

PODSUMOWANIE REALIZACJI PROJEKTU ARCHITEKTURA KOMPUTERÓW 2

TEMAT:

Wirtualizacja - teoria a praktyka - wpływ na wydajność przetwarzania

Prowadzący:

mgr inż. Tomasz Serafin

Autor:

Bartosz Rodziewicz, 226105

1. Wstęp

1.1. Wstęp teoretyczny

Wirtualizacja to zagadnienie bardzo skomplikowane. O wirtualizacji mówimy, za każdym razem, gdy próbujemy pod konkretną architekturą bądź oprogramowaniem (takim jak system operacyjny) emulować funkcję lub polecenia innej.

W moim projekcie analizuję tylko i wyłącznie emulację pełną, czyli sposób wirtualizacji polegający na softwarowej emulacji wszystkich podstawowych elementów komputera na których działa system guest. Program emulujący jest widziany dla komputera hosta w postaci zwykłego procesu/procesów.

Zaletą takiego podejścia jest możliwość emulacji dowolnej architektury, co daje nam możliwość uruchomienia praktycznie dowolnego systemu/oprogramowania. Od dość oczywistych zastosowań jak uruchomienie drugiego identycznego systemu wewnątrz już pracującego zyskujemy możliwość emulowania całkowicie innego sprzętu – jak np. starszej konsoli do gier przez co zyskujemy możliwość zagrania w tytuły niedostępne na dzisiejszych platformach (aspekt legalności i moralności takiego działania to już osobny temat).

Drugą, chyba najważniejszą, z punktu widzenia wielu osób zaletą jest fakt, że emulowany w ten sposób system nie ma żadnej możliwości ingerencji w komputer hosta, co czyni takie rozwiązanie bezpieczną metodą testowania wielu rzeczy.

Kolejną zaletą, z uwagi na mocne komputery osobiste, jest możliwość jednoczesnego uruchomienia wielu maszyn na jednym komputerze i jednoczesny podgląd jak dana rzecz (np. tworzony program) zachowuje się pod różnymi systemami operacyjnymi.

Największą wadą takiego podejścia jest natomiast spadek wydajności emulowanej maszyny w porównaniu do systemu natywnego. W dzisiejszych czasach podejmowane są liczne działania, aby programy odpowiadające za wirtualizację działały jak najbardziej wydajnie, producenci sprzętu (głównie procesorów) wprowadzają technologie umożliwiające tym programom na bardziej bezpośredni dostęp do zasobów komputera hosta (np. technologie Intel VT-x lub AMD-V). Takie technologie umożliwiają maszynom wirtualnym na bardziej bezpośredni dostęp do fizycznego sprzętu (pozostawiając komputer hosta w pełni chronionym) czyniąc ich pracę wydajniejszą i powodując zatarcie granicy pomiędzy innymi metodami wirtualizacji.

1.2. Cel projektu

Celem projektu jest pokazanie czy i jeśli tak, to jak duży spadek wydajności następuje w momencie emulowania systemów operacyjnych na jednej fizycznej maszynie w porównaniu do systemu macierzystego.

2. Specyfikacja projektu

Do wykonania testów wykorzystałem 3 komputery, których wybór uzasadniony był dostępnością ich w moim domu:

1. Procesor: Intel i5 @2,4 GHz, RAM: 4GB, System hosta: Windows 10 Pro
2. Procesor: Intel Pentium G3258 @4.2GHz, RAM: 16GB, System hosta: Windows 10 Pro

3. Procesor: Intel i7-4700MQ @2.4GHz, RAM: 6GB, System hosta: Windows 10

Co do komputera nr 1 to tylko część scenariuszy została na nim przetestowana z uwagi na brak dostępności tego komputera w późniejszym etapie realizacji tego projektu (komputer należał do kolegi, który zrezygnował z kończenia tego projektu ze mną).

2.1. Środowisko testów i jego konfiguracja

Każdy z komputerów miał posiadać zainstalowany natywnie system Windows 10 Pro i Linux Mint 18 oraz przetestowane miały być różne scenariusze.

Każda maszyna wirtualna miała być uruchamiana za pomocą programu VirtualBox z domyślnymi ustawieniami. Program został wybrany ze względu na jego darmowość, popularność i prostotę użycia.

2.2. Sposoby testowania

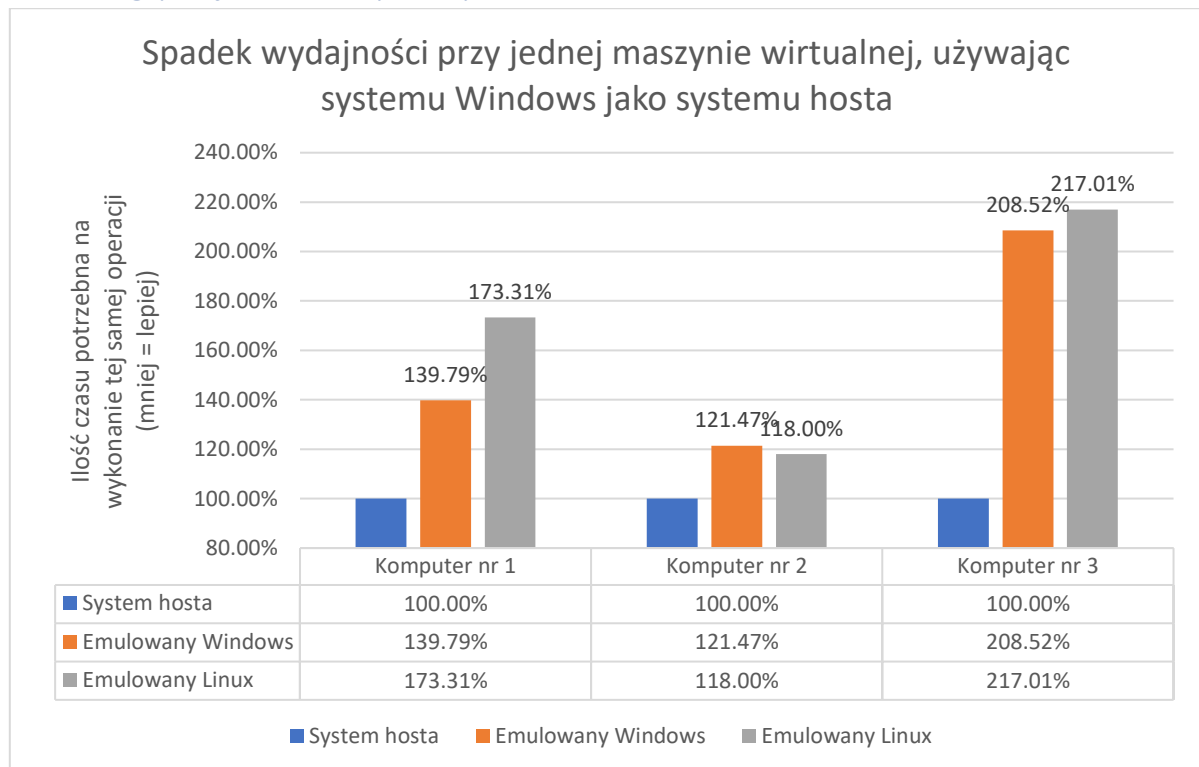
Przetestowane zostały następujące scenariusze:

- Natywny system Windows (jako punkt odniesienia)
- Windows -> Windows
- Windows -> Linux
- Natywny system Windows z dwoma jednocześnie uruchomionymi testami (jako punkt odniesienia)
- Windows -> 2x Windows (dwie jednocześnie działające maszyny wirtualne)
- Windows -> 2x Linux
- Windows -> Windows i Linux
- Natywny system Linux (jako punkt odniesienia)
- Linux -> Windows
- Linux -> Linux
- Natywny system Linux z dwoma jednocześnie uruchomionymi testami (jako punkt odniesienia)
- Linux -> 2x Windows (dwie jednocześnie działające maszyny wirtualne)
- Linux -> 2x Linux
- Linux -> Windows i Linux

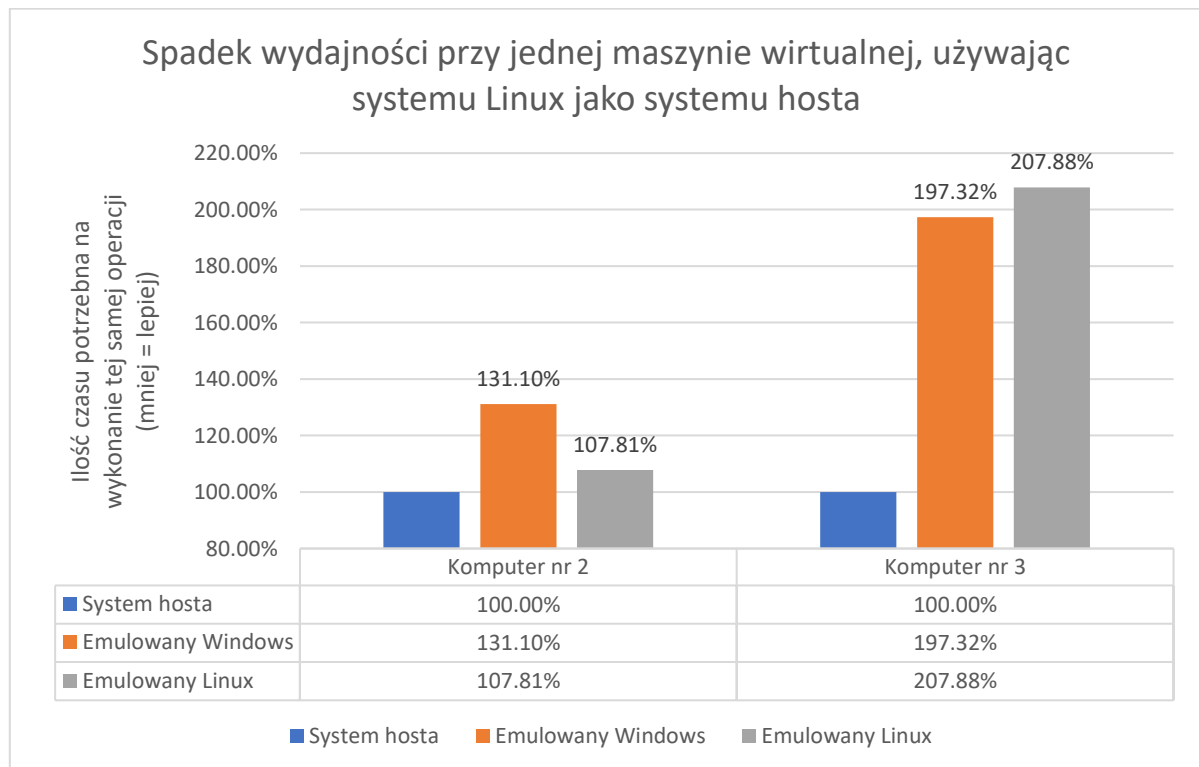
W każdym scenariuszu program benchmarkowy y-chrunch został uruchomiony na ostatnim systemie w łańcuchu, a system hosta trzymany był, aby generował jak najmniejsze obciążenie dla procesora.

Dla każdej maszyny wykonywałem taki sam test – dziesięciokrotne liczenie 50 milionów cyfr rozwinięcia dziesiętnego liczby pi.

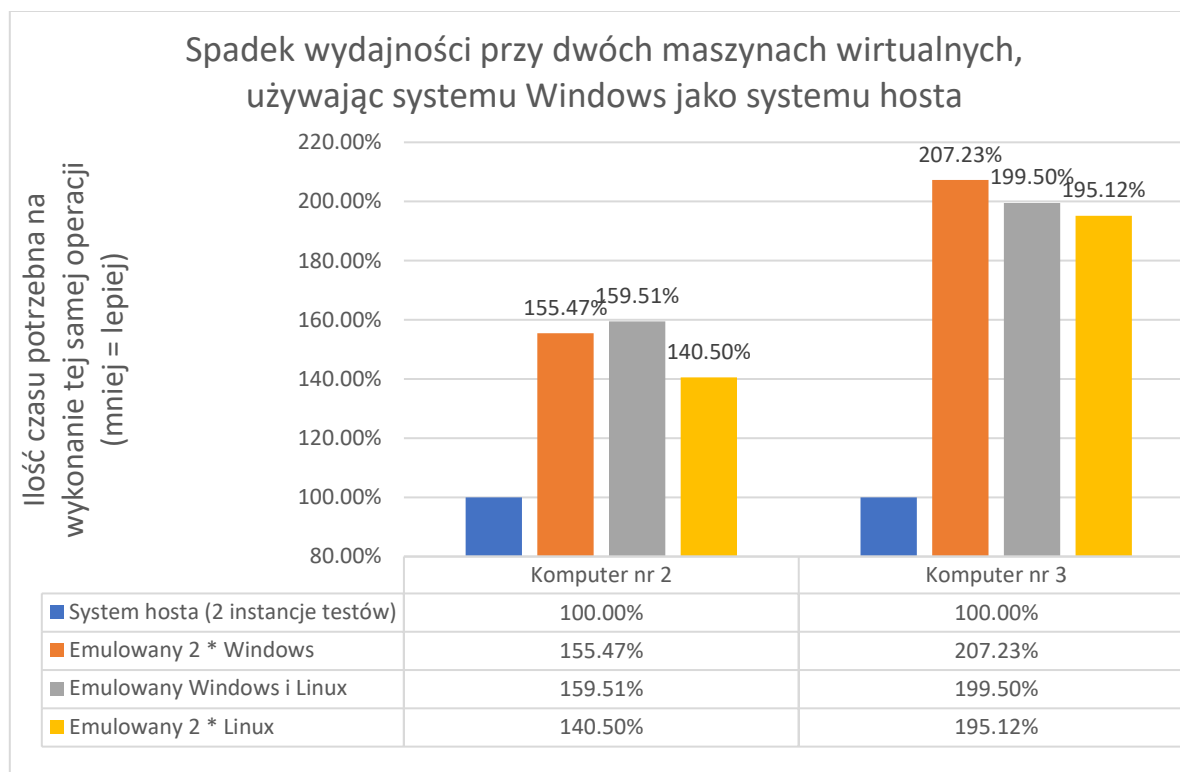
3. Przebieg projektu – wyniki pomiarów



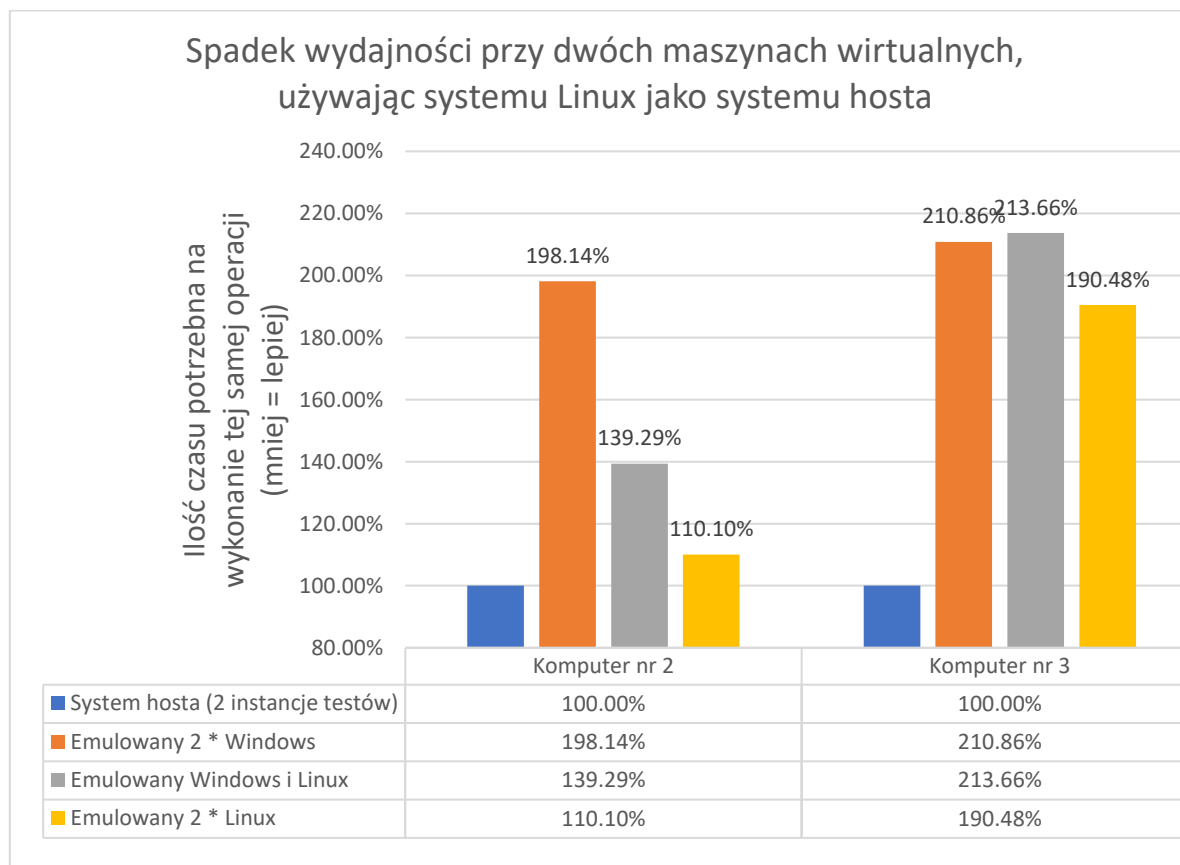
Wykres 1



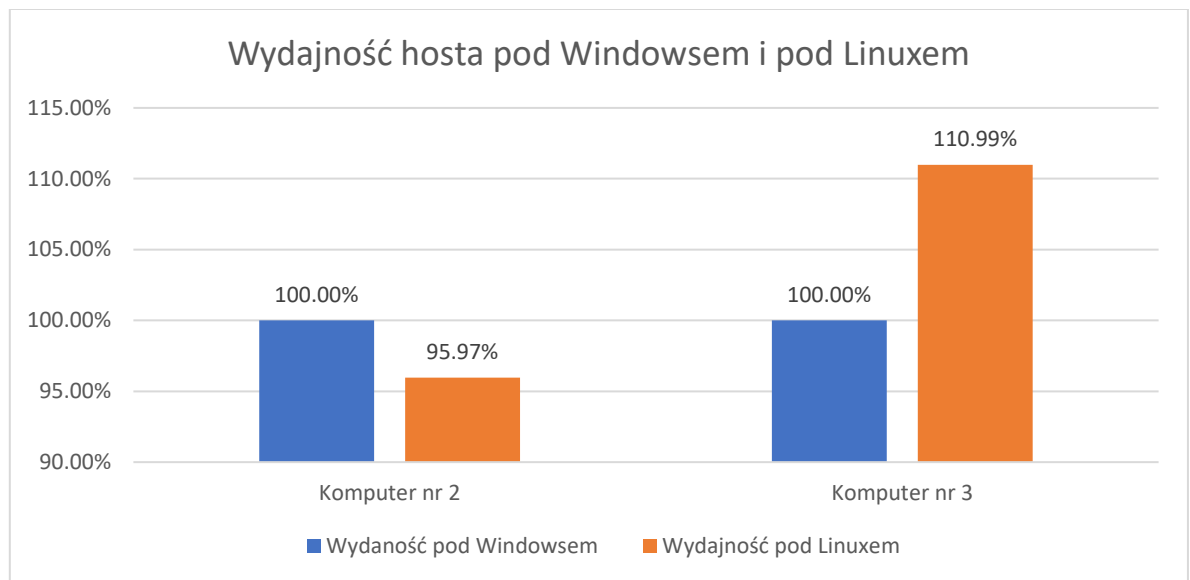
Wykres 2



Wykres 3



Wykres 4



Wykres 5

4. Analiza problemów

W trakcie realizacji projektu nie natrafiłem na zbyt wiele problemów, jednak kilka małych się pojawiło.

Głównym z nich był brak dostępu do komputera nr 1 w dalszej części realizacji projektu. Wyniki z niego zdecydowałem się umieścić tylko ze względu, że w tym jednym teście, który został na nim wykonany pasują do tych uzyskanych na innych urządzeniach.

Innym problemem, na który natrafiłem, była chęć przetestowania przeze mnie trzeciej warstwy wirtualizacji – chciałem wykonać testy, gdzie w maszynie wirtualnej odpalam kolejną maszynę. Próbowałem zrealizować ten scenariusz, jednak problemy jakie nastąpiły uniemożliwiły testowanie takiej konfiguracji:

- Maszyna z Windowsem w ogóle nie chciała wystartować z uwagi na brak VT-x. Wydaje mi się, że to problem Virtualbox, który nie do końca radzi sobie w takich warunkach.
- Maszyna z Linuxem zadziałała, jednak jej wydajność uniemożliwiała jakąkolwiek pracę na niej.
- Testy tej maszyny z Linuxem nie zostały jednak przeprowadzone ze względu na to, że bez technologii VT-x Virtualbox nie pozwala na uruchomienie maszyny 64bitowej, a niestety program benchmarkowy, który wybrałem (y-cruncher) nie posiada wersji działającej na 32bitowych systemach, a wyniki uzyskane innym programem nie pozwalały by na porównanie ich do wyników z innych scenariuszy.

Następnym na jaki natrafiłem była niemożliwość zainstalowania Linuxa, na komputerach testowych, tak aby nie ingerował w aktualny system operacyjny. Ten problem rozwiązałem korzystając z Linuxa w wersji Live na obu komputerach.

5. Analiza wyników – wpływ konkretnych systemów na wydajność

Wyniki pomiarów nieco mnie zaskoczyły, zwłaszcza w miejscach, których nie widać na wykresach. Zaczniemy jednak od analizy tych wykresów.

5.1. Spadek wydajności przy jednej maszynie wirtualnej, używając systemu Windows jako systemu hosta

Pierwszy wykres prezentuje wzrost czasu wykonywania testu, w porównaniu do uruchomienia go na komputerze hosta. Mamy tu 3 komputery, dwa z nich posiadają większą liczbę rdzeni (jeden dodatkowo Hyper-Threading), ale dość niskie taktowanie poszczególnych rdzeni, trzeci natomiast posiada tylko dwa rdzenie, ale prawie dwukrotnie wyższe taktowanie pojedynczego rdzenia.

Komputer drugi (ten, który posiada tylko dwa rdzenie) zanotował najmniejszy ogólnie spadek i emulowany Linux (system ogólnie wydajniejszy i lżejszy) działa na nim lepiej niż emulowany Windows.

Inaczej natomiast ma się sytuacja w przypadku w przypadku komputera nr 1 i komputera nr 3.

Komputer nr 1 (który posiada więcej rdzeni, ale nie ma Hyper-Thredingu) zanotował mocniejszy spadek i emulowany Linux działa na nim wyraźnie słabiej niż emulowany Windows.

Co do pierwszego przypadku jestem pewien, że jest to problem VirtualBoxa, który nie do końca radzi sobie z większą ilością procesorów i odpowiednim rozłożeniem użycia wirtualnych rdzeni na fizyczne odpowiedniki (a na pewno nie robi tego tak skutecznie, jak y-cruncher uruchomiony na systemie hosta).

Nie jestem natomiast w stanie wyjaśnić, dlaczego emulowany Linux działa słabiej na tym i na komputerze nr 3, jednak może mieć to duży związek (przynajmniej w przypadku komputera nr 3) z tym, że natywny Linux działa gorzej na nim, co pozwala mi przypuszczać, że Windows 10 lepiej radzi sobie z aplikacjami wielowątkowymi i ogólną obsługą większej ilości rdzeni (w tym również obsługą Hyper-Thredingu). Widać to na wykresie nr 5.

W przypadku komputera nr 3 zauważamy bardzo duży spadek wydajności przy maszynach wirtualnych (ponad dwukrotny) i wydaje mi się, że głównym powodem jego, jest problem Virtualboxa z obsługą dużej ilości rdzeni i Hyper-Thredingu. Test został wykonany przy ustawionych 4 emulowanych rdzeniach dla maszyny wirtualnej. Z uwagi, że komputer ten posiada Hyper-Threding próbowałem ustawić, aby emulowane były wszystkie 8 wirtualnych rdzeni (mimo ostrzeżeń VirtualBoxa), jednak wyniki były jeszcze słabsze, więc pomysł zarzuciłem.

5.2. Spadek wydajności przy jednej maszynie wirtualnej, używając systemu Linux jako systemu hosta

Analiza tego wykresu nie będzie znacząco różnić się od poprzedniego, ponieważ system hosta znacząco nie wpływa na wynik pomiarów. W tym miejscu (jak i każdym kolejnym) nie posiadam już danych z komputera nr 1.

Można tutaj zauważyć, że Linux jest systemem lżejszym i lepiej dającym sobie radę w tym zastosowaniu.

Wyjątkiem jest jednak emulacja Windowsa, która patrząc na wykres wygląda na bardziej stratną to w porównaniu czystych danych (czyli uwzględnieniu wydajności systemu hosta) daje bardzo podobne rezultaty w obu przypadkach.

Emulacja Linuxa w obu przypadkach jest mniej stratna niż przy emulacji pod Windowsem.

5.3. Spadek wydajności przy dwóch maszynach wirtualnych, używając systemu Windows jako systemu hosta

Ten wykres nie pokazuje nam zbytnio nic nowego.

Większy spadek wydajności na komputerze nr 2 niż przy jednej maszynie rzuciłbym całkowicie na problemy Virtualboxa z zarządzaniem obciążeniem wielu rdzeni.

W przypadku komputera nr 3 spadek wydajności jest porównywalny jak przy jednej maszynie i wydaje mi się, że jest to całkowita zasługa Hyper-Threadingu i działania systemu operacyjnego, który rozłożył dwa procesy VirtualBoxa na osobne wątki.

W dalszym ciągu zauważalna jest tendencja, że Linux, jako lżejszy system działa wydajniej.

5.4. Spadek wydajności przy dwóch maszynach wirtualnych, używając systemu Linux jako systemu hosta

Ten wykres dla komputera nr 3 prezentuje praktycznie takie same wyniki, więc nie ma sensu ich omawiać.

Dla komputera nr 2 natomiast (komputera z mniejszą ilością rdzeni) zauważamy, że Linux (jako host) generalnie o wiele lepiej radzi sobie z rozłożeniem użycia dwóch maszyn z Linuxem, a kompletnie nie radzi sobie z rozłożeniem użycia dwóch maszyn z Windowsem.

5.5. Wydajność hosta pod Windowsem i pod Linuxem

Ten wykres w sumie dodałem tylko jako ciekawostkę, ponieważ sam w sobie za dużo nie wnosi do badania.

Zauważamy na nim jednak, to co wcześniej już wspominałem, że Linux, mimo bycia lekkim, gorzej radzi sobie na wielowątkowym komputerze.

5.6. Inne uwagi, wynikające z obserwacji, których nie dało się umieścić na wykresy

Mimo, że starałem się wszystkie moje obserwacje potwierdzić liczbami i wykresami to były pewne sprawy, których nie dałem rady.

I tak na przykład zauważyłem, że Virtualbox kompletnie nie radzi sobie z równomiernym rozłożeniem obciążenia, gdy uruchomione są jednocześnie dwie maszyny. Jedna maszyna potrafiła skończyć test dwa razy szybciej od drugiej, a w następnym uruchomieniu to ta pierwsza była dwa razy szybsza.

Generalnie miałem też wrażenie, że Windows pod Windowsem (jako host) działał płynniej i wygodniej od Linuxa, jednak gdy hostem był Linux sytuacja się odwracała i to Windows potrafił się przyciąć.

6. Wnioski

- Głównym wnioskiem z mojego testu jest spostrzeżenie, że dzisiejsze komputery posiadają na tyle szybkie procesory, że w codziennych zastosowaniach nawet praca z kilkoma uruchomionymi maszynami wirtualnymi (o ile komputer posiada wystarczającą ilość pamięci RAM) jest wygodna i nie zauważa się ogólnego spadku wydajności (przy 4 jednocześnie uruchomionych maszynach na komputerze nr 2 ogólna wydajność komputera była zadawalająca), a praca w 1-2 to całkowita swoboda.
- Jeśli jednak patrzymy na cyferki i suchą wydajność to lepiej radzą sobie komputery z wyższym taktowaniem pojedynczego rdzenia, ale mniejszą liczbą rdzeni, niż z większą liczbą, a niższym taktowaniem.
- Dodatkowo w przypadku, gdy chcemy emulować Linuxa lepiej jako system host wybrać Linuxa, natomiast dla Windowsa wybrać Windowsa.
- Do emulacji obu systemów naraz lepiej sprawdza się Windows od Linuxa.

7. Bibliografia

- <https://pl.wikipedia.org/wiki/Wirtualizacja>
- <https://en.wikipedia.org/wiki/Virtualization>
- <https://www.virtualbox.org/manual/>
- <http://www.numberworld.org/y-cruncher/>
- różne fora i portale (jak np. superuser.com)