Borland®
THE OPEN ALM COMPANY

Śledzenie problemów wydajnościowych aplikacji aż do ich źródła

Jak skutecznie testować aplikacje pod obciążeniem i dokładnie diagnozować wszystkie pojawiające się wtedy problemy

Luty 2008

Wstęp

Optymalizacja wydajności działania jest kluczowym zadaniem stawianym przed dzisiejszym oprogramowaniem dla biznesu, które z dnia na dzień staje się coraz bardziej złożone, miedzy innymi ze względu na powstawanie nowych standardów oraz trendów takich jak Service Oriented Architectures (SOA) czy Enterprise Application Integrations (EAI).

Jednakże samo narzędzie do testowania obciążeniowego nie jest wystarczającym rozwiązaniem pomagającym dostarczać aplikacje pozbawione słabych stron. Praktyka pokazuje, że o ile narzędzia do badania wydajności są w stanie wykazać, że istnieje problem, o tyle najczęściej wcale nie pomagają nam w odnalezieniu jego źródła. W takich przypadkach mamy najczęściej do czynienia z małą wojną pomiędzy działami jakości i programistów, polegającą na przerzucaniu odpowiedzialności na drugą stronę, co odsuwa w czasie finalne uruchomienie systemu i zwiększa całkowite koszty dostarczenia produktu.

Jedynie narzędzie do testów wydajnościowych z wbudowanym modułem do głębokiej diagnostyki może zapewnić to, czego inżynierowie tak naprawdę poszukują: sposobu na szybkie wykrycie, zlokalizowanie i usunięcie źródła problemu wydajnościowego. Borland dostarcza właśnie takie rozwiązanie w postaci narzędzia SilkPerformer z wbudowanym systemem dynaTrace Diagnostics. Takie podejście umożliwia jasną komunikację podczas automatycznych testów wydajności dzięki możliwości jednoznacznego zidentyfikowania, znalezienia a dzięki temu i łatwego usunięcia wszystkich problemów obciążeniowych krytycznych aplikacji i systemów.

Wydajność jest kluczowym parametrem wszystkich systemów biznesowych

Większość procesów biznesowych o krytycznym znaczeniu dla firmy opiera się dzisiaj na oprogramowaniu, którego wydajność jest kluczowym elementem w ich walce konkurencyjnej. Wydajność software'u to wydajność biznesu – błędy w zainstalowanych systemach czy chociażby zbyt długi czas reakcji oprogramowania mogą prowadzić do olbrzymich strat pieniężnych i wpływać na osłabienie pozycji firmy na rynku. Stąd szczegółowa weryfikacja aplikacji biznesowych pod kątem stawianych przed nią oczekiwań, zarówno funkcjonalnych jak i wydajnościowych stała się dziś oczkiem w głowie zarządów IT.

Budowanie przewagi konkurencyjnej zależy zwykle od czasu dostarczania produktów na rynek oraz od zadowolenia klienta. Problemy z wydajnością mogą opóźnić lub nawet uniemożliwić wdrożenie krytycznych usług biznesowych. Trafne dotarcie do źródła problemów wydajnościowych aplikacji we wczesnych etapach cyklu wytwarzania oprogramowania jest kluczowym wyzwaniem stawianym przed firmami i działami zajmującymi się produkcją i jakością aplikacji.

Zawiły i niewydajny proces rozwiązywania kłopotów z wydajnością systemów nie tylko zwiększa ryzyko utraty satysfakcji klienta, ale również w znaczącym stopniu wydłuża czas dostarczenia produktu – stawiając pod znakiem zapytania przyszłość działalności całego przedsiębiorstwa.

Diagnostyka problemów wydajnościowych jest trudnym zadaniem

Złożoność dzisiejszych aplikacji biznesowych wciąż rośnie wraz z rozwojem nowych technologii oraz zmianami rynkowymi czy wymogami legislacyjnymi. Rozwiązania z obszaru SOA czy EAI coraz sprawniej wspierają złożoność rozbudowanych systemów, w rezultacie oferując mocno rozproszone, heterogeniczne architektury, nie tylko trudne w utrzymaniu, ale jednocześnie wymagające złożonych metod testowania, szczególnie w zakresie ich wydajności.

Dodatkowo spora liczba dostępnych nakładek środowiskowych, jak Java™ EE, .NET, Hibernate® czy Struts, pozwala na jeszcze szybsze tworzenie złożonych aplikacji. Coraz częściej systemy biznesowe działają dzięki jednoczesnej współpracy wielu aplikacji. Przykładem mogą być chociażby struktury typu klient-serwer czy te oparte o Web Service'y.

Podczas gdy najnowsze środowiska programistyczne pozwalają obecnie na znaczne zmniejszenie kosztu produkcji oprogramowania, produkty wynikowe stają się coraz trudniejsze do przetestowania i zoptymalizowania pod kątem wydajności. Wynikiem może być działanie aplikacji z gorszą od zakładanej wydajnością. Efektywna diagnostyka obciążeniowa stanowi podstawę do dalszych oszczędności kosztów produkcji i wsparcia.

Rodzi się pytanie, ile czasu oraz pracy należy włożyć w zapewnienie odpowiednio wydajnego funkcjonowania naszego produktu. W przypadku nowoczesnych, złożonych projektów programistycznych należyta wydajność końcowej aplikacji biznesowej wymaga

ścisłej współpracy pomiędzy wszystkimi działami i zespołami, biorącymi udział w procesie produkcji oprogramowania.

Minimalizacja czasu naprawy błędów oraz ilości cykli testowych

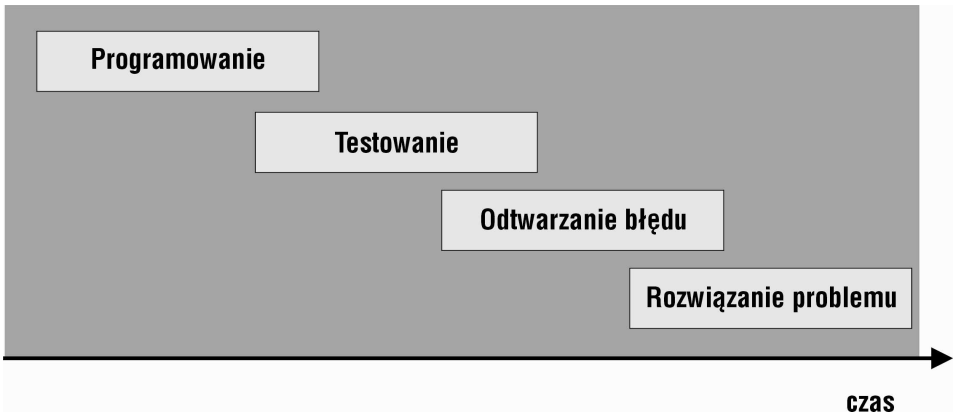
Z punktu widzenia biznesu, wymagania wydajnościowe powinny być uwzględnione z poziomu technicznego. Na poziomie technicznym, IT, możemy dowolnie definiować i przyjmować parametry wydajnościowe takie, które najlepiej pozwolą na zachowanie poprawności i jakości produktu. **Dostępność** jest miarą określającą dostępność serwisu dla użytkowników. **Czas reakcji** mówi nam o tym, jak szybko dany serwis będzie dostępny od jego wywołania – w praktyce jest to kluczowa miara dla użytkowników końcowych. **Wydajność** określa zaangażowanie zasobów systemowych w dostarczanie usług przez system. **Skalowalność** to parametr mówiący o umiejętności radzenia sobie aplikacji z rosnącą liczbą użytkowników i danych.

Testowanie wydajnościowe weryfikuje czas reakcji oraz niezawodność aplikacji pod konkretnym obciążeniem. Jednakże, z uwagi na fakt, że każda aplikacja poddawana testom obciążeniowym, w końcu wykaże problemy z wydajnością, samo testowanie nam nie wystarczy. Oczywiste jest, że stając przed konkretnym problemem wydajnościowym, chcemy się go pozbyć, aby wzmocnić i uodpornić nasz produkt. I tu właśnie pojawia się kwestia właściwej diagnostyki.

Diagnostyka wydajnościowa to analiza problemów wydajnościowych polegająca na coraz głębszym drążeniu źródła problemu. W złożonych architekturach dzisiejszych aplikacji proces ten jest wyjątkowo złożony i skomplikowany. Wymaga przede wszystkim bardzo głębokiej wiedzy na temat wewnętrznej pracy aplikacji jak również analitycznych umiejętności śledzenia problematycznych transakcji w głąb kodu źródłowego. Wszystkie te elementy muszą dodatkowo uwzględniać rozproszone aplikacje działające na wielu popularnych platformach, jak Java™ czy .NET.

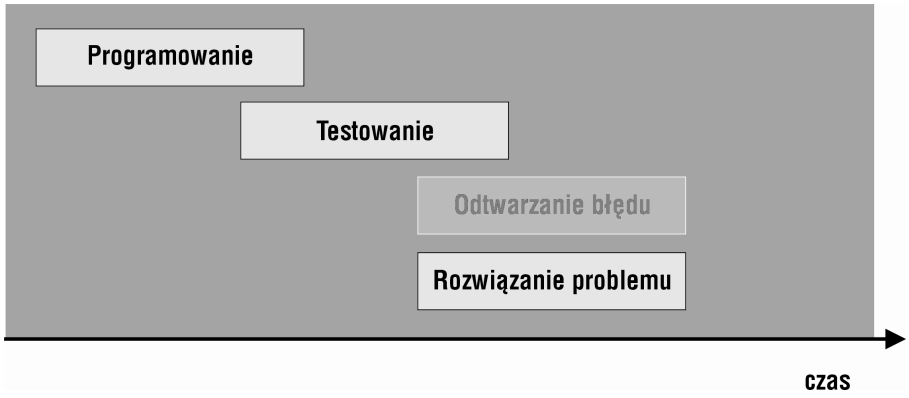
Oprócz wymaganego wsparcia technicznego, w procesie diagnostycznym konieczna jest współpraca pomiędzy wszystkimi uczestnikami procesu wytwarzania aplikacji. Niewystarczająca komunikacja w zespołach i między nimi bardzo często staje się źródłem generowania dodatkowych kosztów wynikających z problemów wydajnościowych.

Częstą praktyką jest wykrywanie przez Dział Jakości problemu z wydajnością systemu i zgłaszanie go programistom. Czasami jednak błąd określony jako: „Transakcja klienta X działa zbyt wolno" nie jest jednoznaczny na tyle, aby mógł zostać naprawiony przez prostą modyfikację kodu aplikacji. Dodatkowo dział programistów, najczęściej operujący na osobnym środowisku, nie zawsze jest w stanie zasymulować podobną sytuację. Dzieje się tak głównie dlatego, że na środowisko programistów składa się przeważnie pojedynczy serwer, a nie jak w docelowym systemie, rozproszony układ wieloklastrowy, często udostępniający kilkanaście lub nawet kilkadziesiąt systemów biznesowych jednocześnie. Taka sytuacja wymaga większych nakładów pracy potrzebnych do powtórzenia problemu zgłoszonego przez QA w celu zidentyfikowania jego źródła. Taki proces może wyglądać jak przedstawiono na *Rysunku 1*.



Rysunek 1. Klasyczne podejście w rozwiązywaniu problemów aplikacji

Najprostszym sposobem jest zgromadzenie tego typu informacji podczas testowania wydajnościowego. Wymaga to jednak specjalnie zaprojektowanego rozwiązania pozwalającego na monitorowanie wszystkich ścieżek transakcyjnych w całym heterogenicznym środowisku rozproszonych komponentów naszej aplikacji. Do tego takie rozwiązanie nie może w żaden sposób dodatkowo obciążać testowanego systemu, ponieważ sam proces testowania mógłby być generatorem problemów, które z uwagi na to, nie mogłyby zostać zasymulowane ani tym bardziej usunięte przez dział programistyczny. Ponadto informacje zgromadzone przez proces diagnostyczny powinny zawierać wszelkie istotne dane niezbędne do rekonstrukcji problemu przez architektów systemowych bez konieczności losowego dobierania ich w procesie reprodukcji stanu aplikacji.



Rysunek 2. Ulepszone podejście w rozwiązywaniu problemów aplikacji

Efektywne testowanie i analiza wydajności

Testowanie wydajnościowe i dostrajanie aplikacji zależy od kilku kluczowych czynników:

- **Dokładna emulacja rzeczywistych użytkowników** korzystających z aplikacji, aby realistycznie odwzorować naturalne zachowanie i warunki obciążenia
- **Efektywne raportowanie** wyników testów wydajnościowych oraz zdiagnozowanych źródeł problemów z punktu widzenia użytkownika
- **Pełna analiza** wykrytych problemów z wydajnością, przez wszystkie poziomy, od symptomów (np. czas reakcji aplikacji nie do przyjęcia) aż do samego źródła w kodzie aplikacji (np. nieoptymalne zapytanie SQL)

Zintegrowany proces diagnostyczny

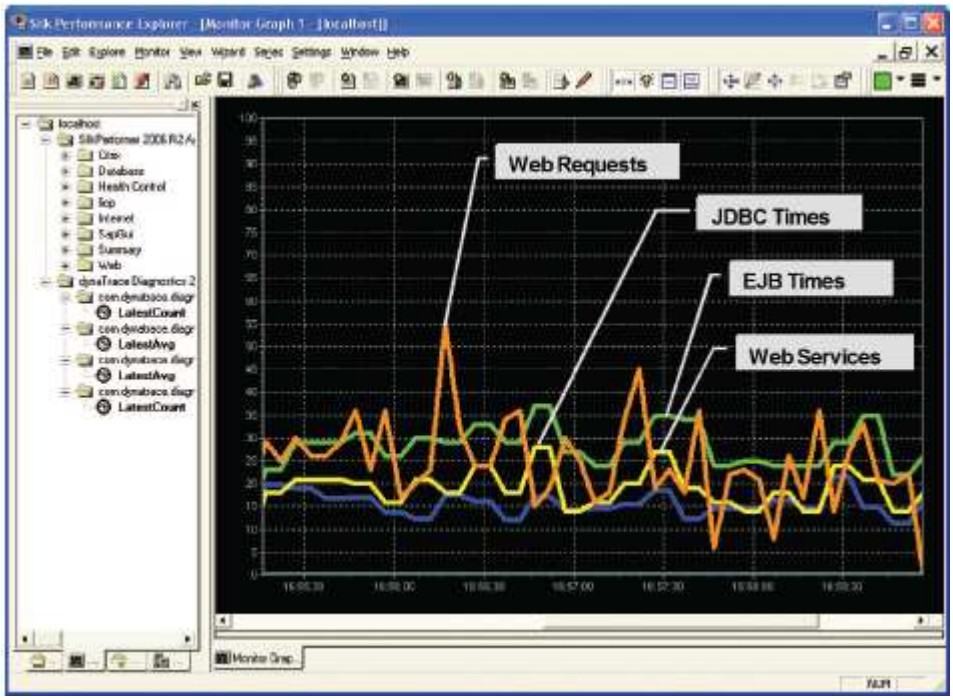
Efektywna diagnostyka wydajności to coś więcej niż narzędzia. W praktyce wymaga ona zintegrowanego rozwiązania umożliwiającego pełną komunikację i automatyzację procesu rozwiązywania problemów wydajnościowych dzięki zwiększeniu stopnia współpracy pomiędzy programistami, inżynierami jakości i architektami systemowymi.

Rozwiązanie Borland SilkPerfomer wraz ze znakomitym dynaTrace Diagnostics zapewnia zintegrowane środowisko, spełniające powyższe założenia. Inżynierowie odpowiedzialni za testy wydajnościowe używają narzędzia SilkPerformer w celu zdefiniowania szczegółowych transakcji użytkownika a następnie wykorzystaniu ich przy uruchomieniu testów obciążeniowych aplikacji. W tym samym czasie zachowanie systemu jest w pełni

monitorowane aż do poziomu komponentu/funkcji/kodu przez rozwiązanie dynaTrace Diagnostics przy prawie zerowym dodatkowym dociążeniu testowanie aplikacji.

Następnie metryki wydajnościowe aplikacji, takie jak czas reakcji, czas spędzony w różnych komponentach systemu, jak stos J2EE czy działania na bazie danych, są korelowane z aktualnym obciążeniem na serwerze. Te informacje mogą być monitorowane w czasie rzeczywistym podczas trwania testu obciążeniowego z wykorzystaniem Performance Explorera w SilkPerformerze tak, jak to widać na *Rysunku 3*. Takie podejście zapewnia nam otrzymanie wyczerpujących danych o zachowaniu się aplikacji. Performance Explorer odpowie nam na pytania:

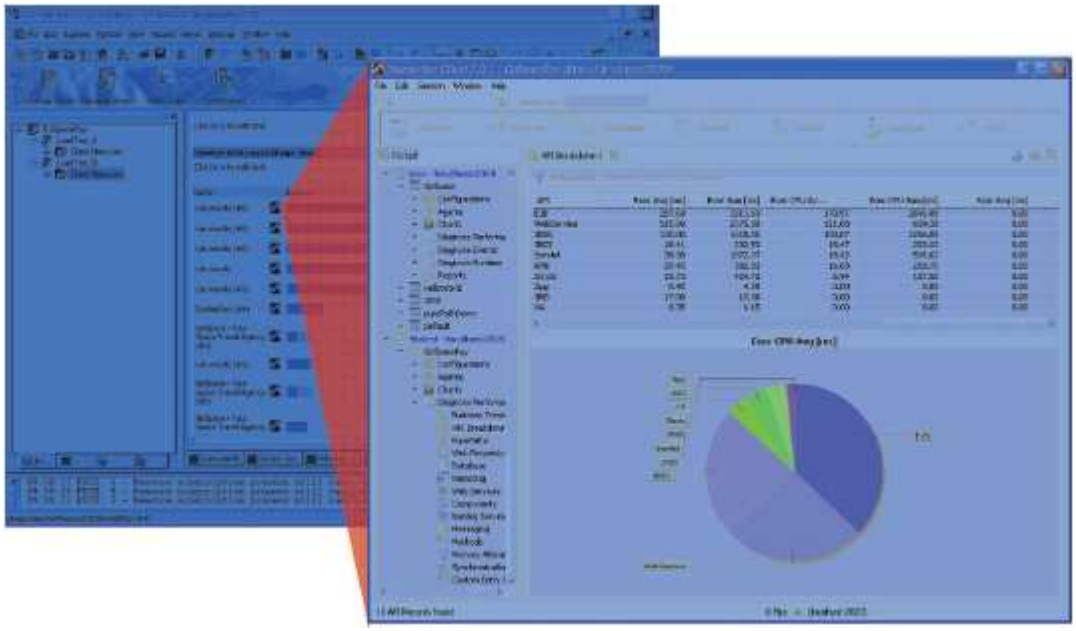
- Które z użytych w aplikacji komponentów zużywają najwięcej zasobów i czas?
- Czy wina za wolne działanie aplikacji leży po jej stronie, czy jest to problem z siecią?
- Jak zachowanie użytkownika wpływa na zarządzanie pamięcią?



Rysunek 3. Powiązanie Czasu Strony Internetowej z dogłębną analizą wydajnościową aplikacji w czasie rzeczywistym

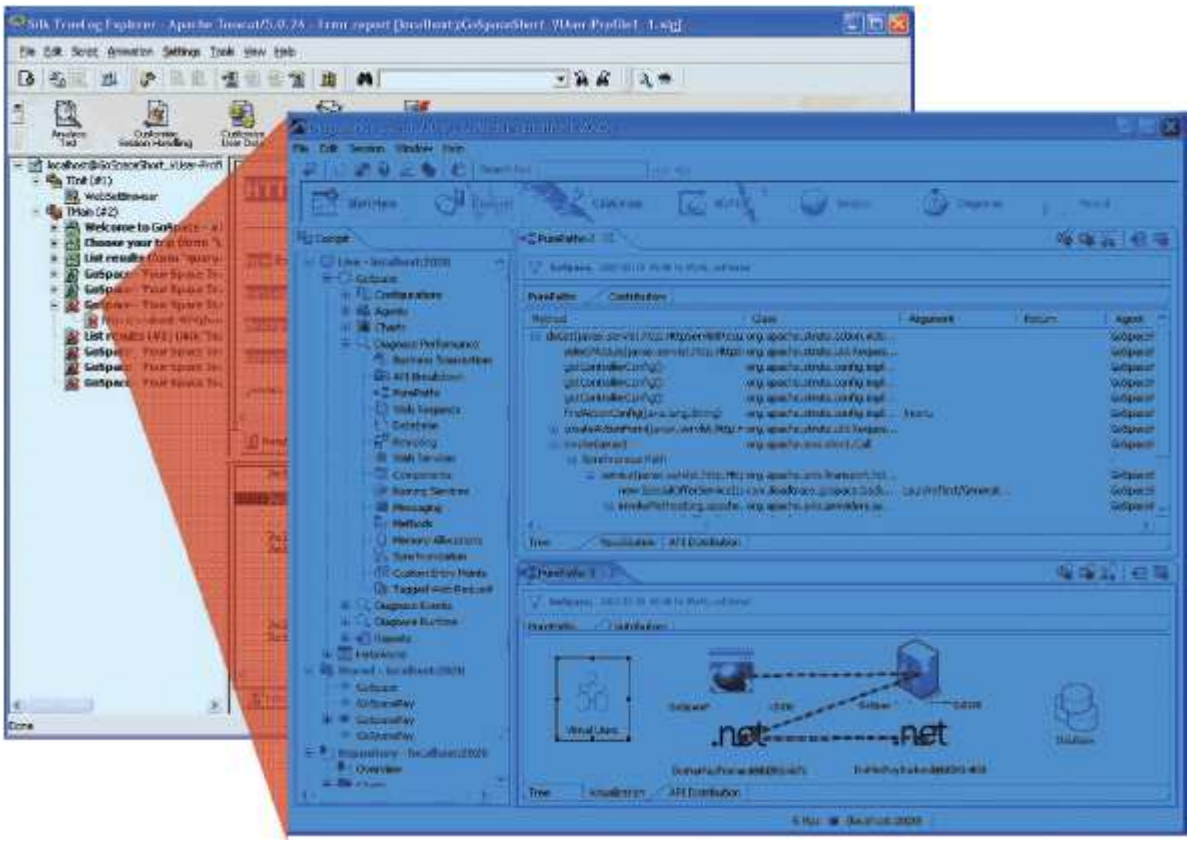
Po zakończeniu testu system raportowy umożliwia pełny wgląd we wszystkie dane zebrane podczas testowania, umożliwiając dogłębną analizę problemów wydajnościowych. W przypadku zintegrowanego rozwiązania, jakim jest SilkPerfomer wraz z dynaTrace

Diagnostics, dostajemy do dyspozycji pełen zestaw funkcjonalności pozwalający na prześledzenie każdego problemu obciążeniowego od jego skutku, aż w głąb do przyczyny, niezależnie od tego, czym ona jest: zbyt długi czas odpowiedzi, przeciążony procesor, zapchana pamięć czy jakakolwiek inna metryka zlokalizowana w różnych komponentach, np. servletach, EJB™ czy JDBC™. Jak pokazano na *Rysunku 4* przy użyciu informacji kontekstualnej (np. alokacja pamięci, problemy z synchronizacją, itp.) dalsza diagnostyka zostaje powiązana z dokładną przyczyną powstania konkretnego problemu, określając tym samym jego źródło.



Rysunek 4. Dogłębna analiza wyników testu pozwala na dotarcie do źródła problemu

W przypadkach, gdzie mamy do czynienia z problemem związanym ze specyficznymi warunkami (na przykład, brakujący serwis czy błędne dane zwrotne z konkretnej transakcji) potrzebna jest bardzo szczegółowa informacja o sposobie, w jaki dana transakcja jest obsługiwana. Ścieżka przepływu danych w takim przypadku może być prześledzona i zrekonstruowana dzięki technologii dynaTrace PurePath®, co zostało zilustrowane na *Rysunku 5*. dynaTrace Diagnostics nagrywa każde odwołanie do wspomnianej ścieżki wraz z pełną informacją kontekstową schwytaną podczas wykonywania testu obciążeniowego. W przeciwieństwie do innych narzędzi, „obecność" systemu diagnostycznego podczas testu w praktyce jest absolutnie przezroczyste i nie modyfikuje w żaden sposób zachowania testowanej aplikacji, zwiększając dokładność i trafność testu a zatem i diagnozy końcowej.



Rysunek 5. Analiza pojedynczej transakcji rozproszonej przy pomocy PurePath

Inne rozwiązania często oferują zbiorcze rezultaty, grupując wszystkie informacje z dostępnych transakcji razem, tym samym powodując, że późniejsza analiza jest dużo bardziej pracochłonna, a dokładne zlokalizowanie konkretnego źródła problemu utrudnione a czasem niemożliwe. To jedna z podstawowych zalet połączenia podejście SilkPerformera i dynaTrace Diagnostics. Dział Jakości jest w stanie odtworzyć każdą transakcję, która spowodowała powstanie problemu bez konieczności zgadywania, które z zarejestrowanych transakcji miały miejsce i z jakich danych korzystały. Dział programistów otrzymuje pełną informację, zawierającą wszystkie szczegóły niezbędne do zrekonstruowania problematycznej sytuacji na poziomie kodu. Dzięki wykorzystaniu technologii PurePath możliwe jest zidentyfikowanie takich źródeł problemów, jak nieszczelność pamięci, desynchronizacja, zła konfiguracja frameworków czy zapchanie kolejki wiadomości w zakresie pojedynczej transakcji w rozproszonych heterogenicznych środowiskach takich jak .NET i Java.

Korzyści biznesowe

Zintegrowane rozwiązanie SilkPerformer wraz z dynaTrace Diagnostics przynosi wymierne korzyści w procesie produkcji oprogramowania głównie ze względu na lepszy przepływ informacji i danych pomiędzy działami jakości i programistów. Zapobiega to powstawaniu kwestii spornych w fazie reprodukcji błędów wydajnościowych oraz sprzyja ich szybkiemu usuwaniu pozwalając unikać zbędnych przestojów i opóźnień.

Testerzy oraz inżynierowie odpowiedzialni za wydajność produktów mogą łatwo i dokładnie dokumentować każdy problem nawet z uwzględnieniem konkretnych linii kod, bez znaczenia czy jest to pojedyncza aplikacja, czy złożone, rozproszone środowisko. Jednocześnie nie muszą dysponować żadną wiedzą programistyczną ani mieć dostęp do faktycznego kodu oprogramowania, aby efektywnie zdiagnozować i przekazać usterki do usunięcia.

Dzięki temu zespoły zaangażowane w produkcję aplikacji, a w szczególności programiści, architekci systemowi i bazodanowi, stają się bardziej produktywni, poświęcając mniej czasu na usuwanie błędów wydajnościowych. Przestarzałe, tradycyjne podejście wymagające analizowania wielu logów, z porównywaniem czasów, aby odgadnąć, jakie zdarzenie mogło mieć wpływ na gorszą wydajność oprogramowania jest zbyt czasochłonne i często niewystarczające do poprawnego zdiagnozowania źródła usterki, szczególnie w przypadku rozwiązań klient-serwer. Korzystając ze zintegrowanych nowoczesnych narzędzi, programiści mogą dzisiaj zawczasu konfigurować dodatkowe systemy monitorujące przepływ informacji w aplikacji, ułatwiając jej gromadzenie w procesie testowania. Te konfiguracje mogą być następnie automatycznie wykorzystywane w środowisku testowym, co jeszcze skutecznie wypiera dawne metody rekonstrukcji powstawania problemów.

Podsumowując, SilkPerfomer wraz z dynaTrace Diagnostics dostarcza wymierne korzyści w postaci:

- Potwierdzenia wydajnego działania aplikacji przed jej uruchomieniem, redukując późniejsze „niespodzianki” oraz koszt ich naprawy
- Zmniejszenia do minimum czasu potrzebnego do wyizolowania problemu wydajnościowego

- Zredukowania czasu potrzebnego na spotkania i rozmowy mające na celu ustaleniu kto odpowiada za dany problem

- Przyspieszenia procesu rozwiązywania problemów obciążeniowych, dzięki dokładnym danym diagnostycznym, odnoszącym się do poszczególnych linii kodu

Pracując całkowicie nieinwazyjnie Borland SilkPerfomer wraz z dynaTrace Diagnostics znacząco usprawnia proces testowania aplikacji, umożliwiając działom jakości i programistom lepszą współpracę nad szybkim rozwiązaniem wszystkich problemów związanych z pracą oprogramowania pod obciążeniem.

Więcej informacji można znaleźć na **www.borland.pl**

Borland is the leading vendor of Open Application Lifecycle Management (ALM) solutions - open to customers' processes, tools and platforms – providing the flexibility to manage, measure and improve the software delivery process.

Borland
THE OPEN ALM COMPANY

Copyright © 2007 Borland Software Corporation. Borland and all other brand and product names are service marks, trademarks, or registered trademarks of Borland Software Corporation or its subsidiaries in the United States and other countries. All other marks are properties of their respective owners. 24479
www.borland.pl