

XII Konferencja PLOUG
Zakopane
Październik 2006

Planowanie wydajności w środowisku Oracle

Tomasz Kazimierski

Oracle Polska

e-mail: tomasz.kazimierski@oracle.com

Streszczenie

Angielski termin Capacity Planning jest tłumaczony na wiele sposobów jako „wymiarowanie”, „planowanie mocy obliczeniowej” czy „planowanie wydajności”. To ostatnie tłumaczenie, choć mało dosłowne, wydaje się najlepiej oddawać istotę tej aktywności. Chodzi tu bowiem o ciągły proces dopasowywania zdolności systemu do obsługi żądań pracy przy zmieniających się wymaganiach lub planów zmiany tych wymagań. Istnienie procesu planowania wydajności lub zarządzania wydajnością jest elementem wielu metodyk i ram procedur operacyjnych w tym ITIL. Celem tego opracowania jest uporządkowanie podstawowych pojęć związanych z planowaniem wydajności oraz wskazanie skutecznych technik i metod planowania wydajności dla złożonych systemów informatycznych ze szczególnym uwzględnieniem aktywności związanej z obsługą bazy danych Oracle.

Informacje o autorze

Tomasz Kazimierski jest Architektem Rozwiązań w Oracle Polska; doświadczenie zawodowe obejmuje budowę złożonych systemów IT opartych o technologię Oracle w aspektach analizy wymagań biznesowych i budowy architektur skutecznych rozwiązań oraz zarządzania takimi projektami.

1. Wstęp

Jednym z podstawowych pytań, na które trzeba znaleźć odpowiedź planując nowe przedsięwzięcie informatyczne jest to, jakie parametry sprzętu będą konieczne dla obsługi zadań wdrażanego systemu. Dokonuje się tego przy pomocy wstępnego wymiarowania – ćwiczenie oczywiste dla produktów standardowych (np. systemy ERP) a trudne dla systemów tworzonych na miarę użytkownika lub znaczących modyfikacji funkcjonalności systemów standardowych. Uruchomienie systemu jest weryfikacją wstępnego wymiarowania i początkiem ciągłego procesu planowania wydajności, który trwa tak długo jak żyje system.

Brak planowania wydajności oznaczałoby narażenie na niespodziewane załamanie wydajności a co za tym idzie:

1. Potencjalne straty finansowe firmy
2. Utrata pozytywnego wizerunku zewnętrznego firmy
3. Spadek produktywności
4. Niezadowolenie użytkowników

W tym artykule omówimy podstawowe pojęcia związane z planowaniem wydajności oraz pokażemy główne techniki przydatne do planowania z uwzględnieniem środowiska bazodanowego Oracle. W ostatniej części zostanie podanych kilka praktycznych przykładów zastosowania.

2. Podstawowe pojęcia

Na potrzeby tego artykułu będzie używaliśmy pojęcia zlecenie. Przez zlecenie rozumiemy wykonanie przez system komputerowy funkcji posiadającej określony sens merytoryczny. Funkcjami takimi może być zbudowanie i przesłanie do przeglądarki strony WWW, pobranie z dysków serwera zestawu danych, wykonanie transakcji w bazie danych; ale zleceniami są również obliczenie miesięcznej listy płac w przedsiębiorstwie czy wykonanie gorącej kopii bazy danych.

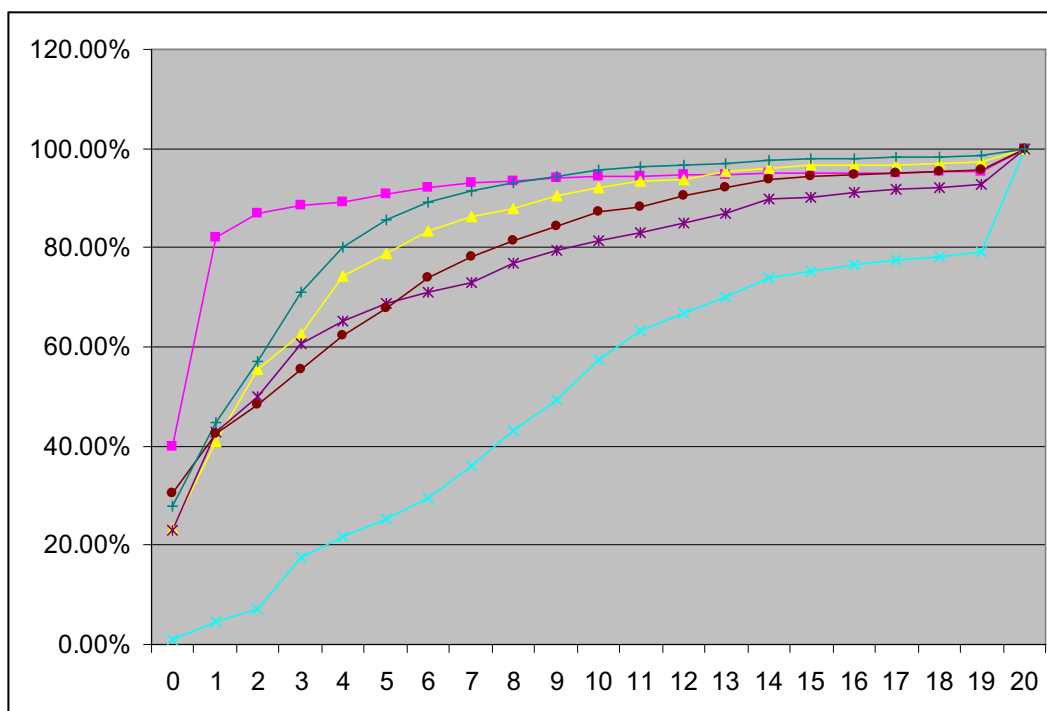
Wydajność systemu jest określona przez dwa podstawowe parametry:

- Przepustowość (throughput) – zdolność systemu do przetwarzania zleceń w jednostce czasu.
- Czas odpowiedzi – czas odpowiedzi systemu dla poszczególnych zleceń.

Ze względu na zdolność systemów do przetwarzania wielu zleceń równocześnie są to dwa niezależne parametry. Na przykład przepustowość systemu może wynosić 200 zleceń / sekundę pomimo, że średni czas wykonania jednego zlecenia wynosi 0.03 sek.

Mówimy, że system jest wydajny, jeśli powyższe parametry mieszczą się w określonym przez wymagania biznesowe limicie. Przykład 1: dla 90% rejestracji przelewu w systemie bankowym czas odpowiedzi < 3 sek. – ponieważ jest to najdłuższy czas możliwy do zaakceptowania przez urzędnika bankowego obsługującego klienta przy okienku; Przykład 2: przepustowość systemu > 30 rejestracji wejścia do zakładu pracy na sekundę ponieważ system zawiera 30-ci czytników kart które pracują równolegle.

Poniższy rysunek pokazuje przykład obserwowanych czasów odpowiedzi w systemie w podziale na typy obciążenia – na osi X przedstawiono czas w sekundach a na osi Y procent zleceń które zostały obsłużone w czasie mniejszym niż wartość na X w ogólnej liczbie zleceń danego typu.

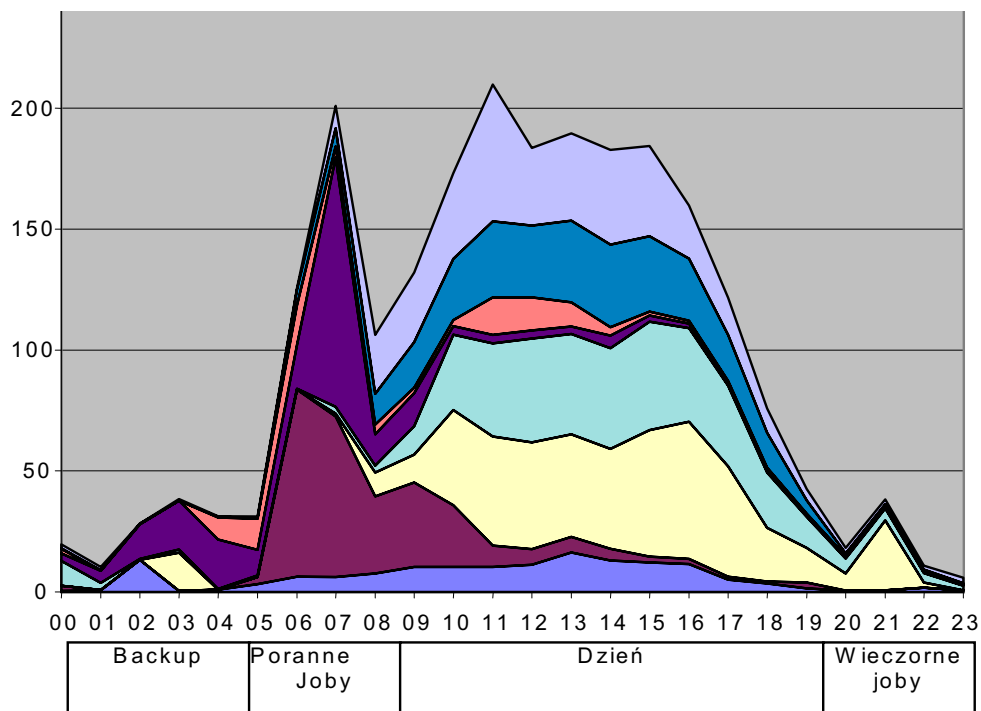


Rys. 1. Przykładowy wykres średnich czasów odpowiedzi w podziale na typy obciążenia.

W każdym systemie istnieje wiele typów zleceń, które różnią się kosztami wykonania – możliwe jest ich pogrupowanie względem charakterystyki wydajnościowej w grupy zwane typami obciążenia (workload). Przykładami grubego podziału na typy obciążenia w środowisku Oracle jest aktywność tylko prostych zapytań, aktywność OLTP i aktywność analityczna. Jednym z najważniejszych elementów koniecznych dla zrozumienia wydajności systemu jest zrozumienie i poprawna identyfikacja wzorców obciążenia. Przy identyfikacji rodzajów obciążeń konieczna jest dość dokładna znajomość rodzajów aplikacji wdrożonych w systemie i ich aktywności.

Obciążenie biznesowe systemu może zostać scharakteryzowane jako liczba zleceń w podziale na typy obciążenia na jednostkę czasu.

W systemie można zaobserwować okna czasowe różniące się obciążeniem. Na przykład okno maksymalnego obciążenia w godzinach 9:00 do 19:00 (duża liczba użytkowników OLTP i sporadyczne inne aktywności) lub okno ciszey nocnej w godzinach 0:00-4:00 (kopia zapasowa, minimalna ilość użytkowników interaktywnych). Przykład dobowej aktywności różnych typów obciążeń przedstawia poniższy wykres:



Rys. 2. Przykładowa zmiana aktywności typów obciążenia w oknach czasowych.

Koszt wykonania zlecenia jest rozumiany jako zapotrzebowanie na zasoby systemowe: CPU, dostęp do pamięci, dostęp do pamięci masowej oraz sieci. Każdy z zasobów posiada specyficzną charakterystykę np. przepustowość strumienia danych, czas reakcji etc. co za tym idzie dla każdego z zasobów istnieje określony punkt nasyceni po osiągnięciu, którego wydajność danego zasoby załamuje się. Konsekwentnie wszystkie zlecenia używające dane zasobu są znacznie wolniej obsługiwane co dalej prowadzi do załamania wydajności całego systemu.

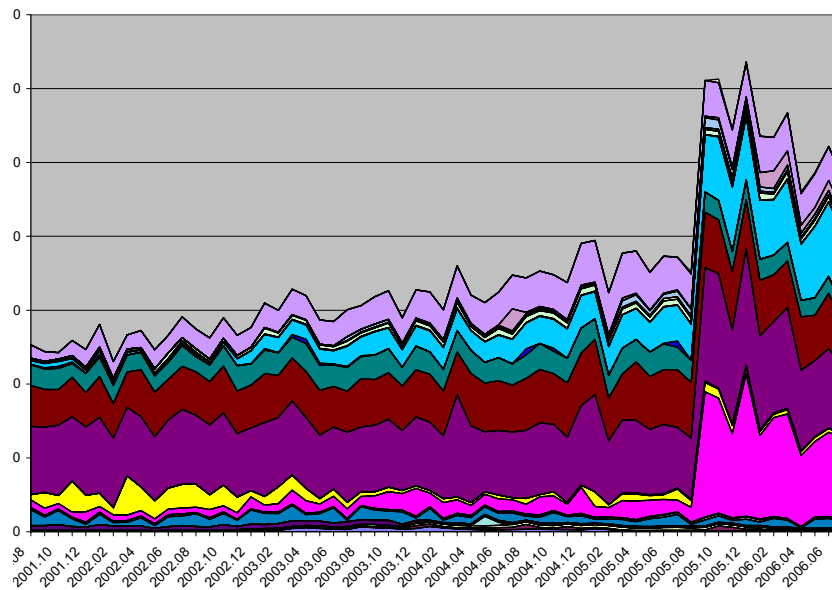
3. Planowanie wydajności

Planowanie wydajności to przewidywanie jak będą się zmieniały parametry wydajności przy zmianie:

- B – obciążenia biznesowego,
- K – zmianie kosztów wykonania zleceń poszczególnych typów obciążenia lub
- Z – przy zmianie zdolności systemu do obsługi zleceń.

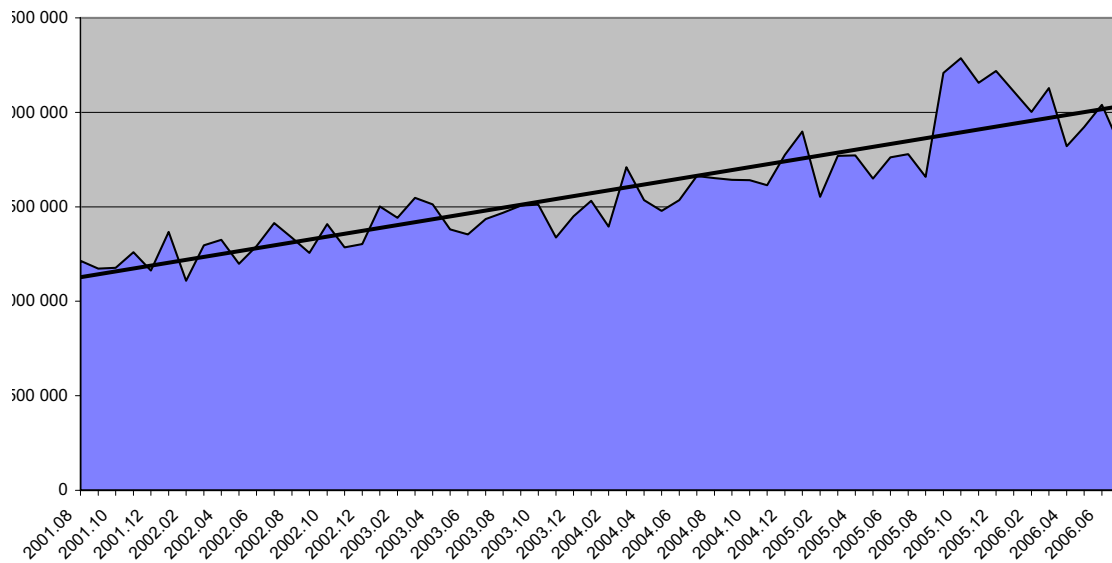
Możemy, więc mówić, że wydajność W jest funkcją w (B, K, Z) .

Zmiana obciążenia B może wynikać ze zmiany wymagań biznesowych, w których działa system, na przykład na wzrost sprzedaży po utworzeniu nowego produktu. Poniższy rysunek przedstawia właśnie taką sytuację obserwowaną jako licznik dokumentów różnych typów powstających w zadanym okresie w systemie:



Rys. 3. Przykładowy wzrost obciążenia systemu na skutek wprowadzenie nowego produktu.

Często obciążenie biznesowe ma stały trend wzrostowy co widać na powiększonej części powyższego wykresu to jest w okresie do wprowadzenie nowego produktu w czerwcu 2005:



Rys. 4. Przykład stałego trendu wzrostowy obciążenia biznesowego.

K – Zmniejszenie kosztów wykonania zlecenia może wynikać z optymalizacji kodu aplikacji bądź strojenia bazy danych lub systemu operacyjnego. Odwrotnie zwiększenie kosztów wykonania zlecenia może nastąpić w czasie życia systemu na skutek wzrostu wolumenu danych lub wprowadzenia nowej funkcjonalności systemów.

Z – Zwiększenie zdolności systemu do obsługi zleceń następuje na skutek rozbudowy sprzętu lub komponentów sprzętu.

4. Ogólna metodyka planowania wydajności

Planowanie wydajności jest procesem składającym się następujących faz:

- Zrozumienie ograniczeń biznesowych, sprzętowych i programowych;
- Identyfikacja typów obciążenia;
- Gromadzenie danych dla analizy;
- Przewidywanie wydajności.

Poniżej opis każdej z faz:

4.1. Zrozumienie

Dla prawidłowego przeprowadzenia planowania wydajności konieczne jest zrozumienie ograniczeń biznesowych, w których działa system. Podstawowymi parametrami koniecznymi do udokumentowania jest akceptowalny poziom usług rozumiany jako limity czasów odpowiedzi i przepustowości systemu. Ponadto konieczne jest wiedza na temat planowanych zmian np. promocji zwiększających sprzedaż (wymagania przepustowości), udostępnienie nowych kanałów dostępu do informacji (zwiększenie ilości obsługiwanych użytkowników), uruchomienie nowych produktów lub usług przed przedsiębiorstwo (dodawanie nowych lub zmienianie istniejących aplikacji). Istotne jest aby użytkownicy biznesowi byli świadomi istnienia procesu planowania wydajności i dostarczali odpowiednie dane z wyprzedzeniem.

W fazie tej dokumentowane jest istniejące środowisko sprzętowe systemu. Zdolność do obsługi zleceń jest zależna od ilości oraz parametrów urządzeń. W przypadku niektórych urządzeń łatwo porównywać ich zdolność do przyjęcia obciążenia, ponieważ jest ona określona przez proste parametry wydajnościowe np. średni czas dostępu do danych dla dysków lub opóźnienie wprowadzane przez urządzenia sieciowe. Pewną trudnością jest porównanie mocy CPU, gdzie moc zależy od częstości zegara oraz złożoności rozkazów wykonywanych przez procesor (architektury RISC¹, CISC², VLIW³) – tu przydatne jest korzystanie z wyników benchmarków np. SPEC. Praktycznie każdy system komputerowy składa się z dużej ilości współpracujących komponentów i wówczas najwygodniejsze jest używać do jednoznacznego określenia zdolności do przyjęcia obciążenia wyników benchmarków porównujących całe system – bardzo użytecznym przykładem są benchmarki publikowane przez **Transaction Processing Performance Council** a przede wszystkim benchmark TPC-C.

Kolejnym elementem, który wymaga zrozumienia to oprogramowanie jak system operacyjny wraz z jego konfiguracją, serwery aplikacyjne i bazodanowe oraz aplikacje biznesowe. Określenie głównych rodzajów aplikacji jest początkiem analizy typów obciążenia, co będzie kontynuowane w fazie kolejnej.

4.2. Identyfikacja typów obciążenia

Właściwa identyfikacja typów obciążenia jest jednym z najtrudniejszych elementów podczas planowania wydajności. Typy obciążenia grupują aktywności o podobnej charakterystyce: częstości występowania, kosztu obsługi, ilości równoczesnych wykonań i itp.

Ogólnie typy obciążeń można podzielić na dwie duże grupy zlecenia interaktywne i zlecenia wsadowe. W przypadku zleceń interaktywnych dodatkowo można wyróżnić zlecenia wykorzystujące pule sesji i sesje bezpośrednie (różnice w zużyciu pamięci i CPU). W ramach każdej grup

¹ **Reduced Instruction Set Computer** np. Ultra Sparc, HP/PA

² **Complex Instruction Set Computer** np. Intel Pentium

³ **Very Long Instruction Word** np. Itanium

istnieje wiele typów obciążeń różniących się częstością wykonań akcji, czasem myślenia użytkownika, czasem wykonania zlecenia, rodzajem przetwarzania danych itp.

Dane o aktywności sesji można poddać analizie automatycznej znajduje tu zastosowanie Oracle Data Mining z algorytmami klastującymi.

Należy dążyć do utworzenia listy kilku dominujących typów obciążeń grupując pozostałe w osobną kategorię. Istnienie zbyt dużej liczby typów obciążeń znacznie utrudnia analizę danych.

Identyfikacja typu obciążenia polega na ustaleniu jego charakterystyki oraz określeniu jak można identyfikować jego aktywność w systemie np. poprzez podanie sposobu, w jaki wpisy w dzienniku pracy aplikacji lub sesje bazodanowe można połączyć z danym typem.

Po określeniu typów obciążenia należy zidentyfikować okna czasowe. Liczba okien czasowych nie powinna przekraczać kilku. Wygodne jest jeśli udało by się zidentyfikować okna czasowe gdzie dominuje dany typ obciążenia – ale nie zawsze jest to możliwe.

4.3. Gromadzenie danych dla analizy

Większość metod przewidywania wydajności wymaga zebrania danych do analizy kosztów poszczególnych typów obciążeń. W zależności od metody gromadzenie następuje w formie chwilowego pomiaru lub długoterminowego monitorowania. Zbierane dane można podzielić na dane niezależne i zależne z punktu widzenia wydajności rozumianej jako funkcji obciążenia.

Dane niezależne to obciążenie. Obciążenie może być mierzone na wiele sposobów, z czego najbardziej skuteczne są:

- Liczba dokumentów powstających w jednostce czasu;
- Liczba akcji wykonywanych przez aplikację na podstawie wpisów w dziennikach działania aplikacji;
- Liczba sesji lub sesji bazodanowych (nie da się zastosować jeśli używane pule sesji);
- Liczba zatwierdzeń wykonywanych przez sesje bazodanowe w jednostce czasu.

Dane zależne to:

- Wyniki chwilowego pomiarów użycia zasobów (liczba odczytów lub zapisów IO, użycie CPU, użycie pamięci, liczba pakietów przesyłanych przez sieć itp.);
- Czasy odpowiedzi dla zleceń na podstawie dzienników pracy aplikacji bądź zewnętrznych pomiarów.

Zbierane dane powinny zawierać znacznik czasowy wykonania pomiaru oraz w środowiskach złożonych z wielu maszyn lub baz danych określenie maszyny bądź instancji bazy danych.

Przykłady użytecznych sposobów zbierania danych podano poniżej w sekcji „Sposoby gromadzenia danych”

4.4. Przewidywanie wydajności

Przewidywanie wydajności opiera się wybraną metodę przewidywania wydajności. Przegląd metod umieszczono poniżej w sekcji „Metody przewidywania wydajności”.

W wyniku fazy uzyskany jest opis wymagań dla systemu przy zmianie obciążenia.

5. Metody przewidywania wydajności

Metody przewidywania wydajności można podzielić na następujące kategorie – podane w kolejności od najmniej kosztownych i skomplikowanych do najdroższych i złożonych lub od najogólniejszych do najbardziej szczegółowych:

- Reguły praktyczne;
- Analiza trendu;
- Modele wydajności;
- Benchmarki.

Poniżej w kolejnych sekcjach opis z każdej z kategorii.

5.1. Reguły praktyczne

Reguły przyjmowane na podstawie ogólnego doświadczenia. Na przykład: „Zajętość CPU przekraczająca 80% istotnie wydłuży czasy odpowiedzi”, „Jeden CPU może obsłużyć 100 równoległych sesji OLTP i do 3 sesji analitycznych”. Wiele z takich reguł zakłada liniową zależność pomiędzy użyciem CPU a obciążeniem np. „Czas wykonania zadania wsadowego rośnie tak jak wolumen przetwarzanych danych”.

Reguły te pozwalają podejmować szybkie decyzje na podstawie ogólnych przesłanek, ale są ich wyniki mają nieokreśloną dokładność.

5.2. Analiza trendu

Przewidywanie przyszłego zachowania systemu na podstawie danych historycznych. Technicznie realizowane jako ekstrapolowanie użycia zasobów o przewidywany wzrost obciążenia. Może znaleźć zastosowanie w systemach słabo obciążonych gdzie koszty użycia zasobów wynikające z konkurencji o nie mogą zostać pominięte.

5.3. Modele wydajności

Możliwe jest skonstruowanie modelu systemu, który będzie wystarczająco dobrze przybliżał zachowanie systemu badanego. Istnieją tu dwa ogólne podejścia: analityczny i symulacyjny.

5.3.1. Analityczne

Model analityczny polega na skonstruowaniu matematycznego opisu obciążenia systemu. Przykładem może być użycie regresji liniowej do określenia utylizacji procesora przy obciążeniu zwiększonym o zadaną wartość i przy znanych wartościach utylizacji przy obciążeniu niższym.

5.3.2. Symulacyjne

Modele symulacyjne polegają na symulacji działania systemu badanego na innych systemach. Funkcje taką może pełnić na przykład środowisko testowe, przy czym wynik jest aproksymowany do środowiska badanego.

5.4. Benchmarki

Wykonanie pomiarów zachowania rzeczywistego systemu w rzeczywistym obciążeniu. Pozwala na jednoznaczne ocenienie zużycia zasobów przez system. Wymaga praktycznego rozwiązania problemu generacji obciążenia analogicznego jak obciążenie rzeczywiste.

6. Sposoby gromadzenia danych

Źródłem danych o obciążeniu mogą być dzienniki pracy aplikacji (np. dziennik pracy serwera WWW), tablice journalowe, dzienniki bezpieczeństwa (łączenie i rozłączanie użytkowników) lub inne dzienniki prowadzone przez aplikacje.

Z punktu widzenia bazy danych informacje o obciążeniu można uzyskać z tabel (G)V\$SESSION i (G)V\$SESSTAT – pierwsza podaje ilość i stan sesji druga ilość transakcji i zatwierdzeń w sekcjach – pomiar polega na okresowym (co kilka minut) tworzenia obrazów tych tabel.

Dane o użyciu zasobów można uzyskać z poziomu systemu operacyjnego. Przykładowe źródła danych w systemie UNIX:

Polecenie	Informacja
sar -u 1 5	Obciążenie CPU
swapinfo -tam	Użycie pamięci w systemie
netstat -i	Ilość pakietów przesłanych i otrzymanych przez sieć
sar -d 1 5	Statystyka wejścia / wyjścia

Tabela 1. Przykładowe polecenia do zbierania statystyk z systemu operacyjnego.

Przy konstrukcji skryptów należy uwzględnić aby używać formatu uśredniającego wynik z czasu od ostatniego pomiaru a nie chwilową wartość – ma to szczególne znaczenie dla parametrów szybkozmiennych np. CPU.

W wersji 10g bazy dane o statystykach systemu operacyjnego można znaleźć w perspektywach bazodanowych np.:

```
select nvl(ss.USERNAME, 'ORACLE PROC') username,
       se.SID,
       VALUE cpu_usage
from v$session ss,
     v$sesstat se,
     v$statname sn
where se.STATISTIC# = sn.STATISTIC#
and NAME like '%CPU used by this session%'
and se.SID = ss.SID
order by VALUE desc
```

Ponadto dane dostępne są w perspektywie V\$OSSTAT.

7. Przykłady

7.1. Przykład #1

W środowisku składającym się z trzech instancji RAC(8,4,4 CPU) uruchomiony jest system obsługujący około 2000 użytkowników. W ciągu roku przedsiębiorstwo planuje wdrażać system w kolejnych jednostkach, co spowoduje wzrost ilości sesji o 30%. Czy obecny system jest w stanie obsłużyć nowe obciążenie? Jeśli nie jak należy go rozbudować.

Zidentyfikowano 4 główne typy obciążenia, z których jeden, OLTP, dominował w godzinach pracy biurowej. Użyto regresji liniowej i na tej podstawie oszacowano, że jeden procesor standardowy w tym badaniu może obsłużyć **183±31** sesji użytkowników oraz pamięć na sesje wynosi **37**

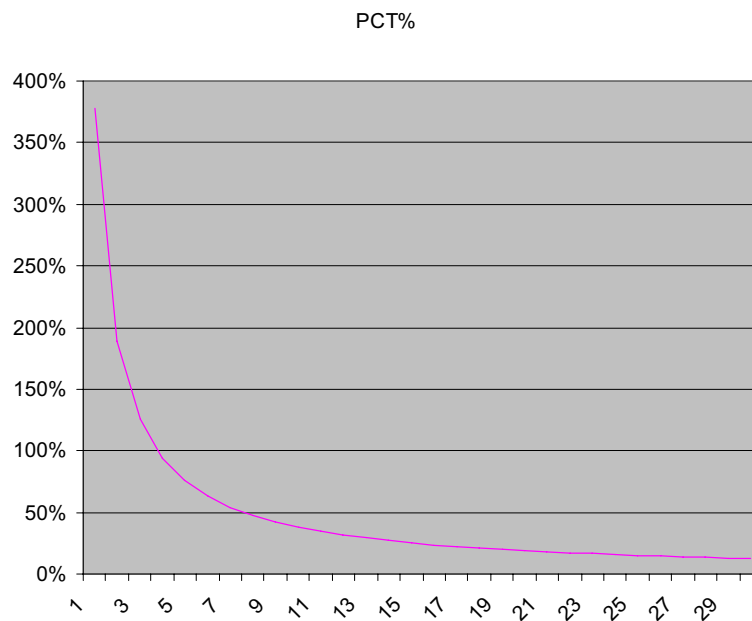
MB ± 0,2MB. Jako wniosek podano, że system może przyjąć zwiększenie obciążenia ze względu na CPU ale ze względu na pamięć sugerowane jest jedno z poniższych:

- Zmiana modelu sesji na pulę połączeń (aby współdzielić pamięć pomiędzy sesjami);
- Rozbudowa fizycznej pamięci maszyn.

7.2. Przykład #2

W systemie istnieją zlecenia o charakterze okresowym dokonujące przeliczenia parametrów biznesowych dla sesji użytkowników. Ze względów biznesowych konieczne jest możliwie częste uruchamianie tego zlecenia. Zmniejszanie częstości odświeżania poniżej 6 minut powoduje, że czasu obsługi innych zleceń przekraczają akceptowalny limit. O ile należy rozbudować maszynę aby częstość wykonania zlecenia można było ustawić na mniej niż 3 minuty.

W ramach prac obserwowano aktywność CPU w kolejnych dniach przy zmianie częstości zlecenia 12, 9 i 6 minut. W godzinach szczytu ilość sesji stabilizowała się na poziomie ok. 260 ale utylizacja procesora zmieniała się w zależności od częstości wykonania zleceń okresowych. Na tej podstawie obliczono przybliżony czas CPU potrzebny na wykonanie zleceń okresowych – czas ten jest niezależny od częstości. Lekceważąc efekt konkurowania o zasoby otrzymujemy wówczas następującą zależność czasu utylizacji CPU od częstości wykonania zlecenia:



Rys. 5. Użycie CPU w zależności od częstości zlecenia.

Wnioskiem z powyższego wykresu jest to, że zmniejszanie częstości wykonania zlecenia powoduje bardzo ostry wzrost wymaganej mocy CPU, co praktycznie uniemożliwia rozwiązanie problemu poprzez rozbudowę sprzętu. Zarekomendowano optymalizację funkcjonalności zlecenia w celu istotnego zmniejszenia wymagań CPU.

7.3. Podsumowanie

Planowanie wydajności jest ważnym elementem w procesie eksploatacji systemów informatycznych. W artykule tym chcieliśmy zwrócić uwagę czytelnika na bezpośrednią zależność planowania wydajności od sfery biznesowej działania systemu w przedsiębiorstwie, z której wynikają

zarówno limity określające postrzeganie systemu jako wydajny jak i zmiany parametrów najbardziej istotnych dla obciążenia.

Zainteresowany czytelnik może znaleźć więcej informacji na temat Planowania Wydajności w pozycjach wymienionych w spisie literatury.

Bibliografia

- [CS95] Craig Shallahamer, 1995, Predicting Computing System Capacity and Throughput, Oracle Corporation White Paper, <http://www.orapub.com>
- [DCEDCS97] Dave Cook, Ellen Dudar and Craig Shallahamer, 1997, The Ratio Modelling Technique, Oracle Corporation White Paper, 1997, <http://www.orapub.com>
- [DMVALD94] Daniel A. Menascé , Virgilio A. F. Almeida, Larry W. Dowdy, Planning and Performance Modeling: from mainframes to client-server systems, Prentice Hall, 1994, ISBN: 0-13-789546-1
- [CS06] Craig Shallahamer, 2006, Essential Performance Forecasting – Presentation RMOUG/IOUG/Collab'06 (March 2006) , <http://www.orapub.com>