JOURNAL of COMPUTER SCIENCES INSTITUTE

JCSI 6 (2018) 77-81
Wysłane: 2017-11-13
Przyjęte: 2017-11-16

Porównanie wydajności emulatora i fizycznego urządzenia z systemem Android w oparciu o algorytm szachowy

Kamil Litkowski*, Jakub Smołka
Politechnika Lubelska, Instytut Informatyki, Nadbystrzycka 36B, 20-618 Lublin, Polska

Streszczenie. W artykule omówione zostały zagadnienia dotyczące różnic wydajności między emulatorami Android Emulator w wersji 26.1.4 i BlueStacks App Player 3 oraz fizycznymi urządzeniami mobilnymi z systemem Android. Wydajność zostaje mierzona poprzez pomiar czasu wykonywania algorytmu szachowego. W artykule opisane zostały dotychczasowe badania związane z daną tematyką. Przedstawiona została także metoda badań, wyniki badań bazujących na algorytmach szachowych oraz wnioski z nich płynące.

Słowa kluczowe: emulator; android; wydajność

*Autor do korespondencji.

Adres e-mail: kamil.litkowski93@gmail.com

Performance comparison of an emulator and physical Android mobile device based on chess algorithm

Kamil Litkowski*, Jakub Smołka
Institute of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract. Article discusses performance differences of emulators Android Emulator version 26.1.4 and BlueStacks App Player 3 and physical devices with Android platform. Performance is measured by the chess algorithm execution time. Article also describes previous research related to this subject. The article also presents the used research methods, results of research based on chess algorithms and conclusions.

Keywords: emulator; android; performance

*Corresponding author.

E-mail address: kamil.litkowski93@gmail.com

1. Wstęp

W ostatnim czasie, ze względu na postęp technologiczny w zakresie miniaturyzacji, można zauważyć gwałtowny rozwój urządzeń mobilnych oraz systemów operacyjnych na nie przeznaczonych. Systemy te przenoszą część funkcjonalności komputerów do urządzeń mobilnych. Dane urządzenia mobilne zostały nazwane smartfonami (*ang. smartphone*). Niewielki rozmiar w połączeniu z mnogością funkcjonalności zaowocował ogromnym wzrostem popularności.

Najbardziej znanymi mobilnymi systemami operacyjnymi są: Android, iOS, Windows Phone oraz BlackBerry. Aktualnie najpopularniejszym z nich jest Android [1]. Jego popularność skutkuje rozwojem emulatorów pozwalających uruchamiać dany system oraz aplikacje przeznaczone na niego na innych urządzeniach. Emulatory w tym przypadku znajdują zastosowanie przy testowaniu tworzonych aplikacji oraz, coraz częściej, wykorzystywane są do korzystania z aplikacji nie mających odpowiedników na inne rodzaje urządzeń lub w celu wypróbowania aplikacji. W przypadku konsol lub innych urządzeń, których głównym celem jest rozrywka, główną funkcjonalnością emulatorów jest korzystanie na innych urządzeniach z aplikacji przeznaczonych tylko na urządzenia mobilne.

Do najbardziej popularnych darmowych emulatorów należą Android Emulator wykorzystujący Android Virtual Device (AVD), BlueStacks App Player, Andy oraz Genymotion [2]. Najpopularniejszym z nich jest Android Emulator zintegrowany ze środowiskiem programistycznym Android Studio. Pozwala na tworzenie wirtualnych urządzeń o wybranych parametrach sprzętowych oraz wybranej wersji systemu. Daje on możliwość przyspieszenia działania emulatora za pomocą specjalnych narzędzi, dostępnych za pomocą Android SDK, także zintegrowanego z Android Studio. Pozostałe wymienione emulatory nie oferują takiej elastyczności w tworzeniu wirtualnych urządzeń, jednak mogą konkurować pod względem wydajności, a do ich głównych zalet zaliczają się prostota i intuicyjność obsługi oraz dodatkowe funkcjonalności przydatne w niektórych zastosowaniach, np. wbudowana możliwość nagrywania filmów i strumieniowania w BlueStacks App Player. Godny uwagi jest także projekt Android-x86 umożliwiający uruchomienie Androida na urządzeniu opartym o architekturę x86.

Celem artykułu jest zbadanie wydajności obliczeniowej mierzonej poprzez czas wykonywania algorytmu szachowego dla wybranych emulatorów urządzeń z systemem Android w porównaniu do fizycznych urządzeń mobilnych przy szczególnym uwzględnieniu wysokiego wykorzystania

zasobów. Analiza zostaje dokonana przy wykorzystaniu algorytmu szachowego z rodziny mini-max cechującego się dużą ilością obliczeń niezbędnych do określenia wartości punktowej ruchów w drzewie ruchów na podstawie aktualnego stanu planszy. Analiza została poprzedzona przeglądem literatury dotyczącej emulatorów urządzeń z systemem Android oraz ich wydajności.

2. Przegląd literatury

Emulatory zawierają dużą część funkcjonalności fizycznego urządzenia, część aplikacji oraz umożliwiają instalowanie nowych [3]. Do najczęściej wykorzystywanych darmowych emulatorów należą między innymi Android Emulator korzystający z Android Virtual Device (AVD), BlueStacks, Andy oraz Genymotion [2].

Zagadnienie wydajności emulatorów podjął Mihai Neacsu w artykule „Benchmark time: Comparing Andy, AmiDuOS, Genymotion, BlueStacks". Stwierdzono w nim, że Andy występuje w wersji darmowej, BlueStacks oraz Genymotion są dostępne w wersji freemium – część funkcjonalności jest płatna, natomiast AmiDuOS jest płatny, a do darmowego korzystania przeznaczona jest jedynie trzydziestodniowa wersja próbna. Wydajność zbadano za pomocą dwóch programów: AnTuTu Benchmark oraz 3D Mark. Oba pozwalały określić szczegółowe wyniki pod kątem wybranych aspektów działania, m. in.: wydajności procesora oraz pamięci RAM, przetwarzania grafiki 2D oraz 3D, odczytu i zapisu danych oraz działania bazy danych. W obu aplikacjach najlepszym emulatorem okazał się Andy. Jednak biorąc pod uwagę składowe wyniki, nie można dokonać tak jednoznacznej oceny, gdyż każdy z badanych emulatorów okazał się najlepszy w co najmniej jednej z badanych kategorii. Wynika z tego, że w zależności od zastosowania najlepszy okazywał się inny z emulatorów. Zaskakująco największą liczbę najlepszych pojedynczych wyników uzyskał Genymotion, plasujący się w zbiorczych testach na przedostatniej oraz ostatniej pozycji [4].

Lauren Darcey oraz Shane Conder omawiają problem wydajności emulatorów w artykule „Supercharge Your Slow Android Emulator". Wskazane zostają tam powody niskiej wydajności wirtualnych urządzeń oraz omówione zostają sposoby na znaczące jej ulepszenie. Jako przyczyna zwiększenia wydajności zostaje wskazane przeniesienie systemu Android na architekturę x86 przez Intel, co pozwoliło na opracowanie obrazów systemu opartych o nią, a w efekcie pozwalających na znacznie szybsze działanie na urządzeniach o nią opartych niż w przypadku emulacji architektury ARM. Do faktycznego przyspieszenia działania konieczne okazało się także opracowanie specjalnego sterownika. Został on stworzony przez firmę Intel i otrzymał nazwę Intel Hardware Accelerated Execution Manager (HAXM). W dalszej części artykułu omówiono zasady działania tego sterownika oraz przedstawiono konfigurację krok po kroku, a następnie przeprowadzono badania porównujące wydajność emulatorów opartych o architektury x86 oraz ARM. Wykazano wielokrotną różnicę w szybkości działania. W kolejnej części artykułu przeprowadzone zostały badania zawierające również pomiary przeprowadzone na fizycznych urządzeniach. Jednak mogły zostać wykorzystane jedynie w celach poglądowych z powodu brakuporównania urządzeń pod względem parametrów sprzętowych, a w efekcie brak miarodajności danych pomiarów [5].

Eugene Shih w artykule „Running Android emulator up to 16x faster on AWS or Google Cloud: performance benchmarks" oraz Manon Lumeau w „How to Speed up the Android Emulator by up to 400%", podobnie jak autorzy powyższego artykułu, opisuje sposób na przyspieszenie działania emulatora oraz porównuje wydajność różnych emulatorów i fizycznych urządzeń. Jednakże prezentuje w tym celu inne podejścia. W badaniach, podobnie jak w poprzednim przypadku, nie zostały uwzględnione różnice sprzętowe między badanymi urządzeniami [6, 7].

Wart odnotowania jest także artykuł „Performance analysis of selected hypervisors (Virtual Machine Monitors VMMs)" autorstwa Adama Arciszewskiego oraz Waldermara Graniszewskiego. Jego celem było określenie wydajności darmowych narzędzi wirtualizacji w kilku wybranych aspektach: szybkości wykonywania obliczeń oraz odczytu i zapisu plików. Badania zostały przeprowadzone dla VirtualBox, VirtualPC oraz porównane z wynikami bez wirtualizacji. Pierwszym rodzajem testu było sprawdzenie szybkości obliczania wartości π oraz kodowania z wykorzystaniem algorytmu AES-256. Rezultatami drugiego badanego przypadku były czasy odczytu oraz zapisu pliku o rozmiarze 1GB. Oba testy zostały powtórzone wielokrotnie, a oprócz średniego wyniku podane zostały również minimalna i maksymalna wartość oraz odchylenie standardowe. Wyniki badań zostały opracowane i zinterpretowane oraz wskazano przesłanki do użycia każdego z rozważanych rozwiązań [8].

Przeprowadzony przegląd literaturowy wskazuje, że tematyka wydajności emulatorów oraz maszyn wirtualnych była wielokrotnie poruszana. Jednak w przypadku emulatorów Androida brakuje obiektywnych badań uwzględniających różnice wydajności urządzeń mobilnych oraz opartych o architekturę x86. We wszystkich przytoczonych przypadkach nie zostały wzięte pod uwagę różnice sprzętowe pomiędzy nimi. Ich zbadanie może umożliwić projekt Android-x86 oferujący wersję systemu Android dedykowaną dla urządzeń o architekturze x86, dzięki czemu aplikacje przeznaczone na urządzenia mobilne można uruchomić bezpośrednio na komputerze, z wyłączeniem konieczności wirtualizacji.

3. Metoda badawcza

Analiza porównawcza wydajności urządzeń mobilnych oraz emulatorów odbywa się poprzez pomiar czasu wykonywania algorytmu szachowego. Do badań wybrana została taka metoda ze względu na dużą ilość obliczeń oraz możliwość sprawdzenia zachowania przy wykorzystaniu wielu wątków.

Badania przeprowadzono na dwóch urządzeniach mobilnych oraz dwóch urządzeniach opartych o architekturę x86, na których uruchomione zostają Android Emulator w wersji 26.1.4 oraz BlueStacks App Player 3.

Badania zostały podzielone na dotyczące aplikacji jednowątkowych oraz wielowątkowych, co skutkuje możliwością zbadania zależności wydajności emulatorów i fizycznych urządzeń w odniesieniu do liczby działających wątków. Pozwala to także na sprawdzenia działania emulatorów w obliczu większej konkurencji wątków – m.in. w przypadku przekroczenia przez liczbę wątków aplikacji liczby wątków procesora. Badania aplikacji wielowątkowej odbywają się w przypadku Android Emulator dla jedynie dwóch rdzeni. Jest to spowodowane fazą eksperymentalną tej funkcjonalności – na obecną chwilę możliwe jest ustawienie maksymalnie takiej liczby. W celu ograniczenia losowości pomiarów, badania zostały przeprowadzone dziesięciokrotnie dla każdego z przypadków. Pod uwagę wzięta została głównie uśredniona wartość pomiarów. Wszystkie czasy wyrażono w milisekundach.

3.1. Etapy badania

Pierwszym etapem badania jest określenie różnic sprzętowych między urządzeniami mobilnymi oraz urządzeniami opartymi o architekturę x86 dla wszystkich przypadków. Odbywa się to za pomocą projektu Android x86, który umożliwia korzystanie z systemu Android na urządzeniu opartym o architekturę x86.

Drugim etapem są pomiary przeprowadzone dla emulatorów, a następnie porównanie ich z wynikami otrzymanymi w trakcie testów urządzeń mobilnych oraz przy wykorzystaniu projektu Android-x86 [9]. Rzeczywiste spowolnienie emulatorów względem maszyn fizycznych pozbawione wpływu różnic wydajności urządzeń zostaje obliczone poprzez podzielenie wyników otrzymanych dla emulatorów przez otrzymane dla projektu Android-x86, przy założeniu, że Android-x86 wykorzystuje pełne możliwości urządzeń i że urządzenie mobilne byłoby równie wydajne jak to oparte o architekturę x86.

3.2. Aplikacja testowa

W aplikacji testowej wykonywany jest algorytm szachowy w przypadku pięciu różnych scenariuszy początkowych (stan początkowy planszy), a następnie sumowane ich czasy wykonania. Aplikacja zostaje wykorzystana w dwóch wariantach o różnym progu odcięcia ruchów, czyli różnicą między aktualnie przetwarzanym oraz najwyższym możliwym wynikiem, po przekroczeniu której badany ruch zostaje odrzucony. Dla obu wariantów te wartości wynoszą 10 oraz 30. Ma to znaczący wpływ na liczbę badanych gałęzi ruchów, a w efekcie na liczbę zadań. Ponadto aplikacja zostaje wykonana w wersjach jedno-, dwu-, cztero- oraz ośmiowątkowej.

Użyty w badaniu algorytm bazuje na algorytmach z rodziny mini-max [10]. Tworzone jest w nim drzewo ruchów i w przypadku każdego możliwego na podstawie stanu planszy określana jest wartość punktowa. Wartości są obliczane dla każdej figury, a następnie w przypadku gracza pierwszego dodawane, a w przypadku drugiego odejmowane od łącznego wyniku. Celem gracza pierwszego jest zmaksymalizowanie tej wartości, a drugiego zminimalizowanie. Ponadto, w celu zmniejszenia liczby przetwarzanych ruchów, ustalona jest różnica między aktualnym ruchem a maksymalną wartością dla danej tury, po przekroczeniu której dana gałąź nie jest dalej przetwarzana, ze względu na niską szansę osiągnięcia w niej najbardziej optymistycznego wariantu ruchu.

3.3. Badane urządzenia

Badane urządzenia podzielono na dwie grupy: z uruchamianym emulatorem oraz urządzenia mobilne. Ich parametry zostały ukazane w tabelach 1 oraz 2.

Tabela 1. Urządzenia z uruchamianym emulatorem

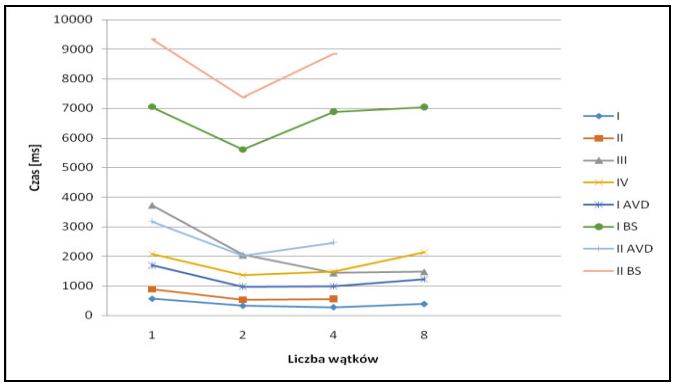
Numer	Procesor	RAM
I	i5-4460, 4 x 3,2GHz	8GB
II	i5-7200U, 2 x 2,5GHz	8GB

Tabela 2. Urządzenia mobilne

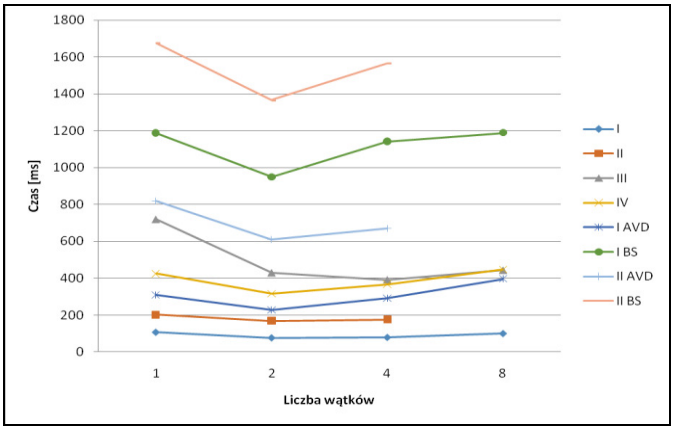
Nr	Wersja systemu	Model	Procesor	RAM
III	7.0	Huawei P9 Lite	HiSilicon Kirin 650 4 x 2,0 GHz + 4 x 1,7 GHz	2GB
IV	7.1	Xiaomi Mi5	Qualcomm Snapdragon 820 4 x Kyro 2,2 GHz	3GB

4. Analiza wyników badań

Zbiorcze wizualizacje średnich wyników w zależności od liczby wątków dla odpowiednio pierwszego oraz drugiego wariantu aplikacji zostały przedstawione na rysunkach 1 oraz 2. Liczbą rzymską zostały na nich oznaczone numery urządzeń, a skrótami AVD i BS rodzaj uruchomionego na nich urządzenia wirtualnego – odpowiednio Android Virtual Device będące wirtualnym urządzeniem tworzonym przez Android Emulator oraz BlueStacks.



Rys. 1. Wizualizacja wyników pierwszego wariantu aplikacji w zależności od liczby wątków



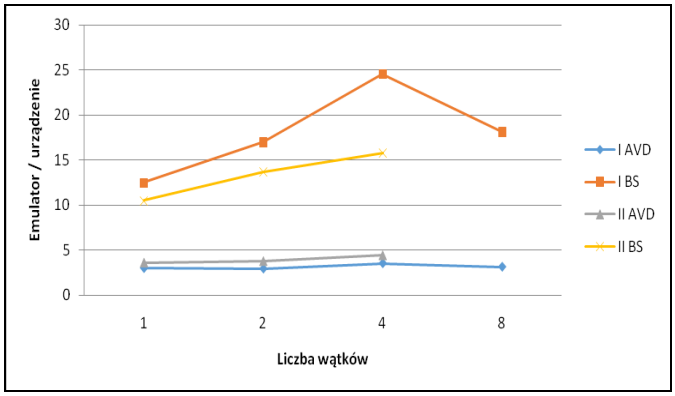
Rys. 2. Wizualizacja wyników drugiego wariantu aplikacji w zależności od liczby wątków

Na przedstawionych wykresach najlepiej widoczne są bezwzględne różnice wyników. Najbardziej negatywnie wyróżnia się BlueStacks App Player. Jest on wielokrotnie wolniejszy od urządzeń mobilnych nawet pomimo kilkukrotnie wyższej wydajności urządzeń w porównaniu do mobilnych. W przypadku Android Emulatora wyraźnie widoczny jest fakt niewiele gorszej lub, w niektórych przypadkach, nawet lepszej wydajności urządzeń mobilnych w porównaniu do wyników tego emulatora. Warte odnotowania jest także to, że wydajność fizycznych urządzeń względem emulatorów rośnie wraz ze zwiększaniem liczby wątków aplikacji.

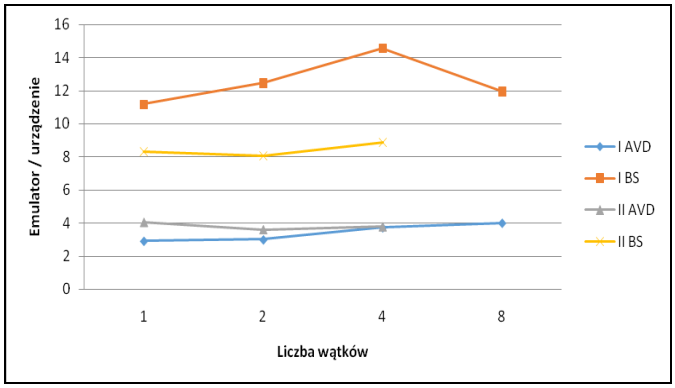
Wyniki dla obu wariantów aplikacji prezentują się dość podobnie. Bardzo zauważalna jest nieco mniejsza różnica w wydajności BlueStacks w stosunku do pozostałych wyników w przypadku drugiego wariantu oraz zdecydowanie bardziej zbliżone wyniki pozostałych przypadków w wariancie pierwszym. W drugim wariancie dużo traciły urządzenia mobilne ze względu na mniejszą liczbę wykorzystywanych wątków, a przez to brak wykorzystania pełni ich mocy. Natomiast korzystniej prezentowały się emulatory, co wskazuje na niezbyt dobre radzenie sobie ich z wielowątkowością.

Wizualizacje stopni pogorszenia wydajności na emulatorach z pominięciem różnic sprzętowych (współczynnik wyliczony poprzez podzielenie wyników dla emulatorów przez osiągnięte dla projektu Android-x86) w zależności od liczby wątków zostały ukazane na rysunkach 3 oraz 4. Znaczenie symboli na wykresach jest identyczne jak w przypadku rysunków 1 oraz 2.

Na wykresach można zauważyć, że Android Emulator okazał się być wolniejszy 3-4 krotnie od fizycznych urządzeń, a różnica okazała się stabilna niezależnie od wariantu aplikacji oraz liczby wątków. Różnice jedynie minimalnie wzrastały wraz z liczbą wątków oraz były nieznacznie wyższe przy mniejszej liczbie wykonywanych zadań, czyli w drugim wariancie aplikacji.



Rys. 3. Wizualizacja stopnia pogorszenia wydajności dla pierwszego wariantu aplikacji w zależności od liczby wątków



Rys. 4. Wizualizacja stopnia pogorszenia wydajności dla drugiego wariantu aplikacji w zależności od liczby wątków

Natomiast BlueStacks okazał się wielokrotnie mniej wydajny. Porównując do fizycznych urządzeń, był od 8 do nawet 25 krotnie wolniejszy. Warto zwrócić uwagę na ogromny rozrzut powiększający się wraz z liczbą wykorzystywanych wątków – pomijając aplikację ośmiowątkową, gdy różnica okazała się mniejsza niż w przypadku czterowątkowej, jednak większa niż przy dwuwątkowej. Ponadto różnica wydajności okazała się dużo mniejsza w przypadku słabszego urządzenia, co wynika z proporcjonalnie mniejszej różnicy wyników w przypadku emulatora niż w przypadku wybranych urządzeń fizycznych.

5. Wnioski

Przeprowadzone badania ukazały wielokrotną różnicę pomiędzy wydajnością fizycznych urządzeń mobilnych oraz emulatorów. Wybrane do badań urządzenia różniły się typem oraz wieloma parametrami, m.in.: mocą obliczeniową procesora oraz liczbą jego rdzeni i wątków, a także ilością pamięci RAM. Pozwoliło to na zbadanie różnic również w przypadku urządzeń o ograniczonym zużyciu energii, w przypadku gdy liczba wątków wykorzystywanych w aplikacji przekraczała liczbę wątków procesora oraz gdy liczba wątków procesora nie była w pełni wykorzystana.

Bezwzględne wyniki badań w przypadku Android Emulatora okazały się zbliżone do otrzymanych na kilkukrotnie mniej wydajnych urządzeniach mobilnych. Uniknięcie czynnika różnic w wydajności urządzeń pozwoliło

na stwierdzenie, że Android Emulator, w zależności od liczby wątków oraz wariantu aplikacji, okazał się 3-4 krotnie mniej wydajny od urządzeń fizycznych. Pozwala to stwierdzić, że obciążenie oraz wielowątkowość nie mają znaczącego wpływu na jego działanie.

Natomiast w przypadku BlueStacks App Player zauważyć można wielokrotną różnicę w stosunku do urządzeń mobilnych – nawet w przypadku bezwzględnych wyników. Przy uwzględnieniu różnic wydajności urządzeń, dysproporcja ta okazała się bardzo duża – dochodząca do niemal 25 razy.

Znaczącą rolę w wynikach badań odegrały liczba wątków oraz wariant aplikacji. Emulatory, a przede wszystkim BlueStacks, radziły sobie słabiej od fizycznych urządzeń z dużą liczbą wątków oraz ich synchronizacją. Znacznie lepsze wyniki, w porównaniu do urządzeń fizycznych, zostały odnotowane w przypadku mniejszej liczby wątków oraz przy znacznie mniejszej konkurencji w drugim wariancie. O ile w przypadku Android Emulatora różnice te nie były aż tak znaczące, o tyle w przypadku BlueStacks były wielokrotne.

Literatura

[1] Vincent J.,99.6 percent of new smartphones run Android or Ios,https://www.theverge.com/2017/2/16/14634656/android-ios-market-share-blackberry-2016 [31.05.2017]

[2] Pinola M., How to Run Android on Your PC: The Best Android Emulators, https://www.laptopmag.com/articles/run-android-apps-on-pc [31.05.2017]

[3] Saha A. K., A Developer's First Look At Android. Linux For You, 01.2008.

[4] Neacsu M., Benchmark time: Comparing Andy, AmiDuOS, Genymotion, BlueStacks, http://www.download3k.com/articles/Benchmark-time-Comparing-Andy-AmiDuOS-Genymotion-BlueStacks-01443 [31.05.2017]

[5] Conder S., Darcey L., Supercharge your slow Android emulator, http://www.developer.com/ws/android/development-tools/haxm-speeds-up-the-android-emulator.html [31.05.2017]

[6] Shih E., Running Android emulator up to 16x faster on AWS or Google Cloud: performance benchmarks, https://www.ravellosystems.com/blog/android-emulator-faster-performance-aws-google/ [31.05.2017]

[7] Lumeau M.,How to Speed up the Android Emulator by up to 400%, https://doc.nuxeo.com/blog/speeding-up-the-android-emulator/ [31.05.2017]

[8] Arciszewski A., Gramoszewski W., Performance analysis of selected hypervisors (Virtual Machine Monitors - VMMs). INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2016, VOL. 62, NO. 3.

[9] http://www.android-x86.org [20.09.2017]

[10] Lee C.-I. C., The min-max algorithm and isotonic regression. The Annals of Statistics, 1983, Vol. 11, No. 2.