Politechnika Warszawska

Wydział Elektroniki i Technik Informacyjnych
Instytut Automatyki i Informatyki Stosowanej

Działania wstępne dla zadania walidacji i oceny efektywności działania energooszczędnego algorytmu sterowania szybkością pracy serwera w laboratorium

Michał Getka

Raport 17-02

Badanie są wspierane przez Narodowe Centrum Nauki.
Projekt badawczy nr 2015/17/B/ST6/01885.

7 marca 2017

Warszawa 2017

1 Cel dokumentu

Celem dokumentu jest przedstawienie kontekstu i postępu dotychczasowych prac w ramach realizacji działań wstępnych dla zadania szóstego „walidacja i ocena efektywności działania energooszczędnego algorytmu sterowania szybkością pracy serwera w laboratorium".

W pierwszym rozdziale opisana została motywacja prac realizowanych w ramach zadania. W drugim pokrótce opisane jądro systemu Linux oraz istotne dla implementacji cechy modułu CPUfreq. Planowane podejście do definiowania norm wydajności pracy regulatorów opisane w rozdziale trzecim. Czwarty rozdział stanowi podsumowanie dotychczasowego postępu prac wstępnych.

1.1 Motywacja pracy

Empiryczne prawo Moora, stanowi że ekonomicznie optymalna liczba tranzystorów w układzie scalonym mikroprocesora podwaja się co około 24 miesiące. Wynika to z jednej strony, z rozwoju technologii i optymalizacji procesów technologicznych, z drugiej natomiast z dynamicznie wzrastającego zapotrzebowania na wysoko wydajne systemy obliczeniowe. Nieodłącznie, wraz ze wzrostem mocy obliczeniowej wzrasta pobór energii i w efekcie wydzielane ciepło.

Duże centra danych zużywają porównywalną ilość energii co małe miasto. W ich przypadku, z ekonomicznego punktu widzenia, wskazane jest aby nie tylko rozbudowywać infrastrukturę w celu zwiększania zakresu świadczonych usług, ale również inwestować w rozwiązania pozwalające na zwiększenie wydajności energetycznej systemu. Energooszczędność ma również narastające znaczenie w segmencie konsumenckim. Zasilane bateryjnie urządzenia mobilne stały się nieodłączną częścią codziennego życia. Zmniejszenie zużycia energii pozwala na dłuższy czas pracy pomiędzy cyklami ładowania, towarzyszące mu ograniczenie wydzielania ciepła pozwala natomiast na ograniczenie gabarytów układów chłodzących i w konsekwencji postępującą miniaturyzację urządzeń.

Producenci komponentów systemów komputerowych odpowiadają na zapotrzebowanie rynku proponując urządzenia zawierające mechanizmy zarządzania zużyciem energii. W szczególności układy mikroprocesorowe ostatnich generacji mogą pracować na różnych poziomach wydajności, które to nie muszą ograniczać się tylko do zmiany częstotliwości pracy zegara. W celu zwiększenia efektywności energetycznej przy zachowaniu jakości świadczonych usług, bieżąca wydajność procesora powinna być skorelowana z zapotrzebowaniem na moc obliczeniową niezbędną dla zapewnienia oczekiwanej funkcjonalności systemu w danych warunkach. Oportunistyczne regulatory skalujące częstotliwość procesora, działające w obrębie jądra systemu operacyjnego, dążą do zapewnienia takiej korelacji.

W ogólności regulatory te mają za zadanie takie sterowanie pracą procesora aby maksymalizować wydajność energetyczną prowadzonych działań w obrębie urządzenia. Może to zostać zapisane np. jako wskaźnik postaci

$$\text{wydajność energetyczna} = \frac{\text{użyteczność obliczeniowa}}{\text{wykorzystana energia}} \tag{1}$$

Problemem z którym należy się zmierzyć projektując taki regulator jest fakt że wielkości składające się na zaproponowany wskaźnik nie muszą – i najczęściej nie są – w sposób bezpośredni mierzalne. Sprawia to również że opracowanie miarodajnego narzędzia oceniającego pracę układów regulacji, które w szczególności pozwalało by na obiektywne ich porównanie nie jest trywialne.

1.2 Cel pracy

Celem rozpoczętych prac w ramach zadania „walidacja i ocena efektywności działania energooszczędnego algorytmu sterowania szybkością pracy serwera w laboratorium" jest opracowanie zestawu narzędzi pozwalających na przeprowadzenie walidacji i oceny algorytmów sterowania w warunkach roboczych. Konieczne jest zdefiniowanie możliwie obiektywnego estymatora wydajności

energetycznej pracy procesora oraz modelu obciążenia pozwalającego na realizację miarodajnych i powtarzalnych scenariuszy testowych dla zestawu konfiguracji układów regulacji. Ostatecznie narzędzia muszą zostać zaimplementowane w ramach wybranego środowiska testowego (konfiguracja sprzętowa i programowa). Środowiskiem w ramach którego zdecydowano się zaimplementować narzędzia jest system komputerowy wyposażony w procesora oparty o architekturę Intel 64 lub IA-32 pracujący pod kontrola systemu operacyjnego Linux.

Celem prac opisywanych w niniejszym dokumencie jest zapoznanie się z dotychczas prowadzonymi badaniami - w szczególności, z przyjętymi metodykami określania wydajności oraz estymatorami stosowanymi w oportunistycznych regulatorach skalujących częstotliwość procesora dostępnych dla systemu Linux. Ze wstępnej analizy zagadnienia wynika że część lub całość zestawu narzędzi implementowanych będzie w ramach jądra systemu operacyjnego Linux. W ramach zadania, autor miał wdrożyć się w zagadnienia związane z programowaniem komponentów jądra systemu, odnaleźć dostępne mechanizmy wartościowe z punktu widzenia projektu oraz rozpocząć działania implementacyjne.

2 System operacyjny Linux jako środowisko pracy zestawu narzędzi

Regulatory skalujące częstotliwość pracy procesora, mająca za zadanie maksymalizować wydajność energetyczną pracują z poziomu jądra systemu operacyjnego. Również elementy opracowywanego zestawu narzędzie służących ocenie jakości działania wspomnianych regulatorów zdecydowano się zrealizować jako moduły jądra systemu operacyjnego.

W rozdziale zawarto najistotniejsze spostrzeżenia dotyczące jądra systemu Linux, oraz sposobu implementacji nowych jego funkcjonalności. Wyszczególniono również istniejące moduły i funkcjonalności które zdecydowano się zastosować lub zaklasyfikowano jako szczególnie przydatne.

2.1 Jądro systemu Linux

Pierwsza wersja kodu źródłowego jądra systemu Linux opublikowana została w Internecie przez Linusa Torvaldsa w roku 1991. Jądro systemu jest objęte Powszechną Licencją Publiczną GNU (GNU GPL), oznacza to że każdy może bezpłatnie pobrać jego kod źródłowy i wprowadzać w nim dowolne modyfikacje. Jedynym wymogiem jest dalsze udostępnianie wersji zmodyfikowanych na tych samych zasadach. Dzięki temu, dzisiejsza postać jądra jest owocem pracy wielu programistów, komunikujących się głownie za pośrednictwem internetowych list dyskusyjnych.

W przypadku kiedy znaczna grupa osób, o różnych nawykach, pracuje nad jednym projektem, może dojść do sytuacji w której fragmenty przygotowane przez każdą z nich rządzą się zupełnie innymi prawami. Funkcjonalnie, będą one realizować jedną spójną ideę, jednak w kwestii metodyki realizacja poszczególnych funkcjonalności składowych jak i zapisu kodu źródłowego, każda będzie niejako „z innego świata". Aby takich sytuacji uniknąć społeczność zdefiniowała spójny, obowiązujący styl programowania. Składają się na niego zarówno wytyczne dotyczące formatowania kodu, sposobu nadawania nazw zmiennych, unikania pewnych konstrukcji językowych jak i tworzenia komentarzy. Sprawia to że zarówno kod który przygotujemy jest czytelny dla innych członków społeczności, jak i zasoby dotychczas zawarte w repozytoriach będą dla nas łatwiej przyswajalny.

Omówiona kwestia spójności sposobu programowania ma również związek z przyjętym w społeczności sposobem dokumentowania źródeł. Przy modyfikowaniu dotychczas istniejących funkcjonalności, bądź wdrażaniu nowych, które nie są w pewien sposób rewolucyjne, generalnie nie narzuca się tworzenia czy aktualizowania dokumentacji jądra. Przyjmuje się że kod powinien być napisany w sposób na tyle przejrzysty aby sam dla siebie stanowił dokumentację. Pewne wybrane moduły czy aspekty funkcjonowania są co prawda opisane w odpowiednich plikach dokumentacyjnych. Ogranicza się to jednak tylko do wysokopoziomowych funkcjonalności i sposobu ich obsługi, nie natomiast do szczegółowego opisu każdego aspektu implementacji. Za część dokumentacji należy również uznać zapisy dyskusji przeprowadzonych w ramach list dyskusyjnych, na drodze których rzesza autorów skonkludowała ostateczną postać realizacji danej funkcjonalności.

W subiektywnej opinii autora pracy, przyjęte w ramach społeczności standardy mają swoje wady i zalety. Bezdyskusyjnie takie podejście nie zwiększa nakładów pracy, poprzez wymóg tworzenia specjalnej dokumentacji, kładzie to też wyjątkowy nacisk na przestrzeganie przyjętego stylu kodowania, co sprzyja czytelności źródeł. Jako wadę wskazuje się fakt że w celu zrozumienia sposobu realizacji, często najprostszych funkcjonalności, niezbędne jest zapoznanie się z fragmentami kodu zawartymi w wielu funkcjach i definicjach, często na przestrzeni wielu plików źródłowych, co bywa uciążliwe. Niewątpliwie, w posługiwaniu się źródłami jądra systemu Linux w charakterze dokumentacji należy nabrać wprawy.

2.2 Moduł CPUfreq i obsługa procesorów architektury Intel 64 i IA-32

Mechanizmem odpowiedzialnym za zarządzanie mocą procesora w systemie Linux jest CPUFreq, przy czym stwierdzenie o „zarządzaniu" jest tutaj pewnym uszczegółowieniem. W rzeczywistości bowiem CPUFreq, jako taki, nie jest związany z konkretną architekturą procesora. Wspiera on zarówno mikroprocesory klasy x86, jak i inne, takie jak na przykład układy z rdzeniem ARM. W

ogólności procesory miewają wbudowane mechanizmy skalowania częstotliwości zegara w odpowiedzi na obciążenie. W takim przypadku moduł CPUFreq, nie steruje pracą procesora, jedynie definiuje górny i dolny limit częstotliwości. Limity te definiowane są w ramach polityki.

Należy w tym momencie zauważyć że spotykane są układy wielordzeniowe w których rdzenie podzielone są na więcej niż jedną domenę częstotliwościową, tzn. w obrębie jednego procesora, zespoły rdzeni logicznych pracują przy różnych częstotliwościach zegara. Identyczna sytuacja występuje w przypadku systemów wieloprocesorowych – rdzenie logiczne widziane przez system operacyjny pracują w ramach różnych domen częstotliwościowych (w szczególności są to odrębne procesory z pojedynczymi domenami). W takich przypadkach dla każdej z domen definiowana jest odrębna polityka.

W przypadku procesorów w których autonomiczne mechanizmy skalowania nie są dostępne lub zdecydowano się je wyłączyć rola CPUFreq jest znacznie większa. Dla polityki każdej z domen częstotliwościowych przypisywany jest algorytm regulatora skalującego częstotliwość. Algorytmy te nazywane są governorami. W systemie Linux dostępnych jest pięć governorów niezależnych od architektury sprzętowej. Trzy z nich mają charakter statyczny – ustawiają częstotliwość na maksymalną (performance), minimalną (powersave) lub stałą, zdefiniowaną przez użytkownika (userspace), dwa pozostałe ustalają częstotliwość w zależności od obciążenia (ondemand i conservative).

Ostatnim komponentem składowym CPUFreq są sterowniki procesora. Są to moduły, dedykowane konkretnym architekturom sprzętowym, pozwalającym na komunikację z konkretnym procesorem w ramach zdefiniowanego, niezależnego od architektury API.

W procesorach opartych o architekturę Intel 64 i IA-32, w przypadku gdy są one skonfigurowane w taki sposób aby kontrolę nad częstotliwością pracy sprawował system operacyjny, wprowadzanie nowych wartości sygnału sterującego dla domeny realizowane jest za pośrednictwem rejestrów IA32_PERF_CTL. Co istotne każdy rdzeń logiczny posiada swój rejestr IA32_PERF_CTL. Pomimo tego że fizycznie rdzenie będące w tej samej domenie muszą pracować z tą samą częstotliwością (w rzeczywistości, z tym samym p-state), zawartość rejestru każdego z nich może być różna. Związane to jest z faktem że rolą rejestru IA32_PERF_CTL jest definiowania żądania określonego poziomu wydajności z punktu widzenia operacji przeprowadzanych na danym rdzeniu. W obrębie domeny następuje przejście do stanu o najwyższej żądanej wydajności obliczeniowej.

Podsumowując, pomimo powiązań pomiędzy rdzeniami logicznymi poprzez wspólną domenę częstotliwościową, każdy z nich dysponuje niezależnym rejestrem wpływającym na poziom wydajności. Ponadto zgłaszane mogą być wyłącznie żądania oczekiwanych stanów wydajności obliczeniowej i nie ma pewności że faktycznie ta wartość zostanie zastosowana, nawet pomijając wpływ mechanizmów zabezpieczeń termicznych.

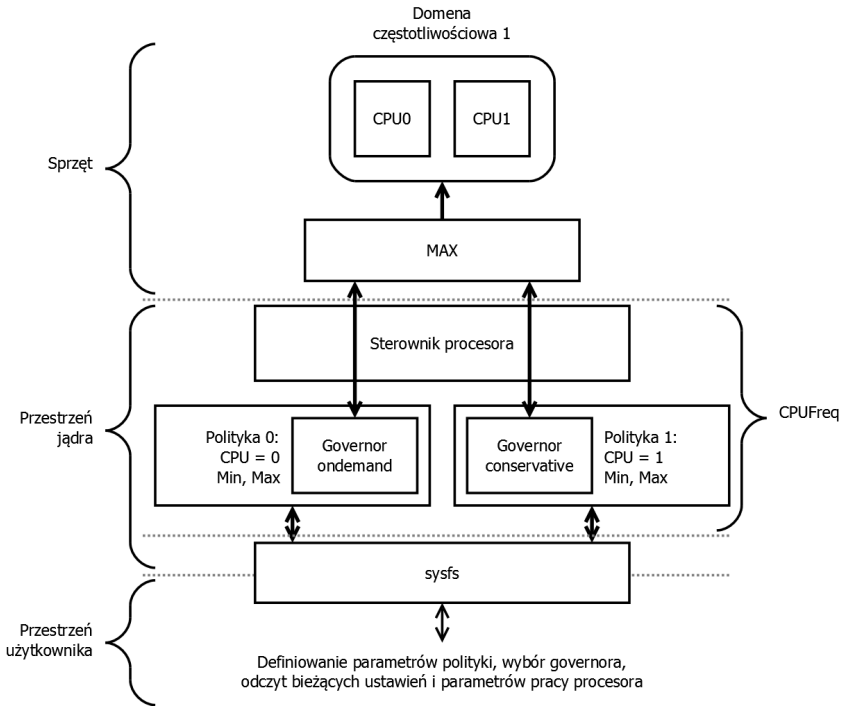
Zdefiniowany w ten sposób interfejs zarządzania wydajnością procesora nie jest w pełni spójny z pierwotną ideologią działania mechanizmów CPUFreq. Z tego powodu od wersji jądra 3.9 wprowadzone zostały zmiany w sposobie obsługi procesorów architektury Intel 64 i IA-32, które naruszają w pewnym stopniu przeznaczenie komponentów mechanizmów CPUFreq, zapewniając jednak obsługę procesora w sposób przewidziany przez producenta.

Niezależnie od zasięgu domen częstotliwości, CPUFreq postrzega każdy procesor logiczny jako niezależną domenę. Powoduje to że dla każdego z nich wyznaczana jest niezależna wartość estymatora obciążenia a następnie wyznaczana jest nowa wartość sterująca. Każdy z governorów (wiele instancji governora dla jednej domeny) niezależnie, za pośrednictwem sterownika procesora ustawia dla obsługiwanego rdzenia logicznego nową częstotliwość pracy. Pierwsza rozbieżność z ideą CPUFreq, polega na tym że wartości zgłaszane przez governory nie są faktycznie interpretowane jako częstotliwość, a identyfikatory stanów wydajności (p-state). Druga rozbieżność, objawia się tym że wywołując funkcję API sterownika odpowiedzialną za definiowanie nowych parametrów pracy domeny, faktycznie zgłaszane są żądania wydajności, spośród których ostateczna decyzja podejmowana jest już w procesorze.

Podsumowując, w przypadku obsługi procesorów opartych na architekturze Intel 64 lub IA-32, działanie CPUFreq wiąże się z następującymi odstępstwami od pierwotnej ideologii:

- Dla każdego z procesorów logicznych, niezależnie od zasięgu domeny, tworzona jest niezależna polityka obsługiwana przez niezależną instancję governora.
- Wypracowywane przez regulator wartości sterujące są interpretowane jako identyfikatory p-state, nie natomiast jako wartości częstotliwości.
- Wyznaczona przez regulator wartość sterująca nie musi faktycznie zostać zastosowana, regulator nie dysponuje również faktyczną wartością p-state przy jakiej pracuje obsługiwany przez niego procesor logiczny.

Przykładowa konfiguracja mechanizmu CPUFreq dla systemu wielordzeniowego opartego o architekturę Intel 64 lub IA-32 bez autonomicznego sterowania częstotliwością przez układ sprzętowy, przedstawiona została na rysunku 1.



Rysunek 1: Przykładowa konfiguracja mechanizmu CPUfreq dla systemu wielordzeniowego opartego o architekturę Intel 64 lub IA-32 bez autonomicznego sterowania częstotliwością przez układ sprzętowy. Symbole CPUx oznaczają procesory logiczne. Blok MAX oznacza wybór wartości p-state odpowiadającej najwyższemu poziomowi wydajności.
Źródło: opracowanie własne.

Część mechanizmów opracowywanego zestawu narzędzi zdecydowano się zaimplementować w ramach przestrzeni jądra systemu operacyjnego – konkretniej, jako modyfikacje funkcjonalności governora ondemand. Z tego też powodu świadomość roli i sposobu działania moduł CPUfreq jak i szczegóły realizacji jego funkcji w przypadku zakładanej konfiguracji sprzętowej została zaklasyfikowana jako istotna dla dalszych działań.

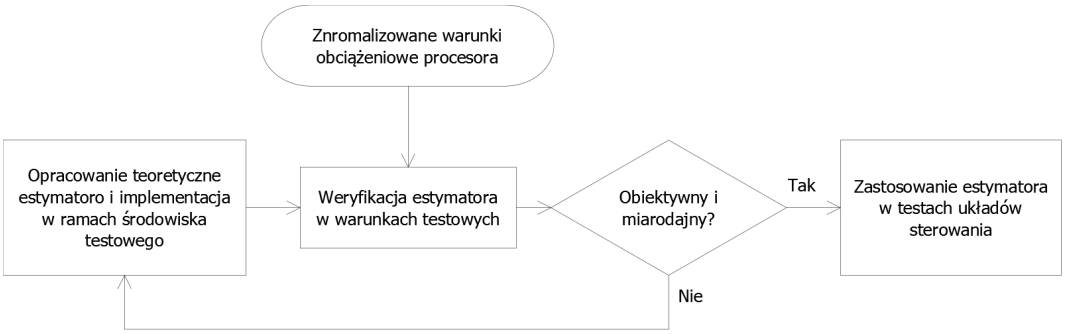
3 Estymator wydajności – metodyka badań i postęp prac

Na potrzeby przeprowadzenia badań przewidywanych w ramach zadania szóstego niezbędne jest opracowanie metodyki obiektywnej oceny efektywności działania algorytmów sterowania szybkością pracy sterowanego urządzenia. Nieodłącznym elementem takiej metodyki jest zdefiniowanie wskaźników liczbowych pozwalających na jednoznaczną ocenę i porównanie zestawu rozwiązań. Rolę tych wskaźników spełniać będą wartości estymatorów wydajności wyznaczanych na drodze zdefiniowanych scenariuszy testowych.

3.1 Metodyka badań

Przyjmuje się że walidacja i ocena efektywności algorytmów będzie dokonywana na zdefiniowanej platformie sprzętowej. Dzięki temu narzędzia temu służące jak i metody ich opracowania bazować mogą również na zasobach i informacjach dostępnych wyłącznie w ramach tej zdefiniowanej platformy, w odróżnieniu od samych implementacji sterowników które to powinny się cechować większą niezależnością od architektury sprzętowej. Potencjalnie, pozwala to na zbudowanie estymatorów o większej wiarygodności niż te których implementacja możliwa jest w ramach implementacji regulatora.

Ogólny zarys metodyki prac od momentu rozpoczęcia badań nad opracowaniem estymatora aż do kampanii testowej regulatorów przedstawiono na rysunku 2.



Rysunek 2: Ogólny zarys metodyki pracy od momentu rozpoczęcia badań nad opracowaniem estymatora aż do kampanii testowej układów sterowania.
Źródło: opracowanie własne.

Jako „znormalizowane warunki obciążeniowe procesora" rozumie się specjalną aplikację benchmarkową działającą w odpowiednio przygotowanym środowisku systemowym (minimalizującym inne obciążenia procesora) realizującą zadania obliczeniowe podzielone na etapy o znanej użyteczności obliczeniowej w rozumieniu ideowego wskaźnika jakości (1). W trakcie poddawania procesora wymuszeniom o znormalizowanej znanej postaci, badane będą wskaźniki poboru energii dla zmiennych wartości stanów wydajności urządzenia. Wynikiem przeprowadzonych badań będą charakterystyki estymatora w funkcji zmiennych warunków obciążeniowych i parametrów pracy urządzenia. Szczegóły kryterium oceny obiektywności i miarodajności estymatora stanowi temat najbliższych działań w ramach realizacji zadania.

4 Postęp prac

W ramach rozpoczętych działań wstępnych zadania zrealizowane zostały następujące aktywności:

- **Zapoznanie się z artykułami naukowymi poruszającymi tematykę energooszczędnego zarządzania energią w systemach komputerowych.**
- **Zapoznanie się z regułami implementacji modułów jądra systemu Linux.**
  W oparciu o źródła książkowe, zawartość list dyskusyjnych oraz dokumentacji utworzonej przez środowisko programistów systemu Linux, zebrano niezbędną wiedzę potrzebną do implementacji własnych funkcjonalności w ramach jądra systemu.
- **Przygotowanie środowiska badawczego i rozpoczęcie prac nad narzędziem do badania estymatorów wydajności.**
  Na komputerze osobistym wyposażonym w procesor Intel Core i5-3230M zainstalowano system Linux. Dla systemu skompilowano jądro w konfiguracji pozwalającej na swobodę modyfikowania modułów obsługi procesora.
  W ramach modułu jądra mechanizm zaimplementowany został mechanizm pozwalający gromadzić charakterystyki czasowe wybranych rejestrów i parametrów pracy procesora. Mechanizm pozwala również na wyłączenie regulatora częstotliwości pracy procesora i utrzymywanie jej na zdefiniowanej stałej wartości.
- **Zapoznanie się z wybranymi fragmentami dokumentacji procesorów Intel opartych o architekturę Intel 64 i IA-32.**
  Z dokumentu wyodrębnione zostały informacje dotyczące obecnych w procesorach mechanizmów zarządzania energią oraz interfejsów pozwalających modyfikować ich zachowania z poziomu oprogramowania, w szczególności z poziomu jądra systemu operacyjnego. Zapoznano się również ze znaczeniem i sposobem interpretacji szeregu aktualizowanych przez sprzęt rejestrów, zawierających informację zwrotną dla systemu operacyjnego, dotyczącą wydajności i parametrów pracy procesora oraz liczników przeprowadzonych operacji.