Inprise Application Server verze 4

Je o co se opřít

V minulých dvou číslech jsme si vysvětlili příčiny současného rozmachu distribuovaných technologií a výhody sestavování aplikací z komponent. V následujícím článku se pro získání konkrétnější představy o možnostech middlewaru, který je dnes na trhu, seznámíme s jedním konkrétním řešením pro podporu tvorby distribuovaných aplikací – aplikačním serverem firmy Inprise (IAS).

Inprise Application Server verze 4 (IAS4) je dobrým příkladem prostředí pro podporu aplikací složených z komponent založeného na jazyce Java. Jako ukázku možností jsem jej z řady alternativních aplikačních serverů (např. IBM WebSphere nebo Oracle Application Server) zvolil proto, že firma Inprise důsledně respektuje formální i de facto standardy, a to v nejnovějších verzích, vyhýbá se řešením proprietárním a na místech, kde svá vlastní nestandardní řešení poskytuje, dodává vždy také alternativní řešení, formálně zcela odpovídající současné verzi příslušného standardu. A právě proto se jedná o softwarové prostředí ideálně vhodné ke tvorbě distribuovaných aplikací, jelikož podpora standardů je zárukou kompatibility se systémy jiných výrobců a nestaví jejich uživatele (v tomto případě návrháře architektury aplikace) do pozice závislosti na další strategii firmy poskytující infrastrukturu (middleware) softwarových komponent. Vyvinutá aplikace by totiž měla být beze změn přenositelná do libovolného jiného prostředí, podporujícího odpovídající verze standardu CORBA, Java či EJB.

Další příčinou, která vedal k volbě IAS jako ukázkového představitele této třídy produktů, je jeho dobrá dokumentace, která je v produktech firmy Borland již tradicí (vzpomeňme zde např. na příručku stařičkého Turbo Assembleru, která je dodnes použitelná jako základní učebnice assembleru 8086, nebo na dobrou učebnici objektového programování, kterou byla např. příručka Turbo C++ v prvních verzích). Srozumitelné, stručné, a přitom technicky správné vysvětlení principů systému CORBA a EJB, které manuály IAS podávají, je totiž s ohledem na doposud nepříliš velký výběr kvalitní literatury pro programátory nezbytnou podmínkou pro rychlý začátek při tvorbě distribuovaných aplikací.

Vnitřní architektura a podporované platformy

IAS je řešením charakteristickým použitím čisté Javy. To znamená, že je celý napsán v jazyce Java a jeho kód neobsahuje tzv. nativní metody specifické pro danou platformu. Je tedy alespoň teoreticky přenositelný na jakoukoli platformu podporující příslušnou verzi virtuálního stroje Java. V současné době oficiálně poskytuje firma Inprise IAS4 pro platformy Linux, Win32 (Windows 95/98/NT), Solaris, AIX a HP-UX.

Jádrem a komunikační vrstvou IAS4 je ORB VisiBroker for Java 4.0, opět založený na čisté Javě. Jedná se o jeden z nejkvalitnějších ORB, které jsou v současné době na trhu. Je charakteristický podporou všech rysů požadovaných specifikací CORBA 2.3, jako je zejména POA (Portable Object Adapter), přenositelné interceptory (Portable Interceptors) a RMI over IIOP. Již v perspektivě samotného využití tohoto ORB pro vývoj aplikací se orientace na IAS jeví jako zajímavá.

Technologicky IAS vychází ze specifikace J2EE, což je univerzálně koncipované prostředí navržené firmou Sun pro běh vícevrstvých aplikací založených na jazyku Java (filozofie ”Write Once, Run Everywhere"). Z nových vlastností, které architektura J2EE definuje, staví IAS4 zejména na integraci Javy se standardem CORBA, novém bezpečnostím modelu Javy a architektuře JDBC na platformě nezávislých ovladačů pro přístup k databázím. Graficky je architektura IAS znázorněna na obr.1.

Jak to funguje dohromady

Z pohledu spolupráce distribuovaných komponent a přístupu k nim poskytuje IAS tři možné metody, které se vzájemně doplňují a prolínají. Je jimi technologie CORBA, vhodná zejména pro heterogenní řešení založené na spolupráci komponent v nejrůznějších programovacích jazycích, technologie Enterprise JavaBeans z dílny Sun Microsystems, využitelná v prostředí Java, a konečně možnost integrace do prostředí WWW, jelikož WWW server podporující Javu je nedílnou součástí aplikačního serveru IAS.

IAS a CORBA

Se základními principy a určením standardu Common Object Request Broker Architecture (CORBA) konsorcia Object Management Group (OMG) jsme se již seznámili v článku v předchozím čísle. IAS se na CORBA napojuje prostřednictvím ORB VisiBroker. Komponenty (objekty) CORBA tak mohou být provozovány na IAS jako serverové aplikace vytvořené v Javě a poskytující služby klientům CORBA, kromě toho však tyto komponenty mohou využívat služeb libovolných jiných objektů CORBA rozmístěných v počítačové síti (ať již jsou provozovány jako aplikace běžných počítačů, či jsou integrovány do libovolných jiných elektronických zařízení připojených k síti, kupř. k internetu).

Díky použití VisiBrokeru verze 4 mohou programátoři komponent CORBA využívat veškeré prvky nejnovější verze standardu CORBA 2.3.

IAS a Enterprise JavaBeans (EJB)

Enterprise JavaBeans je standard navržený firmou Sun Microsystems v rámci technologie Java pro tvorbu spolupracujících distribuovaných komponent. Komponenta je zde nazývána komponentou Enterprise JavaBean. V současné verzi standardu (1.1) jsou definovány tři typy takovýchto komponent: Stateless a Stateful Session Beans a Entity Beans. Obě verze Session Bean mají reprezentovat komponentu schopnou poskytovat služby, zatímco Entity Beans reprezentují určitou perzistentní entitu systému (např. řádek v databázi).

Komponenta EJB je určena pro běh v určitém, přesně definovaném prostředí. Takovémuto prostředí se říká kontejner. V jednom kontejneru může být umístěna jedna nebo více EJB komponent. Kontejner napojuje komponenty na ostatní části distribuovaného systému a poskytuje jim systémové služby (např transakční nebo bezpečnostní službu). Konkrétní implementace příslušných systémových služeb tak není starostí samotného beanu, ale kontejneru, který je příslušným způsobem uživatelsky nakonfigurován. Může být například stanoveno, do které databáze se má implicitně ukládat vnitřní stav jednotlivých komponent při jejich deaktivaci, aby mohl být při opětovné aktivaci znovu obnoven. U některých služeb je jejich využívání zcela řízeno kontejnerem, u jiných může programátor komponenty její implicitní chování modifikovat (např. u transakční služby).

Přenechání implementace podpůrných služeb kontejneru aplikačního serveru má dvě výhody. Zaprvé se programátor nemusí starat o (z technického pohledu značně komplikované) činnosti, které mu implicitně poskytne aplikační server. Zadruhé je možné mnohem obecnější používání komponent, jelikož konkrétní chování komponenty v daném systému již neurčuje jen její programátor, ale správce aplikace, který komponentu do systému začleňuje. Před začleněním komponenty popíše správce nejprve očekávané chování komponenty vzhledem k systémovým službám pomocí tzv. deployment descriptoru. Tento deployment descriptor pak předá příslušnému kontejneru aplikačního serveru spolu s komponentou jako konfigurační informaci.

Na IAS může být současně spuštěno více kontejnerů, z nichž každý lze nakonfigurovat jiným způsobem. Komponenty jsou pak vkládány do kontejneru nakonfigurovaného tak, jak to vyhovuje jejich bezpečnostním a transakčním charakteristikám a požadavkům na přístup k datovým zdrojům.

AS a WWW

Pro účely zpřístupnění aplikací bežících na aplikačním serveru prostřednictvím WWW stránek je jako součást aplikačního serveru začleněn WWW server. Jedná se o WWW server, jehož možnosti jsou srovnatelné např. se serverem JavaServer firmy Sun Microsystems nebo s množinou nejpoužívanějších funkcí poskytovanou oblíbeným serverem Apache. Správu WWW serveru je možné provádět na dálku prostřednictvím WWW prohlížeče. WWW server podporuje provoz přes SSL (Secure Socket Layer, bezpečnostní vrstva, zajišťující kryptování provozu), na dobré úrovni je také možnost zabezpečení přístupu k jednotlivým WWW stránkám. Sympatická je možnost sledování všestranných statistik serveru.

Z našeho pohledu je zajímavá podpora pro technologie servletů a JSP (Java Servlet Pages), které umožňují integrovat komponenty EJB i CORBA do prostředí WWW.

Aplikace jazyka Java jsou do WWW serveru začleněny dnes již dosti známým a vžitým způsobem – pomocí tzv. servletů. Servlety jsou třídy v Javě s definovaným rozhraním a jsou spouštěny v JVM běžící jako součást WWW serveru. Servlety jsou (podobně jako CGI skripty) spouštěny WWW serverem jako reakce na požadavek na určitý URL. Kód HTML zaslaný prohlížeči je pak generován programově příslušným servletem. Servlety tak tvoří základ pro tvorbu

dynamických WWW stránek prostřednictvím aplikací v Javě na straně serveru.

Pro integraci komponent zapouzdřených podle standardu JavaBean podporuje WWW server další z technologií firmy Sun: Java Server Pages (JSP). JSP může být využívána dvěma způsoby, z nichž první je bližší klasickému programování, zatímco druhý přístup je spíše uživatelský a nevyžaduje prakticky žádnou znalost programátorských technik. Programátorský přístup je založen na vkládání kódu přímo do WWW stránek a na jeho interpretaci na straně WWW serveru (podobně jako např. u technologií PHP). Ze skriptu je možné také využívat komponenty JavaBeans. Důležité je poznamenat, že výsledný kód HTML, který bude na základě běhu skriptu zaslán prohlížeči, je vytvářen v rámci tohoto skriptu programově. Pro urychlení generování stránek pro prohlížeč je zdrojový kód v Javě, uvedený na stránce při prvním přístupu, nejprve zkompilován do bytecodu v podobě servletu a jako běžný servlet je pak dále používán.

Druhý přístup ocení zejména návrháři WWW stránek, jejichž specializací je spíše grafický návrh a kteří se programátorskými detaily nechtějí zabývat. Ti mohou jednoduchým příkazem v HTML kódu spustit zadanou metodu určité komponenty a výsledky jejího běhu vložit pomocí speciálních značek na požadované místo HTML stránky. Princip tohoto mechanismu spočívá v tom, že komponenta po spuštění vygeneruje výsledky ve tvaru pole položek < jméno, hodnota > a ty při zpracovávání příslušné značky HTML pro vložení hodnoty určitého jména vloží do dynamicky generované stránky.

K dispozici je samozřejmě i podpora klasického CGI, avšak využití servletů je výrazně zajímavější jak z hlediska výsledného výkonu, tak bezpečnosti a v neposlední řadě i z hlediska rychlosti tvorby serverových aplikací.

Vzájemné propojení diskutovaných architektur a jejich využití z klientských aplikací (v tomto případě třívrstvé architektury) je patrné z obr. 2.

Administrace

Administrace IAS je prováděna pomocí příjemné grafické aplikace v Javě (Inprise Application Server Console). Je zde možné monitorovat a ovládat běh jednotlivých komponent aplikačního serveru: kontejnerů EJB, jmenné služby JNDI, transakční služby a registrovaných CORBA serverů. Jsou zde také k dispozici funkce pro ovládání a sledování činnosti ORB VisiBroker a jeho podpůrných komponent (Interface Repository, Implementation Repository, Location Service atd.). Konfigurovat a administrovat WWW server lze pomocí libovolného prohlížeče WWW podporujícího applety. Systém správy WWW serveru je přehledný a příjemný, integrace správy IAS a jeho WWW serveru do jednoho prostředí by však byla uživatelsky jistě příjemnější.

Jelikož jsou komponenty distribuované aplikace často rozmisťovány na řadu spolupracujících aplikačních serverů, pamatuje konzola IAS i na možnost správy klastru aplikačních serverů společně z jednoho prostředí.

Vzhled prostředí administrační konzoly ukazuje obr.3.

Bezpečnost aplikací

Architektura komponent EJB implementuje svůj vlastní bezpečnostní model založený na principů rolí, které stanovují jednotlivým uživatelům práva pracovat s jednotlivými instancemi EJB komponent a volat jejich metody. Navíc je možné jako zvláštní část dokoupit k IAS mnohem propracovanější bezpečnostní službu, která může být používána jak v prostředí kontejnerů EJB, tak v návaznosti na standardizovanou službu COS Security systému CORBA. Zde je k dispozici vrstva SSL, vzájemná autentizace klientů a serverů certifikáty X.509, návaznost systému uživatelů a jejich hesel na systémy NIS (Network Information System) a mechanismus domén v prostředí Windows NT.

Zajímavá je podpora IAS4 pro běh v prostředí internetu, kde je spolupráce aplikací omezována bezpečnostní politikou firewallů. Jako součást IAS je totiž dodávána aplikace Gatekeeper, což je v podstatě GIOP gateway mezi klienty a servery určená pro překonání firewallů a omezení ukládaných WWW prohlížeči na činnosti dovolené appletům. K dispozici je podpora tunelování protokolu GIOP (komunikační protokol CORBA) v protokolu HTTP (ve standardu CORBA označované jako HIOP) a podpora obvykle problematického provozu protokolu GIOP v prostředí s překladem adres (NAT). GateKeeper tak činí z IAS prostředek pro provozování bezpečných aplikací nejen ve firemních intranetech, ale i v prostředí dnešního internetu, plného bezpečnostních rizik a komplikovaných mechanismů, které mají tato rizika omezovat.

Petr Grygárek, http://www.cs.vsb.cz/grygarek