Piotr ARABAS[1], Michał KARPOWICZ[1]

Naukowa i Akademicka Sieć Komputerowa Instytut Badawczy (1)

doi:10.15199/48.2016.03.08

Wykorzystanie informacji z rejestrów procesora do identyfikacji modelu poboru mocy przez serwer

Streszczenie. *Model poboru mocy jest istotnym elementem rozwiązań zwiększających efektywność energetyczną serwerów sieciowych. Należy spodziewać się, że profile poboru mocy zależą od zastosowania danej maszyny. Przeprowadzenie szczegółowych eksperymentów pomiarowych wymaga odpowiedniego sprzętu – przede wszystkim miernika mocy. Tematem przedstawionej pracy jest próba znalezienia zależności między całkowitą mocą pobieraną przez serwer a mocą odczytaną z rejestrów* `msr` *procesora, Wyznaczenie takiej zależności pozwoli na stosunkowo szybkie dostrajanie generycznego modelu poboru mocy do konkretnego zastosowania i konfiguracji sprzętowej.*

Abstract. *Precise model of power consumption is crucial for improving energy effectiveness of network servers. Model construction requires numerous measurement experiments employing appropriate equipment – power meter and data collection system. The subject of paper is assessment if power statistics available via* `msr` *registers of some Intel processors may be correlated to total power consumed by the server. Finding such relation will allow to construct generic total power consumption model which may be easily tuned to suit specific application.****(Model of server power consumption based on msr registers data)***

Słowa kluczowe: Energooszczędne systemy komputerowe, modelowanie, identyfikacja
Keywords: Power saving in computer systems, modeling, identification

Wstęp

Problematyka ograniczenia zużycia energii przez serwery i urządzenia wykorzystywane w teleinformatyce staje się ostatnio coraz bardziej istotna [1, 2]. Dzieje się tak ze względów ekonomicznych i technicznych, nie bez znaczenia jest także wzrost świadomości ekologicznej. W związku z ciągle niewielkim wykorzystaniem źródeł odnawialnych, redukcja poboru mocy jest jedną z podstawowych metod pozwalających na zmniejszenie emisji gazów cieplarnianych. Należy przy tym pamiętać, że oprócz dostarczenia urządzeniom energii, konieczne jest również odprowadzenie wytwarzanego przez nie ciepła, co wiąże się z koniecznością instalacji urządzeń chłodzących, komplikujących konstrukcję serwerowni, pobierających dodatkową moc i generujących przez to pokaźne koszty.

W ostatnich latach producenci sprzętu znacząco poprawili efektywność energetyczną urządzeń. Dotyczy to przede wszystkim zastosowania nowych technologii w warstwie fizycznej – np. pasywnych sieci optycznych, czy doskonalszych procesorów [3, 4]. Wprowadzono też ulepszenia w protokołach i algorytmach działających na poziomie pojedynczego łącza czy urządzenia [1, 5]. Dalsze zmniejszenie poboru mocy wymaga zastosowania zaawansowanych algorytmów lokalnych – np. sterowania częstotliwością taktowania procesorem [6], a przede wszystkim sieciowych, tzn. koordynujących pracę wielu połączonych maszyn [7, 8, 9, 10, 11, 12]. Maszyny, o których mowa mogą pełnić funkcję serwerów dostarczających różnorodnych usług, a także stanowić część infrastruktury sieci jak zapory sieciowe czy rutery programowe.

Dla konstrukcji wspomnianych algorytmów konieczne jest posiadanie adekwatnego modelu poboru mocy. Model ten, szczególnie w przypadku optymalizacji sieciowej, powinien oddawać całkowity pobór mocy przez urządzenie, co wymaga wykonania dość skomplikowanych pomiarów w trakcie jego identyfikacji. W przypadku sterowania grupą heterogenicznych serwerów (np. farmą serwerów bazodanowych czy klastrem obliczeniowym), powinno się zidentyfikować modele dla wszystkich z nich, a przynajmniej dla wszystkich wariantów konfiguracji. Wymaga to wielokrotnego zestawienia eksperymentu pomiarowego z użyciem zewnętrznego miernika mocy.

Rozwiązaniem tego problemu może być wykorzystanie informacji udostępnianych przez nowsze wersje procesorów Intela za pomocą rejestrów `msr` [13]. W ten sposób, stosunkowo małym nakładem środków, można zdobyć szczegółowe informacje o poborze mocy przez procesor. Można przypuszczać, że pobór mocy przez cały serwer jest w pewien sposób skorelowany z poborem mocy przez procesor, który jest przecież głównym elementem maszyny. Wiele jednak wskazuje, że postać tej zależności może być różna w przypadku wykonywania przez serwer różnych zadań, angażujących w różnym stopniu zasoby i komponenty składowe. Celem przedstawionych prac jest sprawdzenie czy korelacja taka istnieje i w przypadku pozytywnej odpowiedzi na to pytanie podjęcie próby wyznaczenia stosownego modelu.

Koncepcja pomiaru – scenariusze pomiarowe

W celu określenia zależności miedzy mocą pobieraną przez serwer i mocą raportowaną przez procesor za pośrednictwem rejestrów `msr` konieczne jest przeprowadzenie jednoczesnych pomiarów. Ponieważ wynikiem prac ma być określenie czy i w jaki sposób rodzaj wykonywanych zadań wpływa na przebieg zależności między badanymi wielkościami przyjęto, że badania będą prowadzone dla trzech scenariuszy:

1. serwer obliczeniowy, wykonujący intensywne operacje arytmetyczne bez komunikacji sieciowej,
2. ruter programowy, przekazujący ruch sieciowy między swoimi interfejsami bez dodatkowego przetwarzania,
3. serwer przekodowujący strumień wideo.

Scenariusz pierwszy odpowiada sytuacji, gdy jedynym istotnie obciążonym elementem serwera jest procesor. Różnice w poborze mocy wynikają w tym przypadku głównie z możliwości dopasowania częstotliwości zegara procesora (czy też częstotliwości z jakimi taktowane są poszczególne bloki procesora) do natężenia wykonywanych zadań. W systemie Linux możliwe jest to dzięki działaniu w jądrze odpowiedniego sterownika (tzw. *frequency governor*) [14].

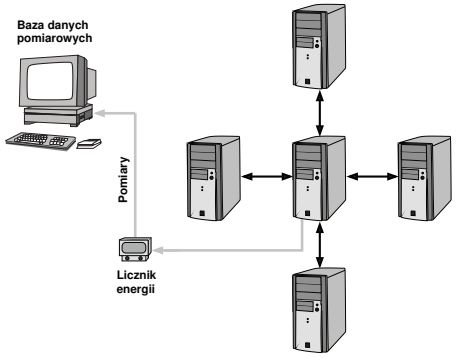
W scenariuszu drugim serwer pracuje jako ruter programowy, czyli jego zadaniem jest przekazywanie ruchu pomiędzy interfejsami. Przyjęto, że ruter nie będzie wykonywał zaawansowanego przetwarzania – np. analizy pakietów, znakowania lub usuwania niechcianego ruchu. W takim trybie działania najbardziej obciążonym elementem maszyny jest karta sieciowa. Należy zwrócić uwagę na fakt, że nowoczesne karty sieciowe są wyposażone w układ (tzw. *offload-engine*), którego zadaniem jest odciążenie procesora poprzez wykonanie części operacji związanych z przesyłaniem pakietów takich jak np. wyznaczanie sum kontrolnych czy segmenta-

cja. Tym niemniej część zadań, w szczególności związanych z wyższymi warstwami stosu protokołów, a więc np. znajdowanie odpowiedniej trasy w tablicy rutingu, jest wykonywanych przez procesor, co czyni z niego drugi najbardziej obciążony element rutera programowego.

Trzeci scenariusz stanowi próbę połączenia poprzednich – serwer przekodowujący strumienie wideo odbiera je na wybranym interfejsie sieciowym i wysyła przez drugi. Ponieważ przekodowanie strumienia wideo jest stosunkowo złożonym zadaniem obliczeniowym, to wydajność procesora stanowi wąskie gardło, tzn. prawdopodobnie nie jest możliwe pełne obciążenie interfejsów sieciowych ruchem. Tym niemniej stanowią one drugi, po procesorze i pamięci najbardziej obciążony komponent systemu i przez to scenariusz ten powinien przypominać poprzedni.

Koncepcja pomiaru – stanowisko pomiarowe

Na stanowisko pomiarowe składało się pięć komputerów klasy PC wyposażonych w procesory Intel i7, 8GB RAM, czteroportowe karty sieciowe Ethernet 1 Gb/s (Broadcom BCM5719), oraz typowy licznik energii elektrycznej z funkcją zdalnego odczytu wraz komputerem monitorującym jego stan oraz bazą danych, do której zapisywane były pomiary pobieranej mocy. Umożliwiało to śledzenie i archiwizowanie poboru mocy czynnej przez badany komputer. Użycie licznika energii zamiast wyspecjalizowanych mierników ogranicza rozdzielczość pomiarów do 1W, przy czym mogą one być wykonywane co około 8 s. Częstotliwość taka pozwala na zebranie dostatecznej liczy próbek przy czasie trwania eksperymentu rzędu pojedynczych minut. Pozostałe cztery komputery były wykorzystywane jako generatory ruchu – w scenariuszu 2, bądź też jako źródło i odbiornik strumienia wideo (patrz Rys. 1).



Rys. 1. Topologia połączeń stanowiska testowego.

Odczyt rejestrów `msr` wykonano za pomocą zmodyfikowanego programu `power_gov`[1]. Modyfikacja polegała na umożliwieniu zapisywania pomiarów w pamięci, przez co zmniejszono obciążenie procesora i uczyniono eksperyment bardziej wiarygodnym. Po zakończeniu eksperymentu pomiarowego i powiadomieniu o tym programu `power_gov` poprzez wysłanie odpowiedniego sygnału, możliwe było zapisanie pomiarów na dysk. Wielką zaletą wykonywania pomiaru mocy za pomocą rejestrów `msr` jest możliwość ich częstego odczytu – w analizowanym przypadku robiono to co 10 ms.

W scenariuszu 1 konieczne jest obciążanie procesora zadaniem obliczeniowym o zmiennej intensywności. Zrealizowano to za pomocą zmodyfikowanego programu `stress`[2], który może wykonywać jedno lub więcej równoległych zadań arytmetycznych, pozwalając na obciążanie kolejno wszystkich rdzeni procesora. Wadą programu `stress` jest fakt, iż obciąża on procesor w zawsze w tym samym stopniu wykonując serię instrukcji. Problem ten rozwiązano wstawiając w wewnętrznej pętli obliczeniowej instrukcje `sleep()` przez co możliwe jest okresowe wstrzymywanie wykonania wątku i zmniejszenie średniego obciążenia procesora.

Scenariusz 2 wymagał generowania ruchu sieciowego o odpowiedniej intensywności. Umożliwia to stosunkowo uniwersalny program `iperf`[3], który wykorzystano do generowania ruchu UDP. Użycie ruchu UDP upraszcza analizę wyników, gdyż pakiety generowane są z zadaną prędkością bez względu na sytuację panującą w sieci. W scenariuszu tym dwa z dodatkowych komputerów działają jako źródła ruchu, pozostałe zaś jako odbiorniki-analizatory, względnie w czasie generowania przepływów w dwóch kierunkach pełnią obie funkcje jednocześnie.

W przypadku scenariusza 3 do generowania strumienia wideo i następnie do jego przekodowania wykorzystano program `mencoder`[4]. Wysyłanie i odbiór strumienia przez sieć był możliwy dzięki wykorzystaniu polecenia `netcat`. Użycie pary programów `netcat` pozwoliło zestawić rodzaj tunelu między komputerami biorącymi udział w eksperymencie. Konsekwencją takiego sposobu przekazywania ruchu jest pominięcie części mechanizmów związanych z przetwarzaniem nagłówków, które działają, gdy realizowany jest scenariusz rutera programowego co może wpływać na wykorzystanie mechanizmów przyspieszających karty (*offloading*).

Eksperymenty pomiarowe – scenariusz 1 – serwer obliczeniowy

W celu zdjęcia charakterystyki odwzorowującej w możliwie pełny sposób zależność między mocą pobieraną przez procesor a całkowitym poborem mocy przez komputer wykonano szereg pomiarów przy różnych poziomach obciążenia procesora. Obciążenie, realizowane za pomocą zmodyfikowanego programu `stress` dobierano zmieniając dwie wartości: czas przerwy między kolejnymi porcjami obliczeń i liczbę jednocześnie uruchomionych wątków. W pierwszym przypadku wybrano 12 poziomów, pozwalających dla pojedynczego wątku, na obciążenie procesora na poziomie od około 3% do 100% (według wskazań programu `top`). Odpowiadające tym obciążeniom czasy bezczynności wątku obliczeniowego przedstawia tab. 1.

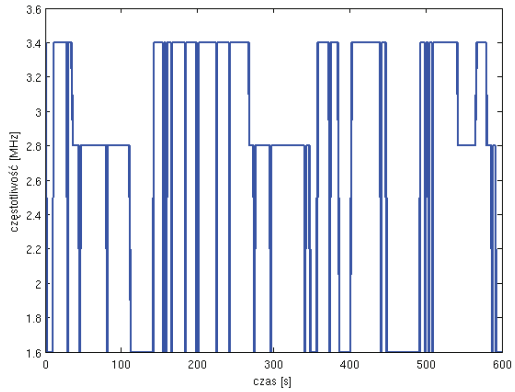
Tablica 1. Nastawy zmodyfikowanego programu `stress` w eksperymencie 1.

czas[μs]	96000	48000	24000	12000	6000	3000
obciążenie [%]	3	4	10	19	30	50
czas[μs]	2000	1200	700	160	10	0
obciążenie [%]	60	70	80	89	95	100

Podane wartości należy traktować orientacyjnie, w przypadku rozważanego pomiaru ich dobór nie jest krytyczny, wskazane jest jedynie aby pokrywały one w miarę równomiernie cały zakres obciążeń. Liczbę wątków zmieniano w zakresie od 1 do 8 – wynika to z faktu, że procesor i7 wyposażony jest w 4 rdzenie, na których może wykonywać po 2 wątki (*Hyperthreading*).

Przy tak prowadzonym eksperymencie system operacyjny dostosowuje częstotliwość taktowania procesora do obciążenia. Z charakterystyki standardowego sterownika `on-demand` wynikają stosunkowo częste skoki częstotliwości sięgające wartości maksymalnej (patrz. Rys. 2). Pomiary

[1] https://software.intel.com/en-us/articles/intel-power-governor
[2] http://people.seas.harvard.edu/~apw/stress/
[3] dostępny w standardowych dystrybucjach Linuksa
[4] http://www.mplayerhq.hu



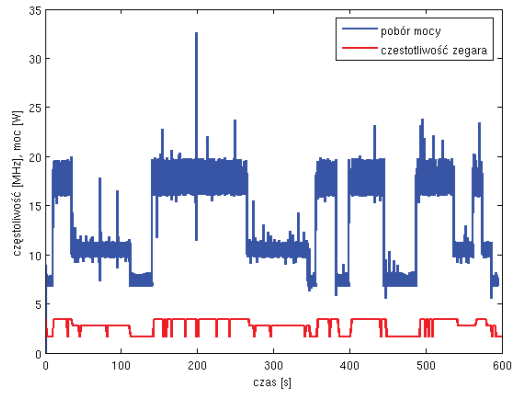
Rys. 2. Zmiany częstotliwości zegara procesora przy obciążeniu o charakterze obliczeniowym.

zostały wykonane co 100 ms, co jest wartością znacznie niższą niż dynamika zmian obciążenia (w tym wypadku przerwa między obliczeniami w programie `stress` wynosiła 160us).

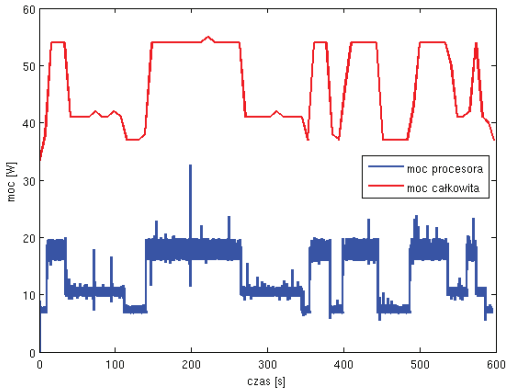
Z faktu takiego sposobu sterowania zegarem wynika również skokowa postać wykresu poboru mocy przedstawionego na Rys. 3. Pomiar ten był wykonywany co 10 ms, co pozwala zauważyć iż dynamika tego procesu jest mniejsza niż zmiany obciążenia. Należy przypuszczać, że wynika to z okresu repetycji sterownika zegara procesora (100 ms) czego potwierdzeniem jest porównanie obu wykresów widocznych na Rys. 3.

Porównanie przebiegu całkowitego poboru mocy i mocy pobieranej przez procesor (Rys. 4) wskazuje na ich dużą zgodność – potwierdza to hipoteze o istnieniu korelacji między tymi wielkościami. Mimo iż moc całkowita jest mierzona co około 8s jej przebieg odwzorowuje co do obwiedni przebieg zapisu poboru mocy przez procesor.

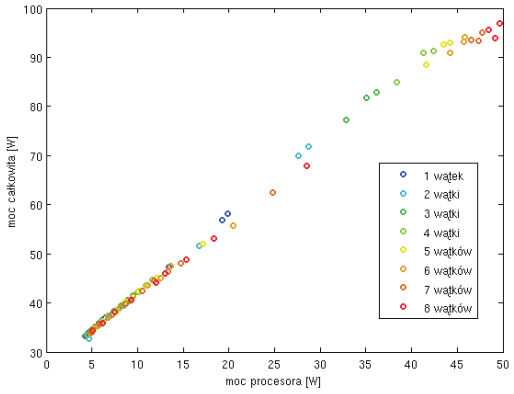
Zmienny charakter poboru mocy wymaga, aby w dalszych analizach posługiwać się uśrednionymi wartościami mocy. Wykres zależności uśrednionego całkowitego poboru mocy od mocy pobieranej przez procesor przedstawia Rys. 5. Bez szczegółowych analiz, tylko na podstawie wykresu, można stwierdzić, że zastosowanie modelu liniowego jest w tym wypadku dopuszczalne. Wiarygodność wyników podnosi fakt użycia stosunkowo dużej liczba punktów pomiarowych – przeprowadzono 96 eksperymentów. Pomiar mocy w każdym z nich był uśredniany z znacznej liczby próbek – ponad 70 dla pomiaru mocy całkowitej i około 60000 dla pomiaru mocy pobieranej przez procesor.



Rys. 3. Porównanie zmian poboru mocy i częstotliwości zegara procesora przy obciążeniu o charakterze obliczeniowym.



Rys. 4. Porównanie zmian całkowitego poboru mocy i mocy procesora odczytanego z rejestrów `msr` przy obciążeniu o charakterze obliczeniowym.



Rys. 5. Zależność całkowitego poboru mocy od mocy procesora odczytanego z rejestrów `msr` przy obciążeniu o charakterze obliczeniowym.

Eksperymenty pomiarowe – scenariusz 2 – ruter programowy

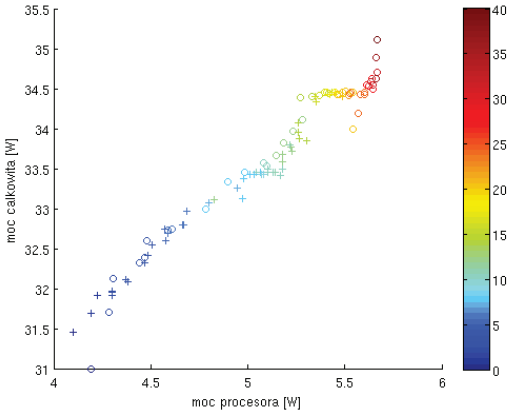
Eksperymenty można podzielić na trzy serie różniące się trybem pracy karty sieciowej: w serii pierwszej maksymalna prędkość transmisji wynosiła 10 Mb/s, w drugiej 100 MB/s, zaś w trzeciej karta pracowała z pełną prędkością czyli 1Gb/s. W każdym przypadku ruch generowano pomiędzy czterema maszynami podłączonymi do portów tej samej karty – w ten sposób zawsze obciążone były wszystkie porty. Natężenie ruchu zmieniano zgodnie z Tab. 2, przy czym dla trybów 10 Mb/s i 100 Mb/s pomijano natężenia większe od nominalnego.

Dla obu par źródło-przeznaczenie rozważono wszystkie kombinacje natężeń ruchu z Tab. 2 – dla trybu 10 Mb/s było to 49 kombinacji, dla trybu 100 Mb/s 144 kombinacje, zaś dla trybu 1 Gb/s 256 kombinacji. Dodatkowo, każdy pomiar powtarzano dla ruchu generowanego jednokierunkowo oraz dwukierunkowo. Ustalenie maksymalnego natężenia ruchu na 800 Mb/s wynika z ograniczeń karty sieciowej – w rozważanej konfiguracji i przy transmisji ramek o pełnej (1500B)

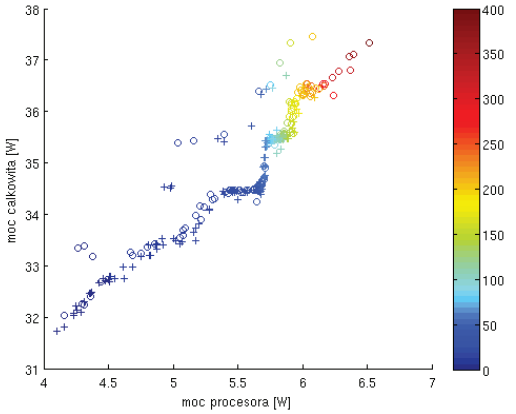
Tablica 2. Natężenia ruchu sieciowego generowanego za pomocą programu `iperf` w eksperymencie 2.

lp.	1	2	3	4	5	6	7	8
natężenie ruchu [Mb/s]	0	1	2	4	6	8	10	20

lp.	9	10	11	12	13	14	15	16
natężenie ruchu [Mb/s]	40	60	80	100	200	400	600	800



Rys. 6. Wyniki pomiarów zależności mocy całkowitej od mocy pobieranej przez procesor dla interfejsu w trybie 10 Mb/s. Skala barw odpowiada sumarycznemu natężeniu ruchu na wszystkich relacjach.



Rys. 7. Wyniki pomiarów zależności mocy całkowitej od mocy pobieranej przez procesor dla interfejsu w trybie 100 Mb/s. Skala barw odpowiada sumarycznemu natężeniu ruchu na wszystkich relacjach.

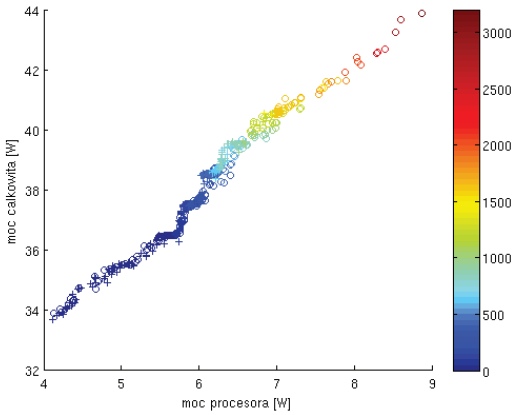
długości osiągnięto maksymalnie 812 Mb/s.

Pomiary dla interfejsu pracującego z prędkością 10 Mb/s przedstawia Rys. 6, przy czym znaczniki „+" odpowiadają transmisji jednokierunkowej, zaś „o" dwukierunkowej. Maksymalny pobór mocy jest znacznie niższy niż dla obciążenia o charakterze obliczeniowym z pierwszego scenariusza. Świadczy to o tym, iż przekazywanie ruchu między portami jest dla maszyny klasy PC zadaniem stosunkowo prostym, nie obciążającym istotnie procesora, który pracuje przez cały czas z minimalną częstotliwością taktowania. Szczególną, nie obserwowaną wcześniej, cechą wykresu jest jego falisty przebieg widoczny w środkowej części. Przypuszczalnie wynika on z ograniczonej do 1W rozdzielczości licznika mierzącego całkowity pobór mocy. Efekt taki nie był widoczny w poprzednim eksperymencie (patrz Rys. 5), należy jednak pamiętać, że wykonano tam mniej pomiarów, a zakres zmian mocy był większy, przez co wahania mogły stać się niewidoczne.

Wyniki pomiarów prowadzonych przy interfejsach pracujących w trybie 100 Mb/s przedstawia Rys. 7. Przebieg charakterystyki jest w zasadzie analogiczny jak w poprzednim przypadku, oczywiście maksymalny pobór mocy jest większy, co wynika z większego natężenia przekazywanego ruchu. Można zauważyć pewną liczbę punktów odznaczających się większym niż reszta poziomem mocy całkowitej. Przypuszczalnie są to pomiary, w trakcie których doszło do uruchomienia procesów systemowych nie związanych z pracą rutera programowego, w szczególności procesów wymagających dostępu do dysku (np. zapis dziennika systemowego). W efekcie całkowity pobór mocy wzrósł znacząco, zaś pobór mocy procesora pozostał na zbliżonym poziomie.

Rys. 8 przedstawia wyniki pomiarów przeprowadzonych przy interfejsach pracujących z maksymalną prędkością – 1 Gb/s. Zauważalną różnicą jest zwiększenie poboru mocy dla analogicznych wartości przepływów. Jest ono widoczne nawet dla pierwszego punktu pomiarowego odpowiadającego brakowi transmisji i wynosi 2W. Zmiana ta może wynikać z uruchomienia innych układów nadajników linii, może być również związana z większym niż w trybach o mniejszej przepustowości zaangażowaniem układów karty w przetwarzanie ramek (*off-loading*). Za tym ostatnim przypuszczeniem może przemawiać fakt nieznacznego obciążenia procesora, który przez cały eksperyment pracował z minimalnym zegarem – nawet podczas transmisji czterech strumieni po 800 Mb/s.

Zestawienie pomiarów w wszystkich trzech trybach – Rys. 9 pozwala wyraźniej zauważyć wspomnianą różnicę



Rys. 8. Wyniki pomiarów zależności mocy całkowitej od mocy pobieranej przez procesor dla interfejsu w trybie 1 Gb/s. Skala barw odpowiada sumarycznemu natężeniu ruchu na wszystkich relacjach.

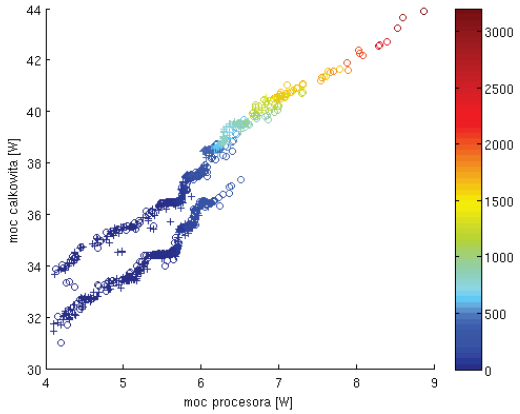
w poborze mocy w trybie 1 Gb/s. Widoczne jest też pokrywanie się punktów pomiarowych dla mniejszych prędkości interfejsów i w przybliżeniu równoległy do nich przebieg charakterystyki dla 1 GB/s.

Na uwagę zasługuje również fakt, że we wszystkich eksperymentach punkty pomiarowe odpowiadające transmisji jednokierunkowej (oznaczone '+') nie układają się w inny sposób niż te zarejestrowane przy przesyłaniu ruchu w dwu kierunkach (oznaczone 'o'). Stanowi to dodatkowe potwierdzenie tezy (patrz. [5, 15]), że zasadniczy wpływ na pobór mocy przez ruter programowy ma sumaryczne natężenie przesyłanego ruchu, nie zaś jego rozłożenie między portami.

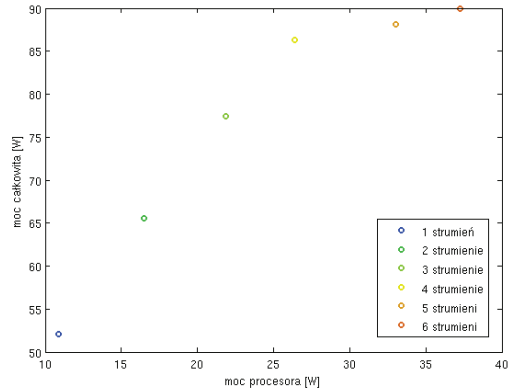
Eksperymenty pomiarowe – scenariusz 3 – serwer przekodowujący wideo

Przekodowywanie wideo jest zadaniem intensywnie obciążającym procesor, przez co możliwe jest przetwarzanie maksymalnie 6 strumieni. Generowane przy tym obciążenie portów ruchem nie jest zbyt duże i wynosi około 20 Mb/s na strumień, czyli przy maksymalnym obciążeniu stanowi około 12% przepustowości portu. Wykres zmierzonej zależności przedstawia Rys. 10.

Przebieg charakterystyki poboru mocy różni się znacząco od przedstawionych wcześniej, połączenie punktów pomiarowych daje linie wklęsłą, o zmniejszającym się nachyleniu. Aby właściwie zinterpretować tę prawidłowość należy zauważyć, że, przekodowywanie strumienia wideo, co prawda obciąża głównie procesor, ale różni się od typowych zadań



Rys. 9. Zestawienie wyników wszystkich pomiarów zależności mocy całkowitej od mocy pobieranej przez procesor. Skala barw odpowiada sumarycznemu natężeniu ruchu na wszystkich relacjach.



Rys. 10. Wyniki pomiarów zależności mocy całkowitej od mocy pobieranej przez procesor przy przekodowywaniu strumieni wideo.

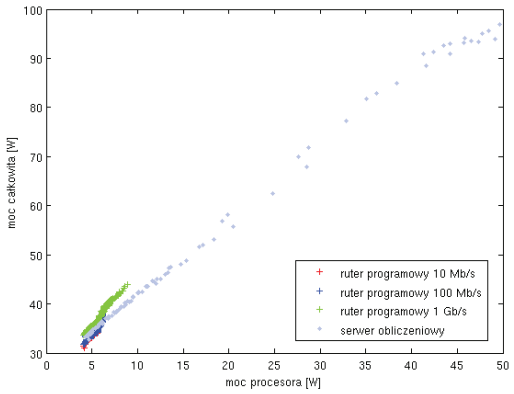
obliczeniowych, gdyż wiąże się z dość ostrymi wymaganiami czasowymi. Zagięcie charakterystyki jest widoczne głównie dla 5 i 6 strumieni, czyli obciążenia, przy którym pojawiają się opóźnienia w przekodowywaniu ramek. Są one na tyle niewielkie, ze nie powodują zerwania transmisji, jest to jednak wynikiem zastosowania dość dużych buforów po stronie odbiorczej. Dla większej liczby strumieni niezakłócona transmisja nie jest już możliwa. Należy zauważyć zbieżność z liczbą rdzeni procesora (4) – sugeruje to, iż źródłem opóźnień mogą być problemy z przełączaniem wątków i dostępem do pamięci.

Dodatkowo, transmisja danych z pomocą tuneli zestawionych programami `netcat` obciąża procesor czynnościami wykonywanymi w scenariuszu rutera programowego przez kartę sieciową (*offloading*). Znajduje to odbicie w nachyleniu początkowej części charakterystyki, które jest wyższe niż w scenariuszach 1 i 2.

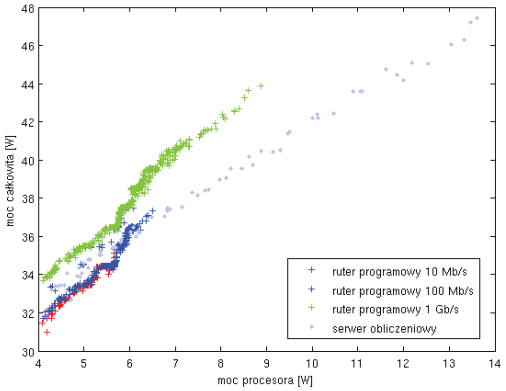
Taki układ punktów pomiarowych wskazuje, że dla w miarę dokładnej aproksymacji przebiegu charakterystyki w pełnym zakresie, potrzebne będzie zastosowanie funkcji nielinowej. Należy przypuszczać, że zastosowanie funkcji liniowej może dać dobre wyniki dla początkowej części charakterystyki – w zakresie, gdzie procesor nie jest całkowicie wykorzystany.

Eksperymenty pomiarowe – porównanie scenariuszy

Zestawienie charakterystyk zmierzonych dla scenariuszy serwera obliczeniowego i rutera programowego przedstawiają Rys. 11 i 12. Ze względu na znacznie mniejszą moc pobieraną w scenariuszu rutera należy skupić się na początkowej części wykresu. Można zauważyć, że przebieg charakterystyki poboru mocy przez serwer obliczeniowy układa się pomiędzy charakterystykami dla rutera programowego, przy czym jej nachylenie jest nieznacznie mniejsze. Można to interpretować jako dowód większego obciążenia procesora w scenariuszu serwera – mniejsze przyrosty mocy całkowitej świadczą o mniejszym wykorzystaniu przez serwer innych niż procesor komponentów maszyny. Jest to więc kolejna przesłanka wskazująca, że karta sieciowa jest w przypadku rutera krytycznym elementem, od którego zależy jego wydajność.



Rys. 11. Zestawienie wyników pomiarów zależności mocy całkowitej od mocy pobieranej przez procesor dla scenariuszy serwera obliczeniowego i rutera programowego.



Rys. 12. Zestawienie wyników pomiarów zależności mocy całkowitej od mocy pobieranej przez procesor dla scenariuszy serwera obliczeniowego i rutera programowego – powiększona początkowa część wykresu.

Analiza wykresu Rys. 13 wykazuje, że charakterystyka poboru mocy w scenariuszu przekodowywania wideo układa się powyżej punktów zmierzonych dla serwera obliczeniowego, przy czym jej początkowe nachylenie jest większe. Wskazuje to na zaangażowanie dodatkowych elementów systemu (karty sieciowej) przez transmisję danych. Zagięcie końcowego fragmentu charakterystyki można tłumaczyć mniejszym wykorzystaniem innych elementów niż procesor (karty sieciowej, pamięci) w sytuacji pojawienia się opóźnień w dekodowaniu ramek. Pobór mocy – tak przez procesor, jak i cały komputer jest znacząco większy niż w przypadku rutera programowego – potwierdza to, że procesor ma decydujący udział w bilansie mocy.

Próba identyfikacji modelu

Wyniki opisanych wcześniej pomiarów wykazują znaczną regularność. Sugeruje to, że powinno być możliwe stworzenie stosunkowo prostych modeli wiążących całkowite zużycie energii przez komputer z poborem mocy przez proce-

Tablica 3. Współczynniki zidentyfikowanych modeli poboru mocy przez komputer.

scenariusz	α_0	α_1	α_2
serwer obliczeniowy	27,33	1,46	0
ruter programowy 10 Mb/s	23,22	2,04	0
ruter programowy 100 Mb/s	22,76	2,19	0
ruter programowy 1 Gb/s	23,37	2,41	0
serwer wideo – pełny zakres	10,6	4,43	-0,06
serwer wideo – 1-4 strumieni	28,49	2,21	0

sor odczytanym z rejestrów `msr`. Dla scenariuszy 1 i 2 wystarczający powinien być model linowy – nie odda on oczywiście wszystkich odchyłek widocznych na niektórych charakterystykach, można jednak spodziewać się, że dokładność aproksymacji będzie dostateczna dla większości zastosowań. Scenariusz 3 wymaga nieco bardziej złożonego podejścia. Wydaje się, że warto rozważyć dwa przypadki różniące się stopniem obciążenia maszyny. Dla mniejszej liczby strumieni (w zakresie 1-4) można użyć modelu liniowego. Dla modelowania zależności w pełnym zakresie konieczny jest model nieliniowy, przy czym stosunkowo nieskomplikowany przebieg charakterystyki jest argumentem za zastosowaniem funkcji drugiego stopnia.

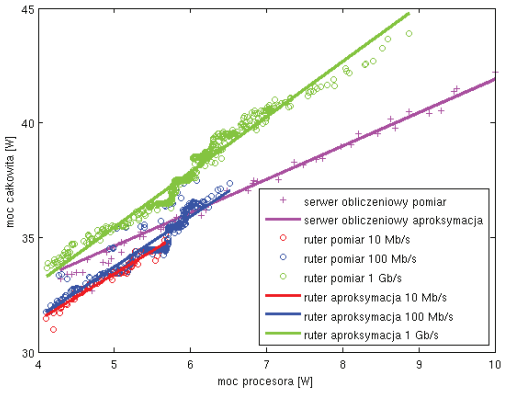
Proponowane modele są stosunkowo proste co powinno ułatwić porównanie i ocenę ogólnych właściwości omawianych przypadków. Ich zaletą jest zdolność do odfiltrowania i uśrednienia części zakłóceń oraz możliwość przeprowadzenia procedury identyfikacyjnej przy stosunkowo niewielkiej liczbie danych – ma to znaczenie szczególnie dla scenariusza 3.

Model drugiego stopnia stosowany w scenariuszu 3 można zapisać następująco:

$$p(w) = \alpha_0 + \alpha_1 w + \alpha_2 w^2, \tag{1}$$

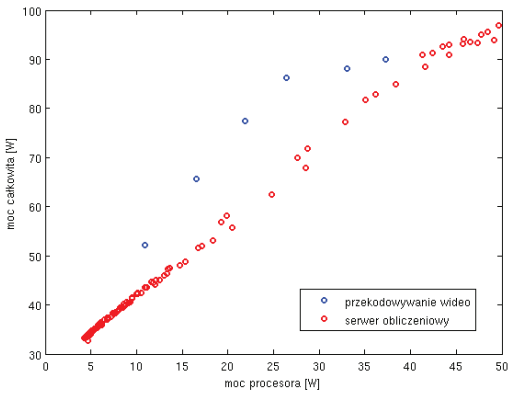
gdzie: w jest mocą odczytaną z rejestrów `msr` procesora, $p(w)$ jest całkowitym poborem mocy przez komputer a α_0, α_1 i α_2 są zidentyfikowanymi współczynnikami modelu. Należy przy tym zauważyć, że model liniowy, użyty w przypadku pozostałych scenariuszy, jak również dla zawężonego zakresu zmian mocy pobieranej przez procesor w scenariuszu 3, można traktować jako szczególny przypadek modelu (1), w którym współczynnik α_2 jest równy zero. Wartości zidentyfikowany współczynników przedstawia tabela 3.

Na Rys. 14 przedstawiono porównanie modeli dopasowanych dla trzech scenariuszy pracy rutera programowego (tryby 10 Mb/s, 100 Mb/s i 1 Gb/s) oraz serwera obliczeniowego. Wykres ograniczono do początkowego zakresu ze względu na mniejszy pobór mocy w scenariuszu rutera niż w przypadku serwera obliczeniowego. Wyraźnie widoczny jest wzrost poboru mocy przy przejściu z trybów 10 Mb/s i 100 Mb/s do trybu pełnej prędkości (1 Gb/s). Prosta dopasowana dla przypadku serwera obliczeniowego rozpoczyna się na podobnym poziomie jak dla rutera programowego pracującego w trybie 1 Gb/s co można tłumaczyć faktem, iż interfejsy sieciowe były w tym przypadku skonfigurowane w tym właśnie trybie (choć nie obciążone). Warto natomiast zauważyć, że nachylenie linii jest w przypadku serwera obliczeniowego mniejsze, co stanowi istotną różnicę, wskazującą na celowość konstrukcji osobnych modeli.

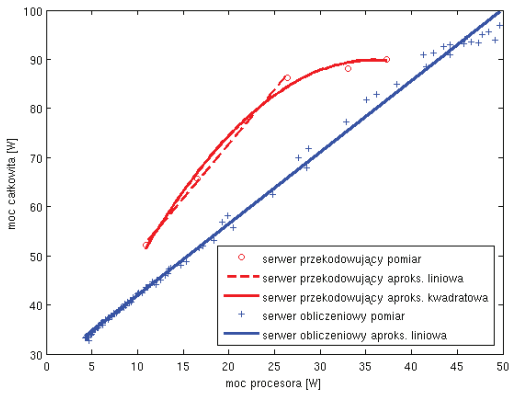


Rys. 14. Porównanie dopasowania modeli do danych pomiarowych dla przypadków rutera programowego i serwera obliczeniowego.

Nachylenie linii dopasowanych do danych zebranych dla scenariusza przekodowywania wideo jest znacząco większe niż przy obciążeniu czysto obliczeniowym (patrz Rys 15), ich przebieg w początkowym zakresie jest zbliżony do przypadku rutera programowego z interfejsami działającymi w trybie 1 Gb/s. Istotną różnicę stanowi zagięcie charakterystyki dla 5-6 strumieni – może ono być oddane wyłącznie przez model nieliniowy i jak zostało wspomniane wcześniej wiąże się najprawdopodobniej z osiągnięciem przez system kresu wydajności. W przypadku rutera programowego wykonywane operacje są mniej wymagające obliczeniowo, co więcej w pewnym stopniu oddelegowane do układów umieszczonych na karcie sieciowej. Znajduje to odbicie w mniejszym nachyleniu prostej dopasowanej w przypadku 1-4 strumieni (por. współczynniki dla rutera programowego 1 Gb/s i serwera wideo dla 1-4 strumieni w Tab. 3).



Rys. 13. Zestawienie wyników pomiarów zależności mocy całkowitej od mocy pobieranej przez procesor dla scenariuszy serwera obliczeniowego i serwera przekodowującego wideo.



Rys. 15. Porównanie dopasowania modeli do danych pomiarowych dla przypadków serwera obliczeniowego i serwera przekodowującego wideo.

Weryfikacja modelu

Dla sprawdzenia poprawności identyfikacji przeprowadzono dodatkową serię eksperymentów pomiarowych. Umożliwiło to ocenę dokładności w nowych punktach, a przez to weryfikację poprawności założenia o kształcie modelu. Jako miarę dopasowania modelu wykorzystano średnią wartość bezwzględną błędu wyznaczoną zgodnie z wzorem poniżej:

$$e_0 = \frac{1}{N}\sum_{i=1}^{N} \mid p(w_i) - p_i \mid, \tag{2}$$

gdzie w_i jest i-tą próbką mocy odczytana z rejestrów `msr` w czasie eksperymentu weryfikacyjnego, p_i jest odpowiadającą jej wartością mocy całkowitej, $p(w_i)$ jest wartością mocy całkowitej odpowiadającej w_i wyznaczoną z pomocą zidentyfikowanego wcześniej modelu, zaś N jest liczbą próbek.

Podobnie jak poprzednio przeprowadzono trzy serie eksperymentów pomiarowych polegających na: wykonywaniu obliczeń o zadanej intensywności (program `stress`), przekazywaniu ruchu sieciowego generowanego programem `iperf`, oraz przekodowywaniu wideo. Parametry pierwszych dwóch grup eksperymentów pomiarowych przedstawiają tabele 4 i 5, eksperymenty polegające na przekodowywaniu wideo prowadzono dla od 1 do 6 strumieni o przepływności zmniejszonej do około 80% w stosunku do wartości wykorzystywanych przy identyfikacji.

Tablica 4. Nastawy zmodyfikowanego programu `stress` w eksperymencie weryfikacyjnym 1.

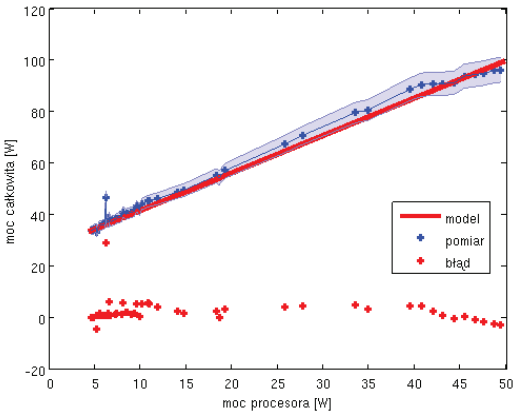
czas[μs]	32000	10000	4500	1200	90	5
obciążenie [%]	8	22	42	71	90	96

Tablica 5. Natężenia ruchu sieciowego generowanego za pomocą programu `iperf`w eksperymencie weryfikacyjnym 2.

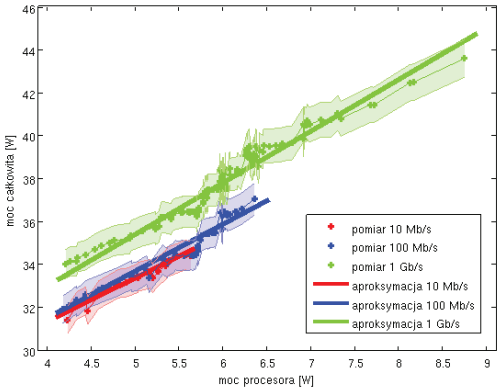
lp.	1	2	3	4	5	6	7	8
natężenie ruchu [Mb/s]	0,5	3	7	15	50	90	300	700

Graficzne przedstawienie wyników eksperymentu weryfikacyjnego dla przypadku serwera obliczeniowego zawiera Rys. 16. Jasnym kolorem oznaczono margines o szerokości $\pm 5\%$ w stosunku do zmierzonych wartości. Jak można zauważyć, poza pojedynczym punktem w początkowej części wykresu, linia odpowiadająca modelowi mieści się w tak określonym zakresie dokładności, przy czym średnia wartość bezwzględna błędu (2) nie jest większa niż 3,1%. Można to interpretować jako potwierdzenie słuszności założenia o liniowości modelu, przy czym przyjęta dokładność (5%) powinna być wystarczająca dla większości zastosowań związanych z sterowaniem. Pojedyncza próbka wykazująca większą niż 5% odchyłkę jest najprawdopodobniej wynikiem błędu pomiarowego, czy też efektem uruchomienia się w czasie pomiaru jednego z procesów systemowych – np. indeksacji plików, który to proces intensywnie używa dysków a przez to może znacząco podwyższać całkowity pobór mocy. Falisty przebieg punktów pomiarowych w końcowym odcinku charakterystyki może jednak świadczyć o występowaniu pewnych nieregularności wynikających z nierównomiernego obciążenia elementów składowych maszyny.

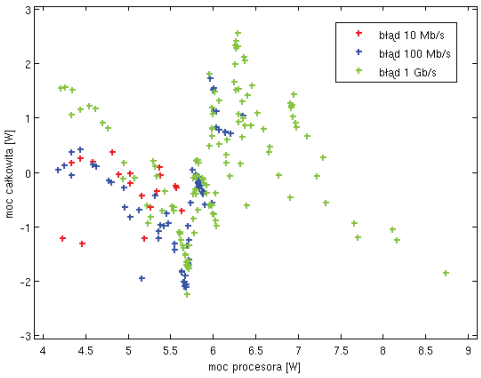
Weryfikacja modeli dla scenariusza rutera programowego dała również podobne wyniki, przy czym modele te charakteryzują się nieco większą dokładnością – oznaczone na Rys. 17 jaśniejszym kolorem obszary odpowiadają odchyłce od zmierzonych wartości o 2%. Średnia wartość bezwzględna błędu wynosi odpowiednio 0,4%, 0,8% i 0,9% dla odpowiednio trybów pracy 10 Mb/s, 100 Mb/s i 1 Gb/s. Osiągnięcie lepszej dokładności możliwe jest dzięki zebraniu większej liczby próbek, na podstawie których identyfikowano model, nie bez znaczenia jest też fakt, że zakres zmian argumentu identyfikowanej funkcji (czyli mocy pobieranej przez procesor) jest mniejszy niż dla serwera obliczeniowego. Podobnie jak w poprzednim przypadku liniowy model nie oddaje nierównomierności poboru mocy całkowitej widocznych jako zafalowania na Rys. 17. Dokładniejsza analiza błędów modelu przedstawiona na Rys. 18 pokazuje, iż dla wszystkich trzech modeli mają one zbliżony przebieg. Obserwacja ta wskazuje, że mogą one mieć wspólne źródło, a co za tym idzie powinno być możliwe stworzenie dokładniejszego modelu – o ile zaszłaby taka potrzeba.



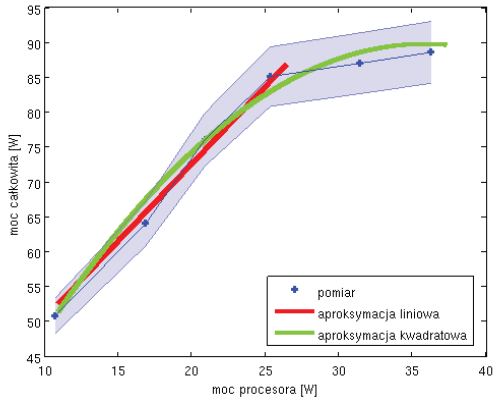
Rys. 16. Weryfikacja modelu dla scenariusza 1 – serwera obliczeniowego



Rys. 17. Weryfikacja modelu dla scenariusza 2 – rutera programowego



Rys. 18. Błędy modeli dla scenariusza 2 – rutera programowego



Rys. 19. Weryfikacja modelu dla scenariusza 3 – serwera przekodowującego wideo

Przypadek serwera przekodowującego wideo jest, jak wskazano wcześniej bardziej skomplikowany. Modelowanie zależności całkowitego poboru mocy od mocy odczytywanej z rejestrów procesora w pełnym zakresie dopuszczalnych obciążeń wymaga użycia funkcji nieliniowej. Na Rys. 19 przedstawiono wyniki weryfikacji nastrojonego wcześniej modelu, oznaczony kolorem błękitnym margines wynosi podobnie jak w przypadku serwera obliczeniowego 5%. Oba modele: kwadratowy pokrywający pełny zakres zmian i liniowy działający tylko dla mniejszych obciążeń mieszczą się w założonej dokładności. Układ punktów pomiarowych wskazuje, że zakres, który przyjęto dla modelu liniowego jest właściwy. Oznacza to, że punkt zagięcia charakterystyki całkowitego poboru mocy jest dobrze określony poprzez wartość mocy pobieranej przez procesor.

Podsumowanie

Zaprezentowane pomiary pozwoliły na zidentyfikowanie modeli wiążących całkowite zużycie mocy przez wykonujący określone operacje komputer z poborem mocy przez procesor odczytanym z jego rejestrów `msr`. Istotną obserwacją jest fakt, że postać modelu jest uzależniona od rodzaju wykonywanych zadań. W stosunkowo prostych przypadkach – serwera wykonującego wyłącznie obliczenia lub rutera programowego nie stosującego zbyt złożonych reguł przetwarzania ruchu, dopuszczalne jest użycie modelu liniowego co znacząco ułatwia identyfikację. W przypadku serwera przekodowującego wideo kształt charakterystyki jest bardziej złożony, udaje się jednak aproksymować go funkcją kwadratową. W obu wariantach możliwe jest więc stosunkowo szybkie zidentyfikowanie parametrów modelu, nawet gdy nie jest dostępne wyspecjalizowane oprzyrządowanie – dla w miarę dokładnego określenia wartości dwóch (dla modelu liniowego) lub trzech (dla modelu kwadratowego) współczynników wystarczy wykonanie kilku pomiarów, co jest wykonalne poprzez bezpośredni odczyt z licznika energii lub użycie prostego miernika.

Głównym ograniczeniem proponowanego modelu jest wspomniana wcześniej zależność jego postaci od rodzaju obciążenia. W przypadku konstrukcji energooszczędnych algorytmów sterujących urządzeniami wynika stąd potrzeba ich dostosowania do konkretnych warunków. Cechę tę można odbierać jako wadę, decyduje ona jednak o możliwości zwiększenia efektywności enegetycznej ponad poziom wyznaczany przez sterowniki uniwersalne, jak np. dostępny w Linuksie `ondemand` – por. [6, 16].

Prezentowane scenariusze, choć typowe, nie wyczerpują wszystkich możliwych zastosowań. Dla szerszego wykorzystania konieczne byłoby przeprowadzenie pomiarów dla co najmniej kilku dodatkowych scenariuszy, w szczególności takich, w których intensywnie używana jest pamięć dyskowa a wykonywane obliczenia w pełni korzystają z możliwości procesora. Pewnym rozwiązaniem może być też wzbogacenie modelu o dodatkowe zmienne wejściowe dostępne w systemie operacyjnym – np. statystyki ruchu na interfejsach sieciowych.

***Autorzy**: dr Piotr Arabas, dr Michał Karpowicz, NASK (Naukowa i Akademicka Sieć Komputerowa Instytut Badawczy) ul. Wąwozowa 18, 02-796 Warszawa; Instytut Automatyki i Informatyki Stosowanej Politechniki Warszawskiej, Wydział Elektroniki i Technik Informacyjnych, ul. Nowowiejska 15/19, 00–665 Warszawa; email: parabas@ia.pw.edu.pl, M.Karpowicz@elka.pw.edu.pl*

LITERATURA

[1] Nedevschi S., Popa I., Iannacone G., Wetherall D., Ratnasamy S.: Reducing network energy consumption via sleeping and rate adaptation, Proc. 5th USENIX Symposium on Networked Systems Design and Implementation, str. 323–336, 2008.
[2] Chabarek J., Sommers J., Barford P., Estan C., Tsiang D., Wright S.: Power awerness in network design and routing, Proc. 27th Conference on Computer Communications (INFOCOM 2008), str. 457–465, 2008.
[3] Venkatachalam V., Franz M.: Power reduction techniques for microprocessor systems, ACM Computing Surveys, vol. 37, no. 3, str. 195-237, 2005.
[4] Tucker R.S., Parthiban, R., Baliga, J., Hinton, K.: Evolution of WDM Optical IP Networks: A Cost and Energy Perspective, Journal of Lightwave Technology, vol. 27 , issue: 3, str. 243-252, 2009.
[5] Bolla R., Bruschi R., Carrega A., Davoli F.: Theoretical and technological limitations of power scaling in network devices, Proc. 2010 Australasian Telecommunication Networks and Applications Conference, str. 37-42, 2010.
[6] Karpowicz M.: Energy-efficient CPU frequency control for the Linux system, Concurrency and Computation: Practice and Experience, doi:10.1002/cpe.3476, 2015.
[7] Chiaraviglio L., Mellia M., Neri F.: Energy-aware backbone networks: a case study, Proc. IEEE International Conference Communications Workshops 2009, str. 1-5, 2009.
[8] Vasić N., Kostić D.: Energy-aware traffic engineering, Proc. 1st International Conference on Energy-Efficient Computing and Networking E-ENERGY, 2010.
[9] Niewiadomska-Szynkiewicz E., Sikora A., Arabas P., Kamola M., Mincer M., Kołodziej J.: Dynamic power management in energy-aware computer networks and data intensive systems, Future Generation Computer Systems, vol. 37, str. 284-296, 2014.
[10] Qureshi A., Weber R., Balakrishnan H., Guttag J., Maggs B.: Cutting the electric bill for internet-scale systems, SIGCOMM Comput. Commun., Rev. 39,4, str. 123-134, 2009.
[11] Kozakiewicz A., Malinowski K.: Network traffic routing using effective bandwidth theory. European Transactions on Telecommunications, 20(7), str. 660-667, 2009.
[12] Karpowicz M., Arabas P.: Energy-aware multi-level control system for a network of Linux software routers: design and implementation IEEE Systems Journal, *w druku*, 2015.
[13] Intel: Intel® 64 and IA-32 Architectures Software Developer's Manual Combined Volumes: 1, 2A, 2B, 2C, 3A, 3B and 3C, http://www.intel.com/content/dam/www/public/us/en/documents/manuals/64-ia-32-architectures-software-developer-manual-325462.pdf, 2015.
[14] Pallipadi V., Starikovskiy A.: The Ondemand Governor: Past, Present, and Future, Proc. Linux Symposium, vol. 2, str. 215-230, 2006.
[15] Bolla R., Bruschi R., Carrega A., Davoli F., Suino D., Vassilakis C., Zafeiropoulos A.: Cutting the Energy Bills of Internet Service Providers and Telecoms Through Power Management, Comput. Netw., vol. 56,no. 10, str. 2320-2342, 2012.
[16] Karpowicz M., Arabas P,: Preliminary results on the Linux libpcap model identification. Proceedings of the 20th IEEE International Conference on Methods and Models in Automation and Robotics, 2015.