnału (DSP), programowalna matryca bramkowa (FPGA) i układ scalony specyficzny dla aplikacji (ASIC) są głównie przystosowane do rozwoju sprzętu. W wyniku tego zastrzeżone platformy powodują słabe problemy z kompatybilnością i kompatybilnością, podczas gdy dedykowane użycie chipsetu powoduje problemy ze słabą programowalnością, słabą przenośnością i powolnym rozwojem dodatkowym. Ponadto branża ta wymaga inżynierów o wysokich umiejętnościach, którzy łatwo można znaleźć. W ten sposób ograniczona jest konkurencyjność i innowacyjność usług w całej branży. Poza tym sprzęt zazwyczaj wymaga długiego cyklu programowania, złożonego testowania i weryfikacji. Poza tym, sieć telekomunikacyjna zużywa ogromną ilość energii przy niskim wykorzystaniu. Można przewidywać, że stale rosnące wymagania mobilnego Internetu zwiększą wymagania sieci bezprzewodowych następnej generacji pod względem możliwości rozbudowy sieci, innowacji i terminowego wprowadzania usług, wydajności energetycznej itd., które są tym, czym są tradycyjne tryby rozwoju sprzętu. niezdolny do spełnienia. Następnie zwróćmy naszą uwagę na branżę IT. Krajobraz jest inny. Branża IT słynie z otwartości i miękkości. Różne standardowe interfejsy znacznie ułatwiają współpracę, dzięki czemu programiści mogą skoncentrować się na rozwoju i optymalizacji niektórych komponentów bez konieczności uwzględniania problemów z połączeniami. Dobrymi przykładami są karty wideo, procesor, wspomnienia i tak dalej. Ponadto platforma ogólnego przeznaczenia (GPP), np. Architektura x86 jest powszechnie stosowana, co ułatwia tworzenie, utrzymywanie i przenoszenie oprogramowania. To, w pewnym sensie, zachęca do innowacji usług i przyspiesza czas wprowadzenia produktu na rynek. Rozwój branży IT, w szczególności dobrobytu centrów danych, jest w dużej mierze zasługą kluczowej technologii, a mianowicie technologii wirtualizacji.

## 10.2 Podstawy wirtualizacji

Jeśli chodzi o pochodzenie technologii wirtualizacji, po raz pierwszy użył tego terminu dr Christopher Strachey z Oxford University. W książce "Time Sharing in Large Fast Computers" wspomniano o koncepcji wirtualizacji w latach sześćdziesiątych. Następnie w 1964 roku IBM zainicjował projekt, który nazwano systemem CP / CMS i ewoluował do Virtual Machine Facility / 370. Od tej pory do odróżnienia

od maszyny fizycznej używano pojęcia maszyny wirtualnej (VM). W obszarze technologii wirtualizacji IBM i środowisko badawcze stwierdziły podstawowe pojęcia związane z wirtualizacją, a Goldberg opublikował prace [3], które dotyczyły koncepcji i korzyści tej technologii. Jednak od kilku lat zainteresowanie spadło, a nic związanego z technologią wirtualizacji nie miało miejsca aż do 1999 roku, kiedy firma VMware Inc. zaprezentowała swój produkt VMware Virtual Platform dla architektury x86-32. Ale czym jest w skrócie wirtualizacja? Jedna opinia jest taka, że wirtualizacja jest aktem ABSTRAKCJI fizycznych granic technologii. Aby osiągnąć fizyczną abstrakcję, istnieje kilka sposobów. Jednym z nich jest to, że aplikacje lub serwery nie potrzebują już dedykowanego sprzętu fizycznego, aby działać jako niezależne jednostki i że maszyna wirtualna może wejść w grę. W przypadku maszyn wirtualnych tradycyjny sprzęt jest emulowany i prezentowany systemowi operacyjnemu tak, jakby rzeczywiście istniał. Poza tym wiele maszyn wirtualnych może działać równolegle w różnych systemach operacyjnych na tym samym sprzęcie. Inna opinia preferuje prostą analogię wirtualizacji do obrazu-obrazu niektórych telewizorów i dekoderek, ponieważ wyświetla mały wirtualny obraz telewizyjny na drugim obrazie telewizyjnym, umożliwiając jednocześnie odtwarzanie obu programów [6]. W oparciu o zwirtualizowaną platformę, która jest połączeniem wirtualizacji komputerów, sieci i pamięci masowej, tradycyjne aplikacje lub wiele zestawów oprogramowania może być wykonywanych w tym samym czasie. W ścieżce ewolucji technologii wirtualizacji istnieją dwa typy monitorów maszyn wirtualizacji (VMM). Jeden uruchamia się bezpośrednio na sprzęcie z takimi samymi funkcjami, jak w tradycyjnych systemach operacyjnych (OS) w celu kontrolowania lub zarządzania sprzętem. Nazywa się to architekturą "Hypervisors". Drugi działa na istniejącym systemie operacyjnym i zarządza systemem operacyjnym. W tym scenariuszu system operacyjny poniżej VMM nazywa się Hostem, a ten jest nazywany systemem gościa. Tryb nazywa się architekturą "Hosted". Rysunek 10.1 pokazuje przykład tych dwóch typów VMM i hipernadzorców.



W przypadku wirtualizacji na poziomie systemu istnieją trzy podejścia: pełna wirtualizacja, parawirtualizacja i wirtualizacja systemu operacyjnego. Możemy po prostu wprowadzić różnice między tymi trzema metodami:

- Pełna wirtualizacja ma jeden dedykowany VMM lub hiperwizor do zarządzania bazowym sprzętem. Dzięki asystentowi sprzętowemu procesory Intel i AMD mogą obsługiwać takie wirtualizacje. Może obsługiwać różne systemy operacyjne gościa.
- Parawirtualizacja różni się nieco od pełnej wirtualizacji. Ma lepszą wydajność niż pełna wirtualizacja. Ale wymaga również modyfikacji systemu gościa.
- Wirtualizacja systemu operacyjnego wymaga jednej warstwy do wirtualizacji i ma najlepsze skalowalności wydajności. Ale obsługuje tylko jeden system operacyjny. W przypadku powyższych trzech podejść nie ma potrzeby wprowadzania żadnych zmian w aplikacjach. Różnica polega na złożoności, możliwościach i liczbie obsługiwanych systemów operacyjnych.

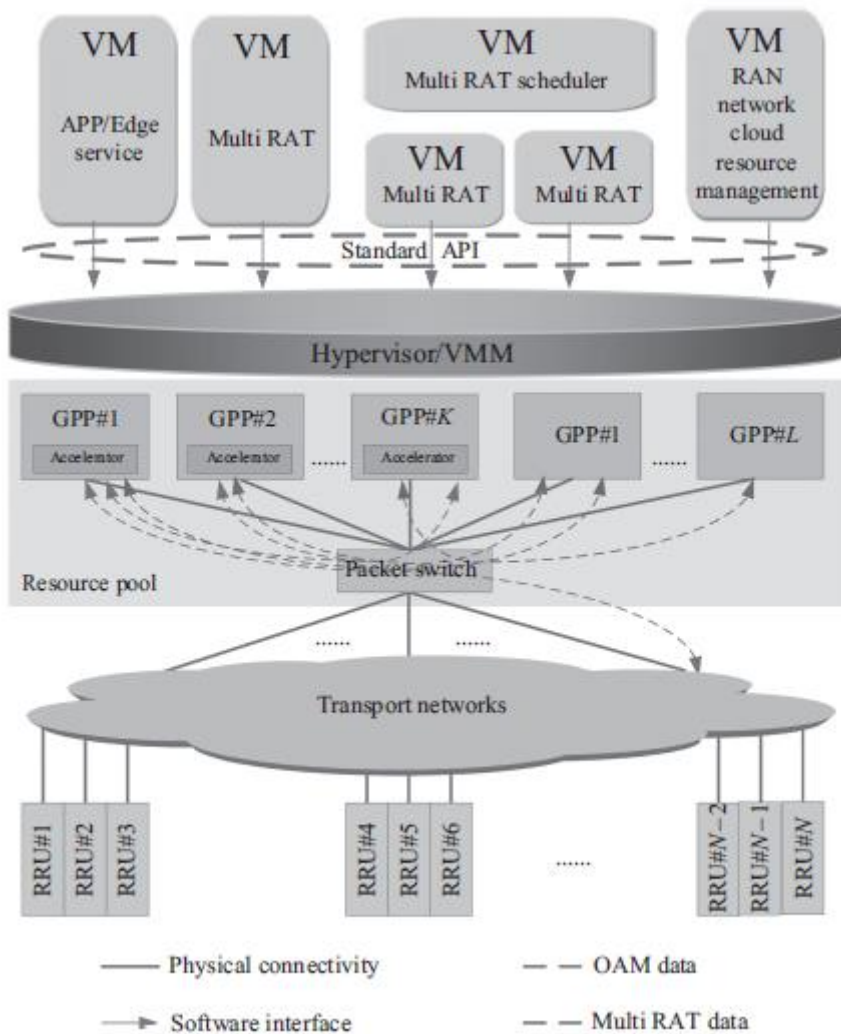
### **10.3 Wirtualizacja Cloud RAN i jej wyzwania**

Z jednej strony operatorzy stają przed wyzwaniami narzucanymi przez coraz bardziej ruch mobilny. Z drugiej strony jest świadkiem, że technologia wirtualizacji wykroczyła poza branżę IT i rozszerzyła swój wpływ na kilka innych branż. Ludzie zaczynają myśleć o możliwości wprowadzenia wirtualizacji do branży telekomunikacyjnej. W 2012 r. Utworzono Industry Specification Group (ISG) o nazwie Network Functions Virtualization (NFV) pod patronatem European Telecommunications Standards Institute (ETSI). Podstawową ideą NFV jest "konsolidacja wielu typów urządzeń sieciowych na standardowe, o wysokim natężeniu serwery, przełączniki i pamięć masową, które mogą znajdować się w centrach danych, węzłach sieci oraz w pomieszczeniach użytkownika końcowego ". Ludzie oczekują, że wirtualizacja może przynieść operatorom takie korzyści, jak szybszy proces rozwoju, uaktualnienie oprogramowania podobne do oprogramowania, mniej skomplikowane zarządzanie, szybszy czas wejścia na rynek i lepszą moc wydajność i tak dalej. Sieci bezprzewodowe składają się głównie z sieci rdzeniowej i sieci dostępu radiowego (RAN). Jest oczywiste, że RAN jest znacznie bardziej skomplikowany niż sieć rdzeniowa z punktu widzenia obliczeń i wymagań w czasie rzeczywistym. W związku z tym istnieje szeroki konsensus w branży, że realizacja wirtualizacji RAN jest znacznie trudniejsza niż w przypadku sieci rdzeniowej. W tej części skupimy się głównie na wirtualizacji RAN. W szczególności skupimy się na Cloud RAN (C-RAN), która jest niezbędną architekturą w stosunku do 5G.

#### **10.3.1 Podstawowy C-RAN**

Koncepcja C-RAN została po raz pierwszy zaproponowana przez China Mobile w białej księdze w 2009 roku. "C" tutaj oznacza "Scentralizowany, współpracujący, Cloud i Clean". Podstawowa idea C-RAN rozpoczyna się od centralizacji, która polega na agregowaniu różnych jednostek pasma podstawowego (BBU), które w tradycyjnym wdrożeniu są geograficznie rozdzielone na tę samą lokalizację. Po scentralizowaniu możliwe byłoby, aby różne jednostki BBU komunikowały się ze sobą w bardziej punktualny sposób, łącząc je z szybkimi sieciami przełączników, umożliwiając w ten sposób implementację algorytmów współpracy w celu poprawy wydajności systemu. Prosta centralizacja nie obsługuje udostępniania zasobów między różnymi fizycznymi jednostkami BBU. Dzięki wirtualizacji to się wkrótce zmieni. Ostatecznym celem C-RAN jest realizacja tak zwanej cechy " zasocjowania zasobów ", która polega na połączeniu zasobów przetwarzania pasma podstawowego, aby zasoby mogły być zarządzane i dynamicznie przydzielane na żądanie. Wirtualizacja jest kluczem do realizacji procesu chmury zasobów. Wirtualizacja sieci RAN ma na celu architekturę sieci RAN z ewoluującym standardem IT technologii wirtualizacji w celu konsolidacji tradycyjnych typów urządzeń sieciowych na standardowych serwerach, przełącznikach i pamięciach o dużej pojemności, które będą znajdować się w witrynach bezprzewodowych lub biurach centralnych. Polega ona na implementacji części funkcji RAN w oprogramowaniu, które może działać na szeregu standardowych urządzeń serwerowych i które mogą być przenoszone do lub lokalizowane w różnych lokalizacjach w sieci zgodnie z wymaganiami,

bez konieczności instalowania nowych sprzęt. Rysunek 10.2 ilustruje zwirtualizowany system C-RAN oparty na platformie Commercialoffs-shelf (COTS).



Dzięki wirtualizacji standardowe serwery IT są używane jako ogólna platforma z obliczeniami i pamięcią masową jako wspólne zasoby. Jak pokazano na Rysunku 10.2, na górze serwerów działają różne aplikacje w postaci maszyn wirtualnych (VM). Niezbędne aplikacje w C-RAN to te, które realizują różne technologie dostępu radiowego (RAT), w tym 2G, 3G, 4G i przyszłe 5G. Dodatkowe aplikacje użytkownika, takie jak sieć dostarczania treści, mogą być również wdrażane na tej samej platformie.

### 10.3.2 Wyzwania związane z wirtualizacją C-RAN

Komunikacja bezprzewodowa, w szczególności przetwarzanie pasma podstawowego w RAN, różni się od centrów danych IT tym, że komunikacja bezprzewodowa ma niezwykle surowe wymagania dotyczące przetwarzania w czasie rzeczywistym. Na przykład, w przypadku systemów LTE wymagane jest, aby sygnał ACK / NACK był wytwarzany i wysyłany z powrotem do UE / eNB w ciągu 3 ms po odebraniu ramki. Tradycyjna technologia wirtualizacji centrów danych z procesorem ogólnego zastosowania (GPP) nie może spełnić tego wymogu. Dlatego zastosowanie wirtualizacji w stacjach bazowych wymaga starannego projektowania i specjalnej optymalizacji kluczowych bloków funkcjonalnych. Aby wdrożyć wirtualne stacje bazowe w czasie rzeczywistym w scentralizowanej puli pasm podstawowych, należy rozwiązać następujące problemy:

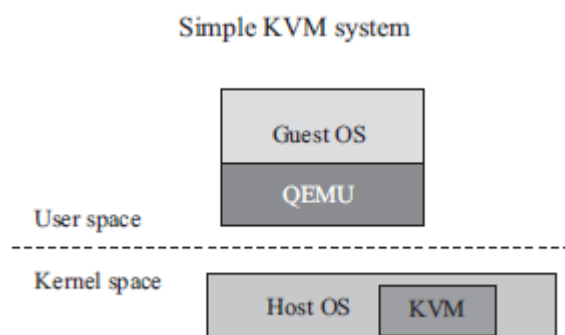
- Procesor ogólnego przeznaczenia i zaawansowany algorytm przetwarzania do przetwarzania sygnałów w czasie rzeczywistym.
- Topologia międzysystemowa BBU o wysokiej przepustowości, niskim opóźnieniu i niskich kosztach wśród fizycznych zasobów przetwarzania w puli pasma podstawowego. Obejmuje on połączenie między układami w BBU, pomiędzy BBU w stelażu fizycznym, a także pomiędzy wieloma stelażami.
- Zdolność tradycyjnej platformy COTS do spełnienia wymagań RAN w czasie rzeczywistym. Obejmuje to optymalizację różnych kluczowych modułów, w tym systemu operacyjnego, hipernadzorca, operacji we / wy, sieci przełączników, systemów zarządzania i tak dalej. Szczegółowe omówienie rozwiązań znajdziecie w następnym rozdziale.
- Wyzwaniem jest również badanie ziarnistości wirtualizacji. Ze względu na różnice w protokołach komunikacyjnych wymagania dotyczące zasobów przetwarzania dla GSM, TD-SCDMA i TD-LTE różnią się znacznie pod względem możliwości obliczeniowych i wydajności w czasie rzeczywistym. W praktyce należy rozważyć różne różnice w wirtualizacji. Na przykład w praktycznych wdrożeniach maszyna wirtualna może obsługiwać tylko jednego przewoźnika w ramach BBU lub nawet mniejszych jednostek. Istnieje inne podejście warstwowe, w którym maszyna wirtualna obsługuje tylko jedną warstwę, taką jak L1, L2 lub L3, lub nawet pewną część warstwy nośnej.

#### 10.4 Najważniejsze technologie i rozwiązania wirtualizacji RAN

Aby osiągnąć wirtualizację RAN, należy zająć się wieloma wyzwaniami technologicznymi. Z punktu widzenia sprzętu i oprogramowania platforma wirtualizacji powinna mieć określone ulepszenia i modyfikacje. Badania i doświadczenie wyraźnie pokazały, że standardowe rozwiązania informatyczne klasy korporacyjnej i korporacyjnej nie zapewniają determinizmu, wydajności i niezawodności wymaganych do skutecznej implementacji RAN - wymagane jest specjalnie zbudowane, zoptymalizowane rozwiązanie. Następujące kluczowe technologie są ważne dla osiągnięcia skutecznej wirtualizacji RAN.

##### 10.4.1 Rozszerzenie na hostowym systemie Linux

Najpierw wprowadzimy Wirtualną Maszynę (KVM) opartą na jądrze. KVM jest przeznaczony dla architektur x86 i x86\_64, aby korzystać z natywnego wsparcia dla wirtualizacji sprzętowej, podczas gdy KVM obsługuje również inne architektury procesorów, takie jak ARM i Power. KVM opiera się na emulacji Quick Emulator (QEMU), aby zapewnić kompleksowe rozwiązanie wirtualizacji. Proste użycie dla KVM jest pokazane na rysunku 10.3.



Wirtualizacja RAN jest bardziej złożonym scenariuszem użycia niż niektórymi aplikacjami sieci rdzeniowej, ponieważ istnieje kilka maszyn wirtualnych lub systemu-gościa, a wydajność jest bardzo wrażliwa na rygorystyczne wymagania (trudne w czasie rzeczywistym, wymagające przetwarzania co 500 ms bez jittera). Podobnie, wydajność bez determinizmu jest bezużyteczna w tym kontekście.

Kompletne zwirtualizowane rozwiązanie musi zapewniać ten sam poziom wydajności i gwarantowaną przewidywalność dla każdego obciążenia związanego ze zwirtualizowaną aplikacją RAN. Należy zająć się następującymi ograniczeniami:

- Obsługa KVM w czasie rzeczywistym i QEMU.
- Lepsza izolacja gości dzięki technikom minimalizacji interferencji procesorów.
- Udoskonalona konfiguracja systemu, partycjonowanie i izolacja maszyn wirtualnych i środowisk z wykorzystaniem elastycznych funkcji skryptowania.
- Dobrze zdefiniowane konteksty bezpieczeństwa dla maszyn wirtualnych i środowisk.
- Obsługa pełnego NO\_HZ (jądro łaskocze). Zestaw poprawek do jądra, aby poprawić izolację określonego procesora ze źródeł zakłóceń systemu, takich jak Inter Processor Interrupts (IPI), tykanie zegara, odzyskiwanie pamięci i inne. Modyfikacje globalnych maszyn stanów (RCU) i usług (pomiar czasu) są wykonywane bez okresowego dostępu do rdzenia.

Ponadto wciąż istnieje wiele innych kluczowych parametrów systemu, które są nazywane opcjami w systemie Linux i mogą wpływać na wydajność w czasie rzeczywistym. W przypadku innego sprzętu i oprogramowania konieczne będzie wykonanie ustawień systemu. Opcje są przekazywane do jądra przez program rozruchowy grub i pojawiają się w pliku / proc / cmdline, gdy cel zakończy proces uruchamiania.<sup>1</sup> W przypadku jądra gościa są one przekazywane do binarnego Qemu-Kvm podczas uruchamiania czas. Poniżej wymieniono najważniejsze opcje, które mogą mieć wpływ na wydajność systemu w czasie rzeczywistym:

### **1. zegar źródłowy**

Służy do przesłonięcia domyślnego źródła zegara, a jiffie to źródło zegara cofania. W architekturze x86\_64 dostępne są opcje hpet i tsc. Podczas gdy QEMU może emulować źródło zegara HPET dla gościa, OVP używa tsc jako źródła zegara, aby uniknąć narzutu emulacji.

### **2. beczynny**

Służy do definiowania zachowania podczas uśpienia procesora, gdy jest on beczynny. Poprawne parametry to pool, mwait, stop i nomwait. Podczas korzystania z odpytywania, beczynne zadanie powoduje przebudzenie CPU przez okresowe pobieranie, uniemożliwiając przejście do trybu oszczędzania energii. Zmniejsza to opóźnienia związane z przełączaniem kontekstów kosztem większego zużycia energii. Podczas używania MWAIT, beczynne zadanie używa specjalnych instrukcji MONITOR / MWAIT, które osiągają prawie taką samą wydajność opcji odpytywania, ale przy mniejszym zużyciu energii. Wszystkie nowsze procesory Intel obsługują te instrukcje. Opcja nomwait pozwala CPU spać w stanach oszczędzania energii C, ale nie stosuje instrukcji MONITOR / MWAIT. Zwróć uwagę, że od czasu Linuksa 3.0 pojawiło się ostrzeżenie o czasie rozruchu po wywołaniu idle ¼mwait, co oznacza, że ta opcja zostanie usunięta w przyszłych wydaniach. Tak właśnie jest, ponieważ jądro 3.9. Jądra skonfigurowane za pomocą CONFIG\_ACPI=n i CONFIG\_INTEL\_IDLE=n po uruchomieniu na sprzęcie obsługującym MWAIT będą po prostu używać HLT. Jeśli MWAIT jest pożądany w tych systemach, te dwie opcje konfiguracji powinny być włączone.

### **3. Niepewność**

Służy do ustawienia domyślnej maski powinowactwa IRQ, czyli zestawu procesorów, które powinny przetwarzać przerwania. Poprawne parametry to lista, zakres lub kombinacja identyfikatorów procesora. Na przykład, jeśli na 16 rdzennych użytkownikach komputerów chcemy zapobiec

przerywaniu przejść do procesorów 1, 2 i 3. Mogliby użyć wartości zakłóceń =0,4-15 na linii startowej hosta. Opcja `irqaffinity` jest szczególnie wygodna w połączeniu z opcją `isolcpus`, aby w pełni wyizolować jeden lub więcej procesorów z planera jądra systemu Linux i podsystemów zarządzania przerwaniem.

#### **4. `isolcpus`**

Ta opcja służy do określenia jednego lub więcej procesorów w celu odizolowania od ogólnych algorytmów równoważenia i planowania SMP. Użytkownicy mogą przenosić proces na izolowany procesor lub z niego, korzystając z wywołań systemowych powinowactwa CPU lub poleceń zestawu zadań i `cpuset`. Numer procesora zaczyna się od 0, a maksymalna to liczba procesorów w systemie -1. Ta opcja jest również używana w wierszu poleceń jądra QEMU-KVM, aby wyizolować wirtualne procesory, które mają być używane do określonych zadań. Na przykład dla dwóch instancji gościa wirtualnego `isolcpus=1` pozostawiłoby zerowe wirtualne CPU dla zadań porządkowych, podczas gdy procesor wirtualny 1 będzie przeznaczony tylko dla tych zadań jawnie ustawionych do działania na nim. Ogólnie, uzyskanie pełnej izolacji procesora nie jest całkowicie możliwe. W szczególności niektóre zadania jądra, takie jak miękkie przetwarzanie IRQ przez wątek `ksoftirqd`, będą prawdopodobnie zawsze obecne. Lista `ksoftirqd` dozwolona dla procesora w jądrze jest zasadniczo tylko do odczytu.

#### **5. `rcu-nocbs`**

Służy do zapobiegania wykonywaniu procedur zwrotnych RCU w docelowych procesorach. Poprawne parametry to lista, zakres lub kombinacja identyfikatorów procesora. Połączenia zwrotne RCU to funkcje, które wykonują porządki po upływie okresu prolongaty RCU. W przypadku niektórych obciążeń liczba wywołań zwrotnych może być dość duża. Na przykład, aby zażądać, aby procesory 1, 2, 3 i 6 nie były używane do wykonywania wywołań zwrotnych RCU, użytkownicy mogą używać `rcu-nocbs=1-3,6` na linii startowej hosta. Nie jest możliwe przypisanie każdego procesora w systemie do listy bez wywołań zwrotnych; co najmniej jeden procesor, CPU 0, musi pozostać w trybie tradycyjnym lub przetwarzanie okresu próbnego RCU nie będzie działać poprawnie. Zazwyczaj zakres `rcu-nocbs` będzie pasował do parametrów `isolcpus`, aby dodatkowo poprawić stan izolacji docelowych procesorów.

#### **6. `rcu_nocb_poll`**

Służy do wymuszania trybu odpytywania na wątkach RCU, które obsługują wykonywanie wywołań zwrotnych RCU. Ta opcja nie ma parametrów. Kiedy procesor działa bez wywołań zwrotnych RCU, będzie osobny proces obciążony obsługą wywołania zwrotnego. Gdy proces ten się obudzi, pobierze listę wyjątkowych wywołań zwrotnych dla przypisanego procesora, poczeka, aż upłynie okres prolongaty RCU, a następnie przejdzie przez wszystkie wywołania zwrotne. Opcja rozruchu `rcu_nocb_poll` nakazuje temu procesowi odpytywanie od czasu do czasu za nową pracę. Tryb odpytywania przyczynia się do zmniejszenia obciążenia odciążonych procesorów, które wciąż wykonują wywołania zwrotne RCU, oszczędzając im starań, aby regularnie budzić procesy czyszczenia.

#### **7. `tsc`**

Służy ona do przyjmowania założeń dotyczących dostępności i zachowania licznika znaczników czasu. Użycie `tsc=perfect` pozwala uniknąć niektórych sprawdzeń wykonawczych wykonywanych na niezawodności TSC. Jest używany w hostach i gościach.

### **10.4.2 Poprawa komunikacji VM**

Komunikacja między VM jest bardzo ważna dla platformy wirtualizacji. Jedną z wyróżniających właściwości wirtualizacji jest możliwość izolowania maszyn wirtualnych współlokatorów na tej samej

fizycznej platformie. Te wirtualne maszyny często muszą się komunikować i wymieniać znaczną ilość danych. Wiele aplikacji RAN ma różne tryby działania w obrębie węzła, które muszą wymieniać wiadomości i dane podczas trwania połączenia głosowego lub transmisji danych. Ważne jest, aby dostawcy RAN mogli zminimalizować zmiany w oprogramowaniu, aby to samo oprogramowanie mogło działać na dedykowanym sprzęcie lub w maszynach wirtualnych, prowadząc do wymogu wysokiej wydajności przesyłania komunikatów między urządzeniami VM. Wydajność będzie wąskim gardłem, jeśli przepustowość lub opóźnienie nie spełni wymagań wirtualizacji RAN. Ponieważ RAN nie jest taki sam jak dotychczasowy system informatyczny, wymagania są niezwykle surowe.

Istnieje wiele rodzajów metod komunikacji między VM, w tym

- Technika współdzielonej pamięci
- Technika oparta na sieci
- \* Przełącznik wirtualny hosta
- \* Przełącznik asystenta sprzętowego

Tutaj skupimy się na komunikacji sieciowej. Główną przyczyną jest to, że korzystanie z komunikacji sieciowej wymaga bardzo niewielu modyfikacji maszyny wirtualnej.

Standardowy KVM i QEMU ma obsługiwane niektóre funkcje, takie jak

- Urządzenia Virtio. VirtIO to specyfikacja parowo-wirtualnych operacji we / wy na maszynie wirtualnej. Tradycyjnie hiperwizor emuluje prawdziwe urządzenia, takie jak interfejs Ethernet lub kontroler dysków, aby zapewnić maszynę wirtualną we / wy. Ta emulacja jest często nieefektywna.
- Vhost. Sterowniki vhost w systemie Linux zapewniają emulację urządzenia virtio w jądrze. Normalnie proces przestrzeni użytkownika QEMU emuluje dostęp I / O z gościa. Vhost umieszcza kod emulacji virtio w jądrze, usuwając przestrzeń użytkownika QEMU poza obraz. Dzięki temu kod emulacji urządzenia może bezpośrednio wywoływać podsystemy jądra zamiast wykonywania wywołań systemowych z przestrzeni użytkownika.

Obecnie dostępnych jest kilka wirtualnych przełączników. Podstawowym, domyślnym wyborem jest stabilny, ale słabo działający Open vSwitch, który jest open source. Istnieją również niektóre komercyjne przełączniki wirtualne, takie jak wydajny przełącznik Wind River Accelerated vSwitch, zintegrowany z platformą infrastruktury NFV Titanium Server.

Aby zapewnić wysoką przepustowość między sieciami VM, przełączniki wirtualne mogą mieć następujące funkcje:

- Obsługa pakietów przestrzeni użytkownika
- Dedykowany rdzeń procesora przypisany

Aby zapewnić sieć VM o małym opóźnieniu, przełączniki wirtualne mogą mieć następujące funkcje:

- Sterownik trybu pobierania
- Brak projektu systemu zamków

### **10.4.3 Ulepszanie w czasie rzeczywistym**



Aby osiągnąć wysoką wydajność i przewidywalność, dystrybucja systemu operacyjnego musi zostać zbudowana i przetestowana przy użyciu określonego zestawu parametrów konfiguracyjnych i tuningowych:

- Preempt-RT powinien obsługiwać jądro hosta i gościa: Zestaw łat jądra dla jądra Linux w celu poprawy determinizmu i zachowania w czasie rzeczywistym poprzez zmiany semantyczne blokad, modyfikacje algorytmów i zmiany sterowników. Wszystkie poziomy pierwszeństwa są obsługiwane do wartości PREEMPT\_RT\_FULL w zależności od przypadku użycia.
- Sterowniki i funkcje w czasie rzeczywistym: adaptacje typowych narzędzi i operacji do pracy w semantyce blokującej preempt-rt.
- Pełny NO\_HZ: zestaw poprawek jądra, które poprawiają izolację określonego procesora ze źródeł zakłóceń systemu, takich jak Inter Interruptor (IPI), tykanie zegara, odzyskiwanie pamięci i inne. Modyfikacje globalnych maszyn stanów (RCU) i usług (pomiar czasu) są wykonywane bez okresowego dostępu do rdzenia.
- Ulepszona obsługa KVM i KVM w czasie rzeczywistym: Oprócz podstawowej funkcjonalności jądra rteci, obsługa gości i hosta KVM została zoptymalizowana pod kątem opóźnień przerwania i izolacji procesu. Pozwala to maszynom wirtualnym na działanie z wydajnością zbliżoną do wydajności macierzystego BSP, z dodatkowym zabezpieczeniem, izolacją, abstrakcją, emulacją i przyspieszeniem przy jednoczesnym wykorzystaniu obsługi funkcji KVM, takich jak migracja na żywo

#### **10.4.4 Migracja na żywo**

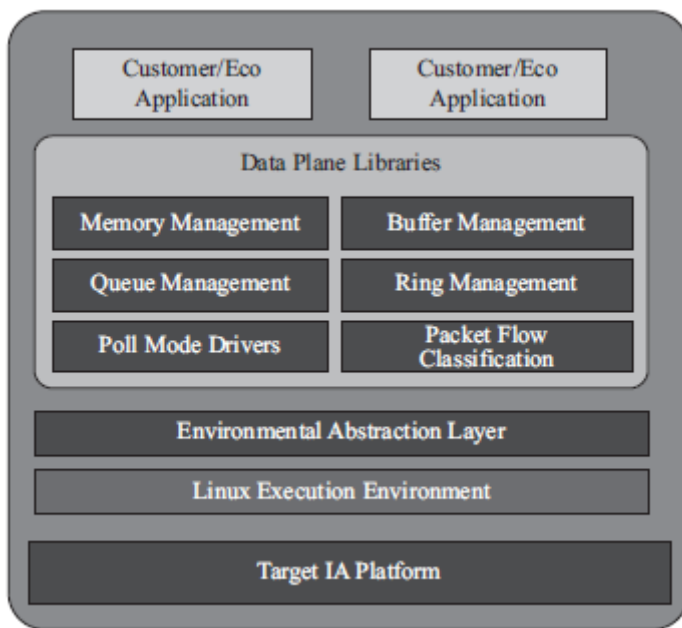
Migracja na żywo odnosi się do procesu przenoszenia uruchomionej maszyny wirtualnej lub aplikacji między różnymi maszynami fizycznymi lub między różnymi procesorami na tej samej maszynie fizycznej bez zatrzymywania maszyny wirtualnej. Pamięć, pamięć i łączność sieciowa maszyny wirtualnej są przenoszone z oryginalnego gościa maszyna do miejsca przeznaczenia. Migracja może być używana w następujących przypadkach użycia:

- Równoważenie obciążenia - goście mogą zostać przeniesieni do hostów z mniejszym użyciem, gdy host zostaje przeciążony.
- Sprzętowe przełączanie awaryjne - gdy urządzenia sprzętowe na hoście zaczynają się zawieszać, goście mogą zostać bezpiecznie przeniesieni, aby host mógł zostać wyłączony i naprawiony.
- Oszczędność energii - goście mogą być redystrybuowani do innych hostów i systemów hostów, aby oszczędzać energię i obniżyć koszty w krótkich okresach użytkowania.
- Migracja geograficzna - goście mogą zostać przeniesieni do innej lokalizacji w celu zmniejszenia opóźnień lub w krytycznych okolicznościach.

Aby uzyskać migrację na żywo, stan systemu-gościa na hoście źródłowym musi być replikowane na hoście docelowym. Wymaga to migracji stanu procesora, pamięci zawartość, pamięć lokalna i stan sieci. Na polu wirtualizacji RAN istotna jest przerwa w migracji na żywo. Migracja na żywo nie powinna wpływać na jakość użytkownika końcowego, a połączenia i dane nie powinny być odrzucane. Aby uzyskać najlepszą wydajność migracji na żywo, można zastosować specjalną technikę migracji pamięci. Coraz ważniejszym czynnikiem, który należy wziąć pod uwagę przy wyborze platformy wirtualizacji do wdrożenia vRAN, jest obsługa migracji na żywo aplikacji opartych na Intel Data Plane Development Kit (DPDK). Ponieważ coraz więcej aplikacji jest budowanych za pomocą DPDK, migracja na żywo dla tych aplikacji zostanie zablokowana, chyba że rozwiązanie wirtualizacji wyraźnie obsługuje tę funkcję.

#### 10.4.5 Przyspieszenie sieci

W celu przyspieszenia sieci skupimy się na zestawie do programowania płaszczyzn danych (DPDK). DPDK to zestaw bibliotek i sterowników do szybkiego przetwarzania pakietów. Został zaprojektowany do pracy na dowolnym procesorze. Pierwszym obsługiwanym procesorem był Intel x86, który został rozszerzony na IBM Power 8, EZchip TILE-Gx i ARM. Działa głównie w przestrzeni użytkownika Linuksa. Port FreeBSD jest dostępny dla podzbioru funkcji DPDK. DPDK jest projektem licencjonowanym na licencji Open Source BSD



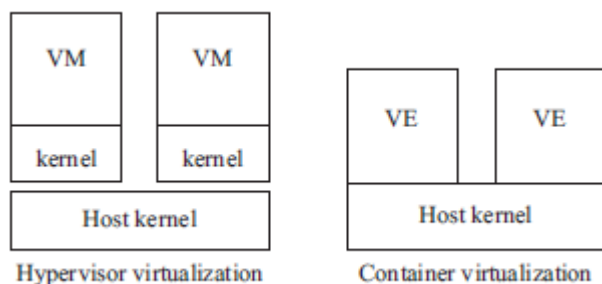
Głównym celem DPDK jest zapewnienie prostej, kompletnej struktury do szybkiego przetwarzania pakietów w zastosowaniach do płaszczyzn danych. DPDK implementuje model "run-to-completion" do przetwarzania pakietów, w którym wszystkie zasoby muszą zostać przydzielone przed wywołaniem aplikacji Płaszczyzny danych, działając jako jednostki wykonawcze na logicznych rdzeniach przetwarzania. Model nie obsługuje programu planującego, a do wszystkich urządzeń można uzyskać dostęp za pomocą odpytywania. Głównym powodem niewykorzystywania przerwań jest ogólny nakład pracy związany z przetwarzaniem przerwań. Oprócz modelu "run-to-completion" można również zastosować model potokowy, przekazując pakiety lub wiadomości między rdzeniami za pośrednictwem pierścieni. Pozwala to na wykonywanie etapów i pozwala na bardziej efektywne wykorzystanie kodu na rdzeniach. Przełącznik Intel DPDK vSwitch jest rozwidleniem wielowarstwowego wirtualnego przełącznika Open vSwitch. Reimplementuje moduł przesyłania dalej jądra (płaszczyznę danych), budując logikę przełączania w górnej części biblioteki Intel DPDK. Znacząco zwiększa to przepustowość przełączania pakietów i przybliża Open vSwitch do ogólnej wydajności komercyjnych rozwiązań vSwitching. Niestety ogólna implementacja jest wciąż niedojrzała, cierpiąca z powodu ograniczeń użyteczności i niezawodności. Chociaż nie jest jeszcze gotowy do użytku komercyjnego, wygląda obiecująco. Silnik spedycyjny zawiera tablice dużych tabel Intel DPDK, a demon kontrolny Open vSwitch jest modyfikowany w celu połączenia z tymi tabelami. Moduł przesyłania działa w przestrzeni użytkownika Linux z prawami licencyjnymi BSD. Intel DPDK vSwitch implementuje podzbiór funkcji przełączania Open vSwitch.

#### 10.4.6 Pojemnik

Kontenery są wynikiem wirtualizacji i separacji systemu operacyjnego. Zapewniają lekkie wirtualne środowisko, które grupuje i izoluje zestaw procesów i zasoby takie jak pamięć, procesor, dysk itp. z hosta i innych kontenerów. Podstawowe technologie kontenerowe obejmują:

- cgroups
- przestrzenie nazw
- chroot
- Linux Security Module (LSM), obowiązkowa kontrola dostępu (MAC)
- Możliwości / SECCOMP
- Tradycyjna wirtualizacja
- Runtime / management

Zrozumienie różnic między hiperwizorem a wirtualizacją kontenerów może lepiej przygotować użytkowników do projektowania i rozwijania wirtualizacji opartych na systemie końcowym. Dwa wirtualne węzły są przedstawione na rysunku 10.5



„jedna uruchomiona wirtualizacja hiperwizora z dwoma maszynami wirtualnymi i inna wirtualizacja kontenera z dwoma wirtualnymi środowiskami (VE). Wspólne, współdzielone jądro Linux zapewnia:

- Narzędzia do zarządzania przestrzenią użytkownika dla funkcji zabezpieczania systemu Linux.
- Izolacja zasobów sprzętowych (procesor, pamięć, blokowe operacje we / wy, sieć itd.) Bez konieczności korzystania z pełnych maszyn wirtualnych.
- Izolacja widoku aplikacji dla zasobów O / S (drzewa procesów, sieć, identyfikatory użytkowników i zamontowane systemy plików).

Technologia pojemników jest lekka niż wirtualizacja poziomu VM. Ponieważ żaden hiperwizor nie jest używany, wszystkie kontenery mają wspólne jądro hosta.

#### 10.4.7 CAT / CMT

Cache Allocation Technology (CAT) to nowa funkcja umożliwiająca hiperwizor systemu operacyjnego lub VMM, aby kontrolować przydział współużytkowanej pamięci podręcznej ostatniego poziomu procesora. Po skonfigurowaniu CAT procesor umożliwia dostęp do części pamięci podręcznej zgodnie z ustaloną klasą usług (COS). Procesor przestrzega zasad COS, gdy uruchamia wątek aplikacji lub proces aplikacji. Można to osiągnąć, wykonując następujące czynności:

- Ustal, czy CPU obsługuje funkcję CAT.

- Skonfiguruj COS, aby określić ilość dostępnych zasobów (przestrzeń pamięci podręcznej). Ta konfiguracja jest na poziomie procesora i jest wspólna dla wszystkich procesorów logicznych.
- Powiązać każdy procesor logiczny z dostępnym COS.
- Uruchom aplikację na procesorze logicznym, który używa żądanego COS.

Cache Monitoring Technology (CMT) to nowa funkcja, która pozwala systemowi operacyjnemu lub hypervisor lub monitorowi maszyny wirtualnej (VMM) na określenie użycia pamięci podręcznej dla aplikacji działających na platformie. CMT może być używany do wykonywania następujących czynności:

- Aby wykryć, czy platforma obsługuje tę funkcję monitorowania;
- W przypadku systemu operacyjnego lub maszyny wirtualnej, aby przypisać identyfikator dla każdej aplikacji lub maszyn wirtualnych, które są zaplanowane do uruchomienia na rdzeniu. Ten identyfikator jest nazywany identyfikatorem monitorowania zasobów (RMID);
- Aby monitorować obciążenie pamięci podręcznej na podstawie RMID, oraz
- W przypadku systemu operacyjnego lub maszyny wirtualnej odczytać zajętość pamięci podręcznej ostatniego poziomu dla danego RMID w dowolnym momencie.

#### **10.4.8 Obsługa OpenStack**

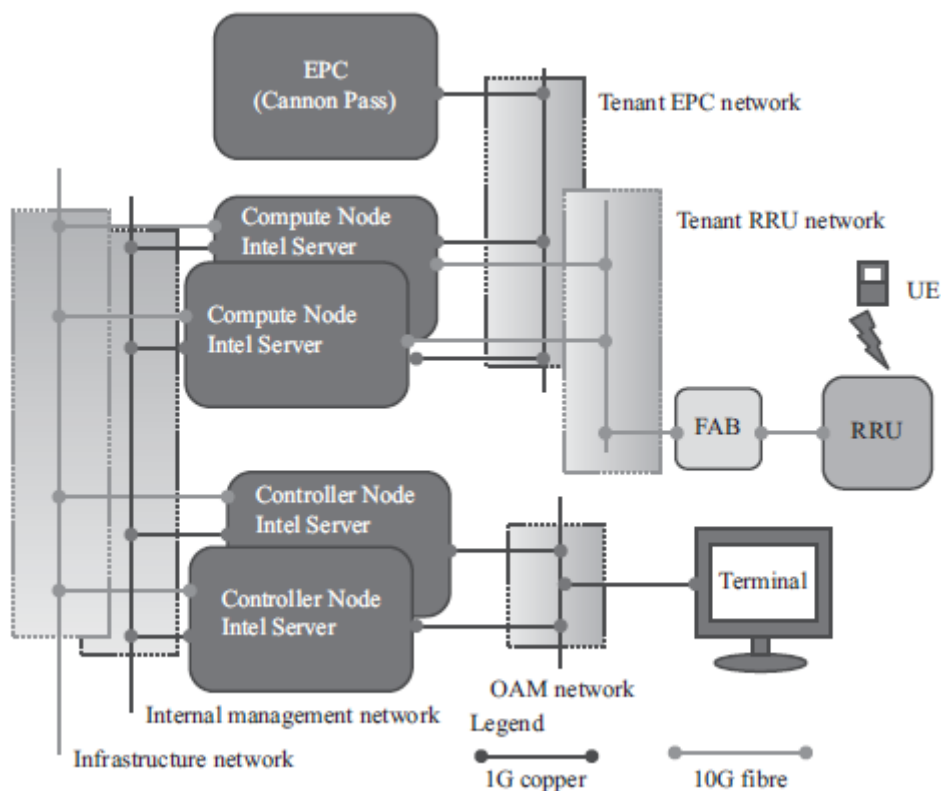
OpenStack [11] to darmowa i otwarta platforma oprogramowania do przetwarzania w chmurze, najczęściej wdrażane jako infrastruktura jako usługa (IaaS). Platforma oprogramowania składa się ze wzajemnie powiązanych komponentów kontrolujących puli sprzętowe przetwarzania, zasoby pamięci i sieci w całym centrum danych. OpenStack zawiera wiele podstawowych usług, takich jak Nova, Swift, Neutron itp., które mają kluczowe znaczenie dla wirtualizacji RAN.

#### **10.5 Proof-of-Concept rozwoju zwirtualizowanego RAN**

W celu oceny wykonalności wirtualizacji RAN utworzono prototyp z różnymi funkcjami optymalizacyjnymi zaimplementowanymi na platformie wirtualizacji, która została opisana w tej sekcji.

##### **10.5.1 Zwirtualizowany system demonstracyjny RAN**

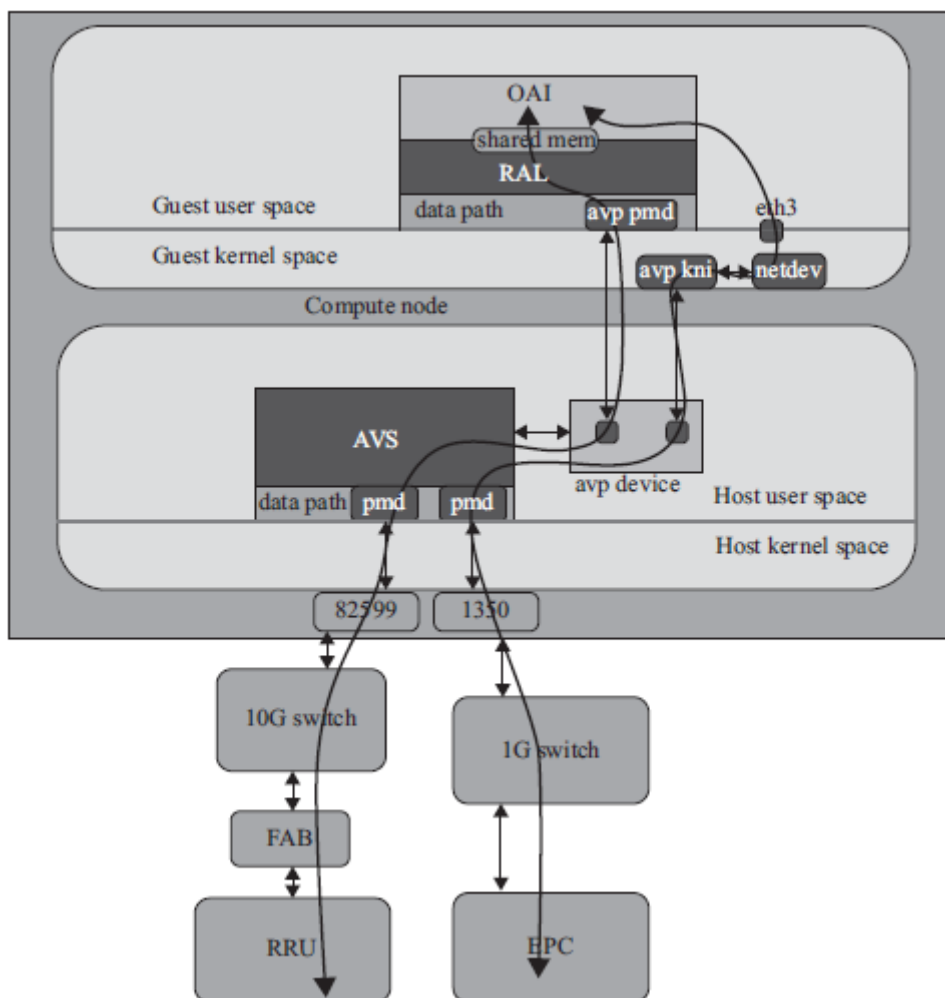
Rysunek 10.6 przedstawia topologię konfiguracji Proof-of-Concept (PoC) C-RAN.



Dwa węzły kontrolera działają w usługach Titanium Server w celu zarządzania infrastrukturą chmury. Dwa węzły obliczeniowe działają na serwerze Titanium, aby zapewnić komputer usługi i host maszyn wirtualnych. System wykorzystuje serwer Wind River Titanium jako infrastrukturę NFV z następującymi funkcjami:

- Automagiczne odzyskiwanie maszyny wirtualnej po awarii węzła obliczeniowego hosta (awaria węzła wykrywanie w sekundach zamiast w minutach typowych dla standardowej klasy IT)
- Automagiczne odzyskiwanie maszyny wirtualnej w przypadku awarii maszyny wirtualnej (wykrywanie awarii maszyny wirtualnej 60x szybciej niż w przypadku standardowego poziomu IT)
- Migracja na żywo maszyn wirtualnych za pomocą procesora Intel DPDK (nie dostępne w OpenStack klasy IT)
- Redundancja węzła sterownika i automagiczne przełączanie awaryjne (nie dostępne w wersji opartej na technologii IT OpenStack)
- Monitorowanie VM związane z kontrolą stanu aplikacji w obrębie gościa (zapewnia precyzyjne wykrywanie błędów aplikacji)
- Grupy zabezpieczeń VM (zapewniające tworzenie maszyn wirtualnych z tej samej grupy na różnych węzłach obliczeniowych)

Przepływ danych jest pokazany na rysunku 10.7.



W węzle obliczeniowym, Titanium Server Accelerated vSwitch (AVS) odpowiada za cały ruch Ethernet. Wysoko wydajny vSwitch przestrzeni użytkownika oparty na DPDK umożliwia wysoką wydajność komunikacji VM-VM bez potrzeby korzystania z jądra Linux, a także wysokowydajne przetwarzanie pakietów z karty sieciowej do aplikacji w maszynach wirtualnych. Ethernet z obsługą Intel DPDK, SRIOV oraz 1G, 10G i 40G zapewnia ultraszybkie przetwarzanie pakietów. Kluczowe moduły na rysunku obejmują:

- Accelerated Virtual Port (AVP) to przyspieszony interfejs sieci Ethernet typu TiS dla zwiększenia przepustowości.
- avp pmd to urządzenie, które jest sterownikiem trybu sond kompatybilnym z Intel DPDK.
- avp\_kni jest urządzeniem zgodnym ze standardowym urządzeniem sieciowym jądra Linux.

Każde wirtualne urządzenie Ethernet w maszynie wirtualnej będzie miało odpowiednie urządzenie w systemie AVS (takie jak "urządzenia avp"). Cały ruch jest dwukierunkowy. Weźmy jako przykład przykład dla uplink. Sygnał ze zdalnej jednostki radiowej jest przekształcany w pakiety Ethernet. Następnie przechodzi w przełącznik 10G. Gdy węzeł obliczeniowy 82599 NIC odbierze pakiety, zostanie on odpytany przez AVS. AVS podejmie decyzję w oparciu o MAC / IP, a następnie przekaże ją do dedykowanej wirtualnej karty sieciowej maszyny wirtualnej. Po zakończeniu procesu VM pakiety zostaną przekazane do sieci szkieletowej. Pakiety przechodzą do określonej wirtualnej karty sieciowej, a następnie wprowadzają ponownie AVS. AVS podejmie decyzję w oparciu o MAC / IP, a następnie

przekazuje pakiet do I350, kolejnej karty sieciowej. W tym eksperymentalnym środowisku serwer HP DL380 jest używany jako serwer, a inne elementy wyposażenia są wymienione poniżej:

- Serwer: HP DL380 Gen8 z 2 gniazdami, 8 rdzeni na gniazdo,
- Procesor: Intel (R) Xeon (R), E-2690 v2, na 3,00 GHz
- Karta sieciowa (NIC): Intel 82599
- Pamięć: 64 GB (węzeł obliczeniowy)

Oprogramowanie uruchomione w systemie jest wymienione poniżej:

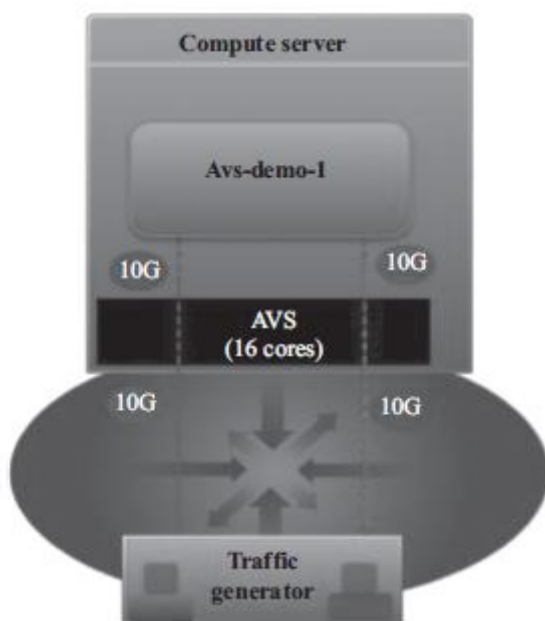
- WindRiver Linux OVP 6 (Open Virtualization Platform)
- Ulepszony kilogram OpenStack
- WindRiver AVS (Advance Vswitch)

### 10.5.2 Wyniki testu

Testy są przeprowadzane w dwóch scenariuszach za pomocą testera Ethernet. W każdym scenariuszu opóźnienie, jitter i przepustowość są odpowiednio mierzone.

#### I. scenariusz vloop

W scenariuszu pokazanym na rysunku 10.8 pakiety są wysyłane z testera Ethernet do systemu operacyjnego.

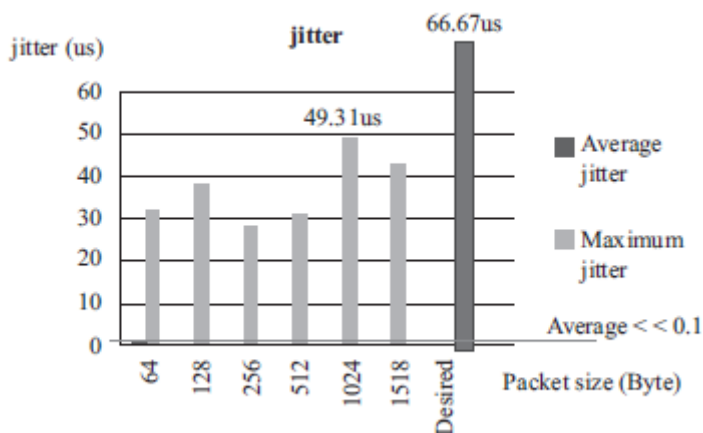
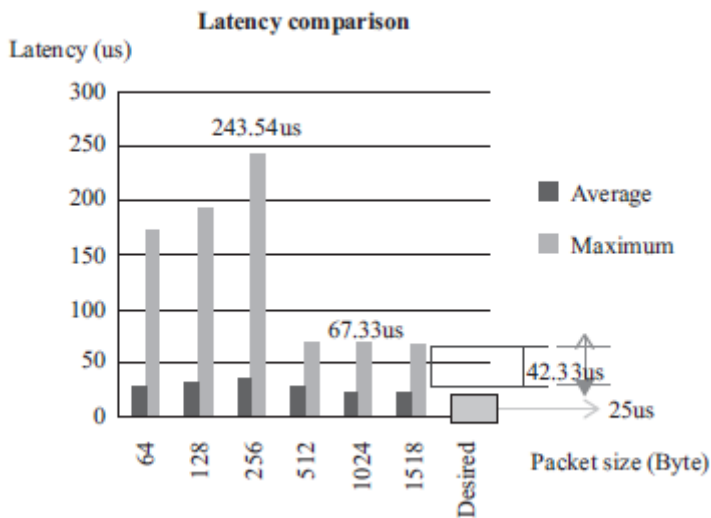


Następnie system hosta przekazuje pakiety do AVS. Gdy pakiety docierają do Avs-demo1, który jest maszyną wirtualną, proces przekazywania L2 odbywa się w ramach demo1. W końcu, po wysłaniu pakietów z vNIC demo1, wrócą oni do testera Ethernet i wykonają analizę pakietów. Przepustowość jest mierzona dla różnych rozmiarów klatki w środowisku wspomnianym powyżej, jak pokazano w Tabeli 10.1.

Rozmiar ramki (bajt) 64 128 256 512 1024 1518

Wydajność (%) 33,03 58,516 99,298 99,298 99,298 99,298

Zgodnie z tabelą, gdy rozmiar ramki staje się większy, przepustowość wzrasta. Powodem jest to, że w przypadku tej samej ilości danych potrzeba więcej pakietów dla mniejszego rozmiaru klatki. System będzie dokładać więcej starań, aby zdemontować pakiety. W rezultacie przepustowość jest niska, gdy rozmiar jest mały. W przypadku różnych rozmiarów ramek opóźnienie pętli pokazano na rysunkach 10.9 i 10.10.



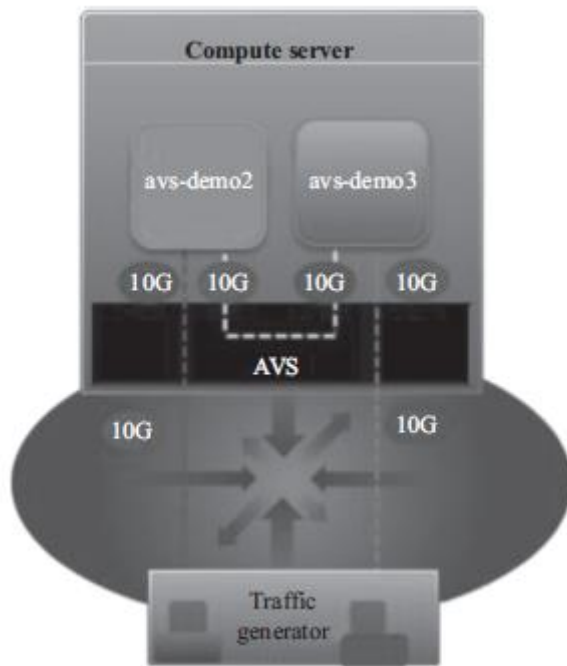
Wykres drgań względem wielkości ramy można zobaczyć na rysunku 10.10. Z rysunków 10.9 i 10.10 można zauważyć, że:

1. Przy zerowym poziomie utraty pakietów, gdy rozmiar pakietu jest nie mniejszy niż 256 bajtów, przepustowość może osiągnąć prawie pełną przepustowość 10 Gb / s.
2. Gdy rozmiar pakietu jest nie mniejszy niż 512 bajtów, opóźnienie transmisji pakietu pozostaje prawie stałe, z maksymalnym opóźnieniem wynoszącym 67.33us.
3. Wszystkie testowane drgania są mniejsze niż 66.67us, co spełnia wymóg komunikacji bezprzewodowej.

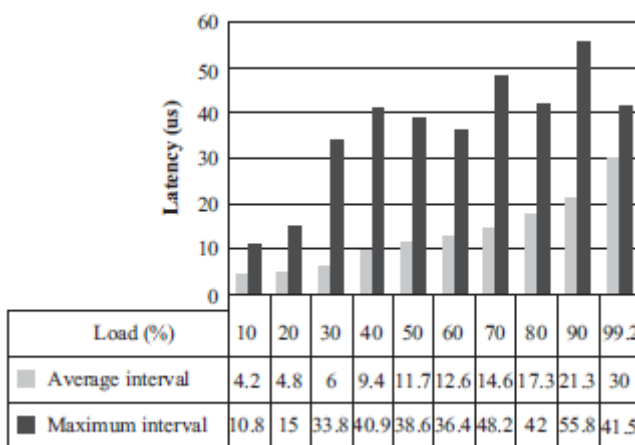
## II. Scenariusz modelu M.



Test ma na celu oszacowanie opóźnienia komunikacji VM, która ma kluczowe znaczenie dla migracji na żywo



W tym teście rozmiar ramki jest ustalony na 1024 bajty. Dane są generowane z generatora ruchu i wysyłane do demo2. Po przekazaniu pakietu L2-forward jest przekazywany do demo3. W odróżnieniu od scenariusza vloop, w tym scenariuszu, po dostarczeniu pakietów testowych do avs-demo2, docelowy adres MAC jest ustawiony jako inna maszyna wirtualna (avs-demo3). Następnie pakiety przechodzą z avs-demo3 z powrotem do systemu operacyjnego. Na koniec pakiety zostaną przechwycone przez centrum testowe Ethernet. Na podstawie opóźnienia scenariusza vloop i bieżącego scenariusza różnica między nimi jest mierzona jako opóźnienie komunikacji VM. Zgodnie z rys. 10.12 opóźnienie między maszynami wirtualnymi w dowolnym momencie jest mniejsze niż 60 sekund.



Opóźnienie to jest lepsze niż wydajność obecnych komercyjnych serwerów i spełnia wymagania dynamicznej migracji

## 10.6 Wniosek

W drodze do 5G wirtualizacja jest postrzegana jako jedna z kluczowych technologii, aby zmaksymalizować elastyczność, skalowalność i wydajność sieci, aby dostosować ją do różnych aplikacji i scenariuszy. Pomimo prostoty idei wdrożenia wirtualizacji w branży telekomunikacyjnej wdrożenie jest trudniejsze w praktyce. Komunikacja bezprzewodowa różni się od centrów danych IT tym, że komunikacja bezprzewodowa ma bardzo surowe wymagania dotyczące przetwarzania w czasie rzeczywistym. W tym rozdziale wirtualizacja w sieciach dostępu radiowego, w szczególności C-RAN, jest uważnie analizowana za pomocą analizy wyzwań i potencjalnych rozwiązań. Biorąc pod uwagę, że funkcje RAN, takie jak przetwarzanie pasma podstawowego, wymagają intensywnych obliczeń i przetwarzania w czasie rzeczywistym, wirtualizacja RAN wymaga następnie wspólnej optymalizacji różnych modułów systemowych, w tym systemów operacyjnych, hiperwizora itp. Wymaga to również ulepszenia i optymalizacji istniejącego tradycyjnego oprogramowania. Technologie informatyczne, w tym przyspieszenie sieci, migracja na żywo, komunikacja VM, wsparcie dla OpenStack i tak dalej. Opracowano prototyp zoptymalizowanej platformy do wirtualizacji o zwiększonej wydajności w czasie rzeczywistym. Testy wykazały, że zoptymalizowany obieg w obie strony L2 Opóźnienie przetwarzania platformy można zmniejszyć średnio do około 30, podczas gdy w tradycyjnym systemie Linux zwykle wynosi około 300-500. W międzyczasie opóźnienie komunikacji VM można zoptymalizować do około 10, w porównaniu do kilkuset mikrosekund dla tradycyjnego systemu Linux. Zwiększenie wydajności platformy w czasie rzeczywistym jest tylko pierwszym krokiem. W przyszłości należy zaangażować znacznie więcej zadań, takich jak zarządzanie w czasie rzeczywistym, szczegółowość wirtualizacji, przyspieszenie sprzętowe, projektowanie szybkich sieci przełączników o małej zwłóce itp. W celu realizacji całkowicie zwirtualizowanych systemów C-RAN