# TPC Benchmarking

Marek Wojciechowski, Maciej Zakrzewicz
*Politechnika Poznańska, Instytut Informatyki*
*ul. Piotrowo 3a, 60-965 Poznań*
*{marek,mzakrz}@cs.put.poznan.pl*

Streszczenie

Analiza porównawcza wydajności różnych systemów baz danych wymaga zbudowania modelowej, reprezentatywnej aplikacji testowej, na której efektywności zostanie zdefiniowana miara porównawcza. Opracowywanie standardowych modeli aplikacji testowych dla różnych charakterystyk obciążenia systemu jest celem istnienia założonej w 1988 roku organizacji TPC (Transaction Processing Performance Council). W artykule omówiono najważniejsze benchmarki TPC, powszechnie wykorzystywane przez dostawców systemów baz danych: TPC-C, TPC-H, TPC-R oraz TPC-W.

## 1 Wprowadzenie

Wdrażanie systemu informatycznego wymaga podejmowania ważnych decyzji o wyborze odpowiedniej infrastruktury, w ramach której zostanie on osadzony: platformy sprzętowej, systemu operacyjnego oraz systemu zarządzania bazą danych. Bardzo często oferta rynkowa sprzętu i oprogramowania oraz wykaz konfiguracji przewidzianych przez twórcę systemu pozwalają na wygenerowanie ogromnej liczby dopuszczalnych kombinacji rozwiązań. Dokonanie niewłaściwego wyboru może oznaczać znaczny wzrost kosztów wdrożenia lub też kłopoty z późniejszą eksploatacją.

W praktyce na pierwszy plan wychodzą kryteria bazujące na dwóch podstawowych współczynnikach oceny rozwiązania: *wydajności* i *cenie*. Wydajność systemu wyraża się najczęściej jego *przepustowością* lub *czasem odpowiedzi* i przekłada się wprost na maksymalną liczbę użytkowników końcowych i rozmiar bazy danych. Cena systemu obejmuje zwykle koszt zakupu serwerów, stacji klienckich, koszty budowy sieci oraz koszty licencji wykorzystywanego oprogramowania systemowego. W porównaniach kosztów przyjmuje się zwykle trzy- lub pięcioletnie koszty eksploatacji. Rozwiązaniem idealnym byłoby wybranie takiej konfiguracji, której cena jest minimalna, a wydajność maksymalna. Niestety, prawa rynku powodują, że zwykle wydajność systemu jest proporcjonalna do jego ceny. To powoduje pewne przeformułowanie definiowanego celu. Poszukujemy takich konfiguracji systemu, które oferują zadowalającą wydajność po jak najniższej cenie.

O ile z wyceną wybranej konfiguracji systemu zwykle nie mamy problemów technicznych, to pomiar wydajności jej pracy przysparza pewnych trudności. Pomiar wydajności systemu praktycznie nigdy nie jest realizowany analityczne – zawsze pociąga konieczność przeprowadzenia eksperymentu. Natomiast eksperyment wymaga symulacji jak najbardziej realistycznego obciążenia wybranej konfiguracji systemu. Rozwiązaniem idealnym byłoby budowanie nowego środowiska eksperymentalnego na potrzeby każdego

wdrożenia. Ponieważ jednak wdrożenia bywają do siebie podobne, dlatego w przemyśle informatycznym narodziła się dziedzina badania wydajności systemów przy użyciu standardowych *benchmarków*.

Benchmark to aplikacja sztucznie obciążająca system informatyczny w celu pomiaru jego wydajności. Zwykle benchmarki dzieli się na *uniwersalne* i *aplikacyjne* (*domain-specific*). Benchmarki uniwersalne starają się jednakowo traktować wszystkie elementy systemu. Benchmarki aplikacyjne specyfikują obciążenie systemu modelujące charakterystykę „typowej aplikacji" z wybranej dziedziny zastosowań. Ponieważ wydajność systemu komputerowego może się istotnie różnić dla różnych typów aplikacji, dlatego w praktyce benchmarki aplikacyjne cieszą się większym zainteresowaniem.

Każdy benchmark aplikacyjny powinien spełniać cztery podstawowe kryteria. Po pierwsze, powinien być *adekwatny* do dziedziny zastosowań wskazanej przez jego twórcę. Po drugie, powinien być *przenaszalny*, czyli powinien umożliwiać implementację w różnych systemach i architekturach komputerowych. Benchmark powinien być *skalowalny*, co oznacza, że może być stosowany zarówno w małych systemach, jak i w wielkich. Po czwarte, powinien być *nieskomplikowany*, aby użytkownicy rozumieli i akceptowali przyjęty model obciążenia testowego [Gray93]. W praktyce, benchmarki najczęściej służą porównywaniu rozwiązań według następujących kategorii: (1) różne oprogramowanie i różny sprzęt dla tej samej klasy aplikacji, (2) różne oprogramowanie na tym samym sprzęcie, (3) różne komputery należące do tej samej rodziny oraz (4) różne wydania oprogramowania na tym samym komputerze.

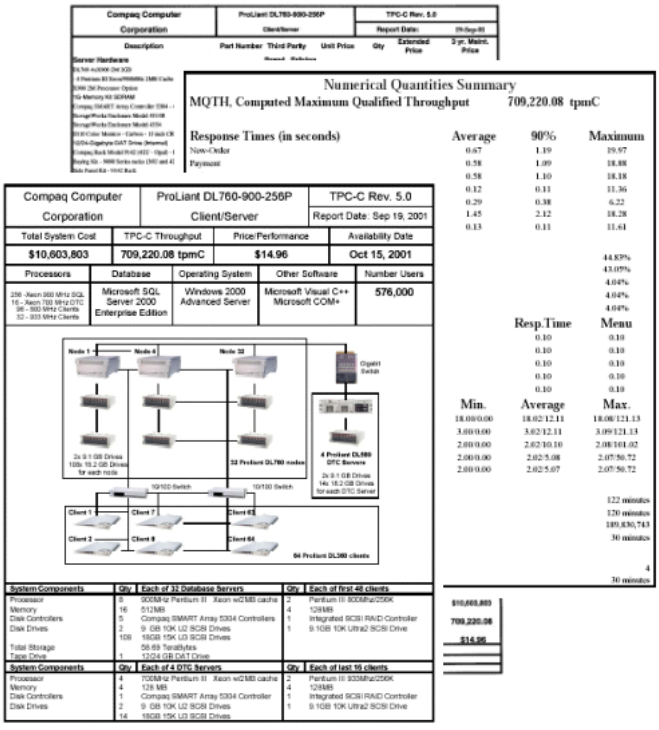
Kto implementuje benchmarki? Najczęściej są to producenci sprzętu i oprogramowania, którzy chcą udokumentować jakość swoich produktów bądź swoją przewagę nad konkurencją. Czasami są to zespoły wdrażające system informatyczny, które mają możliwość testowania sprzętu wypożyczonego przez dostawcę. Kluczową korzyścią, jaką dzięki standardowym benchmarkom uzyskuje konsument jest łatwość porównywania alternatywnych rozwiązań przy pomocy prostych współczynników typu cena/wydajność (*price/performance*).

Należy zwrócić uwagę na to, że w przeszłości dzięki benchmarkom producenci wielokrotnie dopuszczali się nieuczciwych zachowań. W literaturze pojawiły się nawet stosowne terminy dla opisania takich sytuacji: „*wojny na benchmarki*" (*benchmark wars*) czy *benchmarketing*. Ogólny mechanizm typowych nadużyć sprowadzał się zwykle do nierzetelnej implementacji aplikacji testowych i fałszowania wyników eksperymentów.

Obecnie istnieje wiele organizacji i firm zajmujących się definiowaniem standardów benchmarków. Oto kilka przykładów. *SPEC* (Standard Performance Evaluation Cooperation) zajmuje się tworzeniem benchmarków do pomiaru przepustowości systemów jednoprocesorowych, wieloprocesorowych symetrycznych, klastrowych stosowanych dla potrzeb aplikacji ogólnego przeznaczenia i aplikacji internetowych [SPEC]. Najpopularniejsze benchmarki SPEC to: SPECapc (aplikacje z intensywnym przetwarzaniem grafiki), SPEChpc96 (aplikacje przemysłowe na stacjach końcowych), SPEComp2001 (aplikacje wykorzystujące OpenMP), SPECcpu2000 (intensywne wykorzystywanie procesora), SPECweb99 (wydajność serwerów WWW), SPECjbb2000 (Java po stronie serwera), SPECjvm98 (wydajność maszyny wirtualnej Java), SPECsfs97 (wydajność serwerów NFS), SPECmail 2001 (wydajność serwera poczty elektronicznej). *BAPCo* (Business Application Performance Corporation) opracowuje standardy badania wydajności stacji klasy PC [BAPCO]. Najpopularniejsze benchmarki BAPCo to: WebMark (zastosowania internetowe), SysMark (aplikacje biurowe), MobileMark (notebooki - wydajność akumulatorów) i SysMarkDB (serwery baz danych). Firma *SAP* opracowuje rodzinę benchmarków nazywanych SAP (Standard Application Benchmark) dla aplikacji mySAP.com [SAP]. Z kolei *Oracle* publikuje benchmarki nazywane Oracle App Standard

Benchmark, służące do badania wydajności Oracle Applications w różnych architekturach sprzętowych [ORACLE]. Do badania efektywności narzędzi Lotus Notes, *Lotus* stosuje benchmarki NotesBench [LOTUS]. *GPC* (Graphics Performance Characterization Committee) opracowuje testy wydajności aplikacji wykorzystujących bibliotekę graficzną OpenGL [GPC]. Dla potrzeb porównywania wydajności procesorów, *Intel* wykorzystuje grupę benchmarków iComp [INTEL].

Z punktu widzenia zastosowań bazodanowych, największe znaczenie ma konsorcjum *TPC* (Transaction Processing Performance Council), którego celem jest definiowanie standardowych benchmarków dla systemów komputerowych służących do przechowywania danych i przetwarzania transakcji [TPC]. Założone w 1988 roku TPC zrzesza takich producentów, jak: Acer, Bull, Compaq, Dell, Fujitsu, HP, Hitachi, IBM, Intel, Microsoft, Netscape, Oracle, Progress, SCO, Siemens, SGI, Sun, Sybase, Toshiba. Co kilka lat TPC publikuje nowe specyfikacje benchmarków dla różnych klas systemów bazodanowych. Aktualnie stosowane benchmarki to: TPC-C (przetwarzanie OLTP), TPC-R (wsadowe przetwarzanie OLAP), TPC-H (przetwarzanie OLAP ad-hoc) i TPC-W (aplikacje e-commerce). Ogromną zaletą benchmarków definiowanych przez TPC jest ich bardzo dokładna specyfikacja, obejmująca opis warunków testowania systemu, zasady wyliczania kosztów, a nawet format raportów, które muszą być dostarczone TPC jako wyniki eksperymentów. Każdy eksperyment musi zakończyć się przygotowaniem dwóch dokumentów: *executive summary* – kilkustronicowe streszczenie architektury i uzyskanych wyników (rys. 1), oraz *full disclosure report* – duże opracowanie opisujące m.in. fizyczną organizację bazy danych, ustawienia parametrów serwera, implementację generowania bazy danych, implementację aplikacji testowych. Przesłane do TPC wyniki eksperymentów podlegają weryfikacji i dopiero po jej pomyślnym przebiegu mogą zostać opublikowane.




**Rys. 1.** Przykład dokumentu *Executive Summary*, wymaganego przez TPC

W artykule przedstawiono własności najpopularniejszych benchmarków dla systemów baz danych: TPC-C, TPC-H, TPC-R i TPC-W. Struktura pracy jest następująca. Rozdział 2 zawiera rys historyczny rozwoju benchmarków dla systemów baz danych. W rozdziale 3 omówiono benchmark TPC-C, symulujący obciążenie OLTP. Rozdział 4 opisuje benchmarki TPC-R i TPC-H, modelujące obciążenie systemu wspomagania podejmowania decyzji. W rozdziale 5 przedstawiono benchmark TPC-W, przeznaczony dla środowisk e-commerce. Rozdział 6 stanowi podsumowanie.

# 2 Historia benchmarków dla systemów baz danych

W ciągu ostatnich trzydziestu lat intensywnego rozwoju systemów baz danych zaproponowano wiele metod szacowania wydajności tych systemów. Do najbardziej znanych i akceptowanych przez producentów należały niewątpliwie benchmarki: Wisconsin, DebitCredit, TP1, AS$^3$AP, TPC-A, TPC-B, Set Query i TPC-D. W dalszej części rozdziału przedstawiamy ogólną charakterystykę tych historycznych rozwiązań.

Benchmark **Wisconsin** [BITT83] został opracowany na początku lat osiemdziesiątych, początkowo na potrzeby badania wydajności systemu zarządzania bazą danych DIRECT, budowanego na Uniwersytecie Wisconsin. Benchmark Wisconsin posługiwał się bazą danych składającą się z trzech tabel. Jedna z nich, nazwana ONEKTUP, zawierała 1000 rekordów, pozostałe dwie, nazwane TENKTUP1 i TENKTUP2, zawierały po 10,000 rekordów (łącznie ok. 5 MB). Każda tabela była zbudowana z trzynastu kolumn całkowitoliczbowych (dwubajtowych) i z trzech kolumn zawierających 52-bajtowe ciągi znaków. Tabele były wypełniane w sposób jednorodny rekordami generowanymi automatycznie, a wartości kolumn w tych rekordach były bądź losowe, bądź cyklicznie inkrementowane. Kolumny znakowe otrzymywały wartości według wzorca: „losowa litera" – 25 znaków „x" – „losowa litera" – 24 znaki „x" – „losowa litera".

Do pomiaru wydajności służył zestaw pięciu grup zapytań SQL (w sumie 32 zapytania), które realizowały wszystkie podstawowe operacje relacyjne: selekcję z różnymi współczynnikami selektywności, projekcję z różną ilością powtórzonych kolumn, połączenia proste i wielokrotne, proste agregaty i funkcje grupowe oraz dodawanie, usuwanie i modyfikowanie rekordów (tab. 2). Zapytania testowe były szeregowo uruchamiane w ramach pojedynczej sesji użytkownika. Miarami wydajności systemu były zmierzone w sekundach czasy wykonania każdego z zapytań testowych. Uwagi krytyczne w stosunku do benchmarku Wisconsin dotyczyły głównie braku operacji współbieżnych (pojedyncza sesja), braku masowego ładowania danych, braku operacji połączeń zewnętrznych oraz trywialnych zapytań połączeniowych. Niemniej jednak, benchmark Wisconsin stał się na wiele lat standardem de-facto na rynku badania i porównywania wydajności systemów baz danych i systemów komputerowych.

| | | |
|---|---|---|
| `insert into tmp`<br>`select *`<br>`from tenktup1`<br>`where unique2`<br>`  between 0 and 99` | `insert into tmp`<br>`select *`<br>`from tenktup1 t1, tenktup2 t2`<br>`where t1.unique2=t2.unique2`<br>`and t2.unique2 < 1000` | `insert into tmp`<br>`select sum(unique3)`<br>`from tenktup1`<br>`group by onePercent` |

**Tab. 2.** Wybrane zapytania testowe benchmarku Wisconsin

Nowy benchmark dla systemów baz danych zaproponowano w roku 1985. Benchmark **DebitCredit**, nazywany też benchmarkiem Datamation [ANON85], posługiwał się modelem bankowej bazy danych, na której wykonywane były operacje charakterystyczne dla

bankomatów. Baza danych, zbudowana na przykładzie banku Bank of America, informatyzowanego w 1973 roku, składała się z czterech tabel: BRANCH (oddziały banku, 1000 rekordów, 1MB), TELLER (bankomaty, 10,000 rekordów, 1MB), ACCOUNT (konta klientów, 10,000,000 rekordów, 1GB) i HISTORY (90-dniowa historia operacji, 10 GB). Do obciążania bazy danych wykorzystywane były jednakowe transakcje obejmujące: odczyt komunikatu z terminala X.25 w sieci WAN, modyfikację rekordu tabeli ACCOUNT, wstawienie rekordu do tabeli HISTORY, modyfikację rekordu tabeli TELLER, modyfikację rekordu tabeli BRANCH, wysłanie komunikatu do terminala X.25 w sieci WAN i zatwierdzenie transakcji. Pomiędzy kolejnymi transakcjami bankomatu wprowadzane były przestoje, średnio 100 sekund. 95% wszystkich transakcji musiało być ukończonych w czasie jednej sekundy. Podstawowymi miarami wydajności systemu były: przepustowość, mierzona jako maksymalna liczba transakcji na sekundę (*TPS – transactions per second*), oraz cena systemu przypadająca na 1 TPS (koszt pięcioletniego użytkowania sprzętu i oprogramowania).

Wzrost popularności benchmarku DebitCredit przełożył się na większe zainteresowanie opracowanym już wcześniej przez IBM benchmarkiem **TP1**. Benchmark ten, również posługujący się modelem Bank of America (któremu IBM usiłował sprzedać serwery mainframe), zakładał obciążanie systemu bazy danych transakcjami wsadowymi, bez stusekundowych czasów oczekiwania jak w DebitCredit. Podstawową miarą wydajności badanego systemu również była maksymalna liczba transakcji na sekundę (TPS).

W roku 1987 zaproponowano nowy benchmark dla systemów baz danych, który rozwiązywał szereg problemów dostrzeżonych m.in. na przykładzie benchmarku Wisconsin – nazwano go **AS$^3$AP** – ANSI SQL Standard Scalable and Portable Benchmark [Tur87]. Benchmark AS$^3$AP posługiwał się syntetyczną bazą danych, generowaną przez program DBGEN [BMO88]. DBGEN tworzył cztery dziesięciokolumnowe tabele w postaci plików tekstowych, które musiały być następnie załadowane do bazy danych. Wszystkie cztery tabele posiadały jednakową liczbę rekordów (od 10000 do 1 miliarda) oraz jednakową średnią długość rekordu (100 bajtów). W kolumnach tabel zapisane były wartości o rozkładzie zarówno jednorodnym, jak i niejednorodnym. Tabela o nazwie UNIQUES posiadała unikalne wartości we wszystkich kolumnach. W tabeli HUNDRED większość kolumn przechowywała dokładnie sto unikalnych wartości (selektywność 100), a ponadto kolumny te były skorelowane. W tabeli TENPCT większość kolumn posiadała 10% unikalnych wartości (selektywność 10%). Tabela UPDATES służyła do badania efektywności modyfikacji bazy danych. Logiczna struktura każdej tabeli posługiwała się wachlarzem różnych typów danych: typ całkowity bez znaku i ze znakiem, typ zmiennoprzecinkowy, typ dziesiętny, typ alfanumeryczny, ciągi stałej i zmiennej długości oraz ośmioznakowe daty. W czasie eksperymentów w bazie danych pojawiała się jeszcze jedna tabela – miniaturowa TINY, zawierająca jeden rekord i jedną kolumnę – służąca do pomiaru narzutów czasowych realizacji zapytania SQL. Logiczny rozmiar bazy danych musiał być z jednej strony co najmniej równy rozmiarowi fizycznej pamięci operacyjnej testowanego systemu, a z drugiej strony, powinien pozwolić na wykonanie zestawu zapytań obciążających w czasie nie dłuższym niż 12 godzin. Specyfikacja benchmarku AS$^3$AP obejmowała również definicje typów indeksów, jakie mogły być utworzone dla tabel w bazie danych.

Do pomiaru wydajności systemu służyły dwa zestawy poleceń SQL: *single-user tests* i *multi-user tests* (tab. 3). Zestaw pierwszy dokonywał obciążenia systemu przez pojedynczą sesję użytkownika, w ramach której szeregowo następowały operacje ładowania i archiwizacji danych, tworzenia indeksów oraz podstawowe operacje relacyjne: selekcje, proste połączenia, projekcje, agregacje, modyfikacje pojedyncze i masowe. Drugi zestaw poleceń SQL był współbieżnie realizowany przez wiele sesji użytkowników. Ich liczba była określona jako rozmiar bazy danych podzielony przez cztery megabajty (np. dla 40-megabajtowej bazy

danych stosowano 10 sesji współbieżnych). Realizowane polecenia obejmowały współbieżne modyfikacje jedno- i dziesięciorekordowe oraz współbieżne selekcje jedno- i dziesięciorekordowe. Miarą wydajności systemu bazy danych, badanego przy użyciu benchmarku AS$^3$AP był maksymalny rozmiar bazy danych, dla której możliwe było ukończenie wszystkich testowych zestawów poleceń SQL w czasie nie dłuższym niż 12 godzin (tzw. *equivalent database size*).

| | |
|---|---|
| `select count(key)`<br>`from tenpct`<br>`where name='THE+ASAP+BENCHMARKS+'`<br>`and int <= 100000000`<br>`and signed between 1 and 99999999`<br>`and not (float between -450000000`<br>`and 450000000)`<br>`and double > 600000000`<br>`and decim < -600000000` | `begin work`<br>`update updates`<br>`set double = double + 100000000`<br>`where key between 1001 and 1100`<br>`rollback work` |

**Tab. 3.** Wybrane zapytania testowe benchmarku AS$^3$AP

Pod koniec roku 1989 pojawił się pierwszy benchmark zdefiniowany przez TPC: **TPC-A**. Benchmark ten bardzo istotnie bazował na DebitCredit, lecz wprowadzał kilka rozszerzeń. Zamiast wymogu realizacji 95% transakcji w czasie poniżej jednej sekundy, wprowadzono obowiązek realizacji 90% transakcji w czasie poniżej 2 sekund. Sieć WAN pomiędzy terminalem a serwerem bazy danych mogła być zastąpiona siecią lokalną. Zmniejszono również wymagania dotyczące minimalnej liczby terminali testowych, a przestoje pomiędzy kolejnymi transakcjami zredukowano ze 100 sekund do 10 sekund. Bardzo podkreślono konieczność zapewniania własności ACID transakcji na poziomie systemu zarządzania bazą danych i dostarczono specjalne metody testowania tego warunku. Proces badania wydajności systemu został w znaczący sposób sformalizowany i ustandaryzowany. Rok później organizacja TPC opublikowała kolejny benchmark, tym razem bazujący na TP1: **TPC-B**. Podobnie jak TP1, tak i benchmark TPC-B skupiał się na wsadowym przetwarzaniu transakcji bankomatowych.

Pojawiające się pod koniec lat osiemdziesiątych zapotrzebowanie na systemy klasy DSS spowodowało prace nad benchmarkami, które pozwoliłyby ocenić wydajność systemu bazy danych w nowych zastosowaniach. Jedną z pierwszych propozycji był benchmark **Set Query**. Benchmark ten posługiwał się jedną tabelą bazy danych, nazwaną BENCH, zawierającą milion rekordów o 13 indeksowanych kolumnach. Rekordy były generowane automatycznie przez niewielki program opublikowany w [CK80] – jego przekład na język PL/SQL przedstawiono na rys. 4. Badany system obciążany był przez zestaw zapytań obejmujących operacje charakterystyczne dla m.in.: wyszukiwania dokumentów, marketingu kierunkowego, wspomagania podejmowania decyzji i raportowania biznesowego (tab. 5). Miarą wydajności testowanego systemu była jego cena (pięcioletnie użytkowanie sprzętu i oprogramowania) podzielona przez średnią liczbę zapytań wykonanych w ciągu sekundy ($/QPS). Niestety, benchmark Set Query, podobnie jak Wisconsin, posługiwał się tylko jedną sesją użytkownika, w ramach której wszystkie zapytania były wykonywane szeregowo.

```
procedure gen is
  i number(10);
  j number(10);
  seed number:= 1;
  type coltype is varray(12) of number(10);
  colval coltype := coltype(0,0,0,0,0,0,0,0,0,0,0,0);
  colcard coltype := coltype(500000, 250000, 100000, 40000,
                             10000, 1000, 100, 25, 10, 5, 4, 2);
begin   for i in 1..1000000 loop
  for j in 1..12 loop
    seed := mod(16807*seed, 2147483647);
     colval(j) := mod(seed, colcard(j))+1;
     end loop;
    insert into bench values (colval(1), colval(2), ..., colval(12));
  end loop;
end;
/
```

**Rys. 4.** Program generujący tabelę BENCH dla benchmarku Set Query

```
select count(*)
from bench
where k2 = 2 and not k4 = 3
```

```
select sum(k1k)
from bench
where (kseq between 400000 and 410000
    or kseq between 420000 and 430000
    or kseq between 440000 and 450000
    or kseq between 460000 and 470000
    or kseq between 480000 and 500000)
and k4 = 3
```

**Tab. 5.** Wybrane zapytania testowe benchmarku Set Query

Pierwszym „prawdziwym” benchmarkiem dla systemów wspomagania podejmowania decyzji był jednak **TPC-D**, opublikowany przez organizację TPC w roku 1990. Był to pierwszy benchmark, który symulował zapytania ad-hoc, czyli takie, które nie były znane przed uruchomieniem benchmarku. Była to istotna własność, ponieważ uniemożliwiała wcześniejsze dostrojenie badanego systemu „pod benchmark”. Benchmark TPC-D wykorzystywał bazę danych składającą się z ośmiu tabel, symulujących duży system sprzedaży części zamiennych. Dwie trzecie objętości całej bazy danych zajmowała kluczowa tabela LINEITEM, przechowująca pozycje zamówień na części. Wynikiem benchmarku były trzy miary wydajności: średnia geometryczna czasów wykonania poleceń SQL - QppD (*query processing power D*), przepustowość - QthD (*query throughput D*) oraz współczynnik cena/wydajność.

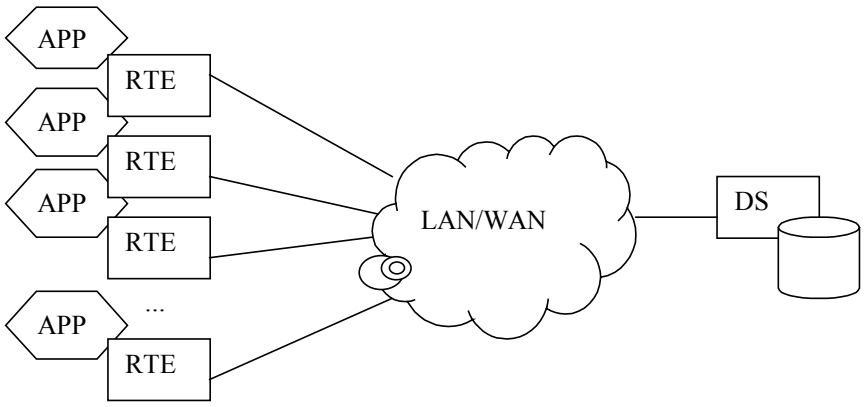
# 3 Badanie wydajności przetwarzania OLTP: TPC-C

We wrześniu 1992 roku organizacja TPC opublikowała nowy benchmark, **TPC-C**, zorientowany na badanie wydajności systemów przetwarzania transakcyjnego (OLAP) – z dużą liczbą użytkowników, z dużą liczbą operacji wprowadzania i modyfikowania danych biznesowych (ostatnia, piąta wersja specyfikacji TPC-C, pochodzi z początku roku 2001) [Tpc01b]. Benchmark TPC-C wyrastał z TPC-A, wzbogacając go dziesięciokrotnie bardziej

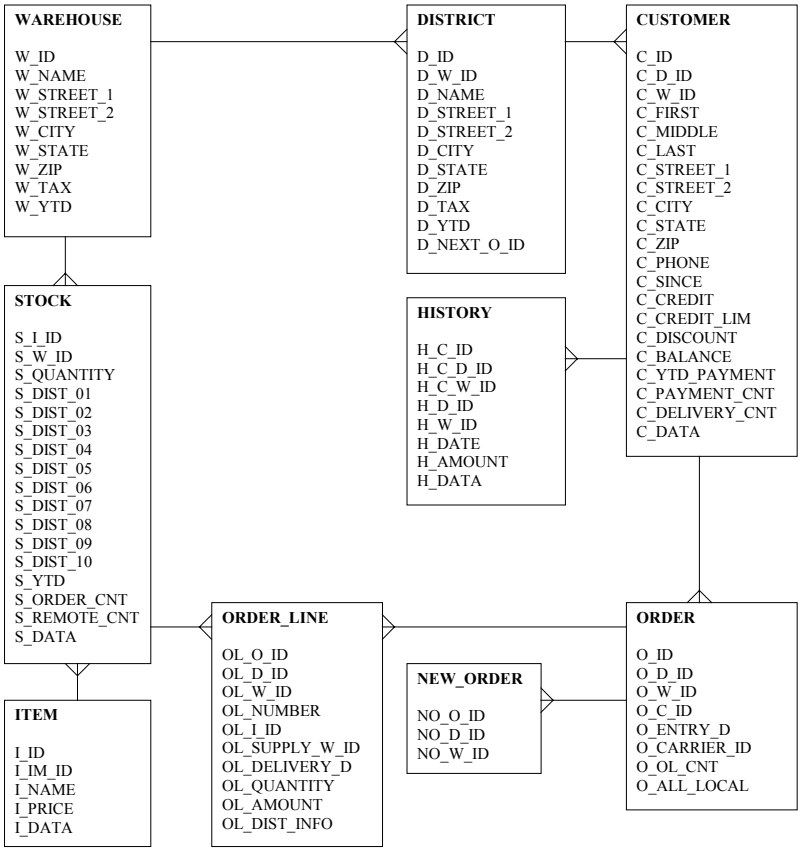
złożonymi transakcjami. Pozwalał na łatwą identyfikację obecności wąskich gardeł w systemie, wynikających m.in. z: rywalizacji o blokady na rekordach współbieżnie modyfikowanych przez wielu użytkowników (tzw. *hotspots*), niewłaściwego poziomu blokowania danych (np. na poziomie strony zamiast na poziomie rekordu), nieefektywnej obsługi buforów dyskowych, nieefektywnej komunikacji sieciowej oraz zapisów do dziennika powtórzeń w celu zapewnienia trwałości transakcji.

Benchmark TPC-C posługuje się dość realistycznym modelem systemu informatycznego dużej sieci hurtowni produktów (rys. 6). Badane transakcje są realizowane za pomocą aplikacji formularzowych (APP), obsługiwanych przez automat (emulator terminala – RTE – *remote terminal emulator*), przygotowany na potrzeby eksperymentu. Czasy odpowiedzi mierzone są po stronie terminala (uwzględniają komunikację sieciową i przetwarzanie w aplikacji-kliencie). Emulowane terminale realizują następujące operacje na dziewięciotabelowej bazie danych (zarządzanej przez serwer bazy danych – DS – *database server*):

- nowe zamówienie – obejmuje wprowadzenie kompletnego zamówienia w ramach pojedynczej transakcji,
- płatność – modyfikacja stanu konta klienta oraz modyfikacja statystyk sprzedaży gromadzonych na poziomie hurtowni i okręgu,
- status zamówienia – odczytanie stanu ostatniego zamówienia danego klienta,
- dostawa – wsadowe przetworzenie dziesięciu zamówień w ramach pojedynczej transakcji,
- stany magazynowe – wyszukiwanie produktów, których stany magazynowe spadły poniżej określonego poziomu,




**Rys. 6.** Typowa architektura systemu testowego TPC-C




**Rys. 7.** Struktura logiczna tabel bazy danych TPC-C

Struktura logiczna wykorzystywanej bazy danych została przedstawiona na rys.7. Pomiary wydajności prowadzone są dla następujących rozmiarów tabel:

- tabela WAREHOUSE: liczba rekordów tak dobrana, aby na każdy rekord przypadało min. 9 tpmC
- tabela DISTRICT: 10 rekordów * liczba rekordów tabeli WAREHOUSE
- tabela CUSTOMER: 30,000 rekordów * liczba rekordów tabeli WAREHOUSE
- tabela HISTORY: 30,000 rekordów * liczba rekordów tabeli WAREHOUSE
- tabela ORDER: 30,000 rekordów * liczba rekordów tabeli WAREHOUSE
- tabela NEW_ORDER: 9000 rekordów * liczba rekordów tabeli WAREHOUSE
- tabela ORDER_LINE: 300,000 rekordów * liczba rekordów tabeli WAREHOUSE
- tabela STOCK: 100,000 rekordów * liczba rekordów tabeli WAREHOUSE
- tabela ITEM: 100,000 rekordów

Zadaniem automatu symulującego pracę użytkownika z aplikacją jest wywoływanie określonych funkcji formularzy oraz wprowadzanie danych zgodnie ze zdefiniowanym profilem obciążenia, obejmującym zarówno czasy przestoju, czasy odpowiedzi, jak i częstotliwości wykonywania aplikacji wzorcowych (rys. 8). Miarą wydajności systemu badanego przy wykorzystaniu benchmarku TPC-C jest przepustowość, wyrażana w jednostkach tpmC (transakcje na minutę – *transactions per minute*). W wyliczeniach przepustowości pod uwagę brane są jednak wyłącznie transakcje typu „nowe zamówienie”. Wtórną miarą jest cena badanego systemu (sprzęt i oprogramowanie) podzielona przez jego przepustowość ($/tpmC). Przy wyznaczaniu ceny pod uwagę brany jest trzyletni okres użytkowania. Przykładowe wyniki benchmarku TPC-C dla różnych systemów przedstawiono w tab. 9.

| Typ transakcji | Częstotliwość | Przestój | Maks. czas odpowiedzi (dla 90% transakcji) |
|---|---|---|---|
| Nowe zamówienie | 45% | 18 s | 5 s |
| Płatność | 43% | 3 s | 5 s |
| Status zamówienia | 4% | 2 s | 5 s |
| Dostawa | 4% | 2 s | 5 s |
| Stany magazynowe | 4% | 2 s | 20 s |

**Tab. 8.** Profil obciążenia TPC-C

| System | Koszt całkowity | TpmC | $/TpmC |
|---|---|---|---|
| Compaq ProLiant DL760-900-256P, MS Windows 2000, MS SQL Server 2000 | $10,603,803 | 709,220.08 | 14.96 |
| Bull Escala PL3200R, AIX 5L V5.2, Oracle9i Database EE | $7,245,205 | 403,255.46 | 17,96 |
| IBM @server xSeries 370 c/s, MS Windows 2000, IBM DB2 UDB 7.1 | $8,530,671 | 440,879,95 | 19,35 |

**Tab.9.** Przykładowe wyniki wydajności dla benchmarku TPC-C

# 4 TPC-H i TPC-R

W kwietniu 1999 roku, dwa nowe benchmarki, **TPC-R** i **TPC-H** zastąpiły dotychczasowy standard TPC-D w dziedzinie badania wydajności systemów wspomagania podejmowania decyzji [Tpc99, Tpc02, PF00]. Benchmark TPC-R symulował obciążenie decyzyjnej bazy danych przez zapytania o charakterze raportowym (dla których uprzednio zestrojono system bazy danych), natomiast benchmark TPC-H symulował obciążenie systemu zapytaniami ad-hoc (nie znanymi wcześniej administratorowi bazy danych).

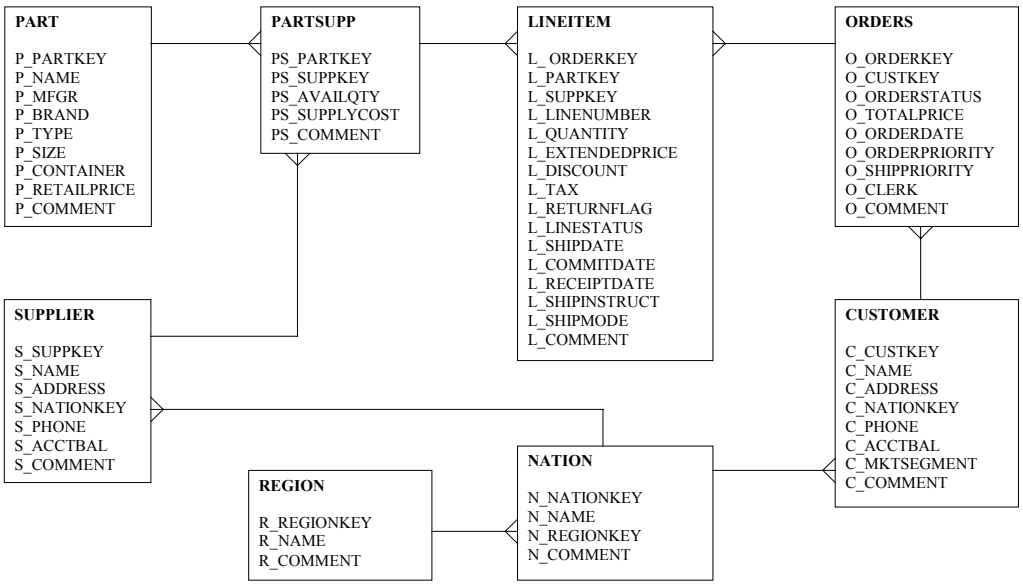
Benchmarki TPC-R i TPC-H posługują się jednakową strukturą bazy danych, zapożyczoną z benchmarku TPC-D (rys. 10). Baza ta składa się z ośmiu tabel, modelujących przedsiębiorstwo globalnej dystrybucji towarów. Organizacja TPC udostępnia program *dbgen*, służący do generowania zawartości testowej bazy danych. Rozmiar całej bazy danych podlega skalowaniu: od 1 GB do 10 TB. Dwie największe tabele, LINEITEM i ORDERS zawierają ok. 83% wszystkich danych.

Do obciążenia systemu, w obu przypadkach, stosuje się zbiory 22 zapytań (16 z nich pochodzi z TPC-D) oraz dwóch funkcji ładowania i usuwania danych. Zapytania modelują następujące operacje biznesowe:

- podsumowanie sprzedaży z podanego dnia,
- wyszukiwanie dostawcy oferującego najniższy koszt,
- wyszukiwanie dziesięciu niezrealizowanych zamówień o najwyższej wartości,
- wyszukiwanie w obrębie kwartału zamówień, których realizacja została opóźniona,
- podliczanie łącznej sprzedaży uzyskanej dzięki dostawcom z wybranego regionu,
- przewidywanie wartości sprzedaży po selektywnej eliminacji ofert promocyjnych,
- podsumowanie międzynarodowej wymiany handlowej (przykład rys. 11),

- zmiana udziału w rynku mierzone w okresie dwuletnim,
- podliczenie zyskowności różnych kategorii towarów,
- podliczenie kosztów przyjmowania zwrotów towarów,
- wyszukiwanie najbardziej zyskownych towarów,
- analiza zmian kosztów przesyłania towarów,
- wyszukiwanie związków pomiędzy klientami a wielkościami ich zamówień,
- monitorowanie odpowiedzi rynku na kampanię reklamową,
- wyszukiwanie największych dostawców,
- wyszukiwanie dostawców wybranej kategorii towarów,
- prognozowanie strat wynikających z odrzucania niewielkich zamówień,
- wyszukiwanie stu największych klientów,
- podliczanie łącznej wartości promocji,
- wyszukiwanie dostawców podatnych na promocję,
- wyszukiwanie dostawców, którzy nie wywiązali się z terminowych zobowiązań,
- wyszukiwanie klientów, którzy od siedmiu lat nie złożyli żadnego zamówienia.

Badanie wydajności systemu odbywa się w toku dwóch procedur testowych. Pierwsza procedura, nazywana *power test*, polega na szeregowym wykonaniu: funkcji ładującej dane, 22 zapytań testowych i funkcji usuwającej dane. Druga procedura, nazywana *throughput test*, polega na wykonywaniu serii zapytań testowych w układzie wielu równoległych sesji, w obecności jednej dodatkowe sesji, w ramach której przeplatają się wywołania funkcji ładowania i usuwania danych.




**Rys. 10.** Struktura logiczna bazy danych TPC-H i TPC-R

```
select supp_nation, cust_nation, l_year, sum(volume) as revenue
from (select n1.n_name as supp_nation, n2.n_name as cust_nation,
      extract(year from l_shipdate) as l_year,
      l_extendedprice * (1 - l_discount) as volume
      from supplier, lineitem, orders, customer, nation n1, nation n2
      where s_suppkey = l_suppkey
        and o_orderkey = l_orderkey
        and c_custkey = o_custkey
        and s_nationkey = n1.n_nationkey
        and c_nationkey = n2.n_nationkey
        and ((n1.n_name = '[NATION1]' and n2.n_name = '[NATION2]')
          or (n1.n_name = '[NATION2]' and n2.n_name = '[NATION1]'))
        and l_shipdate between date '1995-01-01' and date '1996-12-31'
      ) as shipping
group by supp_nation, cust_nation, l_year
order by supp_nation, cust_nation, l_year;
```

**Rys. 11.** Przykładowe zapytanie benchmarku TPC-H

Podstawowymi miarami wydajności systemu badanego przy użyciu benchmarków TPC-H i TPC-R są przepustowości wyrażone w QphH (*TPC-H query per hour*) i QphR (*TPC-R query per hour*). Wartości powyższych miar wyliczane są według następujących formuł:

$$power = \frac{3600 * SF}{\sqrt[24]{\prod_{i=1}^{22} QI(i,0) * \prod_{j=1}^{2} RI(j,0)}}$$

$$throughput = \frac{S * 22 * 3600}{T_S} * SF$$

$$Qph = \sqrt{power * throughput}$$

gdzie:

SF – rozmiar bazy danych wyrażony w GB,
QI(i,0) – wyrażony w sekundach czas wykonania zapytania testowego numer *i*,
RI(j,0) – wyrażony w sekundach czas wykonania funkcji numer *j*,
S – liczba sesji (strumieni) zapytań testowych,
$T_S$- wyrażony w sekundach czas wykonania wszystkich sesji zapytań testowych.

| **System** | **Koszt całkowity** | **Rozmiar bazy danych** | **QphH** | **$/QphH** |
|---|---|---|---|---|
| HP AlphaServer ES45 Cluster, HP Tru64 Unix, Oracle9iR2 | $2,072.818 | 100GB | 5578.4 | 371.58 |
| HP 9000 Superdome ES, HP UX 11.i 64-bit, Oracle9iR2 | $6,024,393 | 3TB | 27,094.3 | 222 |

**Tab.12.** Przykładowe wyniki wydajności dla benchmarku TPC-H

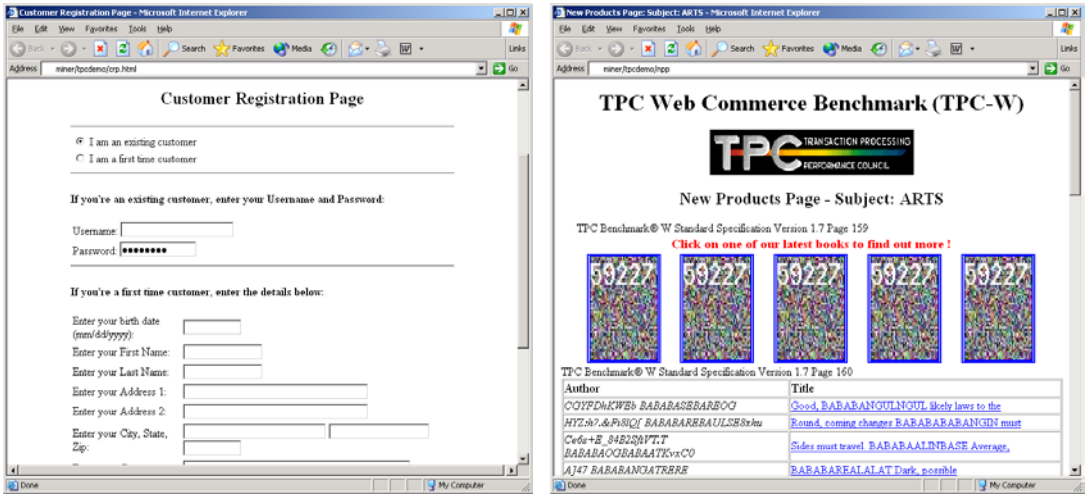
# 5 Badanie wydajności środowisk e-commerce: TPC-W

W lutym 2000, organizacja TPC opublikowała nowy benchmark poświęcony badaniu wydajności środowisk e-commerce: **TPC-W** [Tpc01a, PF00, Smith01]. Podstawę tego benchmarku stanowi symulacja obciążenia systemu księgarni internetowej przez klientów realizujących operacje przeglądania i zakupu produktów (rys. 13). Modelowany system jest obciążany przez emulowane przeglądarki, które zgodnie z zadanym wzorcem przekazują żądania pobrania stron. Specyfikacja TPC-W dokładnie określa strukturę każdej z czternastu stron HTML składających się na cały system:

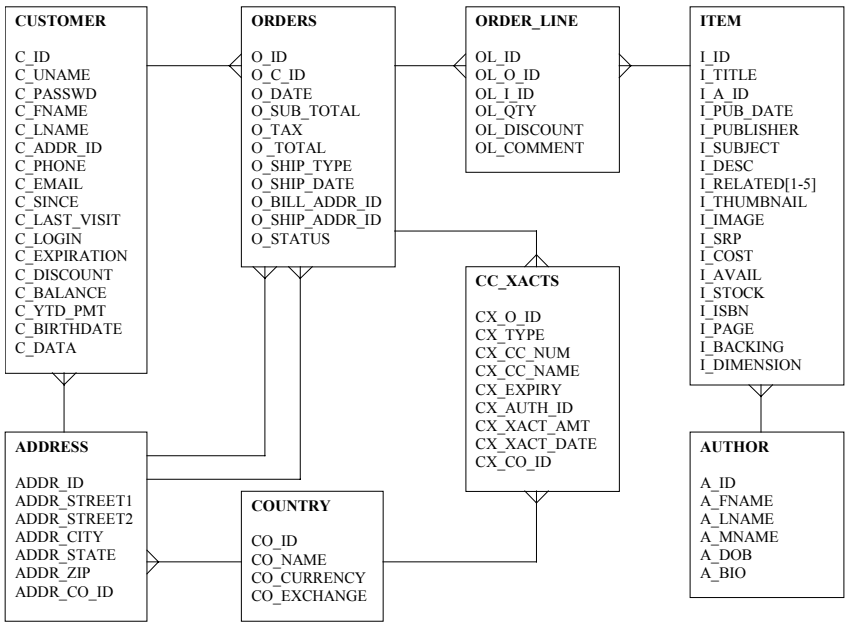
- strona domowa – zawiera łączniki do list nowych produktów, bestsellerów, formularza wyszukiwania, koszyka zakupów i statusu zamówienia; rozpoczyna nową sesję użytkownika,
- koszyk zakupów – przedstawia aktualny stan zamówienia, pozwalając na jego edycję
- rejestracja klienta – umożliwia klientowi zalogowanie się lub założenie nowego konta,
- złożenie zamówienia – służy potwierdzeniu zamówienia, wprowadzeniu informacji o płatności i adresie wysyłki,
- potwierdzenie zamówienia – prezentuje treść przyjętego zamówienia, jego łączną wartość oraz nadany numer zamówienia,
- żądanie wyświetlenia zamówienia – umożliwia klientowi autoryzację w celu przejrzenia treści ostatniego złożonego zamówienia,
- wyświetlenie zamówienia – umożliwia klientowi przejrzenie treści ostatniego złożonego zamówienia,
- formularz wyszukiwania – służy do wprowadzenia warunku wyszukiwania książek,
- wyniki wyszukiwania – prezentuje listę znalezionych książek,
- nowe produkty – wyświetla listę nowowydanych książek,
- bestselery – wyświetla listę najlepiej sprzedawanych książek,
- opis szczegółowy produktu – wyświetla szczegółowe informacje o wybranej książce, wraz z fotografią okładki,
- formularz administracyjny – umożliwia modyfikację niektórych parametrów produktu: ceny, ikony, fotografii okładki,
- potwierdzenie operacji administracyjnej – wyświetla potwierdzenie wykonania modyfikacji wybranych parametrów produktu,

Baza danych wykorzystywana przez oprogramowanie sklepu internetowego składa się z ośmiu nieskomplikowanych tabel, przechowujących informacje o klientach, książkach i zamówieniach (rys. 14). Pomiary wydajności prowadzone są dla następujących rozmiarów bazy danych:

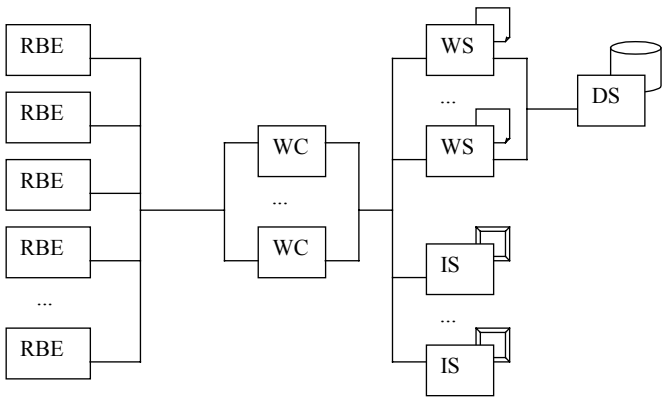
- tabela CUSTOMER: liczba emulowanych przeglądarek * 2880 rekordów
- tabela COUNTRY: 92 rekordy
- tabela ADDRESS: 2 * liczba rekordów tabeli CUSTOMER
- tabela ORDERS: 0.9 * liczba rekordów tabeli CUSTOMER
- tabela ORDER_LINE: 3 * liczba rekordów tabeli ORDERS
- tabela AUTHOR: 0.25 * liczba rekordów tabeli ITEM
- tabela CC_XACTS: 1 * liczba rekordów tabeli ORDERS
- tabela ITEM: 1k, 10k, 100k, 1M, 10M rekordów




**Rys. 13.** Przykładowe strony księgarni internetowej TPC-W




**Rys. 14.** Struktura logiczna tabel bazy danych TPC-W




**Rys. 15.** Ogólna architektura systemu testowego

Ogólna architektura systemu testowego została przedstawiona na rys. 15. System składa się z: (1) emulatorów przeglądarek, (2) bramy do serwera obsługi płatności, (3) serwerów WWW, (4) serwerów obrazów graficznych, (5) serwerów buforowych WWW i (6) serwera bazy danych. Zadaniem emulatorów przeglądarek (RBE – remote browser emulator) jest generowanie obciążenia systemu, zgodnie z narzuconym wzorcem. Specyfikacja TPC-W posługuje się trzema modelami obciążenia (tab. 16): zrównoważonym WIPS (shopping mix), zorientowanym na serwer WWW WIPSb (browsing mix) oraz zorientowanym na serwer bazy danych WIPSo (ordering mix). Brama do serwera obsługi płatności (PGE – payment gateway emulator) służy do symulacji autoryzacji kart kredytowych wykorzystywanych do realizacji zakupów. Serwery WWW (WS – web server) dostarczają statyczną treść HTML, natomiast wszystkie obrazy graficzne są udostępniane przez serwery obrazów graficznych (IS – image server). Udostępniana treść HTML może być buforowana przez serwery buforowe WWW (WC – web cache). Wszystkie dane są przechowywane w bazie danych, obsługiwanej przez serwer bazy danych (DS – database server).

| | Browsing Mix (WIPSb) | Shopping Mix (WIPS) | Ordering Mix (WIPSo) |
|---|---|---|---|
| **Przeglądanie** | **95 %** | **80 %** | **50 %** |
| Strona domowa | 29.00 % | 16.00 % | 9.12 % |
| Nowe produkty | 11.00 % | 5.00 % | 0.46 % |
| Bestsellery | 11.00 % | 5.00 % | 0.46 % |
| Opis szczegółowy produktu | 21.00 % | 17.00 % | 12.35 % |
| Formularz wyszukiwania | 12.00 % | 20.00 % | 14.53 % |
| Wyniki wyszukiwania | 11.00 % | 17.00 % | 13.08 % |
| **Zamawianie** | **5 %** | **20 %** | **50 %** |
| Koszyk zakupów | 2.00 % | 11.60 % | 13.53 % |
| Rejestracja klienta | 0.82 % | 3.00 % | 12.86 % |
| Złożenie zamówienia | 0.75 % | 2.60 % | 12.73 % |
| Potwierdzenie zamówienia | 0.69 % | 1.20 % | 10.18 % |
| Żądanie wyświetlenia zamówienia | 0.30 % | 0.75 % | 0.25 % |
| Wyświetlenie zamówienia | 0.25 % | 0.66 % | 0.22 % |
| Formularz administracyjny | 0.10 % | 0.10 % | 0.12 % |
| Potwierdzenie operacji administracyjnej | 0.09 % | 0.09 % | 0.11 % |

**Tab. 16.** TPC-W: Profile obciążenia

Podstawowymi miarami benchmarku TPC-W są: liczba obsłużonych żądań na sekundę (*web interactions per second*) dla każdego modelu obciążenia (WIPS, WIPSb, WIPSo) oraz współczynnik cena/wydajność wyznaczony jako koszt całego testowanego systemu podzielony przez liczbę obsłużonych żądań modelu WIPS na sekundę. W tab. 17 przedstawiono wybrane wyniki benchmarków TPC-W, zatwierdzonych przez TPC.

| System | \|ITEM\| | Koszt całkowity | WIPS | WIPSo | WIPSb | $/WIPS |
|---|---|---|---|---|---|---|
| Dell PowerEdge 6400/900MHz, MS IIS 5.0, MS SQL Server 2000 | 10K | $190,000 | 7783.3 | 1209.8 | 7828.8 | 24.50 |
| IBM @server xSeries 350, MS IIS 5.0, MS SQL Server 2000 | 10K | $224,698 | 7073.7 | 1042.2 | 7772.4 | 31.77 |
| Unisys e-@ction Enterprise Server ES7000 (16P), MS IIS 5.0, MS SQL Server 2000 | 100K | $1,114,138 | 10,439.6 | 4,406.5 | 11,122.7 | 106.73 |

**Tab. 17.** Wybrane wyniki benchmarków TPC-W

# 6 Podsumowanie

Benchmarki służą do symulowania realistycznego obciążenia systemu informatycznego w celu pomiaru wydajności jego pracy. W artykule omówiono cztery najważniejsze benchmarki stosowane w ocenie systemów baz danych. Benchmark TPC-C modeluje obciążenie OLTP, realizowane przez wielu współbieżnych użytkowników wykonujących krótkie transakcje. Benchmarki TPC-R i TPC-W symulują pracę użytkowników w środowisku hurtowni/magazynu danych, wykorzystywanej do wspomagania podejmowania decyzji. Zapytania są skomplikowane, posługują się wielokrotną agregacją i złożonymi połączeniami. Benchmark TPC-W dzieli obciążenie systemu pomiędzy serwer bazy danych oraz serwer aplikacji internetowych.

Znajomość zasad działania benchmarków pozwala projektantom i managerom IT na podejmowanie celniejszych decyzji o wyborze właściwej konfiguracji sprzętowo-programowej. Dzięki analizie raportów TPC, koszty wdrożenia systemu informatycznego mogą być łatwiej optymalizowane. Nie należy jednak zapominać o tym, że najlepszym benchmarkiem jest rzeczywisty system obsługiwany przez rzeczywisty użytkowników. Wtedy jednak jest zwykle za późno na zmianę konfiguracji.

# 7 Bibliografia


[ANON85] Anon. et al. (artykuł anonimowy, m.in. Jim Gray, Hector Garcia-Molina, Stefano Ceri, Mike Stonebraker, Omri Serlin), "A Measure of Transaction Processing Power", Datamation, Vol. 31(7), pp.112-118, 1985

[BAPCO] Business Applications Performance Corporation, http://www.bapco.com

[BITT83] Bitton, D., DeWitt, D.J., Turbyfil, C., „Benchmarking Database Systems: A Systematic Approach", Proc. Of. VLDB 1983, 1983

[BMO88] Bitton, D., Millman, J., Orji, C., "Program Documentation for DBGEN, a test database generator", University of Illinois at Chicago, 1988

[CK80] Cheney, W., Kincaid,, D., "Numerical Mathematics and Computing", Brooks/Cole Publishing Company, 1980

[GPC] Graphics Performance Characterization Committee, http://www.specbench.org/gpc/

[Gray93] "The Benchmark Handbook", Morgan Kaufmann Publishers, ed. Jim Gray, 1993

[INTEL] Intel, http://www.intel.com

[LOTUS] Notesbench, http://www.notesbench.org
[ORACLE] Oracle Applications Standard Benchmark, http://oracle.com/apps_benchmark
[PF00] Poess, M., Floyd, C., "New TPC Benchmarks for Decision Support and Web Commerce", ACM SIGMOD Record, 29(4), 2000
[SAP] SAP Benchmark, http://www.sap.com/benchmark
[Smith01] Smith, W.D., "TPC-W: Benchmarking An Ecommerce Solution", Intel Corporation, 2001
[SPEC] Standard Performance Evaluation Cooperation, http://www.spec.org
[Tpc01a] Transaction Processing Performance Council, "TPC Benchmark W (Web Commerce) Specification", TPC, 2001
[Tpc01b] Transaction Processing Performance Council, "TPC Benchmark C Standard Specification", TPC, 2001
[Tpc02] Transaction Processing Performance Council, "TPC Benchmark H (Decision Support) Standard Specification", TPC, 2002
[Tpc99] Transaction Processing Performance Council, "TPC Benchmark R (Decision Support) Standard Specification", TPC, 1999
[Tur87] Turbyfill, C., "Comparative Benchmarking of Relational Database Systems", Ph.D. Dissertation, Cornell University, 1987