

Rafał RÓŻYCKI, Sylwia ZDEB

*Instytut Informatyki
Politechnika Poznańska
Wydział Inżynierii Zarządzania
Politechnika Poznańska*

WYBRANE ZAGADNIENIA SPOŁECZNEJ ODPOWIEDZIALNOŚCI BIZNESU W BRANŻY IT

SELECTED ISSUES OF CORPORATE SOCIAL RESPONSIBILITY IN IT INDUSTRY

In the paper the meaning of CSR (corporate social responsibility) is explained in the context of it industry. Some practical activities in the area of green computing are demonstrated on the example of novel architectures of variable speed processors. Moreover general methodology and algorithms designed to reduce energy consumption in a single computer system are demonstrated.

1. Wprowadzenie

W pracy rozważane są wybrane zagadnienia społecznej odpowiedzialności biznesu (ang. CSR – *Corporate Social Responsibility*) w branży technologii infomacyjnej (ang. IT – *Information Technology*).

We współczesnym świecie branża IT zaczyna zdobywać pozycję gałęzi gospodarki o znaczącej w skali globalnej energochłonności. Stały przyrost ilości zużywanej przez komputery energii sprawił, że rozwój technologii informatycznych nie może być już wyłącznie determinowany chęcią poprawy efektywności obliczeń za wszelką cenę. Rosnąca świadomość społecznej odpowiedzialności biznesu branży IT wymusza takie podejście do zagadnienia informatyzacji, w którym ważnym imperatywem staje się dbałość o ograniczone zasoby naturalne. Terminem obliczenia ekologiczne określa się dziś szereg działań podejmowanych właśnie w tym celu. Do działań takich zaliczyć można na przykład: sytuowanie dużych centrów obliczeniowych w pobliżu odnawialnych źródeł energii, opracowywanie zaawansowanych systemów komputerowych, które potrafią w stanie bezczynności automatycznie przechodzić w tryb oszczędzania energii, wreszcie nawet zwykłą utylizację zużytych podzespołów systemów komputerowych.

W niniejszej pracy szczególna uwaga poświęcona zostanie przykładowym działaniami, które mogą podejmować firmy z branży IT w celu obniżenia zużycia energii przez pojedynczy system komputerowy. Działania te, w największym skrócie, polegać mogą na opracowywaniu nowoczesnych architektur systemów komputerowych oraz dostosowanych do nich algorytmów szeregowania zadań, które to rozwiązania, stosowane w

skali masowej, przynieść mogą wymierne korzyści zarówno w sferze ekonomicznej jak i ekologicznej. Firmy produkujące sprzęt jak i oprogramowanie implementując te rozwiązania dać powinny dowód świadomości swej społecznej odpowiedzialności.

W pracy zaprezentowane zostaną przykładowe architektury energooszczędnych systemów komputerowych. Przedstawione będą też podejścia do problemu szeregowania zadań obliczeniowych w nowoczesnych systemach komputerowych. Problem taki modeluje się, jako deterministyczny problem szeregowania zadań na maszynach równoległych i identycznych, zasilanych ze wspólnego ograniczonego źródła mocy. Jest to sytuacja charakterystyczna dla najnowszych procesorów wielordzeniowych, w których każdy z rdzeni może być traktowany jako odrębna maszyna. Przyjmować się przy tym będzie, że ze względu na niebezpieczeństwo przegrzania układu, maksymalna ilość zużywanej przez procesor mocy jest wielkością znaną i ustaloną. Ponadto rozważane będą też odmiany problemu, w których energia stanowić będzie dodatkowe ograniczenia w problemie, albo stanowić będzie kryterium optymalizacji. Pierwszy przypadek odpowiada to sytuacji, w której zadania obliczeniowe szeregowane są na pewnym urządzeniu przenośnym zasilanym z baterii lub akumulatora o skończonej pojemności. Drugi pozwala na uzyskanie odpowiedniej efektywności pracy systemu komputerowego podłączonego do stałego źródła zasilania przy minimalnym zużyciu energii.

Praca ma głównie charakter przeglądowy, zawiera jednak szczegółowe omówienie niektórych modeli i algorytmów obliczeń ekologicznych.

Układ pracy jest następujący. Rozdział 2 jest wprowadzeniem w zagadnienia społecznej odpowiedzialności biznesu. Rozdział 3 koncentrować się będzie już na rozwiązaniach, które będą wyrazem społecznej odpowiedzialności biznesu z branży IT. Następny rozdział poświęcony został prezentacji przykładowych architektur energooszczędnych systemów komputerowych. W Rozdziale 5 przedstawiono podstawowe modele i algorytmy szeregowania zadań w nowoczesnych architekturach systemów komputerowych. Ostatni rozdział zawiera podsumowanie i uwagi końcowe.

2. Społeczna odpowiedzialność biznesu

Współczesny, oraz nowoczesny biznes to nie tylko działalność, która zmierza do wzrostu konkurencyjności firmy poprzez wprowadzanie innowacyjnych technologii i rozwiązań oraz oferowanie coraz lepszych produktów i usług. Umiejętności szerszego spojrzenia na rolę przedsiębiorstwa w otoczeniu, także mają wpływ na pozycję przedsiębiorstwa. Zagadnienie społecznej odpowiedzialności przedsiębiorstw wydaje się należeć do zjawisk trochę innego rodzaju niż terminy dotyczące bezpośrednio podmiotów ekonomii społecznej to jednak łączy w sobie dwa związane z omawianą tematyką aspekty ekonomiczny i kluczowy dla zakresu ekonomii społecznej. Znaczenia odpowiedzialności można pogrupować w trzy kategorie: zobowiązań społecznych, reakcji społecznej i wrażliwości społecznej [8]. Światowa Rada Biznesu ds. Zrównoważonego Rozwoju traktuje odpowiedzialność społeczną, jako etyczne zachowanie przedsiębiorstwa wobec społeczeństwa polegające na zaangażowaniu biznesu w postępowanie etyczne i przyczynianie się do ekonomicznego rozwoju z równoczesnym ulepszaniem życia pracowników i ich rodzin, jak również lokalnej społeczności i całego społeczeństwa.

Zbiegień-Maciąg [14] definiując społeczną odpowiedzialność opisuje ją, jako moralną odpowiedzialność firmy i zobowiązania do rozliczania się przed społeczeństwem ze swojej działalności zwłaszcza przed grupami wewnętrznymi-właścicielami i pracownikami oraz grupami zewnętrznymi-akcjonariuszami i klientami; władzami lokalnymi, grupami nacisku; ruchami ekologicznymi, konsumenckimi i dostawcami oraz kooperantami i administracją państwową.

W definicjach społecznej odpowiedzialności pojawia się także spojrzenie ekonomiczne. Społeczna odpowiedzialność traktowana jest w tym przypadku, jako element przemyślanej strategii przedsiębiorstwa, który może stać się nowym źródłem przewagi konkurencyjnej. Z punktu widzenia rachunku ekonomicznego bycie społecznie odpowiedzialnym jest postrzegane, jako bardzo opłacalne.

Kolejna grupa definicji koncentruje się na charakterze relacji biznesu i społeczeństwa. W tym ujęciu społeczna odpowiedzialność traktowana jest jako zjawisko naturalne, będące konsekwencją istniejących powiązań między społeczeństwem a przedsiębiorstwami. Przedsiębiorstwa, jako organizacje otwarte muszą brać pod uwagę interakcje z otoczeniem, które na zasadzie sprzężenia zwrotnego wpływają na ich funkcjonowanie a społeczna odpowiedzialność jest naturalnym efektem występowania tych współzależności [2].

Szeroką definicję CSR zaproponowali twórcy idei Global Compact. Global Compact to inicjatywa Sekretarza Generalnego Organizacji Narodów Zjednoczonych, Kofi Annana, przedstawiona w czasie Światowego Forum Ekonomicznego w Davos w 1999 roku, adresowana do przedsiębiorców i przedstawicieli świata biznesu na całym świecie. Stanowi ona wezwanie, aby w ramach prowadzonej działalności gospodarczej kierować się określonymi zasadami z zakresu:

- praw człowieka,
- standardów pracy,
- przeciwdziałania korupcji,
- ochrony środowiska naturalnego.

Sformułowano dziesięć następujących zasad społecznej odpowiedzialności biznesu:

1. Popieranie i przestrzeganie praw człowieka przyjętych przez społeczność międzynarodową.
2. Eliminowanie wszelkich przypadków łamania praw człowieka przez firmę.
3. Poszanowanie wolności stowarzyszania się.
4. Eliminowanie wszelkich form pracy przymusowej.
5. Zniesienie pracy dzieci.
6. Efektywne przeciwdziałanie dyskryminacji w sferze zatrudnienia.
7. Prewencyjne podejście do środowiska naturalnego.
8. Podejmowanie inicjatyw mających na celu promowanie postaw odpowiedzialności ekologicznej.
9. Stosowanie i rozpowszechnianie przyjaznych środowisku technologii.
10. Przeciwdziałanie korupcji we wszystkich formach, w tym: wymuszeniom i łapówkarstwu.

Idea Global Compact opiera się na założeniu, iż przez stosowanie zasad społecznej odpowiedzialności środowiska biznesowe mogą w znaczącym stopniu uczestniczyć w rozwiązywaniu palących problemów gospodarczych współczesnego świata, minimalizować ujemne skutki procesów globalizacji i przyczyniać się do zrównoważonego wzrostu światowej gospodarki.

Większość autorów interpretując pojęcie CRS podkreśla konieczność szerszego formułowania celów przedsiębiorstwa niż tylko a kategoriach interesu strictly ekonomicznego i włączania w działalność biznesową aspektów społecznych, ekologicznych oraz budowania dialogu z interesariuszami [7].

2.1. Zalety stosowania CSR przez firmy

Przedsiębiorstwa ze swoich działań CSR mogą czerpać korzyści przyczyniając się do oszczędności środków w przedsiębiorstwie. W [6] CSR zdefiniowano jako „podejście strategiczne i długofalowe, oparte na zasadach dialogu społecznego i na poszukiwaniu rozwiązań korzystnych dla wszystkich”.

Idea odpowiedzialnego biznesu zakłada budowanie wspólnej wartości i generowanie korzyści zarówno dla społeczeństwa, środowiska naturalnego jak i samego biznesu. CSR nie może być utożsamiany z działalnością charytatywną – ważne jest tutaj, jak firma zarabia pieniądze, a nie jak i na co je wydaje. Odpowiedzialny biznes to strategia, która w perspektywie przynosi firmie korzyści w postaci:

- wzrostu konkurencyjności – społeczna odpowiedzialność biznesu może stanowić istotny czynnik przewagi konkurencyjnej zarówno firmy, jak i całej gospodarki
- wzrostu zainteresowania inwestorów;
- zwiększenia lojalności konsumentów i interesariuszy – wzrost świadomości społecznej konsumentów powoduje, iż w swoich wyborach kierują się oni także zaufaniem do danej firmy i jej wizerunkiem;
- poprawy relacji ze społecznością i władzami lokalnymi - udział firmy w życiu społeczności lokalnej, podejmowanie długofalowych i wymiernych inwestycji społecznych, ułatwia jej pozyskanie przychylności i zaufania społeczeństwa oraz władz lokalnych;
- kształtowania pozytywnego wizerunku firmy wśród pracowników – CSR stanowi jeden z elementów pozafinansowego motywowania pracowników.; dzięki kodeksom etycznym, programom społecznym, programom wolontariatu pracowniczego, efektywnemu zarządzaniu różnorodnością, dbałości o środowisko, pracownicy chętniej identyfikują się z firmą, pracują wydajniej i efektywniej; pomaga to w pozyskaniu i utrzymaniu najlepszych pracowników, zmniejsza koszty związane z fluktuacją kadr, zwiększa efektywność firmy i tym samym wpływa na poprawę wyników finansowych;
- wzrostu wydajności – z jednej strony CSR jako czynnik motywacyjny prowadzi do wzrostu wydajności pracowników. Z drugiej strony firma działając zgodnie z zasadami zrównoważonego rozwoju, inwestując np. w technologie proekologiczne generuje znaczne oszczędności, zwiększa swoją efektywność kosztową i optymalizuje procesy produkcji;
- wzrostu efektywności zarządzania ryzykiem - dbanie o relacje z interesariuszami firmy, ułatwia przewidywanie różnego rodzaju ryzyka, przeciwdziałanie mu oraz rozwiązywanie ewentualnych konfliktów
- podnoszenia poziomu kultury organizacyjnej firmy poprzez wdrażanie strategii CSR firma podnosi swoje standardy postępowania wobec interesariuszy, co wpływa na kształtowanie kultury organizacyjnej firmy opartej na zaufaniu, odpowiedzialności i przejrzystości dla wszystkich zainteresowanych;

2.2. Narzędzia realizacji CSR

Narzędzia realizacji społecznie odpowiedzialnego biznesu ściśle związane są z kulturą organizacyjną firmy i profilem działalności. Wśród najbardziej popularnych narzędzi stosowanych do zarządzania społeczną odpowiedzialnością biznesu należy wymienić:

- Kampanie społeczne - działania ukierunkowane na zmianę postaw lub zachowań oraz wzrost świadomości wybranych grup społecznych oraz w określonych kwestiach
- Programy etyczne – ich zastosowanie wpływa m.in. na wzrost kultury w miejscu pracy, a tym samym na poprawę warunków pracy oraz ma na celu integrowanie pracowników firmy wokół wspólnych wartości
- Elastyczne formy zatrudnienia – rozwiązania ułatwiające godzenie życia zawodowego z prywatnym
- Marketing zaangażowany społecznie (ang. *cause related marketing*) – uwzględnienie w działalności firmy zarówno celów marketingowych jak i potrzeb społecznych
- Inwestowanie w pracowników - dotyczy to zarówno rozwoju zawodowego, osobistego jak i zdrowia
- Eko-znakowanie i znakowanie społeczne polegające na umieszczaniu na opakowaniach lub etykietach produktów dodatkowych informacji z zakresu ekologicznej lub społecznej odpowiedzialności
- Profilowanie produkcji/usług w kierunku rozwiązań proekologicznych.

Z założenia odpowiedzialny biznes ma być czymś więcej niż jedynie strategią komunikowania się firmy z jej otoczeniem, ma on angażować społecznie oraz ekologicznie zarówno firmę jak i jej pracowników. Tym, co łączy częściowo odpowiedzialny biznes i Public Relations jest cel długofalowy - wzrost konkurencyjności firmy poprzez budowę jej przyjaznego wizerunku i utrwalenie go w świadomości konsumentów i współpracowników.

3. Społeczna odpowiedzialność biznesu w branży IT

Terminem „branża IT” zwykło się określać wszelkie formy działalności biznesowej związane z szeroko rozumianą informatyzacją. Jako, że informatyzacja różnych form aktywności człowieka w dzisiejszych czasach staje się zjawiskiem powszechnym, tak też do branży IT zalicza się ogromną liczbę firm produkujących sprzęt komputerowy, wytwarzających do niego oprogramowanie, jak i świadczących różnorodnego rodzaju usługi związanego z jego obsługą.

Niektóre aspekty społecznej odpowiedzialności biznesu omówione w Rozdziale 2 w przypadku branży IT wpisują się w zakres działań podejmowanych w ramach tzw. obliczeń ekologicznych (ang. *Green computing*). Do działań tych zalicza się na przykład:

- wykorzystanie proekologicznych technologii wytwarzania sprzętu komputerowego;
- opracowywanie technologii umożliwiających wydłużanie okresu użytkowania sprzętu komputerowego;
- racjonalne gospodarowanie odpadami po sprzęcie komputerowym (recykling);

- upowszechnianie rozwiązań telekomunikacyjnych (telekonferencje zamiast spotkań wymagających kosztownego podróżowania);
- techniki obliczeń energooszczędnych.

Tutaj szczególną uwagę zwracać będziemy na ostatni z wymienionych aspektów obliczeń ekologicznych.

Energooszczędne techniki obliczeń obejmują na przykład takie zagadnienia jak: poprawianie sprawności energetycznej zasilaczy komputerowych, sytuowanie dużych centrów obliczeniowych w pobliżu odnawialnych źródeł energii, wykorzystywanie ciepła powstającego jako efekt uboczny obliczeń do ogrzewania budynków, opracowywanie zaawansowanych systemów komputerowych, które potrafią w stanie bezczynności automatycznie przechodzić w tryb oszczędzania energii, itd., itp.

W niniejszej pracy postulat oszczędnego, ekologicznego używania komputerów realizuje się w nieco inny sposób. Podejmuje się mianowicie próbę opracowania takiego sposobu sterowania kolejnością wykonywania zadań obliczeniowych w ramach pojedynczego systemu komputerowego by ograniczone zasoby energii wykorzystać w sposób optymalny. Zakłada się przy tym możliwość wykonywania obliczeń z różną, zależną od ilości pobieranej mocy, szybkością. Jest to założenie uzasadnione, gdyż na rynku znajduje się dziś wiele procesorów dających taką możliwość a określanych jako procesory o zmiennej szybkości przetwarzania (ang. *Variable Speed Processor* - VSP).

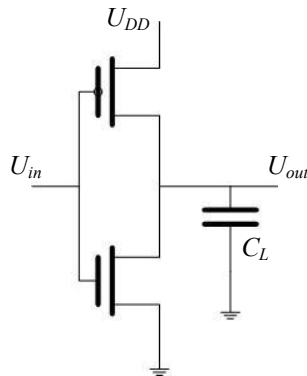
4. Architektury procesorów energooszczędnych

W rozdziale tym rozważania nasze ograniczymy do przykładowych architektur procesorów o zmiennej szybkości przetwarzania. Najpierw podane będą jednak podstawowe informacje dotyczące modeli zużycia energii w układach cyfrowych CMOS.

4.1. Zużycie energii w układach CMOS

Układy cyfrowe wykonane w technologii CMOS zdecydowanie zdominowały współczesny rynek układów mikroprocesorowych. Modele zużycia energii w układach wielkiej skali integracji opierają na podstawowym modelu bramki inwertera CMOS.

Na Rys.1 prezentującym ten model, przez U_{DD} oznaczono napięcie zasilania, U_{in} - napięcie wejściowe, U_{out} jest napięciem wyjściowym a C_L - pojemność obciążenia bramki.



Rys. 1. Model bramki inwertera CMOS

Fig. 1. Model of the CMOS inverter gate

Na całkowity pobór mocy w układzie bramki inwertera CMOS składają się dwie główne składowe:

$$P_{inv} = P_{dyn} + P_{stat} \quad (1)$$

gdzie

P_{dyn} – moc dynamiczna, pobierana przy przełączaniu układu

P_{stat} – moc statyczna pobierana w związku z występowaniem prądów upływu (bez przełączania układu).

Drugi składnik w układach CMOS jest znikomo mały i dlatego o pobieranej przez bramkę mocy decyduje składnik pierwszy. Z oczywistych względów całkowity pobór mocy dynamicznej wynika z częstotliwości przełączeń układu. Jeśli bramka przełączana jest z częstotliwością $f_{0 \ll 1}$, wtedy zużycie mocy dynamicznej wyrazić można wzorem:

$$P_{dyn} = C_L U_{DD}^2 f_{0 \ll 1} \quad (2)$$

Model poboru mocy bramki inwertera (1)-(2), wykorzystywany jest w modelowaniu poboru mocy i zużycia energii w złożonych układach kombinacyjnych CMOS. W konsekwencji, w układach takich przyjmuje się, że energia zużywana przez układ podczas wykonania jednej operacji wyraża się wzorem:

$$E_{op} = C_{eff} U_{DD}^2 \quad (3)$$

gdzie C_{eff} – jest efektywną pojemnością przełączania (zależną między innymi od obciążenia układu)

Natomiast pobór mocy na jedną operację wykonaną w układzie kombinacyjnym CMOS pracującym z częstotliwością f_{clk} jest bezpośrednią konsekwencją (2) i wynosi:

$$P_{op} = C_{eff} U_{DD}^2 f_{clk} \quad (4)$$

Z zależności (3) i (4) wynikają następujące interesujące konsekwencje.

- Na energię zużywaną przez układ kombinacyjny CMOS nie ma wpływu częstotliwość, z jaką pracuje ten układ. Energia ta jest proporcjonalna do kwadratu napięcia zasilania.
- Zmniejszanie częstotliwości pracy układu kombinacyjnego CMOS powoduje proporcjonalne zmniejszenie poboru mocy, co może być działaniem pożądanym ze względu na ryzyko przegrzania układu. Pobór mocy układu jest również proporcjonalny do kwadratu napięcia zasilania.

Jak widać na pobieraną moc i zużywaną energię przez układy CMOS można mieć wpływ przez dobór napięcia zasilania i - co dotyczy tylko mocy - częstotliwości pracy układu. Niestety, nie są to zmienne od siebie niezależne.

W podstawowym modelu procesora synchronicznego, procesor taktowany jest przez zegar systemowy. Zegar ten generuje sygnał prostokątny o założonej częstotliwości nazywanej częstotliwością taktowania. Jeden okres zegara systemowego nazywany jest cyklem maszynowym procesora. W zależności od typu procesora rozkazy procesora mogą wykonywać się w jednym lub wielu cyklach maszynowych. Zbocze narastające sygnału zegara (w większości przypadków) inicjuje między innymi wykonanie podstawowych operacji logicznych realizowanych przez przerzutniki układu kombinacyjnego stanowiącego, obok układu sekwencyjnego, integralną część każdego procesora. Minimalna długość cyklu maszynowego wyznaczona jest więc na podstawie czasu propagacji sygnału przez najdłuższą ścieżkę pomiędzy przerzutnikami w układzie kombinacyjnym (ang. *Critical path*).

Łatwo wykazać, że na czas propagacji sygnału przez układ kombinacyjny CMOS bezpośredni wpływ ma napięcie zasilania. Im wyższe napięcie zasilania, tym czas propagacji krótszy. Krytyczny czas propagacji, a więc najdłuższy, w danym układzie cyfrowym czas propagacji, musi być brany pod uwagę przy doborze maksymalnej częstotliwości pracy tego układu. Innymi słowy, częstotliwość zegara jest dobierana tak, by kolejny cykl maszynowy rozpoczynał się zawsze po stabilizacji sygnałów w układzie kombinacyjnym procesora wykonującego dowolną fazę dowolnego rozkazu procesora wykonywanego w poprzednim cyklu. Doświadczalnie pokazano [5], że zależność maksymalnej częstotliwości pracy wybranych układów CMOS jest w przybliżeniu liniowo proporcjonalna do napięcia zasilania:

$$f_{max} \sim U_{DD} \quad (5)$$

We współczesnych systemach komputerowych próbuje się wykorzystać związki (3) i (4) w celu zmniejszania poboru mocy i/lub zużycia energii. Wypracowane na ich podstawie techniki nazywa się odpowiednio:

- *Dynamic Voltage Scaling* (DVS) – technika ta pozwala na oszczędzanie energii przy zachowaniu danej szybkości przetwarzania
- *Dynamic Frequency Scaling* (DFS) – służy do zmniejszenia poboru mocy (nie zużycia energii!) przez zmniejszenie częstotliwości taktowania zegara w momentach wykonywania niekrytycznych zadań obliczeniowych

Wykorzystanie obu powyższych technik przy dodatkowym założeniu (5) prowadzi do koncepcji architektur układów mikroprocesorowych określanych dla uproszczenia mianem procesorów o zmiennej szybkości przetwarzania.

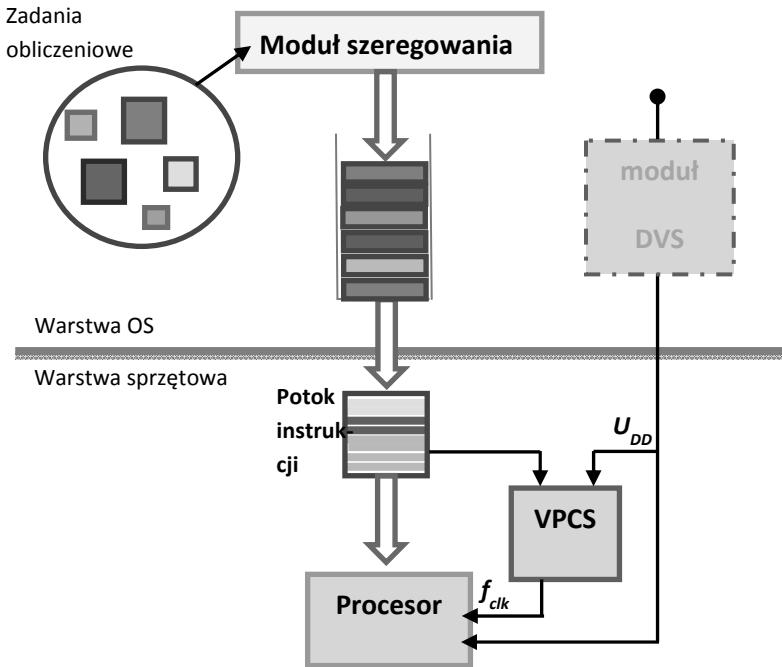
4.2. Procesory o zmiennej szybkości przetwarzania

W praktyce istnieje już wiele rozwiązań typu VSP (Intel — SpeedStep i Foxton, AMD — PowerNow). Każde z nich zakłada jednak, że przetwarzane aplikacje są nierozróżnialne i charakteryzują się takim samym typem zależności pobieranej mocy od szybkości przetwarzania.

W [4] zaproponowano architekturę VSP, w której częstotliwość zegara dobiera się dynamicznie uwzględniając rodzaj wykonywanego przez procesor zadania oraz stan systemu. Dobór odpowiednich dla zadania częstotliwości pracy procesora wykonywany jest przez odpowiedni moduł programowy na poziomie systemu operacyjnego. Na poziomie sprzętowym realizowany jest natomiast dobór odpowiedniego napięcia zasilania. Niestety, w architekturze tej nie wykorzystano w pełni możliwości doboru pary: częstotliwość–napięcie do każdego zadania indywidualnie. Napięcie zasilania układu zależało wyłącznie od wcześniej wybranej częstotliwości.

Bardziej zaawansowaną architekturę VSP zaproponowano w [1]. Wykorzystano w niej fakt, że nie dla wszystkich rozkazów i/lub faz wykonania rozkazów generowane sygnały muszą pokonywać najdłuższą ścieżkę w układzie kombinacyjnym procesora. W takim przypadku bowiem czas pomiędzy ustabilizowaniem się sygnałów a kolejnym zboczem zegara wyznaczającym początek kolejnego cyklu jest praktycznie czasem bezczynności procesora. Oznacza to, że wśród rozkazów procesora istnieją takie, które z powodzeniem mogłyby być wykonywane przy większej częstotliwości zegara. Tradycyjnie zakłada się jednak, że częstotliwość zegara jest stała i taka sama dla wszystkich rozkazów w procesorach synchronicznych. Oczywiście jest to z jednej strony marnowanie teoretycznego potencjału obliczeniowego procesorów synchronicznych, z drugiej jednak strony stwarza możliwość oszczędniejszego gospodarowania energią zużywaną przez procesor przy zachowaniu jego standardowej szybkości przetwarzania.

Koncepcja procesora synchronicznego o zmiennej szybkości przetwarzania zaprezentowana w [1] wykorzystuje syntezytor zmiennego cyklu zegara (ang. *Variable Period Clock Synthesiser* – VPCS). VPCS generuje cykl zegara, który steruje pracą większości elementów systemu, w tym również samego syntezytora. Schemat budowy procesora VSP wyposażonego w układ VPCS pokazany jest na Rys. 2.

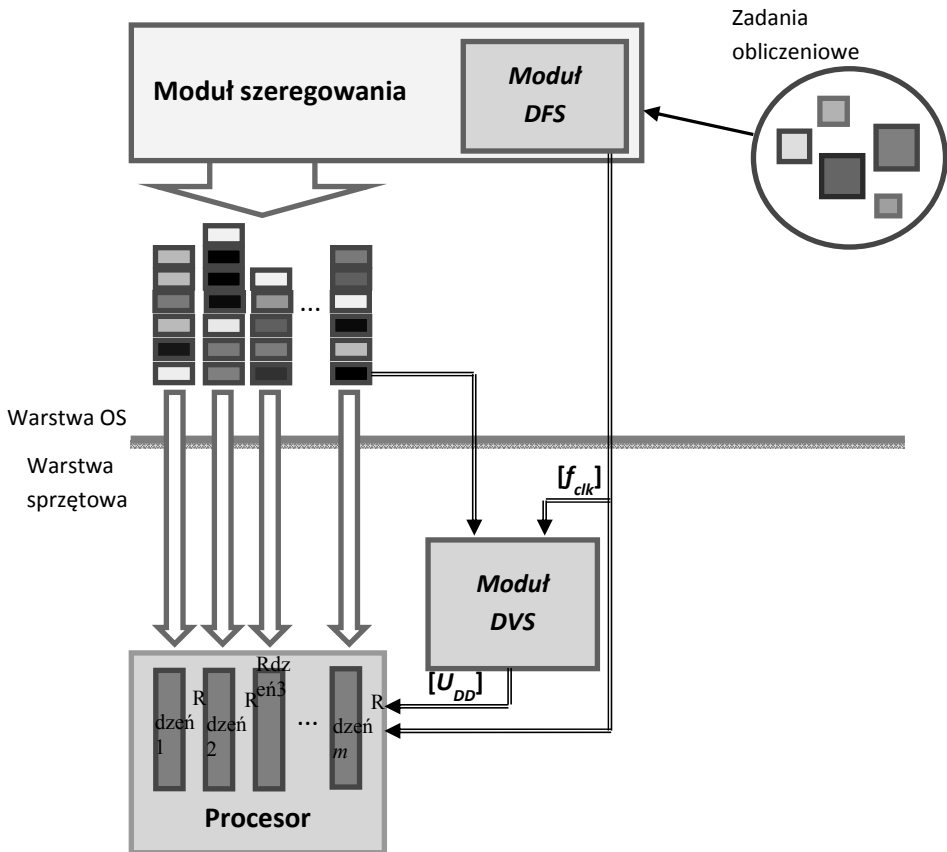


Rys.2. Przykładowa architektura VSP

Fig. 2. An example of VSP architecture

W zaproponowanej architekturze poszczególne instrukcje procesora wykonywane są potokowo. Zasadniczą cechą tego projektu jest jednak to, że częstotliwość zegara dobierana jest sprzętowo (moduł VPCS) do każdej instrukcji. Wynika to z faktu, że dla każdej instrukcji czas propagacji sygnału przez układ kombinacyjny jest inny i przy znanym napięciu zasilania częstotliwość zegara może być indywidualnie dobierana do instrukcji. Mimo, że proponowana architektura spełnia założenia *dynamic frequency scaling*, nie realizuje w pełni koncepcji procesorów VSP. Wynika to z faktu, że napięcie zasilania układu jest niezmiennie. Instrukcje procesora są traktowane przez moduł VPCS dokładnie tak samo, niezależnie od tego, do jakiego zadania obliczeniowego (programu) należą. Autorzy proponują jednak modyfikację układu polegającą na wprowadzeniu modułu programowego (moduł DVS), który na poziomie systemu operacyjnego dobieierałby napięcie zasilania do poszczególnych zadań obliczeniowych.

Rozwinięciem architektur przedstawionych w [1] i [4] może być koncepcja systemu z wielordzeniowym procesorem VSP ukazana na Rys.3.



Rys.3. Koncepcja wielordzeniowej architektury VSP

Fig. 3. A concept of a multicore VSP architecture

W architekturze tej zakłada się, że każdy z rdzeni procesora wielordzeniowego może mieć indywidualnie dobraną częstotliwość pracy oraz napięcie zasilania. Moduł szeregujący spełnia dwie funkcje: określa porządek wykonania poszczególnych zadań obliczeniowych (lub ich fragmentów) na poszczególnych rdzeniach i dobiera do uporządkowanych zadań wektor właściwych częstotliwości zegara na rdzeniach (za pomocą modułu DFS). Sprzętowy moduł DVS na podstawie określonej wcześniej częstotliwości zegara oraz rodzaju wykonywanej instrukcji dobiera właściwe napięcie zasilania dla każdego rdzenia. Dzięki takiej architekturze otrzymujemy najbardziej elastyczny model przetwarzania zadań obliczeniowych, w którym szybkość przetwarzania oraz pobierana moc przez rdzeń procesora zależą od wykonywanego zadania obliczeniowego, a nie tylko od konkretnej instrukcji wykonywanej w ramach takiego zadania. W modelu tym moduł szeregujący systemu operacyjnego jest w pełni odpowiedzialny za aspekt energie-

tyczny wykonania zbioru zadań obliczeniowych i dlatego winien być wyposażony w mechanizmy pozwalające na optymalizację wybranych kryteriów szeregowania. Mechanizmy takie opracowuje się traktując całe zagadnienie na przykład jako pewien wielomaszynowy problem szeregowania zadań z dodatkowymi ograniczeniami zasobowymi.

5. Problemy szeregowania zadań w obliczeniach ekologicznych

Problemy szeregowania na procesorach VSP były już rozważane w literaturze. W zdecydowanej większości przypadków prace te dotyczą jednak problemów szeregowania na jednym procesorze. Generalnie problemy takie mają charakter problemów dwukryterialnych, gdyż jako cel stawiać można sobie zarówno minimalizację zużycia energii, jak i minimalizację wybranego kryterium czasowego uszeregowania (np. długość uszeregowania, średni czas przepływu, maksymalne opóźnienie). Oczywiście oba tak określone kryteria stają do siebie w opozycji - poprawa jednego kryterium prowadzi do pogorszenia drugiego. Możliwe są zatem dwa podejścia: albo poszukiwać się będzie rozwiązań sprawnych (czyli takich, dla których nie istnieją inne rozwiązania o lepszych wartościach obu kryteriów), albo poszukiwać się będzie optimum jednego kryterium przy założonej wartości drugiego. Drugie podejście w literaturze (por.[3]) doczekało się nawet swoistej klasyfikacji. Problemy szeregowania na procesorach VSP dzieli się bowiem na: serwerowe (ang. *server problem*) i laptopowe (ang. *laptop problem*). W problemach serwerowych ustala się wartość kryterium czasowego uszeregowania i poszukuje się takiego uporządkowania zadań na maszynach, które minimalizuje zużycie energii. Odpowiada to na przykład praktycznej sytuacji, w której serwer obliczeniowy na stałe podłączony do źródła zasilania ma wykonać pewien zbiór zadań obliczeniowych w zadanym czasie. W tym wypadku racjonalne jest poszukiwanie uszeregowania zadań na serwerze, które spełnia wszystkie ograniczenia a dodatkowo minimalizuje zużyta energię. Z kolei, w problemach laptopowych zakłada się istnienie dla procesora źródła energii o skończonej pojemności i poszukuje się uszeregowania, które przy takich ograniczonych zasobach minimalizują wybrane kryterium uszeregowania. Zauważmy, że zarówno w przypadku znanych z literatury problemów serwerowych jak i laptopowych nie wprowadza się ograniczenia na chwilowy pobór mocy, co w praktyce prowadzić może do rozwiązań nierealnych lub wręcz wiążących się z ryzykiem przegrzania układu procesora.

5.1. Modele wykonania zadań

W odróżnieniu od klasycznej teorii szeregowania, w przypadku problemów szeregowania na procesorach VSP do parametrów opisujących zadanie nie jest zaliczany czas jego wykonania. Jak łatwo zauważyć, czas wykonania zadania zależy bowiem od jego rozmiaru oraz od szybkości jego wykonania, która z kolei wiąże się poborem pewnej mocy. Dlatego zamiast pojedynczego parametru charakteryzującego czas wykonania zadania mówi się o modelu wykonania zadania. W literaturze znaleźć można dwa podstawowe modele wykonania zadania, tzw. model klasyczny i model dynamiczny.

5.1.1. Model klasyczny zadania

Model klasyczny zadania wykonywanego na procesorze VSP zaproponował w 1995 Yao, Demers i Schenker [13]. W modelu tym zadanie i charakteryzowane jest przez parametr w_i określany jako rozmiar tego zadania. Rozmiar zadania najczęściej interpretowany jest jako całkowita liczba cykli maszynowych procesora wymagana do wykonania zadania. Moc pobierana w trakcie wykonania zadania zależy od szybkości jego wykonania s i wyrażona jest za pomocą ciągłej wypukłej funkcji rosnącej $p(s)$, $s \in (0, \infty)$. Najczęściej przyjmuje się, że funkcja ta jest funkcją potęgową postaci:

$$p(s) = s^\alpha, \quad \alpha > 1 \quad (6)$$

Przy czym, jeśli wziąć pod uwagę zależności (4) i (5), dla technologii CMOS można przyjąć, że $\alpha=3$.

Jak łatwo zauważyć, w modelu tym zależność moc/szybkość jest dla każdego zadania taka sama, co nie daje szansy na wykorzystanie pełni możliwości oferowanych dzięki koncepcji procesora VSP z Rys.3.

5.1.2. Model dynamiczny zadania

Model dynamiczny zadania jest rozwinięciem znanego już dużo wcześniej ogólnego modelu zadania, do którego wykonania niezbędny był zasób ciągły, podwójnie ograniczony. W modelu tym chwilowa szybkość przetwarzania zadania i zależy od ilości mocy $p_i(t)$ przydzielonej temu zadaniu w tej chwili t a relację tę wyraża się przy pomocy funkcji $s_i(\cdot)$, zwanej dalej funkcją szybkości przetwarzania, w następujący sposób:

$$\dot{x}_i(t) = \frac{dx_i(t)}{dt} = s_i(p_i(t)), \quad x_i(0) = 0, \quad x_i(C_i) = w_i \quad (7)$$

gdzie:

- $x(t)$ – stan zadania i w chwili t ,
- $s_i(\cdot)$ – rosnąca (nieujemna), ciągła funkcja szybkości wykonywania zadania i , $s_i(0)=0$,
- $p_i(t)$ – ilość mocy przydzielona zadaniu i w chwili t ,
- w_i – zapotrzebowanie na przetwarzanie zadania i ,
- C_i – moment zakończenia (nieznany z góry) zadania i .

Każde z zadań charakteryzuje funkcja szybkości przetwarzania $s_i(\cdot)$ oraz zapotrzebowanie na przetwarzanie w_i , które podobnie jak w modelu klasycznym może być traktowane jako miara wielkości zadania.

Dodatkowo w modelu dynamicznym zakłada się trzy możliwości:

a) dostępne ilości mocy P i energii E są ograniczone i znane, czyli:

$$\sum_{i=1}^n p_i(t) \leq P \quad (8)$$

oraz

$$\sum_{i=1}^n \int_0^{C_{\max}} p_i(t) dt \leq E \quad (9)$$

gdzie

C_{\max} – jest nieznanym zawczasu momentem zakończenia ostatniego zadania w uszeregowaniu

b) znana i ograniczona jest moc (por. (8)) a minimalizowane jest zużycie energii

c) znana i ograniczona jest energia (por. (9)) a minimalizowany jest pobór mocy

Przypadek a) jest uogólnieniem znanego z literatury problemu laptopowego o dodatkowe ograniczenie na pobór mocy. Przypadki b) i c) są modyfikacjami problemu serwerowego. W każdej z tych modyfikacji, jeden z parametrów z pary moc-energia jest minimalizowany a drugi stanowi ograniczenie.

Model dynamiczny ma wiele zalet w porównaniu do modelu klasycznego. Jest to przede wszystkim model ogólniejszy a jego dynamiczny charakter w naturalny sposób pozwala rozważać sytuacje, gdy zadania mogą być wykonywane ze zmienną szybkością. W modelu klasycznym każde zadanie wykonywane być musiało ze stałą szybkością. Ponadto w modelu dynamicznym nie zakłada się o funkcji szybkości nic więcej ponad to, że jest ona ciągła i rosnąca. Dzięki temu wykazać można niektóre własności rozwiązań optymalnych dla szerszej klasy funkcji szybkości. Możliwość przypisania różnych funkcji szybkości do zadań daje z kolei możliwość rozważania problemów szeregowania w najnowocześniejszych architekturach procesorów VSP. Uwzględnienie w modelu ograniczenia na pobór mocy (lub minimalizacja tego poboru w problemach serwerowych) powoduje, że znajdowane uszeregowania spełniają ograniczenia zawarte w zaleceniach producenta procesora oraz są bezpieczne ze względu na ryzyko przegrzania układu procesora. Jest bowiem oczywiste, że pobierana przez procesor (lub ściślej - jego rdzeń) moc ma bezpośrednie przełożenie na proces nagrzewania się układu scalonego.

Modele klasyczny i dynamiczny są równoważne jeśli przyjąć, że:

$$s_i(\cdot) = s(\cdot) = p^{-1}(\cdot) = p^{1/\alpha}, \quad i = 1, 2, \dots, n, \alpha > 1$$

W szczególności, dla tradycyjnych architektur procesorów wykonanych w technologii CMOS można przyjąć, że:

$$s_i(\cdot) = p^{1/3}, \quad i = 1, 2, \dots, n \quad (10)$$

5.2. Sformułowanie problemu szeregowania na wielordzeniowym procesorze VSP

Problem szeregowania zadań obliczeniowych na wielordzeniowym procesorze VSP sformułowany może być jako deterministyczny problem szeregowania zadań na maszynach równoległych.

Rozważmy zbiór n zadań podzielnych oraz zbiór m maszyn równoległych identycznych. Każde z zadań do swego wykonania wymaga maszyny oraz pewnej ilości energii. Może być ono wykonywane w danej chwili na co najwyżej jednej maszynie, a każda maszyna może wykonywać w danej chwili co najwyżej jedno zadanie. Zadanie charakteryzowane jest przez rozmiar w_i oraz funkcję szybkości $s_i(\cdot)$ zgodnie z modelem dynamicznym zadania (7). Funkcja szybkości, potencjalnie różna dla różnych zadań, daje możliwość różnicowania zadań ze względu na konkretny typ relacji szybkość przetwarzania/moc. Przyjmuje się jednak, że we współczesnych technologiach mikroprocesorowych relacja ta ma postać funkcji ściśle wklęsłej. Takie założenie przyjmujemy i w tej pracy.

Należy znaleźć taki porządek zadań (lub ich fragmentów) na maszynach i jednocześnie taki przydział mocy do zadań (niekoniecznie stały), który minimalizuje wybrane kryterium szeregowania przy znanych ograniczeniach.

5.3. Czasooptymalny przydział mocy do zadań

Zanim przytoczymy kluczowe dla algorytmów szeregowania zadań na procesorach o zmiennej szybkości przetwarzania twierdzenie o czasooptymalnym przydziale mocy, pokażemy równie istotną własność klasy problemów rozważanych w tej pracy. Węglarz w [12] wykazał, że dla ściśle wklęsłych funkcji szybkości przetwarzania warunek:

$$\lim_{T \rightarrow \infty} T \sum_{i=1}^n s_i^{-1}(w_i / T) < E \tag{11}$$

jest spełniony dla dowolnego $E > 0$.

Oznacza to, że znalezienie rozwiązania dopuszczalnego ze względu na ilość dostępnej energii jest zawsze możliwe, poprzez odpowiednie wydłużenie czasu przetwarzania zadań (czyli zmniejszenie szybkości ich przetwarzania).

Twierdzenie o czasooptymalnym przydziale mocy do zadań sformułowane zostało w 1981 r. w [12]. Dotyczy ono wprawdzie przypadku, w którym między zadania rozdziela się optymalnie jedynie zasób ciągły (nie występują tu zasoby dyskretne, jak na przykład zbiór maszyn), ale w kolejnych rozdziałach zademonstrujemy jak można twierdzenie to wykorzystać w rozważanym przypadku problemu szeregowania zadań na maszynach równoległych.

Twierdzenie 1 (Węglarz [12])

Dla zadań charakteryzowanych ściśle wklęsłymi funkcjami szybkości wykonywania $s_i(\cdot)$, $i = 1, 2, \dots, n$ czasooptymalny przydział mocy do dany jest jako:

$$p_i^*(t) = p_i^* = s_i^{-1}(w_i / T^*), \quad i = 1, \dots, n, \quad t \in \langle 0, T^* \rangle$$

gdzie T^* jest jedynym dodatnim rozwiązaniem równania:

$$T \sum_{i=1}^n s_i^{-1}(w_i / T) = E, \tag{12}$$

jeśli $\sum_{i=1}^n s_i^{-1}(w_i / T) \leq P$

albo równania:

$$\sum_{i=1}^n s_i^{-1}(w_i / T) = P \quad (13)$$

w przeciwnym przypadku.

Z powyższego twierdzenia wynikają następujące wnioski dotyczące czasooptymalnego przydziału mocy do zadań:

- w uszeregowaniu czasooptymalnym zadania wykonywane są ze stałym poborem mocy,
- w uszeregowaniu czasooptymalnym wszystkie zadania wykonywane są równoległe,
- w uszeregowaniu czasooptymalnym wszystkie zadania kończą się w tym samym momencie,
- w uszeregowaniu czasooptymalnym zużywana jest cała dostępna energia lub pobierana jest cała dostępna moc (albo występują obie te sytuacje jednocześnie).

Zauważmy, że (12) jest naturalną konsekwencją (9) a (13) wynika z (8). Szczególnie warte podkreślenia jest jednak to, że równanie (13) można rozwiązać analitycznie dla praktycznego przypadku funkcji szybkości postaci:

$$s_i = p^{1/\alpha_i}, \quad i = 1, 2, \dots, n \quad (14)$$

gdzie $\alpha_i \in \{2, 3, 4\}$

Natomiast równanie (12) przyjmuje postać równania algebraicznego stopnia co najwyżej 4 (czyli rozwiązywalnego analitycznie) dla funkcji (14) i $\alpha_i \in \{2, 3, 4, 5\}$.

5.4. Ogólna metodyka znajdowania rozwiązań optymalnych problemu

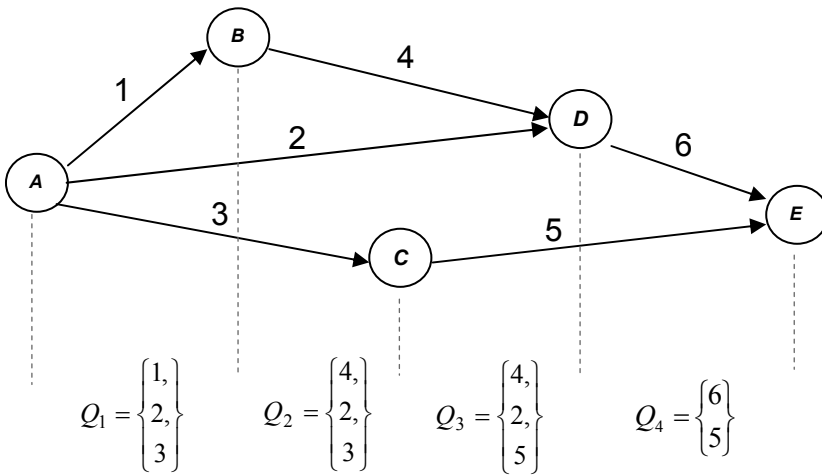
Ogólna metodyka rozwiązywania problemów szeregowania zadań opisanych za pomocą modelu dynamicznego (7) na maszynach równoległych opiera się na Twierdzeniu 1. Polega ona na sformułowaniu odpowiedniego problemu nieliniowego problemu programowania matematycznego [9]. W problemie tym poszukuje się rozdziału rozmiarów zadań w_i , $i = 1, 2, \dots, n$, pomiędzy wszystkie dopuszczalne kombinacje n zadań na m maszynach, który minimalizowałby długość całego uszeregowania. Każda kombinacja zawiera indeksy dokładnie m zadań, które mogą być wykonane równoległe na maszynach. Wszystkich takich kombinacji, dla przypadku zadań niezależnych jest więc: $\binom{n}{m}$.

Długość uszeregowania odpowiadającego każdej kombinacji dla określonego przydziału fragmentów rozmiarów zadań wykonywanych w tej kombinacji znajduje się z Twierdzenia 1. Całkowita długość uszeregowania jest więc sumą długości części uszeregowania odpowiadających poszczególnym kombinacjom.

Podobna metodyka może być stosowana również dla bardziej skomplikowanych przypadków, co opisane zostało w kolejnych rozdziałach.

5.4.1. Problem laptopowy - zadania zależne

Rozważmy teraz laptopowy problem minimalizacji długości uszeregowania zadań zależnych przy ograniczonej ilości mocy (8) i energii (9). Założymy przy tym dla ułatwienia, że przetwarzanie wszystkich zadań może być rozpoczęte w tej samej chwili (momenty gotowości zadań są takie same). Do reprezentowania ograniczeń kolejnościowych występujących między zadaniami wykorzystywać będziemy graf skierowany typu ToA (ang. *Task-on-Arc*), w którym łuki reprezentować będą zadania a węzły zdarzenia. Będziemy mówić, że uporządkowanie wierzchołków w grafie jest dopuszczalne, jeśli dla każdego zadania, reprezentujący je łuk będzie się zaczynał w wierzchołku poprzedzającym wierzchołek końcowy łuku w tym uporządkowaniu. Przykładowy graf ToA z dopuszczalnym uporządkowaniem wierzchołków ($A < B < C < D < E$) dla przypadku sześciu zadań zależnych pokazano na Rys. 4. W takiej reprezentacji wygodnie jest wygenerować sekwencję podzbiorów zadań Q_k , $k = 1, 2, \dots, r$, które mogą być wykonywane równoległe bez naruszenia ograniczeń kolejnościowych. Jeśli w zbiorze Q_k występuje nie więcej niż m zadań, oznacza to, że wykonanie zadań z takiego zbioru jest również dopuszczalne ze względu na ograniczoną liczbę maszyn i zbiór Q_k stanowi zarazem dopuszczalną kombinację $|Q_k|$ zadań na m maszynach. W przeciwnym przypadku ($|Q_k| > m$) podzbiór Q_k należy podzielić na sekwencję (w dowolnym porządku) wszystkich kombinacji $|Q_k|$ zadań na m maszynach (na Rys.4 pokazano to na przykładzie instancji problemu gdzie $m = 2$). Następnie rozmiary zadań należy optymalnie rozdzielić pomiędzy te kombinacje z wynikowej sekwencji kombinacji zadań na maszynach, do których te zadania należą. W tym celu konieczne jest rozwiązanie odpowiedniego problemu nieliniowego programowania matematycznego [10]. Optymalny dla danego porządku wierzchołków grafu ToA rozdział rozmiarów zadań to oczywiście taki, który dzięki zastosowaniu Twierdzenia 1 do każdej kombinacji, zagwarantuje, że długość uszeregowania będzie minimalna. Takie podejście gwarantuje zachowanie ograniczeń kolejnościowych (sekwencja kombinacji jest dopuszczalna) jak i ograniczenia wynikającego ze skończonej liczby maszyn (kombinacje zawierają co najwyżej m zadań). Niestety, sekwencja kombinacji zadań, a tym samym długość uszeregowania, zależy od konkretnego dopuszczalnego uporządkowania wierzchołków, a to może być niejedyne (dla przykładowego grafu z Rys.4 alternatywnym uporządkowaniem dopuszczalnym wierzchołków jest: $A < C < B < D < E$). W takim przypadku, w celu znalezienia uszeregowania o minimalnej długości należy rozwiązać problem programowania nieliniowego dla każdego dopuszczalnego uporządkowania wierzchołków grafu. Liczba takich uporządkowań w ogólności rośnie wykładniczo wraz ze wzrostem liczby wierzchołków w grafie, co czyni metodę pełnego przeglądu nieefektywną z obliczeniowego punktu widzenia.



Sekwencja kombinacji dla danego uporządkowania wierzchołków:

$$\begin{Bmatrix} 1, \\ 2 \end{Bmatrix}, \begin{Bmatrix} 1, \\ 3 \end{Bmatrix}, \begin{Bmatrix} 2, \\ 3 \end{Bmatrix}, \begin{Bmatrix} 4, \\ 2 \end{Bmatrix}, \begin{Bmatrix} 4, \\ 3 \end{Bmatrix}, \begin{Bmatrix} 2, \\ 3 \end{Bmatrix}, \begin{Bmatrix} 4, \\ 2 \end{Bmatrix}, \begin{Bmatrix} 4, \\ 5 \end{Bmatrix}, \begin{Bmatrix} 2, \\ 5 \end{Bmatrix}, \begin{Bmatrix} 6, \\ 5 \end{Bmatrix}$$

Rys. 4. Przykładowy graf ToA i odpowiadająca mu sekwencja kombinacji zadań

Fig. 4. An exemplary ToA graph and related sequence of combinations of jobs

5.4.2. Problem serwerowy – zadania niezależne

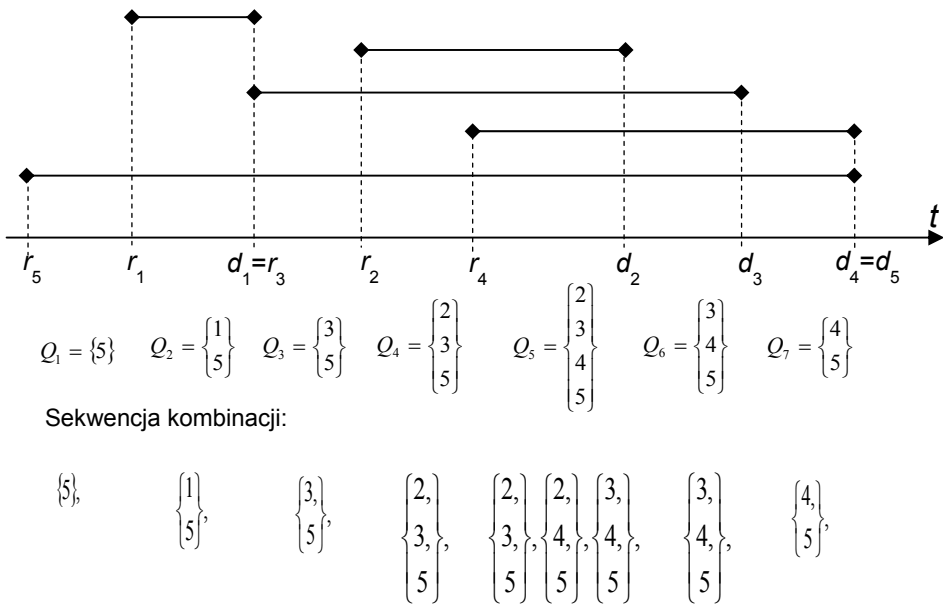
W niniejszym rozdziale zademonstrowane zostanie zastosowanie ogólnej metodyki zaprezentowanej na początku Rozdz.5.4. do przypadku problemu serwerowego. Zakładając teraz będziemy że zadania niezależne są opisane dodatkowo za pomocą czasów gotowości $r_i, i = 1, 2, \dots, n$, oraz pożądaných terminów zakończenia $d_i, i = 1, 2, \dots, n$, przy czym porządek numerów zadań ustalony jest zgodnie z niemalejącym porządkiem $d_i, i = 1, 2, \dots, n$.

Kryterium optymalizacji jest minimalizacja ilości energii, niezbędnej do tego by wszystkie zadania zakończyły się w założonym terminie, przy założeniu ograniczonego chwilowego poboru mocy (8).

Do znalezienia rozwiązania optymalnego rozważanego tu problemu serwerowego z powodzeniem zastosować można ogólną metodykę polegającą na sformułowaniu odpowiedniego nieliniowego problemu programowania matematycznego [11]. Tym razem jednak rozmiary zadań dzielone będą między kombinacje zadań, których równoległe wykonanie zagwarantuje uzyskanie rozwiązania dopuszczalnego ze względu na czasy gotowości i pożądanę terminy zakończenia zadań, a także ze względu na skończoną liczbę maszyn. Sposób generowania odpowiedniej sekwencji kombinacji dla przykładu pięciu zadań i trzech maszyn pokazany został na Rys.5. Twierdzenie 1 w problemie programowania matematycznego jest tym razem wykorzystywane do obliczenia poboru

mocy a zarazem długości części uszeregowania odpowiadającej każdej kombinacji z sekwencji. Oczywiście obliczony z Twierdzenia 1 pobór mocy przez zadania w każdej kombinacji, który wynika z rozdziału rozmiarów zadań między te kombinacje, musi gwarantować globalnie minimalne zużycie energii w całym uszeregowaniu.

Warto zauważyć, że przyjęcie założenia (8) może spowodować, że rozwiązanie dopuszczalne danej instancji problemu nie istnieje. Przypadek taki wystąpi gdy pobór mocy niezbędny do wykonania zadań przed pożądanymi terminami ich zakończenia przekracza dopuszczalny limit mocy P . Sprawdzenie istnienia rozwiązania dopuszczalnego wymaga rozwiązania odpowiedniego układu równań i nierówności nieliniowych.



Rys.5. Przykładowy dopuszczalny porządek zadań i odpowiadająca mu sekwencja kombinacji zadań

Fig. 5. An exemplary feasible ordering of jobs and related sequence of combinations of jobs

5.5. Algorytmy heurystyczne

Ogólna metodyka zastosowana do przypadków przedstawionych w Rozdziale 5.4. ma jedną zasadniczą wadę. Liczba zmiennych w nieliniowych problemach programowania matematycznego formułowanych w tym podejściu rośnie wykładniczo wraz z liczbą zadań w instancji. W konsekwencji, metodyka ta, choć prowadzi do rozwiązań dokładnych, z praktycznego punktu widzenia jest niepraktyczna. Rozwiązanie odpowiedniego problemu

programowania matematycznego wymagałoby uruchomienia zaawansowanego solwera nieliniowego, co poważnie wpłynęłoby na efektywność zaimplementowanego w ramach systemu operacyjnego modułu szeregującego zadania na rdzeniach procesora.

Jako, że nie jest znany inny algorytm dokładny znajdowania rozwiązań optymalnych problemu w czasie, który wielomianowo zależałby od rozmiaru instancji, uzasadnione jest wykorzystanie algorytmów heurystycznych.

Generalnie, możliwe są dwa główne podejścia heurystyczne do rozwiązywania rozważanych problemów szeregowania. W pierwszym podejściu najpierw przydziela się do zadań stałą ilość pobieranej przez nich mocy (ustalając w ten sposób czas ich trwania), a następnie buduje się dopuszczalne uszeregowanie zadań na maszynach z uwzględnieniem konieczności minimalizowania ustalonego kryterium. W drugim podejściu najpierw określa się porządek wykonania zadań na maszynach, później oblicza się przydział (najlepiej optymalny dla danego porządku) mocy do zadań. W podejściu tym zakłada się jednak, że obliczenie przydziału mocy do uporządkowanych na maszynach zadań nie wymaga użycia solwera nieliniowego, a może być dokonane analitycznie. Wymaga to oczywiście przyjęcia stosownych założeń dotyczących zasad tworzenia porządku zadań na maszynach.

Osobną kwestię stanowi problem poszukiwania najlepszego uporządkowania wierzchołków w grafie ToA dla problemu rozważanego w Rozdz. 5.4.1. W pracy [10] zaprezentowano wiele ciekawych heurystyk, które wykorzystują informacje na przykład na temat długości ścieżek do danego wierzchołka, liczby ścieżek do danego wierzchołka, liczby wierzchołków poprzedzających, itd. itp.

Wszystkie proponowane podejścia heurystyczne muszą spełniać jeden podstawowy wymóg: muszą być efektywne – a więc w krótkim czasie muszą dawać rozwiązanie, które jest bliskie (w sensie wartości funkcji celu) rozwiązaniu optymalnemu. Dzięki takim heurystykom moduł szeregujący na poziomie systemu operacyjnego, wspierany przez odpowiednie rozwiązania sprzętowe, będzie w stanie sprawnie i racjonalnie przydzielać zadania obliczeniowe do rdzenia procesora o zmiennej szybkości przetwarzania.

6. Podsumowanie

W pracy społeczną odpowiedzialność biznesu ukazano w kontekście przykładowych działań, które mogą być podejmowane przez firmy z branży IT w celu ochrony środowiska naturalnego. Spodziewane zyski z tytułu wdrożenia proponowanych rozwiązań zarówno sprzętowych (architektury systemów komputerowych) jak i programowych (moduł szeregowania zadań na procesorze wielordzeniowym) są trudne na razie do oszacowania. Rachunek kosztów i ewentualnych zysków wymaga dokładnej analizy efektywności proponowanych algorytmów heurystycznych oraz empirycznej weryfikacji stosowanego dynamicznego modelu wykonywania zadań. Dlatego kolejnym etapem badań nad potencjałem proponowanych rozwiązań powinno być albo przeprowadzenie badań na fizycznym prototypie procesora o zmiennej szybkości przetwarzania, albo dokonanie stosownej symulacji.

Wypada mieć nadzieję, że w obliczu ciągle rosnącego udziału zużycia energii przez branżę IT na globalnym rynku energii, zaproponowany w pracy kierunek badań nad zmniejszeniem energochłonności obliczeń jest bardzo obiecujący. Firmy, które podążają tą lub podobną ścieżką rozwoju swoich produktów i/lub usług można z powodzeniem zaliczyć do biznesu społecznie odpowiedzialnego.

Podziękowania

Praca jest częściowo finansowana z grantu Ministerstwa Nauki i Szkolnictwa Wyższego, grant nr N N519 403437.

Bibliografia

- [1] Boyer F.R., Epassa H.G. and Savaria Y., Embedded power-aware cycle by cycle variable speed processor, *IEE Proc.-Comput. Digit. Tech.*, 2006, 153(4) .283-290.
- [2] Budzyński W., Public Relations. Zarządzanie reputacją firmy, Poltex, Warszawa 2005.
- [3] Bunde, D.P., Power-aware Scheduling for Makespan and Flow, *Proceedings of The Eighteenth Annual ACM Symposium on Parallelism in Algorithms and Architectures*, Cambridge, Massachusetts, USA, 2006, 190-196.
- [4] Burd T.D., Pering T.A., Stratakos A.J., Brodersen R.W., A Dynamic Voltage Scaled Microprocessor System, *IEEE Journal of Solid-state Circuits*, 2000, 35(11) 1571-1579.
- [5] Chandrakasan A., Brodersen R.W., Low Power Digital Cmos Design, Kluwer Academic Publisher, 2002
- [6] Koźmiński A., Piotrowski W., Zarządzanie Teoria i Praktyka, Wyd. Naukowe PWE, Warszawa 1995.
- [7] Lewicka-Strzałecka A., Odpowiedzialność moralna w życiu gospodarczym, Wyd. IFiS PAN, Warszawa 2006.
- [8] Rok B., Odpowiedzialny biznes w nieodpowiedzialnym świecie, Akademia Rozwoju Filantropii w Polsce. Forum Odpowiedzialnego Biznesu, Warszawa 2004.
- [9] Różycki R., Węglarz J., On job models in power management problems, *Bulletin of the Polish Academy of Science: Technical Sciences*, 2009, 57(2), 147-151.
- [10] Różycki R., Węglarz J., Power-aware scheduling of preemptable jobs on identical parallel processors to minimize makespan, *Annals of Operations Research*, 2011, doi:10.1007/s10479-011-0957-5.
- [11] Różycki R., Węglarz J., Power-aware scheduling of preemptable jobs on identical parallel processors to meet deadlines, *European Journal of Operational Research*, 2012, 218(1), 68-75.
- [12] Węglarz, J., Project scheduling with continuously-divisible doubly constrained resources, *Management Science*, 1981, 27 (9), 1040-1053.
- [13] Yao F., Demers A., & Shenker, S., A scheduling model for reduced cpu energy, *Proceedings of IEEE Symposium on Foundations of Computer Science*, 1995, 374- 382.
- [14] Zbiegień-Maciąg L., Etyka w zarządzaniu organizacją, w: J. Dietl, W. Gasparski, Etyka biznesu, Warszawa 2002, 217-229.

