

Maciej MŁYŃSKI
Politechnika Śląska, Instytut Informatyki

ANALIZA ZASTOSOWANIA MASZYNY WIRTUALIZACYJNEJ DO ZWIĘKSZENIA EFEKTYWNOŚCI SYSTEMÓW KOMPUTEROWYCH

Streszczenie. Atrykuł przedstawia analizę zastosowania narzędzia *Virtualization Engine* management collection w wersji 2.1 do zwiększenia efektywności systemów komputerowych w zastosowaniach czasu rzeczywistego o miękkich wymaganiach czasowych. *Virtualization Engine* jest zintegrowanym oprogramowaniem, które składa się z modułów *IBM Director*, *Enterprise Workload Manager* oraz *Virtualization Engine Console*. Wszystkie moduły mają na celu zwiększenie efektywności systemów komputerowych. W pracy omówione zostały poszczególne moduły oraz koncepcje ujednoczenia zarządzania infrastrukturą informatyczną, takie jak: serwer CIM (ang. *Common Information Model*), architektura zorientowana na usługi oraz standard ARM (ang. *Application Response Measurement*).

Słowa kluczowe: Maszyna wirtualizacyjna, Resource Dependency Service, IBM Director, Enterprise Workload Manager, zarządzanie infrastrukturą informatyczną, ARM, SOA

ANALYSIS OF USEFULNESS OF VIRTUALIZATION ENGINE TO IMPROVE EFFICIENCY OF COMPUTER SYSTEMS

Summary. Article has shown analysis of usefulness of *Virtualization Engine* Version 2.1 for improvement of efficiency of computer systems. The *Virtualization Engine* is a complex utility composed of *IBM Direction*, *Enterprise Workload Manager* and *Virtualization Engine Console*. All of the modules are dedicated for efficiency of management of computer systems. This article has shown also all modules and concepts of unity of managements of system infrastructure such as: CIM server, SOA architecture, ARM (*Application Response Measurement*) standard.

Keywords: *Virtualization Engine*, Resource Dependency Service, IBM Director, zarządzanie infrastrukturą informatyczną, Enterprise Workload Manager, ARM, SOA

1. Wstęp

W latach dziewięćdziesiątych powstało wiele narzędzi, które pełnią rolę swego rodzaju nakładek na infrastrukturę informatyczną, są to narzędzia takie, jak: Tivoli Enterprise Console [22], BMC Patrol [4] oraz narzędzia do pomiaru ich wydajności takie, jak: Rational Robo [23, 27] i Mercuri TestDirector. Narzędzia te mają za zadanie monitorować pracę systemu oraz wysyłać powiadomienia do użytkowników w przypadku, gdy działa on niezgodnie z przyjętymi przez projektanta założeniami. Ponieważ przełom nowego tysiąclecia przyniósł ze sobą trend przechodzenia z zarządzania zasobami do zarządzania usługami, dużą popularnością zaczęły się cieszyć systemy i narzędzia, umożliwiające realizację takiego podejścia [1, 3, 14, 16]. Lepsze zarządzanie efektywnością usług możliwe jest dzięki wprowadzeniu mechanizmu wirtualizacji. Zagadnieniem wirtualizacji interesuje się również dość intensywnie środowisko naukowe, dlatego w 2005 roku ACM wspólnie z USENIX zorganizowano konferencję naukową, na której opublikowano wiele prac odnoszących się do systemów z ograniczeniami czasowymi. Dotyczą one zmian w jądrze systemów operacyjnych, analizy różnego typu maszyn wirtualizacyjnych, a nawet programowania systemów czasu rzeczywistego [5, 7, 15, 29]. Przełom w zastosowaniach maszyn wirtualizacyjnych przyniosło wprowadzenie, w 2005 roku, mechanizmów wirtualizacji w systemach komputerowych [20]. Te interesujące rozwiązania zostały zaimplementowane w środowisku sprzętowym i programowym, tworzonymi przez wieloprocessorowe systemy IBM pSeries p5. Jedną z najciekawszych możliwości tych systemów jest dynamiczne tworzenie logicznych partycji. W roku 2006 ukazało się narzędzie *Virtualization Engine*, łączące możliwości sprzętowe systemów p5 ze środowiskiem zorientowanym na usługi [1, 14]. Dlatego jest to narzędzie, które zapowiada szybki rozwój tej gałęzi badań i rozwiązań. Ponieważ systemy komputerowe są obecnie bardzo złożone, dlatego zarządzanie nimi jest coraz trudniejsze [2, 9, 16]. Poszukiwania nowych metod efektywnego zarządzania realizowane są w obszarach programowych i sprzętowych [20, 21]. Niniejszy artykuł przedstawia szereg aspektów rozwiązania, jakim jest maszyna wirtualizacyjna oraz jego analizę pod kątem zastosowania w systemach przetwarzania informacji w czasie rzeczywistym.

Oprogramowanie *Virtualization Engine management collection* wersja 2.1 składa się z czterech modułów: *Enterprise Workload Manager*, *IBM Director*, *Resource Dependency Service* oraz *Virtual Engine Console* [14, 16]. *Enterprise Workload Manager* jest narzędziem, służącym do zarządzania mocą obliczeniową serwerów. Istotne z punktu widzenia użytkownika końcowego jest to, że nie jest ono zorientowane na zasoby tak jak *Workload Manager* w systemie AIX 5L [17, 18, 19], ale na cele biznesowe [3, 14]. Celem dla *Enterprise Workload Managera* jest osiągnięcie zdefiniowanej wcześniej odpowiedniej wydajności systemu informatycznego.

Enterprise Workload Manager może mierzyć wydajność wszystkich elementów systemu, zarówno systemu operacyjnego, serwerów, baz danych, jak i aplikacji. Właściwy pomiar wydajności uwarunkowany jest tym, aby aplikacja miała możliwość obsługi instrumentów ARM. Dlatego komercyjne systemy operacyjne i bazy danych umożliwiają zazwyczaj pomiar czasu odpowiedzi dla tych instrumentów. Pomiar wydajności aplikacji dla instrumentów ARM jest możliwy również dla niektórych serwerów aplikacyjnych [1, 14]. Takie możliwości daje na przykład serwer aplikacyjny WebSphere firmy IBM. Nowe aplikacje zwykle nie mają takich możliwości, dlatego podczas ich tworzenia należy myśleć o późniejszej konieczności zastosowania ich w środowisku *Enterprise Workload Managera*. Interesującą cechą *Enterprise Workload Managera* jest to, że swobodnie potrafi on współpracować z nowoczesnymi technologiami sprzętowej wirtualizacji serwerów p5. W obecnej implementacji pozwala on na zarządzanie mocą obliczeniową tylko w serwerach firmy IBM. Cecha ta bierze się z tego, że obecnie jedynie firma IBM produkuje serwery ze stabilnie funkcjonującymi mechanizmami sprzętowej wirtualizacji.

Drugi moduł to *IBM Director*, przeznaczony do zarządzania kompleksowym środowiskiem informatycznym. Może on monitorować wartości wielu parametrów takich, jak: zajętość systemu plików oraz pamięci operacyjnej, wykorzystanie procesora, ilość zalogowanych użytkowników itd. Tak jak pozostałe podsystemy maszyny wirtualizacyjnej, może on być instalowany na wielu systemach operacyjnych: AIX, Linux (*RedHat Enterprise* i *SLES*), AS/400 (obecnie i/OS) oraz z/OS (dawniej OS/390). Możliwe jest w związku z tym równoległe monitorowanie wielu różnych parametrów systemów. Nie ma także potrzeby, aby administratorzy systemów byli wysokiej klasy specjalistami w zakresie wszystkich systemów operacyjnych. W przypadku problemów sieciowych możemy na przykład sprawdzić, jakie są ustawienia *TTL*, *MTU* czy *ip_forwarding*, bez konieczności logowania się do wszystkich serwerów. Moduł *IBM Director* może służyć nie tylko do sprawdzania czy regulacji ustawień parametrów systemowych, umożliwia on również dystrybucję plików pomiędzy zarządzanymi serwerami, instalację i wgrywanie nowych wersji oprogramowania, a także wykonywanie zadań w określonych, wcześniej zdefiniowanych godzinach.

Trzecim modułem jest RDS (ang. *Resource Dependency Service*). Jest to nowoczesne i nowatorskie narzędzie [14], które opuściło laboratoria firmy IBM w listopadzie 2005 roku. Podobnie jak pozostałe komponenty maszyny wirtualizacyjnej, jest oparte na architekturze zorientowanej na usługach (SOA), która jest nową technologią budowy oprogramowania [1]. Architektura zorientowana na usługi jest dość nowym trendem budowy oprogramowania, który jest silnie wspierany przez technologię J2EE. Moduł *RDS* pozwala na analizę wzajemnych zależności pomiędzy aplikacjami zwanymi także serwisami bądź też usługami. Główną jego funkcjonalnością jest możliwość mapowanie aplikacji, systemów i serwerów do procesów biznesowych [16]. Ponieważ najważniejsza z punktu widzenia funkcjonalności aplikacji

jest realizacja procesów biznesowych, dlatego jest to technologia, która w najbliższym czasie wzbudzać będzie spore zainteresowanie użytkowników.

Czwarty moduł to konsola zarządzająca (ang. *Virtual Engine Console*), która umożliwia zarządzanie wszystkimi komponentami maszyny wirtualizacyjnej z jednego punktu. Ma też wbudowane inne mechanizmy, podnoszące jej użyteczność, takie jak definiowanie ról użytkowników czy delegacja ich uprawnień. Konsola zarządzająca maszyną wirtualizacyjną jest narzędziem łączącym konsolę służącą do zarządzania eWLM oraz RDS.

2. Architektura zorientowana na usługi

Zmieniające się środowisko systemów informatycznych wymusza konieczność tworzenia oprogramowania zorientowanego na usługi. Jednak wymagania co do jakości usług, a także wydajności aplikacji są zmienne w czasie i niedeterministyczne. Architektura SOA podobna jest do obiektów rozproszonych, jednak opisuje rozwiązanie na wyższym poziomie abstrakcji [1]. Interfejsy usług są zazwyczaj definiowane w sposób abstrakcyjny i niezależny od platformy programistycznej. Również same usługi są często implementowane na bazie różnych technologii i udostępniane za pomocą niezależnego protokołu komunikacyjnego [1]. Rezultatem stosowania takich architektur jest zmiana nie tylko z punktu widzenia odbiorcy, ale zazwyczaj także zmiana u dostawcy usług. Jak dotąd większość implementacji architektury zorientowanej na usługi dotyczy aplikacji opartych na serwerach webowych, a to dlatego, że większość interfejsów komunikacyjnych jest oparta na protokole HTTP. Architektura zorientowana na usługi może być realizowana w różnych rozwiązaniach, również w systemach czasu rzeczywistego.

3. Maszyna wirtualizacyjna w środowisku AIX 5L

System AIX 5L jest jednym z najbardziej nowoczesnych systemów operacyjnych typu UNIX. Jest on jednocześnie systemem stabilnym, wydajnym i skalowalnym, który posiada w sobie szereg mechanizmów podwyższających jego dostępność takich, jak: klastry HACMP (ang. *High Availability Cluster Multi-Processing*), mechanizmy szybkiego odtwarzania systemu po awarii takie jak NIM (ang. *Network Installation Manager*) oraz *mksysb*, które umożliwiają wykonywanie kopii systemu i jego późniejsze uruchomienie z taśmy lub płyty DVD. Nowatorskie jest również narzędzie *alt_disk_install*, umożliwiające wykonywanie kopii systemu operacyjnego na zapasowy dysk. Najbardziej interesujące są mechanizmy partycjonowania logicznego i wirtualizacji. Możliwości takie, jak przemieszczanie procesów

rów i pamięci operacyjnej pomiędzy partycjami bez potrzeby restartu systemu operacyjnego powodują, że dość łatwo można za jego pomocą zarządzać wydajnością procesów biznesowych. Natomiast to, że system posiada mechanizmy wirtualizacji i mikropartyjonowania powoduje, że jest on najlepszym obecnie dostępnym na rynku systemem operacyjnym. W tabeli 1 pokazane zostały wybrane parametry partycji logicznej w serwerze IBM p5, którymi można manipulować w maszynie wirtualizacyjnej.

Tabela 1

Parametry maszyny wirtualizacyjnej w zakresie manipulacji jednostką centralną

Grupa	Nazwa parametru	Opis
Tryb pracy procesora	<i>Dedicated</i>	Parametr wskazuje, czy procesory będą pracować w trybie dedykowanym, czy też mogą być współdzielone.
	<i>Shared</i>	
Zajętość procesora	<i>Minimum procesing unit</i>	Parametr wskazują wartości graniczne dla przydzielonej mocy obliczeniowej do partycji logicznej
	<i>Desired procesing unit</i>	
	<i>Maximum procesing unit</i>	
Procesory wirtualne	<i>Minimum virtual processors</i>	Parametry wskazują ile procesorów wirtualnych będzie przydzielonych do wskazanej partycji logicznej.
	<i>Desired virtual processors</i>	
	<i>Maximum virtual processors</i>	
Tryb współdzielenia procesora	<i>Capped</i>	Tryb capped określa, że przydzielona moc obliczeniowa nie może być przydzielona innej partycji. Wright określa wagę partycji w przypadku współdzielenia mocy obliczeniowej procesora pomiędzy różnymi partycjami.
	<i>Uncapped</i>	
	<i>Weight</i>	

Zaleca się, aby w ramach jednej grupy *Enterprise Workload Manager* nie ustawiać dla partycji ich wag. Nie ma przeciw temu technicznych przeciwwskazań, natomiast późniejsza analiza pracy *Enterprise Workload Managera*, w takim przypadku, może się okazać bardzo pracochłonna.

W trybie współdzielonym możemy do partycji przydzielić minimalną wartość 0,1 procesora, natomiast dynamicznie przydzielać można w tym trybie kwanty wykorzystania procesora co 0,01. Tryb *dedicated* pozwala na przydzielanie do partycji pojedynczych procesorów bez możliwości ich współdzielenia. Maksymalna i minimalna liczba procesorów wirtualnych wskazuje, ile system operacyjny utworzy urządzeń typu *CPU*.

Ponieważ AIX 5L wspiera mechanizmy wirtualizacji, instalacja na nim agentów maszyny wirtualizacyjnej jest jak najbardziej odpowiednia. Przed instalacją agenta eWLM należy zainstalować w systemie pakiet *bos.net.ewlm.rte*, a następnie włączyć instrumenty ARM. Zrobić to można, wykorzystując komendę *smitty ewlm*. W przypadku potrzeby zarządzania

partycjami logicznymi, należy w sprzętowej konsoli zarządzającej HMC (ang. *Hardware Management Console*) wpisać do profilu partycji numer grupy dla *Enterprise Workload Managera* oraz ustawić odpowiednie limity dla procesorów. Zalecane jest, aby tryb pracy procesorów był trybem współdzielonym oraz, aby nie była limitowana dla niego moc obliczeniowa.

3.1. Planowanie instalacji

W celu zaplanowania instalacji oprogramowania *Virtualization Engine* zostało dostarczone oprogramowanie wspomagające projektowanie i planowanie instalacji, zwane centrum projektowym (ang. *Planning Advisor*). Zwykle, przy instalacji rozbudowanego oprogramowania, dość dużo czasu zajmuje sprawdzenie kompatybilności wszystkich elementów systemu. Ponieważ baza centrum projektowego jest aktualizowana na bieżąco, daje to możliwość śledzenia możliwości wsparcia zarówno dla systemów konsol zarządzających, jak i systemów agentowych. Centrum projektowe pozwala również na szacowanie mocy obliczeniowej maszyn oraz zajętości pamięci dyskowej w zależności od ilości komponentów oprogramowania i ilości agentów. Jest to dobre i profesjonalne podejście do projektowania systemów oraz inżynierii oprogramowania.

3.2. Proces instalacji komponentów

Skomplikowana architektura rozwiązania powoduje, że proces instalacji komponentów *Virtualization Engine* jest czasochłonny i dość złożony. Na początku należy zainstalować komponenty wspólne, z którymi instalują się również: server aplikacyjny WebSphere, baza danych DB2 oraz ITDS (ang. *Identity Directory Server*), który jest serwerem autoryzacyjnym. Następnie, po zainstalowaniu komponentów wspólnych, należy rozpocząć instalację odpowiednich modułów maszyny wirtualizacyjnej. Cały proces instalacji kończy się instalacją agentów, w nomenklaturze maszyny wirtualizacyjnej są to serwery zarządzane.

4. Wspólny model przekazywania informacji

Przesyłanie informacji pomiędzy różnymi systemami operacyjnymi jest możliwe dzięki zastosowaniu wspólnego modelu informacji CIM, który jest konceptualnym modelem, służącym do opisywania właściwości zarządzania, niepowiązanym z konkretną implementacją [10]. Dzięki wykorzystaniu menedżera CIMMON (ang. *Common Information Model Object Manager*) możliwa jest wymiana informacji między systemami zarządzania a aplikacjami. Server *Pegasus* jest jedną z implementacji menedżera CIMMON w kodzie otwartym,

która jest zgodna ze standardem CIM Distributed Management Task Force (DMTF) i Web-based Enterprise Management (WBEM). Server *Pegasus* może zostać uruchomiony w systemie AIX, Linux, i Microsoft Windows.

5. Pomiar wydajności

Wydajność systemów i aplikacji w środowisku maszyny wirtualnej mierzona jest poprzez wykorzystanie bibliotek ARM. Grupa wywodząca się z Computer Measurement Group (CMG) opublikowała swój standard ARM w wersji 1.0 w czerwcu 1996 roku. Specyfikacja ARM w wersji 2.0 API powstała w listopadzie 1997. W marcu 1998 zostało przekazane do publikacji w Open Group, a następnie zostało zaakceptowane do publikacji jako Open Group Technical Standard [24]. API instrumentów ARM udostępnia wiele funkcji, pozwalających na badanie efektywności systemu operacyjnego i aplikacji. Funkcje te dzielimy potocznie na grupę podstawową i rozszerzoną. Wykorzystanie bibliotek rozszerzonych wiąże się z utratą wydajności spowodowaną zbieraniem informacji.

5.1. Podstawowe możliwości bibliotek ARM

Zaimplementowane funkcje ARM pozwalają odpowiedzieć na pytania:

- Dlaczego aplikacja jest niedostępna? lub, dlaczego nie jest wydajna i gdzie leży problem?
- Kto wykorzystuje te aplikacje, jak często jej używa i jaki rodzaj operacji wykonuje?
- Czy aplikacja pracuje poprawnie i czy jest dostępna?
- Jak aplikacja pracuje? Jaki jest czas odpowiedzi?
- Jaka jest oczekiwana przez użytkowników jakość usług (ang. *service level*)?

Powyższe informacje są przydatne do planowania zasobami (ang. *capacity-planing*) oraz do rozliczania zasobów wykorzystania systemów informatycznych przez użytkowników.

5.2. Zaawansowane możliwości bibliotek ARM

Biblioteki ARM mogą dostarczyć także informacji rozszerzonych, niektóre z nich wymienione są poniżej:

- Korelacja transakcji. Transakcje mają być wykonywane na wielu systemach. Te funkcje pozwalają zrozumieć bardziej naturę aplikacji.
- Metryki pomiaru. Więcej informacji o transakcji, ilość bajtów lub rekordów, które uległy przetworzeniu, lub o stanie aplikacji podczas przetwarzania danego rekordu, np. długość kolejki zadań oczekujących, oraz jak sparametryzować aplikację, aby była bardziej wydajna.

Podczas planowania metryk należy pamiętać, że pobieranie zbyt dużej ilości informacji do analizy zwykle powoduje spadek wydajności całego systemu.

6. Analiza zastosowania maszyny wirtualizacyjnej w systemach specjalizowanych

Ze względu na zastosowane komponenty, produkt zdecydowanie nie znajduje zastosowania w systemach czasu rzeczywistego (tzw. twardych). W przypadku konieczności zwiększenia mocy obliczeniowej wybranej partycji logicznej zasoby zostają zdjęte z pozostałych partycji, a następnie są dodawane do innej. Cały ten proces trwa kilka sekund i jest dłuższy, jeśli partycje są mocno obciążone. Przeprowadzone badania wskazują, że proces ten nie jest deterministyczny, czyli nie możemy określić czasu, w jakim zasoby zostaną przydzielone. W pracy [17] zostały przedstawione wyniki badań, w których do przydziału zasobów został zastosowany *Workload Manager* w systemie AIX, nawet tam nie można mówić o zastosowaniu w systemach o twardych ograniczeniach czasowych. Zastosowanie *Enterprise Workload Managera* powoduje, że czas, w jakim przydzielana jest moc obliczeniowa, jest znacznie dłuższy ze względu na złożoność rozwiązania i wymianę informacji pomiędzy wieloma serwerami i systemami zewnętrznymi. Przeprowadzone testy w środowisku laboratoryjnym wskazują, że ustawienie odpowiednich limitów w politykach systemu nie jest sprawą trywialną. Aplikacje i bazy danych często korzystają ze struktur wspólnych, z tego punktu widzenia istotne są zależności pomiędzy aplikacjami. Przydzielenie zbyt dużych ilości zasobów dla aplikacji może przecież spowodować zmniejszenie jej wydajności. Dzieje się tak często, gdy systemowi operacyjnemu zabraknie mocy obliczeniowej na zarządzanie dostępem do urządzeń wejścia-wyjścia.

Koncepcja rozwiązania maszyny wirtualizacyjnej jest natomiast bardzo dobra dla systemów o miękkich wymaganiach czasowych. Ramy czasowe muszą być jednak tak duże, aby system zdążył zmienić przydział procesorów pomiędzy partycjami. Zależnie od wielkości systemu czasy te wahają się od 1 minuty do 0,5 godziny. Rozwiązanie to może natomiast dobrze sprawdzać się w zastosowaniach biznesowych takich, jak: bank, giełda papierów wartościowych czy systemy internetowe. Wszędzie tam, gdzie ilość transakcji i użytkowników jest niedeterministyczna.

6.1. Korzyści wynikające z wykorzystania maszyny wirtualizacyjnej

Zastosowanie maszyny wirtualizacyjnej może w znacznym stopniu usprawnić zarządzanie w kompleksowym środowisku użytkownika. Można powiedzieć, że, w nowoczesnym środowisku zorientowanym na usługi, zwirtualizowane podejście do zarządzania zasobami

jest nieuniknione. Dzięki niemu możemy zmniejszyć koszty eksploatacji systemów oraz usprawnić procesy planowania zasobów. W celu osiągnięcia zamierzonych korzyści niezbędne jest zdefiniowanie odpowiedniej polityki. Dlatego projektanci systemów powinni zacząć interesować się uważnie możliwościami wirtualizacji w systemach informatycznych oraz systemami służącymi do zarządzania wirtualnym środowiskiem. Nie wszystkie aplikacje mogą w takim środowisku być zaimplementowane, ponieważ, aby skutecznie je monitorować, należy wykorzystywać w nich mechanizmy ARM. Dlatego konieczność stosowania mechanizmów wirtualizacji należy planować już na etapie tworzenia aplikacji. Istotna jest również architektura sprzętowa całego systemu komputerowego. Z pewnością większe korzyści osiągniemy, wykorzystując kompleksowe rozwiązania wirtualizacyjne, zarówno na poziomie sprzętu, systemu operacyjnego, jak i aplikacji.

7. Zakończenie

W dobie szybkich zmian w koncepcjach infrastruktur informatycznych oraz przejścia na gospodarkę usługami, racjonalne jest, aby korzystać z wszystkich dobrodziejstw, jakie oferuje maszyna wirtualizacyjna. Ta nowa koncepcja daje wiele możliwości elastycznego tworzenia środowiska. Jedną z ciekawych jest to, że zasoby obliczeniowe można przydzielać na poziomie sprzętowym i dlatego możliwe jest bardziej efektywne wykorzystanie systemów informatycznych. Zastosowanie maszyny wirtualizacyjnej daje szansę na dynamiczne przydzielanie wielu zasobów informatycznych. W pracy [17] autorzy przedstawili mnóstwo aspektów dynamicznego przydziału zasobów w systemie AIX 5L. Wartości parametrów dla *Workload Managera* były w tej pracy ustawiane ręcznie metodą ad-hoc. W pracy przedstawionych zostało również wiele aspektów wskazujących, że do procesu doboru wartości parametrów warto wykorzystać bardziej wyrafinowane metody, takie jak choćby algorytmy ewolucyjne. Można powiedzieć, że jest to częściowo kontynuacja niekonwencjonalnego rozwiązania zaproponowanego przez autora w pracy [13], w której przedstawiona została metoda skracania mikrocykli w sieciach przemysłowych za pomocą algorytmów ewolucyjnych.

Ponieważ systemy rzeczywiste o twardych ograniczeniach czasowych charakteryzują się tym, że o ich efektywności myśli się już na etapie projektowania, dlatego zastosowanie wyrafinowanych metod doboru parametrów w systemach o łagodnych wymaganiach czasowych może dać jeszcze lepsze rezultaty. Nowe technologie maszyn wirtualizacyjnych udostępniają wiele nowych bibliotek systemowych oraz interfejsów programowania aplikacji, które umożliwiają bardziej elastyczną parametryzację systemu, dlatego możliwe jest opracowanie procedur strojenia systemu poprzez dobór wartości jego parametrów. Autor artykułu

zamierza kontynuować badania w kierunku zbudowania automatu, który będzie przeprowadzał szereg testów i szukał parametrów optymalnych. Prawdopodobnie do tego celu zostaną wykorzystane algorytmy genetyczne [28] lub też inne, bardziej wyrafinowane metody.

LITERATURA

1. Architektura zorientowana na usługach: http://pl.wikipedia.org/wiki/Service-oriented_architecture.
2. Autonomic Computing: <http://www.research.ibm.com/autonomic/>.
3. Bari P., Dillenberger D., Morrill H.: Enterprise Workload manager Overview. IBM ITSO, Austin 2005.
4. BMC Patrol: www.bmc.com/products/products_services_detail/0,,0_0_0_19,00.html
5. Clitherow D., Herzog S., Salla A., Sokal V., Trethewey J.: OS/390 Workload Manager Implementation and Exploitation. Austin, May 1999.
6. Cunei A., Vitek J.: A new approach to real-time checkpointing. Proceedings of the 2nd international conference on Virtual execution environments, 2006.
7. Castro S., Tezulas N., BooSeon Y., Berg J., Kin H., Gfroerer D.: AIX 5L Workload Manager (WLM). Austin, June 2001.
8. Czachórski T.: Analytical Queuing Model for Performance Evaluation of Computer Systems and Computer Networks. ProDialog, no. 16, Poznań 2003.
9. Darmawan D., Kamers C., Pienaar H., Shiu J.: AIX 5L Performance Tools Handbook. Austin, August 2003.
10. IBM Infocenter, <http://publib.boulder.ibm.com/infocenter/eserver/v1r2/index.jsp>.
11. Jastratiew A., Kwiecień A.: Zastosowanie algorytmów ewolucyjnych w poszukiwaniu szeregowanych zbiorów wystąpień zadań cyklicznych. XII Konferencja Systemy Czasu Rzeczywistego, Ustroń 2005.
12. Kirsch C., Sanvido M., Henzinger T.: A programmable microkernel for real-time systems. ACM/Usenix International Conference On Virtual Execution Environments Proceedings of the 1st ACM/USENIX international conference on Virtual execution environments, 2005.
13. Kwiecień A.: Rozprawa habilitacyjna. Analiza przepływu informacji w komputerowych sieciach przemysłowych.
14. Matthys C., Cook J., Kimachi T., Ko E., Perez A. E., Schittko E., Vanel L.: Virtualization Engine Platform Version 2 Technical Presentation Guide. IBM ITSO Poughkeepsie 2006.

15. Menon A., Santos J., Turner Y., Janakiraman G., Zwaenepoel W.: Diagnosing performance overheads in the xen virtual machine environment. Proceedings of the 1st ACM/USENIX international conference on Virtual execution environments, 2005.
16. Mlynski M., Tollet N., Barbati G., Matthys C., Chauhan H., Dierberger B., Marchini R., Wittmann H.: Planning, installing and using Virtualization Engine Version 2, IBM ITSO, Poughkeepsie, w recenzji.
17. Mlynski M., Rumik P.: Badanie użyteczności mechanizmu workload manager systemu AIX 5L do polepszenia wydajności wybranych aplikacji korzystających z bazy MySQL. II Konferencja Bazy Danych Aplikacje i Systemy, WKŁ, 2006.
18. Mlynski M.: Dynamic Resources Allocation in AIX 5L, Real Time Systems – Design and Applications. XII Conference of Real-Time Systems. Ustroń 2005.
19. Mlynski M.: Analysis of Using An AIX Dynamic Resource Allocation Mechanism to Describe a Utility Level of Server in Oracle Data Bases Environment. *Studia Informatica*, vol. 26, no. 3(64), Gliwice 2005.
20. Mlynski M.: The analysis of influence of IBM pSeries servers' virtualization mechanism on dynamic resources allocation in AIX 5L. W recenzji.
21. Open Pegasus: <http://www.openpegasus.org>.
22. Tivoli Enterprise Console: <http://www-306.ibm.com/software/tivoli/products/enterprise-console/>.
23. Rational: <http://www-306.ibm.com/software/rational/>.
24. Standard serwera CIM: <http://www.dmtf.org/standard/cim>.
25. Standard serwera CIM: <http://www.dmtf.org/standards/wbem>.
26. Standard serwera CIM: <http://www.wbemsolutions.com/tutorials/CIM/cim.htm>.
27. Stasiak A., Zieliński Z.: Projektowanie i testownie przykładowych systemów sterowania z wykorzystaniem narzędzi IBM Rational Realtime. XII Konferencja Systemy Czasu Rzeczywistego, Ustroń 2006.
28. Tuning The Kernel With A Genetic Algorithm: <http://kerneltrap.org/node/4493>.
29. Yuting Z., Azer B., Mina G., Ibrahim M., Richard W.: Friendly virtual machines: leveraging a feedback-control model for application adaptation. Proceedings of the 1st ACM/USENIX international conference on Virtual execution environments, 2005.
30. Węgrzyn S.: Informatics as Specific Domain on Motion and Processing of Information. *ProDialog*, no. 16, Poznań 2003.
31. Węglarz J.: Resource Management in Grids. *ProDialog*, no. 16, Poznań 2003.

Recenzent: Dr hab. inż. Zygmunt Mazur Prof. Pol. Wrocławskiej

Wpłynęło do Redakcji 24 lipca 2006 r.

Abstract

The Virtualization Engine is a new brand product from IBM. The latest technology is helpful for management of complex environment. Virtualization engine software includes Enterprise Workload Manager (EWLM), IBM Director, Resource Dependency Service (RDS), and Virtualization Console (VEC). By using those components you can:

- Manage IT infrastructure e.g. monitors operating system parameters, simultaneously install and upgrade software – IBM Director.
- Manage the efficiency and dynamically distribute processing power between applications – EWLM.
- Monitor and analyze dependency between resources – RDS.
- Manage all VE components from one Console – Virtualization Engine Console (VEC).

Computer infrastructure in customer's site must be very flexible. Therefore dynamic resource allocation can help with them. By using these tools you can reallocate and redistribute resources between applications. Some applications needs processing power at night whilst other in the day. Moreover the components have embedded features to manage big data centers. The Virtualization Engine works very well with other IBM hardware virtualization features such as dynamic logical partitions, micro-partitioning, shared and virtual Ethernet. The technologies were introduced into IBM eSeries servers based on latest POWER5 processors. The article also introduces ARM (Application Response Measure). The purpose of the ARM is to enable applications to provide information to measure business transactions from a user perspective and the contributing components of response time in distributed applications. This information can be used to support service level agreements and to analyze response time across distributed systems.

Adres

Maciej MŁYŃSKI: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16,
44-101 Gliwice, Polska, maciej.mlynski@acm.org .