

# Maciej Roszkowski

---

## Metody analizy wydajności systemów operacyjnych w kontekście rozwoju społeczeństwa informacyjnego

---

Ekonomiczne Problemy Usług nr 117, 759-766

---

2015

Artykuł został opracowany do udostępnienia w internecie przez Muzeum Historii Polski w ramach prac podejmowanych na rzecz zapewnienia otwartego, powszechnego i trwałego dostępu do polskiego dorobku naukowego i kulturalnego. Artykuł jest umieszczony w kolekcji cyfrowej [bazhum.muzhp.pl](http://bazhum.muzhp.pl), gromadzącej zawartość polskich czasopism humanistycznych i społecznych.

Tekst jest udostępniony do wykorzystania w ramach dozwolonego użytku.

*MACIEJ ROSZKOWSKI*

Zachodniopomorski Uniwersytet Technologiczny<sup>1</sup>

## METODY ANALIZY WYDAJNOŚCI SYSTEMÓW OPERACYJNYCH W KONTEKŚCIE ROZWOJU SPOŁECZEŃSTWA INFORMACYJNEGO

### Streszczenie

Jednym z warunków właściwego rozwoju społeczeństwa informacyjnego jest możliwość szybkiego dostępu do wydajnego systemu e-usług informatycznych. Dlatego artykuł podejmuje temat metod analizy wydajności platformy e-usług. Przedstawiono cztery antymetody i dwie właściwe metody analizy wydajności systemów operacyjnych (metoda USE i metoda TSA).

**Słowa kluczowe:** analiza wydajności, metoda USE, metoda TSA.

### Wprowadzenie

Spółeczeństwo informacyjne to takie społeczeństwo, które użytkuje systemy informatyczne oraz usługi telekomunikacyjne do przekazywania i przetwarzania informacji (Nowak 2008, s. 25). Systemy informatyczne są wykorzystywane w wielu dziedzinach życia społecznego (np. opiece zdrowotnej, administracji, edukacji, bankowości) w procesie obsługi społeczeństwa informacyjnego (Bytniewski 2013, s. 9).

Systemy informatyczne e-usług rezydują na wielu platformach: na systemie operacyjnym Android (np. realizacja e-usług bankowości elektronicznej poprzez aplikacje zainstalowane na smartfonach), na systemie operacyjnym Microsoft Windows (np. do obsługi wersji klienckich na komputerach stacjonarnych: obsługa pacjentów przychodni medycznych, obsługa klientów usług bankowości), na systemie operacyjnym Unix/Linux w różnych wersjach dystrybucji (np. serwery umożliwiające komunikację jednostek podległych NFZ), na systemie operacyjnym VM-

---

<sup>1</sup> Wydział Informatyki, Katedra Inżynierii Systemów Informacyjnych.

ware vSphere do wirtualizacji środowisk (np. farmy serwerów wysokiej dostępności do autoryzacji kart kredytowych).

Spółeczeństwo informacyjne może rozwijać się poprzez rozwój infrastruktury, kompetencji elektronicznych i e-usług. Te trzy elementy stanowią całość i warunkują właściwy rozwój społeczeństwa informacyjnego. U podstaw działania infrastruktury i e-usług stoi platforma (system operacyjny), na której rezydują urządzenia sieciowe (infrastruktura) i system informacyjny (e-usługi). Kompetencje elektroniczne są wykształcane w toku użytkowania infrastruktury i e-usług.

Jeżeli system operacyjny działa mało wydajnie, to ma to znaczący wpływ na chęć wykorzystywania kompetencji elektronicznych, za pośrednictwem infrastruktury informatycznej, w celu skorzystania z e-usług. Zbyt wiele jednoczesnych połączeń klienckich w infrastrukturze informatycznej do serwera e-usług może spowodować jego przeciążenie, a w konsekwencji bardzo powolne działanie e-usług lub przerwy w dostępie do nich. Aby optymalizować wydajność systemów operacyjnych, tak aby mogły działać efektywnie i niezawodnie, potrzebne jest przeprowadzenie właściwej analizy wydajności.

Celem niniejszej publikacji jest przedstawienie metod analizy wydajności systemów operacyjnych w kontekście rozwoju społeczeństwa informacyjnego. Poniżej zostaną przedstawione antymetody analizy wydajności, pokazane jako tło dla właściwych metod analizy wydajności.

## 1. Antymetody analizy wydajności

Przedstawione antymetody analizy wydajności stanowią przykład dosyć częstych zachowań w organizacjach odpowiedzialnych za obsługę systemów operacyjnych (Usenix 2014). Prezentowane są nie po to, żeby je powielać, tylko po to, aby się ich wystrzegać.

### Jawna antymetoda

Jawną antymetodę (ang. *Streetlight Anti-Method*) cechuje brak metody. Analiza wydajności następuje za pomocą dobranych przypadkowo narzędzi monitorowania, w nadziei na znalezienie czegoś ciekawego. Tak badana wydajność może przyczynić się do odkrycia jakichś problemów, ale niekoniecznie tych, które są szukane i odpowiadają za niższą wydajność. Próby wpływania na wydajność przyjmują najczęściej formę prób i błędów, zachodzących podczas zmian różnych parametrów.

Jeżeli nawet działania te przyniosą jakieś efekty, to cały proces dochodzenia do nich będzie bardzo powolny, a przy okazji można znaleźć parę innych problemów, które wywiodą badacza na manowce.

### Antymetoda losowej zmiany

Antymetoda losowej zmiany (ang. *Random Change Anti-Method*) opiera się na eksperymentach. Dokonując losowego wyboru jakiegoś elementu, modyfikuje

się jego wartość w jednym kierunku (zwiększając wartość), jednocześnie dokonując pomiaru wydajności. Następnie dokonuje się modyfikacji wartości tego samego elementu w drugim kierunku (zmniejszając wartość) i ponownie dokonuje się pomiaru wydajności. Pomiar wydajności może obejmować różne metryki: czas wykonywania operacji, czas działania aplikacji, opóźnienie, przepustowość. Porównując uzyskane w ten sposób dwie wartości można wywnioskować, która z nich spowodowała wzrost wydajności.

Wykonane w ten sposób działania mogą spowodować, że zostanie osiągnięty wzrost wydajności, ale tylko przy określonym obciążeniu. Prezentowana antymetoda nie wyszukuje błędów, ale je obchodzi poprzez modyfikację parametrów. Usunięcie błędu, który powodował spadek wydajności, może wręcz obniżyć badaną wydajność, ze względu na wcześniej poczynione modyfikacje za pomocą tej antymetody. Konsekwencją wprowadzania losowych zmian może być spadek wydajności w innym przypadku, który nie został wzięty pod uwagę, np. przy maksymalnym obciążeniu środowiska produkcyjnego.

#### **Antymetoda obwiniania kogoś innego**

Antymetoda obwiniania kogoś innego (ang. *Blame-Someone-Else Anti-Method*) jest stosowana w środowiskach ścisłej specjalizacji. Każda osoba odpowiada za działanie konkretnego systemu lub komponentu środowiska i w tej wąskiej dziedzinie się specjalizuje. W przypadku pojawienia się problemów z wydajnością, dotyczącą jego systemu lub komponentu środowiska, zamiast diagnozować problem, stara się go przetrzucić na innych. W tym celu stawia hipotezę, że obniżona wydajność jest spowodowana błędami w konfiguracji innego systemu lub komponentu środowiska, za który on sam nie odpowiada. Dlatego też w jego mniemaniu osoba odpowiedzialna za ten fragment powinna zdiagnozować problem. W przypadku gdy oskarżenie nie znajdzie uzasadnienia, stawiana jest nowa hipoteza i nowa osoba przystępuje do testów wydajnościowych.

Zrzucanie winy na innych prowadzi do marnowania zasobów organizacji. Zespoły, których problem nie dotyczy – muszą go rozwiązywać tylko po to, żeby udowodnić swoją rację. Winę za taki stan rzeczy ponosi kierownictwo lub mała asertywność samych zespołów. W przypadku wysnucia takiej hipotezy wystarczy poprosić o dowód świadczący o tym, że wadliwym punktem może być nasz system lub komponent środowiska.

#### **Antymetoda pasywnych testów wydajności**

Pasywne testy wydajności (ang. *Passive Benchmarking Anti-Method*) mają za zadanie zebrać dane o wydajności bez wnikania w szczegóły. Po uruchomieniu testów wyniki są prezentowane dla ogółu.

Wygenerowane dane końcowe nie zawsze muszą odzwierciedlać rzeczywistość, a co za tym idzie wnioski, które są wysnute na ich podstawie, mogą być nieprawidłowe. Przykładowo można sądzić, że sprawdza się wydajność komponentu X, natomiast w rzeczywistości dokonuje się pomiaru wydajności komponentu Y,

a wnioskuje się o pomiarze komponentu Z. Tak przeprowadzone testy wydajności mogą nie brać pod uwagę różnych ograniczeń, niezwiązanych z celem testu, np. obciążenie sieci, nieoptymalna konfiguracja sprzętowa. Tego rodzaju testy są bardzo podatne na perturbacje (np. tworzenie kopii zapasowych w czasie przeprowadzania testów).

## 2. Metody analizy wydajności

W większości zastosowań analizy wydajności sprawdzają się dwie metody: metoda USE i metoda TSA. Obydwie metody wzajemnie się uzupełniają, gdyż każda z nich opiera się na innym podejściu. Metoda USE opiera się na analizie zasobów (podejście sprzętowe), natomiast metoda TSA – na analizie procesów (podejście programowe).

Zasób systemu operacyjnego można zdefiniować jako komponenty sprzętowe, zarządzane przez menedżera zasobów (system operacyjny). Proces może uzyskać, wykorzystać i zwolnić dostęp do zasobu. System operacyjny dba o uporządkowany i kontrolowany przydział zasobów. Wątek systemu operacyjnego to wydzielony fragment programu wykonywany przez procesor. Każde uruchomienie dowolnej aplikacji w systemie operacyjnym uruchamia jeden lub wiele procesów, a każdy powołany do życia proces tworzy wiele wątków. Wątki są wykonywane współbieżnie w obrębie jednego procesu, współdzieląc przestrzeń adresową pamięci.

### Metoda USE

Metoda USE jest stosowana na początkowym etapie analizy wydajności, gdyż umożliwia wykrycie wąskich gardeł w systemie operacyjnym (Gregg 2014, s. 42). W myśl metody USE dla każdego zasobu systemu operacyjnego powinny być sprawdzone trzy typy metryk: poziom wykorzystania (ang. *Utilization*), poziom nasycenia (ang. *Saturation*) i liczba błędów (Gregg 2014).

Metryka błędów jest liczbą błędów.

Poziom wykorzystania zasobu może być wyrażony w postaci czasu lub pojemności. Wyrażony w postaci czasu oznacza ilość czasu poświęconego na aktywne wykonywanie zadania przez zasób, w danym okresie czasu (np. zasób procesor). Wyrażona w postaci pojemności oznacza ilość zużytej pojemności (np. zasób pamięć masowa).

Poziom nasycenia to stopień zajętości zasobu, w którym zasób nie może już wykonać kolejnych zadań w danej chwili.

Algorytm metody USE został przedstawiony na rysunku 1. W kroku pierwszym metody USE należy wyselekcjonować listę zasobów, wraz z konkretnymi przykładami (Tanenbaum 2010, s. 35):

1. Procesory – liczba rdzeni, procesory wirtualne, wielowątkowość współbieżna (*hyper-threading*),

2. Pamięci operacyjne – RAM,
3. Pamięci masowe – pamięć magnetyczna (dyski twarde talerzowe), pamięć półprzewodnikowa (dyski SSD, USB),
4. Interfejsy sieciowe – karty ethernet.

Nie wymienione komponenty sprzętowe, takie jak kontrolery, magistrale, bufor, szyny wejścia–wyjścia, najczęściej można pominąć przy ocenie wydajności, gdyż projektuje się je w celu zapewnienia bardzo dużej przepustowości. W drugim kroku metody USE należy określić dla każdego zasobu – typ metryki (poziom wykorzystania, poziom nasycenia, liczba błędów), który może zostać wykorzystany w jego analizie (tabela 1):

Tabela 1

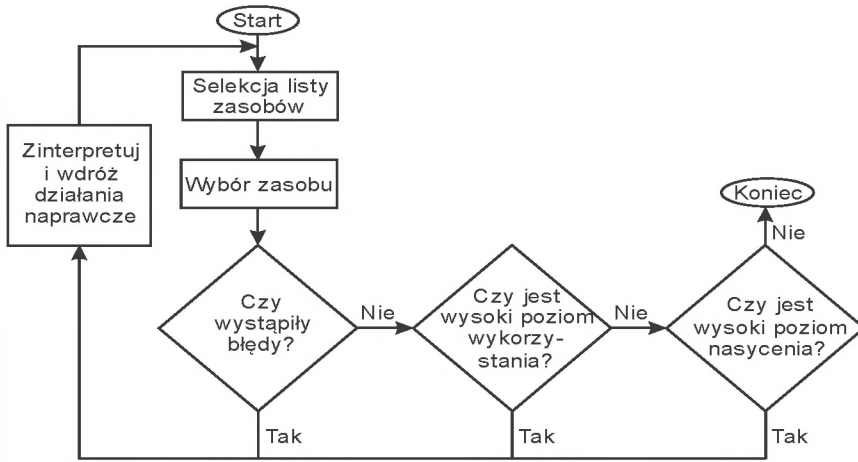
Przykładowe wyrażenie dwóch metryk dla zasobów  
w metodzie USE (poziom wykorzystania i poziom nasycenia)

Lp.	Zasób/Metryka	Poziom wykorzystania	Poziom nasycenia
1	Procesory	Oceniany na podstawie każdego procesora oddzielnie lub wartości średniej dla wszystkich procesorów	Zależny od długości kolejki zadań bieżących, czekających na dostęp do procesora
2	Pamięci operacyjne	Zależny od ilości wolnej pamięci (dostępnej dla systemu operacyjnego)	Zależny od wielkości stronicowanej pamięci
3	Pamięci masowe	Wskazuje na wielkość zajętej przestrzeni pamięci masowej	Zależny od długości kolejki oczekiwania
4	Interfejsy sieciowe	Zależny od przepustowości i szybkości transmisji	Zależny od możliwości przepełnienia bufora

Źródło: opracowanie własne.

Bardzo rzadko mierzy się błędy w przypadku zasobów: procesor i pamięć. Występowanie błędów w przypadku tych zasobów świadczy najczęściej o przegrzewaniu się tych zasobów lub o postępującym uszkodzeniu fizycznym zasobów. Występujące błędy na pamięciach masowych są bardzo często spowodowane postępującą degradacją powierzchni dysków. Mechanizm S.M.A.R.T. jest odpowiedzialny za monitorowanie i powiadamianie o błędach dysków w systemie operacyjnym. Model warstwowy struktury protokołów transmisji TCP/IP umożliwia komunikację internetową za pomocą protokołów komunikacyjnych, które umożliwiają przesyłanie danych w postaci pakietów za pomocą interfejsów sieciowych. W przypadku wystąpienia błędów podczas transmisji protokoły komunikacji umożliwiają ich retransmisję lub poprawianie błędów. Najczęstszą przyczyną błędów jest fizyczne uszkodzenie łącza danych lub rzadziej interfejsu sieciowego.

Ostatnim krokiem metody USE jest właściwa interpretacja trzech rodzajów metryk dla poszczególnych zasobów oraz podjęcie działań naprawczych.



Rys. 1. Algorytm metody USE

Źródło: opracowanie własne.

### Metoda TSA

Metoda TSA (ang. *Thread State Analysis Method*) umożliwia ustalenie wydajności danego procesu, co w konsekwencji umożliwia określenie wydajności wszystkich aplikacji (Gregg 2014, s. 168).

W kroku pierwszym metody TSA należy dla każdego procesu określić stany, w których przebywa, w całym okresie badawczym. Najczęściej wyróżnia się następujące stany procesu (podział ten nie wyczerpuje wszystkich możliwych stanów w systemie operacyjnym) (Silberschatz 2003, s. 117):

1. Stan wykonywany (ang. *executing*) – proces jest wykonywany przez procesor (po stronie użytkownika i jądra) (czasami nazywany aktywnym),
2. Stan gotowy (ang. *runnable*) – proces oczekuje na dostęp do procesora,
3. Stan oczekujący (ang. *waiting*) – proces oczekuje na zdarzenie lub sygnał.

Diagram stanów procesu został przedstawiony na rysunku 2.

W kroku drugim metody TSA dla każdego z podanych stanów należy zmierzyć czas trwania danego stanu.



Rys. 2. Diagram stanów procesu

Źródło: opracowanie własne.



Krok trzeci ma charakter ekspercki, ze względu na konieczność posiadania wiedzy o budowie aplikacji, która powołała do życia konkretny proces, a w konsekwencji wątek. Należy wyselekcjonować te stany, w których proces przebywa najdłużej (najdłuższy czas trwania) i poddać je analizie.

1. Dłuższe przebywanie w stanie wykonywania może oznaczać, że proces nie-ekonomicznie zajmuje czas procesora. Metoda „profilowania procesora” umożliwia ustalenie, które linie kodu źródłowego aplikacji są za to odpowiedzialne, a w konsekwencji ekonomiczne zaplanowanie zużycia procesora.
2. Spędzanie większej ilości czasu w stanie gotowości (oczekiwania na dostęp do procesora) może oznaczać potrzebę zapewnienia większej mocy obliczeniowej. Większą moc obliczeniową można zapewnić przydzielając dodatkowe procesory lub usuwając limity nałożone na procesor.
3. Stan oczekiwania jest przyczyną dużych opóźnień w działaniu aplikacji, gdyż większość zdarzeń może być długotrwała lub zależna od programisty aplikacji i jego wizji zapobiegania konfliktom w dostępie do zasobów w środowisku wielozadaniowym. Zajście zdarzenia lub wysłanie sygnału kontynuacji umożliwia wznowienie wykonywania procesu.

Ostatnim krokiem metody TSA jest właściwa analiza najdłuższych stanów procesów oraz podjęcie działań naprawczych.

## Podsumowanie

Przedstawione antymetody są przykładem dosyć częstych zachowań w organizacjach odpowiedzialnych za obsługę systemów operacyjnych. Najczęściej wykorzystywanie antymetod w codziennej pracy jest spowodowane chaosem i chęcią doraźnego naprawienia sytuacji spadku wydajności. Jedynie działania zorganizowane, oparte na wypracowanych metodach analizy wydajności, mogą się sprawdzić w toku obsługi systemów operacyjnych (platform e-usług).

Właściwe metody analizy wydajności (metoda USE i metoda TSA) zostały pokazane na tle antymetod. Należy zwrócić uwagę na zestawienie tych metod razem. Obydwie metody wzajemnie się uzupełniają, gdyż prezentują odmienne podejście. Metoda USE prezentuje podejście sprzętowe, w którym analizowane są pod względem wydajności komponenty serwerów, na których jest umiejscowiony system operacyjny. Natomiast metoda TSA prezentuje podejście odmienne, które uzupełnia działanie metody USE. W metodzie TSA prezentowane jest podejście programowe, gdyż analizie podlegają procesy samej aplikacji, czyli wewnątrz systemu informatycznego wraz z platformą e-usług.

Wyposażeni w możliwości, które daje zestawienie metody USE i metody TSA do analizy wydajności systemów operacyjnych, można w bardzo prosty sposób przyczynić się do rozwoju społeczeństwa informacyjnego.



**Literatura**

1. Bytniewski A. (red.) (2013), *Systemy informatyczne a rozwój społeczeństwa informacyjnego*, Uniwersytet Ekonomiczny we Wrocławiu, Wrocław.
2. Gregg B. (2014), *Systems Performance: Enterprise and the Cloud*, Prentice Hall, Michigan.
3. Nowak J.S. (2008), *Spoleczeństwo informacyjne – geneza i definicje*, w: P. Sienkiewicz, J.S. Nowak (red.), *Spoleczeństwo informacyjne. Krok naprzód, dwa kroki wstecz*, PTI, Katowice.
4. Silberschatz A., Galvin P.B., Gagne G. (2003), *Podstawy systemów operacyjnych*, WNT, Warszawa.
5. Tanenbaum A S. (2010), *Systemy operacyjne*, Helion, Gliwice.
6. [www.brendangregg.com/methodology.html](http://www.brendangregg.com/methodology.html) (2014).
7. [www.usenix.org/conference/lisa12/performance-analysis-methodology](http://www.usenix.org/conference/lisa12/performance-analysis-methodology) (2014).

**METHODS OF OPERATIONAL SYSTEMS' EFFICIENCY ANALYSE  
IN THE CONTEXT OF THE INFORMATION SOCIETY****Summary**

A possibility of a quick access to the efficient system of e-services is one of the basic condition for an information society development. For this reason this article describes methods of operational systems' efficiency analyses. There are presented four anti-methods and two appropriate methods of operational systems' efficiency analyses (USE Method, TSA Method).

**Keywords:** analysis of efficiency, USE Method, TSA Method.

*Translated by Maciej Roszkowski*