

Metody zabezpečení dostupnosti MS SQL serveru 2005/2008

Methods of ensuring the availability of MS SQL Server 2005/2008

Pálka Lukáš, DiS.

Bakalářská práce
2010



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2009/2010

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Lukáš PÁLKA, DiS.**
Osobní číslo: **A07262**
Studijní program: **B 3902 Inženýrská informatika**
Studijní obor: **Informační a řídicí technologie**

Téma práce: **Metody zabezpečení dostupnosti MS SQL serveru 2005/2008**

Zásady pro vypracování:

1. Popište a proveďte instalaci a konfiguraci služby Microsoft Cluster services pro OS 2003/2008 S. Ed.
2. Nainstalujte a popište služby potřebné pro chod Cluster services
3. Nainstalujte a popište instalaci a konfiguraci MS SQL v cluster módu
4. Popište a realizujte služby mirroring databázi a Log Shipping v MS SQL
5. Popište a realizujte MS SQL pomocí virtualizace pod VMwarevSphere
6. Navrhněte a popište prostředky hardwaru spojené s vyšší dostupností služby MS SQL
7. Popište technologie redundance HW spojené s vyšší dostupností služeb a navrhněte jejich použití.
8. Popište a otestujte realizaci navržených technologií a proveďte měření dostupnosti při výpadku.
9. Porovnejte všechny uvedené metody vyšší dostupnosti a uveďte, kdy a jak je vhodné je použít, případně jak a proč je kombinovat.

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. OSIF, Michal. **Windows Server 2003**. 1. vyd. Praha : Grada Publishing a.s., 2003. 612 s. ISBN 80-247-0396-3. Kapitola 1 – Active Directory 2003, , s. 10-221.
2. RUSSEL, Charlie, CRAWFORD, Sharon, GEREND, Jason. **Windows Server 2003 : Velký průvodce administrátora**. Ana Rychetská. 1. vyd. Brno : CPBooks a.s., 2005. 1374 s. ISBN 80-251-0579-2. Kapitola 18 – Použití clusterů 2003, s. 545--577.
3. **Configuring ServerRAID controllers for clustering**. IBM Systems Software Information Center [online]. 2003 [cit. 2010-01-28]. Dostupný z WWW: http://publib.boulder.ibm.com/infocenter/eserver/v1r2/index.jsp?topic=/dirinfo/fqy0_acluscf
4. **IBM Totalstorage FastT Storage manager Version 9 Installation and Support Guide for Intel-based Operating System Environments : Installing Storage manager software in a standard Windows configuration** [online]. 2003 [cit. 2010-01-28]. Dostupný z WWW: ftp://ftp.software.ibm.com/systems/support/system_x.../gc26-7648-00.pdf
5. **Configuring MicrosoftCluster Service in a VMwarevSphere 4 Environment UsingInfortrend Storage** [online]. 2009 [cit. 2010-01-28]. Dostupný z WWW: http://www.infortrend.com/doc/appNote/APP_Note_MSACS%20in%20vSphere4.pdf
6. **Windows Server 2008 Instalace Active Directory** [online]. 2008 [cit. 2010-01-28]. Dostupný z WWW: <http://www.mstv.cz/it/videos/190/WS2008-serial-Instalace-Active-Directory>

Vedoucí bakalářské práce:

Ing. Petr Šilhavý, Ph.D.

Ústav počítačových a komunikačních systémů

Datum zadání bakalářské práce:

5. března 2010

Termín odevzdání bakalářské práce:

1. června 2010

Ve Zlíně dne 5. března 2010

prof. Ing. Vladimír Vašek, CSc.
děkan



doc. Ing. Ivan Zelinka, Ph.D.
ředitel ústavu



ABSTRAKT

Bakalářská práce se skládá z několika částí. První z nich seznamuje čtenáře s instalací a nasazením clusteringu. Další část práce se zabývá softwarovými nástroji SQL Serveru pro zvýšení dostupnosti a to zrcadlením databází a předáváním transakčních logů. Dále práce obsahuje i informace o zajištění vysoké dostupnosti pomocí hardwaru a to nástroji redundance a použitím technologií virtualizace. Poslední část práce se týká porovnání všech uvedených technologií, testy a vhodnou kombinací těchto technologií.

Klíčová slova: Clustering, Mirroring, Log Shipping, Virtualizace, HW Redundance

ABSTRACT

The thesis consists of several parts. The first of these introduces the reader to install and deploy clustering. Another part deals with the SQL Server software tools to improve the availability and mirroring of database and transaction log forwarding. Further work also includes information on high availability through hardware redundancy and tools and the use of virtualization technology. The last part concerns the comparison of all these technologies, tests and appropriate combination of these technologies.

Keywords: Clustering, Mirroring, Log Shipping, Virtualization, Hardware Redundancy

PODĚKOVÁNÍ

Tato bakalářská práce je výsledkem už šestileté praxe ve správě databázového serveru na Magistrátě města Přerova, dříve Městského úřadu. Proto bych chtěl poděkovat této instituci co se týká profesionálního růstu v oboru IT.

Rád bych touto cestou poděkoval i mému garantovi práce Ing. Petru Šilhavému, Ph. D.

Na závěr pak musím vyjádřit svůj dík veškerým lidem v mém okolí, kteří mě podporovali a tlačili dál ve studiu a samozřejmě k vypracování této bakalářské práce.

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, 25.5.2010

.....
podpis diplomanta

OBSAH

ÚVOD	11
I TEORETICKÁ ČÁST	12
1 VYSOKÁ DOSTUPNOST	13
1.1 PŘEHLED NÁSTROJŮ K ZAJIŠTĚNÍ VYSOKÉ DOSTUPNOSTI.....	13
2 HAVARIJNÍ CLUSTERING	15
3 ZRCADLENÍ (MIRRORING) DATABÁZÍ	16
3.1 JAK ZRCADLENÍ FUNGUJE	16
3.2 KOMUNIKACE.....	17
3.3 REŽIMY ZRCADLENÉ DATABÁZE.....	17
3.4 HIGH AVAILABILITY (VYSOKÁ DOSTUPNOST).....	18
3.4.1 Role Witness serveru.....	19
3.5 HIGH PERFORMANCE (VYSOKÝ VÝKON).....	19
3.6 HIGH SAFETY (VYSOKÁ BEZPEČNOST)	20
3.7 MIRROR VS CLUSTERING RAM	20
4 PŘEDÁVÁNÍ TRANSAKČNÍCH LOGŮ – LOG SHIPPING	21
4.1 ZÁKLADNÍ KOMPONENTY PŘEDÁVÁNÍ PROTOKOLŮ	21
5 REPLIKACE	22
5.1 PLATFORMA MS SQL SERVER PODPORUJE TŘI RŮZNÉ REPLIKAČNÍ MODELY	22
5.1.1 Replikace na snímcích.....	22
5.1.2 Replikace na transakcích.....	22
5.1.3 Slučované replikace	23
6 PROSTŘEDKY HARDWARU SPOJENÉ S VYŠŠÍ DOSTUPNOSTÍ SLUŽBY MS SQL	24
6.1 PŘÍPRAVA SERVERU, VÝBĚR HARDWARU SERVERŮ.....	24
7 TECHNOLOGIE REDUNDANCE HW SPOJENÉ S VYŠŠÍ DOSTUPNOSTÍ SLUŽEB	26
7.1 ÚVOD KAPITOLY REDUNDANCE	26
7.2 TECHNOLOGIE REDUNDANCE	26
7.2.1 Redundance Serverů.....	26
7.2.2 Redundance úložiště (polí).....	26
7.2.3 Redundance Switchů	26
7.2.4 Redundance cest.....	27
7.2.5 Redundance LAN.....	27
7.3 KDY A ZA JAKÝCH OKOLNOSTÍ REDUNDANCI POUŽÍT	27
7.3.1 Tabulka pravděpodobnosti výpadku	28
7.3.2 Tabulka pravděpodobnosti údržby	28
7.4 KDY POUŽÍT REDUNDANCI.....	29
7.4.1 Nejprve je vhodné začít s redundancí serveru.....	29
7.4.2 Redundance switchu a cest k externímu poli je také nesmírně důležitá	30
7.4.3 Redundance úložiště (polí).....	30
7.4.4 Redundance LAN.....	32

7.5	BĚŽNÉ CHYBY	33
II	PRAKTICKÁ ČÁST	34
8	POPIŠTE A PROVEĎTE INSTALACI A KONFIGURACI SLUŽBY MICROSOFT CLUSTER SERVICES PRO OS 2003 S. ED.....	35
8.1	HW POŽADAVKY	35
8.2	KONFIGURACE SÍŤOVÝCH KARET	35
8.3	POŽADAVKY NA ÚLOŽIŠTĚ	36
8.4	INSTALACE CLUSTER SLUŽBY POD WINDOWS 2003	36
8.5	KONFIGURACE SÍŤOVÝCH KARET	37
8.5.1	Konfigurace sítě	37
8.5.2	Konfigurace privátní sítě.....	37
8.5.3	Konfigurace veřejné sítě	38
8.6	VYTVOŘENÍ ÚČTU CLUSTERU	38
8.7	VYTVOŘENÍ A KONFIGURACE ŘÍDÍCÍHO DISKU PRO CLUSTER (QUORUM DISK)	39
8.8	INSTALACE A KONFIGURACE CLUSTER SLUŽBY	40
8.9	INSTALACE A KONFIGURACE PRVNÍHO NODU CLUSTERU	40
8.10	INSTALACE A KONFIGURACE DRUHÉHO NODU CLUSTERU	42
8.11	DOKONČENÍ CLUSTERINGU.....	43
8.12	NASTAVENÍ QUORUM DISKU	44
8.13	TEST CLUSTERU.....	44
8.14	DALŠÍM KROKEM JE KONTROLA ZDROJŮ CLUSTERU	45
9	POPIŠTE A PROVEĎTE INSTALACI A KONFIGURACI SLUŽBY MICROSOFT CLUSTER SERVICES PRO OS 2008 S. ED.....	46
9.1	PŘÍPRAVA INSTALACE FCM	46
9.2	KONFIGURACE SÍŤOVÝCH KARET	46
9.3	INSTALACE FAILOVER CLUSTER SLUŽBY POD WINDOWS 2008	47
9.3.1	Kroky instalace pomocí průvodce.....	47
9.4	KONFIGURACE FAILOVER CLUSTERU	49
9.4.1	Failover Cluster manager po konfiguračních úpravách	50
9.5	PŘIDÁNÍ FILE SERVERU DO FCM (SKLASTROVÁNÍ FILE SERVERU).....	50
10	INSTALACE A POPIS SLUŽEB ZAJIŠTUJÍCÍCH CHOD SLUŽBY CLUSTER SERVICES	52
10.1	INSTALACE ACTIVE DIRECTORY.....	52
10.1.1	Samotná instalace Active Directory.....	53
10.2	DNS.....	54
10.3	OSTATNÍ SLUŽBY A PROSTŘEDKY PRO CLUSTERING.....	54
11	INSTALACE A KONFIGURACE MS SQL V CLUSTER MÓDU – MSSQL 2005	55

11.1	INSTALACE SLUŽBY	55
11.2	PO-INSTALAČNÍ KROKY INSTALACE	59
11.3	OVĚŘENÍ CHODU CLUSTERU – TEST FUNKČNOSTI A REÁLNÁ MIGRACE SLUŽEB	61
11.4	INSTALACE SERVICE PACKU DO SQL 2005 V CLUSTER MODU	62
11.5	AKTIVE NODE CLUSTERING	62
11.6	SCHÉMA INSTALACE SQL 2005:	63
12	INSTALACE A KONFIGURACE MS SQL V CLUSTER MÓDU – MSSQL 2008	64
12.1	SCHÉMA INSTALACE.....	64
12.2	PŘED INSTALACÍ	65
12.3	INSTALACE DTC (MSDTC)	65
12.3.1	Instalace MSDTC	65
12.4	INSTALACE SQL SERVERU 2008 JAKO KLASTROVÝ VIRTUÁLNÍ ZDROJ.....	66
12.5	PŘIDÁNÍ DALŠÍHO NODU PRO SQL SERVER 2008	71
12.6	KONTROLA FUNKČNOSTI CLUSTERU	72
12.7	POST INSTALAČNÍ RUTINY	73
12.8	INSTALACE SERVICE PACKU NA SQL 2008 SE PROVÁDÍ NÁSLEDOVNĚ.....	73
13	ZRCADLENÍ DATABÁZÍ MS SQL (MIRRORING).....	74
13.1	PRAKTICKÁ REALIZACE POMOCÍ PRŮVODCE	74
13.2	POPIS INSTALACE	74
13.3	SCHÉMA INSTALACE MIRRORINGU.....	76
14	PŘEDÁVÁNÍ TRANSAKČNÍCH LOGŮ – LOG SHIPPING	77
14.1	POPIS INSTALACE	77
14.2	INSTALACE	77
14.3	KONFIGURACE.....	78
14.4	SCHÉMA INSTALACE LOG SHIPPINGU	80
15	REPLIKACE	82
15.1	NASAZENÍ REPLIKACE V PROVOZU	82
15.1.1	Vytvoření distributora	82
15.2	POSTUP INSTALACE	82
15.2.1	Schéma Instalace Replikace	84
16	PRAKTICKÁ REALIZACE SQL SERVERU POMOCÍ VIRTUALIZACE – STANDALONE VERZE	85
16.1	VYTVOŘENÍ VIRTUÁLNÍHO STROJE STAND-ALONE SLUŽBA STROJE	85
16.2	PRŮVODCE NÁS PROVEDE NÁSLEDUJÍCÍ KROKY	85
16.3	KONFIGURACE VIRTUÁLNÍ MAŠINY PRO STAND-ALONE SERVER	86
16.4	PROBLÉMY Z PRAXE O VIRTUALIZACI SQL SLUŽBY	88
17	PRAKTICKÁ REALIZACE SQL SERVERU POMOCÍ VIRTUALIZACE - CLUSTERED VERZE	89

17.1	VYTVORENÍ VIRTUÁLNÍHO STROJE CLUSTER VIRTUAL 2 NODY	89
17.2	KONFIGURACE VIRTUÁLNÍHO STROJE CLUSTER VIRTUAL 2 NODY	89
17.3	SCHÉMA INSTALACE	90
18	PROSTŘEDKY HARDWARU SPOJENÉ S VYŠŠÍ DOSTUPNOSTÍ SLUŽBY MS SQL	91
18.1	INSTALACE OPERAČNÍHO SYSTÉMU SERVERŮ	91
18.2	DISKOVÝ SUBSYSTÉM.....	91
18.3	RAID 1,5,10	92
18.3.1	RAID 1	93
18.3.2	RAID 5	93
18.3.3	RAID 10	94
18.4	KONFIGURACE A INICIALIZACE EXTERNÍCH POLÍ POMOCÍ MANAGEMENTU IBM	95
18.5	INSTALACE SERVERU POMOCÍ PRŮVODCE SETUP GUIDE OD IBM	96
19	REALIZACE NAVRŽENÝCH TECHNOLOGIÍ A MĚŘENÍ DOSTUPNOSTI PŘI VÝPADKU	97
19.1	TESTY.....	97
19.2	CLUSTERING.....	98
19.3	MIRRORING, LOG SHIPPING, REPLIKACE	99
19.4	VIRTUALIZACE (S VI CENTER TECHNOLOGIÍ).....	100
20	POROVNÁNÍ VŠECH UVEDENÝCH METOD VYŠŠÍ/VYSOKÉ DOSTUPNOSTI	102
20.1	TABULKA DOSTUPNOSTI - POROVNÁNÍ VŠECH UVEDENÝCH METOD	102
20.1.1	Metody zajišťující vysokou dostupnost z pohledu SW.....	102
20.1.2	Metody zajišťující vysokou dostupnost z pohledu HW	103
21	VÝSLEDEK A NÁVRH ŘEŠENÍ.....	104
21.1	KOMBINACE REDUNDANCE.....	104
21.1.1	Clustering v kombinaci zdvojení FC switchu a cest	104
21.1.2	Použití zrcadlení databází v kombinaci se svědkem a redundancí LAN ...	106
21.1.3	Třetí variantou je Virtualizovat jeden z nodu clusteru	107
21.1.4	Čtvrtou variantou je clustering se zrcadlením databází, redundancí cest a virtualizací.....	107
22	ZÁVĚR.....	109
	ZÁVĚR V ANGLIČTINĚ.....	110
	SEZNAM POUŽITÉ LITERATURY	111
22.1	KAPITOLY V KNIZE	111
22.2	ELEKTRONICKÉ ZDROJE (WWW)	111
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	113
	SEZNAM OBRÁZKŮ	114
	SEZNAM TABULEK.....	115
	SEZNAM PŘÍLOH.....	116

ÚVOD

Cílem práce je prozkoumat, popsat a vyhodnotit metody zabezpečení dostupnosti služby databázového serveru či serverů různými metodami jako jsou clustering, virtualizace, redundance, zrcadlení databází, replikace databází a předávání transakčních logů nad SQL službou firmy Microsoft.

Úkolem je prozkoumat a popsat celou infrastrukturu zabývající se dostupností databázové služby jako takové vůči stanicím a aplikacím.

Práce je rozdělena na oddělené části zabývající se konkrétní problematikou.

První kapitola této práce se zabývá teoretickou částí, kde každé řešení vysoké dostupnosti popíšeme z hlediska fungování té dané metody a dozvíme se více o jejich možnostech nasazení. Dále se v této kapitole zabýváme návrhem jak postupovat při výběru hardwaru a technologií zajišťující chod MS SQL Serveru.

Další kapitolou je praktická část, která popisuje instalaci a konfiguraci všech metod, které zabezpečují vysokou dostupnost služeb. Práce se zabývá popisem instalace a konfigurace každé výše zmíněné metody a podrobným popisem podpůrných technologií redundance. V technické části kapitoly otestujeme všechny prostředky a technologie vysoké dostupnosti, porovnáme je a vypíchneme výhody a nevýhody.

V závěru bakalářské práce, navrhne konkrétní řešení a dostaneme se ke kombinaci metod zajišťující chod SQL Serveru. V rámci celé infrastruktury je popsáno, jak použít softwarové nástroje, tak i hardwarové nástroje, které nám zajistí dostupnost služeb i v případě poruchy hardwaru. Jednou z metod zamezení poruchy hardwaru je například řešení za pomoci technologií redundance.

Bakalářská práce lze použít jako návod pro budování infrastruktury zabývající se databázovými servery v organizaci, ale také, jak začlenit službu SQL Serveru do stávajícího prostředí.

Zejména je rozebrána technologie clusteringu a to jak v prostředí OS 2003/2008, tak i služby MS SQL 2005/2008. Clustering metoda k zajištění vysoké dostupnosti databázového serveru se v praxi používá nejčastěji a paradoxně je i nejvíce komplikovaná.

Co se týká virtualizace, je to nový rostoucí trend, jak vyřešit řadu problémů v případě havárie jednoho ze serverů a i když se nejedná o standardní metodu z pohledu služeb SQL Serveru firmy Microsoft, začlenili jsme i tuto technologii do bakalářské práce.

I. TEORETICKÁ ČÁST

1 VYSOKÁ DOSTUPNOST

Vysoká dostupnost je systém pro zajištění co největší dostupnosti služeb, které servery nabízejí. Navzdory veškerému plánování musíme počítat s tím, že někdy dojde k havárii či chybě systému a služba databázového serveru nebude dostupná.

K tomu, abychom zajistili co nejvyšší dostupnost služby SQL Serveru, můžeme použít dnes už pět možných způsobů, které nám pomohou zautomatizovat nežádoucí stavy systémů a automaticky je uvést do takového stavu, kdy se opět stanou dostupnými.

Takové systémy pak můžeme nazývat odolnými systémy pro nečekané havárie, chyby a kolize.

Nástroje, které nám dovedou zajistit vyšší, nebo chceme-li, vysokou dostupnost se nazývají:

- Havarijní clustering
- Zrcadlení databází
- Replikace databází
- Předávání transakčních logů
- Virtualizace

1.1 Přehled nástrojů k zajištění vysoké dostupnosti

Havarijní Clustering [3]

Havarijní clustering je systém postaven na clusteringu operačního systému. Díky této funkci lze instalovat danou službu či instanci na více serverů a uzlů. V případě nečekané chyby softwaru nebo havárie hardwaru nám clustering zajistí převzetí klastrové služby automaticky jiným serverem. Tím docílíme dostupnost služeb v maximální možné míře s minimálním výpadkem. Tento systém řeší chyby serverů, nejsou zde ale řešeny havárie úložišť dat, které jsou umístěny vždy odděleně od operačního systému.

Zrcadlení databází [3]

Zrcadlení databází je model zajištění vysoké dostupnosti, kde se řeší totální havárie úložiště, nedostupnost serverů a prostředků ke komunikaci se servery. Cílem zrcadlit databáze je zvýšit dostupnost a zároveň zabezpečit odolnost vůči výpadkům systémů, způsobených selháním zejména hardwaru, či kolapsu sítě. Zrcadlení databází je softwarové

řešení, založené na komunikaci SQL Serverů pomocí předávání takzvaných transakčních logů.

Replikace databází [3]

Replikace databází je obdobou zrcadlení databází, ale na úplně jiném principu. Replikace pracuje na přenosu transakcí s daty. Tyto transakce se přenášejí na všechny replikované uzly pro danou databázi. Replikace se ve stručnosti dá charakterizovat jako mechanismus distribuce dat na nejzákladnější úrovni s použitím synchronizace.

Předávání transakčních logů – Log Shipping [3]

Předávání transakčních protokolů je metoda přenosu dat zakomponována od samého vzniku SQL Serveru. Cílem předávání protokolů je zachovat alespoň jednu kopii databáze, abychom zajistili ochranu této databáze před havárií hardwaru, nebo poškozením dat. Jedná se o nejzákladnější myšlenku vyšší dostupnosti, kdy předáváme v jednom sledu řetězec transakčního logu databáze jiné instanci SQL Serveru. Zatímco zrcadlení databází pracuje na principu vytvoření právě jedné kopie dané databáze, tak u transakčního modelu můžeme kopií vytvořit mnohem více.

Technologie Virtualizace

Virtualizace znamená vytvoření virtuálního počítače uvnitř skutečného počítače.

Jinými slovy lze říci, že pomocí virtualizace jsme schopni jeden zdroj (server, případně jeho části – procesor, paměť, síťová karta, datové úložiště) využít pro více než jeden operační systém.

Systém řízení těchto virtuálních počítačů nám umožňuje dosáhnout vyšší dostupnosti a spolehlivosti pro případ výpadku serverů nebo havárii hardwaru.

2 HAVARIJNÍ CLUSTERING

Havarijní clustering systému SQL Server 2005/2008 je založen na Clustering Services, který zajišťuje ochranu jak databázové služby, tak i ostatních služeb systému a v neposlední řadě i Operačního systému samotného.

Instalace služby SQL Serveru bude proto instalována nad dvěma servery v takzvaném cluster balancingu, která běží nad službou Cluster Services.

Od verze SQL Serveru 2005 jsou použity technologie clusteringu i pro služby jako jsou Analysis Services a Notification Services, které jsou součástí SQL Serveru.

Verze SQL Serveru Standard Edition podporuje dva nody clusteru. Verze 2005 Enterprise podporuje až osm uzlů a verze 2008 Enterprise Edition dokonce šestnáct uzlů.

Podrobný popis instalace a konfigurace naleznete v praktické části této bakalářské práce.

3 ZRCADLENÍ (MIRRORING) DATABÁZÍ

Zrcadlení databáze je prostředek vysoké dostupnosti, kdy druhou kopii databáze uchováваме na jiné vybudované infrastruktuře (serveru s externím polem). Je to silná metoda k ochraně dat a velice významná, kdy bereme v potaz všechny možná výpadky jak na straně HW, tak i SW.

Zrcadlení databází je tedy softwarové řešení pro zvýšení dostupnosti databází.

Technologie zrcadlení umožňuje vytvořit přesnou kopii databáze na jiném serveru.

Zrcadlené databáze pracují pouze v režimu s plnou obnovou databáze (FULL RECOVERY MODEL), jelikož pro zrcadlení se používá transakční log.

Zrcadlit se dají pouze uživatelské databáze a to jak ve verzi SQL Serveru Enterprise tak i Standard.

Zrcadlení databází nelze provést pod jednou instancí SQL serveru. Vždy jsou potřeba oddělené instance SQL Serveru. Pod jedním serverem by teoreticky mirroring rozchodit šel, ale nic to neřeší v problému vysoké dostupnosti.

Databázové zrcadlení poskytuje úplnou, nebo téměř úplnou redundanci dat v závislosti na provozních režimech a jednoznačně zvyšuje dostupnost mirrorovaných databází.

3.1 Jak zrcadlení funguje [3]

V této kapitole budeme hovořit o třech typech serveru z pohledu mirroringu ve vazbě na databázi. Je to Principal server, Mirror server a Witness server.

Principal je hlavní, řídicí databáze, dalo by se říci aktivní databáze.

Mirror je zrcadlená databáze, která je obrazem principal databáze.

Witness je řídicí člen, který definuje, která databáze je hlavní a která zrcadlo. Tento řídicí člen řídí transakce a předávání dat zrcadlení mezi servery.

Vždy je třeba jednu databázi nastavit jako hlavní (principal) a druhou databázi (FULL backup) do role mirror.

Role svědka (Witness) řídí automatické převzetí služeb a to zrcadlenou databází, která se stane hlavní (Principal) databází, v případě havárie principal databáze.

Principal a Mirror role jsou povinné, Witness role je volitelná.

3.2 Komunikace

Každá transakce, která je směřována do databáze (principal db) je směřována automaticky do zrcadlené (mirror) databáze. Naopak mirror databáze je databáze, která přijímá všechny transakce aplikované v řídicí (principal) databázi.

Proces zrcadlení databáze přesunuje záznamy z transakčního protokolu z řídicí databáze do zrcadlené, která na tomto základě bude obsahovat stejné záznamy v tabulkách, jako hlavní databáze.

Zrcadlená databáze je ve stavu obnovování (restore) a proto nepřijímá žádné požadavky a proto neprovádí žádné transakce ze strany klientů. Ze zrcadlené databáze lze však vytvořit snímek databáze, který poskytne klientům přístup pro čtení.

Hlavní i zrcadlená role jsou ve stavu, kdy si mohou své stavy prohodit, protože obě databáze jsou naprosto shodné a udržují si navzájem konzistenci dat pomocí synchronizace. Proto se v libovolném okamžiku může kterákoliv databáze stát hlavní. Proces je řízen buď Witness serverem, nebo administrátorským zásahem, který rozhodne, která databáze je v jaké stavu, zda principal, či mirror.

Role svědka Witness je volitelná, leč významná část této technologie, která řídí celý proces předávání dat a v případě výpadku je to právě ona, která rozhodne, že dojde k záměně rolí u principal a mirror serverů. Witness roli proto můžeme chápat jako rozhodující faktor který určí, který ze serverů bude aktivní instance. Witness funguje na principu potvrzování událostí a řízených instrukcí.

Hlavní databáze nemůže mít více jak jedno zrcadlo a naopak.

3.3 Režimy zrcadlené databáze [3]

Synchronní režim: Tento režim pracuje v režimu vysoké bezpečnosti.

Asynchronní režim: Režim bez použití zachování stejného obrazu v reálném čase.

Zrcadlení databáze umožňuje chod ve třech různých konfiguracích:

High Availability (Vysoká dostupnost)

High Performance (Vysoký výkon)

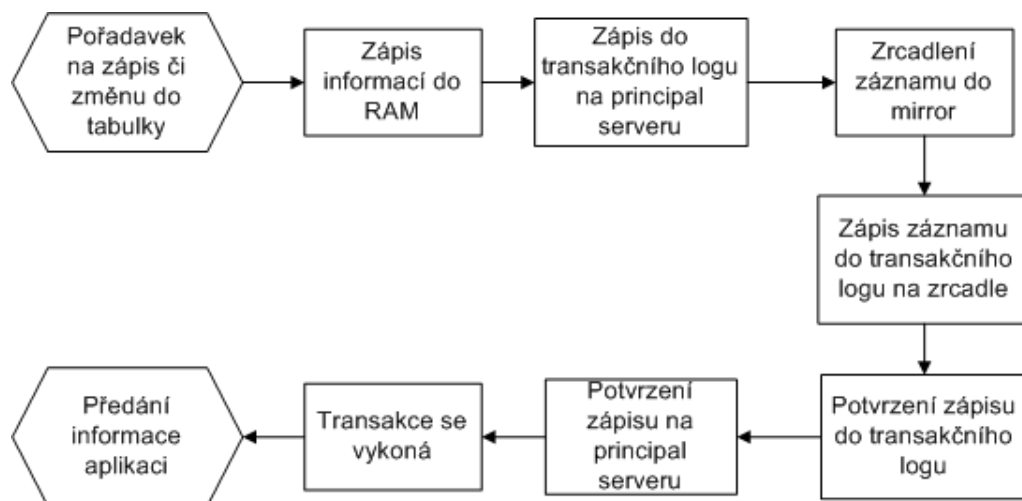
High Safety (Vysoká bezpečnost)

3.4 High Availability (Vysoká dostupnost) [3]

Tento režim zajišťuje odolný synchronní přenos mezi hlavní a zrcadlenou databází.

Je zde kontrola detekce selhání a automatické převzetí služby při havárii.

Pracujeme zde s potvrzováním veškerých transakcí. Proces potvrzování je proces, kdy se daná transakce nejdříve zapíše do vyrovnávací paměti (RAM) ve svém paměťovém bloku, SQL služba následně zaznamená (zapíše) transakci do transakčního logu na hlavní (principal) databázi. Až je transakce zapsaná do transakčního logu, dojde k zrcadlení databáze přenosem nové části transakčního protokolu (například řádku s transakcí) do transakčního logu zrcadlené databáze. Následně dojde k potvrzení zápisu transakce na zrcadle a dojde k potvrzení události, že se operace zdařila. Tato informace se pošle zpět do hlavní databáze, kde se transakce opět potvrdí jako provedená. Po vydání potvrzení v hlavní databázi je operace fyzicky vykonána (sql operace typu transakce sql) a poslána zpět do aplikace, která dala podnět k vykonání činnosti. Aplikace poté pokračuje ve zpracování výsledku sql operace. Tento proces zaručuje, že než je informace o potvrzení vrácena aplikaci, jsou všechny transakce potvrzeny a uloženy do transakčního protokolu nad oběma databázemi. Hlavní i zrcadlená databáze proto vždy budou obsahovat stejná data.



Obr. 1: Princip komunikace mirroringu

Oproti jiným technologiím vysoké dostupnosti tato řešení fyzicky přenášejí záznamy transakčního protokolu tak, jak jsou zapisovány. U technologie předávání protokolů a replikace je tento proces daleko složitější, protože je třeba vyčkat na dokončení transakce a až po té vyhodnotit, zda se proces provedl správně, či nikoliv. Proto je zrcadlení vhodné

použít především tam, kde pracujeme s velkým počtem řádků v transakci, protože to nebude mít dopad na počet synchronizačních přenosů. Vliv se snižuje proto, že při větší velikosti transakcí nedochází ke zvětšování potvrzovacích pokynů. Naopak procento potvrzování je pak nízké oproti počtu operací v transakci.

3.4.1 Role Witness serveru

Pomocí této role zde vstupuje další prvek řízení, kdy je neustále kontrolován stav hlavní databáze pomocí detekování chyb a zajišťuje havarijní převzetí služby zrcadlenou databází (serverem).

Pokud dojde k havarijnímu stavu, Witness role nastaví u zrcadla stav na hlavní a po opětovném chodu hlavní databáze je stav nastaven jako mirror role a to vše plně automaticky.

Witness server tedy obrací role hlavní a zrcadlené databáze. Zrcadlená databáze je povýšena na hlavní databázi. Následně sníží úroveň hlavní databáze na roli zrcadla a je otočena komunikace transakcí tak, aby bylo zachováno předávání informací hlavní db na zrcadlenou db.

Oproti technologiím replikace a předávání protokolů vše proběhne automaticky bez nutnosti zásahu administrátora, či změny konfiguračních souborů.

Dříve, než server svědka (witness server) zajistí výměnu rolí, je třeba, aby s danou operací souhlasily dvě ze tří rolí mirroringu. Tím se ověří, že došlo k výpadku pouze jednoho serveru (role). Řídící prostředek odborně nazýváme Quorum.

Vždy před výměnou rolí se automaticky ověří, zda se zrcadlená role může připojit k serveru svědka a dosáhnout tak kvóra.

3.5 High Performance (vysoký výkon) [3]

V tomto modelu vysoké dostupnosti se nepoužívá role svědka. Tento režim provozu poskytuje připravenou pohotovostní konfiguraci, která ale nepodporuje detekci chyb, neumožňuje automatické převzetí služeb a rovněž negarantuje pravý obraz databází v reálném čase.

Data na zrcadlenou databázi jsou posílána asynchronně, proto nelze tento model použít pro automatické převzetí služby. Transakce jsou posílány na zrcadlenou databázi se

zpožděním, proto nelze zaručit, že v okamžiku výpadku zrcadlo již přijalo všechny transakce a že je databáze naprosto shodná s hlavní databází.

Výhodou je fakt, že zrcadlená databáze díky asynchronnímu provozu může být vzdálena od hlavního serveru a rovněž zde odpadá vliv synchronizačních procesů na výkon databázového serveru.

Tento model nezajišťuje stoprocentní shodu dat při výpadku. Proto může dojít ke ztrátě právě probíhajících transakcí i při vynucení havarijního stavu administrátorem.

3.6 High Safety (Vysoká bezpečnost) [3]

Přenos transakčních záznamů je přenášen synchronně, takže je zaručena shodnost obrazu v reálném čase, ovšem oproti modelu vysoké dostupnosti bez nutnosti použití role svědka.

Opět zde platí, že první jsou potvrzeny transakce na úrovni zrcadla a až poté je zpětně potvrzeno zapsání transakce v hlavní databázi.

Jelikož se zde nepoužívá role svědka, není možné zajistit automatické převzetí služby v případě výpadku hlavní databáze zrcadlenou databází. Pokud dojde k havárii hlavní databáze, můžeme zde ručně zvýšit úroveň zrcadla na úroveň hlavní databáze.

Pokud operaci převzetí hlavní role provedeme, SQL server odpojí všechna připojení, dokončí probíhající transakce a znemožní navázání či vytvoření nových transakcí. Po dokončení všech běžících transakcí dojde k odeslání transakčního logu na server s rolí zrcadla a následně se role zrcadla a hlavní databáze vymění. Jakmile dojde k výměně, SQL server odblokuje službu pro příjem transakcí a aplikace či klienti mohou pokračovat v posílání transakcí na server. Ten je běžným způsobem začne zpracovávat a chod SQL služby je zajištěn.

3.7 Mirror vs clustering RAM

Pokud porovnáme technologii mirroringu a clusteringu po stránce operací v RAM paměti, tak clustering na pasivním nodu nezaznamenává žádné indexy a nenačítá a nealokuje si žádnou paměť pro jakékoliv transakce. Naopak mirroring transakce na zrcadle alokuje, takže pokud dojde k přehození rolí, služba SQL mnohem rychleji reaguje na transakce, jelikož řada informací je již obsažena v paměti RAM a není třeba přistupovat k datům na disk. Poškozené stránky řeší mirroring pomocí obrazu z mirroru druhého serveru oproti jiné technologii.

4 PŘEDÁVÁNÍ TRANSAKČNÍCH LOGŮ – LOG SHIPPING

Log Shipping je technologie vysoké dostupnosti, založená na udržování aspoň jedné kopie databáze, aby byla zajištěna její ochrana ať už chybou hardwaru, či softwaru.

Princip je hodně podobný s technologií zrcadlení, protože se předávají řetězce transakčního protokolu databáze jiné databázi.

Zásadní rozdíl je v tom, že mirroring umožňuje pořízení pouze jednoho obrazu, ale předávání transakčního logu je technologie, kdy získáme více kopií a tím i zvýšení redundance dat.

4.1 Základní komponenty předávání protokolů [3]

Master databáze (primární)

Slave databáze (sekundární)

Search databáze (sledování)

Primární databáze: Stejně jako běžná databáze udržuje a poskytuje svá data klientům a aplikacím. Navíc zde dochází k automatickému vytváření zálohy transakčního protokolu, který následně primary server kopíruje na sekundární server.

Sekundární databáze: nejedná se o běžnou databázi a proto není ani dostupná pro aplikace. Obnovují se do ní v pravidelných intervalech zálohy z primární databáze. Databáze se může nacházet ve dvou stavech.

Standby: speciální stav, kdy k databázi mohou přistupovat stanice a aplikace a data jsou zde pouze pro čtení. (Pokud jsou v databázi přítomny aktivní konekce, nelze databázi obnovit z transakčních logů).

NoRecovery: výchozí nastavení databáze, kdy databáze není dostupná a plní funkci pouze obnovovacího algoritmu.

Sledovací databáze: Je to nepovinná role, stejně jako Witness režim při technologii zrcadlení a úloha tohoto serveru je dozírat na stav synchronizace a řídit tuto synchronizaci.

5 REPLIKACE

Replikace je rovněž jeden ze způsobů zajištění vysoké dostupnosti. Replikaci lze zařadit mezi metody vyšší dostupnosti předávání transakčního logu nebo mirroringu. Řada principů je zde více či méně podobná, proto i tuhle technologii rozebereme do detailu.

Replikace databází, nebo chceme-li replikace tabulek, je už roky zaběhlá technologie pod MS SQL serverem. Už ve verzi SQL 2000 jsme se mohli setkat s tímto řešením, ale my se budeme zabývat replikací na MS SQL 2005/2008. Tato technologie je na MS vcelku propracovaná a patří k těm lepším replikacím.

5.1 Platforma MS SQL Server podporuje tři různé replikační modely

[3]

Replikace na snímcích (Snapshot Replication)

Replikace na transakcích (Transaction Replication)

Slučované replikace (Merge Replication)

5.1.1 Replikace na snímcích

Tento model je založen na kopírování dat platných v daném okamžiku. Princip replikace je proto velice jednoduchý, protože zde není třeba sledovat změny zdrojových dat. Požádání o data proběhne na základě požadavku druhé strany, nebo pomocí plánovaných úloh, kdy data porovnáme se zdrojem, zda se změnila a přenáší se pouze novější záznamy v tabulce. Proto se tento systém replikace nazývá snímkovací, protože vyhodnocujeme a kopírujeme snímky změn. Tento typ replikace je vhodné použít jen tehdy, pokud víme, že se nám data v databázi či v tabulce nebudou měnit moc často.

5.1.2 Replikace na transakcích

Jak název napoví, u tohoto modelu se budou data replikovat podle změn uvedených v transakcích. Oproti zrcadlení zde ale nedochází k přímému zapisování dat do dvou databází, ale transakce pro replikaci sbírají a v ucelených balících či intervalech se odesílají do repliky. Opět toto řešení přináší řadu výhod a nevýhod. Pokud se nám často bude měnit jedna hodnota v tabulce, budou narůstat nároky na síťový provoz a dostupnost repliky. Výhodou je aktuálnost dat a možnost zapojení více replik a tímto zapojení více serverů.

5.1.3 Slučované replikace

Tento model je poměrně nejsložitější, co se týká funkcionality. Princip replikace je založen na počátečním pořízení kopie zdrojových dat a následnou replikací dat na základě změn v datech. Aktualizace dat se provádí v pravidelných naplánovaných časových intervalech a nebo na základě požadavků repliky. Tento model se nejčastěji používá tam, kde není možné neustále zajistit komunikaci mezi zdrojovou databází a replikou. Podobné replikování je v praxi běžné u synchronizace kontaktů či emailu v mail serverech a podobně. Výhoda tohoto řešení je pořizování slučovacích zdrojů dat, kdy se opravdu posílají jen data aktuální a data pouze změněná. Oproti transakční replikaci, kdy posíláme neustále změnu jedné buňky, zde dojde k přenosu jen poslední aktuální informace. Oproti snímkovací replice zde nepřenášíme celé balíky tabulek, ale opět jen ty, které se změnilo. Proto je tato aplikační logika nejsložitější ze všech systémů vysoké dostupnosti. Výhodou je i malé množství přenášených dat a aktuálnost dat v replice. Nevýhodou je možnost konfliktu transakcí, které zde nečekají na potvrzení.

6 PROSTŘEDKY HARDWARU SPOJENÉ S VYŠŠÍ DOSTUPNOSTÍ SLUŽBY MS SQL

6.1 Příprava Serveru, výběr hardwaru serverů

Nejspíše nejdůležitější a nejtěžší krok celé této problematiky. To, jaké HW prostředky pořídíme, se nám přímo projeví v samotném řešení celého návrhu vysoké dostupnosti. Jednoznačně doporučuji koupit robustní řešení dané platformy. Server je stroj, který pojede nonstop a každá špatně ušetřená koruna se projeví dvojnásob a náklady na zpětný upgrade jsou mnohem vyšší, než pořídít daný stroj hned na počátku.

Zejména se to týká externích polí a jejich pojetí řadičů. Integrované diskové řadiče doporučuji rovnou zavrhnout. Papírově mají stejné parametry jako externí, ovšem I/O operace a veškeré práce s takovými poli (přerozdělování disků, výměny disků, diagnostické utility, případně řešení jakýchkoliv problémů a firmwary), je jeden veliký problém. Interní řadič disků je vhodné mít pouze na systémový oddíl OS, na který použijeme RAID1.

Paměť RAM, jednoznačně čím více, tím lépe. Dle nastavení SQL služby nám paměť sníží požadavky I/O operací na disky a tím se nám několikanásobně zrychlí veškeré dotazy a operace nad SQL. Musíme si uvědomit, že operace I/O, tzn. diskové operace, jsou nejužší místo, co se týká rychlosti. Co se týká velikosti paměti RAM vzhledem k velikosti databází, nelze přesně říci, kolik paměti budeme potřebovat. Je zde potřeba brát na zřetel samotné operace nad danými databázemi (transakční zpracování dat, indexování apod.) Velikost RAM má vliv i nad výběrem OS a SQL ve verzi 32b či 64b. Jednoznačně doporučuji 64b. Je zde vyšší výkon SQL a netrápí nás limit 4GB, respektive 6GB paměti. 6GB pouze pomocí zdvojené alokace paměti s parametry v boot.ini a verzi OS Enterprise Edition. Standard tuhle funkci nemá.

Výběr CPU je rovněž důležitý, ale v problematice SQL serveru ne tolik jako rozhodnout, jaké diskové pole a kolik paměti. Platí, že životnost CPU a chipsetu je jednou taková, jako životnost velikosti RAM a diskové kapacity. Sporné je i jak má být základní deska rozšiřitelná. V dnešní době dokupovat procesory a paměť např. po 4 letech chodu serveru je na pováženou vzhledem k ceně takovýchto komponent vzhledem k výkonu, dostupnosti těchto komponent a záruce. Když už tedy upgrade serveru, tak co nejdříve a jednoznačně značkovými komponenty od stejného výrobce, jako je server samotný.

Co se týká CPU, tak je zde další zásadní věc, a to licence. Pokud nás zajímá licencování per CPU, mluví se zde o licencování na patici. Tzn. V dnešní době koupíme serverový CPU i se 4 či 6 jádry. Doporučuji minimálně 2 CPU. Musíme myslet na to, že zde máme operace nad I/O a operace nad transakcemi. Určitě se vyplatí některá jádra přiřadit pouze daným operacím, to ale níže v kapitole o konfiguraci MSSQL Serveru.

Důležité je i rozhodnutí, kolik nodů budeme potřebovat. Pokud více jak dva, mluvíme o MSSQL Enterprise Edition. Pokud dva, tak stejné stroje, či nikoliv? V kapitole Virtualizace MSSQL Serveru budeme hovořit o virtualizaci této služby, ale tento hit dnešní doby není zrovna vhodný pro SQL službu jako takovou. Tahle věta platí pro danou dobu, tzn. rok 2010. Jakým trendem se bude virtualizace vyvíjet a zda zasáhne výrazně i do této problematiky, se jen těžko odhaduje. Vráťím se k počtu nodu, pokud tedy dva, určitě doporučuji stejné oba servery nad jedním společným diskovým prostorem. Můžeme, a je to velice výhodné, použít více diskových polí a oddílů, už kvůli rychlosti. U více nodu jak dvou, bych virtualizaci klidně doporučil. Praxe ukazuje, že jednou za čas bude třeba celý cluster vypnout, ať už jde o firmware HW, či změnu koncepce. Na toto je vhodné virtualizaci použít, protože se jedná o dočasný provoz, kde nás nemusí trápit snížení výkonu a zpomalení operací nad db.

Z historických důvodů doporučuji zvážit koupit do serveru i FDD mechaniku, zejména nad OS 2003 Server, který neumí mapovat flashdisk jako FDD při instalaci OS v sekci instalaci driveru řadiče disků. Firmwary biosu, polí, disků, switchu a podobně, se většinou dělají přes bootovací CD, takže v tomhle směru problém mít nejspíše nebudeme.

7 TECHNOLOGIE REDUNDANCE HW SPOJENÉ S VYŠŠÍ DOSTUPNOSTÍ SLUŽEB

7.1 Úvod kapitoly Redundance

Redundance jsou prostředky hardwaru, které jsou použity duplicitně tam, kde předpokládáme nejčastější možné chyby hardwaru, či totální havárie operačních systémů.

Redundanci lze použít jak pro výpadky hardwaru, tak i softwaru. Pro výpadky HW to jsou prostředky jako jsou switche, úložiště, či síťové prostředky. Pro výpadky SW to jsou servery, které hlídají dostupnost služby, nebo replikují svá data pro případ havárie hlavního serveru.

Redundanci můžeme chápat i jako technologii, která používá jeden aktivní a minimálně jeden pasivní člen (prvek), který čeká na výpadek členu aktivního.

7.2 Technologie redundance

7.2.1 Redundance Serverů

Pod tímto pojmem můžeme chápat použití více serverů. Jeden server je aktivní a další servery tvoří pasivní část s tím, že ověřují chod aktivního serveru. V případě výpadku či havárie přebírá pasivní prvek funkci aktivního a to i v případě, že původní server se dostane do funkčního stavu. O tomto systému mluvíme jako o pasivním clusteringu. Podrobným popisem clusteringu se zabývá jiná kapitola.

7.2.2 Redundance úložiště (polí)

Replikují se zde fyzická data SQL, či jiné služby. Redundance probíhá na nejnižší vrstvě tzn. řídicí prvek zde tvoří řadič polí (controller) a redundance, neboli mirror polí se děje bez použití serveru. Tato redundance se chová tak, že co se stane na jednom poli, to se v reálném čase stane i na druhém. Takže pokud dojde k chybě softwaru, která způsobí chybu na oddíle s daty, tak nám tento mirror úložiště neochrání.

7.2.3 Redundance Switchů

Jedná se o duplicitní připojení diskových úložišť k serveru. V případě výpadku některého ze switchů dojde k automatickému předání úlohy redundantnímu switchu, který bez ztráty dat dokončí požadavek původního switchu. Jedná se tedy o redundantní komunikaci, kdy

jeden pracuje v režimu aktiv a druhý pasiv. Koordinace a kontrola probíhá opět na hardwarové úrovni, kterou si vyhodnocují samy switche bez přímé kontroly serverů. Servery se s touto událostí nemusí zatěžovat, protože netuší, že došlo k nějaké havárii.

7.2.4 Redundance cest

Zde mluvíme o použití redundance v přímé komunikaci server – switch. V serveru je použito více komunikačních karet. V našem případě FC (Fibrechannel) karty. Co karta, to jedna cesta ke switchi. Propoj je u FC karet realizován pomocí optických kabelů. Takováto redundance přes switche lze použít pouze u technologie SERVER – SWITCH – POLE. Pokud máme SCSI karty v serverech, které komunikují přímo s úložištěm (polem), není zde možné redundanci realizovat, protože nejsme schopni uřídit přenosy dat s řízením, kdy se jaká cesta k úložišti použije.

7.2.5 Redundance LAN

Redundance LAN je dostupnost služby směrem ke klientovi. Je to komunikace mezi client – LAN switch – server. Můžeme zde použít i technologii zvanou trunkované linky, což je technologie spojování více komunikačních prvků do jednoho. Nejčastěji trunkování LAN kabelů za účelem jak vysoké dostupnosti, tak i zvyšování přenosové rychlosti. Do redundance LAN chápeme i použití více síťových karet pro komunikaci, a to jak při použití clusteringu, tak i standalone serveru.

Do redundancí můžeme zahrnout i použití technologií v kombinaci s mirroringem, či použitím náhradních serverů. Jakékoliv řešení, které hlídá stav primárního systému, můžeme chápat jako sekundární systém redundance.

7.3 Kdy a za jakých okolností redundanci použít

Budeme vycházet z technologií, které jsou použity pro redundanci.

Jelikož redundanci můžeme jakkoliv kombinovat, je na zváženu, kterou z technologií redundance použijeme a jak je zkombinujeme.

Dobré je si uvědomit, kde dochází k nejvíce haváriím a pak rozhodnout, jakou cestou tyto havárie řešit. Asi nemá smysl řešit havárie teoretické, které téměř nikdy nenastanou. Takové redundance jsou zbytečné a je to plýtvání penězi.

7.3.1 Tabulka pravděpodobnosti výpadku

Serveru	60%
Úložiště dat (externí pole) *	5%
FC Switch	7%
Datová cesta k poli server – switch – pole	15%
LAN – směr ke klientovi (kabely, centrální switche)	13%

Tab. 1: Pravděpodobnost výpadku

* U úložiště dat počítáme s kompletní havárií úložiště. Havárie disků v poli řeší RAID technologie a vzhledem ke kapitole Redundance budeme tento fakt akceptovat. Redundanci je třeba chápat na daný prvek, který chceme duplikovat. Disky v poli na RAID technologii redundanci používají a proto pravděpodobnost výpadku je tak malá. V případě disků by jsme mluvili o pravděpodobnosti výpadku okolo 25% vzhledem k celému řešení.

Rovněž doporučuji externí pole připojovat dle dokumentace k externím řadičům popsané v dokumentaci. Řada serverů má svůj interní řadič, ovšem ten zvyšuje možnost havárie jak pole, tak i disků v poli.

Z naší tabulky vyplývá, že k nejvíce výpadkům dochází právě na serverech. Důvodem je použití služeb operačního systému a samotný provoz operačního systému.

7.3.2 Tabulka pravděpodobnosti údržby

Do údržby chápeme instalace aktualizací, instalace záplat systémů, firmware zařízení, či diagnostiku při potížích.

Je to stav, kdy zařízení není ve stavu výpadku, ale je s ním potřeba učinit nějakou nestandardní operaci, při které nemůže plnit svou běžnou funkci.

Serveru	76%
Úložiště dat (externí pole)	7%
FC Switch	7%
Datová cesta k poli server – switch – pole *	5%
LAN – směr ke klientovi (kabely, centrální switche)	5%

Tab. 2: Pravděpodobnost údržby

* U datových cest údržbu chápeme jako přeměření spojení a vyhodnocení, zda je třeba datové kabely vyměnit či nikoliv, případně je opravit. (Twist-kabely, FC kabely).

Z výše uvedených dvou tabulek je zřejmé, kde je vhodné redundanci nasadit. Jelikož jde redundance kombinovat, je vhodné zvolit následující postup.

7.4 Kdy použít redundanci

7.4.1 Nejprve je vhodné začít s redundancí serveru

Redundance serveru: máme dvě možnosti, buď použijeme clustering serveru, nebo můžeme použít mirroring (zrcadlení) databází.

Pokud použijeme clustering, tak lze použít pasivní nebo aktivní režim. Pasivní od aktivního se liší tak, že u aktivního clusteringu jsou 2 SQL servery, každý nainstalován pod jinou instancí (default 1433, druhá instance jiný port). Takový aktivní clustering, spojený s SQL službou, vystupuje vůči klientům pod jednou IP, ale jiným portem.

Nejpoužívanější je pasivní clustering. Instalaci jsme popsali v kapitole o clusteringu.

Výhody či nevýhody aktivního či pasivního clusteringu už chápeme, ale je třeba myslet na licence. Při aktivním clusteringu potřebujeme jednou tolik licencí pro chod SQL služby a to jak v případě licencování per-CPU, nebo per-USERS.

Druhou volbou je zrcadlení databází. Zde rovněž potřebujeme dvojí licencování, jelikož se bere v potaz to, že aktivní služba SQL serveru je zde dvakrát, a proto dvojí licencování.

Aktivní zde má významnou roli, protože u pasivního clusteringu je vždy aktivní pouze jeden nod, to znamená jedna služba SQL, takže úspora na licencích.

Pokud porovnáme mirroring databází a clustering, tak u clusteringu nepotřebujeme další úložiště a ani nakupovat zdvojeně licence. Licence jsou jednoznačně klad této technologie, ovšem použití úložiště je na zváženu. Z hlediska nákladů je to klad, ale z hlediska havárie pole je to špatně.

S touto technologií (ať už mirror db nebo clustering) má smysl začít, jelikož právě zde dochází k nejčastějším výpadkům. Cena řešení na pořízení serverů a FC switche je nejméně nákladná s porovnáním jiných redundancí. U mirroringu db dokonce nepotřebujeme ani FC switch a externí pole, vše lze provést na konkrétních serverech.

Databázové úložiště je vhodné vždy stavět jako externí model, ať už z důvodu flexibility, tak i dalších rozšiřujících možností a využitelnosti. Servery se mění mnohem častěji jak datové úložiště, a proto bych se vyhnul ukládání dat přímo v serverech.

Více na Obr. 2: redundance dvou serverů, cest a switchu.

7.4.2 Redundance switchu a cest k externímu poli je také nesmírně důležitá

Redundance switchu a cest je věc, která se určitě vyplatí. Spoléhat na jedno přenosové vlákno (kabel) je z hlediska koncepce špatně. Redundance cesty přes jeden FC switch lze provést také, dokonce můžeme redundovat jak cesty směrem k serveru (dvě FC karty v serverech), tak i cestu směrem k úložišti (pole s více rozhraními). Tato redundance se pak nastavuje na switchi, kde se definuje zónování a podobně. Na straně serveru se musí nainstalovat podpůrný software, jinak server uvidí duplicitní násobek disků a docházelo by ke kolizím.

Pokud se rozhodneme redundovat i switche, tak se na kabeláži (FC kabely) vše opět násobí. Do pole tedy vedou čtyři kabely, z každého switche dva a servery mají po dvou FC kartách.

V této technologii mluvíme o nákupu dvou switchů (případně placenou registraci portů), nákupu FC konektorů a nákup kabelů a FC karet do serverů. Náklady jsou tak poměrně vysoké v porovnání s náklady na nákup dalšího serveru, ale jsou jednoznačně nižší, než náklady na redundanci úložiště.

Více na Obr. 2: redundance dvou serverů, cest a switchu.

7.4.3 Redundance úložiště (polí)

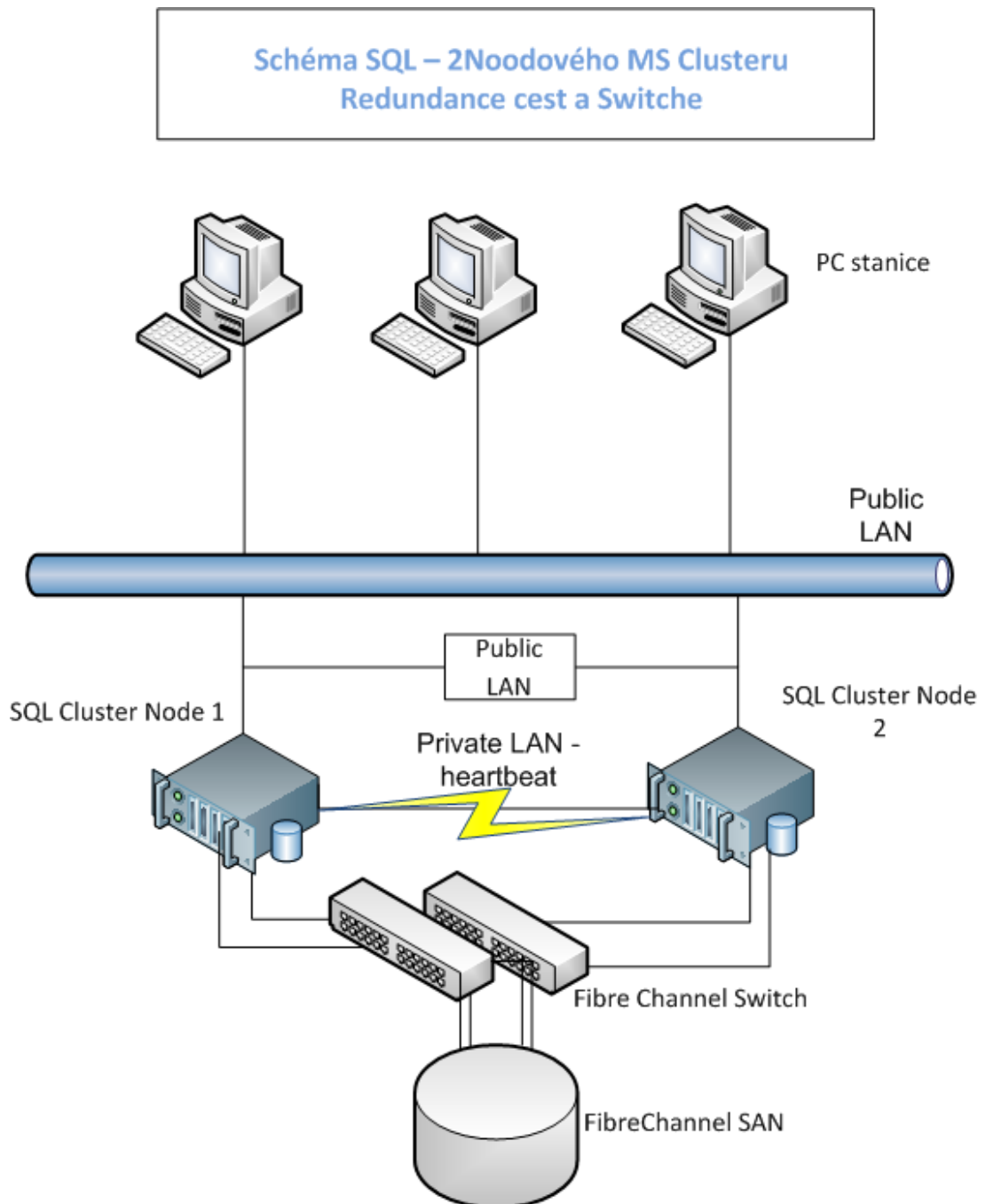
Jedná se o duplicitu na hardwarovém modelu, kdy data z FC switchu zapisujeme na dvě úložiště. Zde je podmínkou použití polí, které tuto funkci podporují. Obě pole by měla být shodná a musí podporovat komunikaci v daném režimu s FC switchem.

Zde jdou použít i technologie, které mezi FC Switch a úložiště vsunou další HW část technologie, která řídí komunikaci mezi poli nezávisle na switchích. FC Switch pak neví nic o redundanci úložiště, protože tuto funkci tvoří další prvek. Tento prvek je vhodné rovněž zapojit v redundanci, a tím opět zdvojujeme cesty k polím.

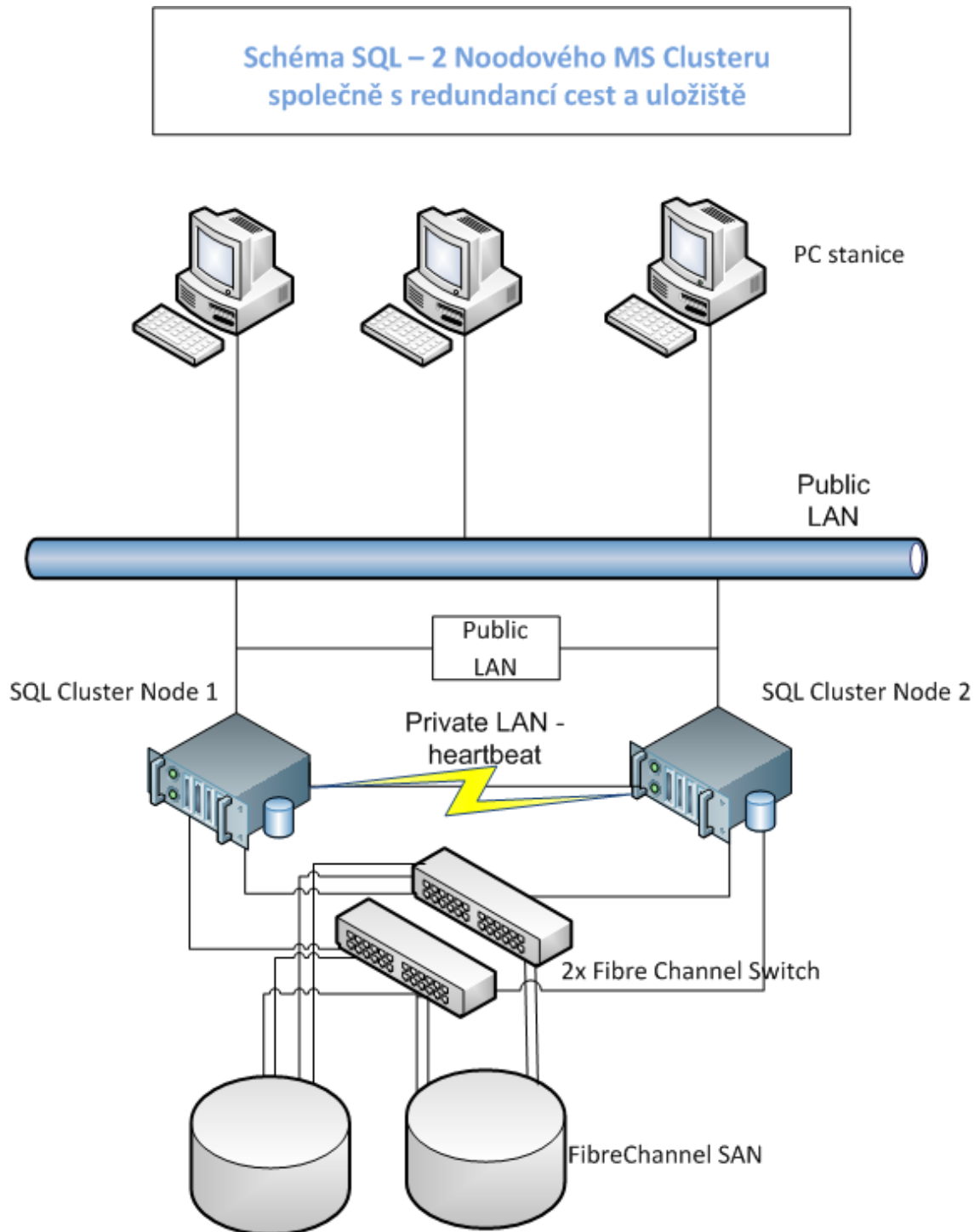
Více informací zde nebudeme uvádět, protože to už není předmětem této práce.

Tato technologie obnáší asi nejvyšší náklady na pořízení. Musíme zde započítat nákup FC karet do serverů, nákup switchu, FC konektorů, kabelů, disků a diskovou šasi s FC řadiči. Proto je třeba si vše pořádně rozmyslet a zvážit, co nám tato technologie přinese.

Více na obrázku Redundance úložiště.



Obr. 2: redundance dvou serverů, cest a switchu



Obr. 3: Redundance úložiště

7.4.4 Redundance LAN

Redundance LAN, neboli trunk linky, je technologie, kdy zvyšujeme rychlost a dostupnost. Trunkování linek musí v první řadě podporovat switche, ke kterým jsou servery připojeny. Doporučuje se zejména zdvojit a ztrojit linky mezi switchi, samotný server je vhodné

připojit pomocí dvou linek a nastavit typ komunikace. Datový přenos v našem případě SQL Serveru je natolik malý, že redundance má význam pouze pro vyšší dostupnost a ne rychlost. Servery samozřejmě doporučuji připojovat na minimálně 1GB switch a i zásuvka k serveru by měla být 1GB. Trunk linky nám umožní mezi switchi komunikovat až 4GB (v budoucích technologiích i výše), tím nemůže dojít k zahlcení switchů a uspání přenosové linky.

Cena tohoto řešení se může lišit, pokud už v síti používáme novější technologie switchu, tak z 99% trunk linky takový switch podporuje a je jen otázkou šikovnosti takovou technologii rozchodit bez nějakých dalších nákladů.

7.5 Běžné chyby

Použití externích polí SCSI a zapojení v redundanci. Toto řešení je velice nešťastné při použití redundance. SCSI pole často disponují více než jedním komunikačním rozhraním a nabízí se zde použití redundance například pomocí clusteringu. Bohužel zde chybí ta část technologie, která kontroluje stavy serverů, a podle těchto stavů vyhodnocuje komunikaci. V našem případě tuto úlohu musí vykonávat sám server, to nese řadu potíží, a spíše než řešení automatického výpadku docílíme nevhodnou technologií snížení dostupnosti služeb. Pokud nám řídicí funkci úložiště provádí server a ne switch, jsme závislí na tom, jaký nod (server) je či byl aktivní při posledních operacích. Musíme zde dodržet například pořadí náběhu serverů. Ten server, který pracoval s úložištěm naposledy (poslední aktivní nod), tak musí předat řízení pasivnímu nodu, jinak nedojde k převzetí služeb. Jelikož nejčastější havárie či údržby se dějí na straně serverů, nedojde-li k předání služeb, celá technologie vyšší dostupnosti nám zhavaruje na softwarové chybě.

Platí zde pravidlo, pro řízení vysoké dostupnosti je nevhodnější HW řízení.

II. PRAKTICKÁ ČÁST

8 POPIŠTE A PROVEĎTE INSTALACI A KONFIGURACI SLUŽBY MICROSOFT CLUSTER SERVICES PRO OS 2003 S. ED.

Před samotnou instalací clusteringu je potřeba provést řadu instalací jiných služeb. Je třeba vhodně zvážit hardwarové požadavky pro instalaci této služby a rovněž vše správně nakonfigurovat. Všechny tyto kroky zde popíšeme a jelikož je clustering nejobsáhlejší a nejvíce komplikovaná část vysoké dostupnosti, budeme se jí zabývat do detailů.

Instalaci Cluster služby provedeme na verzi OS Windows Server 2003 Enterprise Edition:

Před-instalace Microsoft Cluster Service (MSCS)

- OS Microsoft Server 2003 Enterprise Edition nebo Datacenter Edition
- DNS Server
- Domain Controller

8.1 HW požadavky

Hardware serverů, které chceme použít pro clustering, musí splňovat požadavky kompatibility (hardware compatibility list – HCL). Databáze hardwaru najdeme přímo na stránkách Microsoftu, pod názvem HCL SQL Clustering.

Externí úložiště SCSI nebo Fibre channel (komunikace přes optické switche).

Dvě samostatné síťové karty.

Před samotnou instalací doporučuji všechny HW komponenty přehrát nejnovější verzí firmwaru. Zejména u externích polí, disků a switchu je to žádoucí.

8.2 Konfigurace síťových karet

Je třeba zapnout funkci NETBIOS s vyplněným unikátním jménem.

Statickou IP adresu pro obě síťové karty.

Zařazení serveru v doméně, kde primary domain controller běží pod oběma nody.

Před instalací cluster služby musí být servery členy domény a být ve skupině administrators group.

Oddělenou komunikaci pro obě síťové karty. Jedná síťová karta nakonfigurována pro veřejnou (PUBLIC) síť, druhá použita pouze pro komunikaci uvnitř clusteru (PRIVATE síť).

8.3 Požadavky na úložiště

Diskový prostor pro data SQL serveru musí být připojen samostatnými kartami (controllery).

System zapisu dat musí být v systému ukládání dat RAID. (RAID1, 5, 10)

Samostatné oddíly na externím úložišti pro qourom disk a sql server.

Controllery musí mít nadefinovaný jedinečný identifikátor. Důvodem je komunikace cluster manageru vůči HW.

Všechny disky na externím poli musí být ve formátu single disk jako master MBR ve formátu NTFS.

8.4 Instalace Cluster služby pod Windows 2003 [1], [8]

Oproti 2008 Serveru je potřeba dodržet daleko více restriktivních nařízení před instalací cluster služby. Tato nařízení jsou nesmírně důležitá a je potřeba dodržet postup bod od bodu. Proto provedeme každý bod v tabulce. Tabulka je zde jen pro informaci o budoucích krocích instalace, proto s ní budeme pracovat až do konce této kapitoly.

Nastavení	První Server	Druhý Server	Úložiště	Poznámka
Síťová komunikace	ON	ON	OFF	Konfigurace a kontrola funkčnosti síťových karet
Dostupnost externího pole	ON	OFF	ON	Viditelnost externího diskového prostoru
Kontrola dostupnosti externího pole	OFF	ON	ON	Kontrola dostupnosti u druhého nodu, vypneme první Nod a ověříme dostupnost na druhém nodu
Instalace a konfigurace na prvním nodu	ON	OFF	ON	Instalace prvního nodu, ostatní nody jsou vypnuty

Instalace druhého nodu	ON	ON	ON	Po úspěšné instalaci 1. Nodu vypneme tento nod a je třeba opakovat operaci na druhém nodu
Konec instalace	ON	ON	ON	Ověříme správný chod cluster instalace

Tab. 3: Postup instalace clusteru na OS 2003[8]

8.5 Konfigurace síťových karet

Cluster služba vyžaduje dvě samostatné síťové karty. Na veřejnou PUBLIC komunikaci je doporučována 1GB karta, pro interní PRIVATE komunikaci dostačuje 100MB karta.

Síťová karta by opět měla splňovat standard kompatibility tzn. HCL (Hardware kompatibility list).

Je vyžadována statická IP v4 pro komunikaci mezi servery, a také pro veřejnou komunikaci. Doporučuji použít co nejstandardnější řešení pro clustering komunikaci pod PRIVATE síti z důvodů optimálního chodu služby. Je vhodné přepnout rychlost síťové karty na **100MB/Half Duplex** a použít překřížený lisovaný kabel cross-over cable.

Více na Obr. 4: schéma clusteru základ

Privátní komunikace je použita na komunikaci mezi oběma nody, pro informaci o stavech a pro management konzoli.

8.5.1 Konfigurace sítě

Pojmenujeme obě síťové karty a nastavíme tak, aby PUBLIC síť byla začleněna do struktury sítě. Naopak PRIVATE síť je dobré nastavit z jiného rozsahu než je veřejná síť. Síť na sebe sice nevidí, ale je to optimální z důvodu přehlednosti.

8.5.2 Konfigurace privátní sítě

Zapojíme síťovou komunikaci vůči switchům u PUBLIC sítě a křížený kabel mezi oběma nody u PRIVATE komunikace.

Na Privátní síťové kartě musíme vypnout na kartě DNS registraci komunikace v DNS (Register this connections addresses in DNS).

Na kartě WINS vypneme položku NetBIOS over TCP/IP.

U této sítě nevyplňujeme bránu a DNS server.

Více na Obr. 4: schéma clusteru základ

8.5.3 Konfigurace veřejné sítě

Na veřejné síťové kartě je vhodné ponechat zatržené tlačítko na kartě DNS registraci komunikace v DNS (Register this connections addresses in DNS)

Na kartě WINS volíme výchozí položku NetBIOS over TCP/IP (NetBIOS Settings – Dafault)

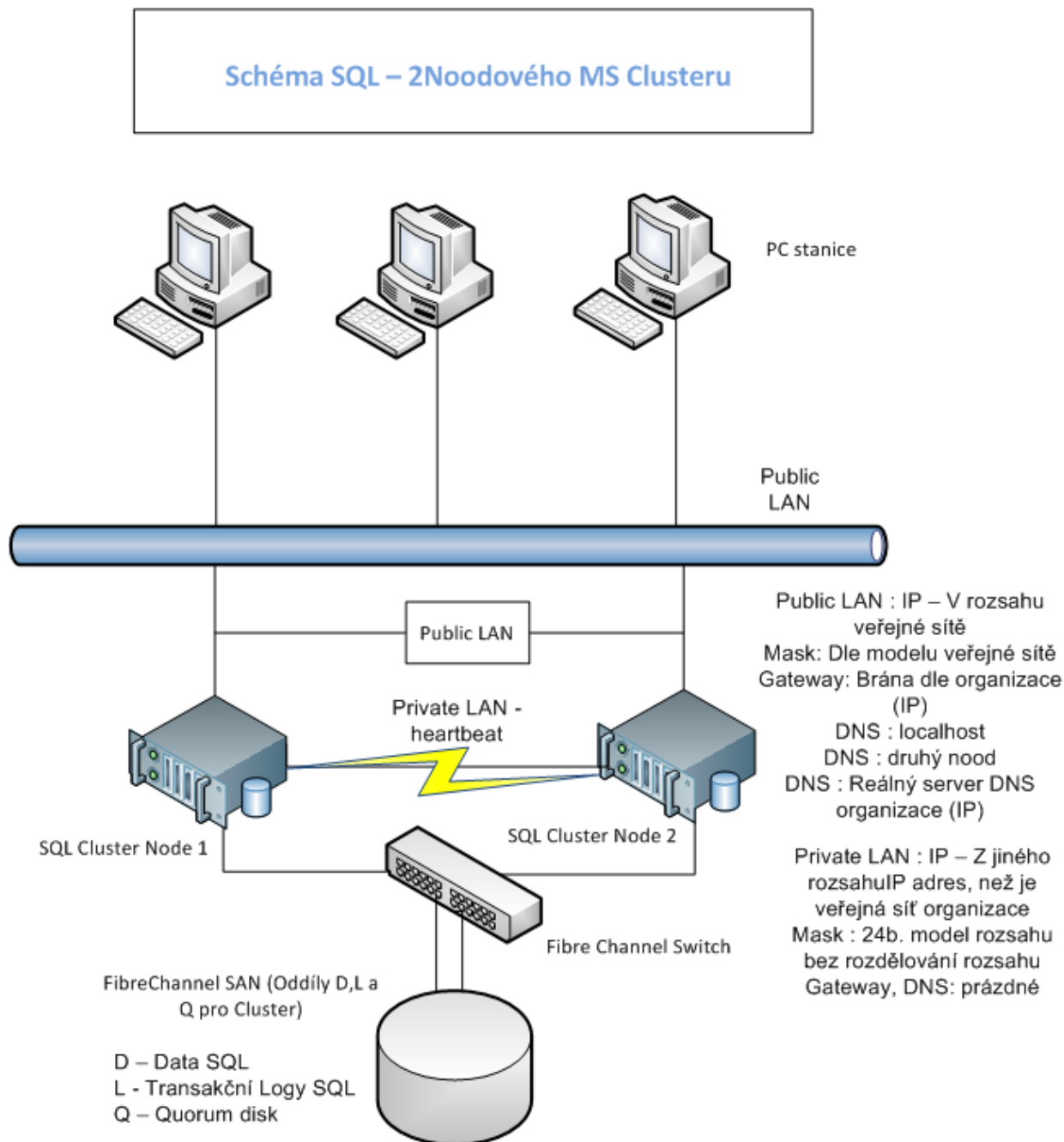
Nastavení DNS z Obr. 4: schéma clusteru základ

8.6 Vytvoření účtu clusteru

Aby nám cluster služba mohla existovat a běžet, je třeba ji navázat na nějaký systémový účet společný pro oba nody. Takový účet proto musíme vytvořit před samotnou instalací cluster služby. K tomu použijeme účet v AD, který bude ve skupině Local Administrator a Domain Users. (Local Administrator mám namysli skupinu Administrators pro každý nod).

Tento účet doporučuji používat jen pro vnitřní chod clusteru. Není třeba se pod takovýmto účtem hlásit do Active Directory, proto považujeme cluster účet jen jako systémový a podle toho zvolme vhodné jméno a heslo. Heslo jednoznačně nastavme takové, aby bylo dostatečně bezpečné, ale co víc, nesmíme nechat zatrženou volbu vypršení hesla. Proč je asi jasné.

Nadefinování účtů pro cluster službu v AD je běžná rutina, proto myslím, že k tomu není potřeba další postup, protože už víme, do jakých skupin musíme účet zařadit.



Obr. 4: Schéma clusteru základ

8.7 Vytvoření a konfigurace řídicího disku pro cluster (Quorum Disk)

Quorum disk je použit pro uchování konfigurační tabulky clusteru, záchranných procedur, výpisu (logů) pro konfigurační nástroj clusteru.

Standalone disk. Proč právě tento typ, je vysvětleno v kapitole, jak definovat hardware pro clustering.

Quorum disk musí mít minimálně 100MB, optimální je 1GB jako Dedikovaný separátní samostatný základní oddíl. (standalone basic disk).

Po instalaci cluster služby a nakonfigurování quorum disku nám vznikne adresář MSCS. V tomto adresáři se nachází soubor quolog.log. Tento soubor je použit jako transakční koordinátor clusteru. Soubor není čitelný, nejedná se o řádkový textový formát.

Jakákoliv změna či zásah do quorum disku je nesmírně citlivá, proto doporučuji naddimenzovat tento oddíl na oněch 1GB. Časem se nám může stát, že tento disk je potřeba například pro řízenou přímou kontrolu zálohování třetích stran (TIVOLI apod.).

Doporučené a nepsané pravidlo je tomuto disku dát písmenko Q.

8.8 Instalace a Konfigurace Cluster služby

Před spuštěním instalace pojmenujeme název serveru tak, aby bylo zřejmé, o jaký server (NOD) se jedná.

A je třeba rovněž dodržet postup uvedený výše (tabulka) – Instalace cluster služby 2003

Minulý bod je nesmírně důležitý, proto ještě malé vysvětlení:

První nod, na kterém zahájíme instalaci clusteru, zapneme (diskové pole nám běží neustále), přihlásíme se účtem v doméně a ověříme, zda je přístupný sdílený disk Quorum.

Druhý nod vypneme !

8.9 Instalace a Konfigurace prvního nodu clusteru

Nainstalujeme službu Cluster Administrator pokud není již nainstalována. Instalace této služby je výchozím balíčkem instalace OS.

Spustíme Cluster Administrator a založíme nový cluster volbou Create new cluster Create – new – cluster.

Přihlásíme se do již existující domény a zadáme jméno clusteru. Pokud nejsme přihlášení na serveru jako administrátor AD, ale pouze lokální administrátor, musíme provést přihlášení i do domény.

Přidáme první server, na kterém právě pracujeme, do nového clusteru.

Průvodce ověří, zda nod (oba nody) má nadefinován přístup na sdílený quorum disk.

Důležitou volbou je konfigurační volba, zda provedeme instalaci Typically, nebo Advanced configuration. Jakou volbu zvolíme, má zásadní vliv na vytvoření Clusterových zdrojů. Typically nám většinu zdrojů vytvoří automaticky, je ale třeba tyto zdroje provést a překontrolovat vazby a nastavení. O tomto se zmíníme vzápětí.

Schéma instalace [8]:

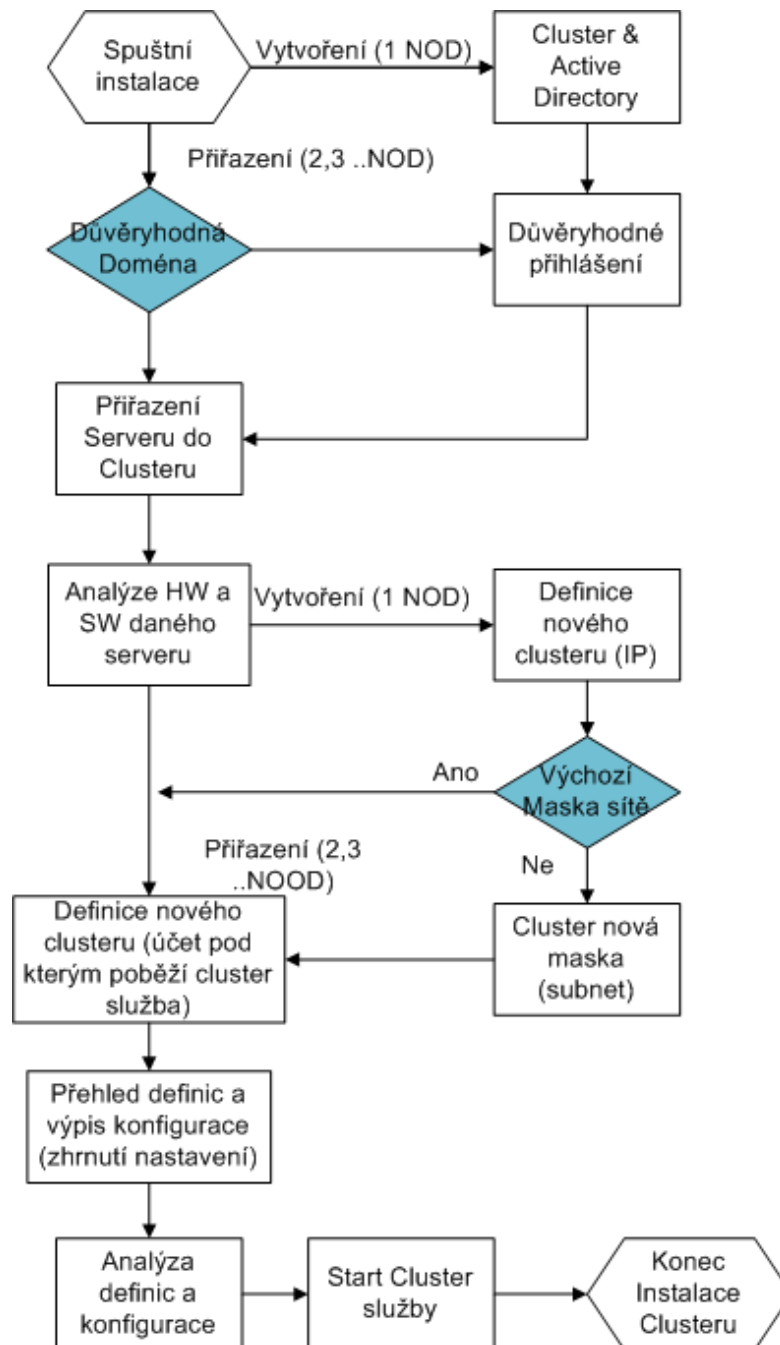


Schéma instalace clusteru pod OS 2003[8]

Pokud zvolíme volbu **Advanced**, cluster nevytváří žádný clusterový zdroj a všechny zdroje si následně vytvoříme ručně. Obě volby mohou být správnou volbou, je to dle letité zkušenosti a testů, jak se cluster bude chovat a co se bude v jakém pořadí migrovat. Automatická volba může být správná, ale taky úplně špatná. Nic zásadního tím ale nepokážeme, takže zvolíme tu, na kterou si troufáme.

Dalším krokem je analýza konfigurace a výsledek testů. Testují se jak HW problémy, tak i SW chyby či nedostatky.

Nyní nás čeká zadání IP adresy clusteru. IP adresa musí být z rozsahu veřejné sítě. Pod touto IP běží služba clusteru a ne samotné služby jako takové, takže není třeba se obávat sdílených služeb pod jednou IP.

Tak jak jsme zadali IP adresu clusteru, tak musíme nadefinovat jméno a heslo účtu, pod kterým poběží služba clusteru. Tento účet jsme již přichystali, byl to ten, který byl členem skupiny **Local Administrator** a **Domain Users**.

Tímto jsme nadefinovali vše potřebné pro první nod a služba clusteru se nám zdárně nainstaluje a nakonfiguruje.

Provedeme ověření funkčnosti, tzn. pustíme přes Administrátorské toolsy aplikaci Cluster Administrator. Vidíme, že je cluster vytvořen a zatím služba běží na prvním serveru.

8.10 Instalace a Konfigurace druhého nodu clusteru

Oproti prvnímu nodu než začneme, je třeba nechat zapnutý první nod clusteru z důvodu přiřazení druhého nodu do již existujícího modelu. Tím, že necháme zapnutý první nod, tak ochráníme společný quorum disk, který řídí clustering. Služba clusteringu běží na prvním nodu, dostupnost této služby a chod této služby ověříme a pokud je vše v pořádku, můžeme pokračovat v instalaci.

Instalace bude obsahovat mnohem méně kroků, než instalace prvního nodu. Můžeme tedy začít. Instalace druhého nodu neproběhne na nodu druhém, jak bychom čekali, ale na nodu prvním.

Spustíme tedy cluster administrator.

Přes položku **File – New** vybereme volbu **NODE** a průvodce zahájí instalaci.

Pokud jsme přihlášení do domény na všech serverech (2), průvodce se nás neptá na doménový účet a heslo oprávněné k přidání dalšího nodu do clusteru.

Podle pojmenovaného serveru vybereme druhý nod a potvrdíme tlačítkem ADD. Pokud máme více nodů, přidáme zde všechny a můžeme pokračovat (v našem případě máme jen 2 nody).

Průvodce spustí analýzu stavu serveru (HW i SW kontrola), zda server splňuje vše pro chod cluster služby.

Dále zadáme jméno účtu pro chod cluster služby. Jméno účtu doporučuji mít stejné na všech serverech pro přehlednost. A opět zde platí členství ve skupinách Local Administrator a Domain Users.

Dalším krokem dokončíme celý proces přidání serveru do clusteringu, spustí se výsledná zpráva s případnými chybami a varováními, potvrdíme opět kontrolu a tím je instalace hotová.

Doporučuji oba nody zrestartovat a to v pořadí: Vypneme nod 2, restartujeme nod první a po naběhnutí AD zapneme nod druhý.

8.11 Dokončení clusteringu

Než nám bude fungovat clustering jak má, je potřeba nakonfigurovat síťovou komunikaci pro clustering. Nadefinujeme, která komunikace má po jaké síťové kartě probíhat a co se má stát, pokud dojde k výpadku karty či kabelu.

Opět pustíme Cluster Administrator nad oběma nody a v sekci Cluster Configuration vybereme část Networks. Tím se dostaneme nad obě síťové karty, respektive na názvy síťových karet, které nyní můžeme nadefinovat jak potřebujeme.

Privátní síťovou kartu nadefinujeme tak, že komunikace bude použita pouze pro Internal cluster communications only.

Naopak veřejnou PUBLIC síťovou kartu nadefinujeme pro veškerou komunikaci – ALL communications.

Dalším důležitým krokem je nastavení priority cluster komunikace. V nastavení clusteru a jejich vlastnostech se dostaneme k záložce nastavení priority pro komunikaci - Network Priority. Zde z popisu je zřejmé, k čemu je toto nastavení dobré. Pro interní komunikaci clusteru proto vybereme privátní síťovou kartu jako nadřazenou před public síťovou kartou.

Všechna tato nastavení týkající se dokončení nastavení clusteringu provádíme pod oběma nody. Pokud máme nodů více, je nastavení podobné v tom smyslu, že určujeme priority komunikací.

8.12 Nastavení Quorum disku

Po nainstalování clusteru se nám může stát, že průvodce instalací nám zvolil špatný disk jako řídicí disk (quorum disk). Jak se to mohlo stát a proč nás v průběhu instalace instalační průvodce na tuhle skutečnost neupozornil? To je správná otázka, ale situace se má tak, že se tahle konfigurace, která je nesmírně důležitá, provedla automaticky podle pravidla: Instalátor vybral nejmenší společný disk, který v systému figuroval a který byl předem naformátován v basic modu s master particí. Ve většině případů je to v pořádku, protože disk Data, Log, případně Temp budou mít větší velikost, ale je lepe takovou věc přezkontrolovat dřív, než budeme instalovat službu SQL jako klastrovanou službu.

Kontrolu a případnou změnu provedeme pod záložkou Quorum ve vlastnostech pod cluster schématem. Root directory by měla být nastavena na `\MSCS\` na diskovém oddíle Q: Tento oddíl jsme vytvořili už dříve, takže pokud je konfigurace v pořádku, můžeme provést kontrolu i fyzicky pod aktivním nodem a přezkontrolovat strukturu adresáře a existenci souboru `clulog.log`, o kterém jsme se již zmínili.

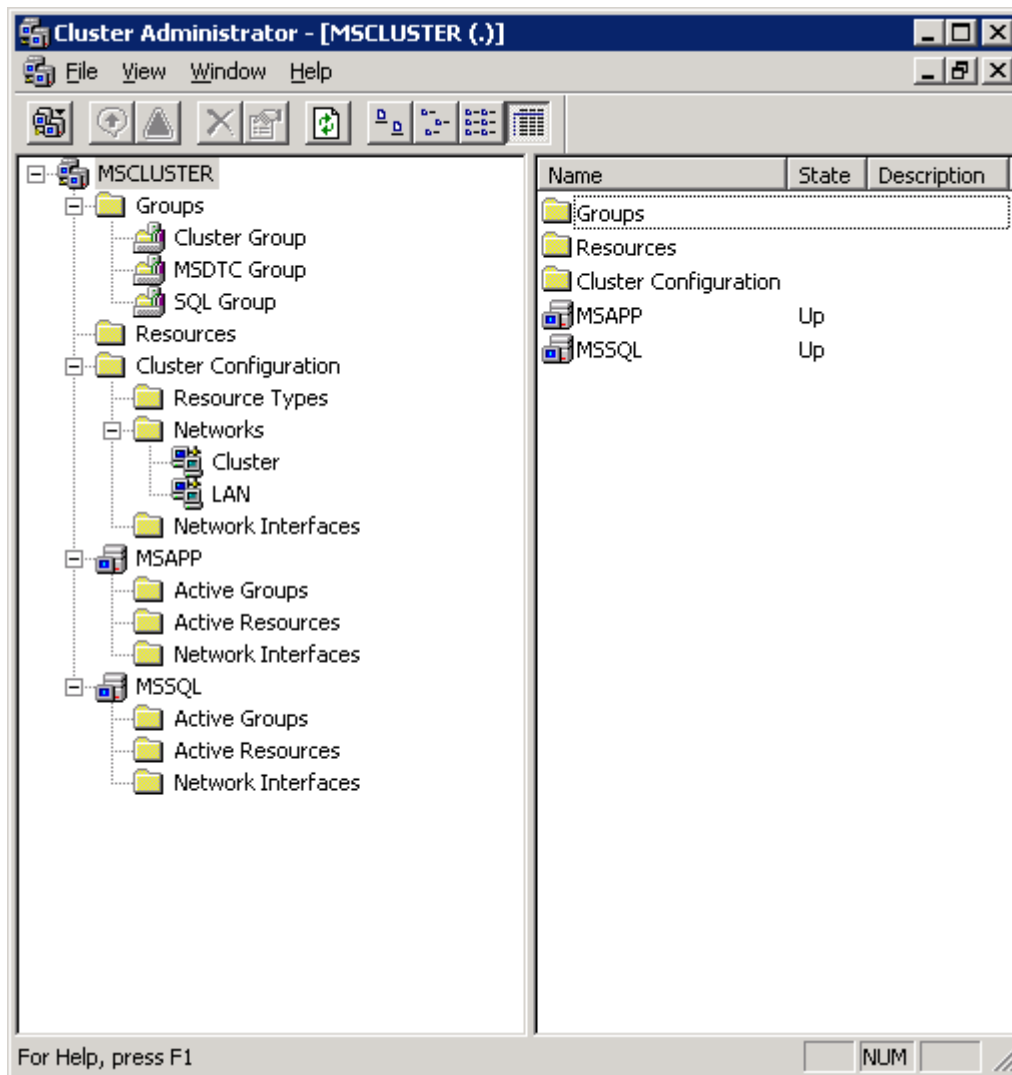
Zde je nastavení cluster služby u konce a zbývá jen test, jak se nám cluster chová ve skutečnosti.

8.13 Test clusteru

Po instalaci a restartu všech nodů clusteru pustíme na jednom z nodu cluster manager a podíváme se, jak vše dopadlo.

První věc, kterou je dobré zkontrolovat, je přítomnost zařazení všech serverů v clusteru. Vybereme náš cluster a na obrazovce vidíme, jaké servery jsou zde zařazené a zda je služba v provozu, či nikoliv (UP).

Pokud je vše jak má, cluster služba ve výpisu služeb je tedy na všech serverech spuštěna, doporučuji ověřit, zda je všude v režimu automatického spuštění po startu OS.



Obr. 5: Instalace Clusteru

8.14 Dalším krokem je kontrola zdrojů clusteru

Po instalaci nás zajímá: cluster, jméno, cluster IP adresa a quorum disk. Všechny tyto 3 zdroje jsou jako cluster zdroj v jedné skupině (groups).

Po instalaci SQL služby zde budeme mít další skupinu s mnoha dalšími cluster zdroji, ale to později.

Doporučuji vyzkoušet migraci ať už s restartem aktivního nodu, tak i ruční migraci cluster skupiny.

Tím je cluster rozchozen a nachystán pro službu SQL.

9 POPIŠTE A PROVEĎTE INSTALACI A KONFIGURACI SLUŽBY MICROSOFT CLUSTER SERVICES PRO OS 2008 S. ED.

Stejně jako pod OS 2003 provedeme instalaci cluster služby pod Operačním systémem: OS Windows Server 2008 Enterprise Edition.

9.1 Příprava instalace FCM

Velká část před-instalačních kroků je shodná s postupem ve verzi 2003, proto není nezbytně nutné postup opakovat, ale uvedu jen konkrétní kapitoly, které je dobré si přečíst.

3.1 *HW požadavky:*

3.2 *Konfigurace síťových karet:*

3.3 *Požadavky na úložiště:*

3.5 *Konfigurace síťových karet:*

3.6 *Vytvoření účtu clusteru:*

3.7 *Vytvoření a konfigurace řídicího disku pro cluster (Quorum Disk):*

Obrázek 2 Nodového clusteru je rovněž stejný: Obr 3.1 v kapitole 3

9.2 Konfigurace síťových karet

Připomenu, že oba servery musí mít dvě samostatné síťové karty a na samotném počátku je vhodné si názvy síťových karet popsat v síťovém nastavení jako Public a Private.

V popisu a schématu budeme brát v potaz 2 nodové řešení. Nastavení síťových karet je shodné pro OS 2003 i 2008 Server.

Komunikace cluster služby probíhá přes protokol TCP/IP. Nastavení se tedy odvíjí od PUBLIC (veřejné) sítě, kde máme jasný rozsah adres, který musí být ve stejném subnetu jako okolní síť. Privátní síť, která je zde mezi dvěma nody, naopak musí být z jiného rozsahu, než je PUBLIC síť. Za povšimnutí stojí i nastavení DNS, které je zde třeba nastavit v rámci clusteru na oba servery pro každý nod. V této fázi prozatím nemáme nainstalován DNS server, ale nastavení můžeme provést již nyní. Aby nám fungoval Internet a aktualizace, musíme v nastavení public síťové karty pro oba nody uvést třetí DNS server, tak jak je definován veřejný DNS server pro organizaci, ve které cluster provozujeme. DNS server, který budeme instalovat a který nám poběží na všech nodech

clusteru, bude použit čistě jen pro cluster službu. Brána a maska PUBLIC sítě je opět totožná jako politika rozsahu IP uvnitř organizace.

Z ostatních protokolů síťové komunikace zde budeme používat ještě klient sítě Microsoft a sdílení složek. Důvodem je Active Directory, kdy oba nody budeme instalovat jako Primary Domain Controllery.

9.3 Instalace Failover cluster služby pod Windows 2008

Jako první krok je instalace samotné služby failover cluster, která není součástí standardní instalace OS. Nainstalujeme nad oběma budoucími nody přes Server manager a nástroj Features službu Failover Clustering. Nyní máme v systému nainstalovanou utilitu failover cluster manager. Spustíme FCM přes Administrative tools.

Před samotnou instalací clusteru doporučuji pustit test nad oběma budoucími nody a to pomocí nové utility, která není součástí W 2003, která je zakomponována ve službě failover cluster = FCM. Spustíme tedy proces v záložce management - Validate and Configuration, který nám nahlásí všechny možné problémy, které je dobré před samotnou instalací odstranit. Chyby hlášeny červeně se odstranit musí, varování nikoliv.

Před instalační kontrola nás může nutit instalovat i co my nechceme, proto takové věci neinstalujeme, pokud nejsou podmínkou k chodu clusteru. Pokud jsme odstranili všechny potíže a nainstalovali potřebné služby a záplaty os, můžeme se pustit do instalace cluster služby.

Spustíme průvodce instalací pomocí nabídky create cluster a provedeme následující kroky:

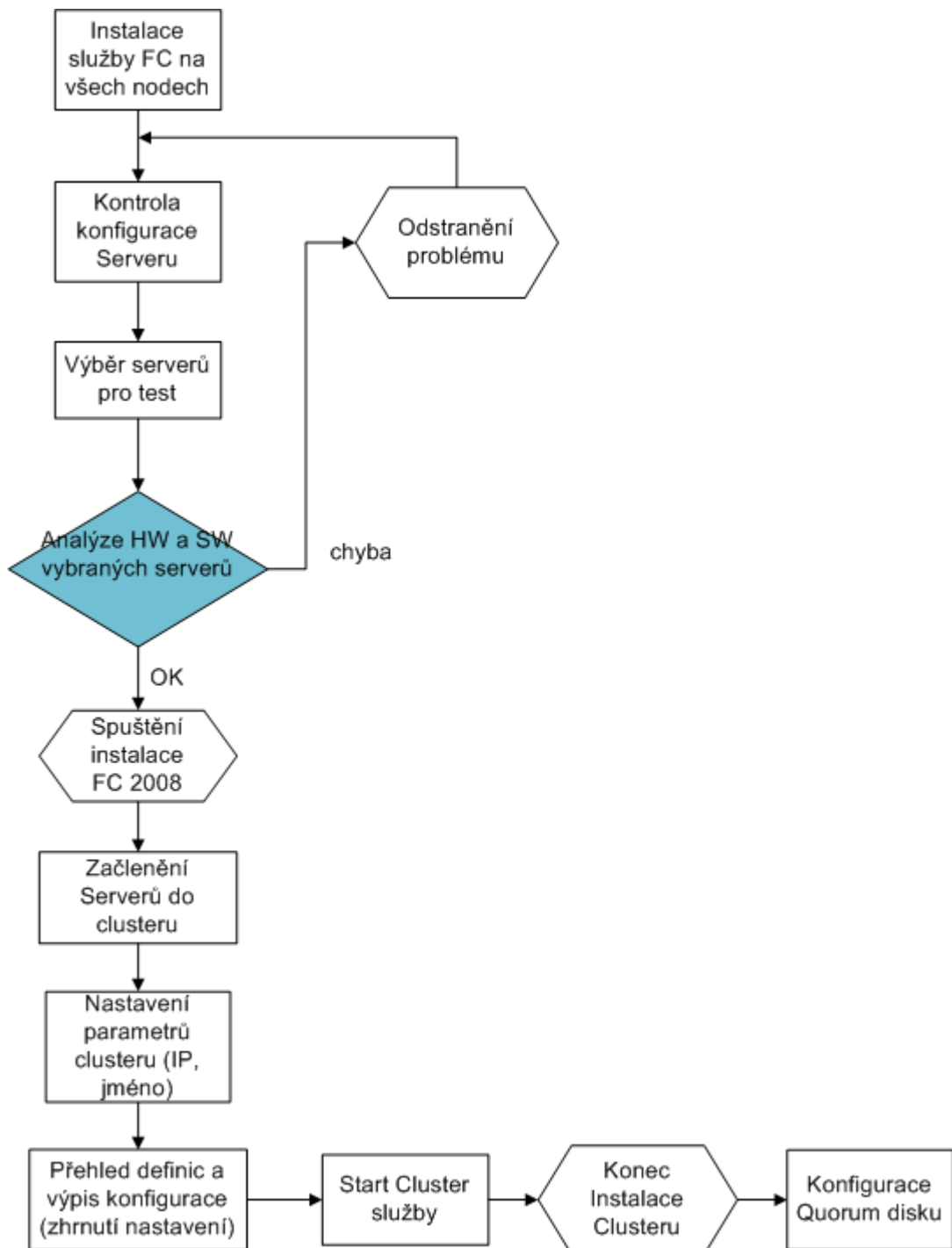
9.3.1 Kroky instalace pomocí průvodce [9],[10]

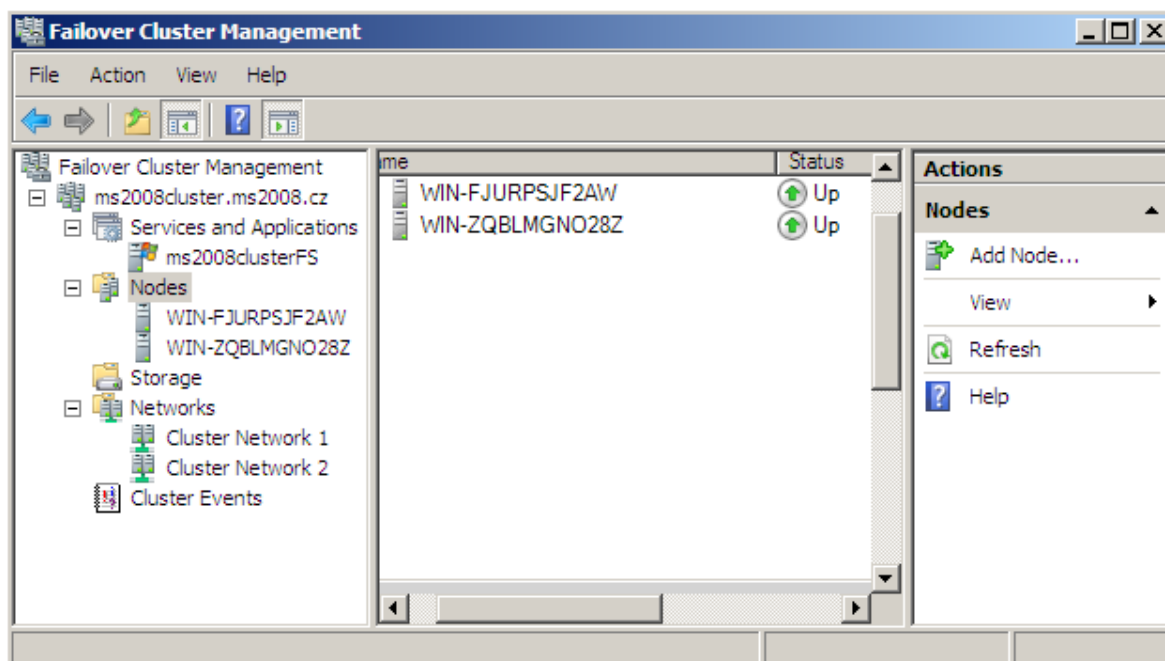
Přidáme všechny nody clusteru (v našem případě 2), přeskočíme testy, které jsme již v předchozím kroku provedli.

Přiřadíme jméno a IP adresu clusteru (veřejnou IP) pro účely admin-konfigurace. Pod touto IP poběží samotný cluster, což přinese řadu výhod, jako například ověřování chodu služby apod.

Dokončíme průvodce instalace Failoveru a doporučuji uložit detail instalace. Nyní máme nainstalován Failover cluster – spustíme Failover Cluster Manager a vypadá to následovně: (obrázek Failover Cluster Manager 4.2).

Schéma instalace:

*Schéma instalace Cluster Services 2008*



Obr. 6: Failover Cluster Manager

9.4 Konfigurace Failover Clusteru [10]

Vytvoření a konfigurace Quorum disku – doporučuji začít právě touto konfigurací, protože je to jádro samotného clusteru. Z před-instalace máme nachystán quorum disk o velikosti 1GB, proto tento disk inicializujeme a použijeme jako Quorum disk.

Disky pro clustering vždy vytváříme jako simple volume.

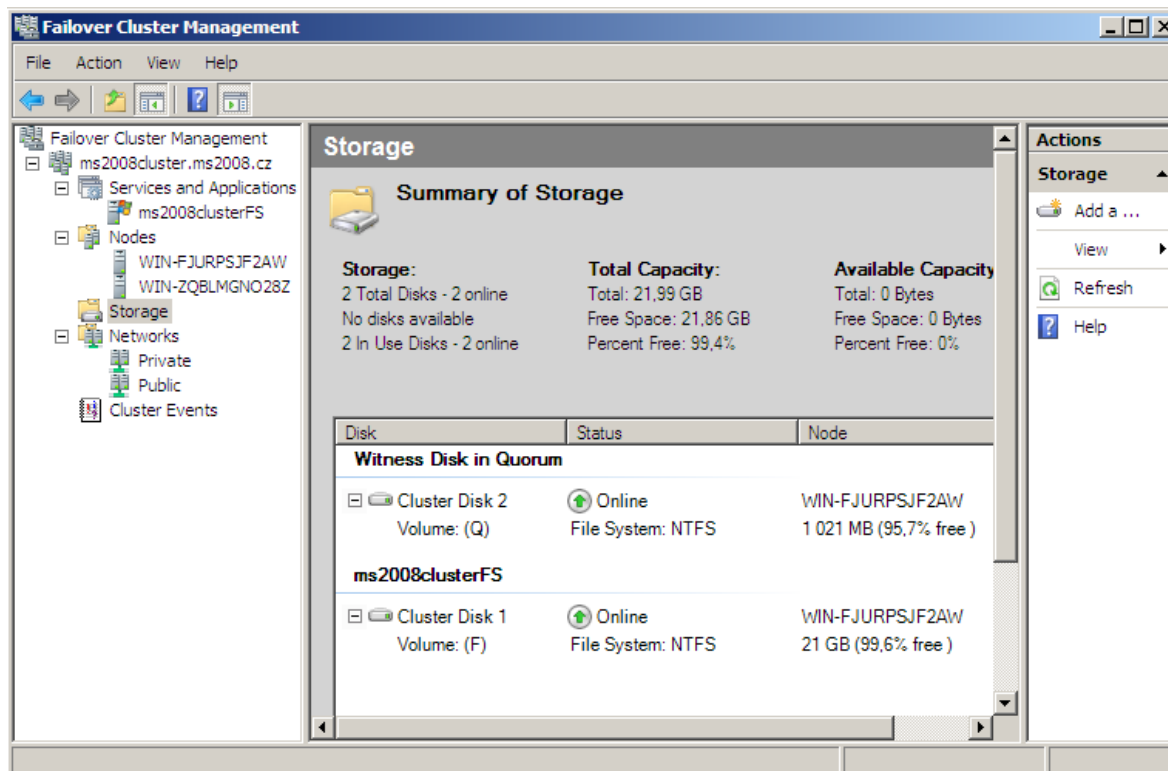
Disk máme tedy vytvořen a pustíme Failover Cluster Manager a provedeme přidání Quorum disku do clusteru. V operacích Actions vybereme položku ADD a disk a přidáme Quorum disk (doporučuji si disky pojmenovávat, pro lepší orientaci při správě) . Přidání disku můžeme provést hromadně, tzn. nachystáme si přes disk manager všechny disky, všechny disky vytvoříme jako simple volume a pojmenujeme.

V Cluster Manageru pak můžeme všechny disky přidat.

Nyní můžeme nadefinovat Quorum disk – v Failover Cluster Manageru (FCM) pod vytvořeným clusterem v nabídce More Actions vybereme položku Configure Cluster Quorum Settings a vybereme model Node and Disk Majority. Vybereme Quorum disk a FCM provede vytvoření a konfiguraci Quorum disku.

Nejdůležitější operace je tedy dokončena a opět si uložíme report pro případné řešení problémů.

9.4.1 Failover Cluster manager po konfiguračních úpravách



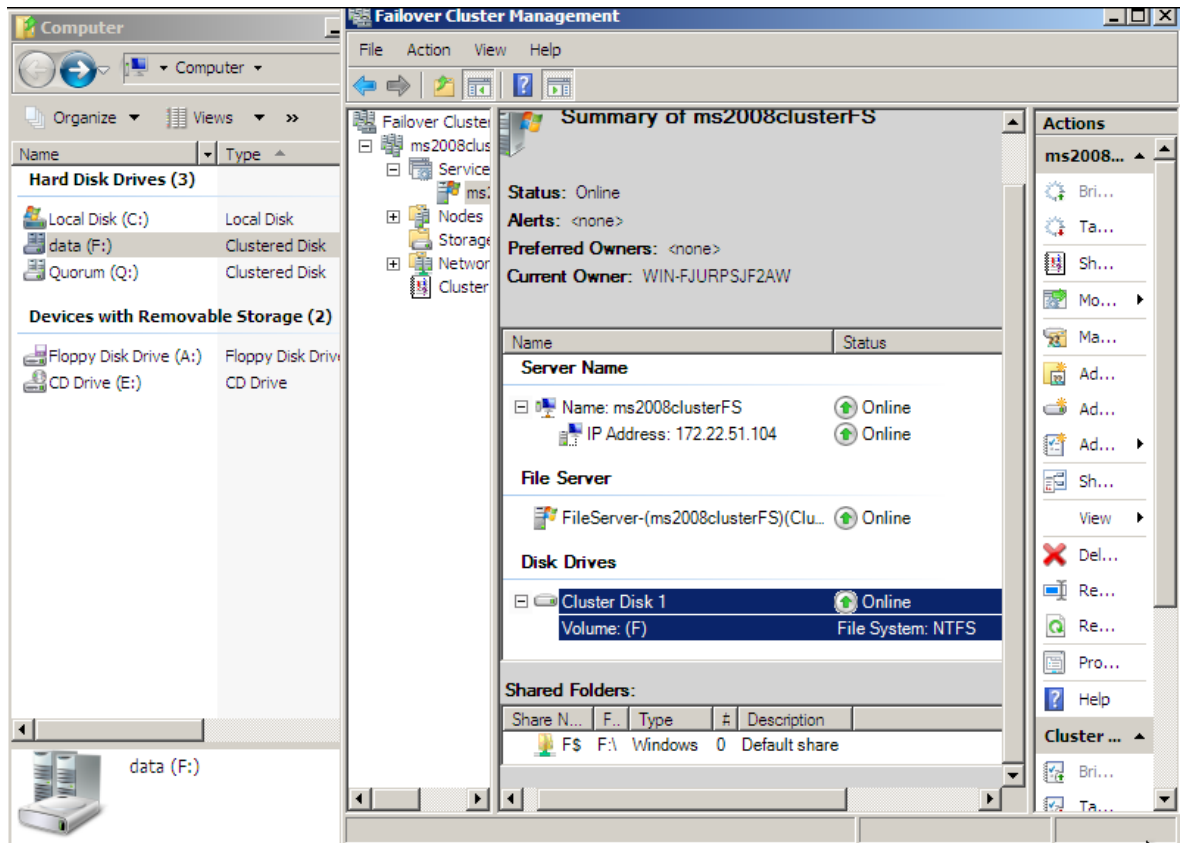
Obr. 7: Failover Cluster manager 2

9.5 Přidání File Serveru do FCM (Sklastrování File Serveru)

V FCM přidáme novou službu do clusteringu opět pomocí průvodce Configure a Service or Application.

Vybereme Službu File Server a přidáme všechny disky nachystané pro SQL službu. Tím jsou disky přidány do schématu clustery a ten zajistí, že v daném okamžiku vidí tyto disky pouze ten server, který je aktivní. V popisu disku pak nalezneme, že se jedná o clustered disk. Správu a konfiguraci takových disků pak provádíme jen přes Failover Cluster Manager. Dostupnost služby, případně její zastavení, provádíme na stejném místě, to znamená v FCM.

Tímto máme cluster nachystán pro instalaci služby SQL. Služba SQL Serveru se nepřidává přes FCM, ale provede se instalace SQL v clustered modu.



Obr. 8: Přidání File Serveru do FCM Failover Cluster manager

Na závěr technická připomínka. Při instalaci SQL služby se ověřuje, jaká verze z hlediska 32bit či 64bit bude použita. Není podporována instalace 64bitového SQL Serveru na 32bitovém operačním systému. Z hlediska koncepce je to samozřejmě špatně a proto je dobré, že systém ověřuje i takovéto kompatibility už před instalací produktu.

10 INSTALACE A POPIS SLUŽEB ZAJIŠTUJÍCÍCH CHOD SLUŽBY CLUSTER SERVICES

10.1 Instalace Active Directory

Instalace jako taková je zde opět společná jako pro OS 2003/2008 Server. Co ale totožné není, je kde AD (Active directory) může či musí běžet. Na Serverech 2003 tzn. Cluster službě pod OS 2003 se musí instalovat AD lokálně a to jako primary domain controllery pro oba nody. Naopak na serverech 2008 Server Edici nesmí AD běžet na daných serverech. Je to celkem zásadní věc, protože zde Microsoft překopal celou logiku clusteru.

OS Server 2003 – Musíme instalovat AD jako Primary Domain Controller pro každý nod. Proto zde musíme nejprve nastavit síťovou komunikaci a doporučuji nastavit i jméno serveru, od kterého se odvíjí jméno domény.

OS Server 2003 – nesmíme zde instalovat AD. Pokud bychom tak učinili, instalace sql služby jako Custerové služby nám nahlásí kolizi při předinstalační kontrole. Platí zde samozřejmě, stejně jako u 2003 Serveru, nastavení síťové komunikace, ale jméno serveru se zatím nikde neprojeví.

Výhody a nevýhody řešení nad OS Server 2003 jsou ty, že zde máme samostatné řešení nezávislé nad jiným serverem zabezpečující AD. Pokud by došlo k výpadku takového serveru cluster služba sice bude stále běžet a poběží zde i cluster služby, ale nebude nám zde fungovat migrační politika. Tzn. při výpadku aktivního nodu nám nedojde k přemigrování cluster služeb (sql služba, disky, dns atd.). Nevýhodou je to, že pokud máme v organizaci již AD, tak stavíme další Domain Controller, což vede k nepřehlednosti síťového prostředí.

Výhody a nevýhody řešení pod OS Server 2008 jsou následující. V tomto OS se počítá se začleněním clusteru do už existující instance AD. Pokud AD budeme instalovat na některý z nodu, nedovolí nám následně sql setup instalaci sql jako cluster službu. Pokud AD v našem prostředí nemáme, musíme ji nainstalovat na jiném stroji nebo použít virtualizaci. Z politiky nasazení vyšší dostupnosti vyplývá, že máme již službu AD v provozu a dostatečně zajištěnou proti výpadku.

Vidíme, že politika AD se výrazně změnila z verze 2003 na 2008. Na jednu stranu je to dobře, na stranu druhou nikoliv. Vychází to i z politiky firmy Microsoft jako takové, kdy nás nutí, abychom používali co nejvíce jejich systémů.

Co je dobré si uvědomit, jsou i úzká místa vyšší dostupnosti. Co se stane, když nám vypadne AD vůči sql clusteru už víme, ale jak to bude vůči stanicím? Pokud se uživatelé neautentizují vůči AD a přistupují k sql serveru přes sql autentizaci, tak nás výpadek AD zajímá jen jako prostředek ke clusteringu. Pokud ale k autentizaci klientů dochází, což je dnes u většiny společností, tak k čemu vlastně dojde? Uživatel nebude v síti a leckdo by mohl říci, že služba sql nebude stejně využívána, protože uživatel se nepřihlásí do sítě, nedostane se k síťovým aplikacím a nemá se jak autentizovat vůči sql. To je všechno pravda, ale ne vždy. Pokud se uživatelé autentizují vůči jinému AD a autentizace sql služby není v rámci AD, který zrovna neběží, tak vše funguje jak má. Může nastat i to, že uživatelé se nebudou přihlašovat do domény a sql aplikace bude komunikovat v rámci sql (mixed) autentizace a zde rovněž sql aplikace fungují. Proto je nesmírně důležité ke všem těmto službám přistupovat globálně a uvědomit si co se stane, když něco nebude fungovat. Nemá smysl řešit redundanci jen jedné služby. Sql (mixed) autentizace se rovněž hodí při přístupu aplikací, které jsou vázány na autentizaci ve vnitřní síti, jako například web aplikace a podobně.

10.1.1 Samotná instalace Active Directory

Instalace pro verzi 2003 i 2008 je skoro totožná. Spustíme tedy příkaz dcpromo, doporučuji si vše pečlivě poznamenat, nejlépe zapsat.

Zadáme FQDN jakožto jméno AD, domain netbios name, autentizaci a to je vše.

Postup instalace AD je natolik jednoduchý, že je potřeba si nejdříve vše důkladně rozmyslet a spíš než řešit v průběhu instalace, co vlastně chceme, tak instalaci nadefinovat dle našich potřeb, ke kterým je budeme využívat.

V téměř každé organizaci již máme AD někde nainstalovánu, proto Instanci SQL 2008 doporučuji začlenit do struktury firmy, a není třeba se zabývat vysvětlováním co je dobře, a co špatně.

V případě SQL 2003 musíme nainstalovat AD novou. Použijeme standardní konfiguraci, která je vyhovující pro chod SQL služby a to vše v režimu dvou primary domain controlleru.

Pro Server 2003 - oba servery instalujeme tak, že instalaci provádíme na jednom a následně na druhém serveru. Servery nevypínáme, pouze první restartujeme a až server naběhne, provádíme instalaci na druhém nodu s tím, že jej přidáme do stávajícího AD,

který nám právě naběhl na prvním serveru. Provedeme restart i druhého serveru a můžeme pokračovat dále.

Pro Server 2008 – AD neinstalujeme, pouze zařadíme oba servery do Active Directory. Pokud AD nemáme, instalujeme AD na jiném stroji, nebo použijeme virtualizaci.

10.2 DNS

Provádí se před instalací Active directory. Služba se přidá před Administrators Tools, kde zaškrtneme službu DNS.

Po úspěšné instalaci zkontrolujeme, zda se nám služba DNS dostala mezi služby v záložce services a zda má definován stav spustit automaticky.

10.3 Ostatní služby a prostředky pro clustering

Jako první služba se vždy instaluje **cluster services**, následně **file services s DNS serverem** a to vše na všech budoucích nodech clusteru.

Instalace **File Services** je navíc v režimu clustered služby. Více o File Services v kapitole: *POPIŠTE A PROVEĎTE INSTALACI A KONFIGURACI SLUŽBY MICROSOFT CLUSTER SERVICES PRO OS 2003/2008 S. ED.*

11 INSTALACE A KONFIGURACE MS SQL V CLUSTER MÓDU – MSSQL 2005

Jelikož máme vše nachystáno, vše co je třeba pro instalaci SQL serveru v režimu clustrované služby, můžeme se pustit do samotné instalace.

Instalaci provádíme na aktivním nodu, tzn. na tom, na kterém jsou přístupny všechny disky, které máme již vytvořeny jako Clusterové zdroje pro SQL.

Před zahájením instalace je třeba oba nody zrestartovat. Před instalací je třeba zkontrolovat stav všech nodů a pokud restart neprovedeme nyní, budeme o něj požádáni v průběhu zahájení instalace.

Před instalací ještě vytvoříme samostatnou skupinu v cluster zdrojích a pojmenujme ji například SQL Server Group.

A poslední věc, která nás při instalaci čeká a co je dobré si nachystat, je na všech serverech vytvořit systémový účet, pod kterým poběží SQL Server. Můžeme zde vytvořit jak doménový účet tak i lokální účet, důležité je, aby byl členem administrators skupiny, respektive domain administrators v případě domain autentikace.

11.1 Instalace Služby

Spustíme z instalačního media setup.exe (nebo využijeme autorun).

Jak už jsme zvyklí, před zahájením instalace provede systém kontrolu a před-instalační procedury. V našem případě to budou dvě komponenty, které musí být nainstalovány. Obě tyto komponenty musíme nainstalovat na všechny nody, které budeme instalovat v clusteru. Zajímavé je to, že obě komponenty se nám nainstalují na všechny nody automaticky tzn. i na pasivní nody, které dále přidáme do clusteringu. Proto není třeba instalovat tyto komponenty ručně, ale provedeme instalaci pouze na aktivním nodu. Jedná se tedy o následující balíčky:

Microsoft SQL native Client

Microsoft SQL Server 2005 Setup Support Files

Pokud jsme se drželi rad psané na začátku této kapitoly, tzn. systém nás nevyzval k restartu všech nodů, dostali jsme se až do části průvodce instalací, který provede test konfigurace serveru – Systém Configuration Check.

Test na systémovou kontrolu provedete diagnostiku celé řady komponentů. Varování opět nejsou povinná, ale failed výsledky musíme nejdříve odstranit, než přejdeme k instalaci SQL Serveru. Varování vždy není dobré za každou cenu odstraňovat. Musíme na cluster pohlížet jako na servery, které poskytují nějaké služby a pokud nás některé služby jako IIS nezajímají, není důvod je instalovat. Nejenže to může znamenat bezpečnostní díru do systému, ale také to zpomaluje servery při náběhu. Platí obecné pravidlo – instalujeme na servery pouze ty služby, které chceme, aby server poskytoval.

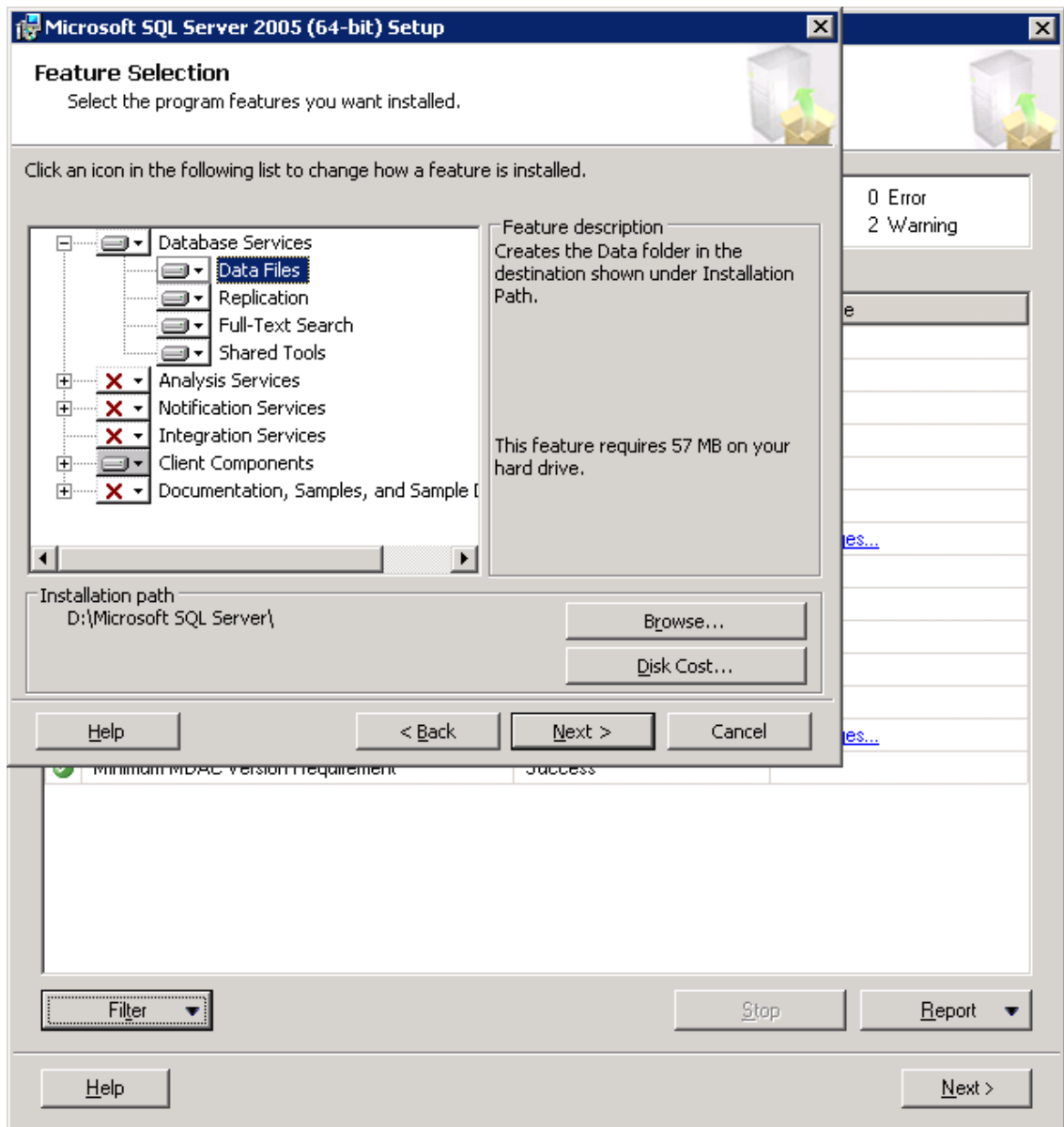
Pokud jsme se drželi všech doporučení obsažených v tomto dokumentu, tak kontrola proběhne v pořádku a můžeme se pustit dále.

Následuje výběr komponent, které chceme na servery instalovat. Záměrně píší na servery, protože vše, co teď provádíme, je konfigurace průvodce instalací a samotná instalace se provede až na konci instalační procedury, instalace se provede na všechny nody, jsou součástí clusteringu.

Z komponent opět doporučuji instalovat jen to, co využijeme. V našem případě to je SQL Server Database Services v režimu clusteru. Tzn. nezapomeňme zatrhnout Create s SQL Server failover cluster. Vše ostatní nás v našem případě nezajímá, protože vývojářské komponenty Analysis Services nepotřebujeme, Reporting, Notification a Integration Services nás také nezajímají a dokonce ani Workstation components. Důvod je prostý. Server má vykonávat jen danou činnost a není třeba instalovat věci jako komponenty pro management, nebo dokonce knihovny s literaturou. Takové věci patří na stanici administrátora a ne na server. *Obrázek Instalace SQL 2005 Cluster Mod 6-1.*

Co je na této obrazovce nesmírně důležité je tlačítko Advanced. Díky tomuto tlačítku se dostaneme do podrobnější instalace, kde ovlivníme, kde budou uloženy transakční logy (na kterém disku), kde budou data SQL a také jakou collation použijeme pro model databáze a model sql.

Na další obrazovce zvolíme instanci SQL Serveru. Co to je instance jen v krátkosti. Default Instance řekne SQL službě, že poběží na defaultním portu 1433. Více instancí má význam pro tříštění SQL databází, ovšem užitek nulový, protože jakékoliv následné propoje mezi db se musí psát plnou cestou s delší autentizací. Použití více instancí má význam, pokud chceme fyzicky oddělit dva či více SQL serverů pod jedním strojem. Rozhodně tím ale neušetříme a neuřídíme systémové zdroje HW. (kromě RAM paměti). Celá správa je pak mnohem složitější a použitelnost je jen ne výjimečných případech.



Obr. 9: Instalace SQL 2005 Cluster Mod

V cluster režimu to je zajímavější, protože z pasive clusteringu lze udělat aktive clustering. Jak to? No je to prosté, každá SQL služba poběží na jednom ze serveru (nodu) a pokud obě služby SQL budou v cluster režimu, tak nám budou migrovat v případě výpadku jednoho z nodu. Výhody a nevýhody tohoto řešení necháme do jiné kapitoly, nyní provádíme instalaci SQL služby v cluster modu a mohli bychom se hravě dostat do nepřehledné části práce, která pak nebude úplně dobře srozumitelná.

Takže instalujeme první SQL v clusteringu, zvolíme Default Instance.

Dále zvolíme Virtual Server Name. Toto jméno definuje, pod jakým názvem se budou stanice nebo aplikační služby připojovat na SQL Server. Je to jméno v našem případě defaultní instance.

Pokračujeme na další obrazovce. Zde definujeme IP adresu, pod kterou bude SQL Cluster služba vystupovat. Tahle IP adresa bude vytvořena jako cluster zdroj, takže pokud dojde k výpadku serveru, tak IP adresa migruje se službou SQL Serveru tak, aby byl zachován připojovací algoritmus (Conner string) aplikací. Jak služba klastruje budeme řešit v další kapitole, zatím nás zajímá jen to, že IP adresa je jedinečné číslo SQL služby a v 99% případů ji navážeme na public síť (PUBLIC síťovou kartu).

Dále zde máme výběr klastrové skupiny, pod kterou budou všechny SQL zdroje vytvořeny. Doporučuji použít samostatnou skupinu pro SQL zdroje už kvůli přehlednosti a také rychlosti migrace. Přiřadit SQL cluster zdroje jdou do výchozí cluster groupy, kde máme prozatím tři zdroje (ip adresa clusteru, jméno clusteru a quorum disk), ale takového kroku bych se vyvaroval.

Dále zde máme nastavení umístění datových souborů SQL serveru. Vybereme tedy cestu D:\Microsoft SQL Server\. Samotnou strukturu adresáře doplní ještě jméno instance a název služby, to ale ničemu nevádí. Transakční logy, které chceme umístit na jiné písmenko, zde bohužel nenadefinujeme. Proto cesta bude totožná jako je nastavení pro data. Je to jedna z nevýhod průvodce SQL Serveru 2005. Pod SQL Serverem 2008 tato nastavení už lze provést.

A máme zde opět další obrazovku s nastavením a to přiřazení nodu do clusteru. Podmínkou je aspoň jeden nod, uvedený jako Required node (je to ten, na kterém instalaci provádíme a zároveň aktivní nod). Jelikož máme ostatní nody spuštěny, začleňme všechny tyto servery do clusteru. Docílíme tím automatickou instalaci SQL Serveru (automatické nahrání a konfigurace souboru, definice služeb, konfiguraci registru a úpravy či přidání systémových knihoven). Podmínkou je Administrátorský přístup Active Directory, do které jsou dané servery zařazeny pomocí autentizace, která nás čeká dále.

Nejdříve vyplníme pod jakým účtem proběhne vzdálená instalace SQL serveru na ostatních nodech. Je to ta automatická instalace služby o které jsme již hovořili v této kapitole a která nám velice usnadní instalaci, protože ji provádíme pouze jednou. Proto zde vyplňujeme doménový účet, použijeme autentizaci serveru pod společnou doménou,

kteřou jsme nainstalovali a nadefinovali. Účet musí být opět členem domain administrators a local administrators. Můžeme pokračovat dále.

Na další obrazovce definujeme, pod jakým účtem poběží služba SQL Serveru. Můžeme použít jak lokální účet, tak i doménový účet, opět v režimu administrators, respektive Domain Administrators. Doporučení je použít doménový účet, ale troufnu si říci, že je zde mnohem lepe použít lokální účet a to z jednoho prostého důvodu. Účet pod kterým běží tato služba je svázána s dostupností AD, pokud tato služba vypadne, služba zhavaruje. Tím, že zvolíme lokální účet se zbavíme této závislosti, což je ještě žádoucí. Ostatní služby jako SQL Agent, který jsme použili pro Maitanance plány, svážeme opět s SQL službou a zatrhneme tuto volbu z důvodu, že služba agenta bude rovněž vytvořena jako clustered zdroj.

Můžeme pokračovat dále a definujeme pod jakými účty poběží podpůrné služby SQL serveru jako Agent a Full-text Search. Pokud jsme na minulé obrazovce zvolili lokální účet, tak se nás toto nastavení netýká. Doporučuji nadefinovat spouštění všech služeb pod jedním účtem, ať už jsme zvolili lokální účet pro autentizaci, nebo doménový účet.

Tím jsme se úspěšně dostali až na konec průvodce a instalace SQL služby započne. Doporučuji žádný ze serverů nevypínat nebo restartovat, i když se může zdát, že se na nich nic neděje. Do okamžiku, než nám průvodce nahlásí konec instalace na servery tzn. na všechny nody, které jsme začlenili, do instalace raději nesahejme. Procedury jako automatické instalace apod. jsme doufejme provedli těsně před instalací SQL Služby, takže by nás ani tato věc neměla potrápit, pokud se spustí. Instalace bude trvat poměrně dlouho, záleží to server od serveru, takže můžeme pozorovat, jak se nám vše krásně instaluje a služby postupně zelenají.

11.2 Po-instalační kroky Instalace

Provedeme kontrolu cluster služby SQL v cluster manageru. Spustíme cluster manager a podíváme se, jaké zdroje nám přibyly do skupiny SQL Server Group. Skupina musí obsahovat povinně následující položky. Pokud nějaká chybí, něco jsme zanedbali, nebo jsme nedodrželi postup, který zde byl popsán. Není se čeho obávat. Nejenže jsem se nesetkal s tím, že by nějaké zdroje byly špatně nadefinovány nebo chyběly, ale zdroje můžeme dodefinovat jak potřebujeme, případně je předělat. Týká se to například cluster disků, které nám mohou časem přibýt, třeba z důvodu rozšiřování kapacity.

SQL Server Group obsahuje následující clusterové zdroje

Disk D (datový disk pro databáze sql serveru, neboli sdílený disk clusteru)

SQL network Name (jméno sql serveru pro připojování aplikací a agend)

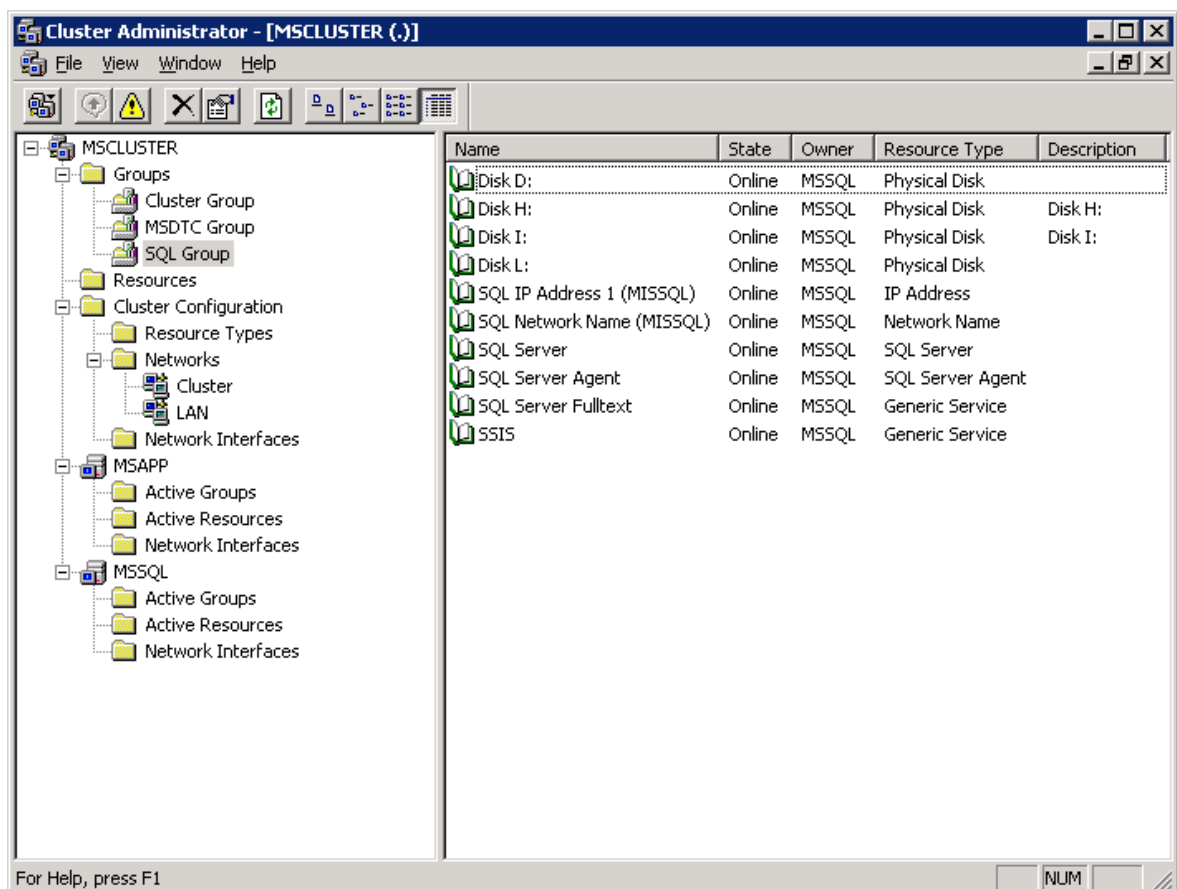
SQL IP Address (IP adresa pod kterou vystupuje SQL služba. Důležité je si uvědomit, že SQL server nevystupuje pod IP adresou serveru, ale pod touto IP adresou a to vždy, nezávisle na tom, kde tato služba ve skutečnosti běží)

SQL Server

SQL Server Agent

SQL Server FullText

Jelikož nám zde chybí disk s Logy, přidáme tento disk do clusteru a závislost tohoto disku nastavíme stejně jako disk D. Jak zařazení dalšího disku provést si řekneme v kapitole dokonfigurace a ladění SQL Serveru.



Obr. 10: Cluster Administrator Console

11.3 Ověření chodu clusteru – Test funkčnosti a reálná migrace služeb

Jako jednoduchý první test provedeme restart aktivního nodu. Pustíme tedy cluster manager na pasivním druhém nodu a pozorujeme, co se děje, když první nod provádí ukončení služby.

Jako druhý test doporučuji ruční migraci skupin. Je to mnohem rychlejší než jako restart serveru. Nejdříve vždy provádíme migraci nesystémových zdrojů tzn. nejdříve přemigrujeme SQL Server Group a až je vše přemigrováno provedeme migraci Cluster Groupy. Až budeme mít více skupin, je možné aplikační skupiny migrovat zaráz.

Tím máme otestován ideální stav clusteru, zbývá otestovat migraci, pokud dojde k nějaké události.

Test sítě: Z aktivního nodu vytáhneme public síťový kabel a podíváme se, co se děje. Cluster se zachová tak, že prověří komunikaci mezi clustery přes PRIVATE síť, provede test PUBLIC síťové karty na pasivním nodu a pokud jsou obě podmínky splněny, započne migrace clusteru na pasivní druhý nod.

Test dostupnosti disku: ač následující věta bude znít šíleně, musíme provést test i výpadku dostupnosti disku. Ryze teoreticky vytáhneme optický kabel či jiný, kterým máme připojen FS nebo SCSI pole k serveru a sledujeme, co se děje. Prakticky si pomůžeme nástroji managementu, protože zásahy do FC kabelů nebo i SCSI kabelů jsou riskantní a může dojít k jejich poškození a nebo opotřebení (chybovost). Opět by nám veškeré služby tzn. zdroje měly přemigrovat na druhý nod. Pokud máme řešení dostupnosti diskového pole pomocí Fibre Channel switche, je mnohem rozumnější test provést přes management switche a jednoduše vypnout daný konektor přímo na switchi. Switch nedoporučuji vypínat stejně jako pole, protože dojde k migraci i v případě, že je pole nedostupné pro druhý nod. Je to chyba clusteringu, která s touto situací nepočítá. Možná se jedná o chybu nedokonalé komunikace dostupnosti pole s podporou softwaru Microsoft tzn. podpora třetí strany, ale to je věc, kterou neovlivníme. Proto provádíme tyto testy a zjišťujeme, co se bude dít, až se stane nečekaná událost.

Ještě není špatné zmínit, že pokud z nějakého důvodu nedojde k náběhu služby na druhém nodu, tak defaultně dojde ke třem pokusům migrace a následně se uvede služba do stavu faild. V takovém případě pokud dojde k opravě, je potřeba ověřit, zda je tedy vše v pořádku a zdroj, nebo celou skupinu zdrojů nechat naběhnout do stavu online opět přes

cluster manager. Celá tato operace se uvede do daného stavu přes menu Take online/Take offline.

11.4 Instalace Service Packu do SQL 2005 v Cluster modu

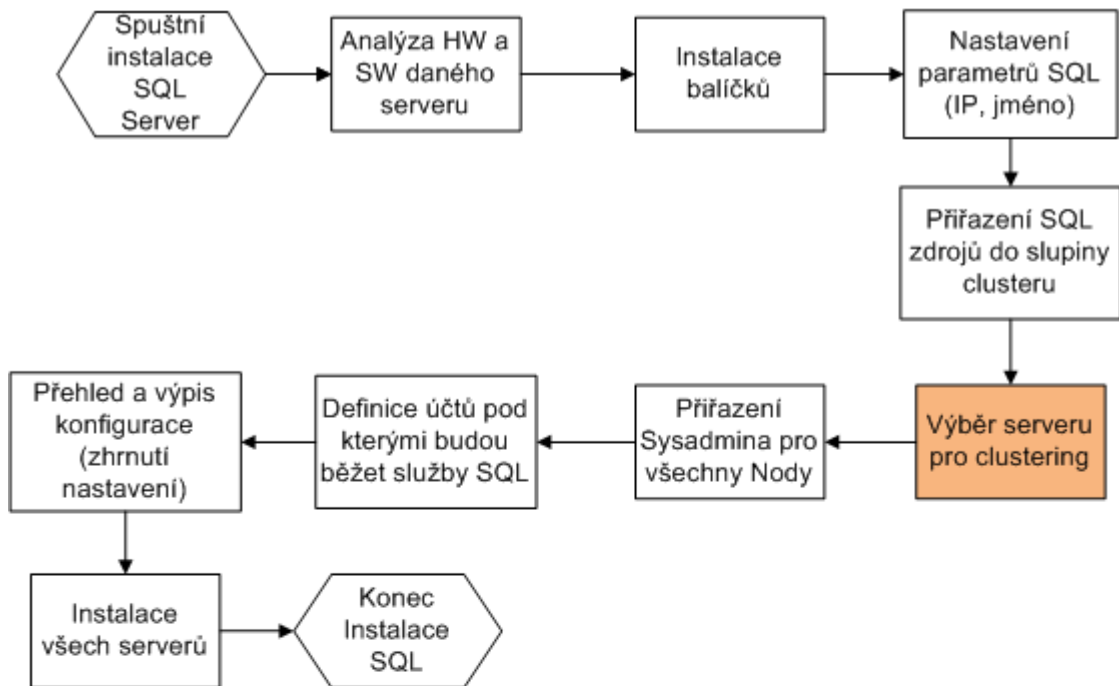
Pokud se rozhodneme, že chceme instalovat SP SQL Serveru 2005 v dané verzi 64bit. postupujeme následovně. Jako první věc zkontrolujeme, zda jsou oba nody plně funkční v cluster administrátoru. Zde vidíme, zda služba clusteringu funguje bez problémů. Pokud je vše v pořádku, instalujeme SP pouze na aktivním nodu. Provedeme instalaci tím, že je SQL klastrová služba, provede se instalace automaticky na druhém nodu. Po instalaci bude třeba oba nody zrestartovat. Nejdříve vypneme server, kde služba sql neběží (pasivní nod) a zrestartujeme aktivní nod. Po náběhu prvního nodu pustíme server druhý a instalace SP je dokončena. Zda úspěšně, ověříme opět v cluster administrátoru spuštěním Management Studia a provedeme SQL dotaz pod master databází.

```
SELECT @@VERSION, GO
```

11.5 Aktive node clustering

Aktivní clustering je velice prostá a jednoduchá záležitost, týkající se instalace. O to horší jsou pak licence a jak udržovat oba SQL servery po stránce konfigurace a účtů. Nejedná se totiž o active clustering, jak si většina administrátorů může představit. Jde o dvě samostatné instance SQL serveru pod jiným portem. Defaultní instance má port 1433, druhá instance musí mít port jiný, stejně jako jméno SQL serveru. Instalace clusteru je jednoduchá v tom, že SQL server vytvoříme jako další zdroj v clusteru, který navážeme na další potřebné zdroje jako jsou disky a podobně. Zásadní nevýhodou jsou tedy jak řešit licence, protože zde máme rázem dva SQL servery a už se neřeší, že jsou nainstalovány, nebo dokonce běží na jednom OS a stroji. Druhá nevýhoda je zmiňovaná údržba, zálohování, případně správa účtu, pokud používáme SQL autentizaci v mixed modu. Active clustering díky těmto zásadním vlastnostem není ideální řešení a mnohem lepší je použití například zrcadlení databází.

11.6 Schéma Instalace SQL 2005:



Přřidání Nodu do clusteru:

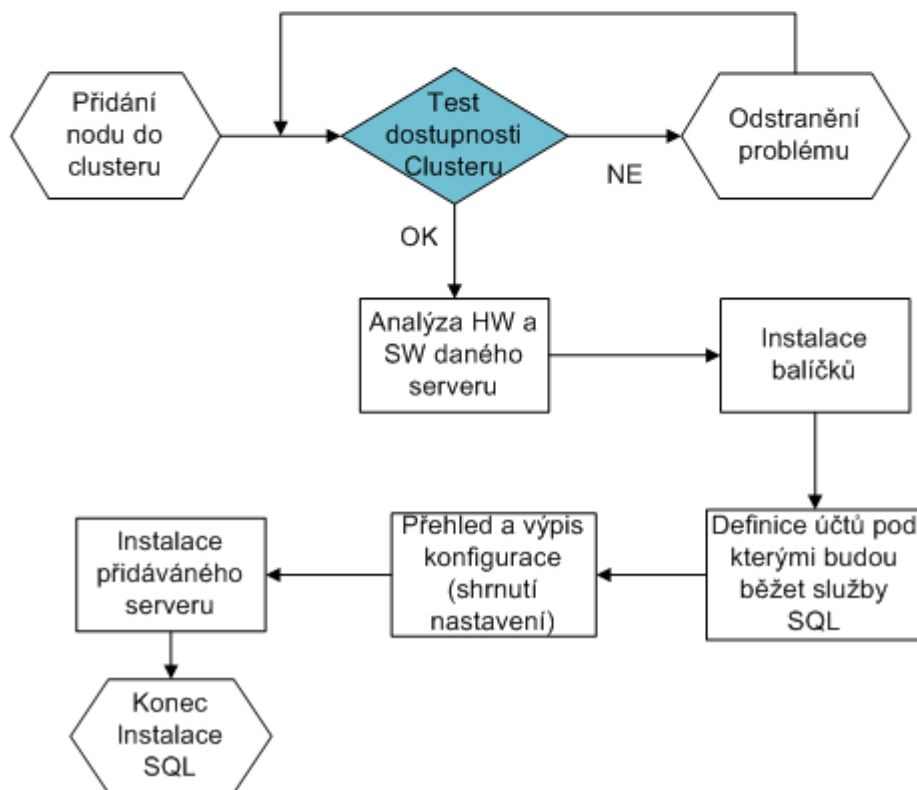


Schéma instalace SQL 2005

12 INSTALACE A KONFIGURACE MS SQL V CLUSTER MÓDU – MSSQL 2008

Jak už jsme si zvykli, před zahájením instalace opět nainstalujeme a nakonfigurujeme vše, co by nás mohlo při instalaci překvapit. Tím se vyhneme možným problémům a instalace bude čistá bez zbytečných nestandardních zásahů.

12.1 Schéma Instalace

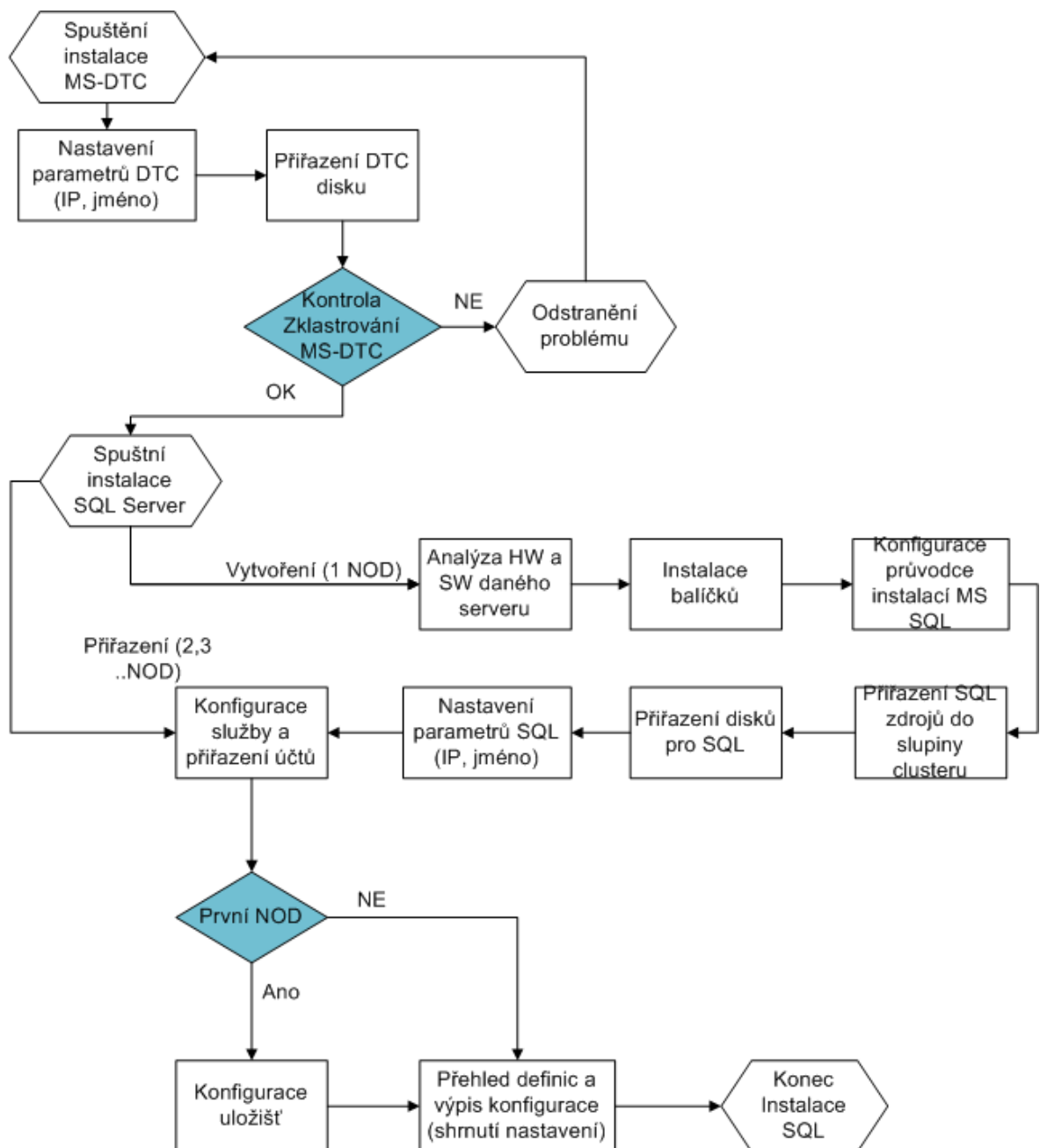


Schéma instalace SQL 2008

12.2 Před instalace

Provedeme instalaci **.NET Framework 3.5 společně se SP1**. Pokud je novější verze, zřejmě bude vhodnější nainstalovat právě onu poslední.

Provedeme instalaci aplikace **Windows Installer 4.5 se SP1**. Verze aplikace a verze záplat je chápána jako minimální potřebná konfigurace. Pokud se nám časem objeví novější verze, bude asi vhodnější použít ty novější.

12.3 Instalace DTC (MSDTC)

Jako první věc kterou nainstalujeme je služba MSDTC. Služba MSDTC je transakční manažer, který dovolí klientským aplikacím seskupovat různé datové zdroje do jedné transakce a následně je tato transakce distribuovaná všem serverům, kterých se daná transakce týká. Tím se nám několikanásobně zrychlí čas potřebný ke zpracování a díky transakčnímu manageru je celá úloha mnohem lépe řízená.

MSDTC službu je potřeba nainstalovat už před instalací MS SQL Serveru z důvodu použití distribuovaných transakcí a také z důvodu instalace služby MS SQL v clusteru, která právě distribuované transakce používá. SQL Server v cluster modu pak používá službu MSDTC pro distribuované dotazy, commit dotazy, nebo pro replikaci, o které si řekneme také, ale v jiné kapitole. Replikace databází s použitím MSDTC není podmínkou pro instalaci clusteringu.

MSDTC manager je oproti verzi SQL Serveru 2008 mnohem kompaktnější, mnohem snadněji se konfiguruje a je méně náchylný na špatné nadefinování. Konfigurací máme na mysli proces, při kterém aplikační část musí kontaktovat manager a předat mu parametry ke zpracování.

12.3.1 Instalace MSDTC

Spustíme Failover Cluster Manager.

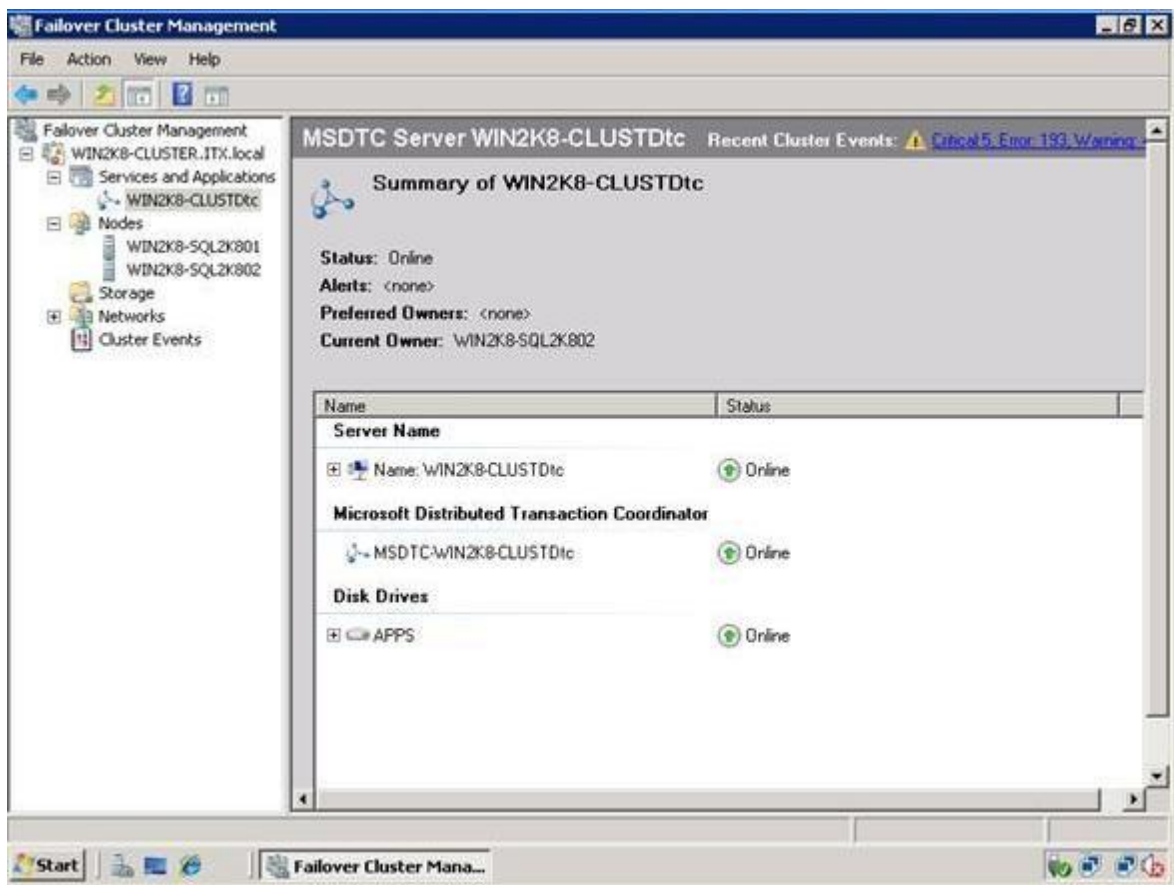
Na položce, kde je náš cluster (cluster jméno), zvolíme volbu Configure a Service as Application. Tím docílíme spuštění průvodce vyšší dostupnosti a z menu vybereme naši službu DTC (MSDTC). Po potvrzení instalace nás čeká následující nastavení.

Nadefinujeme jméno služby a IP adresu služby. Pod tímto jménem a pod touto IP adresou se pak služba hlásí navenek a právě přes toto jméno či IP adresu lze s touto službou pracovat. IP adresu opět zvolíme z rozsahu veřejné PUBLIC sítě.

Dále definujeme disk, který jsme si přichystali a pojmenovali MSDTC. Disk jsme rovněž zařadili mezi cluster zdroje, takže se nám zde nabídne. Pokud bychom tuto věc opomenuli, nemohli bychom jej nyní použít. MSDTC disk nakonfigurujeme jako quorum disk.

Tím jsme u konce, ještě dostaneme výpis konfigurace a pokud se všim souhlasíme, zahájí se instalace.

Služba je vytvořena jako klastrový zdroj a můžeme tento zdroj vidět v FCM v menu services and Applications.



Obr. 11: Instalace MSDTC SQL 2008

12.4 Instalace SQL Serveru 2008 jako klastrový virtuální zdroj

Než začneme instalovat tak zásadní službu jakou je SQL Server, otočíme (zrestartujeme) všechny nody clusteru, zkontrolujeme, zda cluster funguje a instalaci provedeme na aktivním nodu.

Spustíme **setup.exe** z instalačního média a z menu vybereme instalaci SQL Serveru v režimu failoveru (New SQL Server failover cluster installation).

Průvodce prověří stav systému a kontrolu nainstalovaných potřebných aplikací a balíčků. V našem případě je zapotřebí mít nainstalovány následující aplikace:

.NET Framework 3.5 verze s SP1 a Windows installer 4.5 verze s SP1 (SP – service pack).

Oba balíčky je třeba instalovat v jakékoli verzi SQL Serveru 2008, a to jak ve verzích Standard, Enterprise a v dalších, nebo v režimech standalone nebo cluster.

Obě aplikace máme nainstalovány, proto můžeme pokračovat dále. Pokud by tomu tak nebylo, museli bychom restartovat oba nody a zopakovat průvodce instalace.

Licenční číslo a licenční ujednání si samozřejmě pečlivě přečteme a můžeme pokračovat dále

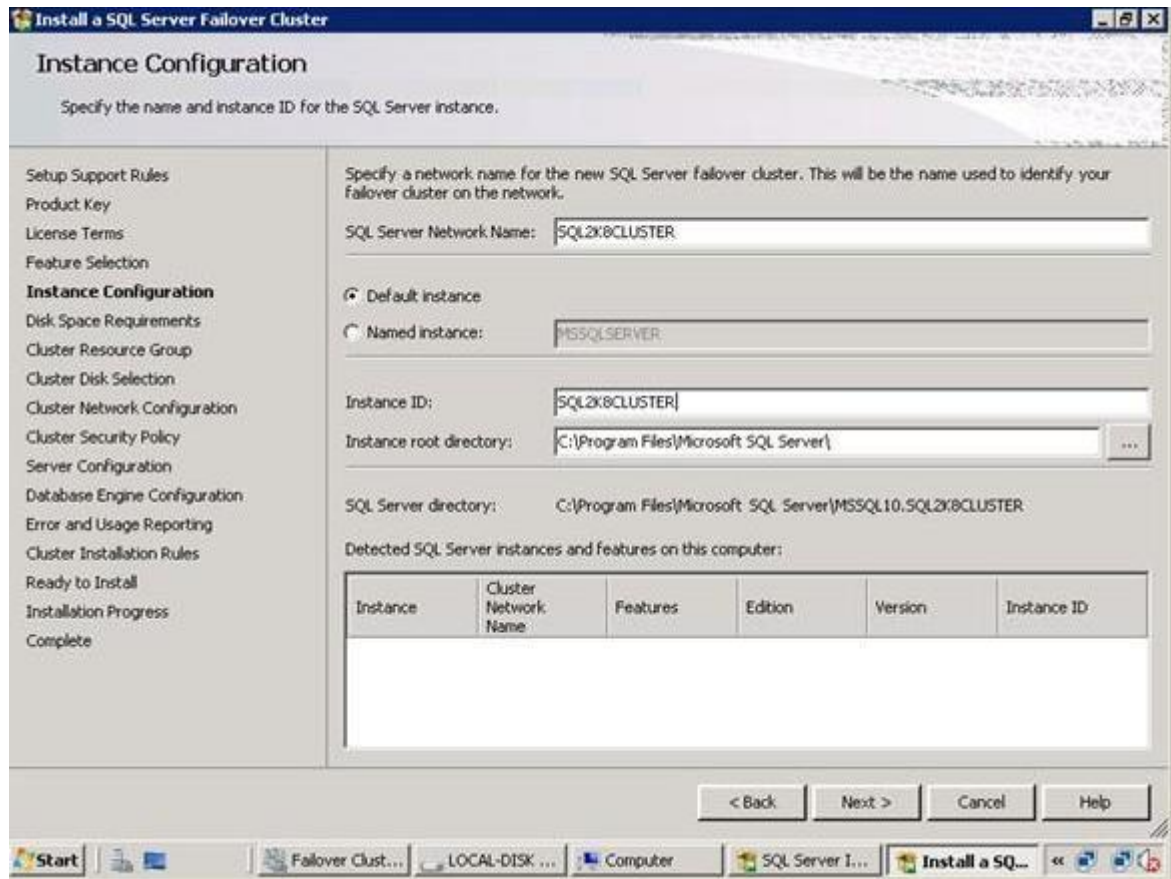
Na dalším kroku se nám spustí validátor stavu OS a HW. Ověřuje se celá řada aplikací, ať už ve verzích, nebo dostupnost dané služby. Na tento test jsme si už zvykli. Kontrolujeme varování (Warnings), musíme odstranit FAILED služby. Tak jak ve verzi SQL Serveru 2005, všechna varování odstraňovat nemusíme, FAILED ano. Pro jistotu zopakují to, co u instalace SQL Serveru 2005 – Instalujeme na servery jen to nejnnutnější z důvodu rychlosti, zatíženosti, údržby a bezpečnosti. Takže některá varování můžeme v klidu ignorovat. Ignorujeme ta varování, které služby nebudeme nikdy na daném serveru potřebovat, nebo ty záplaty, které nejsou žádoucí pro náš server.

Po úspěšné kontrole se dostaneme na seznam dodávaných utilit a služeb, kde opět volíme tzn. instalujeme jen ty, které budeme používat. Opět uvažujeme tak, že instalujeme jen to, co server potřebuje. Instalace řady komponent je vhodná například jen na administrátorskou stanici (BOOKs Online, Tools apod.).

Pokračujme dále. Čeká nás definice SQL Server Network jména pod kterou se bude SQL služba prezentovat vůči aplikacím a stanicím připojené k SQL Serveru.

Zvolíme instalaci, v našem případě první default instanci. Proč default a proč budeme instalovat jen jednu a co to instance je, si vysvětlíme dále v kapitole o robustním řešení drive clusteringu. Nyní instalujeme pasive cluster a proto vyberme default instanci.

Do položky Instance ID vyplníme název SQL serveru a tento název bude použit pro defaultní instanci sql služby, tedy zdroje. V položce Instance ID doporučuji napsat stejné jméno jako jméno uvedené v SQL Serveru network name.



Obr. 12: Instalace SQL 2008 v Cluster modu

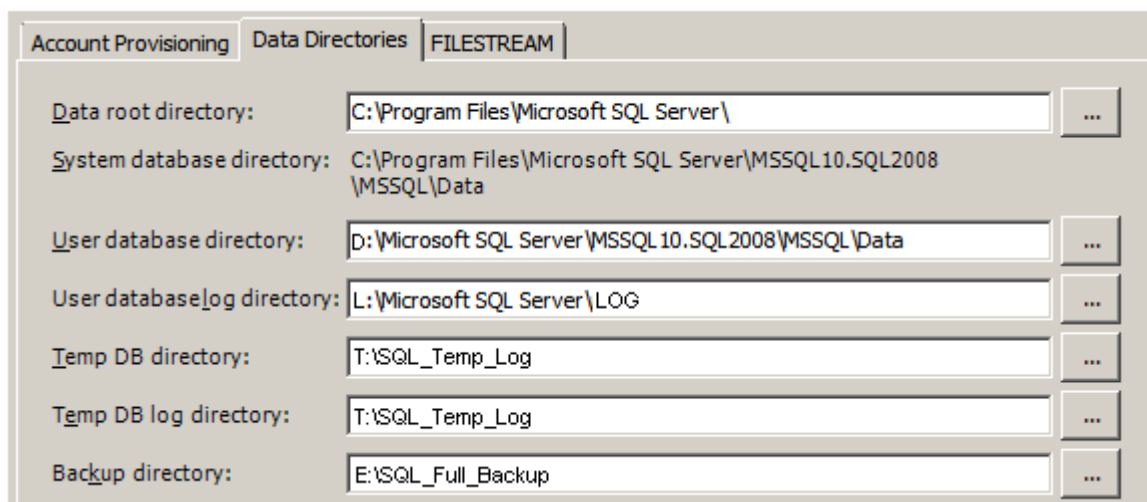
Cesta k aplikační části služby SQL (Instance root directory) je cesta, kde budou uloženy aplikační soubory služby SQL Serveru, neboli binaries data.

Můžeme pokračovat dále, vybereme lokální disk, nejrozzumnější je použít systémový disk. Pozor na velikost volného místa (doporučuji aspoň 5GB volného místa na systémovém disku, Instalace zabere cca 1GB, ale režie a rezerva systému je vždy nutná) a můžeme pokračovat.

Na další obrazovce si můžeme zvolit, do jaké skupiny failoveru bude zařazena služba sql serveru společně s dalšími podpůrnými zdroji sql serveru (Menu Cluster Resource Group). Stejně jako u SQL Serveru 2005 doporučuji samostatnou skupinu failoveru z důvodu systematičnosti a také přehlednosti. Samostatnost této skupiny se nám může v budoucnu hodit například při restartu služeb ve skupině a podobně.

Důležitým bodem je následující krok. Zde se definují cluster disky, které budou použity pro SQL službu. Před instalací SQL Serveru jsme si vytvořili a pojmenovali disky pro Data a Logy.

Proto je nyní zařadíme jako klastrové. Je důležité provést toto nastavení právě teď, protože se nám provází zdroje uvnitř cluster SQL grupy ve Failoveru. Na serverech máme ještě další disky jako Qourum disk, MSDTC disk, ty ale už máme nakonfigurovány (tyto disky jsou automaticky zařazeny do výpisu). Disk pro TEMP db, pokud máme, tak jej rovněž zařadíme do failover clusteringu.



Obr. 13: Instalace SQL 2008 v Cluster modu 2

Na další obrazovce máme nastavit IP adresy pro clusterovou službu SQL serveru. Tahle služba je migrační zdroj, kterou lze chápat jako virtuální. Virtuální chápeme tak, že s pádem serveru nebo služby failover clusteru nebo výskytem nějakého problému dojde k migraci služeb a migruje i IP adresa naší služby SQL serveru. Jako vtípek programátorů lze považovat nastavení IP adresy brané z DHCP serveru. Nenapadá mě, kdy pro statickou virtuální službu vyšší dostupnosti je vhodné použít dynamickou IP adresu. Prostě tuhle volbu odškrtneme a zapomněl jsem dodat, že IP adresu volíme z rozsahu veřejné IP adresy. Následující část průvodce nás zavede ke konfiguraci Cluster Security Policy. Zde použijeme defaultní volbu a to Use service SIDs. Ve verzi SQL Serveru 2003 jsme mohli specifikovat skupinu pro všechny SQL Servery (služby), ale ve verzi SQL Serveru 2008 toto již nedefinujeme a je to recommended (výchozí) volba instalace.

Pokračujeme dále a blížíme se doufejme ke konci definice průvodce. Čeká nás nastavení služeb, respektive pod jakými účty se budou spouštět dané služby. Zde je dobré se zamyslet nad tím, co je doporučeno a co je pro nás nejvhodnější. Doporučené nastavení je použít doménový administrátorský účet pro všechny služby, které se nám zde nabízí. S jedním účtem pro všechny služby se dá souhlasit asi vždy, ale zda použít doménový účet, těžko říci. Jelikož máme Active Directory na jiném serveru, co se bude dít, pokud tento

AD Server zkolabuje, nebo služba nebude dostupná? SQL služba nám začne zběsile kolabovat při volání autentizace a pokud dojde i k migraci SQL zdrojů, tak jsme rázem nahraní. Proto zvažme, zda není vhodnější použít lokální administrátorský účet pro sql služby (zdroje). Pokud se vrátím k jednomu účtu, tak jedinou výjimkou by snad mohlo být spouštění agenta, který zajišťuje spouštění maintenance plánu (zálohování, údržba db a podobně).

Další důležitým nastavením je autentizace vůči SQL serveru. Mixed autentizace je autentizace, kdy se k SQL serveru dá přistupovat jak pod doménovým účtem, tak i pod účtem SQL serveru. O licenčních ujednáních zde ale mluvit nebudeme, to je další obšrná kapitola, která by zabrala mnoho řádků a jelikož se politika firmy Microsoft mění poměrně často, je dobré se vždy obrátit na dodavatele systému, který nám prodal licence.

Na první záložce tedy definujeme autentizaci a účet pod samotným SQL, kterým se budeme moci autentizovat k SQL Serveru. Sysadminovské přístupy budou automaticky vytvořeny dva, první je ten účet, který zde uvedeme a pak účet **sa**, který je zde z historických důvodů.

Na druhé záložce definujeme kde, jaká a na jakém disku, budou data sql serveru uložena. Proto jsme si přichystali již dříve disky pro Logy, Data případně temp disk. Rozdělení těchto dat je nesmírně důležité a řešili jsme to již dříve.

Za zmínku ještě stojí backup directory. Doporučuji zvolit další samostatný disk pro tato úložiště. Dokonce bych volil pokud to je možné takový diskový oddíl, který není svázán s cestou k externímu poli (switchi). Laicky by se dalo říci, oddělit tento oddíl od všech prostředků, které SQL server používá. Při totální havárii switchu nebo externího pole se hravě dostaneme k zálohám a o to nám u záloh jde. Pokud zálohujeme jinými nástroji třetích stran, přesto tohle nastavení doporučuji neignorovat. S SQL zálohami se jednoduše pracuje a jakákoliv manipulace například s páskami a obnova db pro nějaké testování a podobně, je velice náročná.

Už jsme na konci průvodce, čeká nás už jen definice reportu pro Microsoft. Já doporučuji nic nikam neposílat.

Na další obrazovce nám proběhne kontrola SW a HW prostředků a definice, jak jsme průvodce nakonfigurovali (Cluster Instalation Rules a ready to Install).

Nyní můžeme spustit instalaci.

Instalace je dokončena. Ověříme cluster zdroje služby SQL v aplikaci Failover Cluster Manageru a provedeme po instalační kroky:

Oproti verzi SQL Serveru 2003 se nám služba SQL Serveru nainstalovala pouze na 1 nod. Důvod, proč to v průvodci instalace SQL Serveru 2008 není a proč se Servery nepřidávají automaticky, je jednoduchý. Na serverech, kde běží služba SQL serveru 2008 v cluster modu, nesmí být služba Active Directory. Na SQL Serveru 2003 to bylo nutností, zde AD být nesmí.

12.5 Přidání dalšího nodu pro SQL Server 2008

Přidání dalšího serveru do clusteru bude velmi podobné jako instalace nového clusteru, pouze budou vynechány konfigurační nastavení služeb a clusteru. Z tohoto důvodu uvedu v bodech jen stručný popis a postup, ale vysvětlivky nebudeme opakovat.

Spustíme instalaci z média a tentokrát vybereme položku Add node to a SQL Server failover cluster (přidáváme další nod do clusteru pro SQL zdroje).

Průvodce prověří stav systému a kontrolu nainstalovaných potřebných aplikací a balíčků, potvrdíme licenční ujednání a zadáme licenční číslo.

V dalším kroku se nám spustí validátor stavu OS a HW.

Dále se provede ověření Cluster Node Configuration, který ověří načtení údajů pro zařazení do clusteringu.

Na další obrazovce specifikujeme, do jaké skupiny failoveru jsme zařadili službu sql serveru společně s dalšími podpůrnými zdroji sql serveru (Service Accounts). Nadefinujeme stejný domain účet jako při instalaci prvního nodu, tím docílíme stejného nastavení pro přidaný nod. Tzn. bezproblémové volání služeb SQL zdrojů.

Opět test s výsledkem testů (Error and Usage Reporting).

Přidání nodu do již existující skupiny Failoveru a kontrola dostupnosti (Add Node Rules).

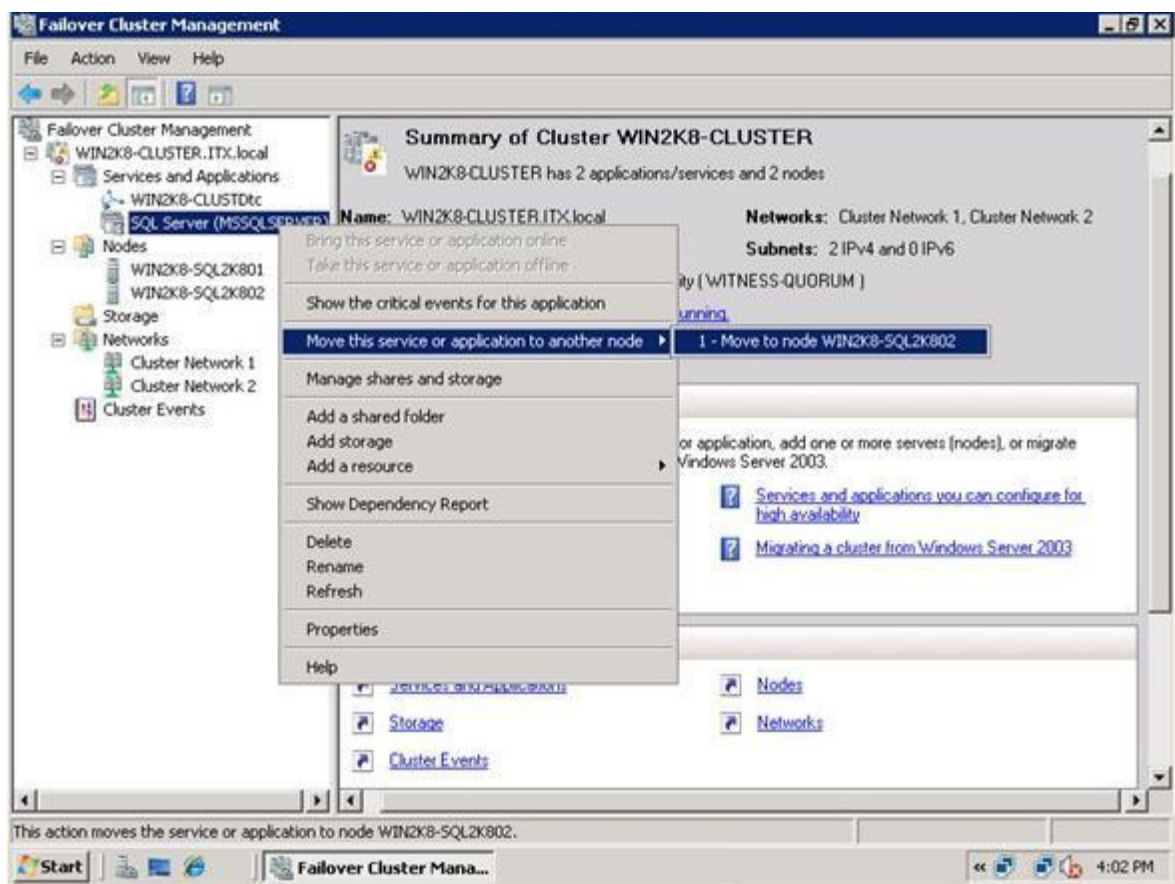
Průvodce vygeneruje sumář nastavení ke zhlédnutí a po potvrzení instalace se zahájí proces začlenění.

Instalace je u konce a sever se stává členem Failoveru.

12.6 Kontrola funkčnosti clusteru

Testy provedeme stejné jako v případě Clusteru pod OS 2003. Dané testy jsou v kapitole ověření chodu clusteru – Test funkčnosti a reálná migrace služeb:

Co se změnilo je prostředí, kde u OS 2003 vše provádíme přes Cluster Administrator, u OS 2008 to je Failover Cluster Manager. Aplikace se nadají ani v nejmenším srovnat, ohledně obsluhy, protože práce v nich je naprosto odlišná. Funkce a procesy na pozadí se nezměnily z pohledu administrátora, pouze se vyvolají z jiného místa a postup je rovněž jiný.



Obr. 14: Instalace SQL 2008 v Clusteru modu 3

Migrace Serveru pod OS 2008 respektive Failover Cluster Manageru

Přes položku název clusteru, volbou Move the services or application to another node, se dostaneme do seznamu možných serverů, na kterých lze provést migraci. Pokud máme 2 nody, v nabídce se nám zobrazí pouze jeden server, na který lze zdroje zmigrovat.

12.7 Post instalační rutiny

Zrestartujeme všechny nody (začneme s pasivním nodem, který vypneme a poté zrestartujeme aktivní nod. Po náběhu můžeme zapnout i druhý nod)

Spustíme aktualizaci SP pro SQL server na aktivním nodu a opět provedeme restart serveru, tentokrát ale stačí jen aktivního nodu.

12.8 Instalace Service Packu na SQL 2008 se provádí následovně

Provedeme instalaci na každém nodu kde SQL služba běží, a to v pořadí. První aktive nod a poté v libovolném pořadí pasive nody. Po dokončení instalace doporučuji zrestartovat všechny servery. Pasivní nody clusteru vypneme a až zrestartujeme aktivní nod, poté pasivní nody zapneme a zkontrolujeme příkazem v Management Studiu, zda vše proběhlo v pořádku.

```
SELECT @@VERSION, GO
```

13 ZRCADLENÍ DATABÁZÍ MS SQL (MIRRORING)

13.1 Praktická realizace pomocí průvodce

Zvolím metodu vytvoření zrcadlení pomocí průvodce v naší známé utilitě Management Studio. Skriptem je to mnohem jednodušší, ale skript je pro neznalého administrátora méně čitelný.



Obr. 15: Instalace mirroringu MS SQL

13.2 Popis instalace

Nastavíme na databázi, kterou chceme zrcadlit full recovery model. Doporučuji, pokud to je možné, i compatibility level minimálně 90 (SQL 2005), 100 (SQL 2008).

Provedeme FULL Backup databáze a následně provedeme restore DB na druhém serveru s tím, že recovery provedeme s parametrem NORECOVERY (z důvodu konzistence s full backupem).

Na serveru kde bude hlavní databáze klikneme na databázi – Task – Mirror a zvolíme Configure Security. Zvolíme, zda budeme používat roli svědka.

Na další obrazovce definujeme, zda chceme od samotného začátku řídit zrcadlení svědkem.

Automaticky je vybrán hlavní (Principal) server a nastavíme Listener Port a Jméno EndPointu pro Principal Server Instanci. Doporučuji ponechat zatržítko Encrypt.

Na další obrazovce vybereme zrcadlenou roli (server kde bude zrcadlo), zadáme port (doporučuji o jedno větší než jsme zadali u principal serveru), zadáme stejný název EndPointu a ponecháme zatrženou volbu Encrypt.

Totéž pro Witness (svědka) roli, port opět o jedno větší, než jsme zadali u zrcadla. Takže bude platit, že každá role má svůj Listener port. Na další obrazovce definujeme, pod jakými účty poběží zrcadlení na všech serverech s danými rolemi. Posledním krokem je výpis naší konfigurace. Provede se instalace na serverech a proběhne spuštění zrcadlení.

Pokud se vrátíme do nastavení databáze – Task – Mirror, tak zde uvidíme, že služba zrcadlení běží, pokud jsme tedy potvrdili v předešlém kroku spuštění mirroringu. Vidíme zde i porty všech rolí serveru a důležitá věc je nastavení Operating mode. Zde nastavujeme v jakém režimu zrcadlení poběží, máme na výběr jak synchronní tak asynchronní režim. Posledním důležitým leč nenápadným stavem je STATUS s tlačítkem znovuobnovení.

Ensure that security is configured for mirroring this database. Configure Security...

Server network addresses:

Principal: Start Mirroring

Mirror: Pause

Witness: Remove Mirroring

Note: Use fully-qualified TCP addresses. For example:
TCP://svr5.corp.abc.com:5022 Failover

Operating mode:

High performance (asynchronous) -- Commit changes at the principal and then transfer them to the mirror.

High safety without automatic failover (synchronous) -- Always commit changes at both the principal and mirror.

High safety with automatic failover (synchronous) -- Requires a witness server instance. Commit changes at both the principal and mirror if both are available. The witness controls automatic failover to the mirror if the principal becomes unavailable.

Status: Refresh

Obr. 16: Instalace mirroringu MS SQL

13.3 Schéma instalace mirroringu

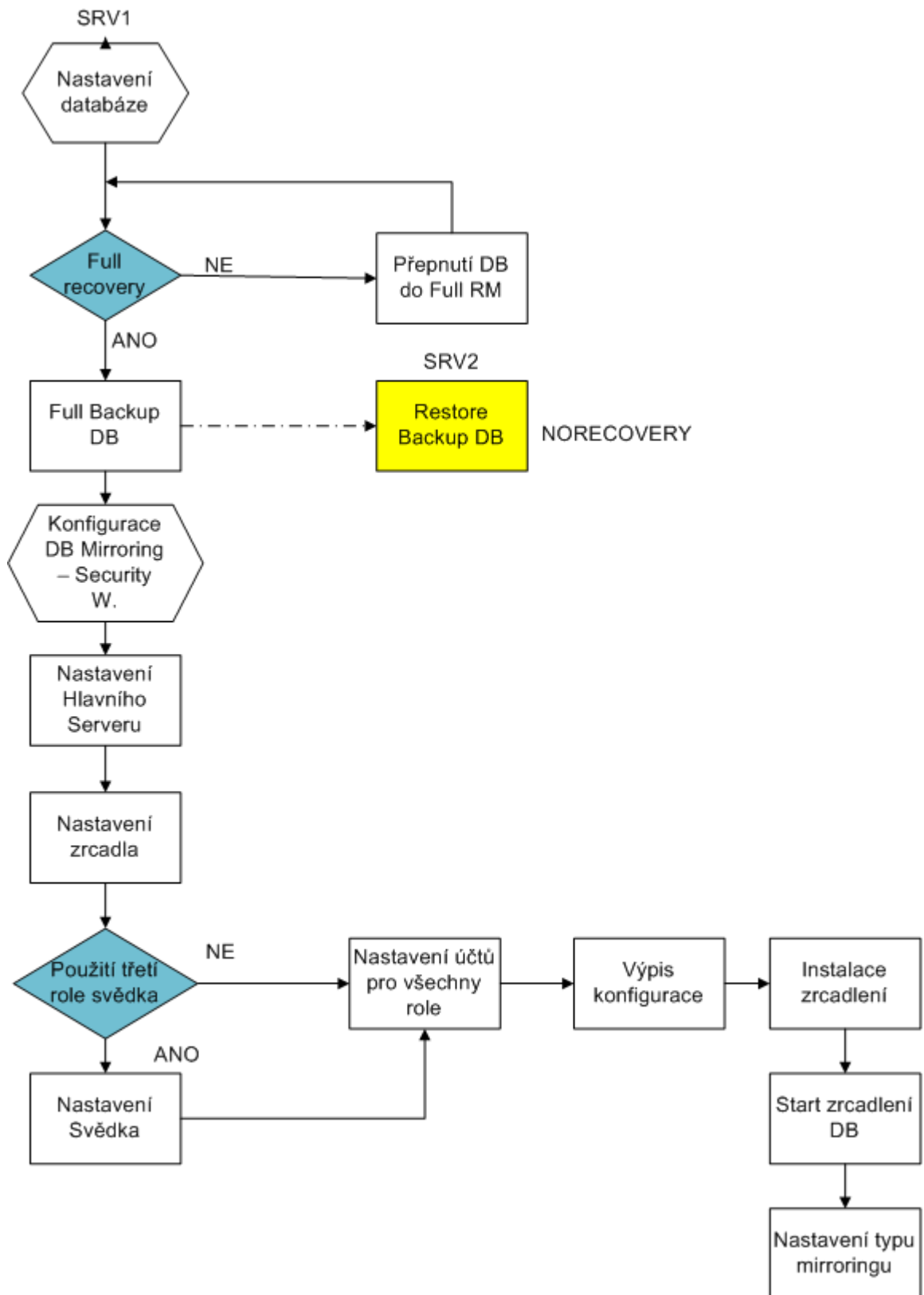


Schéma instalace MS SQL Mirroringu

14 PŘEDÁVÁNÍ TRANSAKČNÍCH LOGŮ – LOG SHIPPING

14.1 Popis instalace

Před instalací log shippingu musíme nastavit řadu věcí, které jsou podmínkou pro správný chod tohoto řešení.

Vytvoření nového účtu na všech serverech, kterých se dotkne předávání transakčních dat.

Nastavení sdílení. Je třeba, aby jsme určili server, kde budeme v pravidelných intervalech ukládat transakční logy a rovněž, kam budou mít přístup všechny sekundární servery. Vhodné je použití serveru vyčleněného v síti, jehož úloha je file server a pomocí sdílení, či mapování disků nastavit práva tak, aby pod účtem, který jsme vytvořili v předešlém kroku, mohli s takovýmto místem pracovat.

Přístupová práva jak pro server primary tak i secondary musí být Full Access.

Kontrola síťových karet, zda jsou nastaveny pro komunikaci při rychlosti 1GB.

Kontrola služby SQL Server Agent. Služba musí být na všech serverech spuštěna, jelikož právě tato služba bude automaticky provádět pravidelné operace s předáváním dat a následných procesů.

14.2 Instalace

Nastavení databáze do režimu Full Recovery modelu.

Full Backup (provedeme plnou zálohu) databáze a zálohu překopírujeme na secondary server(y).

Poté provedeme backup transakčního logu dané databáze a opět překopírujeme zálohu na secondary server(y).

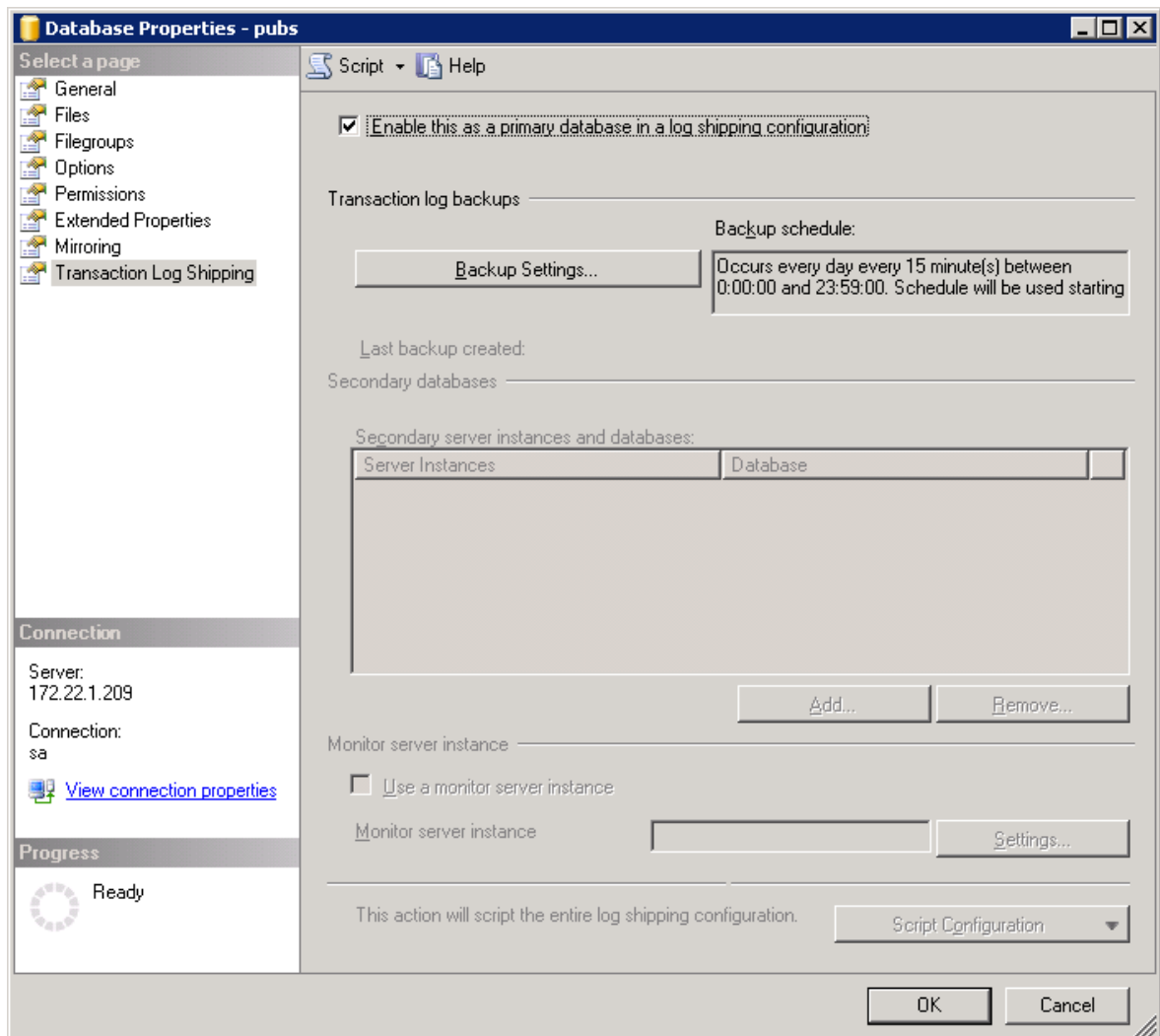
Po překopírování záloh, provedeme obnovu na secondary serveru v pořadí full backup a následně i obnovu transakčního logu. Obě zálohy je třeba obnovit do db se stejným jménem a v režimu RESTORE WITH STANDBY.

Na primární databázi provedeme shrink transakčního logu (příkazem nebo volbou shrink v aplikaci Management studio)

Tím máme vše nachystáno pro zprovoznění log shippingu.

14.3 Konfigurace

Ve vlastnostech databáze na záložce Transaction Log Shipping zapneme volbu Enable this as a primary database in a log shipping configuration.



Obr. 17: Instalace Log Shipping

Provedme nastavení pomocí Backup Settings.

Nastavíme síťovou cestu pro zálohy transakčních logů (UNC cesta). Povedeme konfiguraci automatické úlohy pomocí Scheduleru. Scheduler vykonává SQL Agent, který jsme už nastavili.

V této fázi definujeme, jak často se budou zálohovat transakční logy a také definujeme zpětnou kontrolu (Alert). Tato kontrola by měla být minimálně dvakrát tak delší, než je opakující se čas zálohy.

Rozumné je nastavit i automatické odmazávání transakčních logů a kompresi u záloh (pouze u sql 2008).

V dalším kroku přidáme nový sekundární server a to pomocí tlačítka ADD.

Klepnutím na tlačítko Connect na první záložce se připojíme ke svému sekundárnímu serveru, kde definujeme název databáze, kterou zde již máme obnovenu, takže název db neměňme.

Na záložce Initialize Secondary Databáze zvolíme, že databázi už máme na secondary serveru připravenou a to volbou No, the secondary database is initialized.

Na záložce Copy Files nastavíme cílovou sdílenou složku, do které se budou kopírovat zálohy transakčního protokolu.

Musíme zde nastavit i interval kopírování souborů. Ten nastavíme stejný jako interval pro zálohování transakčního logu.

Na záložce Restore Transaction Log nadefinujeme, do jakého stavu se nám bude secondary databáze obnovovat. Máme dvě volby a to no recovery mode a standby mode. No recovery mode je režim, kdy db je v konzistentním tvaru jako master databáze, ale uživatelé s ní nemohou pracovat. Standby mód je přístupný i pro koncové aplikace, ale pouze pro čtení. Při obnově databáze v tomto režimu je vhodné odpojit všechny aktivní konekce kvůli zaručení správnosti reprezentovaných dat. Tato volba je ovšem pro koncové aplikace nepříjemná, proto je vhodné tento režim použít jen v případě, kdy obnovu secondary databáze neprovádíme příliš často.

Nastavíme zpoždění (data restoring), Alert a plán obnovení.

Po nastavení Backup Settings můžeme dále definovat Monitor server Instance, který automaticky kontroluje stav služby. Vše se opět děje přes SQL Server Agentu.

Vše nyní máme nastaveno a můžeme potvrdit volbou OK. Tím se vše zkonfiguruje a služba začne být aktivní dle scheduleru.

Na závěr pustíme SQL Server Agentu a v úkolech najdeme Log Shipping. Zkontrolujeme stavy úloh a pozorujeme, zda nedojde v některém z kroků k chybě.

Na kartě Log File Viewer v job history můžeme vidět, jak dopadly všechny předešlé úkoly a vidíme zde i časy, jak dlouho job běžel. Vidíme zde, zda jsme správně zvolili časové odstupy mezi navazujícími úkoly.

14.4 Schéma instalace Log Shippingu

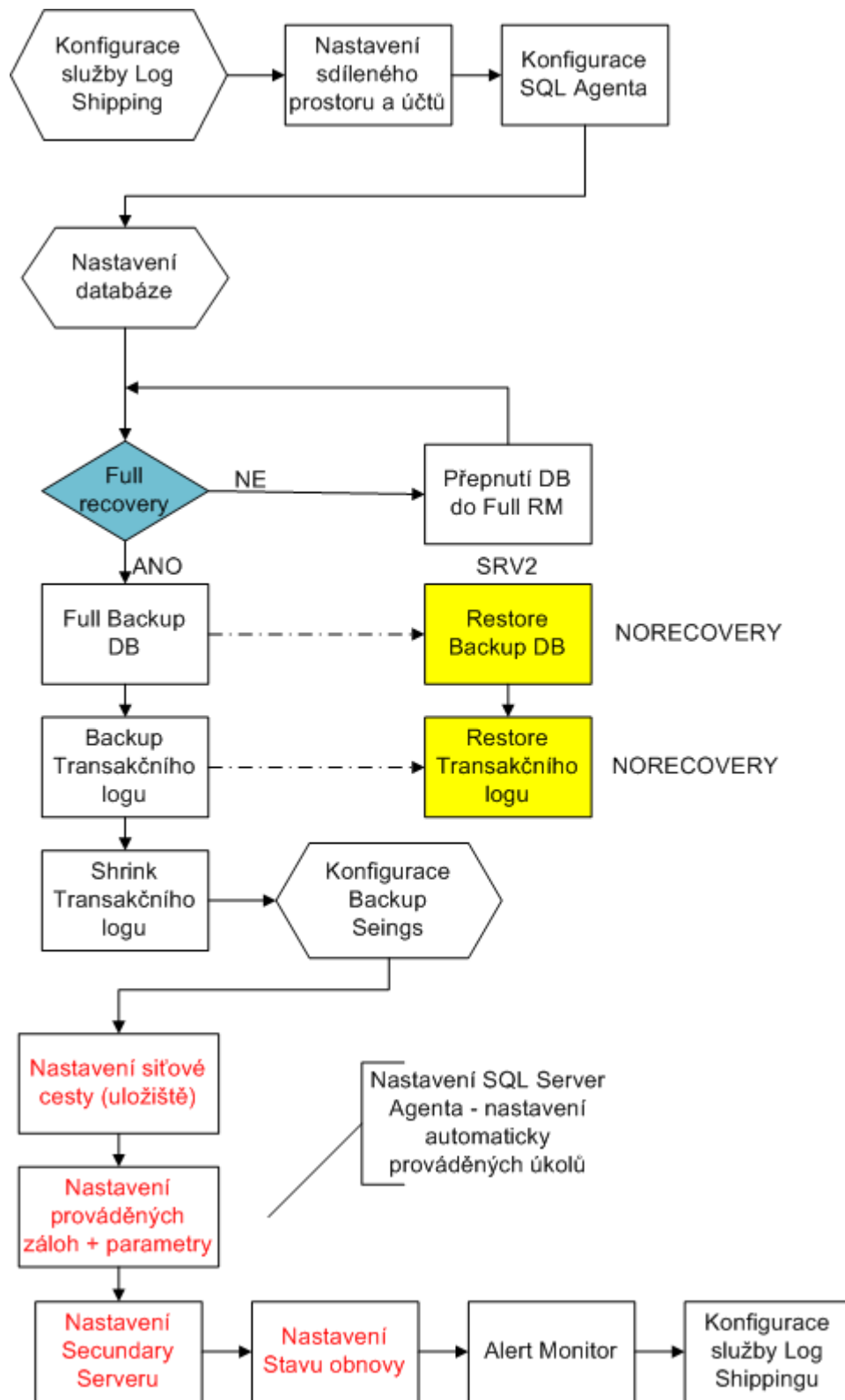
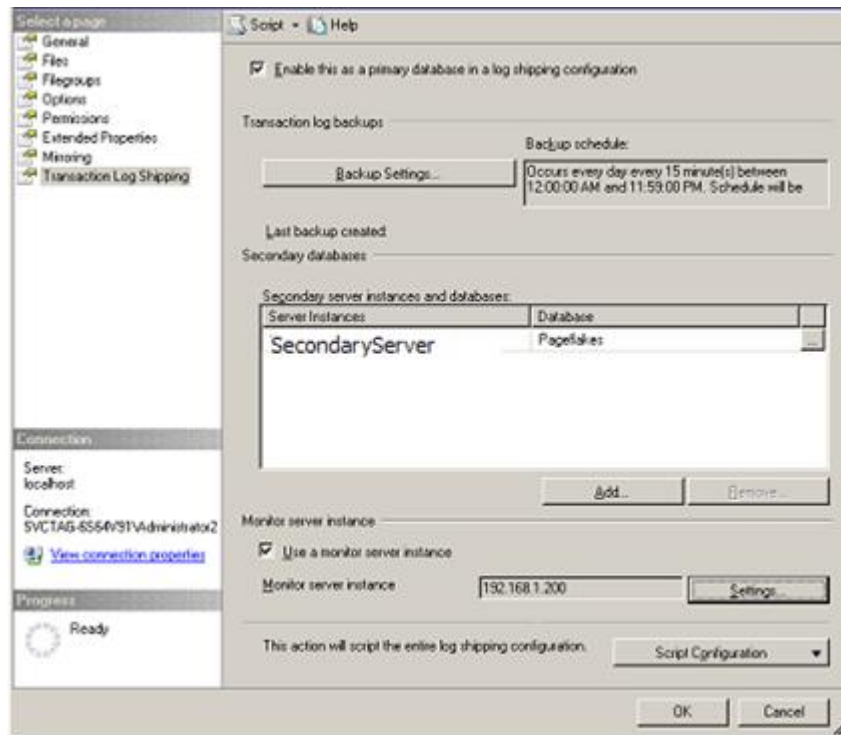


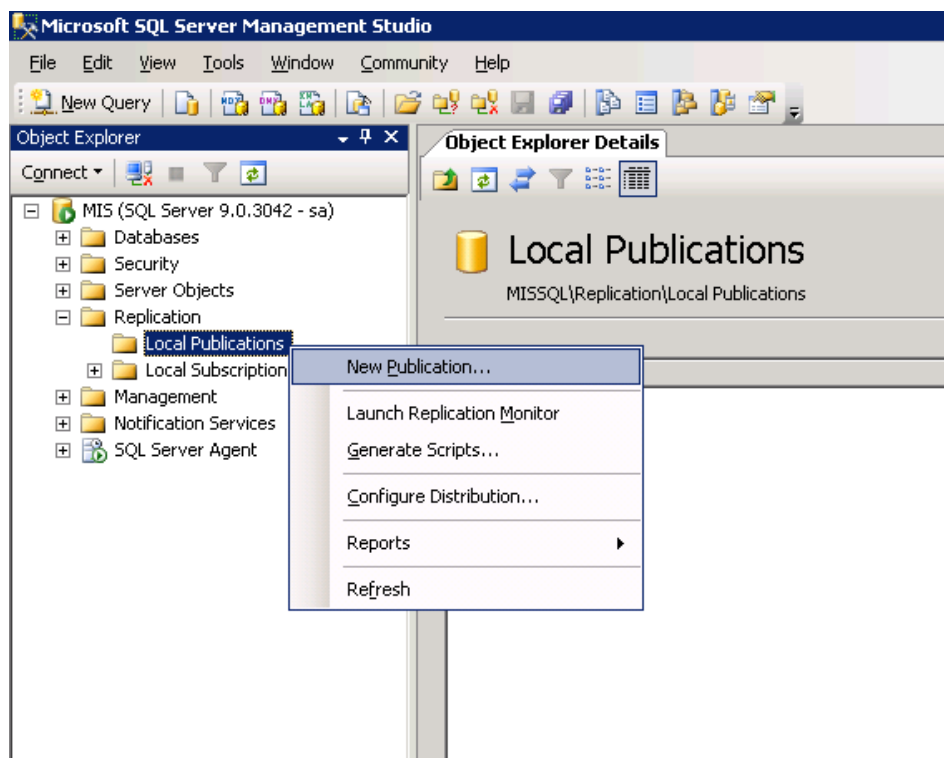
Schéma instalace MS SQL Log Shipping

Obrázek Dokončení Instalace Log Shippingu



Obr. 18: Instalace Log Shipping 2

Obrázek z následující kapitoly 16 - Instalace MS SQL Replikace



Obr. 19: Instalace Replikace

15 REPLIKACE

15.1 Nasazení replikace v provozu

Replikace byla vytvořena na základě potřeb firem s vysokou dis-akací na velké vzdálenosti. Veškeré technologie dostupnosti, které jsme zde popisovali, neřeší to, co role replikace. Klasickým příkladem je firma o několika pobočkách, kde každá pobočka si řeší své potřeby, mají své samostatné servery a je potřeba nějak provádět synchronizace části dat mezi sebou. Modely replikace, které jsme zde popsali, dokáží při správném zvolení typu replikace zajistit právě to, co potřebujeme. To znamená, zda potřebujeme aktuální data v daný okamžik, nebo pracuje li firma v místech, kde není možné zajistit stálou datovou komunikaci s centrálou.

Na samotném začátku zprovoznění replik je třeba provést full backup databáze a následný restore na replikace. Je potřeba si uvědomit, že v této fázi přípravy je potřeba operaci udělat nad databází a ne tabulkou. Potřebujeme replice říci, jaké tabulky má replikovat, ale musíme zde splnit podmínky jako: stejný název databáze, stejný model db, full recovery model db, totožné schéma db a až poté můžeme replikovat záznamy v tabulkách, nikoliv že bychom mohli v databázi mít jiné tabulky či struktury a podobně.

15.1.1 Vytvoření distributora

Distributor je databáze, která prezentuje svá data na repliky, proto předpokládáme, že bude někde v hlavní serverovně.

15.2 Postup Instalace

Rozhodit replikaci není nic složitého, jen je třeba si uvědomit, co od systému vlastně chceme.

Spustíme si Management Studio, předpokládám, že už nějakou uživatelskou databázi v SQL Serveru máme a přes volbu Replication spustíme Configure Distribution a zvolíme distribuční server.

Ověříme automatické spuštění SQL Server Agentu pod platným účtem s administrátorskými právy nad daným počítačem.

Konfigurace Snapshot Folderu. Je to místo na disku, kde se budou ukládat replikovaná data pro agenty, kteří s touto replikou budou dále pracovat.

Konfigurace Distribuční databáze. Zde uvedeme cestu k distribučním souborům a zvolíme jméno databáze distribuce.

Nyní máme distribuční role vytvořeny a můžeme vybrat data, která chceme publikovat do repliky. To provedeme přes volbu Local Replications.

Vyberme databázi, které se replikace bude týkat. Respektive z které budeme replikovat nějaké tabulky či záznamy.

V dalším kroku se dostaneme do nabídky, kde definujeme, jaká data z dané databáze budeme publikovat. Neřekli jsme si, že aby šla replikovat tabulka, musí mít primární klíč (jedinečné číslo záznamu).

Na další obrazovce můžeme určit, jaká část tabulky se bude replikovat. Neboli jaký sloupec (jakou položku) chceme replikovat. A také filtr Table, kde můžeme přímo vyselektovat konkrétní části dat.

Snapshot Agent – Definujeme snímek tabulky, který zamkne všechny záznamy a následně tato data jsou kopírována do replikačního souboru. Lze říci, jak často nám SQL Agent provede tohle snímkování dat.

Na další obrazovce definujeme Snapshot agent Security. Není to nic jiného než nastavení práv na daný snímek.

Tím je jedna část replikace hotová. Je nám vygenerován kompletní záznam nastavení pro Distributora.

To znamená, data už poskytujeme, teď je potřeba nastavit druhou stranu.

Pod kategorií replikace klikneme na Publication a vybereme volbu New Subscriptions. Tím vytvoříme repliku, která bude data sbírat z distributora.

Vybereme ze seznamu Publisheru databázi a vybereme naši pojmenovanou repliku. Dále provedeme autentizaci na distributora.

Na další obrazovce potvrdíme roli Subscribera a databázi, která je zdrojem dat.

Dále definujeme autentizaci MS SQL Agentu, který zajišťuje chod replikace.

Už nám zbývá jen říci, kdy synchronizace repliky proběhne a jak často. Vše pomocí SQL Agentu.

Tím se vydefinuje i subscriber (replika) a pokud nepotřebujeme nic dodatečně změnit, můžeme synchronizaci spustit.

Pustíme si Replication Monitor, ten ověří znovu načtení úloh synchronizace, kterou jsme definovali a spustí sadu procesů zajišťující replikaci. Ve výpisu můžeme vidět, jak dlouho každá operace trvala, případně nám vypíše chyby replikace, které musíme následně opravit.

15.2.1 Schéma Instalace Replikace

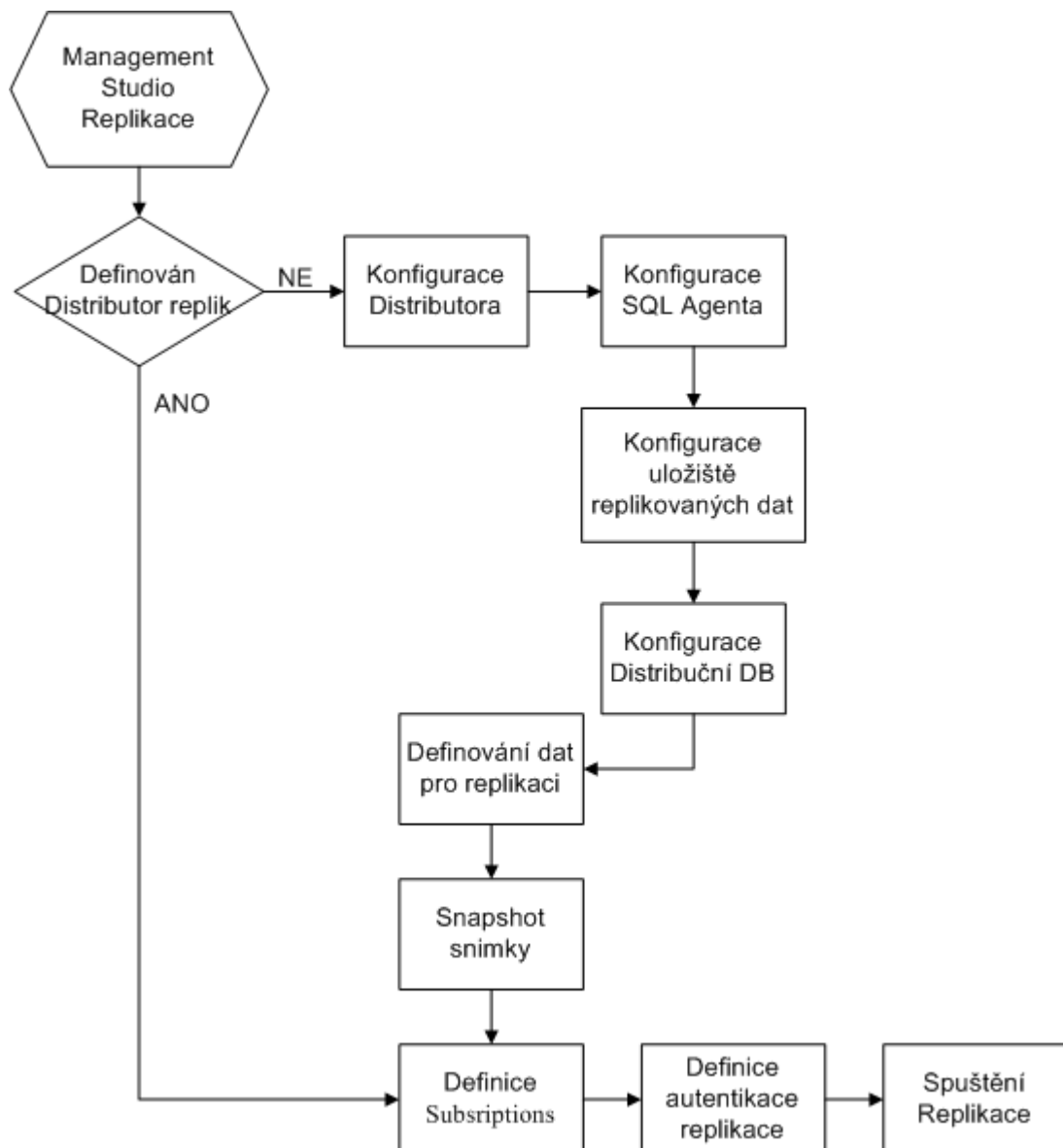


Schéma instalace MS SQL Replikace

16 PRAKTICKÁ REALIZACE SQL SERVERU POMOCÍ VIRTUALIZACE – STANDALONE VERZE

Instalace SQL serveru pomocí virtualizace nám rovněž přinese vyšší dostupnost této služby a to pomocí služeb virtualizace. Popíše zde virtualizaci pomocí nejsilnějšího hráče na trhu a to virtualizaci pomocí VMWARE. HyperV od Microsoftu jistě nebude v budoucnu špatný nástroj, ale pro dnešní dobu je produkt VMWARE mnohem více robustní a kompaktnější. Xen postavený na Linuxu nebo HyperV prozatím nedosahují takových kvalit jako virtualizace od VmWare.

Popis virtualizace a jak virtualizace funguje, není tématem této práce, proto zde bude popsáno pouze řešení týkající se sql služby a možnosti, co z virtualizace lze použít a jak se systém nakonec chová.

Virtualizaci použijeme ve verzi Vmware Vcenter Enterprise+ v.4 ESX.

Vytvoření Stand-Alone Virtuálního stroje pro použití SQL Serveru vytvoříme pomocí nástroje pro management virtuálních strojů VMware vSphere Client.

V práci bude popsáno pouze to, jak vytvořit, zkonfigurovat a nainstalovat SQL Server s OS a podpůrnými aplikacemi. Práce, jak postavit Virtual server a proč, a jak použít konkrétní verzi aplikace pro virtualizaci, zde popisovat nebudeme. Přistupujeme k Virtualizaci jako ke koncové službě SQL Administrátora.

16.1 Vytvoření Virtuálního stroje Stand-Alone služba stroje

Pomocí průvodce virtualizace v aplikaci VMware vSphere Client vytvoříme virtuální stroj pomocí volby Create new Virtual Machine.

Vytvoření probíhá pomocí průvodce, samotný proces vyvoláme až na konci průvodce, proto není třeba se bát vracet se v průvodci zpět, pokud chceme změnit některou z nadefinovaných vlastností VS.

16.2 Průvodce nás provede následující kroky

Definice jména a umístění Virtuálního stroje, správce Virtuálního stroje. Úložiště fyzických dat Virtuálního stroje (dále jen VS). Verze VS (možnost definovat VS ve zpětné kompatibilitě), doporučuji verzi VS-7 (nejnovější). Typ VS (definice OS a verze 32bit či 64b.) OS který budeme ve VS provozovat. Počet CPU ve VS (nenechme se zmást názvem

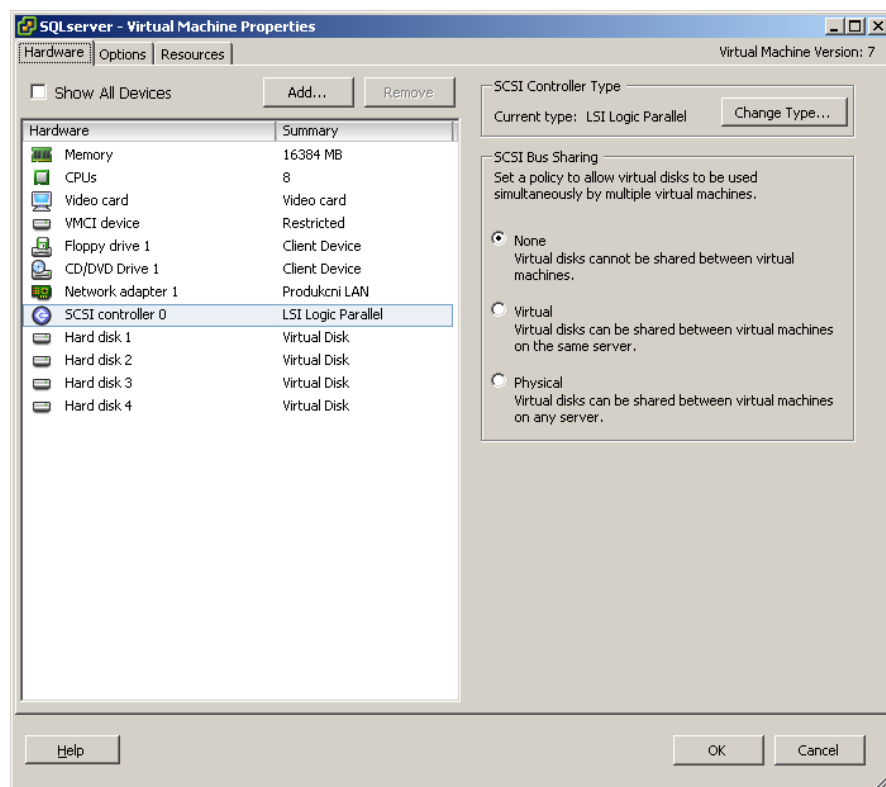
CPU, definujeme počet jader CPU). Paměť přidělená VS (max. velikost paměti). Komunikace LAN v různých režimech, výběr adaptéru LAN, NICs.

SCSI Controller (typ řadiče disku, volba přístupu k úložišti dat pro VS). Doporučuje se LSI logic SAS nebo VMware Paravirtual (pro naše účely LSI Logic Parallel). Výběr typu Controlleru je vhodné použít dle OS vzhledem k podpoře řadičů a práce operačního systému vs diskového subsystému. Vytvoření nového file souboru pro VS, nebo použití stávajícího z úložiště (my New). Velikost disku (velikost souboru z pohledu fyzického uložení) se kterým budeme pracovat ve VS. Umístění datového souboru VS.

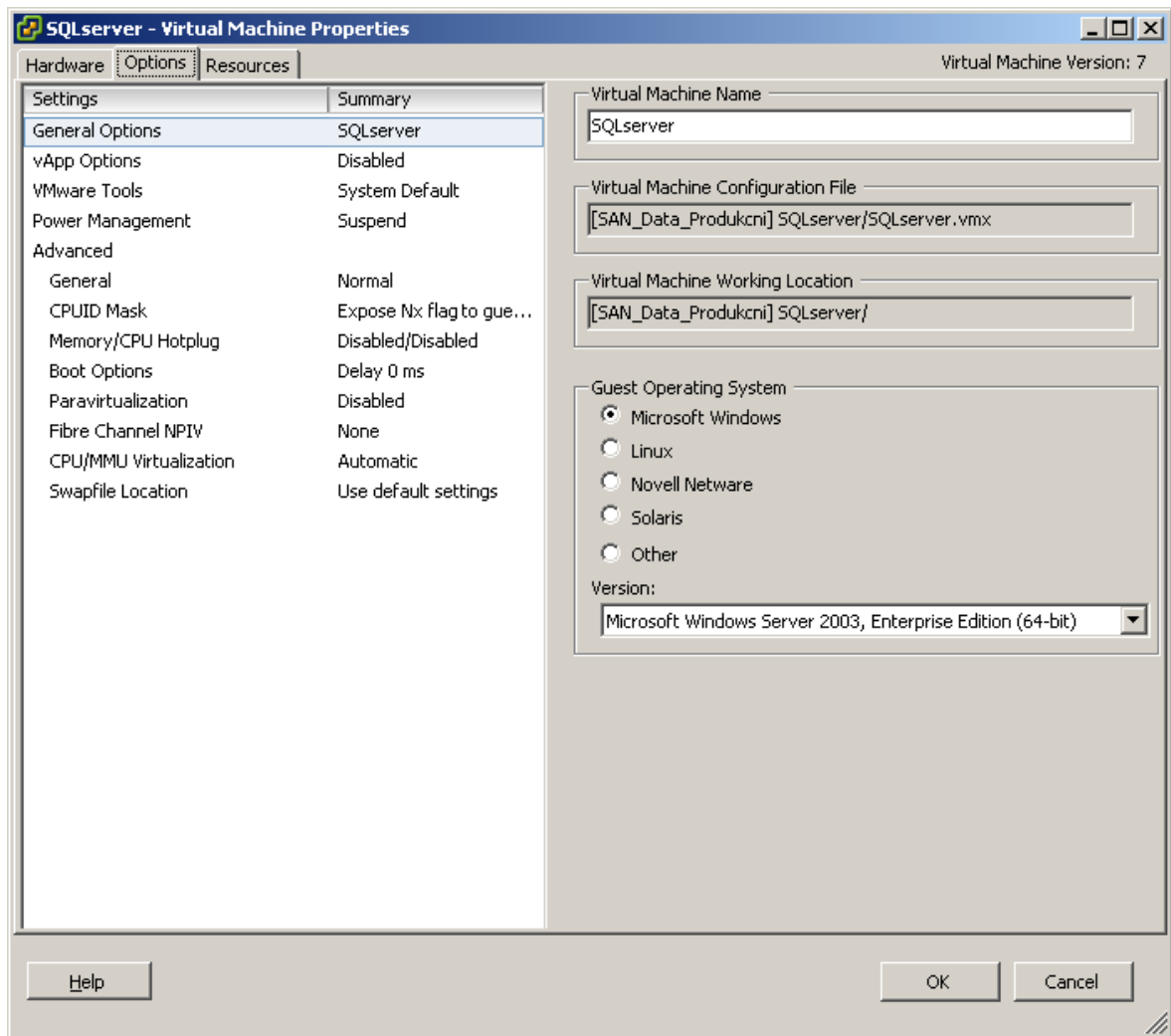
Controller (řadič), který bude obsluhovat konkrétní disk ve VS (ke každému disku můžeme přiřadit vlastní controller). Sumář instalace a po odkliknutí FINISH se ukončí průvodce a vytvoří se Virtuální stroj.

16.3 Konfigurace Virtuální mašiny pro Stand-Alone Server

Pro SQL server v Stand-Alone instalaci použijeme 1 controller se čtyřmi disky. (disky: OS, Data, Log, Zálohy), SCSI Controller bude typ LSI logic Parallel bez sharingu (neinstalujeme cluster). RAM 16GB, 8 CPU Jader, 1 LAN, 64bit, OS W2003 Ent. Ed. Fyzické úložiště dat na SAN Externím poli (Fibrechanell).



Obr. 20: Virtualizace SQL Stand-Alone



Obr. 21: Virtualizace SQL Stand-Alone

Nainstalujeme OS klasicky jako na fyzickém stroji. Protože budeme instalovat StandAlone verzi SQL stroje, je zde instalace totožná jak pro verzi OS 2003, tak i 2008 (my 2003).

Instalace driveru není ve virtuálním prostředí třeba, protože se o to postará samotná virtualizace, je zde ale virtualization tools, které zrychlí samotné ovládání virtuálního stroje a upraví některé drivery pro chod OS.

Provedeme klasicky upgrade všech komponent, případně provedeme instalaci posledního service packu.

Použijeme Snapshot utilitu před instalací samotného SQL. Tzn. Vypneme OS a pomocí nástrojů virtualizace provedeme snapshot disku.

Spustíme instalaci SQL Serveru. Pokud instalujeme verzi SQL Server 2005 (v našem případě Standard Edition 64b.), instalace nás upozorní na následující potřebné balíčky:

Provedeme instalaci Prerequisites tzn. Framework ve verzi 2.0 a vyšší, doporučuji 3.5, následně nainstalujeme Native Clienta SQL. Doporučuje se instalovat i Internet Information Server (IIS). Ten ale není k ničemu potřeba, tímto toto upozornění ignorujeme. Obecně zde platí to, že instalujeme jen ty služby, které na chod systému potřebujeme. Čím více věcí je nainstalováno, tím větší pravděpodobnost, že něco zkolabuje. O snižování výkonu při běhu nepotřebných služeb se nemá cenu ani zmiňovat.

Zahájíme tedy instalaci SQL Serveru a postupujeme dle připraveného instalačního balíčku.

Po kontrole připravenosti systému – System Configuration Check máme možnost instalovat konkrétní balíčky spojené s SQL. Opět doporučuji instalovat jen ty komponenty, které budeme opravdu potřebovat. Dále budeme definovat instanci SQL a pod jakým účtem poběží služba SQL. V rámci virtualizace je nejvhodnější použít default instanci a systémový účet pro běh SQL služby. Následně zde máme SQL autentizaci a collaction. Autentikaci volíme takovou, pomocí které se budou uživatelé hlásit vůči SQL serveru. Pokud máme účty přímo v SQL, volíme mixed autentizaci. Collaction doporučuji volit z nabídky SQL collaction a vybrat takovou, která nám nejvíce vyhovuje ke stávajícím databázím (case sensitive – bere v potaz malá a velká písmena, CZ apod.). Dokončíme instalaci, provedeme restart a doporučuji i instalaci posledního SP pro SQL. Service packy jsou kumulativní, takže není třeba instalovat SP od první verze.

16.4 Problémy z praxe o Virtualizaci SQL služby

Zahlcování enginu pod virtuální mašinou na VmWare.

Obecně se dá říci, že virtualizace ve stavu jaká je pro dnešní den, není vhodné virtualizace použít pro velké databáze. Analýza stavu proč to tak je, je ve většině případů jen na úrovni teoretizování a jelikož to není tématem této práce a ani neznám důsledky chyb, tak se nebudu pouštět do hlubších teorií. Víme ale co se děje a k čemu dochází. Veškeré problémy jsou na I/O vstupně výstupních operacích a na vazbě cpu do virtální OS. Do budoucna se dá očekávat spolupráce produktů Microsoft s Virtualizací VmWare, kde by takovéto chování mělo být už minulostí.

Doporučovaným řešením je použití virtualizačních disků přímo pod službou VmWare, kde by k těmto problémům docházet nemělo, ovšem dochází.

Obecně o virtualizaci jde ale říci, že Xen technologie a HyperV budou v tomto okamžiku na tom ještě hůře než řešení od VmWare.

17 PRAKTICKÁ REALIZACE SQL SERVERU POMOCÍ VIRTUALIZACE - CLUSTERED VERZE

17.1 Vytvoření Virtuálního stroje Cluster Virtual 2 NODY

Vytvoření VS proběhne obdobně jako u Stand Alone Serveru, pouze budeme mít více disků. Ty budou v režimu share a dále budeme mít k dispozici dva nody serveru. Disky budou pracovat pod sdíleným úložištěm dat SQL serveru.

Stand Alone server ve Virtuálu jsme instalovali OS 2003 a SQL 2005, proto nyní provedeme instalaci pod OS 2008 a SQL 2008. Jde nám ovšem jen o definici VS, instalace je v 98% totožná s postupem instalace, jako by stroje byly fyzické servery. Pouze nám vypadnou instalační postupy týkající se podpory úložiště SAN a konfigurace externích polí. To se pod VS dělá přímo v konfiguraci vytvoření nových VS. To si ukážeme právě teď.

A dovolím si říci už nyní, že cluster SQL pod Virtuál serverem není ideální řešení co se týče kombinace vyšších dostupností. Jak správně kombinovat a nastavovat vysoké technologie dostupnosti si řekneme v závěru této práce v samostatné kapitole.

17.2 Konfigurace Virtuálního stroje Cluster Virtual 2 NODY [6]

Pro SQL server v Cluster instalaci použijeme 3 controllery s 3 disky. (disky: OS, Data, Log).

SCSI Controllery budou typu LSI logic SAS s i bez sharingu (Virtual). Bez sharingu tzn. SCSI Bus Sharing nedefinujeme systémové disky (umístění instalace OS), všechny ostatní však budou typ Virtual kvůli clusteringu. Typ SCSI Controlleru tentokrát SAS vzhledem k robustnosti řešení nad clusteringem.

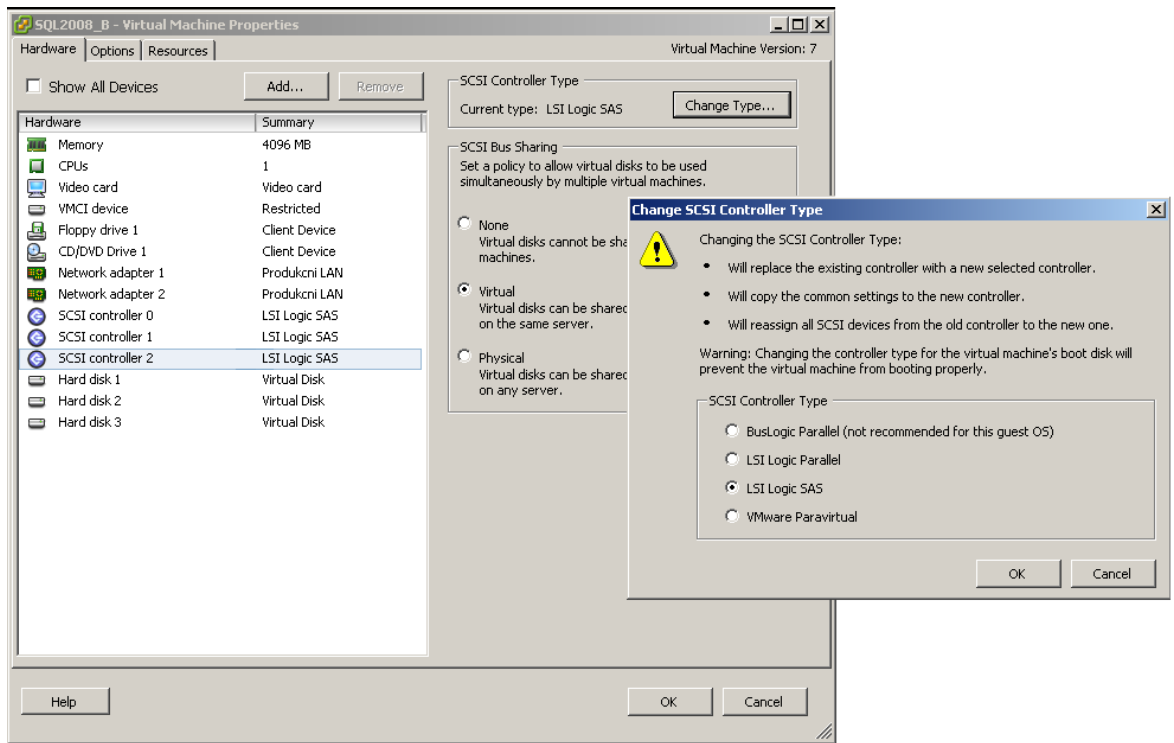
Každý řadič SCSI bude na vlastním kanále. Je to z důvodu rychlosti a samostatnosti diskových oddílů, pokud bychom chtěli některý rozšířit či jakkoliv změnit (každý disk má svůj SCSI řadič).

RAM 4GB, 1 CPU Jader, 2 LAN, 64bit, OS W2008 Ent. Ed.

RAM a CPU pro instalaci stačí v této konfiguraci, dvě LAN rozhraní z důvodu PUBLIC a PRIVATE sítě. Více v kapitole zabývající se clusteringem).

Fyzické úložiště dat na SAN Externím poli (Fibrechanell).

Oba VS budou nadefinovány stejně s tím, že je třeba nastavit práva na sdílené Virtuál disky tak, aby je viděly oba nody clusteru. Nastavení probíhá v rámci VS, pod OS pak nainstalujeme pouze službu clusteru se službou sharing disk v cluster režimu.



Obr. 22: Virtualizace SQL Clustered

17.3 Schéma instalace

Schéma instalace SQL služby ve virtuálu je naprosto totožná s instalací na fyzický server. Vychází to z podstaty věci Virtualizace.

18 PROSTŘEDKY HARDWARU SPOJENÉ S VYŠŠÍ DOSTUPNOSTÍ SLUŽBY MS SQL

18.1 Instalace Operačního systému Serverů

Standardní instalace OS (v našem případě Windows 2005/2008 Enterprise Edition 64b.)

64bit vzhledem k velikosti paměti

Enterprise Ed. z důvodu Cluster Services služeb

Instalace ve výchozím stavu, doporučuji EN verzi s podporou českého prostředí

Instalace ovladačů všech zařízení i externích, aktualizace OS. Instalace ovladačů je nesmírně důležitá, v dalších krocích na toto nebude již místo a mohly by nastat další problémy či zborcení celého OS. Vždy záležitosti týkající se HW doporučuji instalovat jako první, kdy pod OS neběží žádné služby a server nevykonává žádnou úlohu vůči stanicím.

Dále doporučuji firmware všech zařízení tzn. polí, disků, biosu, diskových switchů apod. Platí zde to, co v minulém bodě. Proč řešit firmware za chodu služby SQL a ne dříve, kdy můžeme HW řádně otestovat a máme prostor pro ladění a konfiguraci.

18.2 Diskový subsystém

Inicializace a rozdělení disků v režimu simple volume (NTFS).

Provedeme standardní inicializaci všech disků, se kterými budeme pracovat. Píši záměrně disků a ne polí, protože diskové oddíly se vytváří na polí tak, aby o nich OS nevěděl. Tzn. rozhodně nevytvářejte disky logicky, ale co diskový prostor, to oddíl na poli. Každý takovýto oddíl může mít jiný RAID, dokonce je to vhodné.

Disky se musí vytvořit jako Simple volume, protože to je podmínkou cluster oddílu tzn. co disk, to master oddíl. Víceméně je jasné, proč samostatné oddíly, ale pro jistotu to napíšu. Každý disk bude mít svůj zdroj, který se bude migrovat v rámci cluster služby.

Na cluster nejméně 4 další disky/oddíly (QUORUM DISK – 1GB, MSDTC – 1GB, DATA, LOG).

Quorum disk je řídicí disk celého clusteru, bez něj nepojede cluster, proto 1GB je dostatečný prostor, který nebude třeba nikdy zvětšovat, či s ním nijak manipulovat.

Manipulace s tímto oddílem je nesmírně citlivá pro celý cluster, jakýkoliv zásah do těchto dat vede ke zhroucení cluster služby. Obsahuje čistě jen řídicí instrukce clusteru a nastavení.

MSDTC je transakční koordinátor. Systémový svazek, velikost a obsah dat je hodně podobný jako u quorum disku. Obsahuje však jiná data, vesměs data systému a ne cluster služby.

Doporučuji i 5. disk a to disk TEMP disk cca 2G na každých 50GB fyzických dat v DB na disku DATA. To znamená, máme-li 50GB dat na disku Data, tak 2GB oddíl pro db temp. Je třeba počítat s tím, že db se nafukují, takže je lépe zvolit dostatečnou rezervu.

Disk LOG – zde si MSSQL ukládá všechny transakční logy, zde je nesmírně důležité tento disk fyzicky oddělit od disku DATA, aby nedocházelo ke zpoždění při zápisu. Jedná se zde o rychlost I/O operací, která nám tímto enormně stoupne. Disk LOG by měl být ve formátu RAIDU 1 nebo 10. Kapacita tohoto disku při full recovery modelu DB cca o 1/2 menší jako oddíl data. Při simple recovery modelu (nedoporučuji) 1/4 oddílu data. Největší chyba začínajícího administrátora, který se nesetkal s databázemi je ta, že LOGy chápe jako nějaké odkladiště událostí, které není vůbec důležité. Opak je pravdou, je to nejchoulostivější oddíl co se týče rychlosti a dalo by se říci i dat, protože pokud nám sql služba (server) zkolabuje, nemáme data uložena fyzicky, ale po dalším spuštění sql serveru dojde k dokončení zápisu všech transakcí do fyzických dat na disku DATA. Pokud bychom ony logy neměli, přišli bychom o konzistenci těch databází s kterými se před pádem pracovalo, ve smyslu přepisování a vkládání nových záznamů.

DATA – disk s daty. Jednoznačně disk na RAID 1,5,10. Kapacita dle velikosti DB.

Pokud nebudeme vytvářet samostatný disk pro TEMP SQL, doporučuji využít disk DATA. TEMP DB, či chcete-li soubor, je dočasný prostor pro odkládání operací nad SQL.

18.3 RAID 1,5,10

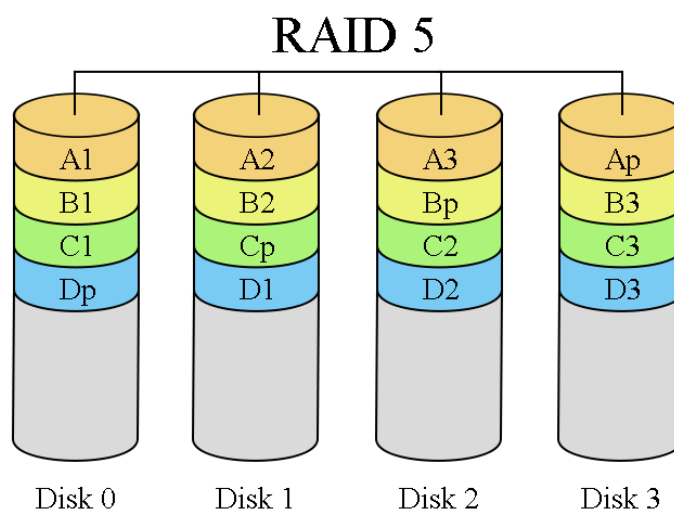
Obecně pro diskový prostor pro SQL platí: Používejte co nejmenší disky s co nejvíce otáčkami. Dosáhnete tím rozložení zátěže na více hlaviček a minimalizujete tím nejužší místo pomalého chodu SQL Serveru. Je potřeba se ale zamyslet nad rozumným řešením, jaké disky do pole koupit. Do diskového pole jde dát pouze omezený počet disků a rozšiřující expanzní moduly jsou fajn, ale vše má své limity. Ve své práci popíší pouze RAID 1,5,10 protože žádný jiný RAID není vhodné používat.

18.3.1 RAID 1

Raid 1 je zrcadlo, data se zapisují na oba disky zároveň. V případě výpadku jednoho z disků pracuje systém s druhým. OS tento výpadek nezaznamená, pouze utility k poli mohou nahlásit kolizi. V zapojení nad jedním řadičem nejde o duplexing metodu. Při duplexing metodě je zapotřebí dvou stejných řadičů (doporučuji použít naprosto stejné se stejným firmwarem), dochází ke zrychlení čtení i zápisu. Rovněž dojde ke zvýšení bezpečnosti při výpadku jednoho s řadičů. V RAID1 přicházíme o 50% diskové kapacity. Tento typ RAIDU je proto nejvýhodnější pro systémový oddíl.

18.3.2 RAID 5

U pole RAID5 hovoříme o minimálně 3(4) discích. RAID5 doporučuji použít na 6 discích a víc, protože se pro ukládání paritních dat používá jiný (rychlejší) algoritmus a obecně v db platí, čím více menších disků, tím rychlejší práce s nimi. Určitě není rozumné koupit dva velké disky s co největší kapacitou a použít je na DATA oddíl jako koncové úložiště našich databází. RAID5 trpí jedním neduhem a to je pomalým zápisem, to nás ale nemusí na tomto oddíle trápit, protože SQL služba si data ukládá data v blocích v porovnání s LOG diskem. RAID5 díky redundantním informacím nad paritním diskem řeší výpadek jednoho z disků. Po takovém výpadku, stejně jako u všech RAIDu, dojde k přepočtu dat, což trvá několik hodin, kdy není vhodné cokoliv dělat s polem, či jej dokonce vypínat, aby nám zhavaroval další disk. Proto nad tímto oddílem je vhodné vytvořit ho hotspare disk, který se zařadí do RAIDu v případě havárie disku. Kapacita disku se dá vyjádřit jako: všechny disky -1 –hotspare (nepovinný).



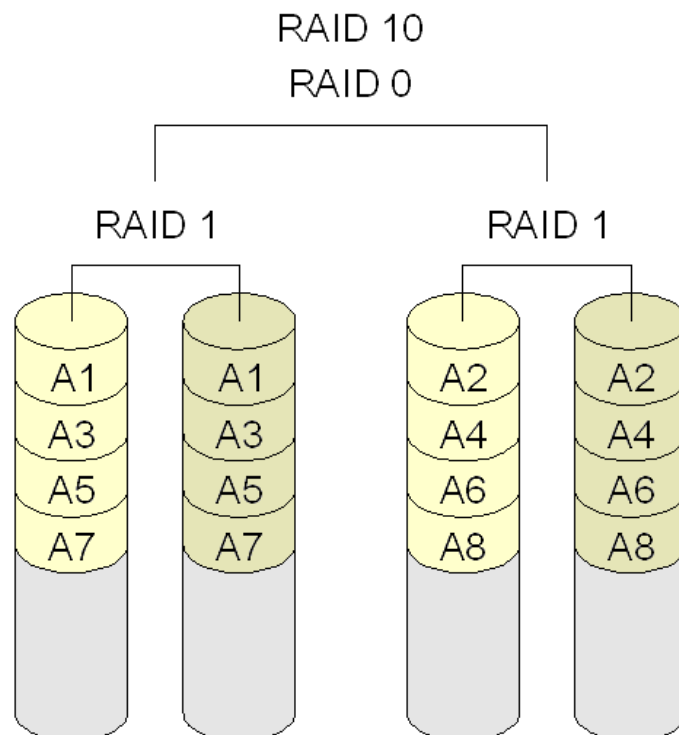
Obr. 23: Prostředky HW RAID5 [wikipedia]

18.3.3 RAID 10

RAID10 je jeden z nadstandardních typů RAIDu, je více než vhodný pro oddíl LOG a TEMP.

Pro oddíl LOG potřebujeme co nejrychlejší zápis a čtení, protože všechny transakce nad SQL se nejdříve uloží do „souboru“ a následně se zapisují po blocích do databáze. Proto oddíl LOG nedoporučuji jako RAID5 (pomalý zápis), ale je třeba použít buď RAID1 nebo RAID10. Od RAID 0+1 se liší tím, že se data nejdříve v diskových polích zrcadlí a teprve pak se tato pole vloží do dalšího diskového pole typu RAID 0 pro větší zrychlení přenosových rychlostí. Maximální počet pevných disků, který může selhat bez jakýchkoliv následků, je jeden v každém poli. Rychlost především, jelikož se nemusí počítat paritní data, čímž se vše zrychluje (případně zlevňuje).

Celková kapacita se pak dá vyjádřit jako celkový počet disků v poli * kapacita jednoho z disku (pokud máme všechny disky stejné)/2.

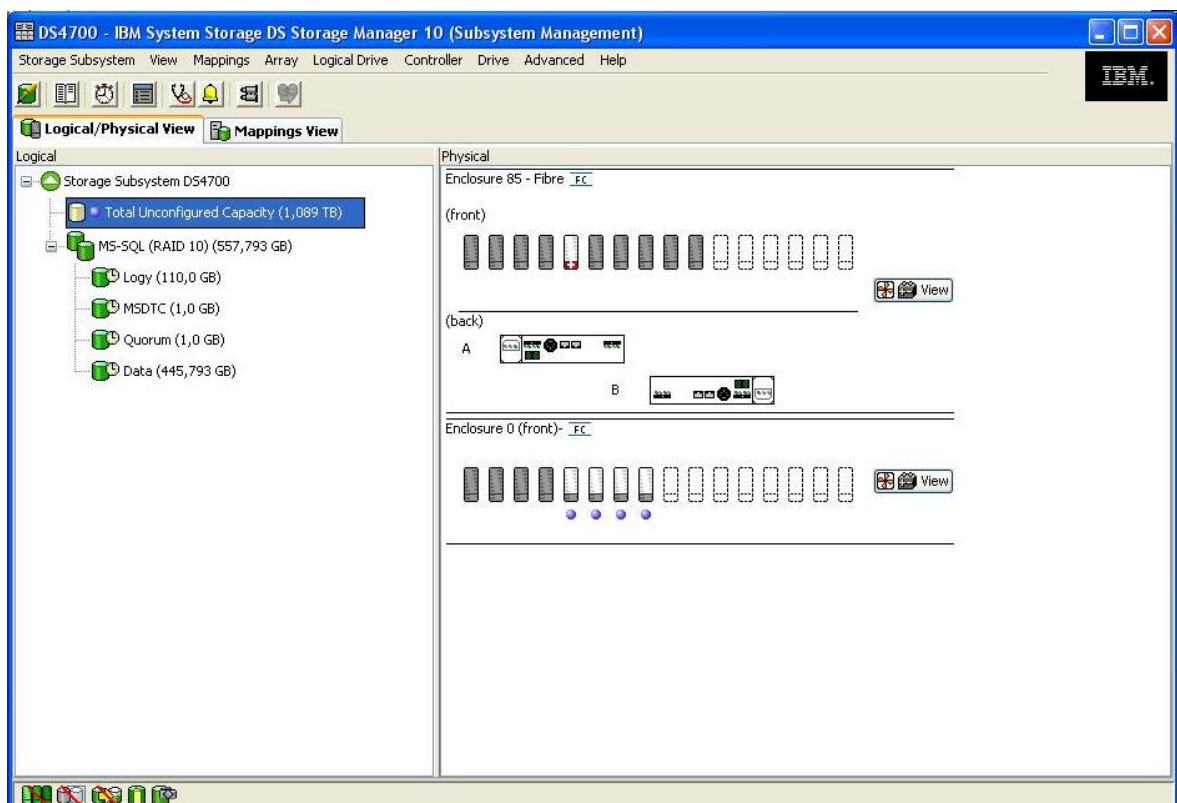


Obr. 24: Prostředky HW RAID10 [wikipedia]

18.4 Konfigurace a Inicializace externích polí pomocí managementu IBM

Konfigurace a Instalace se provádí z administrátorské konzole, kde provedeme instalaci storage manageru určenému ke konfiguraci daného externího pole. Provedeme konfiguraci, tak jak jsme si už řekli, a tím je systém téměř připraven k instalaci cluster služeb.

Na OS je třeba nainstalovat manager cest, který uzpůsobí viditelnost oddílu pro OS. Jelikož použijeme redundanci cest, tak bez těchto pomocných driverů by se s oddíly nedalo pracovat a OS by nepracoval korektně.

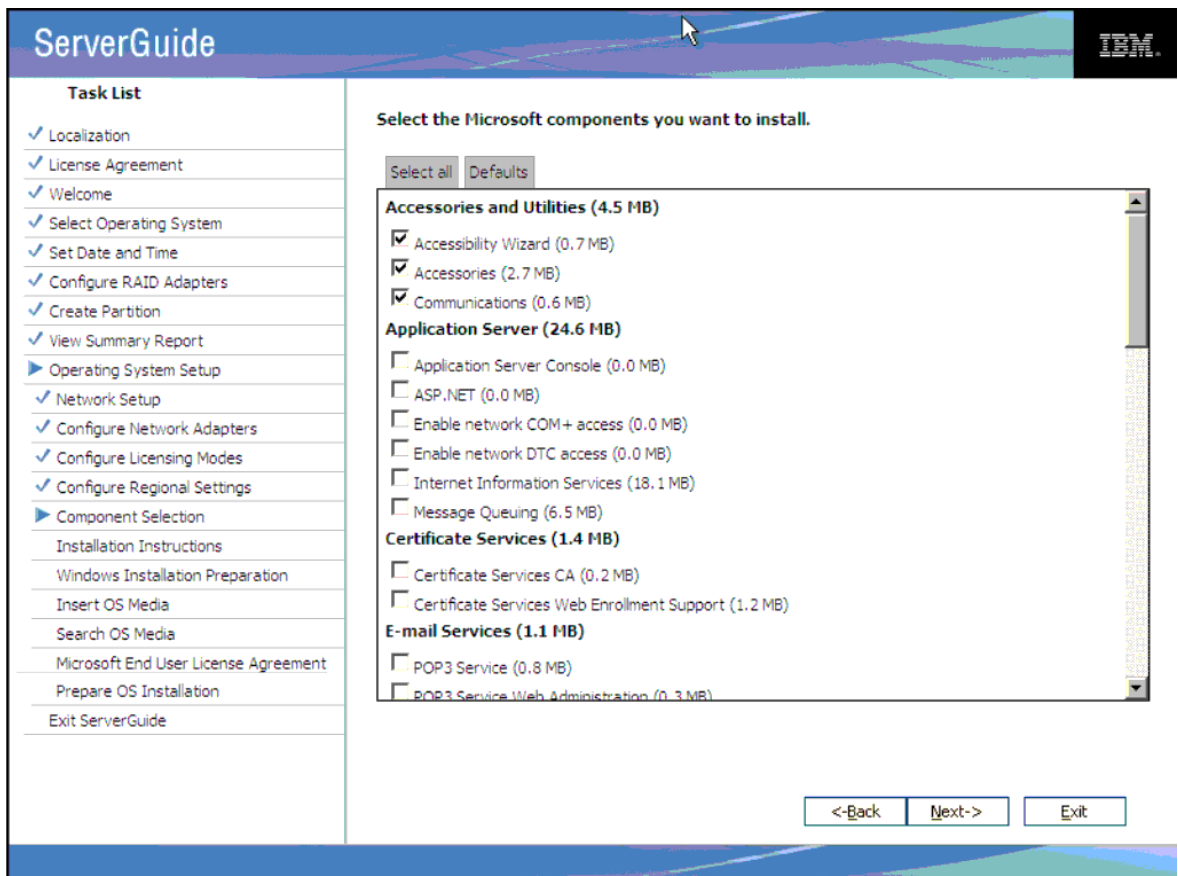


Obr. 25: Inicializace Externího pole

Další informace je možné najít v [4],[5].

18.5 Instalace serveru pomocí průvodce SETUP GUIDE od IBM

Instalace se provádí pomocí boot média, na kterém jsou obsaženy veškeré drivery a software pro všechny části systému pod různými Operačními systémy. My instalujeme OS Windows 2003/2008 Enterprise Edition 64b. Instalace vypadá následovně



Obr. 26: Instalace serveru IBM

Po úspěšném nakonfigurování průvodce, instalace GUIDE nás vyzve ke vložení instalačního média OS a přes vygenerovanou konfiguraci se OS nainstaluje.

19 REALIZACE NAVRŽENÝCH TECHNOLOGIÍ A MĚŘENÍ DOSTUPNOSTI PŘI VÝPADKU

V rámci testování podrobíme všechny technologie vysoké dostupnosti různým testům a popíšeme si, jak se technologie zachovala, případně jak se měla zachovat.

Chování jednotlivých řešení je rozebráno v jednotlivých kapitolách. Zde se ale zaměříme na testy a pokusíme se každé řešení prověřit přímo kolizními či havarijními situacemi.

Každá technologie bude prověřena několika testy a bude měřen celkový výpadek vůči cílovým stanicím či aplikacím.

Testy vychází čistě z práce, a proto není špatné se nad testy zamyslet i jako nad danými scénáři, co se může vlastně stát. Řada testů může pokrýt problémy i nepřímých systémů, které mají zásadní vliv na chod serverů.

Havárie UPS – test totální havárie serveru

Havárie klimatizace – vypnutí serveru standardním ukončením OS přes management serveru

Havárie OS (blue screen bez příčiny HW) - test totální havárie serveru

Havárie RAM modulu – dle serveru, dle typu paměti

Z testů předpokládáme, že databáze jsou v konzistentním stavu, kdy nedojde například díky dedlocku k nedostupnosti db, nebo dojde k rozbití indexu a podobně. Údržba databází nemá nic společného s tématem vysoké dostupnosti, proto zde tyto nástroje a prostředky nebudou popsány.

19.1 Testy

Pro zestručnění, budeme používat test1 až 11 v konkrétním řešení:

Totální havárie serveru (vypnutí serveru z elektřiny) – *test 1*

Vypnutí serveru standardním ukončením OS – *test 2*

Havárie síťové karty LAN v serveru – *test 3*

Havárie síťového kabelu LAN k serveru – *test 4*

Havárie diskového úložiště – *test 5*

Havárie FC switche – *test 6*

Havárie LAN switche – *test 7*

Firmware HW – *test 8*

Havárie FC konektorů nebo optických kabelů – *test 9*

Havárie disku v RAID poli – *test 10*

Havárie SQL nebo některé služby pod OS zajišťující prostředky pro SQL – *test 11*

19.2 Clustering

Test 1 – Dojde k detekci výpadku serveru druhým nodem, neboli výpadku služby zajišťující clustering a server přebírá roli řízení. Všechny clusterové zdroje migrují na druhý nod v určitém pořadí dle závislostí daných rolí. Dostupnost služby do 1 minuty.

Test 2 – Stejné chování serveru jako při testu 1.

Test 3 – zde má server minimálně dvě síťové karty. Pokud dojde k výpadku PRIVATE síťové karty, cluster komunikace automaticky začne probíhat přes PUBLIC síť. Migrace služby se nekoná a server zprostředkovává všechny služby, jako by k žádné události nedošlo.

Pokud dojde k výpadku PUBLIC síťové karty, clusterová služba hlavního serveru započne migraci všech rolí clusteringu. Dostupnost služby do 1 minuty.

Test 4 - Stejné chování serveru jako při testu 3.

Test 5 – V případě výpadku diskového úložiště clustering nedisponuje režimy, které by situaci uměly automaticky vyřešit.

Test 6 - Stejné chování serveru jako při testu 5.

Test 7 – Havarijní clustering provedeme migrací služeb na druhý nod clusteru, kde jsou plně automaticky obnoveny všechny clusterové role společně se službou SQL.

Test 8 – Pokud se rozhodneme firmwarovat jakoukoliv část z HW týkající se služby SQL, je vždy potřeba mít stejný firmware na všech nodech, pokud se jedná o stejné servery. Rovněž zvažme, zda provést odstávku celého clusteru a řádně firmware nasadit a otestovat, nebo zda bude firmware prováděn na pasivním nodu. Při firmwaru na pasivním nodu je třeba si ověřit, zda firmware nezasahuje do aktivních společných částí HW. Pokud ano, rozhodně zastavte chod služby SQL. I zdánlivě jednoduchý firmware serveru, pokud

inicializuje firmware společných části clusteru, může způsobit totální katastrofu s daty a pokud nemáme zálohy, je to vcelku problém.

Test 9 - Stejné chování serveru jako při testu 5.

Test10 – havárie disku v RAIDu řeší diskové pole automaticky s tím, že proběhnou rutiny pro inicializaci záložního disku (hotspare), nebo pokud hotspare nemáme, tak diskové pole pokračuje v RAIDu bez vadné části. Dle zvoleného modelu RAIDu (RAID 1,5,10) můžeme vyhodnotit, jak moc je kritické, pokud nám pole vyřadí jeden disk jako vadný.

Test 11 – Migrace služby SQL na druhý nod. Plně automatická obnova služby do 1 minuty.

19.3 Mirroring, log shipping, Replikace

Jde o podobný přístup technologie vysoké dostupnosti, proto pro zkrácení shrneme tyto technologie do jedné kapitoly.

Předpokládejme následující režimy:

Mirroring s rolí Witness (svědka).

Log Shipping s třetí rolí sledovací databáze.

Replikaci celé databáze tzn. všech tabulek včetně všech záznamů (režim replikace na transakcích nebo snímcích).

Test 1 – Při výpadku serveru přebírá řídicí roli druhý server.

Při mirroringu se automaticky stává druhý server hlavním serverem a nedochází k žádné ztrátě dat.

Dostupnost služby do stanice je okamžitá. Clientská část musí podporovat změnu connect stringu. Napojení stanice proběhne znovu spuštěním aplikace.

Při Log Shippingu nedojde k předání posledních záznamů, které se nestihly předat přes zálohu transakčního logu. Je třeba tyto logy předat a nedojde ke ztrátě dat. Je ovšem nutný zásah administrátora.

Je třeba provést zálohování transakčního logu, ten přenést na druhý server a spustit obnovu transakcí.

Při Replikaci je situace obdobná jako u Log Shippingu, kdy je třeba zásah administrátora za účelem získání posledních záznamů z transakčního logu.

Test 2 - Stejné chování serveru jako při testu 1. Důvodem je časový plán SQL Agentu a jeho pravidelné provádění úloh nezávislé na okolních událostech.

Test 3 - Stejné chování serveru jako při testu 1.

Test 4 - Stejné chování serveru jako při testu 1. Pokud nepoužijeme roli svědka, nebo roli sledovací databáze, tak nedojde k žádnému předání, protože SQL server tuhle událost nesleduje a ignoruje ji.

Test 5 - Stejné chování serveru jako při testu 1.

Test 6 - Stejné chování serveru jako při testu 1.

Test 7 - Stejné chování serveru jako při testu 4.

Test 8 – Zde doporučuji převést hlavní roli na druhý server ručně. V případě replikace provést nasměrování klientů na replikovaný server. Po firmwaru HW otestujte dostupnost daného zařízení pro SQL Server a pokud je vše v pořádku, server můžeme začlenit opět do provozu.

Test 9 - Stejné chování serveru jako při testu 1.

Test 10 - Záležitost RAID pole, vše v režimu zvoleného RAIDu a managementu pole.

Test 11 - Stejné chování serveru jako při testu 1.

19.4 Virtualizace (s Vi center technologií)

Vi Center je technologie Virtualizace, kdy nad dvěma či více Servery pro virtualizaci běží řídicí server, který rozděluje a řídí veškerou komunikaci mezi virtualizačními servery.

Test 1 – V případě havárie serveru dojde k restartu virtuální mašiny na serveru druhém. Tzn. druhý server virtualizace převezme operační systém, nastartuje ho, a pokud máme službu SQL serveru nastavenou aby naběhla automaticky po zapnutí OS, tak je služba SQL obnovena cca do 3 minut.

Pokud nepoužijeme Vi Center – nedojde k automatickému předání VM a OS s SQL službou zůstane nedostupná.

Test 2 – Nejedná se o havárii serveru jako takovou, služba nemigruje na druhý virtuální server. Po naběhnutí OS dojde ke spuštění SQL služby. Služba SQL je dostupná do 3 minut. Pokud nepoužijeme Vi Center – nedojde k automatickému předání VM s SQL službou.

Test 3 – Při havárii síťové karty dojde k přepnutí vizualizačního serveru do Isolate stavu. Server ukončí legálně všechny virtuální mašiny a úlohu převezme druhý vizualizační server, který startuje všechny VM. Pokud nepoužijeme Vi Center – nedojde k automatickému předání VM s SQL službou.

Test 4 – Stejné chování serveru jako při testu 3.

Test 5 – Pokud dojde k havárii diskového úložiště kde jsou uloženy SQL databáze, nebo dokonce samotný virtuální OS, dojde k havárii, kterou virtualizace neumí řešit. Tento stav se týká jak při použití Vi Centra, tak i bez něj.

Test 6 – Stejné chování serveru jako při testu 5.

Test 7 - Stejné chování serveru jako při testu 4.

Test 8 – V případě firmwaru HW pod virtualizací nedojde k žádné změně z důvodu virtualizační vrstvy, která odděluje HW od SW. Virtuální OS nepozná, že se nějaká část HW změnila, a pokud se firmware povedl, dostupnost služby po spuštění serveru bude znovu obnovena.

Test 9 - Stejné chování serveru jako při testu 5.

Test 10 – Havárie disku je oddělena od služeb serveru, proto zde opět funguje role RAID technologie.

Test 11 – Zde je zásadní nedostatek virtualizace. Virtualizační vrstva není svázána se službou SQL Serveru a proto, pokud dojde k havárii služby, nebo nějaké chyby služby, virtualizace tento stav neřeší. SQL služba je v takovém případě nedostupná a do zásahu administrátora nedojde k jejímu opětovnému zprovoznění.

Replikace db – slučováním	N	N	N	N	N	N	N	N	A	N
Virtualizace	A	A	A/N	A/N	N	N	N	A/N	A	N

Tab. 4: Přehled vysoké dostupnosti z pohledu SW

A/N je kombinace, že za určitých okolností Ano – použití svědka, použití Vi centra a podobně. Obecně řečeno použití třetí role, která řídí komunikaci daného SW řešení.

ZA – Nutný zásah Administrátora.

20.1.2 Metody zajišťující vysokou dostupnost z pohledu HW

Test číslo:	1	2	3	4	5	6	7	8	9	10	11
Redundance Serverů	A	A	A	A			A				
Redundance Úložiště (polí)					A						
Redundance Switchů						A			A		
Redundance cest						A			A		
Redundance LAN			A	A							

Tab. 5: Přehled vysoké dostupnosti z pohledu HW

Podrobný rozpis a návrhy realizací jsem uvedl do kapitoly VÝSLEDEK A NÁVRH ŘEŠENÍ. Tabulky softwarové redundance a hardwarové redundance přehledně znázorňují, které technologie co řeší.

21 VÝSLEDEK A NÁVRH ŘEŠENÍ

Z Tabulky dostupnosti - Porovnání všech uvedených metod plyne vcelku jasně, že žádné řešení vysoké dostupnosti nepokrývá všechny možné události, které mohou zapříčinit nedostupnost služby MS SQL. Proto pokud chceme takového stavu docílit, musíme použít redundanci, případně použít kombinaci více metod, zajišťující vysokou dostupnost.

Na zřetel musíme brát i fakt, jak často může daný problém způsobit stav, kdy je třeba redundance. Mnohem menší pravděpodobnost je havárie na FC switchi, než třeba na chybě OS a služby SQL. Pravděpodobnost výpadku a možné administrátorské zásahy do systému jsme probrali v kapitole 7 o redundanci.

Jak tedy dosáhnout pokrytí co nejvíce možných problémů tak, aby služba SQL zůstala pro koncové stanice v provozu? Je vhodné použít správnou kombinaci jak redundance, tak použití více technologií založených na SW. Jako u všech analýz můžeme říci i zde, že není jednoznačný vítěz a jednoznačná cesta, jak takovou dostupnost zajistit. My si proto popíšeme nejvhodnější kombinace s ohledem jak na pořizovací náklady, tak i z hlediska dis-alokace a možnosti postupného rozšiřování. Důležitým faktorem je i jaká data, respektive, kolik dat či databází je třeba zajistit vzhledem k vysoké dostupnosti. Nemá smysl stavět robustní řešení vždy pro každou aplikaci.

Dobré je si nejdříve vyčíslit, kdy musí být služba dostupná vždy, kdy to vadí méně, kam data budeme distribuovat a jaká data a kdo s nimi bude pracovat. Případně se zamyslet nad tím, co se stane, pokud je služba nedostupná, nebo jestli nám jde především o dostupnost dat pro potřeby rychlé obnovy a zajistit tak v každém okamžiku konzistenci tabulek. Leckdy stačí mít nejaktuálnější data na nějakém místě v pořádku v případě havárie, ale není nutné zajistit okamžitě provoz databázových aplikací. I takto jde chápat redundanci a zajištění služby SQL.

21.1 Kombinace redundance

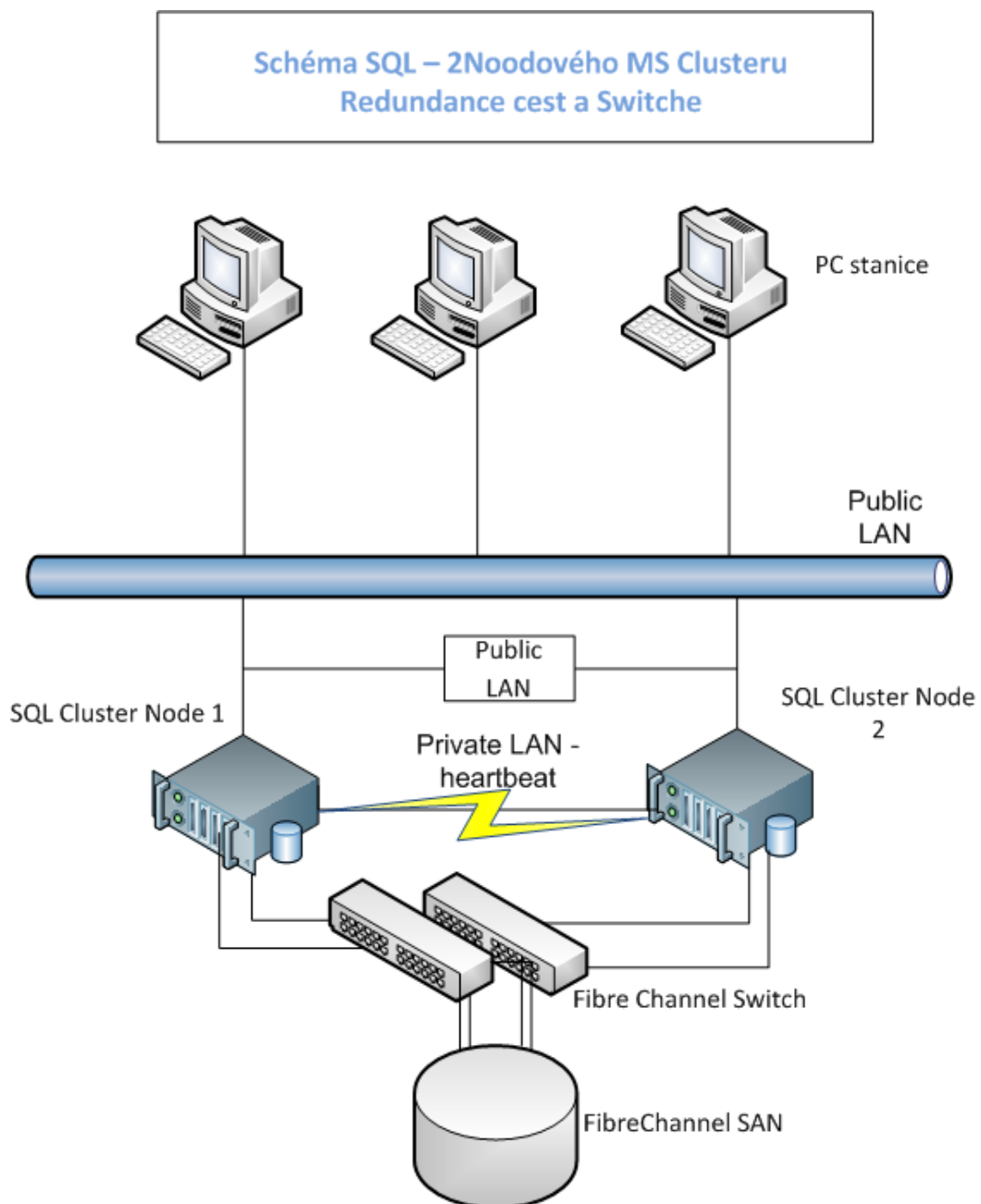
21.1.1 Clustering v kombinaci zdvojení FC switchu a cest

Asi nejpoužívanější je kombinace redundance serveru – Clustering v kombinaci zdvojení FC switchu a cest. Redundance pole je možná, ale provádí se až na samém konci řešení, jelikož částka na pořízení dalšího FC externího pole s disky je dost veliká. Redundance

cest server – switch se zapojuje tak, aby každý server viděl oba switche a oba switche měly redundanci vůči externímu poli.

Takovéto řešení pokryje výpadky všech částí systémů a to i LAN karet, jelikož komunikaci vůči klientům ověřuje služba clusteringu. (všechny testy při tomto zapojení jsou pak vyhodnoceny jako A).

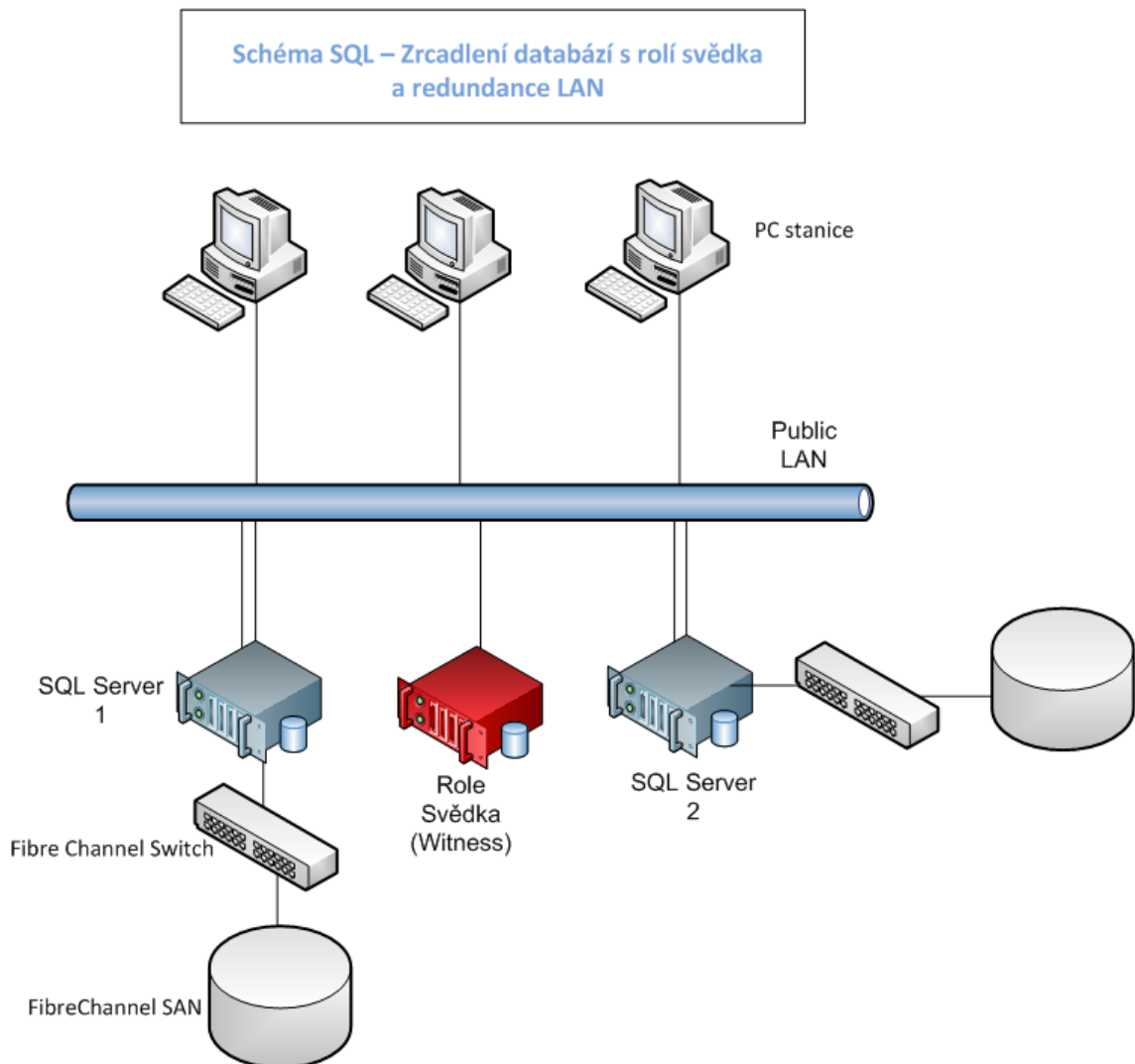
Vhodné použití, pokud máme jednu centrální serverovnu a nechceme budovat další, kde je třeba řešit klimatizace, zabezpečení přístupu, výpadky elektřiny (UPS) a podobně.



Obr. 27: Kombinace redundance - cesty, switche

21.1.2 Použití zrcadlení databází v kombinaci se svědkem a redundancí LAN

Tato metoda se nabízí k použití, pokud máme více serveroven a kdy hrozí zvýšené riziko totálního kolapsu celé serverovny (požár, povodeň a podobně). V porovnání s clusteringem je zapojení naprosto odlišné, ale použití počtu serverů a switchů je velice podobné.



Obr. 28: Kombinace redundance - zrcadlení databází

U tohoto řešení můžeme využít interní řadič disku a data mít interně uvnitř serveru. Tím se vyhneme nákupu externích FC switchů a polí. Jednoznačně to ušetří náklady, ale toto řešení pokulhává na robustnosti, možnosti rozšíření diskových kapacit a i interní řadiče jsou mnohem hůře upravovatelné a nedosahují takových kvalit, jako jsou externí úložiště. Velká část interních řadičů diskových polí neumožňuje kombinaci více RAID polí a pro naše potřeby SQL služby jsou tedy podstatnou nevýhodou.

Celé řešení opět pokrývá všechny možné kombinace výpadků (všechny testy při tomto zapojení jsou pak vyhodnoceny jako A) a to jak při použití interního či externího úložiště.

Při použití interních polí přímo v serveru dosáhneme nejlevnější možné varianty vzhledem k nákupu HW technologií. V případě použití externích polí je to varianta dražší než clustering.

A nejdůležitější fakt na závěr. Při použití tohoto řešení je potřeba řešit licence pro 2 SQL servery. Při licencování per CPU je to zdvojování nákladů oproti pasivnímu clusteringu.

21.1.3 Třetí variantou je Virtualizovat jeden z nodu clusteru

Nejčastějším důvodem je využití pasivního nodu clusteru pro výpočetní výkon jiných aplikačních služeb.

Schéma zapojení je totožné se schématem Clusteringu v kombinaci s redundancí switchu a cest. Tahle varianta může být v jistých situacích velice lehce implementována do stávajícího řešení Virtualizace. Pokud tedy virtualizujeme a máme vybudovanou část infrastruktury, stačí nám pouhé dokoupení serveru a FC karet.

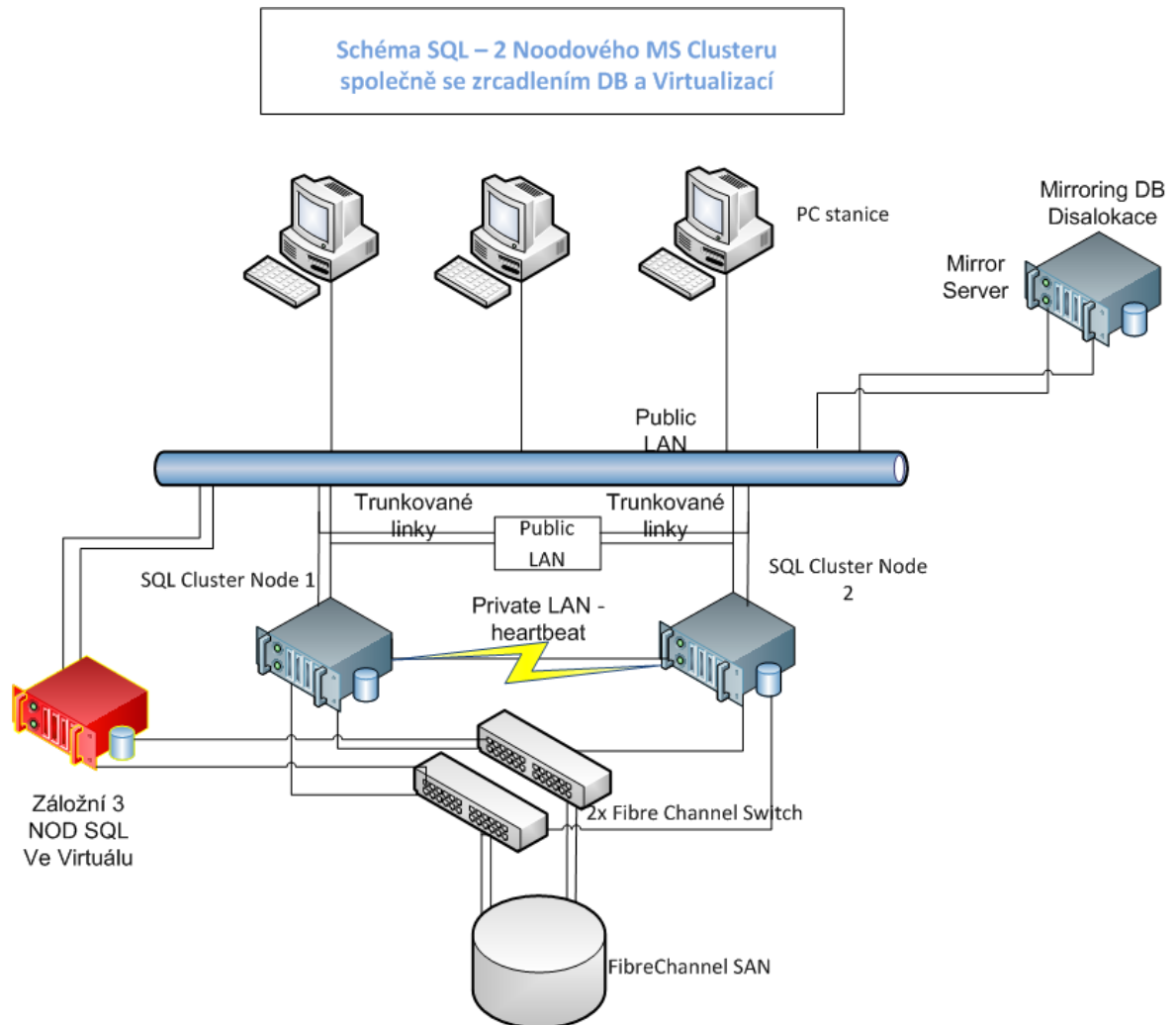
Pokud stavíme řešení redundance na zelené louce, je na zváženu, kolik máme aplikačních serverů a zda i u nich chceme řešit vyšší dostupnost pomocí virtualizace. Pokud ano, jednoznačně se vyplatí spojit virtualizaci s clusteringem pod externím úložištěm. Pokud virtualizovat aplikační servery nechceme, doporučuji použít pasivní clustering.

21.1.4 Čtvrtou variantou je clustering se zrcadlením databází, redundancí cest a virtualizací

Zajímavou kombinací je využití řešení clusteringu se zrcadlením db, redundancí cest a virtualizace:

Toto řešení eliminuje stav totálního výpadku clusteru, případně diskového subsystému. Dobrou otázkou je, proč tak přehánět redundanci, ale odpověď bude velice jednoduchá. V případě použití clusteringu se jednou za čas dostaneme do situace, kdy se nám bude hodit zmigrovat všechny služby jinam, například kvůli údržbě, nebo reinstalaci serverů. Pokud používáme v prostředí firmy virtualizaci, nabízí se mít nachystánu virtuální mašinu pro tuto službu. Virtuální mašina běžet nemusí, tím pádem nebude alokovat prostředky serveru. Virtualizaci takového serveru je vhodné provést čistě se službou zajišťující SQL Server bez dat a s tím spojenou i konfiguraci. V případě provozu na takovémto serveru se

pouze přes management FC switche namapují diskové oddíly z clusteru na 3 Virtualizační nod a službu SQL Serveru můžeme spustit. Všechny konfigurace jsou uloženy v master databázi, je to léta odzkoušená metoda, jak rychle zprovoznit SQL službu.



Obr. 29: Kombinace redundance - kombinace VD

Další článkem redundance je použití zrcadlení databází, případně jen konkrétní databáze. Opět i zde má tato metoda uplatnění zejména při dis-alokaci budov, nebo pokud potřebujeme zajistit dostupnost služby SQL v situacích, kdy je třeba zajistit 24hodinový provoz 7 dní v týdnu. V takovém provozu musíme počítat s totálním výpadkem například elektřiny i v době, kdy není žádný administrátor v dosahu a kdy je třeba řešit dostupnost databází. Mirroring databází nebo i klonování databází je vhodný prostředek k pokrytí služeb při totální havárii hlavní části infrastruktury.

22 ZÁVĚR

Hlavním cílem této bakalářské práce bylo seznámit čtenáře se všemi možnostmi zajištění vysoké dostupnosti služby Microsoft SQL Serveru. Práce obsahuje kapitoly týkající se jak softwarového, tak i hardwarového řešení. Softwarové řešení popisují technologie jako je clustering, zrcadlení databází a předávání transakčních logů. Do hardwarového řešení patří virtualizace a redundance. Práce popisuje každou zmíněnou technologii po stránce teoretické, i po stránce praktické. Po stránce teoretické jsou popsány možnosti dané technologie, po stránce praktické je to pak instalace a konfigurace. V závěru práce se dozvíme, kdy a jak je vhodné použít dané řešení vysoké dostupnosti a také jejich vzájemnou kombinaci. Z práce jednoznačně nevyplývá použití jedné metody, každá metoda má své výhody, i nevýhody. Práce obsahuje i testy každého řešení a podrobné srovnání všech zmíněných nástrojů k zajištění dostupnosti služby Microsoft SQL Serveru.

V posledních dvou kapitolách přesně popisují, kdy je vhodné použít konkrétní technologii dostupnosti.

Práce detailně popisuje nejčastější řešení a to pomocí použití metody clusteringu a redundance cest směrem k úložišti dat. Clustering v kombinaci s HW redundancí vyšla nejlépe a to jak z hlediska realizace, tak i z hlediska pořizovacích nákladů při zachování jednoho centrálního místa serverové infrastruktury.

Pokud shrneme do jedné věty cíl bakalářské práce, tak je to studie k zajištění vysoké dostupnosti za účelem chodu služby Microsoft SQL Server a databází.

ZÁVĚR V ANGLIČTINĚ

The main objective of this thesis was to acquaint the reader with all the options to ensure high availability of Microsoft SQL Server. The work contains chapters concerning the software and hardware solutions. Describes the technology software solutions such as clustering, database mirroring and transmission of transaction logs. Hardware virtualization includes redundancy solutions. This work describes each of the technology from a theoretical and a practical point of view. Technology possibilities are described in theoretical point, in terms of practical point, installation and configuration is described. In the end we will find out when and how it is appropriate to use the high availability solutions and also their combination. The work clearly does not imply the use of one specific method, each method has its advantages and disadvantages. Work includes testing of each solution, and a detailed comparison of all these instruments to ensure the availability of Microsoft SQL Server.

In the last two chapters it exactly describes where it is appropriate to use a particular technology availability.

Work describes in detail the most common solution and a method using clustering and redundancy of pathways toward the data store. Clustering in combination with hardware redundancy came out best, both in terms of implementation, as well as for cost, while maintaining a centralized server infrastructure.

To summarize it in one sentence, the objective of the thesis is a study of high availability in order to keep Microsoft SQL Server and databases available.

SEZNAM POUŽITÉ LITERATURY

22.1 Kapitoly v knize

- [1]. OSIF, Michal. *Windows Server 2003*. 1. vyd. Praha : Grada Publishing a.s., 2003. 612 s. ISBN 80-247-0396-3. Kapitola 1 - Active Directory 2003, , s. 10–221.
- [2]. RUSSEL, Charlie, CRAWFORD, Sharon, GEREND, Jason. *Windows Server 2003 : Velký průvodce administrátora*. Anna Rychetská. 1. vyd. Brno : CPBooks a.s., 2005. 1374 s. ISBN 80-251-0579-2. Kapitola 18 - Použití clusterů 2003, s. 545–577.
- [3]. HOTEK, Mike. *Microsoft SQL Server 2008 : Krok za krokem*. Praha : Computer Press, 2009. 488 s. ISBN 978-80-251-2466-6. Kapitola - Vysoká dostupnost.

22.2 Elektronické zdroje (WWW)

- [4]. *Configuring ServeRAID controllers for clustering. IBM Systems Software Information Center* [online]. 2003 [cit. 2010-01-28]. Dostupný z WWW: http://publib.boulder.ibm.com/infocenter/eserver/v1r2/index.jsp?topic=/diricinfo/fqy0_acluscfg.html .
- [5]. *IBM Totalstorage FastT Storage manager Version 9 Installation and Support Guide for Intel-based Operating System Environments : Installing Storage manager software in a standard Windows configuration* [online]. 2003 [cit. 2010-01-28]. Dostupný z WWW: ftp://ftp.software.ibm.com/systems/support/system_x.../gc26-7648-00.pdf .
- [6]. *Configuring Microsoft Cluster Service in a VMwarevSphere 4 Environment Using Infortrend Storage* [online]. 2009 [cit. 2010-01-28]. Dostupný z WWW: http://www.infortrend.com/doc/appNote/APP_Note_MSCS%20in%20vSphere4.pdf .
- [7]. *Windows Server 2008 Instalace Active Directory* [online]. 2008 [cit. 2010-01-28]. Dostupný z WWW: <http://www.mstv.cz/it/videos/190/WS2008-serial---Instalace-Active-Directory> .

- [8]. *Guide to Creating and Configuring a Server Cluster under Windows Server 2003* [online]. 2003 [cit. 2010-01-28]. Dostupný z WWW: <http://www.microsoft.com/downloads/en/confirmation.aspx?familyId=a5bbb021-0760-48f3-a53b-0351fc3337a1&displayLang=en> .
- [9]. *Windows Server 2008 R2 - Instalace a konfigurace Windows Failover Cluster* [online]. 2008 [cit. 2010-01-28]. Dostupný z WWW: <http://www.mstv.cz/it/videos/210/WS2008-R2---Failover-Cluster--konfigurace-a-vytvoreni-> .
- [10]. *Windows Server 2008 R2 - Podrobnejsi konfigurace Windows Failover Cluster* [online]. 2008 [cit. 2010-01-28]. Dostupný z WWW: <http://www.mstv.cz/it/videos/219/WS2008-R2---Failover-Cluster--pokrocila-konfigurace-> .
- [11]. *Latest SQL Server Clustering Articles* [online]. 2007 [cit. 2010-01-29]. Dostupný z WWW: <http://www.sql-server-performance.com/articles/clustering/main.aspx> .
- [12]. *Installing SQL Server 2008 on a Windows Server 2008 Cluster Part 1,2,3,4* [online]. Edwin Sarmiento, 2009 [cit. 2010-01-29]. EN. Dostupný z WWW: <http://www.mssqltips.com/tip.asp?tip=1687,1698,1709,1721> .

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

AD	Active Directory
FC	Fibre Chanell
NOD	Instance Serveru
DB	Databáze
MSCS	Microsoft Cluster Services
HCL	Hardware kompatibility List
GB	Gigabyte
FCM	Failover Cluster Manager
DTC	Data Transaction Controller
SP	Service pack
VS	Virtual Server
VM	Virtual Machine

Standardizované zkratky: HDD, FDD, CPU, SQL, RAM, LAN, SAN, NOD, SCSI, NS, OS, NETBIOS, RAID, MBR, NTFS, MB, GB, TCP/IP, DHCP, SID, LAN, LSI, NIC, SAS, IIS, I/O, HW, SW, UPS

SEZNAM OBRÁZKŮ

Obr. 1: Princip komunikace mirroringu.....	18
Obr. 2: redundance dvou serverů, cest a switchu.....	31
Obr. 3: Redundance úložiště.....	32
Obr. 4: Schéma clusteru základ.....	39
Obr. 5: Instalace Clusteru.....	45
Obr. 6: Failover Cluster Manager.....	49
Obr. 7: Failover Cluster manager 2.....	50
Obr. 8: Přidání File Serveru do FCM Failover Cluster manager.....	51
Obr. 9: Instalace SQL 2005 Cluster Mod.....	57
Obr. 10: Cluster Administrator Console.....	60
Obr. 11: Instalace MSDTC SQL 2008.....	66
Obr. 12: Instalace SQL 2008 v Cluster modu.....	68
Obr. 13: Instalace SQL 2008 v Cluster modu 2.....	69
Obr. 14: Instalace SQL 2008 v Cluster modu 3.....	72
Obr. 15: Instalace mirroringu MS SQL.....	74
Obr. 16: Instalace mirroringu MS SQL.....	75
Obr. 17: Instalace Log Shipping.....	78
Obr. 18: Instalace Log Shipping 2.....	81
Obr. 19: Instalace Replikace.....	81
Obr. 20: Virtualizace SQL Stand-Alone.....	86
Obr. 21: Virtualizace SQL Stand-Alone 2.....	87
Obr. 22: Virtualizace SQL Clustered.....	90
Obr. 23: Prostředky HW RAID5.....	93
Obr. 24: Prostředky HW RAID10.....	94
Obr. 25: Inicializace Externího pole.....	95
Obr. 26: Instalace serveru IBM.....	96
Obr. 27: Kombinace redundance - cesty, switche.....	105
Obr. 28: Kombinace redundance - zrcadlení databází.....	106
Obr. 29: Kombinace redundance - kombinace VD.....	108

SEZNAM TABULEK

Tab. 1: Pravděpodobnost výpadku.....	28
Tab. 2: Pravděpodobnost údržby.....	28
Tab. 3: Postup instalace clusteru na OS 2003.....	36
Tab. 4: Přehled vysoké dostupnosti z pohledu SW.....	102
Tab. 5: Přehled vysoké dostupnosti z pohledu HW.....	103

SEZNAM PŘÍLOH

BP neobsahuje žádné přílohy

PŘÍLOHA P I: NÁZEV PŘÍLOHY