

Maciej MŁYŃSKI
Politechnika Śląska, Instytut Informatyki

ANALIZA MECHANIZMU DYNAMICZNEGO PRZYDZIAŁU ZASOBÓW SYSTEMU AIX W ZAKRESIE JEGO ZASTOSOWANIA DO OKREŚLENIA STOPNIA WYKORZYSTANIA SERWERA W ŚRODOWISKU BAZ DANYCH ORACLE

Streszczenie. W artykule przedstawione zostały wyniki doświadczenia wykorzystania narzędzia *workload manager* systemu AIX 5L do określenia stopnia wykorzystania serwera przez poszczególne instancje bazy danych Oracle. Celem przeprowadzonego eksperymentu było oszacowanie chwilowego wykorzystania serwera przez instancje bazy danych oraz wskazanie, która z nich wykorzystuje najwięcej zasobów. Artykuł wskazuje na możliwość wykorzystania systemowych narzędzi dynamicznego przydziału zasobów do określania stopnia wykorzystania serwera oraz rozliczania kosztów jego eksploatacji.

Słowa kluczowe: system operacyjny, dynamiczny przydział zasobów, baza danych

ANALYSIS OF USING AN AIX DYNAMIC RESOURCE ALLOCATION MECHANISM TO DESCRIBE A UTILITY LEVEL OF SERVER IN ORACLE DATABASES ENVIRONMENT

Summary. The article shows results of experiment that describes the possibility of uses workload manager in AIX5L operating system to describe utility level of server in Oracle databases environments. Goal of the experiment was to show server utilization by database instances. The article shows that it is possible to use dynamic resource allocation mechanism to present a scale and costs of server utilizations.

Keywords: operating system, dynamic resource allocation, database

1. Wprowadzenie

Pierwsze udane konstrukcje elektronicznych maszyn cyfrowych wykorzystywanych do realizacji procesów obliczeniowych dały początek powstaniu i rozwojowi informatyki jako dyscypliny naukowej [9]. Od niemalże początku powstania komputerów prowadzone są badania nad mechanizmami dynamicznego przydziału zasobów informatycznych. Mechanizmy te opierają się na dobrze udokumentowanej teorii, głównie z dziedziny badań operacyjnych takich, jak: teoria kolejek, teoria zapasów, proces Markowa, analiza szeregów czasowych, metody gradientowe. Pomimo dobrych podstaw teoretycznych, ze względu na szybkie zmiany w systemach informatycznych, jest jeszcze w tej dziedzinie dużo do zrobienia. W ostatnich latach zapotrzebowanie na moc obliczeniową serwerów zaczyna przerażać ich możliwości. Duże ilości procesorów i pamięci powodują, że zasoby informatyczne są bardzo kosztowne, sięgające niejednokrotnie kilkunastu milionów dolarów. Prowadzone są w związku z tym prace badawcze, mające na celu możliwie jak najlepsze wykorzystanie zasobów systemowych. Ważniejsze kierunki badań obecnie prowadzone to *grid* oraz *autonomic computing*. Prace badawcze w dziedzinie *autonomic computing* zmierzają do opracowania metod zarządzania i serwisu systemów komputerowych, adekwatnych do bardzo szybkiego rozwoju technologii informatycznych [13]. *Grid computing* jest natomiast, w ogólnym rozumieniu, metodą gospodarki rozproszonymi zasobami informatycznymi [12].

Potrzeby rozwoju mechanizmów dynamicznego przydziału zasobów podyktowane są szybkim rozwojem technologii komputerowych i systemów informacyjnych. Prowadzone są obecnie intensywne badania nad sieciami militarnymi i trzeciej generacji, nanotechnicznymi i kwantowymi systemami informatyki [9, 14, 15], co będzie miało wpływ na potrzeby podziału mocy obliczeniowej we współczesnych systemach komputerowych.

Nowe wersje systemów operacyjnych mają wbudowane mechanizmy, mające na celu lepsze wykorzystanie zasobów obliczeniowych serwera. Są to np.: *workload manager* i *load leveler* w AIX5L, *enterprise workload manager* oraz *intelligent resource director* (IRD) w systemie z/OS [6, 8]. Użycie wymienionych mechanizmów wymaga ich uprzedniej parametryzacji, która jest możliwa jedynie po szczegółowej analizie wymagań oraz warunków środowiskowych. Niektóre z wymienionych mechanizmów dynamicznego przydziału zasobów można wykorzystać do rozliczania wykorzystania zasobów informatycznych serwera [7]. Niniejszy rozdział przedstawia analizę wybranych mechanizmów w zakresie ich przystosowania do szacowania wykorzystania serwerów oraz aspekty ich wykorzystania do obniżenia kosztów eksploatacji systemów informatycznych.

2. Środowisko do prowadzonych eksperymentów

Wynikiem przeprowadzonego eksperymentu miało być oszacowanie stopnia wykorzystania maszyny przez poszczególne instancje baz danych. W tym celu zdefiniowano 5 klas w konfiguracji *workload manager*. Każda z nich odpowiada jednej instancji bazy danych. Ponieważ zamiarem było rozliczenie wykorzystania zasobów serwera, nie dokonano ich szczegółowej parametryzacji. Należy pamiętać, że w przypadku uruchomienia *workload manager* w trybie pasywnym system nie narzuca reguł przydziału zasobów. Kolejnym etapem pracy było znalezienie odpowiednich reguł klasyfikowania procesów (*ang. class assignment rules*). Narzędzie *workload manager* pozwala na budowanie tych reguł według:

- nazwy użytkownika, który uruchomił wskazany proces,
- grupy podstawowej użytkownika, który uruchomił proces,
- nazwy procesu poprzez podanie pełnej ścieżki dostępu.

Żadna z nich w rozważanym przypadku nie mogła być zastosowana, ponieważ wszystkimi instancjami baz danych zarządzał jeden użytkownik, a także wszystkie instancje były uruchamiane przez jeden i ten sam program. W związku z tym reguły klasyfikowania procesów do poszczególnych klas zostały oparte na manualnym mechanizmie, bazującym na nazwie procesu systemowego. Reguła działała w ten sposób, że proces o nazwie oraclePROD (LOCAL=NO) ze względu na to, że w jego nazwie występuje ciąg znaków „PROD”, był kwalifikowany do klasy PROD.

2.1. Opis środowiska

Do przeprowadzenia eksperymentów wybrano partycję logiczną na serwerze IBM p670 o następującej konfiguracji:

- 7 procesorów POWER4+ 1.7 GHz,
- 9,2 GB pamięci operacyjnej,
- system operacyjny AIX 5.2ML04,
- baza danych Oracle 9.2.

Na wydzielonej partycji logicznej zainstalowano 5 różnych instancji bazy danych Oracle. Średnie obciążenie procesorów na partycji często dochodziło do 100%. W systemie pracowało ponad 500 użytkowników.

3. Prezentacja wyników doświadczenia

Po zdefiniowaniu reguł i uruchomieniu *workload manager* w trybie pasywnym otrzymano następujące wyniki:

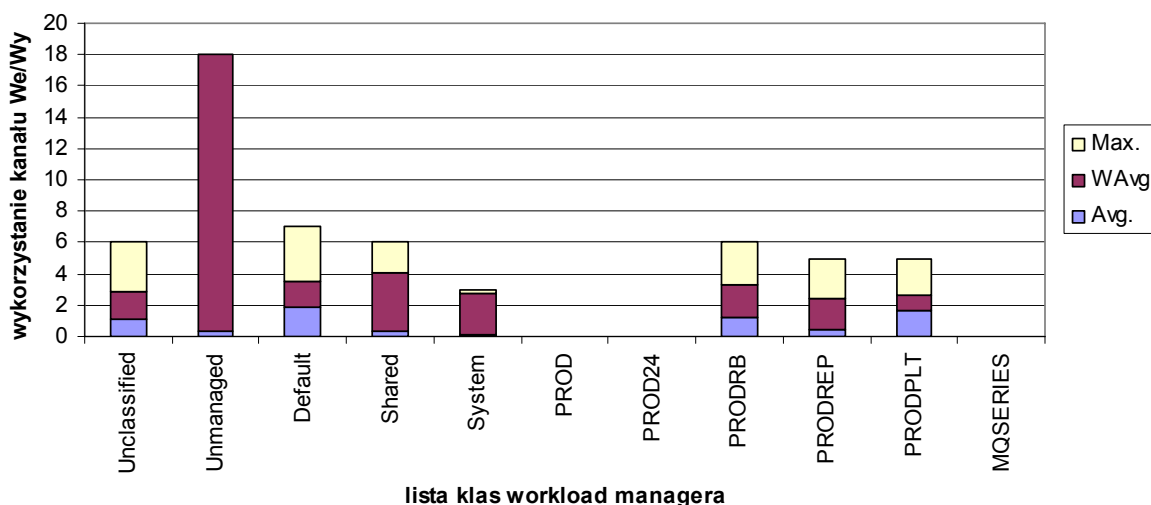
```

mon64 v9a                               Hostname=hqrs6h07 Refresh=2.0secs 11:26.16
Work Load Manager CPU MEM BIO CPU MEM IO CPU MEM BIO Tier Inheritance
Class Name |---Used---| |---Desired---| |---Shares---| Proc's T I Localshm
Unclassified 0% 7% 5% 100 98 100 -1 -1 -1 0 0 0 0
Unmanaged 0% 12% 0% 100 99 100 -1 -1 -1 0 0 0 0
Default 14% 3% 2% 100 98 100 -1 -1 -1 77 0 0 0
Shared 0% 23% 0% 100 98 100 -1 -1 -1 0 0 0 0
System 0% 2% 0% 100 99 100 -1 -1 -1 120 0 0 0
PROD 4% 1% 3% 100 98 100 -1 -1 -1 47 0 0 0
PROD24 0% 2% 0% 100 98 100 -1 -1 -1 99 0 0 0
PRODRB 19% 34% 3% 100 98 100 -1 -1 -1 109 0 0 0
PRODREP 57% 7% 0% 100 98 100 -1 -1 -1 59 0 0 0
PRODPLT 1% 3% 0% 100 98 100 -1 -1 -1 31 0 0 0
MQSERIES 0% 0% 0% 100 100 100 -1 -1 -1 11 0 0 0

```

Z powyższego przykładu widać, że klasa PRODREP wykorzystywała najwięcej (57%) mocy obliczeniowej serwera. Najwięcej pamięci (34%) wykorzystywała klasa PRODRB.

Poniżej przedstawione zostały dane historyczne prezentowanych klas. Statystyki zebrano za pomocą narzędzia *nmon*. Wykres przedstawia wykorzystanie podsystemu wejścia i wyjścia. Na osi y widoczne jest procentowe wykorzystanie kanału dyskowego.



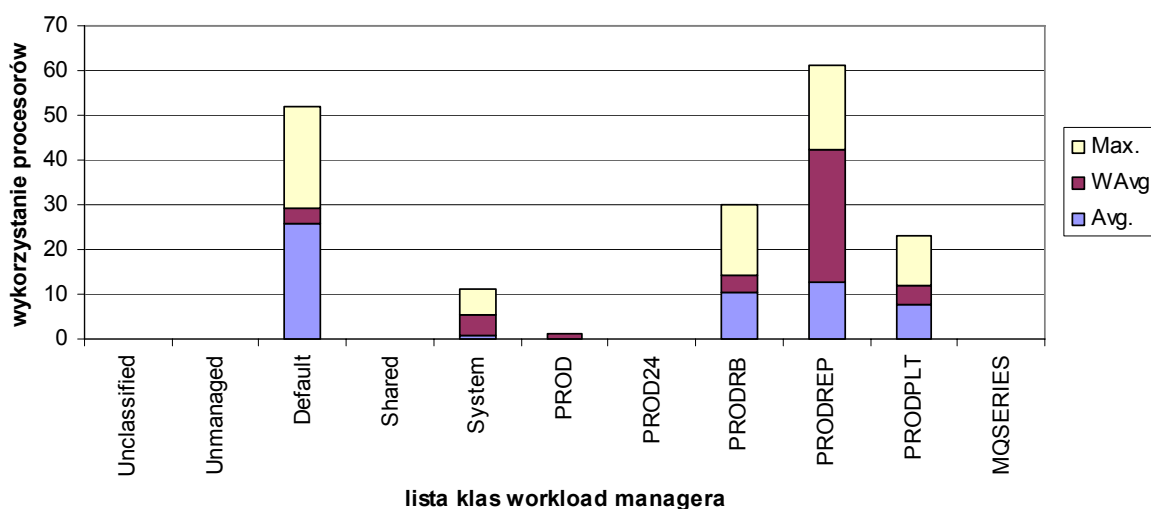
Rys. 1. Wykorzystanie kanału We/Wy

Fig. 1. Utilization of I/O channel

Na wykresie widać, że kanały dyskowe były najbardziej wykorzystywane przez klasy PRODRB, PRODREP i PRODPLT. Najmniejsze ich wykorzystanie jest widoczne dla klas PROD i PROD24. Warto zwrócić uwagę na klasy systemowe typu *unmanaged*, *unclassified* oraz *shared*. Klasy te reprezentują zasoby, którymi z różnych względów nie może zarządzać *workload manager*. Brak możliwości zarządzania występuje dla procesów systemowych, m.in. wymiany stron pamięci do i z przestrzeni wymiany (*ang. paging space*). Tak

wykorzystane zasoby są zliczane w klasie *unclassified*. Mimo że takie wykorzystanie podsystemu dyskowego nie jest zaliczane do poszczególnych klas zasobów, należy pamiętać, że są one przez nie pośrednio generowane. Klasa *unmanaged* jest także poza kontrolą *workload managera*. Należą do niej procesy typu *wait*. Nie należy w związku z tym zwracać na nią szczególnej uwagi. Stopień wykorzystania maszyny przez wszystkie procesy łącznie z procesami *wait* zawsze jest równy 100%.

Następny wykres przedstawia procentowe wykorzystanie procesorów przez zdefiniowane klasy *workload managera*.

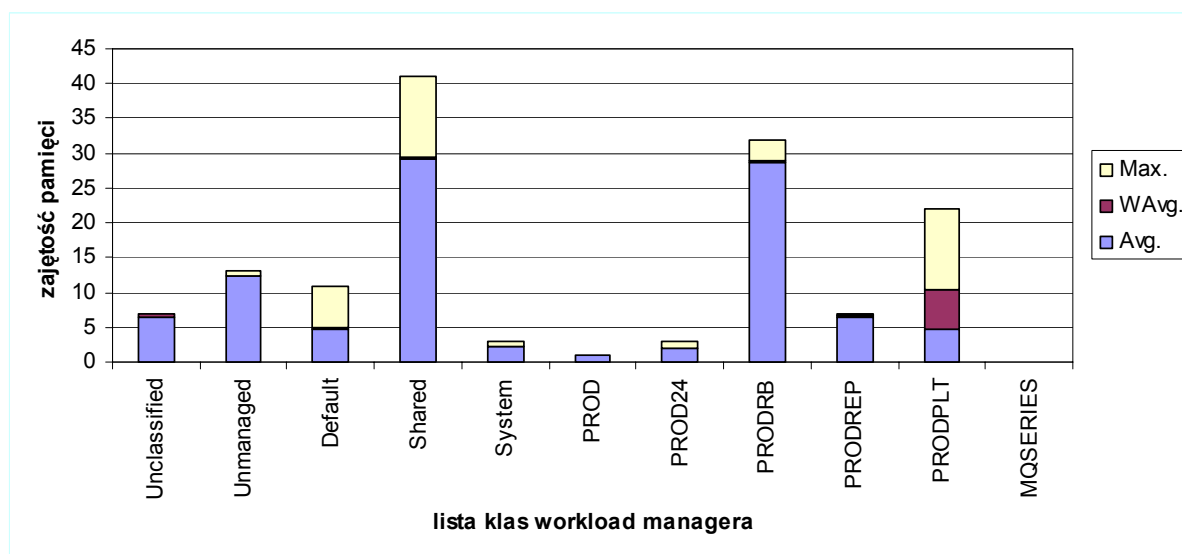


Rys. 2. Procentowe wykorzystanie procesorów

Fig. 2. Utilization of processors

Z wykresu wynika, że procesory serwera były najbardziej wykorzystywane przez klasę PRODREP. W klasie *default* znalazły się natomiast procesy nie zaliczone do żadnej z nowo zdefiniowanych klas. Klasa *default* jest predefiniowaną klasą domyślną. Jeśli procesy nie zostaną zaklasyfikowane przez reguły *workload managera* do żadnej ze zdefiniowanych przez administratora klas, system umieszcza je w tej właśnie klasie. Z analizy procesów znajdujących się w klasie domyślnej wynika, że dość duża część mocy obliczeniowej jest zużywana przez proces nasłuchu bazy danych oraz częściowo przez skrypty służące do obsługi procesów administracyjnych.

Kolejny wykres przedstawia procentowe wykorzystanie pamięci operacyjnej przez zdefiniowane klasy *workload managera*.



Rys. 3. Procentowe wykorzystanie pamięci operacyjnej

Fig. 3. Utilization of memory

Najwięcej pamięci jest wykorzystanej przez procesy w klasie PRODRB. Klasa *shared* należy do tzw. *pseudoklas*, które są poza kontrolą *workload managera*. Strony pamięci typu *shared* to strony wykorzystywane przez więcej niż jedną z klas. Są to strony pamięci współdzielonej (*ang. shared memory region*) oraz pliki wykorzystywane równocześnie przez procesy należące do wielu różnych klas.

3.1. Rezultaty analizy w zakresie dynamicznego przydziału zasobów

Z analizy działania mechanizmu dynamicznego przydziału zasobów wynika, że motor bazy danych Oracle wykorzystuje wspólne zasoby, które nie mogą zostać przypisane do poszczególnych instancji.

Są to procesy:

- podsystemu nasłuchu motoru bazy danych (*ang. listener*),
- skrypty służące do obsługi procesów administracyjnych, takich jak m.in. ładowanie danych.

W związku z tym podczas rozliczania kosztu wykorzystania serwera należy także brać pod uwagę zasoby wspólne, które ze względów technicznych nie mogą być zaklasyfikowane do odpowiednich klas.

Podczas testów zwrócono uwagę na fakt, że nowe procesy nie są przypisywane do klas *workload manager*. Skutkuje to faktem, że część zasobów zostaje zliczona przez klasę *default*.

W celu przypisania zasobów do odpowiednich klas został napisany poniższy program:

```
ps -ae -o pid,user,args | grep -v defunct | while read line
do
PNAME=`echo $line | awk '{ print $3}'`
PID=`echo $line | awk '{ print $1}'`
case $PNAME in
*PROD)
    wlmassign -S PROD $PID
;;
*PROD24)
    wlmassign -S PROD24 $PID
;;
*PRODRB)
    wlmassign -S PRODRB $PID
;;
*PRODREP)
    wlmassign -S PRODREP $PID
;;
*PRODPLT)
    wlmassign -S PRODPLT $PID
;;
esac
done
```

Podczas testu program był uruchamiany co 15 minut. W przypadku konieczności uruchamiania systemu w większym środowisku, należy przypisywać procesy do poszczególnych klas już na etapie ich tworzenia. W systemie AIX jest to możliwe poprzez wykorzystanie RSCT (*ang. Reliable Scalable Cluster Technology*).

4. Podsumowanie

W niniejszym artykule przedstawiona została metoda rozliczania zasobów systemowych serwera poprzez wykorzystanie *workload manager*. Użycie takiego narzędzia powoduje, że trzeba uwzględnić późniejszą konieczność rozliczania zasobów już na etapie projektowania systemów komputerowych.

Przedstawione wyniki eksperymentów pokazały, że nie wszystkie wykorzystane zasoby serwera są zliczane. Dotyczy to zasobów wspólnych, takich jak: pamięci typu *pin*, stron pamięci używanych przez wiele klas jednocześnie i procesu stronicowania pamięci. Zaprezentowany przykład dla baz danych Oracle wskazuje, że duża część zasobów serwera była wykorzystywana przez wspólny proces nasłuchu.

Pomimo tego, że rozliczenia są obciążone dość dużymi błędami i jest wiele aspektów, na które trzeba zwrócić uwagę, a także pamiętać o nich już na etapie projektowania systemu, możliwe jest wykorzystanie systemowych narzędzi dynamicznego przydziału zasobów do określania stopnia wykorzystania serwera oraz rozliczania kosztów jego eksploatacji.

Nowo tworzone procesy motoru bazy danych Oracle zwykle nie są przypisywane prawidłowo do zdefiniowanych klas. Istnieje zatem konieczność poprawy mechanizmu klasyfikowania procesów (*ang. class assignment rules*).

Dzięki wykonanym eksperymentom wykazano, że metody klasyfikowania procesów do poszczególnych klas można usprawnić poprzez:

- wykorzystanie różnych użytkowników do uruchomienia instancji baz danych Oracle,
- wykorzystanie różnych grup użytkowników do uruchomienia instancji baz danych Oracle.

Biorąc pod uwagę wymagania obecnie wprowadzanych technologii informatycznych, konieczne będzie stosowanie sprawnych mechanizmów dynamicznego przydziału zasobów.

LITERATURA

1. Castro S., Tezulas N., Yu B., Berg J., Kin H., Gfroerer D.: AIX 5L Workload Manager (WLM). ITSO, Austin 2001.
2. Clitherow D., Herzog S., Salla A., Sokal V., Trethewey J.: OS/390 Workload Manager Implementation and Exploitation. ITSO, Austin 1999.
3. Cederlof T., Klark A., Herlin T., Ostaszewski T.: IBM Certification Study Guide AIX Performance and System Tuning. ITSO, Austin 2000.
4. Dasgupta T., Sommer S.: IBM eServer Certification Study Guide - AIX 5L Performance and System Tuning. ITSO, Austin 2002.
5. Rozendal K., Albot A., Alford J., Beesley J., Freudenberger A., Speed M.: AIX V4.3.3 Workload Manager Technical Reference. ITSO, Austin 2000.
6. Bari P., Dillenberger D., Morrill H.: Enterprise Workload manager Overview. ITSO Austin 2005.
7. Darmawan D., Kamers C., Pienaar H., Shiu J.: AIX 5L Performance Tools Handbook. ITSO, Austin 2003.
8. Castro S., Tezulas N., Yu B., Berg J., Kin H., Gfroerer D.: AIX 5L Workload Manager (WLM). ITSO, Austin 2001.
9. Węgrzyn S.: O informatyce jako dziedzinie nauki o ruchu i przetwarzaniu informacji. ProDialog, nr 16, Poznań 2003.

10. Drozdowski M., Wolniewicz P.: Szeregowanie zadań jednorodnych w gronach stacji roboczych. ProDialog, nr 10, Poznań 2000.
11. Małyśiak B.: Porównanie czasów wykonania zapytań rozmytych z ich nierozmytymi odpowiednikami. Współczesne problemy systemów czasu rzeczywistego. WNT, Warszawa 2004.
12. Węglarz J.: Zarządzanie zasobami w systemie typu grid. ProDialog, nr 16, Poznań 2003.
13. Autonomic Computing: <http://www.research.ibm.com/autonomic/>.
14. Domańska J., Domański A., Czachórski T.: Self-similarity of network traffic and the performances of QoS mechanisms. Archiwum Informatyki Teoretycznej i Stosowanej, Katowice 2004.
15. Czachórski T.: Analityczne modele kolejkowe w ocenie efektywności pracy systemów i sieci komputerowych. ProDialog, Poznań 2003.

Recenzent: Dr inż. Marcin Gorawski

Wpłynęło do Redakcji 21 lipca 2005 r.

Abstract

This article shows an analysis of usefulness dynamic resource allocations to account a cost of server utilizations. The article demonstrates example of using workload manager in oracle multi instances environment. Figure 1, 2 and 3 describe results of the experiments. The figures show that not all processes were classified to defined classes. Author explains why the processes were not classified and how readers should interpret the predefined classes: default, unmanaged and unclassified. We can also to find how the dynamic allocation mechanism works. In the end the author gives some recommendations how we can improve the dynamic allocation mechanism.

Adres

Maciej MŁYŃSKI: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16, 44-100 Gliwice, Polska.