

**ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA
V PRAZE
PROVOZNĚ EKONOMICKÁ FAKULTA**



Bakalářská práce

TÉMA:

**IBM WebSphere Application Server -
ND V6**

AUTOR:

**JAKUB ŠTĚPÁNEK,
STUDENT OBORU SYSTÉMOVÉ INŽENÝRSTVÍ NA
PEF ČZU V PRAZE**

VEDOUcí PRÁCE

**ING. PAVEL ŠIMEK,
KATEDRA INFORMAČNÍCH TECHNOLOGIÍ NA
PEF ČZU V PRAZE**

V PRAZE DNE 25. 6. 2007

PROHLÁŠENÍ

Prohlašuji, že jsem svou bakalářskou práci na téma „**IBM WebSphere Application Server – ND V6**“ vypracoval samostatně za použití uvedené literatury a po odborných konzultacích s panem Ing. Pavlem Šimkem.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

PODĚKOVÁNÍ

Děkuji tímto panu **Ing. Pavlu Šimkovi** za odborné vedení a rady při zpracování bakalářské práce na téma „**WebSphere Application Server – ND V6**“.

Zároveň děkuji firmě **IBM Česká Republika spol. s r.o.** a jejím zaměstnancům za ochotu při poskytování potřebných podkladů nutných pro vzniku bakalářské práce.

Obsah:

1. Úvod	2
2. Cíl práce a metodika.....	3
3. Základní informace o WebSphere softwaru	4
3.1 Co je to WebSphere?.....	5
3.2 Technologie J2EE.....	6
3.2.1 Klíčové charakteristiky a služby J2EE.....	7
3.3 Aplikační server.....	9
3.3.1 IBM WAS – Marketingový pohled.....	10
3.3.2 Evoluce aplikačního serveru IBM.....	11
3.3.3 WAS V6.1 podrobnější pohled	14
3.3.4 Architektura serveru WAS - jednoduchá	17
3.3.5 Architektura serveru WAS – federovaná	19
3.3.6 Správce konfigurací – Deployment Manager	20
4. Administrace WAS.....	23
4.1 Možné instalační scénáře	23
4.2 Použité prostředky pro praktickou část	24
4.3 Instalace a konfigurace.....	26
4.3.1 Předinstalační úpravy OS	27
4.3.2 Instalace BD2.....	27
4.3.3 Instalace WebSphere Application Server – ND V6.1	28
4.3.4 Instalace IHS.....	32
5. Praktické použití administrační konzole WAS.....	34
5.1 Instalace Trade aplikace	34
5.2 Vytvoření clusteru.....	36
5.2.1 Clusteru.....	38
5.2.2 Demonstrace clusteru	40
6. Závěr	41
7. Zdroje.....	42
8. Přílohy.....	43
8.1 Seznam obázků	43
8.2 Použité zkratky	43

WebSphere Application Server – ND V6

Klíčová slova:

IBM, WebSphere, server, aplikační server, webový server, Internet, cluster, aplikace, relační databáze, počítačová síť, třívrstvá architektura

Souhrn

Hlavním cílem bakalářské práce je vytvoření a následná demonstrace clusterového řešení na WebSphere platformě společnosti IBM.

Teoretická část pojednává především o samotném představení obchodní značky WebSphere a produktů spadajících pod ni. Dále je podrobněji popisován aplikační server a použité technologie, vývoj verzí a další skutečnosti týkající se dané problematiky.

Praktická část demonstruje realizaci a použití clusterového řešení při pádu jednoho z aplikačních serverů. V praktické části se autor mimo jiné zaměřuje na praktické ukázky použití administrační webové konzole.

Summary

A main goal of the baccalaureate work is creating and following demonstration of a cluster solution with the aid of an IBM WebSphere platform.

A theoretical part mostly describes a business brand name of IBM WebSphere and its products. Furthermore, an application server, used technologies, an evolution of application server and more fact about this given theme are described.

A practical part demonstrates a realization and a using of the cluster solution in a case of failing of an application server. In the practical part the writer also focuses on a practical using of a web administration console.

1. Úvod

V dnešní přetechnizované společnosti jsou lidé stále více závislí na Internetu a jeho službách. Mnoho firem na celém světě využívá ke své činnosti služeb Internetu převážně jako marketingový nástroj nebo jako další odbytové místo jejich komodit. Jiní na Internetu staví svůj business, například společnost Google či Seznam.cz jsou ti, kteří by bez sítě Internet nemohly existovat. Mezi další uživatele bezesporu patří státní správa, která prostřednictvím Internetu nabízí různé služby svým občanům. Díky tomu lidé nejsou nuceni trávit hodiny v přeplněných čekárnách na úřadech a mohou stále více úkonů provádět z pohodlí domova či kanceláře.

Internet zasahuje výraznou měrou i do školství na všech úrovních (základního, středoškolského i vysokoškolského). Dnes by se student vysoké školy bez Internetu ztěžilo dokázat obejít a to z jednoho prostého důvodu, všechny studijní procesy, jako přihlašování na zkoušky a rozvrh, zapisování na bakalářskou či diplomovou práci jsou realizovány právě prostřednictvím internetových aplikací. Zajisté student nevyužívá Internet jen ke studijním účelům, ale také pro získávání informací a komunikaci s okolním světem.

Síť Internet lze považovat za nepřebornou studnu informací, která se stala pro počítačově gramotné uživatele nejdostupnějším zdrojem, využívaným ve všech oborech lidské činnosti.

Internet nabízí mnoho služeb, mezi kterými je asi nejznámější služba webových stránek World Wide Web (dále jen www).

2. Cíl práce a metodika

Cílem bakalářské práce je seznámení s produktem WebSphere Application (dále jen WAS) společnosti IBM. Především se bude zaměřovat na jeho administraci, která se dotkne samotné instalace na operační systém Windows Server 2003, základní nastavení serveru, nainstalování ukázkové aplikace a následné vytvoření clusteru aplikačního serveru. Poslední část textu se zabývá využitím clusteru při pádu jednoho ze serverů.

Faktem je, že k tomuto tématu není téměř žádný český literární zdroj. To je hlavní důvod, proč text bakalářské práce bude vznikat na základě anglických zdrojů – manuálů nazývaných Redbooks nebo z Informačního Centra IBM a samozřejmě ze základní literatury uvedené v seznamu. Dále bude čerpáno z interních a školících materiálů společnosti IBM poskytnutých studentovi k prostudování.

Teoretická část je tvořena na základě volně dostupných ať už internetových nebo tištěných zdrojů, které autor začal studovat již v rámci bakalářské praxe.

Kapitola „Základní informace o WebSphere“ představí produktovou značku a její oblasti, základní technologie použité v aplikačním serveru a v neposlední řadě aplikační server jako takový. Ke konci kapitoly jsou popsány hlavní charakteristiky aplikačního serveru a možné způsoby jeho implementace.

Praktická část textu je založena z větší části na základní literatuře a školících materiálech společnosti IBM. Při realizaci se vyskytly určité problémy, které byly vyřešeny až po konzultacích s odborníky z podniku IBM Česká republika, spol. s r.o.. Realizace praktické části trvala zhruba čtyři a půl týdne počínaje přípravou síťové topologie, virtuálních PC, instalací potřebných softwarových komponent IBM, konfigurací a konče doladěním a otestováním funkcionality clusteru. Praktické realizaci předcházela mnoha týdenní studia manuálů a jiných zdrojů nezbytných k pochopení problematiky. Jednotlivé kapitoly a podkapitoly praktické části (4. a 5. kapitoly) na sebe logicky navazují a provázejí tak jednotlivými oblastmi kroků (v omezeném rozsahu – není možné obsáhnout celou konfiguraci do textu bakalářské práce krok po kroku) vedoucími k vytvoření aplikačního clusteru na produktu WebSphere Application Server ve verzi 6.1 Network Deployment.

3. Základní informace o WebSphere softwaru

Éra prvních elektronických dokumentů na síti Internet začala statickými webovými stránkami (dále jen web), které obsahovaly textové informace v takzvané hypertextové formě. Hypertext byl navrhnut v počáteční podobě Timem Berners-Lee roku 1989. Základní vlastnost hypertextu spočívá v možnosti přeskokování kontextu v souladu s logickým uvažováním člověka – tzv. „přeskakování z myšlenky na myšlenku“. Postupem času se webová služba začala nezastavitelně vyvíjet. V současné době je možné na síti Internetu nalézt stránky s různým stupněm složitosti a dynamičnosti. Podle dynamičnosti lze Internetové stránky dělit na statické, statické stránky s částí dynamiky, dynamické až komplexní webové aplikace – dnes často využívané v komerční či veřejné sféře.

Internet se proto stal základem k podnikání, hovoříme-li o obchodních organizacích; v případě veřejné správy a organizací do ní náležejících je informativním kanálem pro občany.

Fungování internetových řešení se neobejde bez softwarových a hardwarových prostředků. Tato práce se dotkne různých softwarových prostředků Internetu, ale hlavně se zaměří na základ dnešních dynamických webových technologií na tzv. aplikační servery.

Před dalším výkladem je nutné uvést, na jaké architektuře stojí zmiňovaná webová služba www. Její prvotní podoba vycházela z koncepce dvouvrstvé architektury klient/server. Na jedné straně klient, reprezentovaný internetovým prohlížečem a na straně druhé HTTP (Hyper Text Transfer Protocol) server, starající se o zpracovávání a zaslání zpracovaných požadavků přicházejících od klienta.

Současný trend přináší (rozšířenou) třívrstvou architekturu složenou ze tří základních komponent (aplikační server / HTTP server / klient), která zajišťuje lepší bezpečnost, širší škálovatelnost a vyšší výkon celého systému. Na nejnižším místě v hierarchii architektury leží aplikační server (např. Jboss, Tomcat, Weblogic, IBM WebSphere Application Server), jenž se stará o zpracování dynamických Java, J2EE (Java 2 Enterprise Edition) a jiných dynamických aplikací. Nad aplikačním serverem stojí HTTP server. Ten zde sehrává funkci převážně bezpečnostní

a výkonnostní. Pokud by byl vystaven aplikační server přímému přístupu z Internetu, mohlo by snadno dojít k jeho napadení, a tím by byly ohroženy zdrojové kódy a jiná důvěrná data jako hesla či přístupové cesty k databázím a jiným zdrojům.

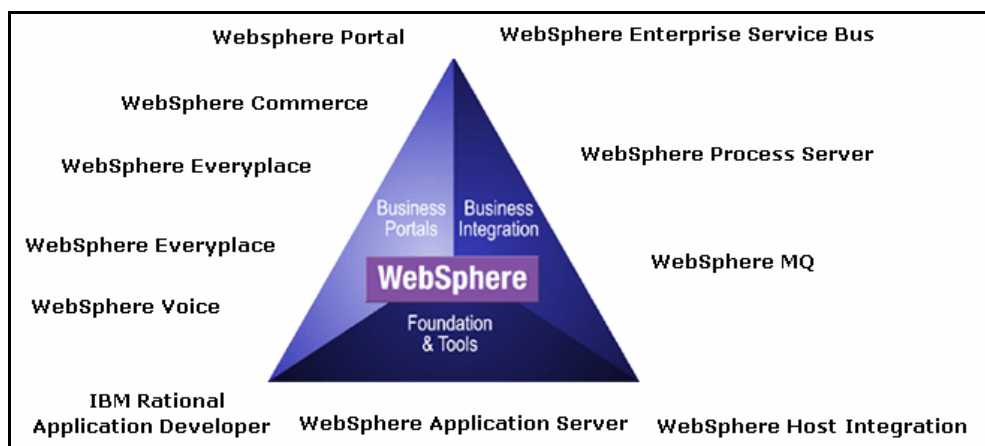
V třívrstvé architektuře slouží HTTP server také jako transportní a vyrovnávací prvek, který již částečně leží mimo demilitarizovanou zónu (dále jen DMZ) na rozdíl od aplikačního serveru. Většina dnešních HTTP serverů je vybavena cache komponentou. Ta umožňuje ukládání webového obsahu do schránky (ENG: cache) pro rychlé reagování na požadavky přicházející od klienta/webového prohlížeče.

Poslední a třetí komponenta v třívrstvé architektuře zůstává totožná jako u dvouvrstvé – klient reprezentovaný webovým prohlížečem (ENG: Web Browser).

3.1 Co je to WebSphere?

Odpověď by mohla znít následovně:

WebSphere je jak druhová (ENG: brand) značka, tak technologie společnosti IBM. Pokrývá širokou řadu technologií, které dohromady dávají různé obchodní řešení. WebSphere produkty jsou rozděleny do několika oblastí, které se zaměřují na aplikační servery, portálové produkty a spolupráci, bezdrátové/hlasové a mobilní zařízení, elektronické obchodování (ENG: E-commerce), integraci podnikových systémů, síťové technologie a v neposlední řadě vývoj softwaru.



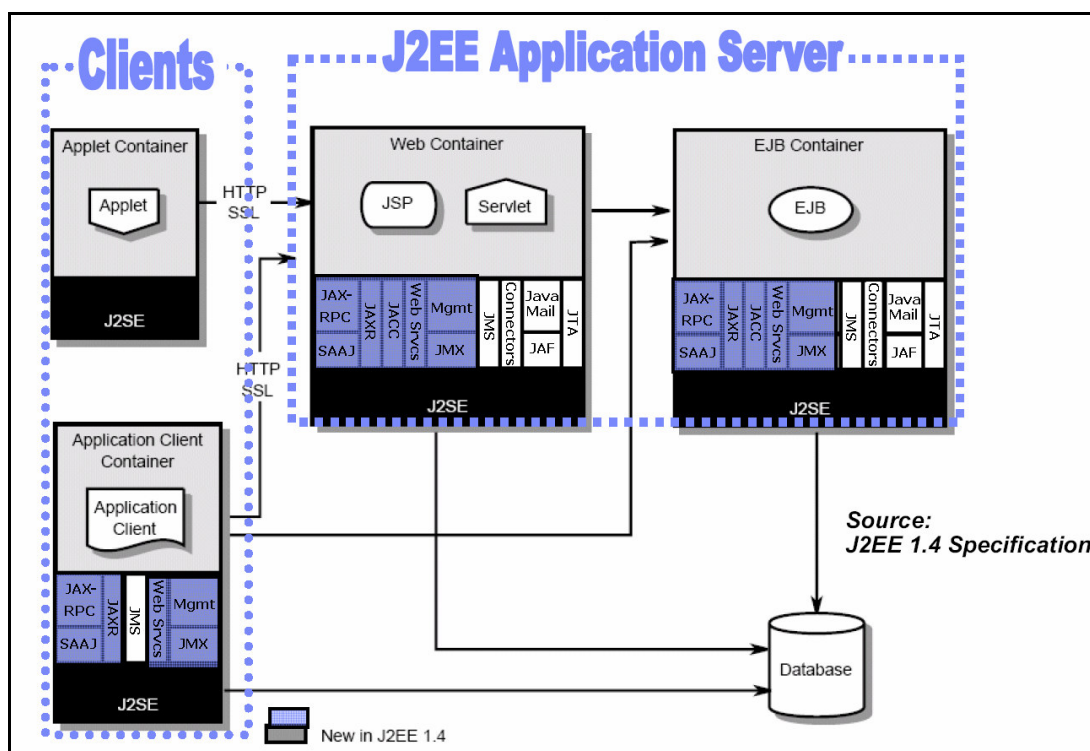
Obr. č. 1 – Základní produktová mapa IBM WebSphere ^[VI]

3.2 Technologie J2EE

Zkratka vznikla z anglických slov **J**ava **2** Platform **E**nterprise **E**dition. J2EE je platformě nezávislé Java prostředí od společnosti Sun spolupracující s IBM, Oracle, HP a dalšími. Umožňuje vývoj, budování a poskytování webově založených on-line podnikových aplikací. J2EE se skládá ze souboru služeb, API (Application Programming Interface) a protokolů poskytujících nezbytnou funkcionalitu pro webové aplikace.^[VI]

Aplikace vyvíjené pro aplikační servery/portály mají zpravidla třívrstvou architekturu. Spodní vrstva představuje databázový systém, prostřední vrstva aplikační logiku a horní, tzv. prezentační vrstva tvoří uživatelské rozhraní. Velikou výhodou této aplikační architektury je, že aplikace dosahuje modulárnosti, tj. možnosti opětovného použití komponenty. Jelikož je architektura rozdělena do tří vrstev, vývojáři se mohou specializovat pouze na jednu z nich a nemusí se zabývat a zatěžovat ostatními vrstvami.^[VI]

Vícevrstvou architekturu využívá kromě standardní J2SE, i právě zde popisovaná J2EE. Jednotlivé vrstvy a vazby mezi nimi znázorňuje následující obrázek číslo dvě.



Obr. č. 2 – Vícevrstvá architektura J2SE^[VI]

3.2.1 Klíčové charakteristiky a služby J2EE

Jak bylo výše uvedeno J2EE tvoří jakýsi soubor různých navzájem se doplňujících technologií, které jsou převážně určeny pro vývoj a provoz serverových aplikačních systémů. Mezi hlavní technologie lze zařadit následující:

JSP (JavaServer Pages)

JSP slouží pro generování dynamického obsahu webového prohlížeče jak v počítačích, tak v mobilních zařízeních jako PDA (Personál Data Assistant) či MDA (Mobile Data Assistant). Nejčastěji se JSP využívá pro generování HTML kódu na základě dotazů uživatelů. Používá se v technologii *server-side scripting* (kód interpretovaný na straně serveru). JSP doplňuje architekturu Java servletů. Poskytuje kontejner, což je „prostředí“ na straně serveru, ve kterém se zpracovává daný skript. Kontejner dále zajišťuje správu JSP stránek a jejich překlad na servlety. JSP stránka se podobá stránce ve značkovacím jazyku HTML (HyperText Markup Language), obsahuje však zvláštní syntaktické elementy, díky kterým může stránka dynamicky měnit svůj obsah.^[V]

JSF (JavaServer Face)

JSF pravděpodobně vznikla reakcí na novou technologii .NET (Network Technology) společnosti Microsoft. JSF je mladá technologie, která je zatím ve fázi specifikačního procesu. Jde o komponenty generující uživatelské rozhraní na straně serveru bez ohledu na výsledný protokol, což velice ulehčuje a zjednodušuje vývoj aplikací.^[V]

Servlety

Java Servlety představují velmi širokou multiplatformní technologii pro rozšíření funkcionality webových serverů. Webové nebo Java aplikační servery většinou umožňují servlet kontejner začlenit jako vlákno do hlavního procesu. Populární webové servery jako *IIS*, *Apache* či *iPlanet* vyžadují diferencované technologie v závislosti na jejich vlastním rozhraní a použité technologii. Součástí architektury servletů je také zásuvný modul webového serveru, který umožňuje přeposlání požadavků servletu do některého odděleného procesu Javy, jenž je implementací servletu kontejneru. Komunikace servlet kontejneru se zásuvným modulem probíhá

tak, že modul je nastaven k naslouchání na určitém síťovém portu, do kterého jsou mapovány přesměrované požadavky. ^[V]

EJB (Enterprise JavaBeans)

EJB je potřeba považovat za hlavní součást J2EE. Kolem nich se soustřeďuje celý aplikační vývoj.

Enterprise Java Beans – EJB jsou komponenty pracující v prostředí EJB kontejneru, které implementují části aplikační logiky. Protože některé aplikace mohou být využívány různými klienty (webový prohlížeč nebo jiná aplikace komunikující přes rozhraní webových služeb), umožňují tedy EJB i jednoduché opakované použití aplikační logiky na více místech. EJB je dnes ve verzi 3.0. Specifikace EJB definuje vlastnosti, rozhraní EJB kontejneru a tři následujících typů EJB komponent: ^[V]

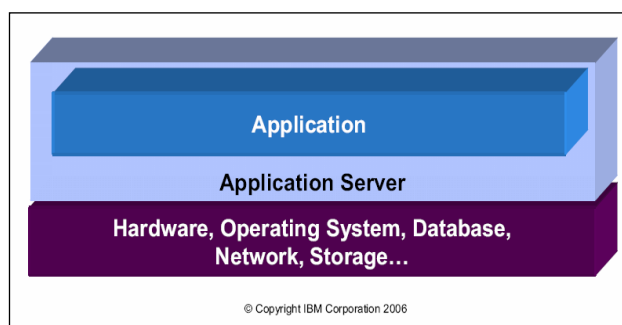
- Session Bean – typicky obsahuje tzv. Business Logic, zpravidla vykonává požadavky klienta ^[V]
- Entity Bean – je persistentní objekt, který reprezentuje určitá data v databázi, zpravidla záznam v tabulce. ^[V]
- Message-Driven Bean – má stejnou úlohu jako Session Bean, je však na rozdíl od něj asynchronní a lze s ním komunikovat výhradně přes rozhraní JMS. ^[V]

Z praktického hlediska je EJB komponenta soubor ve formátu JAR – *Java Archive*, který obsahuje kód komponenty a XML (eXtensible Markup Language) popisovače definující obsah archivu (může obsahovat více EJB komponent) nastavení případných parametrů a informace o napojení EJB na okolí. Tento popisovač se nazývá Deployment Deskriptor a má podobu jednoho nebo více XML souborů. Archiv může obsahovat i potřebné knihovny v podobě JAR archivů a další soubory. Přesný popis XML popisovačů je definován v odpovídající verzi specifikace EJB.

Poznámka: V případě WAS jsou jednotlivé Deployment Deskriptory dostupné pro náhled z webové administrační konzole.

3.3 Aplikační server

Aplikační server tvoří střední vrstvu (ENG: middle tier) takzvané vícevrstvé architektury. Jejím předchůdcem je dvouvrstvá architektura, nazývaná také architekturou klient/server, jak již bylo v úvodu řečeno.^[VII]



Obr. č. 3 – Poloha aplikačního serveru ^[III]

Současným trendem ve vývoji aplikací je odlehčit koncovým zařízením (PC, PDA, mobilním telefonům, atd.) od aplikační logiky a plně využívat komfortu vícevrstvé architektury. Její výhodu ocení především správci podnikových systémů. Centralizovaná údržba, správa a monitoring, snadná distribuce nových verzí aplikací a zabezpečení - to jsou jedny z argumentů hovořících pro aplikační server. Většina dnešních aplikačních serverů disponuje škálovatelností a možností nasazení na významných platformách.^[VII]

V případě architektury klient/server je celá nebo majoritní část aplikační logiky nainstalována a provozována na klientském počítači a serverovou částí je databáze.^[VII]

U vícevrstvé architektury na sebe přebírá veškerou aplikační logiku software provozovaný právě na aplikačním serveru. Klient může být vybaven pouze webovým prohlížečem nebo klientskou aplikací komunikující se serverovou částí na aplikačním serveru. Tato komunikace pak funguje na jednoduchém principu požadavek – odpověď. Aplikační server pak kromě výkonných funkcí business logiky provádí komunikaci s databázovými servery.^[VII]

Aplikační server tedy poskytuje infrastrukturu pro běh nejen podnikových aplikací podporujících obchod (ve smyslu business), ale i pro mnoho různých aplikací

využívaných v celém spektru lidského působení. Infrastruktura WebSphere Application Server přináší nezávislost mezi aplikací a prostředím, ve kterém běží (hardware, databáze operační a jiné využívané systémy). To velice usnadňuje práci vývojářům, kteří při své práci nemusí brát v potaz hardware či operační systém a mohou se zaměřovat na vývoj nových obchodních logik/aplikací, které pro podnik ve finále znamenají konkurenční výhodu!

3.3.1 IBM WAS – Marketingový pohled

Software, díky kterému mohou firmy stavět flexibilní, otevřené, na standardech založené informační infrastruktury umožňující rychle reagovat na nové podmínky a příležitosti trhu. ^[IV]

Produkt pomáhá zkrátit čas a snížit náklady na vývoj a rozvoj e-business aplikací, podporuje vysoce výkonná škálovatelná prostředí, zajišťuje flexibilitu podporou mnoha platforem a konfigurací. Jeho technologie chrání internetové podnikové aplikace před výpadky, které mohou podniky stát nesmírné množství finančních prostředků za minutu ve ztracených službách a produktivitě. ^[IV]

Aplikační server se vyskytuje ve třech základních provedeních: WebSphere Application Server Express, WebSphere Application Server a WebSphere Application Server Network Deployment. Tato vysoce výkonná aplikační platforma založená na technologii J2EE a webových službách s možností vysoké škálovatelnosti transakčního stroje pro dynamické e-business aplikace přináší navíc v edici Network Deployment další zvýšení výkonnosti a disponibilní kapacity pro podporu dynamického aplikačního prostředí, vylepšení webových služeb, pokročilé clusterování, rozhraní síťových služeb a vysokou dostupnost distribuovaných konfigurací. Rozšíření o další vlastnosti a funkcionalitu poskytuje WebSphere Extended Deployment, který přidává dynamické, cíleně řízené, vysoce výkonné prostředí pro WebSphere aplikace. Tyto rozšířené možnosti pomáhají optimalizovat využití a řízení rozložení výkonu a zkvalitnit služby provozních aplikací. ^[IV]

Výhody:

- rychlý vývoj, implementace a nasazení
- snadné přizpůsobení, podpora velkého množství platforem
- nízké provozní a celkové náklady
- rychlá návratnost, vysoká obchodní flexibilita, redukce cen a času
- bezpečnost, dynamičnost, optimální využití zdrojů
- infrastruktura „On-demand“, nepřetržitý bezporuchový provoz, eliminace ztrát obchodních příležitostí
- podpora v ČR
- centrálně spravovaná aplikace, snadná integrace dalších datových zdrojů, vazba na jiné IS přes XML^[IV]

3.3.2 Evoluce aplikačního serveru IBM

IBM přinesla na trh již mnoho verzí svého aplikačního serveru. První beta verze „aplikační serveru“ od společnosti IBM se objevila v červnu roku 1998, nesla název Servlet Express. Ve své podstatě to byl jen servletový stroj (ENG: servlet engine).

Ve **verzi 2** IBM přidala podporu technologie JavaBeans a CORBA a podporu pro operační systém Linux. Verze 2 se prodávala ve dvou edicích – *Standard Edition* a *Advance Edition* (SE, AE).^[X]

Verze 3 byla kompatibilní s JDK 1.2, J2EE 1.0. IBM zároveň přidala mnoho rozšiřujících vlastností do základní specifikace J2EE. Nově byla přidána podpora pro operační systém OS/400 (dnes i5/390) a OS/390 (dnes z/OS). Nově přibyla k předchozím i edice *Enterprise Edition* (EE)^[X]

Verze 4 se stala J2EE 1.2 certifikovaným aplikačním serverem. Tato verze měla, jako její předchůdci, konfiguraci uloženou v databázi, ale edice SE začala používat XML soubory.^[X]

Verze 5 byla uvolněna 19. listopadu 2002. Stala se také J2EE 1.3 certifikovaným aplikačním serverem. Z pohledů verzí je pětka potomkem jejích předchůdců,

ale technologicky musela být změněna od samotného jádra systému. Pětka se stala základem dnešního WebSphere Application Server. Konfigurace serveru se kompletně přesunula na model XML souborů. Byla přidána nová služba nazývaná Deployment Manager, která se stará o konfigurační soubory cell a node jednotek (cell = logický celek instancí-node, node je instance serveru). Do verze 5 byla také implementována odlehčená verze MQ 5.3, což byl Java Message Service (JMS) server. ^[X]

IBM nabízela verzi 5 v následujících edicích: ^[X]

- **Community Edition** – edice založená na open source základě, která je zdarma k dispozici – ideální pro vývoj a testy menších internetových řešení
- **Express Edition** – pro takzvané SMB – malé a střední podniky (ENG: Small and Medium Business)
- **Base Edition** – Edice aplikačního serveru na bázi J2EE vhodná pro méně robustní prostředí bez potřeby vysoké dostupnosti a škálovatelnosti. Obsahuje pouze základní součásti
- **Network Deployment** – tato edice podporovala clusterovou architekturu a J2EE failover ochranu. Dále obsahovala Edge komponenty. Obecně ND edice je vždy vhodná pro robustní nasazení s možností široké škálovatelnosti
- **Enterprise Edition** – disponoval workflow strojem (Process Choreographer), který používal pro modelování logiky předchůdce dnes používaného standardu BPEL (Business Process Execution Language). Také byla přidána kompletní podpora pro aplikační model – WebSphere Asynchronous Beans. Dnes se již tato edice nepoužívá
- **Extended Deployment** – edice určená do prostředí s potřebou maximálního výkonu, efektivního využívání hardware a podporou virtualizace a tzv. grid computingu (Grid Computing lze definovat jako aplikování prostředků z mnoha počítačů v síti (ve stejném okamžiku) na jeden problém. Obvykle se jedná o problém, který vyžaduje velký počet cyklů zpracování nebo přístup k velkému objemu dat. ^[VIII])

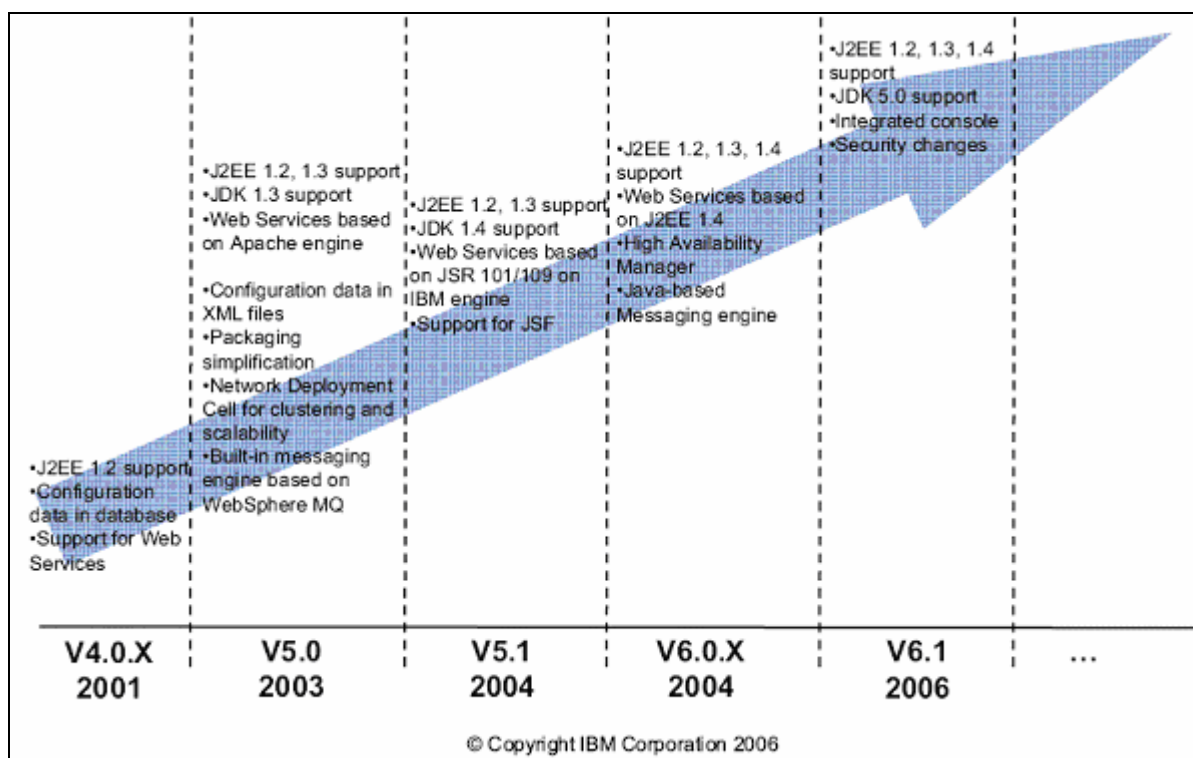
- *WebSphere Application Server for z/OS* – tato edice je v podstatě stejná jako *Network Deployment*. Rozdíl spočívá v optimalizaci pro systém z/OS, který se používá při vysoce vytěžovaných, kriticky důležitých, škálovatelných systémech s požadavkem na extrémní bezpečnost

Verze 6 byla vydána v prosinci 2004. WAS V6 je J2EE 1.4 kompatibilní aplikační server rozšířený o bezpečnostní vlastnosti – podpora JACC 1.0 a WS-Security 1.0. Pro informaci, Express edice se prodává za cenu \$2000/CPU – maximálně na 2 CPU. A nejdražší a nejvýkonnější edice podporující cluster stojí \$16000/CPU. *Ceny jsou pouze orientační.*

Dnes je k dispozici WebSphere Application Server ve **verzi 6.1**. Lze ji charakterizovat následujícími znaky:^[X]

- Podpora Java Standard Edition 1.5
- Podpora chodu Portletů ve standardu JSR 168 na aplikačním serveru
- Session Initiation Protocol (SIP) Servlety
- Přizpůsobení k WebSphere Instrall Faktory
- IBM Support Assistant
- IBM JSF Widget Library
- Zjednodušuje administraci
- Vylepšení správy certifikátů
- Zvýšení bezpečnosti
- Možnost správy IBM HTTP Serveru (IHS) z Admin Console
- Podpora Web Service Resource Framework a WS-BusinessActivity (WS-BA)
- Podpora JSR160 JMX Repote Connection
- Administration Console python Command Assistance^[X]

Na obrázku číslo čtyři jsou zachyceny základní změny v průběhu evoluce WebSphere Application Server od verze 4 do 6.1. Je zde dobře patrná reakce tvůrců WAS na vznik nových verzí J2EE a jiných technologií, které jsou se vznikem nových verzí aplikačního serveru úzce spjaty.^[X]



Obr. č. 4 – Evoluce WAS ^[VI]

3.3.3 WAS V6.1 podrobnější pohled

Verze 6.1 se dodává ve třech edicích a to *WebSphere Application Server Express/Base/Network Deployment*, které se od sebe liší svou použitelností, softwarovými doplňky a výkonností. Největší omezení, které sebou nese Express edice jsou především: absence clusterování a více-serverového řízení. Dále součástí dodávaného softwaru Express edice není Tivoli Direktory Server (LDAP), Tivoli Access Manager Server ani Edge komponenty. Další nepříjemné omezení Express edice spočívá v limitu počtu použitých procesorů. To znamená, že server nelze provozovat na více než dvou CPU. Omezení na procesory u edice Base a ND není!^[VI]

Přehlednější pohled na jednotlivé edice a jejich součásti zachycuje obrázek číslo pět.

	WebSphere Application Server Express	WebSphere Application Server (Base)	WebSphere Application Server Network Deployment
Core Application Server	Standalone Node	Standalone Node	Deployment Manager, Standalone Node, Managed Node
IBM HTTP Server V6.1 Web Server plug-ins	x	x	x
Application Client	x	x	x
Data Direct JDBC Drivers	x	x	x
Development and/or Deployment Tools	Rational Web Developer AST	Rational Application Developer "Trial" AST	Rational Application Developer "Trial" AST
Database included in the package	DB2 Express (Dev. Use only)	DB2 Express (Dev. Use only)	DB2
Edge Components			x
IBM Tivoli Directory Server (LDAP server)			x
Tivoli Access Manager Server			x

© Copyright IBM Corporation 2006

Obr. č. 5 – WAS edice ^[VI]

Je důležité připomenout, že všechny zde zmiňované edice verze 6 jsou postavené na stejném jádru se stejnou funkcionalitou! Pouze jednotlivé podmínky použitelnosti a doplňkový software se liší. To neplatí u edice nazvané Community Editio, která je založena čistě na open source technologiích. ^[VI]

Nově ve verzi 6.1 je zařazena podpora pro protokol Session Initiation Protocol (SIP) implementovaný do servletových aplikací. Podpora přináší možnost provozování chatu, VoIP a video konferencí a jiných komunikačních služeb využívajících právě protokol SIP.

IBM HTTP Server (dále jen IHS):

Součástí WAS je i HTTP server, který je založen na jádře Apache Web Server 2.0.47. Společnost IBM jej rozšířila o podporu SSL (Secure Sockets Layer) zabezpečené připojení, vylepšila možnosti pro ukládání do vyrovnávací paměti statických obsahů k navýšení rychlosti odezvy serveru. Dále byla přidána komponenta pro dynamický překlad HTML kódu do různých jazykových mutací, podpora standardních LDAP (Lightweight Directory Access Protocol)

autentifikačních a autorizačních serverů a v neposlední řadě možnost monitorování stavu IHS z webové administrační konzole WAS. ^[VI]

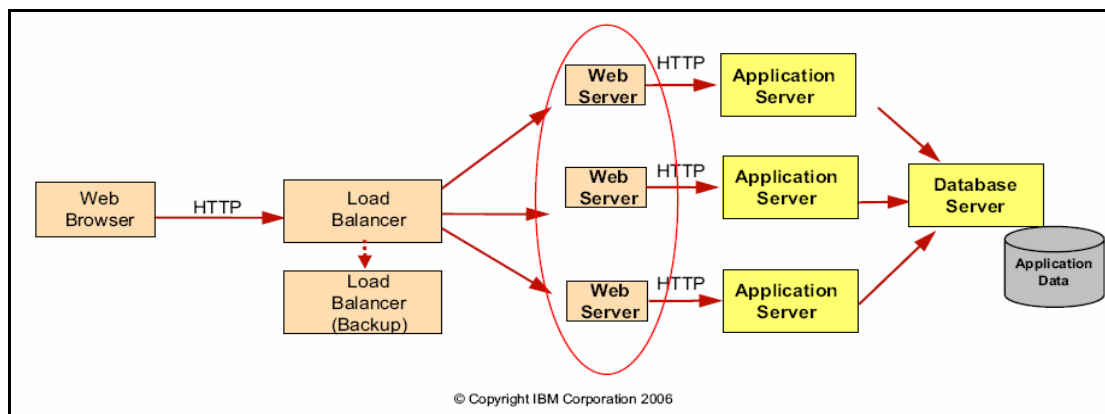
Edge Komponenty:

Jde především o síťově orientovaný software navržený k optimalizaci vysoké zátěže sítě a taktéž samotné architektury (IHS, WAS ...).

Vyrovňovač zátěže (Load Balancer), dříve znám také jako Network Dispatcher – software s širokou funkcionalitou. Umožňuje dynamicky monitorovat stav sítí a usměrňovat provoz takovým způsobem, aby nedocházelo k přetížením či dočasným výpadkům vedoucím k nedostupnosti Internetových aplikací. ^[VI]

Caching Proxy, dříve znám také jako Web Traffic ExpressProxy sloužící k ukládání statických obsahů do paměti Proxy serveru, který je následně schopen na některé požadavky rovnou odpovědět, aniž by musel přeposílat požadavek na aplikační server. Tím je velice přispíváno k hospodárnému a efektivnímu využívání celé infrastruktury a minimalizuje se zatěžování samotného aplikačního serveru. ^[VI]

Obrázek níže zachycuje zjednodušené použití Load Balanceru a Caching Proxy (Web Serveru). Funkcionalita je popsána následovně: Prohlížeč (Web Browser) zašle dotaz na zobrazení nějakého webového obsahu (stránky), dotaz podle aktuálního stavu vytížení sítě je Load Balancerem přeměřován na nejméně zaneprázdněný webový server. Webový server porovná požadovaná data s uloženými daty v jeho vyrovnávací paměti (cache) a pokud najde shodu, data (stránku) zašle prohlížeči zpět. Pokud ve vyrovnávací paměti daná data nemá, například z důvodu, že jde o dynamicky tvořený obsah, přesměruje požadavek na aplikační server. Aplikační server podle charakteru dotazu případně zašle dotaz ještě do databáze a poté stránku vygeneruje. Aplikační server v konečné fázi vygenerovanou stránku přepoše příslušnému webovému serveru, a web server následně zašle finální výsledek do prohlížeče uživateli. ^[VI]



Obr. č. 6 – Edge Komponenty^[VI]

3.3.4 Architektura serveru WAS - jednoduchá

Obrázek číslo sedm znázorňuje základní architekturu samostatného aplikačního serveru (ENG: stand alone server). Základní elementy jsou tvořeny okolními prvky a aplikačním serverem, který obsahuje Java procesor a kontejnery pro zpracovávání aplikační a jiné logiky.

Jádro

S návazností na kapitolu o J2EE technologii lze na obrázku sedm snadno rozpoznat například Web kontejner nebo EJB kontejner. Web kontejner slouží ke spuštění servletů a JavaServlet Pages (JSP stránek) a následné generování HTML kódu pro webový prohlížeč. Vygenerované HTML stránky z Web kontejneru jsou zasílány do interního HTTP serveru (Enhanced HTTP Server viz obr. č.7). Servlety a JSP stránky mohou fungovat bez propojení na další logickou strukturu ovšem v praxi tomu bývá naopak. Servlety či JSP stránky využívají dynamické zdroje dat, které jim zprostředkovává EJB kontejner. Ten umožňuje připojení na běžné relační databáze nebo jiné aplikační zdroje. V EJB kontejneru jsou zpracovávány klasické znovupoužitelné Java třídy volané ze servletu či JSP kódu. Na obrázku je pod EJB a Web kontejnerem vyznačen ještě Messaging Engine a Web Service Engine. Zmíněné komponenty jsou podpůrné subsystémy kontejnerů, které tak rozšiřují jádro WAS o další funkcionalitu. ^{[I], [VI]}

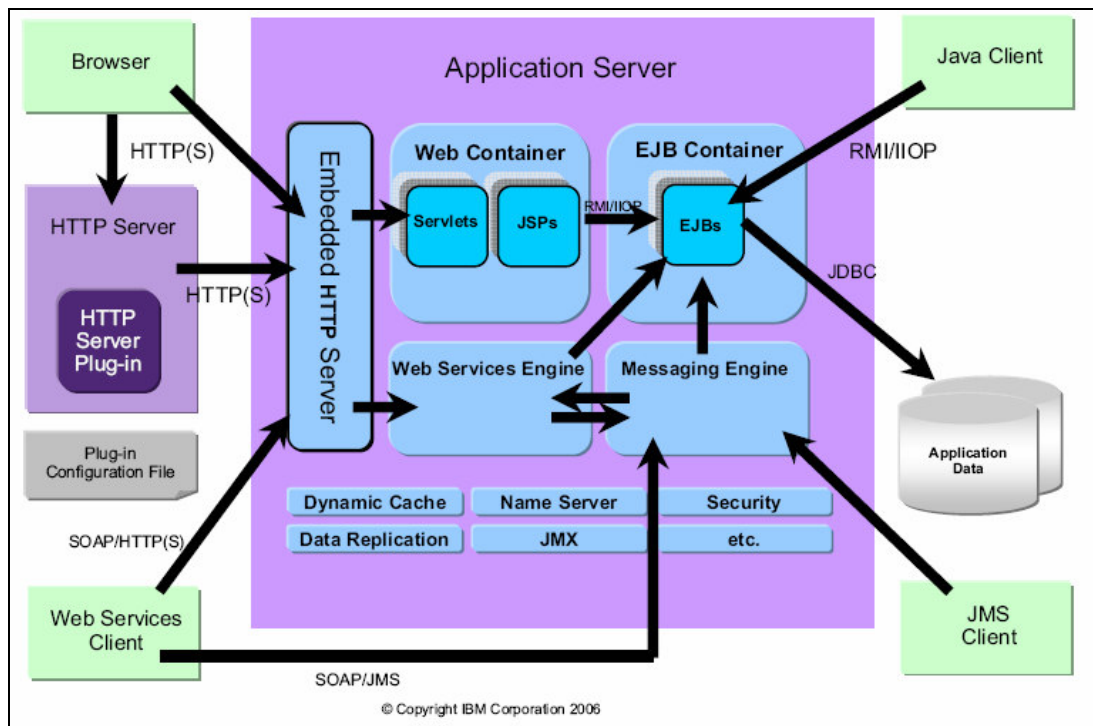
V jádru sehrávají významnou roli další nepostradatelné součásti, které jsou tvořeny jednotlivými komponenty JSM stroje, jenž se stará o asynchronní tok požadavků do a z jádra serveru. JMS je vyvinut ze základního Java-messaging stroje. A poskytuje v zásadě API (API = programové rozhraní) pro přístup k asynchronní sběrnici SI Bus (Service Integration Bus).^{[I], [VI]}

Jádro serveru mimo klasický způsob připojení přes HTTP server a Web kontejner, umožňuje přímý přístup do EJB kontejneru Java klientem pomocí protokolu RMI/IIOP, jak je vyznačeno na obrázku číslo sedm v horním pravém rohu. Ostatní komunikace probíhá přes alespoň jeden okolní subsystém jádra (Web kontejner, Messaging Engine, Web Service Engine)^{[I], [VI]}

Plug-in

Velmi důležitá komponenta, která ovšem neleží v samotném aplikační serveru, je tzv. zásuvný modul (ENG: plug-in). Plug-in zprostředkovává komunikaci mezi aplikačním serverem a předsazeným HTTP serverem (IHS). Jeho další role spočívá v možnosti správy hlavního konfiguračního souboru IHS – (*httpd.conf*) přímo z administrační konzole WAS. To ocení především administrátoři systému díky snazší správě serveru. Dále plug-in nabízí při výpadku jednoho z aplikačního serveru automatické přesměrování na jiný, akceschopný. Ovšem pouze na takový, který je uveden v konfiguračním souboru zásuvného modulu uloženého na IHS serveru – na obrázku číslo sedm „Plug-in Configuration File“ (*plugin-cfg.xml*).^{[I], [VI]}

Vnitřní architektura se při bližším prozkoumání stává více složitější a její popis by se vymykal rámci tohoto textu. Proto je zde popsán pouze hlavní základ.



Obr. č. 7 – Architektura WAS ^[VI]

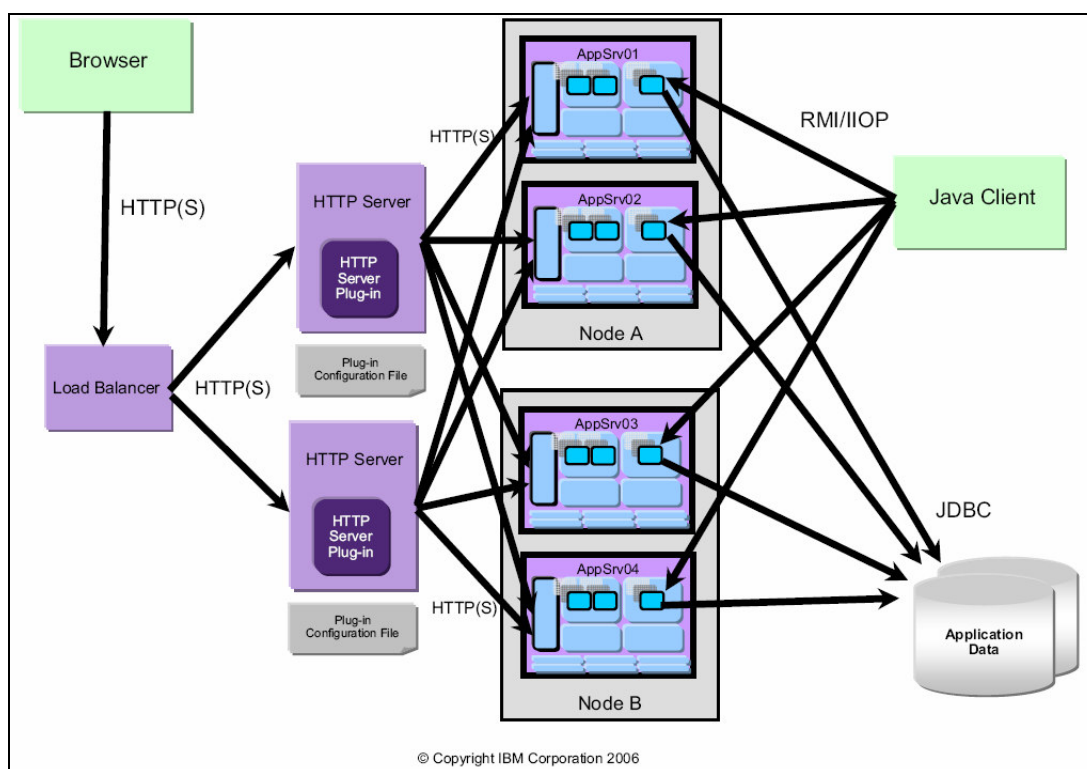
3.3.5 Architektura serveru WAS – federovaná

Federovaná architektura je založena na více logicky propojených serverech za účelem vyšší dostupnosti a spolehlivosti celého topologie. Pro snazší pochopení bude federovaná architektura popisována na modelovém příkladu:

Příklad: Vyšle-li uživatel, reprezentovaný internetovým prohlížečem (ENG: Browser), požadavek na stránku, dostane se k síťovému vyrovnávači (Load Balancer), ten podle aktuální zátěže přeměruje požadavek na určitý HTTP server. V rámci této bakalářské práce se pohybujeme v technologii IBM, tudíž je IBM HTTP Server propojen se zásuvným modulem (HTTP Server plug-in), který zabezpečí další přeměrování na příslušný aplikační server v rámci architektury. Přeměrování mezi IHS a aplikačním serverem je řešeno přes konfigurační soubor zásuvného modulu, ve kterém je mimo jiné pro každý server v „Nodu“ nadefinována prioritita serveru, podle které se požadavky rozesílají. ^[1]

Postupnou selekcí síťové cesty plynoucí z předešlého popisu je zajištěna vysoká dostupnost a spolehlivost poskytované aplikace např. při možném výskytu přetížení či selhání jednoho ze serverů.

Aplikační server je možné propojit s jiným než IHS serverem. Lze použít například Apache Web Server, Lotus Domino Web Server, Microsoft IIS či Sun ONE Web Server a jiné. Ovšem pouze při použití IHS lze automaticky distribuovat konfigurační soubor plug-inu (*plugin-cfg.xml*) za pomoci webové administrační konzole aplikačního serveru. V opačném případě je administrátor nucen vygenerovaný konfigurační soubor ručně zkopírovat k příslušnému HTTP serveru. Pokud by tak neučinil, provozované aplikace by nebyly směrovány na HTTP server a celá konektivita architektury by byla přerušena.



Obr. č. 8 – Federovaná architektura WAS ^[VI]

3.3.6 Správce konfigurací – Deployment Manager

Tím, že se pohybujeme ve federované architektuře, ve které se předpokládá více různých serverů, které ani nemusí fyzicky stát na stejném místě, dostáváme se před problém jejich správy. Řešení mohou přinést dnešní technologie nabízející různé

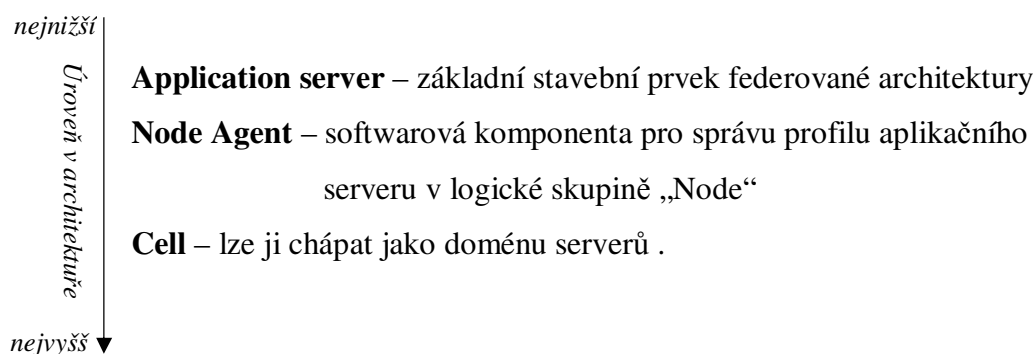
softwarové programy a komponenty pro vzdálené připojení k serveru, asi nejznámější a nejpoužívanější jsou SSH, VNC či vzdálená plocha v prostředí MS Windows. Existují také hardwarové komponenty (např. IBM Remote Supervisor Adapter I, II), které poskytují již sofistikovanější funkcionalitu, jako třeba připojení disku/CD-ROM našeho lokálního zdroje a umožnit tak např. instalovat operační systém vzdáleně na serveru či sledování všech hardwarových stavových hlášení (logů) a v neposlední řadě možnost přenesení obrazovky už v prostředí BIOSu (Basic In/Out System).

Nicméně výše zmiňované neřeší problém, jak sjednotit správu všech aplikačních serverů do jednoho centrálního místa/jedné webové stránky/jedné konzole/jednoho okna. Proto IBM pro své webové servery vyvinula Správce konfigurací (ENG: Deployment Manager), který v zásadě umožňuje spravovat a konfigurovat vzdáleně jednotlivé servery pod něj zařazené.

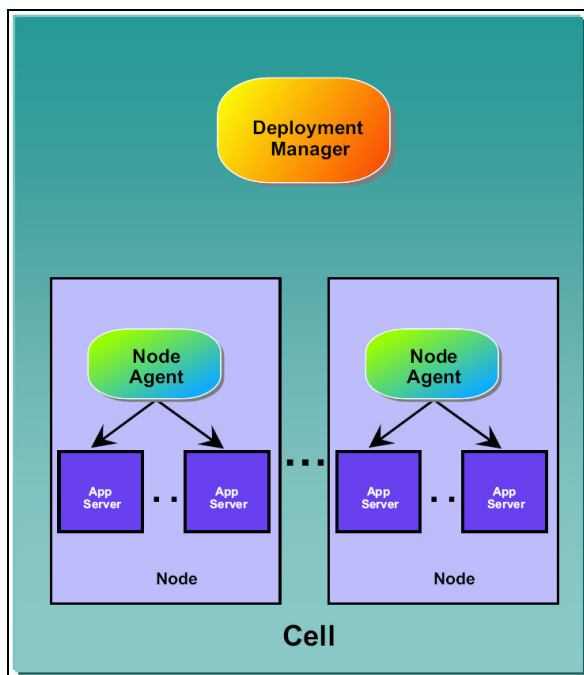
Jak vše funguje?

Na nejnižší úrovni ve federované architektuře stojí samostatný aplikační server, respektive jeho instance (profil). Aby Deployment Manager (dále jen DM) mohl daný profil serveru vzdáleně ovládat, musí nad serverem být spuštěn tzv. Node Agent. Ten zajišťuje „spojení“ mezi DM a jedním nebo více profily aplikačních serverů. Pokud pod sebou Node Agent má více profilů aplikačních serverů, vzniká logická množina Node. Existuje-li v architektuře Node Agent je na něj připojen DM. DM může, obdobně jako Node Agent vs. profily aplikačních serverů, spravovat více Node Agentů. Propojením DM a Node vzniká tzv. buňka (ENG: Cell).^[VI]

Logické uspořádání jednotlivých komponent v architektuře



Obrázek číslo devět demonstruje hierarchické rozložení jednotlivých elementů tvořící ve finále buňku Cell. Z obrázku je patrné, že jeden Node Agent umožňuje správu několika aplikačních serverů. Stejným způsobem Deployment Manager může spravovat více logických jednotek Node.



Obr. č. 9 – Logická struktura buňky Cell^[VI]

4. Administrace WAS

Kapitola Administrace WAS a následující kapitola postupně nastíní možné scénáře instalace, praktickou ukázkou instalace aplikačního serveru na dva virtuální počítače, následné nasazení předpřipravené J2EE aplikace a vytvoření clusteru o dvou aplikačních serverech. Jako prekvizita pro nasazovanou aplikaci J2EE bude zapotřebí relační databáze DB2. Tudíž její základní instalace a konfigurace bude také v textu uvedena. Poslední komponenta, která bude mimo WAS a BD2 instalována je IBM HTTP Server, který bude předsazen před aplikační server.

4.1 Možné instalační scénáře

Existuje mnoho způsobů jak nainstalovat WebSphere Application Server a ostatní komponenty. Jednotlivé scénáře se od sebe liší především možnostmi výkonnostními a bezpečnostními. V praxi je většinou výsledná architektura ovlivněna požadavky na výkon, bezpečnost a v neposlední řadě financemi společnosti.

Vše na jednom serveru - WebSphere aplikační server lze nainstalovat na jeden fyzický server například společně s DB2 relační databází a IHS serverem. Zajisté takovéto uspořádání bude vyhovovat z finančního hlediska, jelikož pro celou architekturu postačí jeden server, ale pro produkční prostředí se takto nainstalovaný systém absolutně nehodí. A to nejen z důvodu výkonnostního, ale i bezpečnostního!^[1]

Oddělení HTTP serveru – Architektura s odděleným HTTP serverem je již o krok dále než předcházející z hlediska bezpečnosti a částečně i výkonnosti. Díky tomu, že HTTP server je na jiném fyzickém stroji než samotný WAS s databází, není velké riziko jej vystavit částečně mimo DMZ. A také samotný běh web serveru neubírá systémové prostředky pro provoz WAS či databáze.^[1]

Oddělení HTTP serveru a databáze – se provádí v případech, kdy očekáváno abnormální zatížení buď WAS nebo databázového serveru. Hlavní důvod oddělení databáze od aplikačního serveru je tedy především z výkonnostního hlediska –

aby jeden systém neubíral příliš mnoho systémových prostředků tomu druhému a neohrozil tak dostupnost aplikace. Nepsané pravidlo pro dostupnost webové stránky hovoří o maximální délce načítání webové stránky, která by neměla přesáhnout maximálně tři vteřiny ^[III]

Další scénáře – plynou z kombinování více funkcionálně stejných komponent do clusterů pro rapidní zvyšování výkonu a dosažení takzvané 24x7 dostupnosti. Systémy postavené na federované architektuře vyžadují nejen kvalitní hardwarové/softwarové, ale i lidské zdroje, které budou schopny kvalifikovaně danou infrastrukturu spravovat. Z finančního hlediska stojí clusterová infrastruktura na nejvyšším stupni nákladovosti, avšak i z výkonnostního hlediska je na tom nejlépe.

Pro cenovou představu výkonný server IBM xSeries 3800 s 4 CPU 64bit 8GB RAM stojí kolem 25 000,- USD k tomu je zapotřebí přičíst licence na software a popřípadě servisního technika IBM – 3300,-Kč/h. To jsou pouze čisté pořizovací náklady, dále by bylo potřeba počítat s administrátorem (celým týmem), který daný stroj nakonfiguruje, nainstaluje WebSphere a zprovozní celou infrastrukturu. Pokud společnost nemá vyvinuté podnikové aplikace, bude muset financovat i vývoj. Vývoj podnikových aplikací se pohybuje v řádech miliónů korun. Takové investice se mohou pro podnik zdát neúnosné, ale musí se vzít v úvahu, co podnik v budoucnu novými aplikacemi získá, jakou návratovou hodnotu budou mít a jak rychle budou vynaložené prostředky na aplikace vráceny.

4.2 Použité prostředky pro praktickou část

Praktická část tohoto textu byla realizována na dvou PC. Bližší specifikace viz. tabulka:

Typové označení	Specifikace
IBM ThinkPad T41	<ul style="list-style-type: none">• Intel Pentium M 1.7GHz• 2GB DDR RAM• HDD 40 GB, 5700 RPM• ATI Mobility Radeon 9000• 1000 MBit LAN

Typové označení	Specifikace
IBM ThinkPad T42p	<ul style="list-style-type: none"> • Intel Pentium M 1.8GHz • 2GB DDR RAM • HDD 60 GB, 7200 RPM • ATI Mobility Radeon 9000 • 1000 MBit LAN

Síť LAN (Local Area Network)

Pro síťové propojení počítačů byl použit router SMC BARICADE 7004ABR, který podporuje rychlost sítě pouze 100 Mbit, tudíž nebyla využita kapacita síťových karet.

Virtualizace

Z důvodů usnadnění práce byla použita virtualizace PC. K tomuto účelu byl vybrán software od společnosti An EMC Company - VMware Workstation 6.0.0 Build-45731. VMware umožňuje vytváření zálohy celého virtuálního systému a jeho následné rychlé obnovování například při nepovedených konfiguracích či kolizních instalací komponent do systému. Tato funkcionality byla několikrát velice oceněna v průběhu realizace praktické části bakalářské práce.

Hardwarová konfigurace virtuálních PC

Konfigurace jednotlivých virtuálních PC pojmenovaných **was6host01** a **was61host02** byla pro oba stejná, viz. obrázek číslo deset.

Memory	1512 MB
Hard Disk (SCSI 0:0)	10.0 GB
CD-ROM (IDE 1:0)	Using drive E:
Ethernet	Bridged
USB Controller	Present
Display	1 monitor
Processors	1

Obr. č. 10 – Konfigurace virtuálních PC

Operační systémy na virtuálních PC

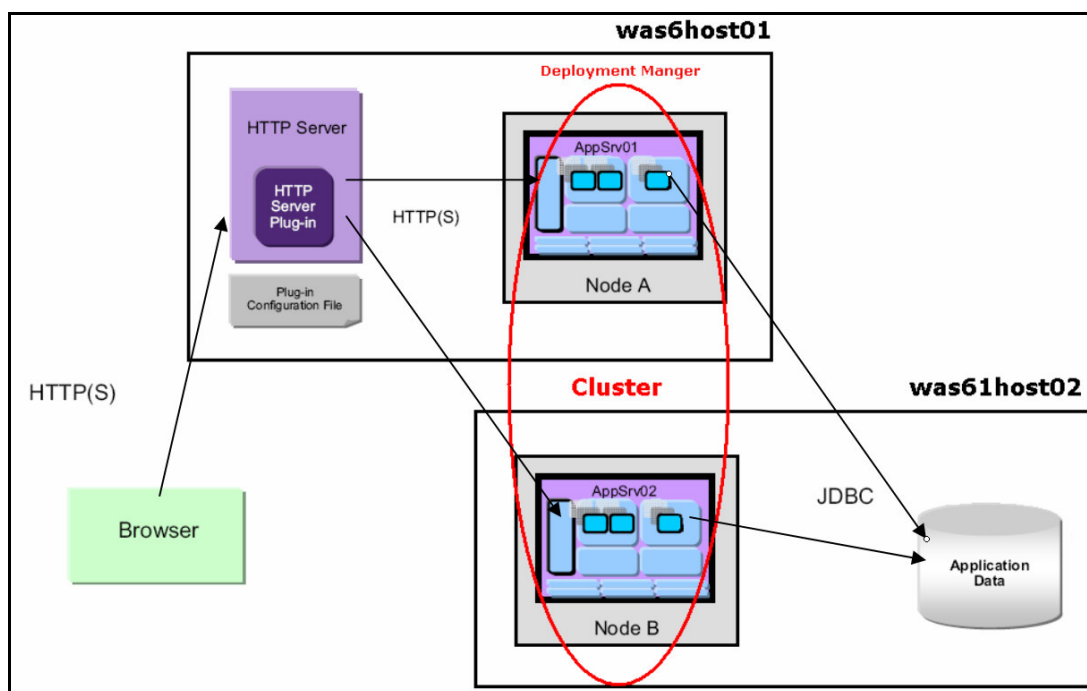
Na virtuálním PC **was6host01** byl nainstalován Windows Server 2003 Standard Edition SP1. Na PC **was61host02** byl také nejdříve nainstalován Windows Server

2003 Standard Edition SP1, ale při realizaci se objevily určité komplikace s databázovým systémem, které byly řešeny instalací SP2 (ServicePack 2).

V dalších kapitolách bude popsána instalace jednotlivých komponent na virtuální počítače. Pouze IHS a Deployment Manager je instalován pouze na virtuální počítač **was6host02**.

4.3 Instalace a konfigurace

Navzdory tomu, že celou architekturu lze implementovat na větší množství hardwarového zařízení, pro omezené prostředky bylo nutno zvolit trochu netradiční instalační scénář plynoucí z použití pouze dvou PC. Na jednom virtuálním PC (**was6host01**) byl nainstalován WebSphere aplikační server spolu s IBM HTTP Serverem a dočasně DB2 relační databází. Na druhém (**was61host02**) byl nainstalován také aplikační server s databází DB2. V průběhu vývoje konfigurace se využití relační databáze na prvním PC stalo redundantní, proto na obrázku není zobrazeno. Další raritou se stalo samotné použití virtualizačního softwarového nástroje pro usnadnění případné obnovy systému při realizaci praktické části.



Obr. č. 11 – Realizovaná topologie

4.3.1 Předinstalační úpravy OS

Před samotnou instalací je zapotřebí provést kontrolu kompatibility použitého hardwaru a softwaru. Detailní výčet všech HW/SW požadavků je zanesen v příloze Systémové požadavky WAS nebo na stránce Systémových požadavků pro WAS na adrese: <http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html> Pokud systém splňuje všechny předepsané požadavky, je možné přejít k části konfiguračních úprav operačního systému. V první fázi je důležité nastavit doménové jméno, které po instalaci WAS již nepůjde změnit. Respektive v operačním systému změnit půjde, ale je tak hluboko zaneseno v samotné konfiguraci aplikačního serveru, že není jiný způsob jak ho změnit, než reinstalaci serveru. Dále je nutné nastavit požadovaná systémová práva pro administrátora. Detaily naleznete v základní literatuře nebo v Informačním centru IBM.^[1]

Instalační CD

Pro instalaci bylo zapotřebí následujících softwarových balíčků:

- WebSphere Application Server V6.1 Network Deployment for Windows
- WebSphere Application Server V6.1 Network Deployment for Windows – Supplemental
- DB2 UDB Enterprise Edition V8.2 for Windows FP14

První softwarový balíček obsahuje základní instalaci aplikačního serveru. Druhý obsahuje IHS, HTTP Plug-in a aplikačního klienta. Třetí, jak již z názvu plyne, obsahuje relační databázi DB2 pro systém Windows.

4.3.2 Instalace BD2

Instalace serveru DB2 UDB Enterprise Edition V8.2 lze provést přes grafického instalačního průvodce nebo přes textový režim nebo v tzv. „silent modu“. Grafický a textový průvodce administrátora provede všemi nezbytnými nastaveními. V průvodci lze nechat hodnoty ve výchozím nastavení, pouze uživatelské jméno a heslo pro přístup do databáze je zapotřebí změnit. K případné kontrole správnosti

nastavení instalačního průvodce je v rámci příloh začleněn výpis nastavení grafického průvodce před spuštěním instalace viz. Instalace DB2.^[IX]

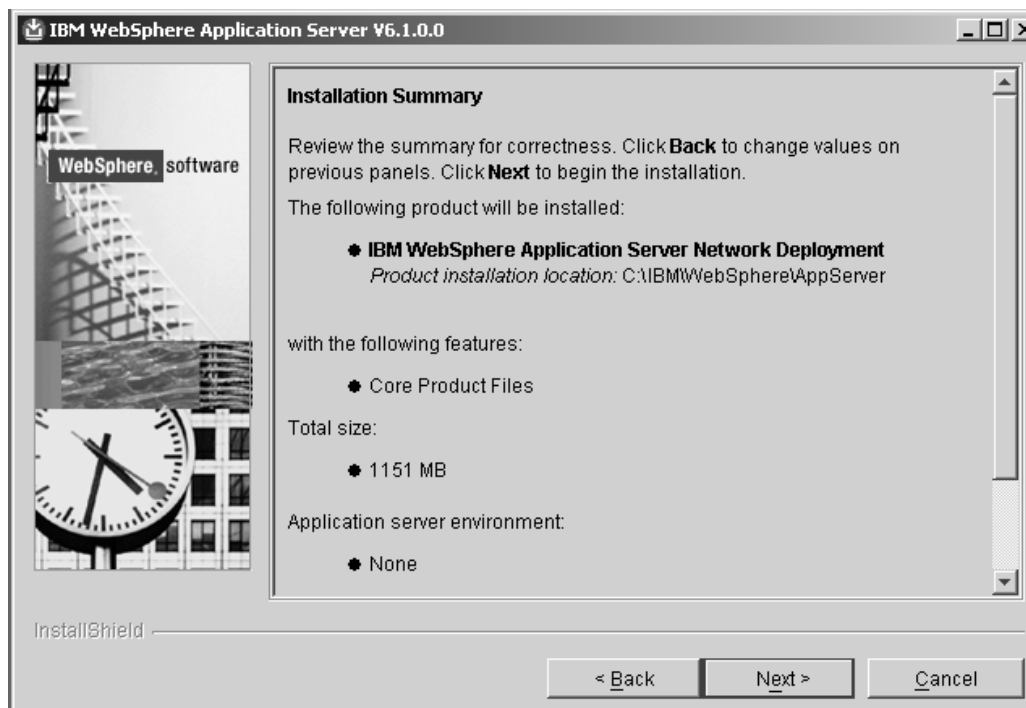
K realizaci praktické části byla použita modelová J2EE aplikace „Trade“ společnosti IBM využívaná na různých školeních a prezentacích – aplikace byla získána v rámci bakalářské praxe k studijním účelům. Její součástí je i malá relační databáze, která byla pomocí jednoduchého skriptu implementována do databáze DB2 na obou virtuálních PC.

4.3.3 Instalace WebSphere Application Server – ND V6.1

Samotná instalace aplikačního serveru vyžaduje určité předinstalační kroky, které byly popsány v kapitole 4.3.1. Po jejich provedení již nic nebrání k přechodu do fáze instalace aplikačního serveru. Instalace lze provést podobným způsobem jako výše popisovaná instalace DB2 serveru v kapitole 4.3.2. K dispozici je opět grafické rozhraní, textové rozhraní a „silent mode“, pro který je zapotřebí konfigurační soubor obsahující informace o požadované instalaci.^[III]

Doporučení:

Praxe ukázala, že je v některých případech výhodné, změnit výchozí adresář **C:\Program Files** na **C:** z důvodů možných budoucích problémů při příliš dlouhé adresářové struktuře. WebSphere server některé složky pojmenovává dlouhými unikátními klíči, a právě díky tomu by mohlo dojít k dosažení maximální povolené délky názvů celého stromu adresářů.



Obr. č. 12 – Shrnutí instalace WAS

Při instalaci nebyla vybrána volba vytvoření instance aplikačního serveru z důvodu absence podrobnějšího nastavení instalace. Instance byla vytvořena v následujícím kroku grafickým průvodcem pro tvoření profilů takzvaným „Profile Manager Tool“. V něm je důležité na začátku zaškrtnout volbu „Advance profile creation“, pokud by tomu tak nebylo, v následujících krocích by grafický průvodce nezobrazil všechny dostupné možnosti nastavení instalace.

Profile Manager Tool – postup

Průvodce lze spustit z nabídky Start nebo pomocí instalačního CD s aplikačním serverem.

Průvodcem necháme nainstalovat administrační konzoli a základní aplikace jako Snoop, Hello atd.

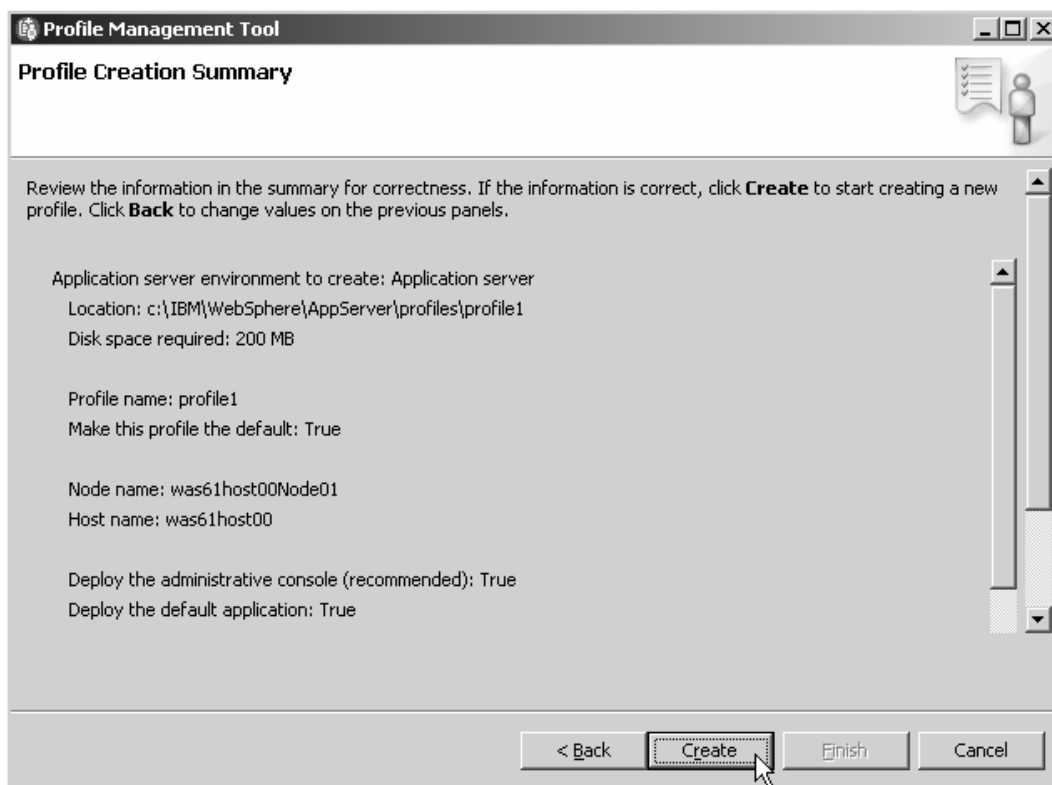
Na další stránce byly zanechány výchozí adresáře pro uložení profilů ve výchozím stavu a přešlo se k pojmenování Node. Zde bylo zadáno **was6host01Node01** a na druhém virtuálním PC **was61host02Node01**, jméno host bylo ponecháno ve výchozím tvaru tj. **was6host01** a **was61host02**.

Další stránka v grafickém průvodci slouží pro nastavení globální bezpečnosti serveru. V případě praktické realizace byla bezpečnost nastavena do aktivního stavu.

Poslední stránka slouží pro nastavení jednotlivých portů služeb serveru, i zde byly zanechány výchozí hodnoty.

Doporučení:

Před dalšími kroky je vhodné tabulku s porty uložit pomocí kombinace kláves Alt+PrintScreen a vložit vyfocené okno do aplikace malování, která je součástí standardní instalace Windows, jako obrázek. Tabulka totiž obsahuje kompletní seznam portů, které budou použity na serveru pro vytvářený profil aplikačního serveru.



Obr. č. 13 – Shrnutí instalace profilu

Ověření instalace

Před pokračováním v dalších krocích, je nutné ověřit správnost instalace z tzv. „First steps“ konzole, kliknutím na odkaz „Installation verification“ a zkontrolovat zda se shodují poslední dva zakroužkované řádky s obrázkem číslo čtrnáct:^[111]

```
First steps output - Installation verification
IVTL0010I: Connecting to the WebSphere Application Server was61host00 on port: 9080
IVTL0020I: The Installation Verification Tool cannot connect to WebSphere Application Server; waiting for
Start running the following command:cmd.exe /c "c:\IBM\WebSphere\AppServer\profiles\profile1\bin\startServer.bat"
>ADMU0116I: Tool information is being logged in file c:\IBM\WebSphere\AppServer\profiles\profile1\logs\server1\startServer.log
>
>ADMU0128I: Starting tool with the profile1 profile
>ADMU3100I: Reading configuration for server: server1
>ADMU3200I: Server launched. Waiting for initialization status.
>ADMU3000I: Server server1 open for e-business; process id is 2220
IVTL0015I: WebSphere Application Server was61host00 is running on port: 9080 for profile profile1
Testing server using the following URL:http://was61host00:9080/ivtserver?parm2=ivtserverlet
IVTL0050I: Servlet engine verification status: Passed
Testing server using the following URL:http://was61host00:9080/ivtserver?parm2=ivtAddition.jsp
IVTL0055I: JavaServer Pages files verification status: Passed
Testing server using the following URL:http://was61host00:9080/ivtserver?parm2=ivtejb
IVTL0060I: Enterprise bean verification status: Passed
IVTL0035I: The Installation Verification Tool is scanning the file c:\Program Files\IBM\WebSphere\AppServer\profiles\profile1\logs\server1\startServer.log
[6/9/06 15:53:03:048 EDT] 0000000a WSKKeyStore W CWPJK0041W: One or more key stores are us
[6/9/06 15:53:22:966 EDT] 0000000a ThreadPoolMgr W WSVR0626W: The ThreadPool setting on the
IVTL0040I: 2 errors/warnings are detected in the file c:\IBM\WebSphere\AppServer\profiles\profile1\logs\server1\startServer.log
IVTL0070I: The Installation Verification Tool verification succeeded.
IVTL0080I: The installation verification is complete.
```

Obr. č. 14 – Ověření instalace WAS – výpis konzole

Další způsob, kterým lze ověřit instalaci, spočívá v samotném přihlášení pod administračním jménem a heslem, které bylo zadáno při aktivaci globální bezpečnosti v grafickém instalačním průvodci, do administrační konzole, která je dostupná na adrese: <http://localhost:9060/ibm/console>

Port musí odpovídat portu administrační konzole z tabulky zobrazené při instalaci.

Třetí způsob ověření běhu serveru lze provést pomocí příkazové řádky příkazem **serverStatus -all**, který vypíše stav všech nalezených serverů respektive profilů. Jednotlivé parametry jsou odděleny mezerou a pomlčkou. Parametr **-all** lze nahradit za jméno profilu např. **server1** již bez pomlčky, v takovém případě příkaz vypíše stav pouze daného profilu. Pokud je aktivována globální bezpečnost serveru jako další parametry jsou zadány přihlašovací údaje administrátora podle následující syntaxe:

```
C:\IBM\WebSphere\AppServer\bin\serverStatus.bat -all
-user <jméno_administrátora> -password <heslo>
```

Poznámka: Příkaz je psán v jednom řádku.

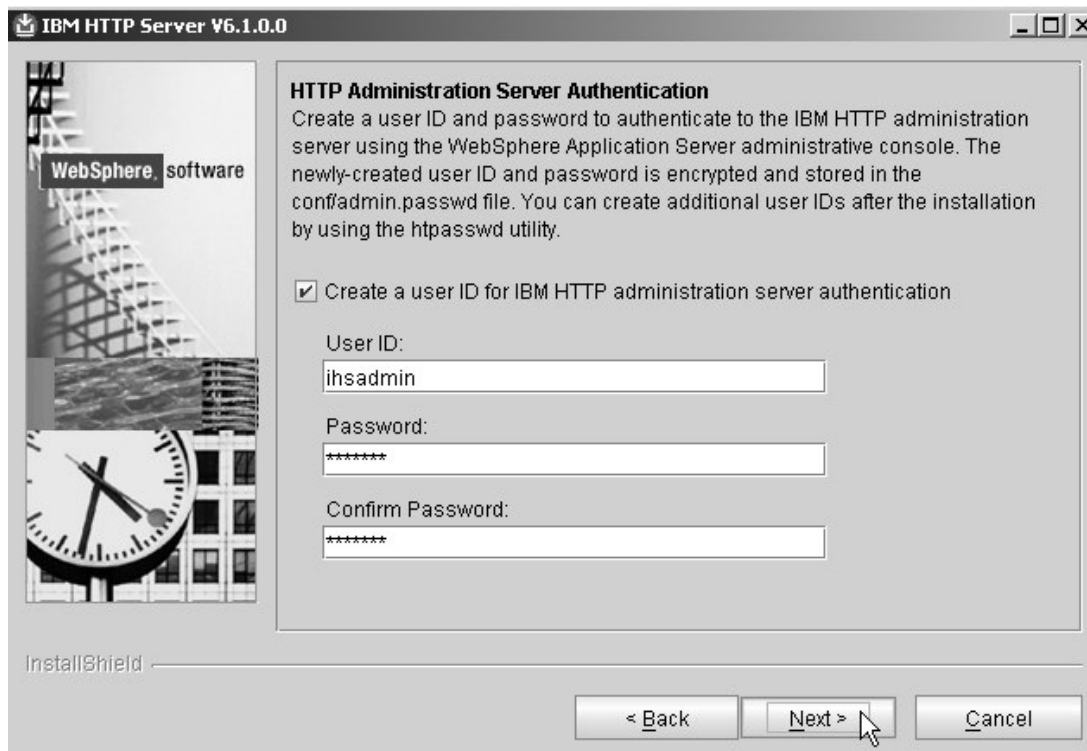
4.3.4 Instalace IHS

Po úspěšné instalaci aplikačního serveru a vytvoření profilu, lze přejít k instalaci webového serveru. Na demonstrovaném případě to bude webový server od společnosti IBM – IBM HTTP Server. IHS byl především vybrán díky pokročilé integraci s WebSphere aplikačním serverem.

Instalace IHS na was6host01

Instalaci je opět možno realizovat obdobně jako u DB2 či WAS a to prostřednictvím grafického či textového rozhraní. Grafický průvodce je vytvořen velmi intuitivně, ke každé volbě je připsán podrobný popis, tudíž při instalaci je nutné dát pozor pouze na část potřebnou pro budoucí integraci mezi aplikačním serverem a IHS.

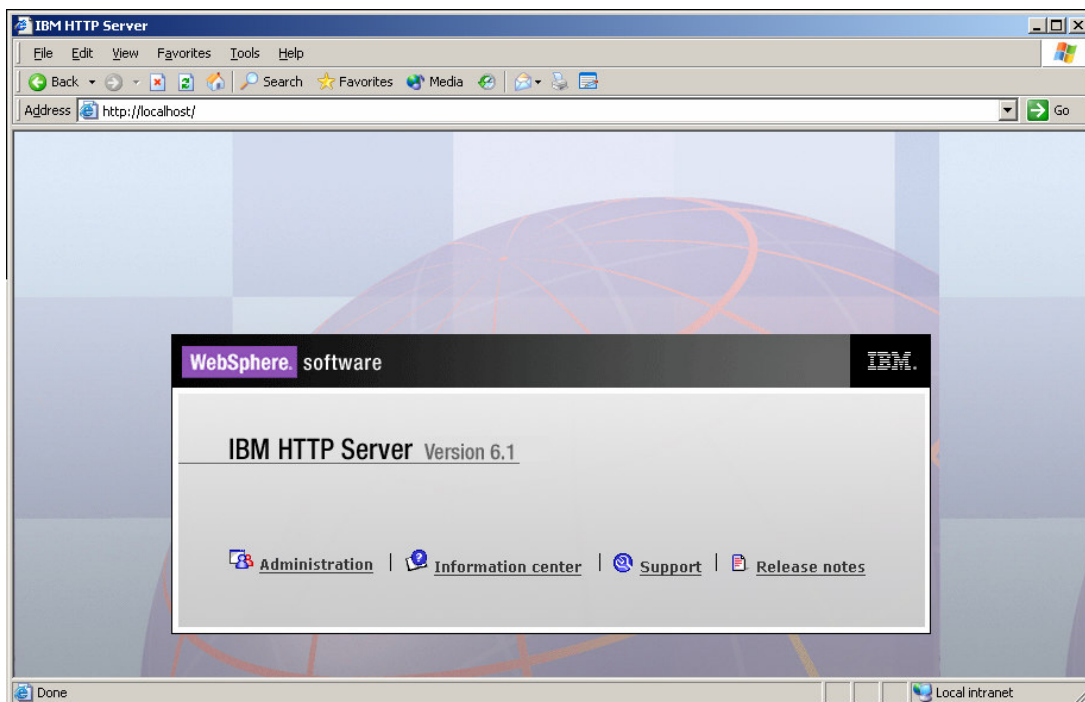
Jde především o krok „HTTP Administration Server Authentication“ který slouží pro nastavení uživatelského jména a hesla, přes které se administrační konzole WAS bude přihlašovat k IHS, a umožní tak jeho snadnou vzdálenou správu. V rámci instalace IHS grafický průvodce nabízí možnost nainstalování plug-inu, která byla při realizaci praktické části tohoto textu využita.^[III]



Obr. č. 15 – HTTP Administration Server Authentication

Po dokončení instalace a nastartování serveru prostřednictvím správce služeb, pouze v případě je-li IHS nainstalován jako služba, lze jeho funkčnost ověřit na adrese: <http://localhost/>

Pokud je vše korektně nainstalováno, měla by se zobrazit úvodní stránka webového serveru. [III]



Obr. č. 16 – Úvodní stránka IHS

5. Praktické použití administrační konzole WAS

V kapitole pět je demonstrováno použití administrační konzole při implementaci modelové aplikace Trade společnosti IBM. Po jejím nasazení na jeden z virtuálních počítačů bude nastíněna realizace clusteru o dvou aplikačních serverech a následné otestování funkcionality vypnutím jednoho z aplikačních serveru při běhu Trade aplikace.

5.1 Instalace Trade aplikace

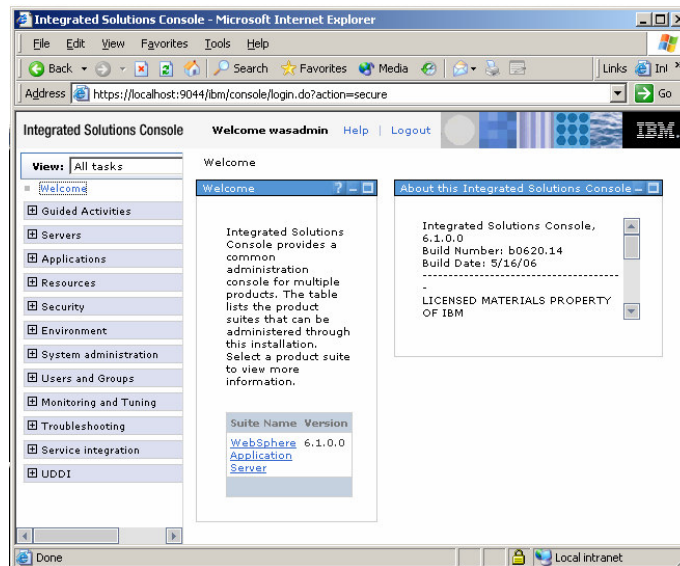
Instalace Trade aplikace byla provedena přes webovou administrační konzoli WAS dostupnou z následující URL (s příslušným portem daného aplikačního serveru) <http://localhost:9080/ibm/console> po nastartování aplikačního serveru např. příkazem z příkazové řádky.

```
C:\IBM\WebSphere\AppServer\profiles\<>jmeno_složky
profilu>\bin\startServer.bat -server1 -user
<jméno_administrátora> -password <heslo>
```

Poznámka: Příkaz je psán do jednoho řádku.

Popis příkazu:

První parametr příkazu je vždy jméno profilu serveru **server1**, další oddělen mezerou a pomlčkou je uživatelské jméno administrátora **-user <jméno _administrátora>** a poslední parametr se stejnou syntaxí oddělení je heslo **-password <heslo>**.

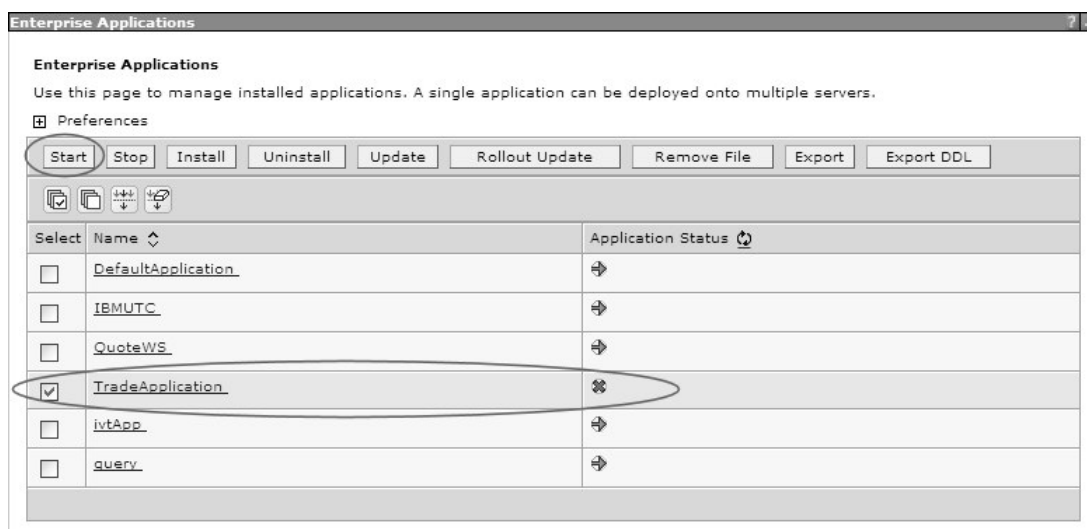


Obr. č. 17 – Administrační webová konzole WAS

Samotnou aplikaci IBM dodává v částečně nedodělaném stavu. Proto si realizace praktické části bakalářské práce vynutila použití vývojové nástroje dodávaného k aplikačnímu serveru – WebSphere Application Toolkit. Pomocí tohoto vývojového nástroje byly do Trade aplikace dodefinovány některé XML soubory pro přístup k databázi. Především byl nadefinován DB2 Universal JDBC Driver Provider (XA) poskytovatel dále jméno, heslo, port a další potřebné informace pro připojení do databáze využívané aplikací. Poté byla aplikace uložena a vyexportována ve formátu EAR (Enterprise Archive), který se snadno instaluje do aplikačního serveru IBM.

Instalace aplikace

Proces instalace aplikace je velice jednoduchý díky webové administrační konzoli. V části **Application – Install new application** nainstalujeme vyexportovaný EAR soubor. Důležitá část konfiguračního procesu po instalaci aplikace zahrnuje nastavení poskytovatele spojení do databáze se stejnými parametry jako byly zadány pomocí vývojového nástroje WebSphere Application Toolkit. Po úspěšné instalaci a konfiguraci dalších potřebných parametrů, které zde nebudou z hlediska rozsahu popisovány, je možné aplikaci spustit pomocí tlačítka **Start** v části **Application – Enterprise Application**.



Obr. č. 18 – Start aplikace

Potvrzení funkcionality na adrese <http://localhost9080/Trade/web>

5.2 Vytvoření clusteru

Ve fázi vytvoření clusteru jsou na obou virtuálních počítačích nainstalovány BD2 UDB servery s databází potřebnou pro Trade aplikaci. Dále jsou nainstalovány aplikační servery, každý s jedním profilem jako samostatný server (ENG: stand alone server - znamená, že server není řízen žádným Node agentem). Jako poslední krok byla nasazena Trade aplikace a zároveň otestována její funkcionality.

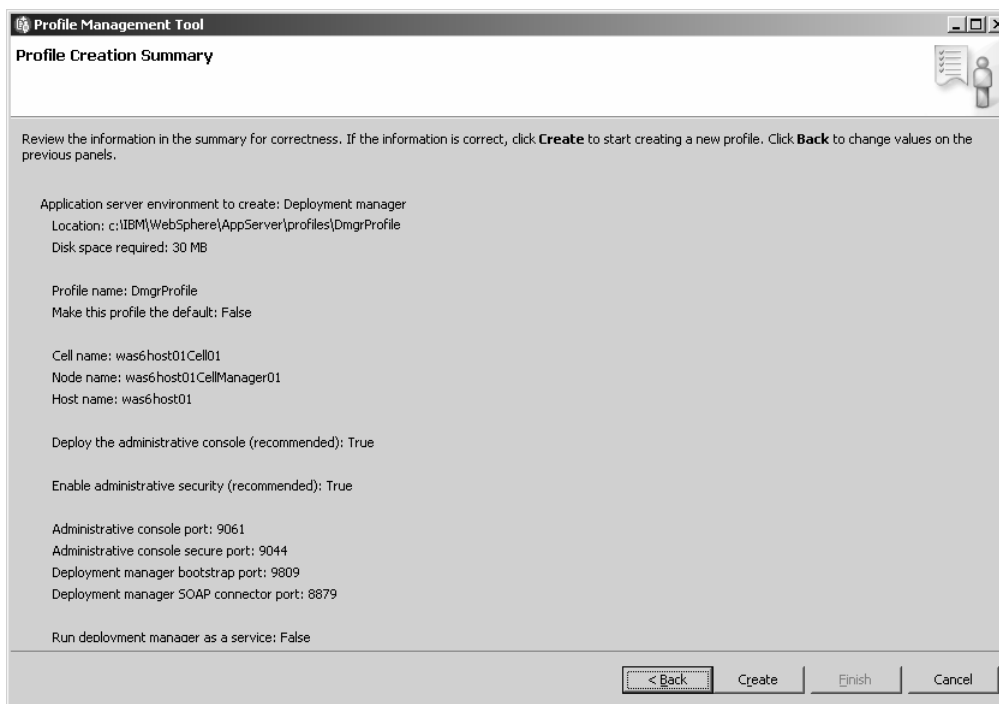
V následující fázi bude nastíněno vytvoření z těchto dvou aplikačních serverů pomocí Deployment Managera buňky Cell, z které bude v dalších krocích nakonfigurován aplikační cluster.

Instalace DM

Nad jednotlivými aplikačními servery, respektive jejich profily je zapotřebí nasadit Node agenty, které se napojí na správce konfigurací – Deployment Managera. Bližší funkcionality byla popisována v kapitole 3.3.6, z toho důvodu zde není rozsáhlejší popis.

Pro instalaci Deployment Managera na **was6host01** je nutné využít již zmiňovaný „Profile Manager Tool“ v kterém je vybrána volba – vytvořit Deployment Manager.

Další postup je obdobný s instalací profilu aplikačního serveru v kapitole 4.3.3 v části Profile Manager Tool – postup.



Obr. č. 19 – Shrnutí instalace DM

Po instalaci DM byly přidány na serveru **was6host01** webovou administrační konzolou v části *System administration – Nodes* a tlačítkem *Add Node*, oba Node z obou virtuálních PC. Při procesu přidávání Node do buňky Cell je důležité, aby na obou PC byl nastaven stejný čas – maximální povolená odchylka je stanovena na pět minut.

Objevený problém:

V průběhu realizace praktické části nebyl znám požadavek na časovém omezení a proto se přidání nepovedlo hned na poprvé. Ovšem v zápětí se po dodatečném sladění času vyskytl další problém. Deployment Manager neakceptoval dodatečné sladění a stále varoval, že není na obou PC stejný čas. Po dlouhém zkoumání možných variant nastavení byly opět systémové hodiny vráceny, čímž nastala původní odchylku, která činila téměř devět hodin. V tom okamžiku DM přidání Nodu do buňky Cell provedl bez jakýchkoliv chyb. Tento problém byl konzultován s odborníkem v IBM ČR, který navrhol tuto chybu nahlásit centrále pro sběr chyb

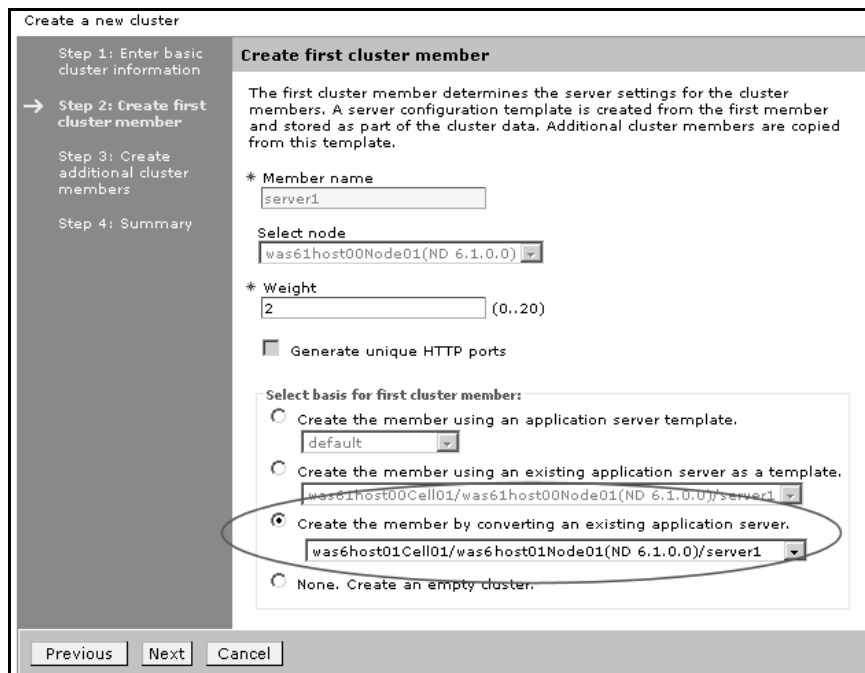
v softwaru IBM. Před tím byl proces chyby ještě jednou zopakován, aby se docílilo stoprocentní jistoty výskytu dané chyby a postupu při kterém vzniká.

Následující krok spočívá ve vytvoření, pomocí webové administrační konzole, nového Unmanage Node a následně jeho zařazení také do buňky Cell, kde již jsou was6host01 a was61host02 Node. Unmanage Node bude sloužit pro řízení HTTP Serveru (IHS) prostřednictvím webové administrační konzole.

5.2.1 Clusteru

Nasazením DM se dostáváme do finální fáze praktické části, ve které bude nastíněn postup vytvoření části clusteru. Cluster je specifický tím, že při pádu jednoho ze členů dokáže přenést stavové informace potřebné pro zachování běhu aplikace na provozuschopný aplikační server. Pro replikování těchto informací byl použit tzv. mechanismus z paměti-do-paměti (EMG: memory-to-memory). Existuje ještě další, který je založen na replikování dat za pomoci databáze.

Vytvoření clusteru bylo provedeno opět prostřednictvím uživatelsky přívětivého prostředí webové administrační konzole Deployment Managera, která se liší pouze jiným portem v URL adrese (port viz. instalace DM). Cluster se vytváří v části **Servers – Clusters** a tlačítkem **New**. Jednotlivé členy clusteru byly vytvořeny zkonvertováním stávajících profilů aplikačních serverů na obou virtuálních PC tak, jak je demonstrováno na obrázku:



Obr. č. 20 – Vytvoření clusteru

Po vytvoření clusteru je nutné namapovat aplikace, které na něm mají být provozovány. Při mapování aplikací se nesmí zapomenout přidat i Node pro HTTP Server (IHS), ve kterém je následně vygenerován XML konfigurační soubor s informacemi pro plug-in. XML soubor je po stisknutí tlačítka **Propagate Plug-in** automaticky distribuován na IHS. Kopii vygenerovaného XML souboru si lze prohlédnout v příloze „XML Soubor plug-inu“.

Tím že je vytvořena nová architekturu na serveru, je zapotřebí znova nastavit na správnou úroveň, tzn. na **was6host01** a **was61host02** Node, JDBC databázový zdroj podle již nastíněného postupu.

Memory-to-Memory

Poslední částí vytvoření clusteru spočívá v propojení té části operační paměti, kam se ukládají stavové informace o běžících aplikacích na daném aplikačním serveru respektive o vytvořených session. Jak již v úvodu této kapitoly bylo podotknuto, k realizaci praktické části bakalářské práce byla využita metoda nazvaná memory-to-memory.^[11]

Přenos sessiony pomocí databáze

WebSphere Application Server umožňuje ještě další ekvivalentní metodu pro zajištění předávání session mezi servery. Jde o způsob využívající databázový server, do kterého si každý aplikační server při vytvoření nové session uloží její kopii. Pokud dojde k selhání jednoho ze serverů v clusteru, přebírající server si z databáze zreplikuje kopii sessiony spadlého serveru do své operační paměti a tím na sebe přebírá provoz spadlého serveru.^[I]

5.2.2 Demontrace clusteru

Výsledkem celé praktické části je modelová demonstrace použití clusteru, při pádu jednoho ze serverů, respektive při jeho okamžitém vypnutí. Samotná demonstrace je popsána na následujícím postupu:

Primárně je zapotřebí nastartovaného clusteru obsahujícího oba aplikační servery (**was6host01** a **was61host02**) a dále spuštěnou Trade aplikaci na celém clusteru tzn. na obou serverech. V následujícím kroku je otevřeno okno webového prohlížeče s URL aplikací směřovanou přes IHS server: <http://localhost/Trade/web>

Po načtení stránky je možno se přihlásit pod přihlašovací jméno „client“. Načtení nové webové stránky je signalizací pro úspěšné přihlášení do aplikace. V levé části stránky je zobrazeno, na kterém ze dvou serverů v clusteru byla při přihlášení vytvořena sessiona. Pokud byla vytvořena sessiona na serveru **was6host01**, pro otestování clusteru je nutné zastavit server pomocí webové administrační konzole v části *Servers – Application Servers* tlačítkem „*Stop immediately*“. Poté co je server zastaven, přepnutím zpět do okna s Trade aplikací a následnou jakoukoliv akcí na stránce vyvoláme nový požadavek, který bude odeslán na cluster. Za povšimnutí stojí fakt, že v levé části okna se změnil v tomto případě server **was6host01** na **was61host02**. Změnou je prověřena funkcionality správného předávání sessiony mezi jednotlivými servery metodou memory-to-memory.^[III]

6. Závěr

WAS je velice propracovaný a vysoce škálovatelný produkt IBM. Jeho nespornou výhodou je schopnost snadné integrace díky použitým technologiím, ale hlavně díky široké softwarové základně IBM WebSphere produktů, které se mohou pochlubit nejen vysokou kompatibilitou sami mezi sebou, ale i mezi dalšími technologiemi.

Z pohledu administrátora, do kterého se autor tohoto textu dostal, je WAS velice obsáhlý aplikační server s rozsáhlými možnostmi nastavení a konfigurací plynoucích z jeho samotného charakteru jakožto základního prvku celé WebSphere technologie. Avšak kvůli rozsáhlým možnostem nastavení se aplikační server místy stává nepřehledný, čímž si jeho správa říká o leckdy hluboké znalosti v problematice a dovednosti. Proto se staly součástí realizace praktické části bakalářské práce konzultace s odborníky, které pomohly při řešení určitých problémů.

V průběhu konfigurace byla zřejmě objevena závažná chyba aplikačního serveru, která po dalším prozkoumání a potvrzení bude nahlášena na centrálu pro sběr chyb softwaru IBM. Více informací viz kapitola 5.2 část - Objevený problém.

Po téměř čtyř a půl týdenní usilovné práci na praktické části se podařilo dojít k cíli praktické části bakalářské práce, jejíž jádro bylo založeno na realizaci clusterového řešení na platformě IBM WebSphere.

7. Zdroje

- [I] Carla Sadtler, Fabio Albertoni, Bernardo Fagalde, Thiago Kleinubing, Henrik Sjostrand, Ken Worland, Lars Bek Laursen, Martin Phillips, Martin Smithson, Kwan-Ming Wan. *WebSphere Application Server V6.1- System Management and Configuration*: IBM Press, 2006. 984 p. ISBN 0-73-849652-9.
- [II] Carla Sadtler, Fabio Albertoni, Bernardo Fagalde, Thiago Kleinubing, Henrik Sjostrand, Ken Worland. *WebSphere Application Server V6.1 - Planning and Design*: IBM Press, 2006. 324 p. ISBN 0-73-849450-X.
- [III] IBM WebSphere Training and Technical Enablement. *IBM WebSphere Application Server V6 Administration - Student Exercise*: IBM Press, 2005. Part No. CT3H6NA, Product Code SW246.
- [IV] *Komix, s.r.o.* [online].
URL: http://www.komix.cz/Produkty/SW_na_miru/Pouzivane_produkty/IBM_WebSphere.aspx, [cit. 2006-12-25].
- [V] *CESNET, z. s. p. o.* [online].
URL: <http://www.cesnet.cz/fond-rozvoje/dalsi/pdf/043r1.pdf>, [cit. 2006-12-25].
- [VI] *IBM WebSphere Application Server Administration – Instruktor Guide ERC 6.1*, IBM Press, 2006. Course code: WA361.
- [VII] *OKsystem* [online].
URL: http://portal.oksystem.cz/portal/page?_pageid=53,42315&_dad=portal&_schema=PORTAL, [cit. 2007-02-13].
- [VIII] *IBM CZ – Podniky se strategií On Demand – Technologický vývoj* [online].
URL: <http://www-05.ibm.com/e-business/cz/evolving/ondemand/technology/grid.html>, [cit. 2007-06-02].
- [IX] *IBM Information Center DB2 – Instalace* [online].
URL: <http://publib.boulder.ibm.com/infocenter/db2luw/v8//index.jsp>, [cit. 2007-06-05].
- [X] *Wikipedia – The Free Encyclopedia* [online].
URL: http://en.wikipedia.org/wiki/WebSphere_Application_Server, [cit. 2007-05-20].

8. Přílohy

1. Instalace DB2 – nastavení instalačního průvodce
2. Systémové požadavky WAS
3. XML soubor Plug-inu

8.1 Seznam obázků

Obr. č. 1 – Základní produktová mapa IBM WebSphere.....	5
Obr. č. 2 – Vícevrstvá architektura J2SE.....	6
Obr. č. 3 – Poloha aplikačního serveru.....	9
Obr. č. 4 – Evoluce WAS	14
Obr. č. 5 – WAS edice.....	15
Obr. č. 6 – Edge Komponenty	17
Obr. č. 7 – Architektura WAS	19
Obr. č. 8 – Federovaná architektura WAS.....	20
Obr. č. 9 – Logická struktura buňky Cell ^[VI]	22
Obr. č. 10 – Konfigurace virtuálních PC.....	25
Obr. č. 11 – Realizovaná topologie	26
Obr. č. 12 – Shrnutí instalace WAS	29
Obr. č. 13 – Shrnutí instalace profilu	30
Obr. č. 14 – Ověření instalace WAS – výpis konzole.....	31
Obr. č. 15 – HTTP Administration Server Authentication.....	32
Obr. č. 16 – Úvodní stránka IHS.....	33
Obr. č. 17 – Administrační webová konzole WAS	35
Obr. č. 18 – Start aplikace	36
Obr. č. 19 – Shrnutí instalace DM.....	37
Obr. č. 20 – Vytvoření clusteru.....	39

8.2 Použité zkratky

.NET	Vývojová platforma společnosti Microsoft
API	Application Programming Interface - programové komunikační prostředí
BIOS	Basic Input-Output System - základní programové prostředí osobního počítače
BPEL	Business Process Execution Language - jazyk pro modelování obchodní logiky
CD	Compact Disc - datové médium
CORBA	Common Object Request Broker Architecture -určitý softwarový standard
CPU	Central Processing Unit - je to ústřední výkonová jednotka počítače
DB2	DataBase2 - Označení databázového softwaru společnosti IBM
DMZ	DeMilitarized Zone - důvěryhodná zóna v síťovém prostředí
EAR	Enterprise Archive - komprimovaný soubor obsahující Java strukturu

EJB	Enterprise Java Bean - programová komponenta spouštěná na straně serveru
FP(xy)	Fix Pack - opravný balíček softwaru
HP	Hewlett-Packard - nadnárodní firma zabývající se globálně počítači
HTML	HyperText Markup Language - značkovací jazyk pro hypertext
HTTP	Hyper Text Transfer Protocol - je internetový protokol určený pro přenos dokumentů ve formátu HTML
HW	Hardware - elektronické vybavení počítačů
IBM	International Business Machines - přední světová společnost v oboru informačních technologií
IHS	IBM HTTP Server - webový server společnosti IBM
IIS	Internet Information Services - balík internetových služeb od společnosti Microsoft
J2EE	Java Platform, Enterprise Edition - jde o rozšířenou verzi J2SE
J2SE	Java Platform, Standard Edition - jde o rozšířenou verzi Javy
JDBC	Java Database Connectivity - je API pro programátory v programovacím jazyku Java,
JACC	Java Authorization Contract for Containers – standart pro bezpečnost v prostředí Java
JDK	Java Development Kit - je soubor základních nástrojů pro vývoj aplikací pro platformu Java
JMS	Java System Message - systém pro komunikaci v rámci Javy
JSR 160	Java Management Extensions - specifické API prostředí
JSF	JavaServer Face - technologie generující uživatelské rozhraní
JSP	JavaServer Pager - technologie umožňující generování dynamického obsahu webových stránek
LAN	Local Area Network - označení pro lokální počítačové sítě
LDAP	Lightweight Directory Access Protocol - je definovaný protokol pro ukládání a přístup k datům na adresářovém serveru.
MDA	Mobile Digital Assistant - kombinace PDA a mobilního telefonu
ND	network Deployment - Správce konfigurací IBM WebSphere Application serveru
OS	Operation System - operační systém
OS/400	Operation Systém/400 - operační systém společnosti IBM
PC	Personal Computer - osobní počítač
PDA	Personal Data Assistant - kapesní verze osobního počítače
RAM	Random-Access Memory - jde o elektronickou paměť s náhodným přístupem
RPM	Revolutions per minute — anglická zkratka pro jednotku úhlové rychlosti
RMI/IIOP	Protokol pro komunikaci distribuovaných aplikací bez ohledu na použité platformě a vývojovém prostředí.
SI Bus	Sistem Integration Bus - softwarový integrační prostředek
SIP	Session Initiation Protocol - je internetový protokol určený pro přenos signalizace v internetové telefonii.
SMB	Small and Medium Business - malé a střední podniky/obchod
SP(x)	ServicePack - opravný balíček softwaru
SSH	Secure Shell - je klient/server protokol v síti TCP/IP, který umožňuje šifrovanou komunikaci mezi dvěma počítači.
SSL	Secure Sockets Layer – podpora šifrovaného přenosu
SW	Software - programové vybavení počítače
UDB	Universal DataBase - edice DB2 produktu společnosti IBM
URL	Unikátní Internetová adresa
VNC	Virtual Network Computing - umožňuje přenos grafického prostředí na vzdálený počítač pomocí TCP/IP sítě
WAS	WebSphere Application Server - aplikační server společnosti IBM
WS-BA	WS-BusinessActivity
WWW	World Wide Web - soustava propojených hypertextových dokumentů
XML	eXtensible Markup Language - je obecný značkovací jazyk, který byl vyvinut a standardizován konsorciem W3C
z/OS	zero-downtime/OS - Operační systém společnosti IBM

1. Instalace DB2 – nastavení instalačního průvodce

Instalovaný produkt:	DB2 Enterprise Server Edition
Typ instalace:	Typická
Vybrané funkce:	
Ukázky pro Spatial Extender	
Podpora JDBC	
Sada Java Development Kit	
Prostředí Java Runtime Environment	
Využití LDAP	
Podpora ODBC	
Podpora OLE DB	
Vzorové aplikace	
Vzorová databáze	
Podpora SQLJ	
Nástroje pro vývoj aplikací SQLJ	
Ukázky SQLJ	
Webové nástroje DB2	
XML Extender	
Ukázky pro XML Extender	
APPC	
Základní nástroje pro vývoj aplikací	
Apply	
Základní podpora klientů	
Systémové vázané soubory	
Capture	
Řídicí centrum	
Nástroje klienta	
Příkazové centrum	
Asistent pro konfiguraci	
Podpora Connect	
Databázové nástroje	
Vývojové centrum	
Analyzátor událostí	
Začínáme s produktem DB2	
Klient Spatial Extender	
NetBIOS	
Modul pro naslouchání NetBIOS	
Pojmenovaná propojení procesů	
Modul pro naslouchání pro Pojmenovaná propojení procesů	
TCP/IP	
Modul pro naslouchání TCP/IP	
Cílový adresář:	C:\Program
Files\IBMASQLLIB	
Požadované místo:	464 MB
Nové instance:	
Jméno instance:	DB2
Spustit instanci při startu systému:	Ano
Konfigurace TCP/IP:	
Jméno služby:	db2c_DB2
Číslo portu:	50000
Informace o uživateli instance:	
Jméno uživatele:	db2admin
Server DB2 Administration Server:	
Informace o uživateli instance:	
Jméno uživatele:	db2admin

2. Systémové požadavky WAS

URL: <http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg27007663>

(+) Produkty jsou součástí instalace

Operating Systems supported with 32-bit WebSphere Application Server

- Windows 2000 Advanced Server with SP4
- Windows 2000 Server with SP4
- Windows 2000 Professional Server with SP4 [Windows 2000 product site \(See Required Detail ^{1b}\)](#)
- Microsoft Windows Server 2003, Datacenter with SP1
- Microsoft Windows Server 2003, Enterprise with SP1
- Microsoft Windows Server 2003, Standard with SP1 [Windows Server 2003 product site](#)
- Microsoft Windows XP Professional with SP2 [\(See Required Detail ^{1b}\)](#)

Operating Systems supported with 64-bit WebSphere Application Server

- Microsoft Windows Server 2003 x64 Editions [Windows Server 2003 product site](#)

Hardware Requirements

For Windows (32 bit WebSphere Application Server)

- Intel® Pentium processor at 500 MHz or faster
- Intel EM64T or AMD Opteron (32-bit operating-system support only)
- [Disk space requirements](#)
- Minimum 512 MB physical memory; 1 GB recommended
- CD-ROM drive

For Windows (64 bit WebSphere Application Server)

- AMD Opteron or Intel EM64T
- [Disk space requirements](#)
- Minimum 1 GB of physical memory recommended
- CD-ROM drive

Java™

The WebSphere-supplied Java 2 SDK is required for both the run time and any remote Java clients

- (+) IBM 32-bit SDK for Windows, Java 2 Technology Edition, V5.0 SR2
- (+) IBM 64-bit SDK for Windows on AMD64/EM64T architecture, Java 2 Technology Edition, V5.0 SR2

Databases

[See the JDBC Drivers section](#)

- (+) Cloudscape™ 10.1 [Cloudscape product site \(See Required Detail ³\)](#)
- DB2® for iSeries™ 5.2, 5.3, or 5.4 [DB2 for iSeries product site](#)
- DB2 for z/OS® V7 or V8 [DB2 for z/OS product site](#)
- DB2 for Linux®, UNIX®, and Windows see document [#7009244](#) for supported levels and editions
- WebSphere Information Integrator 8.2 FP4 [WebSphere Information Integrator product site](#)
- Informix® Dynamic Server [9.4C7W1](#) or [10.00C4 Informix Dynamic Server product site](#)
- Oracle 9i Standard/Enterprise Release 2 - 9.2.0.7 [Oracle Database product site](#)
- Oracle 10g Standard/Enterprise Release 1 - 10.1.0.4 [Oracle Database product site](#)
- Oracle 10g Standard/Enterprise Release 2 - 10.2.0.1 or 10.2.0.2 [Oracle Database product site](#)
- Microsoft SQL Server Enterprise 2000 SP4 [Microsoft SQL product support site](#)
- Microsoft SQL Server Enterprise 2005 [Microsoft SQL product site](#)
- Sybase Adaptive Server Enterprise 12.5.2 or 15.0 [Sybase Adaptive Server Enterprise product site](#)

Java Database Connectivity (JDBC) Drivers

For the mapping of the JDBC Driver to supported databases refer to the information center article, [Vendor-specific data sources minimum required settings](#).

- DB2 Driver for JDBC and SQLJ 3.1 (type 2/4)
- DB2 Legacy CLI-based JDBC Driver (type 2) [\(See Required Detail ⁴\)](#)
- DB2 Universal JDBC Driver 2.8 (type 2/4)
- (+) IBM WebSphere embedded Connect JDBC 3.5 build 37 (type 4)
- DataDirect Connect JDBC 3.5 build 37 (type 4) [DataDirect Connect JDBC product site](#)
- (+) IBM WebSphere embedded SequeLink JDBC 5.5 build 5701 (type 3) [\(See Required Detail ⁵\)](#)
- DataDirect SequeLink JDBC 5.5 build 5701 (type 3) [DataDirect SequeLink product site \(See Required Detail ⁵\)](#)
- IBM Toolbox for Java/JTOpen JDBC Driver 4.8 (type 4) [Toolbox for Java product site](#)
- Informix JDBC Driver 3.00 JC3 (type 4) [Informix JDBC product site](#)
- Microsoft SQL Server 2000 JDBC Driver 2.2.0040 (type 4) [\(See Required Detail ⁴\)](#)
- Oracle 9i or 10g JDBC Driver [Oracle JDBC product site \(See Required Detail ⁵\)](#)

	<ul style="list-style-type: none"> ● Sybase jConnect 5.5 EBF 11656 or 6.0 EBF 12723 (type 4) Sybase jConnect product site
<p>LDAP Servers</p>	<ul style="list-style-type: none"> ● IBM Tivoli® Directory Server 5.2 or 6.0 IBM Tivoli Directory Server product site ● IBM z/OS Security Server 1.6 or 1.7 IBM z/OS Security Server product site (See Required Detail ⁵) ● IBM z/OS.e Security Server 1.6 or 1.7 IBM z/OS.e Security Server product site (See Required Detail ⁵) ● Lotus® Domino® Enterprise Server 6.5.4 or 7.0 Lotus Domino product site ● Novell eDirectory 8.7.3 or 8.8 Novell eDirectory product site ● Sun ONE Directory Server 5.1 SP4 or 5.2 Sun product site ● Windows Active Directory 2000 or 2003 Windows Active Directory 2000 product site Windows Active Directory 2003 product site
<p>Web Servers on 32-bit kernel Refer to the information center article, Installing Web server plug-ins.</p>	<ul style="list-style-type: none"> ● Apache HTTP Server 2.0.54 Apache HTTP Server product site (See Required Detail ¹) ● IBM HTTP Server for WAS 6.0.2 IBM HTTP Server product site (See Required Detail ¹) ● (+) IBM HTTP Server for WAS 6.1 IBM HTTP Server product site (See Required Detail ¹) ● Internet Information Services 5.0 IIS product site (See Required Detail ¹) ● Internet Information Services 6.0 IIS product site (See Required Detail ¹) ● Lotus Domino Enterprise Server 6.5.4 or 7.0 Lotus Domino product site (See Required Detail ¹) ● Sun Java System Web Server 6.0 SP9 Sun product site (See Required Detail ¹) ● Sun Java System Web Server 6.1 SP3 Sun product site (See Required Detail ¹) and (See Required Detail ²)
<p>Web Servers on 64-bit kernel Refer to the information center article, Installing Web server plug-ins.</p>	<ul style="list-style-type: none"> ● Apache HTTP Server 2.0.54 (32-bit plug-in support) Apache HTTP Server product site (See Required Detail ¹) ● IBM HTTP Server for WAS 6.0.2 (32-bit plug-in support) IBM HTTP Server product site (See Required Detail ¹) ● (+) IBM HTTP Server for WAS 6.1 (32-bit plug-in support) IBM HTTP Server product site (See Required Detail ¹) ● Internet Information Services 5.0 (32 and 64-bit plug-in support) IIS product site (See Required Detail ¹) ● Internet Information Services 6.0 (32 and 64-bit plug-in support) IIS product site (See Required Detail ¹) ● Lotus Domino Enterprise Server 6.5.4 or 7.0 (32-bit plug-in support) Lotus Domino product site (See Required Detail ¹) ● Sun Java System Web Server 6.0 SP9 (32-bit plug-in support) Sun product site (See Required Detail ¹) ● Sun Java System Web Server 6.1 SP3 (32-bit plug-in support) Sun product site (See Required Detail ¹) and (See Required Detail ²)
<p>Web browsers</p>	<ul style="list-style-type: none"> ● Firefox 1.5 ● Internet Explorer 6.0 SP 1 ● Mozilla 1.7.8 (See Required Detail ⁸)
<p>JMS Provider</p>	<ul style="list-style-type: none"> ● IBM WebSphere MQ 5.3 CSD 12 or 6.0.1.1 WebSphere MQ product site (See Required Detail ⁷) ● IBM WebSphere MQ 5.3.1 for z/OS + PK10303 or 6.0 + PK22728 WebSphere MQ product site (See Required Detail ⁷)
<p>Security</p>	<ul style="list-style-type: none"> ● IBM Tivoli Access Manager 6.0 for trust association interceptor Tivoli Access Manager product site (See Related Information) ● IBM Tivoli Access Manager 6.0 for enhanced trust association interceptor (support for Subject creation) (See Related Information) ● IBM Tivoli Access Manager 6.0 for Java Authorization Contract for Containers (JACC) (See Related Information)
<p>Third Party ORBs</p>	<ul style="list-style-type: none"> ● Iona Technologies Orbix Enterprise Edition 6.3 Iona Technologies product site (See Required Detail ¹¹) (See Related Information) ● Borland VisiBroker 6.5 Borland VisiBroker product site (See Required Detail ¹¹) (See Related Information) ● Java 2 Platform, Standard Edition (J2SE) Version 1.4.2 by Sun Microsystems Inc. Sun J2SE product site (See Required Detail ¹¹) (See Related Information) ● Java 2 Platform, Standard Edition (J2SE) Version 1.5 by Sun Microsystems Inc. Sun J2SE product site (See Required Detail ¹¹) (See Related Information)

3. XML soubor Plug-inu

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <!--
HTTP server plugin config file for the webserver was6host01Cell01.ihsnode.webserver01
generated on 2007.06.12 at 11:15:51 PM CEST
-->
= <Config ASDisableNagle="false" AcceptAllContent="false" AppServerPortPreference="HostHeader"
  ChunkedResponse="false" FIPSEnable="false" IISDisableNagle="false" IISPluginPriority="High"
  IgnoreDNSFailures="false" RefreshInterval="60" ResponseChunkSize="64"
  VHostMatchingCompat="false">
<Log LogLevel="Error" Name="C:\Program
  Files\IBM\HTTPServer\Plugins\logs\webserver01\http_plugin.log" />
<Property Name="ESIEnable" Value="true" />
<Property Name="ESIMaxCacheSize" Value="1024" />
<Property Name="ESIInvalidationMonitor" Value="false" />
= <VirtualHostGroup Name="default_host">
<VirtualHost Name="*:9080" />
<VirtualHost Name="*:80" />
<VirtualHost Name="*:9443" />
<VirtualHost Name="*:5060" />
<VirtualHost Name="*:5061" />
<VirtualHost Name="*:443" />
<VirtualHost Name="quote.trade.com:9080" />
<VirtualHost Name="quote.trade.com:80" />
<VirtualHost Name="quote.trade.com:9443" />
<VirtualHost Name="quote.trade.com:5060" />
<VirtualHost Name="quote.trade.com:5061" />
<VirtualHost Name="quote.trade.com:443" />
</VirtualHostGroup>
= <ServerCluster CloneSeparatorChange="false" IgnoreAffinityRequests="true" LoadBalance="Round
  Robin" Name="TradeCluster_V2" PostBufferSize="64" PostSizeLimit="-1"
  RemoveSpecialHeaders="true" RetryInterval="60">
= <Server CloneID="12cgrm5t4" ConnectTimeout="0" ExtendedHandshake="false"
  LoadBalanceWeight="2" MaxConnections="-1" Name="was6host01Node01_server1"
  ServerIOTimeout="0" WaitForContinue="false">
<Transport Hostname="was6host01" Port="9080" Protocol="http" />
= <Transport Hostname="was6host01" Port="9443" Protocol="https">
<Property Name="keyring" Value="C:\Program
  Files\IBM\HTTPServer\Plugins\config\webserver01\plugin-key.kdb" />
<Property Name="stashfile" Value="C:\Program
  Files\IBM\HTTPServer\Plugins\config\webserver01\plugin-key.sth" />
</Transport>
</Server>
= <Server CloneID="12cgrmd2k" ConnectTimeout="0" ExtendedHandshake="false"
  LoadBalanceWeight="2" MaxConnections="-1"
  Name="was61host02Node01_server1_on_was61host02" ServerIOTimeout="0"
  WaitForContinue="false">
<Transport Hostname="was61host02" Port="9081" Protocol="http" />
= <Transport Hostname="was61host02" Port="9444" Protocol="https">
<Property Name="keyring" Value="C:\Program
  Files\IBM\HTTPServer\Plugins\config\webserver01\plugin-key.kdb" />
<Property Name="stashfile" Value="C:\Program
  Files\IBM\HTTPServer\Plugins\config\webserver01\plugin-key.sth" />
</Transport>
</Server>
= <PrimaryServers>
<Server Name="was6host01Node01_server1" />
<Server Name="was61host02Node01_server1_on_was61host02" />
</PrimaryServers>
</ServerCluster>
= <UriGroup Name="default_host_TradeCluster_V2_URIs">
<Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="/snoop/*" />
<Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="/hello" />
<Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="/hitcount" />
<Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="*.jsp" />
```

```

<Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="*.jsw" />
<Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="*.jsw" />
<Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="/j_security_check" />
<Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid"
    Name="/ibm_security_logout" />
<Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="/servlet/*" />
<Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="/Trade/web/*" />
<Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="/QuoteWSRouter/*"
    />
</UriGroup>
<Route ServerCluster="TradeCluster_V2" UriGroup="default_host_TradeCluster_V2_URIs"
    VirtualHostGroup="default_host" />
- <RequestMetrics armEnabled="false" loggingEnabled="false" rmEnabled="false" traceLevel="HOPS">
- <filters enable="false" type="URI">
  <filterValues enable="false" value="/snoop" />
  <filterValues enable="false" value="/hitcount" />
</filters>
- <filters enable="false" type="SOURCE_IP">
  <filterValues enable="false" value="255.255.255.255" />
  <filterValues enable="false" value="254.254.254.254" />
</filters>
- <filters enable="false" type="JMS">
  <filterValues enable="false" value="destination=aaa" />
</filters>
- <filters enable="false" type="WEB_SERVICES">
  <filterValues enable="false" value="wsdlPort=aaa:op=bbb:nameSpace=ccc" />
</filters>
</RequestMetrics>
</Config>

```