

JVM98

SPEC JVM Client 98

Przeznaczony dla komputerów z dyskiem (PC, stacja robocza) lub bez (komputer sieciowy). Uruchamiany jako applet - potrzebna przeglądarka WWW lub *applet viewer*. Mierzy szybkość wykonania *Java byte-code* przez JVM.

Wymagania

JVM - kompatybilna przynajmniej z Java 1.1.

Pamięć - 3 kategorie: < 48, 48 - 256, > 256 MB.

Dysk - nie wymagany.

System operacyjny - nie sprecyzowany.

Serwer WWW - serwowanie klas i danych dla appletu.

Serwer pocztowy SMTP - wysyłanie wyników.

Testy

Aplikacje testujące w postaci *byte-code* dostarczane przez różnych producentów (którzy nie muszą ujawniać źródeł).

8 różnych programów testujących, z których 5 to rzeczywiste aplikacje lub ich fragmenty.

`_200_check` - sprawdzenie JVM / SPEC

`_201_compress` - kompresja LZW

`_202_jess` - system ekspertowy / Sandia Laboratories

`_209_db` - zarządzanie danymi / IBM

`_213_javac` - JDK Java compiler / Sun Microsystems

`_222_mpegaudio` - dekodowanie strumienia MP3 / FIIS

`_227_mtrt` - raytracing

`_228_jack` - parser-generator / Sun Microsystems

Mierzony czas potrzebny do: załadowania programu, weryfikacji plików klas, ew. skompilowania w locie (JIT) oraz uruchomienia.

Każdy test uruchamiany kilkakrotnie (minimum 2) - dwa wyniki: najlepszy i najgorszy. Ostateczny wynik jest średnią geometryczną z wyników wszystkich testów (oprócz `_200_check`) znormalizowaną względem maszyny odniesienia (133MHz PowerPC 604, RAM 96MB, AIX 4.1.5.0, JDK 1.1.4 (JIT: wył.)).

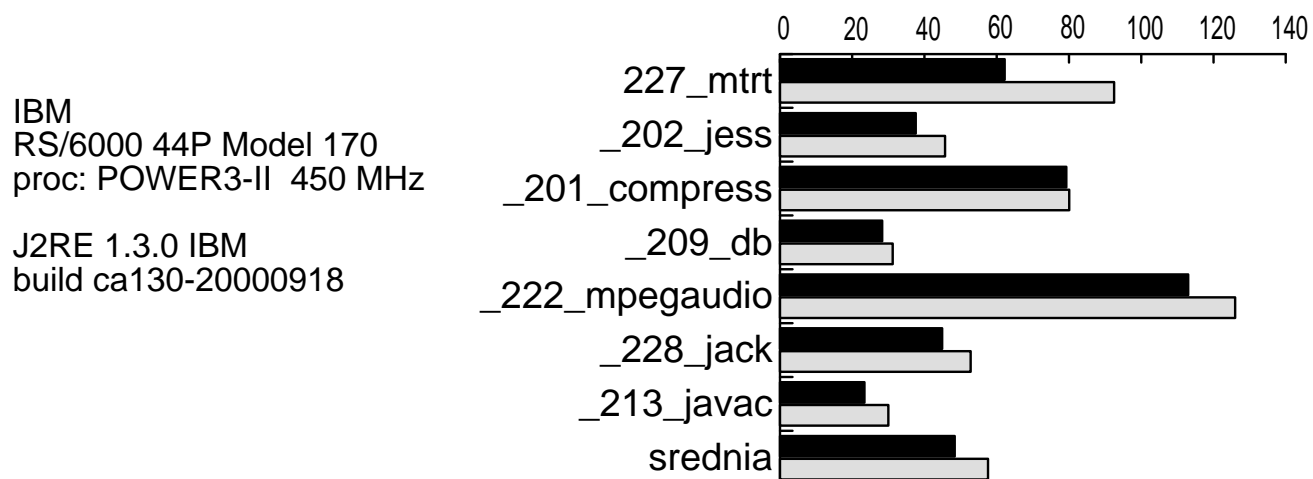
Większa liczba - lepszy wynik.

Metryka `SPECjvm98` - najlepsze wyniki.

Metryka `SPECjvm_base98` - najgorsze wyniki.

Publikowanie wyników

Z każdego z testów, obie metryki, średnia. Tabele, wykresy.



JBB2000

Java Business Benchmark

Zaprojektowany do mierzenia wydajności Javy po stronie serwera (ale nie servlety, EJB czy JSP), na pojedynczej maszynie.

Program emulujący system 3-warstwowy (z naciskiem na warstwę środkową):

1. Generowanie zdarzeń wejściowych przez użytkownika.
2. Logika aplikacji.
3. Dane (baza danych).

Całkowita niezależność (własne dane, własne wielowątkowe operacje, nie wymaga nic poza JRE).

Inspirowany TPC-C (ale niekompatybilny!).

Tabele reprezentowane przez klasy, rekordy przez obiekty.

Obiekty umieszczone w pamięci (BTrees).

Brak dyskowych operacji I/O.

Brak sieciowych operacji I/O.

Model

Modelowanym systemem jest hurtownia z magazynami obsługującymi pewną ilość okręgów.

Klienci (użytkownicy terminali) mogą zainicjować pewne operacje, np.: złożenie zamówienia, sprawdzenie statusu złożonego zamówienia itp.

Operacje generowane przez firmę to np.: przetwarzanie zamówień, płatności itp.

W SPECjbb2000 tylko jeden terminal (klient) jest aktywny dla jednego magazynu. Magazynem jest zbiór ok. 25MB danych (zorganizowanych w BTree). Terminalem jest wątek Javy. Każdy wątek wykonuje po kolei operacje pobierane z pewnym rozkładem prawdopodobieństwa ze zbioru operacji. Ilość wątków (magazynów) zwiększa się w miarę wykonywania testów.

Pomiary

Mierzoną wartością jest ilość wykonanych operacji na sekundę.

Punktem pomiarowym jest dwuminutowy przedział czasu, w którym zliczona zostaje ilość operacji wykonana przy pewnej liczbie magazynów.

Metryka

Pomiary dokonywane są dla wszystkich punktów (ilości magazynów) od 1 przynajmniej do podwojonej ilości magazynów dającej szczytową wydajność. Minimum to punkty od 1 do 8.

Szczytową wydajność obserwujemy w punkcie N .

Wyniki dla punktów od N do $2N$ są uśredniane. Średnia ta stanowi metrykę SPECjbb2000.

Publikowanie wyników

Wydajności dla wszystkich ilości magazynów, średnia wydajność zgodnie z metryką. Wykres wydajności w zależności od ilości magazynów.

