VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

BEZPEČNOST DATABÁZÍ MSSQL

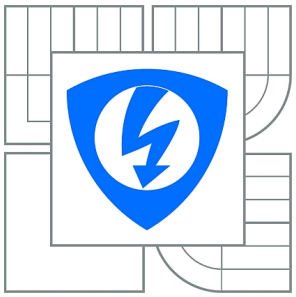
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE PAVEL PYSZKO
AUTHOR

BRNO 2012

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF TELECOMMUNICATIONS

BEZPEČNOST DATABÁZÍ MSSQL

SECURITY OF MSSQL DATABASES

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

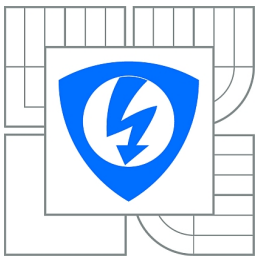
AUTOR PRÁCE — PAVEL PYSZKO

AUTHOR

VEDOUCÍ PRÁCE — prof. Ing. KAMIL VRBA, CSc.

SUPERVISOR

BRNO 2012



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Pavel Pyszko ***ID:*** 119586
Ročník: 3 ***Akademický rok:*** 2011/2012

NÁZEV TÉMATU:

Bezpečnost databází MSSQL

POKYNY PRO VYPRACOVÁNÍ:

Věnujte se problematice bezpečnosti dat v rámci databází MSSQL 2005, 2008 a 2008 R2, kryptografickým algoritmům a šifrování. Navrhněte optimální způsob ukládání dat do databází MSSQL co do nejlepšího poměru zabezpečení/výkon. Popište nejčastější bezpečnostní rizika týkající se přístupu do MSSQL.
Nejdříve by měl být proveden rozbor v současnosti používaných metod a řešení zabezpečení dat v rámci MSSQL, rozbor bezpečnostních rizik a uvedení kladných a záporných vlastností jednotlivých řešení. Poté se předpokládá navržení a realizace nejoptimálnější varianty z hlediska zabezpečení/výkon.

DOPORUČENÁ LITERATURA:

[1] WALTERS, E. Robert. Mistrovství v Microsoft SQL server 2008. 1. vyd. Brno: Computer Press, 2009. 864 s. ISBN 978-80-251-2329-4.
[2] VELTE, T. Antony, VELTE, J. Toby, EISENPETER, Robert. CLOUD COMPUTING – Praktický průvodce. 1. vyd. Brno: Computer Press, 2011. 304 s. ISBN 978-80-2513-333-0.
[3] HANÁK, Ján. C++/CLI – Začínáme programovat. 1. vyd. Brno: Artax, 2009. 371 s. ISBN 978-80-87017-04-3.

Termín zadání: 6.2.2012 ***Termín odevzdání:*** 31.5.2012

Vedoucí práce: prof. Ing. Kamil Vrba, CSc.
Konzultanti bakalářské práce: Ing. Radek Pospíšil

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ANOTACE

Cílem bakalářské práce je rozebrat problematiku bezpečnosti dat v rámci databází MSSQL 2005, 2008, 2008 R2 a 2012, kryptografickým algoritmům, šifrováním, zabezpečením databáze, sledováním výkonu databáze a nástrojům pro správu a analýzu dat. V práci jsou rozebrány bezpečnostní rizika, možnosti řešení jednotlivých rizik a optimální návrh varianty z hlediska zabezpečení/výkon.

Klíčová slova: databáze, zabezpečení, šifrování, výkon a Microsoft SQL

ABSTRACT

The aim of bachelor thesis is to discuss issues of data security within the databases MSSQL 2005, 2008, 2008 and 2012 R2, cryptographic algorithms, encryption, database security database performance monitoring and management tools and data analysis.The thesis contains analyzed the security risks, possible solutions for each risk and the optimal design options in terms of security / performance.

Keywords: database, security, encryption, performance and Microsoft SQL

Bibliografická citace práce

PYSZKO, P. *Bezpečnost databází MSSQL .* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2012. 48 s. Vedoucí bakalářské práce prof. Ing. Kamil Vrba, CSc..

Prohlášení

Prohlašuji, že svou bakalářskou práci na téma Bezpečnost databází MSSQL jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne ……. .....

podpis autora

Poděkování

Děkuji konzultantovi práce Ing. Radku Pospíšilovi za velmi užitečnou metodickou pomoc a cenné rady při zpracování bakalářské práce.

V Brně dne ……. .....

podpis autora

Obsah

Seznam tabulek

Seznam obrázků

Seznam grafů

Úvod

SQL (Structured Query Language) je standardizovaný dotazovací jazyk určen pro práci s daty relačních databází. Využití databázových systémů a potažmo jazyka SQL je v současnosti velice rozsáhlé a v oblasti IT hojně využívané. Od doby první standardizace (rok 1986, ANSI SQL-86) se možnosti značně rozšířily a v současnosti se tak jedná o velice výkonný a rozsáhlý prostředek.

Práce se zaměřuje na implementaci jazyka v prostředí Microsoft SQL Server, výkon databází a jejich bezpečnost, přitom budou zhodnoceny a uvedeny vlastnosti, funkcionality a omezení jednotlivých verzí Microsoft SQL Serveru. Dále budou probíhat testy nad definovanou databází, zejména praktické měření dosažitelného výkonu /bezpečnosti ve vybraných verzích jazyka SQL, návrh optimalizace předložené databáze a její struktury a následně ověření vhodnosti zvoleného návrhu.

1. Režimy ověřování

Režimy ověřování se vybírají během instalace SQL serveru. Existují dva možné režimy: ověřování pomocí systému Windows a kombinovaný režim. Kombinovaný režim je kombinací Windows ověřování a SQL ověřování. Ověřování systému Windows je tak vždy k dispozici a nemůže být zakázáno.

Pokud při instalaci vybereme kombinované ověřování, je nutné zadat heslo, které se ukládá na SQL server do účtu s názvem **sa**[1]. Je-li režim ověřování systému Windows vybrán během instalace, je zakázáno přihlášení k účtu **sa**. Pokud později změníte režim ověřování na kombinovaný, účet **sa** zůstává zakázaný. Pro povolení přihlášení prostřednictvím účtu **sa** lze použít příkaz transakčního jazyka SQL „LOGIN ALTER“. Účet **sa** je často terčem uživatelů se zlými úmysly. Nedoporučuje se tedy účet **sa** používat, pokud to není nutné. Je velmi důležité použít silné heslo pro přihlášení k účtu **sa**. [2]

1.1 Windows ověřování

- Pokud se uživatel připojuje prostřednictvím uživatelského účtu systému Windows, SQL Server ověřuje název účtu a heslo pomocí Windows hlavního klíče v operačním systému. To znamená, že identitu uživatele potvrzuje Windows.
- SQL Server nevyžaduje heslo a neprovádí ověřování identity
- Ověřování systémem Windows je výchozí režim, a je mnohem bezpečnější než SQL ověřování
- Používá bezpečnostní protokol Kerberos, více viz 1.1.1
- Poskytuje vynucení zásady hesel s ohledem na složitost ověřování pro silná hesla
- Poskytuje podporu pro uzamčení účtu
- Podporuje vypršení platnosti hesla

Obr. 1.1: Windows ověřování

[1] Účet sa – účet, jenž má oprávnění vlastníka databáze pro všechny databáze serveru, nezávisle na použitém ověřovacím režimu.

1.1.1 Bezpečnostní protokol Kerberos

Kerberos je primární protokol zabezpečení pro ověřování identity v rámci domény. Protokol ověřuje identitu uživatele požadujícího ověření i identitu serveru provádějícího požadovaného ověření. Toto duální ověřování je označováno jako vzájemné ověření. Důležitou službou v rámci protokolu Kerberos je služba KDC (Key Distribution Center). Služba KDC je spuštěna v každém řadiči domény jako součást adresářové služby Active Directory, která zajišťuje uložení všech hesel klienta a dalších informací o účtu. [3]
Proces ověřování prostřednictvím protokolu Kerberos probíhá následujícím způsobem: [3]

1. Uživatel v systému klienta pomocí hesla prokáže svou identitu službě KDC
2. Služba KDC vydá klientovi zvláštní lístek TGT (Ticket Granting Ticket). Systém klienta pomocí lístku TGT získá přístup ke službě TGS (Ticket Granting Service), která je součástí mechanismu ověřování pomocí protokolu Kerberos v řadiči domény
3. Služba TGS poté vydá klientovi tzv. lístek služby
4. Klient se prokáže tímto lístkem požadované službě v síti. Lístek služby slouží nejen k prokázání identity uživatele pro službu, ale i k prokázání identity služby pro uživatele.

1.2 SQL ověřování

- Název účtu a heslo jsou vytvořeny pomocí SQL Serveru a také zde uloženy
- Uživatelé připojující se pomocí SQL ověřování musí poskytnout své přihlašovací údaje (login a heslo) pokaždé, když se připojí
- SQL ověřování má k dispozici tři bezpečnostní nastavení hesel:
    - Uživatel musí změnit heslo při příštím přihlášení
    - Vynutit vypršení platnosti hesla
    - Vynutit zásady hesel
- Podporuje prostředí se smíšenými operačními systémy, kde ne všichni uživatelé jsou ověřováni Windows doménou

Obr. 1.2: SQL ověřování

2. Vysoká dostupnost/obnovení po havárii

Co vysoká dostupnost znamená? Znamená to, že servery a služby jsou spuštěné a koncoví uživatelé nepociťují zádrhele v chodu systému a že firma/osoba může fungovat na přijatelné úrovni. Používáme pojem přijatelná úroveň, protože jedna organizace může za přijatelnou úroveň považovat stav, kdy systém neběží hodinu týdně, zatímco jiná organizace může považovat za přijatelnou vteřinu týdne. To závisí na úrovni vysoké dostupnosti, jakou potřebujete.
Problémem jsou výpadky, které se vyskytují v následujících podobách [1]:

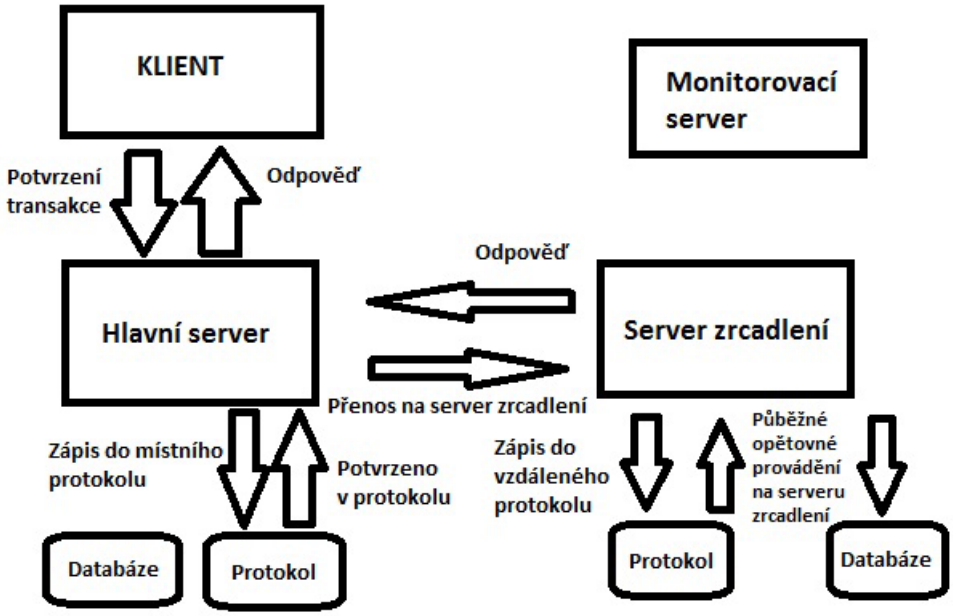
- Neplánované výpadky: toto je to, čemu se většina lidí snaží vyhnout, když chtějí implementovat řešení vysoké dostupnosti. Jedná se o havárie, výpadky proudu, chyby správce atd.
- Plánované výpadky: pravidelná údržba databáze, aktualizace, instalace softwaru atd.

Pojem, který jde ruku v ruce s vysokou dostupností, je pojem obnovení po havárii, což jsou prostředky, pomocí nichž chráníme systém před ztrátou dat a obnovujeme normální operace v případě selhání systému. Obnovení po havárii je součástí strategie vysoké dostupnosti. Dokonce i po havárii musíte být schopni nastartovat a spustit váš systém pro dosažení vysoké dostupnosti. [1]

V podstatě strategie obnovení po havárii je strategie zálohy a obnovy. Dosažení nejvyšší úrovně obnovy po havárii znamená mít jiné pracoviště, umístěné jinde, než je primární pracoviště. V případě, že dojde ke katastrofálním selháním na prvním pracovišti, dojde k přepnutí na druhé pracoviště. Pro dosažení odolnějších serverů můžete kombinovat strategie vysoké dostupnosti a obnovení po havárii jak na primárním pracovišti, tak na sekundárním. Při implementaci řešení vysoké dostupnosti/obnovení po havárii je dostupných mnoho technologií systému SQL Serveru, hlavní z nich jsou popsány níže v textu. [1]

2.1 Zrcadlení databáze

Hlavní myšlenka zrcadlení databáze je taková, že se udržují synchronizované verze databáze na hlavním serveru a na zrcadlovém serveru. Pokud hlavní databáze není dostupná, pak se klientské aplikace přepnou na zrcadlenou databázi a operace (z pohledu uživatelů) budou pokračovat. [1]



Obr. 2.1: Architektura zrcadlení databáze s monitorovacím serverem

Tedy klient je připojen k hlavnímu serveru a odesílá transakci. Hlavní server zapíše požadovanou změnu do hlavního protokolu transakcí a automaticky přenese tuto informaci popisující transakci na zrcadlo, kde se zapíše do protokolu transakcí zrcadleného serveru. Zrcadlo pak pošle potvrzení příjmu hlavnímu serveru. Zrcadlo průběžně používá vzdálený protokol transakcí pro „replikaci“ změn provedených do hlavní databáze na zrcadlový server. [1]

Hlavní výhoda, díky které je zrcadlení výhodnější než odesílání souboru protokolu nebo replikace transakcí, je to, že je eliminována potenciální ztráta dat, je povolené automatické přepnutí v případě havárie, obsahuje transparentní přesměrování klienta a jsou zjednodušeny správa a sledování. [1]

2.2 Clustering

Cluster se skládá z řady volně spojených počítačů, které pracují tak, že v mnoha ohledech mohou na venek vystupovat jako jeden celistvý systém. Komponenty clusteru jsou vzájemně propojeny přes rychlé lokální sítě, kde každý uzel (node) provozuje vlastní instanci operačního systému. Cluster se opírá o centralizované řízení přístupu, který je k dispozici jako řízené uzly sdílených serverů. Cluster může být jednoduchý se dvěma uzly systému (propojení dvou osobních počítačů), nebo mohou být použity u velmi rychlých super počítačů. Clustery můžeme rozlišit na několik typů, zde jsou rozepsány ty hlavní. Ve skutečnosti se však jednotlivé funkce clusterů prolínají, aby bylo dosaženo optimálních parametrů. [1]

Výpočetní cluster

Slouží k zvýšení výpočetního výkonu pomocí více počítačů, které na výpočtu spolupracují. Tímto způsobem vznikne vysoce výkonný celek, který je mnohonásobně levnější, než jeden superpočítač. [1]

Cluster s vysokou dostupností
Zajišťuje pomocí několika počítačů nepřetržité poskytování nějaké služby i při výpadku počítače z důvodu hardwarové závady nebo plánované údržby. Služby poskytuje jeden počítač, který je v případě výpadku automaticky zastoupen jiným počítačem. [1]

Cluster s rozložením zátěže
Snižuje možnou míru zátěže tím, že službu poskytuje několik počítačů, které mají stejný obsah (služba je poskytována paralelně). Stejný obsah je zajištěn replikací obsahu mezi všechny propojené počítače nebo existencí specializovaného centrálního uložiště. [1]

Úložný cluster
Zprostředkovává přístup k diskově kapacitě, která je rozložena mezi více počítačů z důvodu dosažení vyššího výkonu nebo pro zajištění vyšší spolehlivosti. Toho je dosahováno speciálními souborovými systémy, které jsou schopny zajistit rozložení zátěže, redundanci dat, pokrytí výpadku jednotlivých uzlů, distribuovaný mechanismus zamykání souborů a další doprovodné služby. [1]

Clustering lze použít spolu se zrcadlením, avšak musí být použito zrcadlení mezi clustery, nikoli uvnitř clusteru. Zjednodušeně můžete mít hlavní databázi běžící na clusteru složeném ze dvou uzlů, který komunikuje se zrcadlem fungujícím na samostatném clusteru složeném ze dvou uzlů. Jediné věci, na které je třeba dávat pozor při použití zrcadlení a clusteringu, jsou doby přepnutí v případě poruchy a konflikty. Pokud například nastavíme časový limit zrcadlení na jednu minutu a clustery obvykle přepnou po 30 sekundách, mohou se clustery přepnout v případě poruchy dříve než vaše zrcadlo a opačně. Také se můžete dostat do poněkud podivných situací, když cluster a zrcadlo přepnou v případě poruchy ve zhruba stejnou dobu. Určitě si nejdříve vyzkoušejte jakoukoli konfiguraci, kde budete mít spuštěné zrcadlení a clustering dohromady. [1]

Rozdíly mezi zrcadlením a clusteringem
Nejvýraznější rozdíl spočívá v tom, že clustering probíhá na úrovni instancí, zatímco zrcadlení na úrovni databází. Zrcadlení v případě poruchy nepřepíná další služby systému SQL Server nebo další služby operačního systému. Výhoda zrcadlení v porovnání s clusteringem je taková, že zrcadlení nevyžaduje speciální hardware ani nevyžaduje podsystém sdílených disků. Zrcadlová databáze může existovat v různých umístěních. [1]

2.3 Replikace transakcí

Replikace transakcí se typicky používá při replikaci dat mezi servery. Replikace transakcí je také upřednostňovaná metoda pro replikace velkého množství dat, nebo když je vyžadována velmi nízká časová prodleva mezi vydavatelem a příjemcem[2]. Replikace transakcí je schopna fungovat, i když vydavatelem nebo příjemcem není databáze systému SQL Server, ale například systém Oracle.
U této replikace se distribuují pouze změněná data. Replikace transakcí může být nakonfigurována s množstvím různých typů publikací například replikace transakcí v režimu peer-to-peer. [1]

[2] Vydavatel a příjemce – například klientská aplikace

2.4 Porovnání vybraných technologií systému SQL Server

Tab. 2.1: Porovnání vybraných technologií systému SQL Server

Oblast	**Zrcadlení databáze**	**Clustering**	**Replikace transakcí**
Ztráta dat	Žádná ztráta dat	Žádná ztráta dat	Možnost určité ztráty dat
Automatický přechod na záložní systém	Ano, v režimu vysoké dostupnosti	Ano	Ne
Transparentní pro klienta	Ano, automatické přesměrování	Ano, připojení na stejnou IP adresu	Ne, ale pomůže program pro rozložení zátěže sítě (NLB)
Doba mimo provoz	< 3 sekundy	20 sekund i více, plus doba obnovy databáze	<10 sekund
Přístup k záložním datům	Ano, pomocí snímku databáze	Ne	Ano
Nejmenší objekt	Jen databáze	Všechny systémové i uživatelské databáze	Tabulky nebo pohledy
Maskování selhání disku	Ano	Ne, sdílené řešení disku	Ano
Speciální hardware	Ne doporučen je duplikát	Hardware na seznamu HCL hardwaru kompatibilního pro clustering	Ne, doporučen je duplikát
Složitost	Malá	Větší	Větší

3. Zabezpečení

Zabezpečení je v dnešní době důležitou součástí databází. V případě narušení SQL Serveru a proniknutí k datům může pro mnohé organizace znamenat konec konkurenceschopnosti, ztrátu klientů či jiných nepříjemností. Tato kapitola se zabývá funkcemi týkajícími se zabezpečení.

3.1 Zakázané funkce databázového zdroje

Aby se zajistilo, že systém SQL Server bude ve výchozím nastavení zabezpečený tak, jak nejvíc to bude možné, je ve výchozím nastavení zakázáno množství funkcí, které představují bezpečnostní riziko, a tyto funkce musí být před svým použitím povoleny.

To zahrnuje následující funkce: [1]

- Vzdálená připojení
- Vyhrazena linka správce
- Rozhraní .NET Framework
- Databázová pošta
- Funkce SQLMail
- Služba Service Broker
- Připojení pomocí protokolu http
- Zrcadlení databáze
- Služba Web Assistant
- Rozšířená procedura xp_cmdshell
- Vzdálené dotazy Ad hoc
- Rozšířené procedury automatizace OLE
- Rozšířené procedury objektů SMO a DMO

3.2 Principy a zabezpečené objekty

Model zabezpečení systému SQL Server je založen na dvou mechanismech – principy a zabezpečené objekty. Principy jsou takové objekty, které mohou získat oprávnění pro přístup k určitým databázovým objektům. Zabezpečené objekty jsou takové objekty, k nimž lze řídit přístup. [1]

3.2.1 Principy

Principy mohou představovat určitého uživatele, roli, kterou může přijmout více uživatelů, nebo aplikací. Systém SQL Server dělí principy do tří tříd:

Principy systému Windows
Systém SQL server umožňuje vytvořit přihlašovací jména z přihlašovacích účtů nebo skupin systému Windows. Ty mohou příslušet buď místnímu počítači, nebo doméně. Když se uživatelé přihlásí k systému SQL Server pomocí ověření systému Windows, musí být jejich aktuální uživatelský účet vytvořen jako přihlašovací jméno v systému SQL Server nebo musí být členem skupiny uživatelů systému Windows, která existuje jako přihlašovací jméno. [1]

Principy systému SQL Server
Zatímco ověření systému Windows spoléhá na samotný operační systém, že provede ověření (rozhodne, kdo uživatel doopravdy je), a systém SQL Server provede jen autorizaci (rozhodne, které akce může ověřený uživatel provádět), při ověření systémem SQL Server samotný systém SQL Server provádí současně ověření i autorizaci. Podporuje současně jednotlivá přihlašovací jména a role serveru, ke kterým může být přiřazeno více uživatelů. [1]

Principy databáze
Principy databáze jsou ty objekty, které představují uživatele, kterým mohou být přiřazena oprávnění k přístupu do databází nebo k určitým objektům v databázi. Zatímco přihlašovací jména fungují na úrovni serveru a umožňují provádět akce jako připojení k systému SQL Server, principy databáze

fungují na úrovni databáze a umožňují vybrat nebo provádět manipulace s daty, vykonávat příkazy DDL (Data Definition Language) na objektech uvnitř databáze nebo spravovat uživatelská oprávnění na úrovni databáze. [1]

3.2.2 Zabezpečené objekty

Zabezpečené projekty jsou databázové objekty, k nimž můžete řídit přístup a ke kterým můžete udělovat povolení pro principy. Systém SQL Server rozlišuje mezi třemi obory, ve kterých mohou být objekty zabezpečeny:

Obor serveru
Zabezpečené objekty v oboru serveru zahrnují přihlašovací jména, koncové body protokolu http, oznámení o událostech databáze. Tyto objekty existují na úrovni serveru mimo všechny jednotlivé databáze, ke kterým se řídí přístup pro celý server. [1]

Obor databáze
Zabezpečené objekty uvnitř oboru databáze jsou objekty jako uživatelé, role a sestavení CLR (common language runtime – runtime společného jazyka), které existují uvnitř určitých databází, ale ne uvnitř schémat. [1]

Obor schématu
Tato skupina obsahuje ty objekty, které přebývají uvnitř schématu v databázi, jako jsou např. tabulky, pohledy, a uložené procedury. Schéma je vrstva v hierarchii zabezpečení systému SQL Server. Uživatel může vlastnit více schémat a schéma může být vlastněno více než jedním uživatelem. [1]

3.3 Oprávnění

Oprávnění jsou jednotlivá práva, která jsou udělena principu databáze pro přístup k zabezpečenému objektu. Oprávnění povolující přístup uděluje příkaz GRANT, oprávnění odpírající přístup příkaz DENY a příkaz ke zrušení oprávnění, která již byla udělena REVOKE. Přesná oprávnění, která je možná udělit, a formát příkazu GRANT a DENY se mění podle zabezpečeného objektu. Je možně je rozdělit do 12 skupin:

Oprávnění k serveru
Oprávnění, která se vztahují k serveru jako celku, jakými jsou oprávnění k připojení k serveru nebo ke koncovému bodu, oprávnění k vytvoření nebo pro změnu událostí DDL (Data Definition Language) nebo jiných a oprávnění k přístupu k externím zdrojům ze systému SQL Server. [1]

Oprávnění ke koncovým bodům http
Oprávnění k připojením ke koncovému bodu a pro řízení, změnu, zobrazení definice nebo převzetí vlastnictví daného objektu. [1]

Oprávnění k certifikátům
Povolení pro změnu nebo řízení určitého certifikátu. [1]

Oprávnění k databázi
Oprávnění k celé databázi, která se vztahují na všechny objekty v aktuální databázi. Zahrnují například oprávnění k vytváření, změně a spouštění objektů v dané databázi. [1]

Oprávnění ke schématům
Oprávnění, která se vztahují na zadané schéma nebo ke všem objektům ve schématu. Obsahují schopnost provádět operace SELECT, INSERT, UPDATE nebo DELETE na jakémkoliv objektu ve schématu a ke spouštění jakékoliv procedury nebo funkce uvnitř schématu a k řízení, změně nebo převzetí vlastnictví schématu. [1]

Oprávnění k sestavením
Oprávnění k určitému sestavení, jako jsou oprávnění ke spouštění, řízení, změně nebo převzetí vlastnictví sestavení. [1]

Oprávnění k typům
Oprávnění k určitému typu definovaného uživatelem, jako jsou oprávnění ke spouštění, řízení, změně nebo převzetí vlastnictví daného typu. [1]

Oprávnění k full-textovým katalogům
Oprávnění k odkazováním, převzetí vlastnictví, zobrazení definice nebo řízení katalogu. [1]

Oprávnění ke službě Service Broker
Oprávnění k určitému objektu služby Service Broker. Tato oprávnění se mírně mění v závislosti na typ objektu. [1]

Oprávnění k principům serveru
Oprávnění k zosobnění daného přihlašovacího jména nebo ke změně, zobrazení definice, převzetí vlastnictví nebo řízení daného přihlašovacího jména. [1]

Oprávnění k principům databáze
Oprávnění k zosobnění daného uživatele nebo ke změně, řízení nebo zobrazení definice daného principu databáze. [1]

Oprávnění k objektům
Oprávnění udělená k zabezpečeným objektům v oboru schématu, jakými jsou tabulky, pohledy nebo uložené procedury, a ke spouštění nebo k provádění operací SELECT, DELETE nebo dalších operací na daném objektu. [1]

3.4 Zabezpečení přístupu kódu

Zabezpečení přístupu kódu (CAS - Code Access Security) je mechanismus platformy .NET, který umožňuje vývojářům explicitně uvést, která oprávnění jejich kód potřebuje ke spuštění (např. oprávnění ke spuštění nespravovaného kódu nebo oprávnění k přístupu do některé části

souborového systému), a také umožňuje udělit nebo zamítnout oprávnění pro určitý kód, aby provedl určité akce. [1]

S oprávněním CAS je možné provádět čtyři základní akce:

Uplatnit (Assert)
Uplatnění oprávnění umožňuje sekci kódu provést danou akci dokonce, i když metoda, která zavolala aktuální metodu, nemá daná oprávnění. Samostatný kód musí mít oprávnění k provedení dané akce, jinak uplatnění selže. [1]

Odmítnout (Deny)
Odmítnout oprávnění způsobí, že jakékoliv pokusy pro vykonání zakázané akce nebo o vyžádání stejného oprávnění dále v zásobníku volání metody selžou. Avšak odmítnutí je možné přepsat následným voláním metod Assert() nebo PermitOnly().[1]

Požadovat (Demand)
Požadavek na oprávnění signalizuje, že kód vyžaduje ke svému spuštění oprávnění. Požadavek se přidělí, pouze pokud nebyl zakázán dříve v zásobníku volání nebo povolen pouze odlišném prostředku. [1]

Jen povolit (PermitOnly)
Můžete poskytnout přístup jen k určitému prostředku a odmítnou přístup k ostatním prostředkům, které vyžadují stejné oprávnění. Oprávnění PermitOnly může bát přepsáno potvrzením (assert) nebo odmítnutím (deny) oprávnění, ale ne dalším PermitOnly. [1]

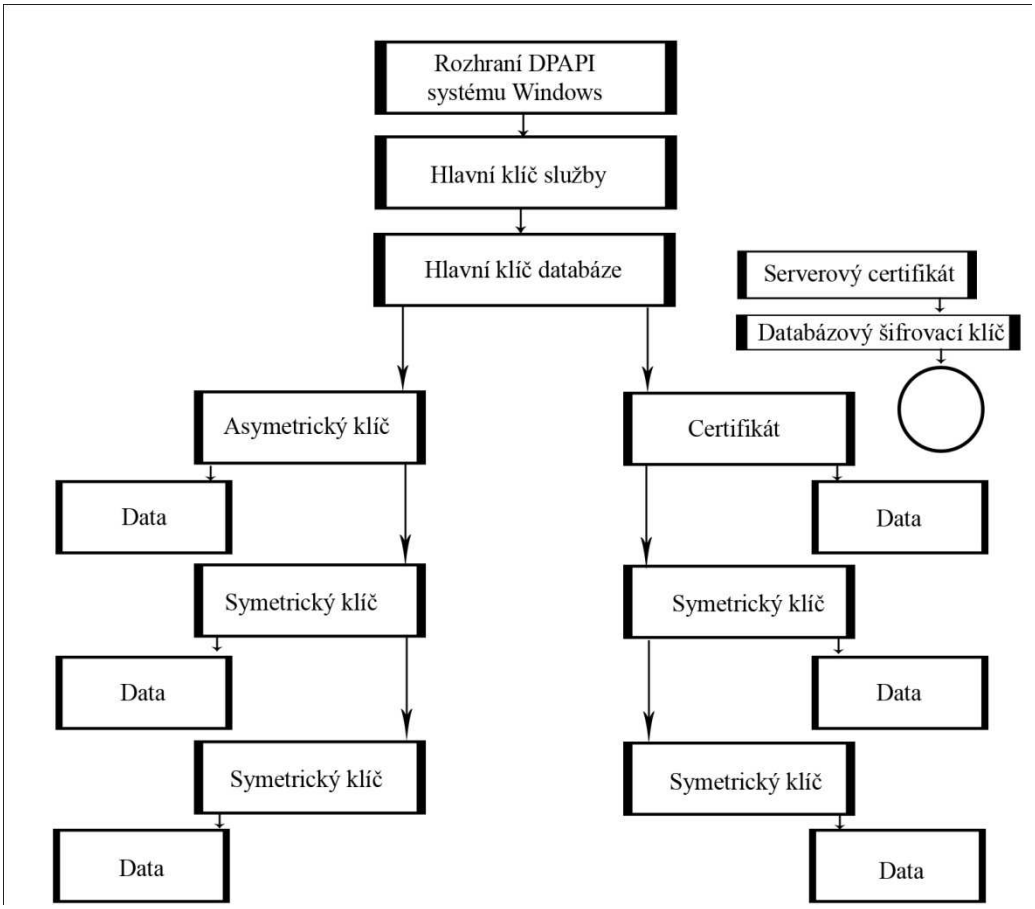
CAS je možné implementovat dvěma způsoby: Deklarativně pomocí atributů pro vyžádání a uplatnění oprávnění, nebo imperativně pomocí volání metod na jednotlivé objekty oprávnění. Oprávnění je také možné konfigurovat pro celá sestavení, skupiny sestavení, nebo dokonce celý místní počítač nebo doménu. [1]

4. Šifrování

Šifrování je proces „zamlžení“ dat pomocí klíče nebo hesla. Data jsou tedy nepoužitelná bez dešifrovacího klíče nebo hesla. Šifrování nevyřeší potíže s řízeným přístupem, avšak zvyšuje zabezpečení, omezení ztráty dat i v případě, že byly obejity ovládací prvky pro přístup.

4.1 Hierarchie šifrovacích klíčů

Model šifrování systému SQL serveru obsahuje zabudovanou správu šifrovacích klíčů ve formátu v souladu se standardem ANSI X9.17. Tento standard definuje několik vrstev šifrovacích klíčů, které se používají k zašifrování dalších klíčů, které se následně používají k zašifrování vlastních dat. [1]



Obr. 4.1.1: Hierarchie šifrovacích klíčů v systému SQL Server

Hlavní klíč služby (SMK – Service Master Key) je klíč nejvyšší úrovně, základ všech klíčů v systému SQL Server. Existuje jediný hlavní klíč služby definovaný pro každou instanci systému SQL Server. Hlavní klíč služby je zabezpečen pomocí rozhraní DPAPI systému Windows (Windows Data Protection API) a je vydán pro zašifrování další vrstvy klíčů, hlavních klíčů databází (DMK – Database Master Key). Hlavní klíč služby se vytvoří automaticky při první příležitosti, když je potřeba. [1]

Hlavní klíče databáze se používají pro zašifrování symetrických klíčů, asymetrických klíčů a certifikátů. Každá databáze můžu mít jediný, pro ni definovaný hlavní klíč databáze. [1]

Další vrstva klíčů obsahuje symetrické klíče, asymetrické klíče a certifikáty. Symetrické klíče jsou primární prostředky pro zašifrování dat v databázi. Zatímco asymetrické klíče a certifikáty je možné použít k zašifrování dat, Microsoft doporučuje, abyste šifrovali data výhradně s pomocí symetrických klíčů. [1]

Ke všem klíčům a certifikátům představeným v systému SQL Server 2005, systém SQL Server 2008 navíc představuje koncept serverových certifikátů a databázových šifrovacích klíčů pro podporu nové funkce transparentního šifrování dat. Serverový certifikát je prostě certifikát vytvořený v databázi master. Databázový šifrovací klíč je speciální symetrický klíč, který se používá k zašifrování celé databáze najednou. [1]

4.2 Mechanismy šifrování

SQL Server poskytuje následující mechanismy pro šifrování:

- Asymetrické klíče

- Certifikáty
- Symetrické klíče
- Transparentní šifrování dat
- Rozšířitelná správa klíčů
- Šifrování bez klíčů

Asymetrické klíče

Asymetrický klíč je tvořen ze dvou šifrovacích klíčů: veřejného klíče a privátního klíče. Privátní klíč může nabývat délky 512, 1024 a 2048 bitů. Microsoft doporučuje, abyste použili asymetrické klíče jen k zašifrování symetrických klíčů a tyto symetrické klíče k zašifrování vašich dat. Jedním důvodem je pro to rychlost. Symetrické šifrování je značně rychlejší než asymetrické šifrování. Dalším důvodem pro zašifrování dat pomocí symetrického šifrování je omezení na velikosti dat, se kterou si poradí asymetrické šifrování. Omezení asymetrických algoritmů založených na délkách privátních klíčů implementovaných v systému SQL Server je uvedeno v tabulce 4.1 [1]

Tab. 4.1: Asymetrické algoritmy, délky klíčů a omezení

Algoritmus	Privátní klíč	Prostý text	Zašifrovaný text
RSA_512	512 bitů	53 bajtů	64 bajtů
RSA_1024	1024 bitů	117 bajtů	128 bajtů
RSA_2048	2048 bitů	245 bajtů	256 bajtů

Certifikáty

Certifikáty jsou dalším nástrojem, který systém SQL Server poskytuje pro asymetrické šifrování. Certifikát je v podstatě pár veřejný a privátní klíč asymetrického klíče, který obsahuje další data popisující daný certifikát. Další data obsahují počáteční datum, datum vypršení a předmět certifikátu. Na rozdíl od asymetrických klíčů systému SQL Server mohou být certifikáty zálohovány do souborů a také z nich obnoveny. Systém SQL Server podporuje certifikáty, které se řídí standardem ITU-T X.509. Microsoft doporučuje, abyste certifikáty, stejně jako asymetrické klíče, použili pro zašifrování vašich symetrických klíčů a tyto symetrické klíče použili k zašifrování vašich dat. Zašifrování pomocí certifikátů má stejná omezení délky jako asymetrické zašifrování. [1]

Symetrické klíče

Symetrické zašifrování vyžaduje jediný klíč současně pro zašifrování i dešifrování vašich dat. Symetrické šifrování se provádí pomocí blokových šifrovacích algoritmů, které šifrují vaše data po blocích o konstantní velikosti, a proudových šifrovacích algoritmů, které šifrují vaše data v průběžném proudu. Blokové šifrovací algoritmy mají nastavenou velikost šifrovacího klíče a velikost šifrovaného bloku, jak vidíte v tabulce 4.2 [1]

Tab. 4.2: Podporované algoritmy v systému SQL 2008

Algoritmus	Délka uloženého klíče	Skutečná délka klíče	Velikost bloku	Poznámka
DES	64 bitů	56 bitů	64 bitů	
Triple_DES	128 bitů	112 bitů	64 bitů	
Triple_DES_3KEY	128 bitů	112 bitů	64 bitů	Tato volba je dostupná pro databázové šifrovací klíče
DESX	192 bitů	184 bitů	64 bitů	
RC2	128 bitů	128 bitů	64 bitů	
RC4	40 bitů	40 bitů	N/A	Microsoft doporučuje nepoužívat algoritmus RC4 pro zašifrování vašich dat. Algoritmus RC4 je proudová šifra, položka velikost bloku se tak na ní nevztahuje.
RC4_128	128 bitů	128 bitů	N/A	Microsoft doporučuje nepoužívat algoritmus RC4_128 pro zašifrování vašich dat. Algoritmus RC4_128 je proudová šifra, položka velikost bloku se tak na ní nevztahuje.
AES_128	128 bitů	128 bitů	128 bitů	
AES_192	192 bitů	192 bitů	128 bitů	

Transparentní šifrování dat

Transparentní šifrování dat šifruje každou stránku vaší celé databáze a automaticky dle potřeby dešifruje každou stránku během přístupu. Tato funkce vám umožňuje zabezpečit celou vaši databázi, aniž byste se starali o podrobnosti zašifrování na úrovni sloupců. Výhoda této funkce je v tom, že vám umožní zabezpečit transparentně vaši databázi bez jakýchkoli změn v koncových aplikacích. Transparentní zašifrování dat nevyžaduje žádné místo navíc a může generovat daleko efektivnější plány dotazů než dotazy na zašifrovaná data, protože transparentní zašifrování dat umožňuje systému SQL Server používat řádné indexy. Slabou stránkou transparentního zašifrování dat je skutečnost, že vyžaduje další režii, protože systém SQL Server musí dešifrovat každou stránku dat při každém dotazu. [1]

Rozšiřitelná správa klíčů

Rozšiřitelná správa klíčů (EKM – Extensible Key Managment) vám umožňuje použít aplikační rozhraní Microsoft Cryptographic API (CryptoAPI) pro zašifrování a generování klíčů. Podpora rozšiřitelné správy klíčů je navržena tak, aby umožňovala prodejcům třetích stran poskytovat hardware pro generování šifrovacích klíčů a další nástroje známé jako moduly hardwarového zabezpečení (HSM – Hardware Security Module). Modul hardwarového zabezpečení může nabídnout mnoho výhod oproti

standardním zabudovaným šifrovacím funkcím, jako je hardwarem urychlené šifrování a dešifrování nebo další správa klíčů. [1]

Šifrování bez klíčů

Kromě použití certifikátů, asymetrických klíčů a symetrických klíčů můžete zašifrovat vaše data pomocí heslových frází. Heslová fráze je řetězec nebo binární hodnota, ze které může systém SQL Server odvodit symetrický klíč pro zašifrování vašich dat. [1]

Další zabezpečení je použití otisků dat. Otisky dat používají algoritmy MD2, MD4, SHA nebo SHA-1. První tři jsou algoritmy Message Digest, které ze vstupu generují 128 bitové hodnoty otisku. Poslední dvě hodnoty jsou algoritmy Secure Hash Algorithm, které generují 160bitový otisk vstupu.

SQL Server také poskytuje funkce pro podepsání dat pomocí certifikátů a asymetrických klíčů a pro ověření těchto podpisů. To je užitečné pro ochranu integrity citlivých dat, protože jakákoli drobná změna dat ovlivní podpis. Délka podpisu závisí na délce certifikát nebo privátního klíče v asymetrickém klíči. [1]

5. Sledování a ladění výkonu

Cílem sledování databáze je posouzení, zda server pracuje správně. Účinné sledování spočívá v dělání pravidelných snímků aktuálního výkonu sloužící k izolaci procesů, které způsobují problémy. Microsoft SQL Server a operační systém Microsoft Windows poskytují nástroje, které umožňují zobrazit aktuální stav databáze a sledovat výkonnost při jakékoli změně.

Sledování SQL Serveru umožňuje: [4]

- Zjištění, zda jde vylepšit výkon. Například sledováním doby odezvy pro často používané dotazy můžeme určit, zda jsou vyžadovány změny indexu tabulky nebo dotazu.
- Vyhodnotit činnost uživatelů. Například pro sledování uživatelů, kteří se snaží připojit k instanci SQL Server, můžete určit, zda zabezpečení je nastaveno přiměřeně.
- Řešení potíží nebo ladění aplikací, jako jsou například uložené procedury.

Systém Windows obsahuje následující nástroje pro sledování aplikací, které jsou spuštěny na serveru: [4]

- Nástroj System Monitor (sledování systému), umožňuje shromáždit a zobrazit data v reálném čase o aktivitách, jako je například využití paměti, disku a procesoru.
- Výstrahy a protokolování výkonu
- Správce úloh

SQL Server nabízí následující nástroje pro monitorování složek SQL Serveru: [4]

- Trasování SQL
- SQL Server Profiler – primární nástroj pro analýzu výkonu a ladění systému SQL Server. Používá se také pro zachytávání dotazů a příkazů, které jsou odesílány specifickému serveru.
- SQL Management Studio Activity Monitor – sledování činnosti graficky zobrazí informace: o procesech spuštěných v instanci SQL Severu, blokovaných procesů, zámků a činnost uživatele. To je vhodné zejména pro ad hoc zobrazení aktuální aktivity.
- SQL Management Studio Graphical Showplan

- Systémové uložené procedury

Tab. 5.1: Příklad systémových úložných procedur a jejich funkce

Procedura	Poznámka
sp_who	zobrazí informace o aktuálních uživatelích a procesů SQL Serveru, včetně aktuálně vykonávajícího výrazu, a zda je výraz blokován
sp_lock	zobrazí informace o uzamčení, včetně ID objektu, ID indexu, typu zámku a typ nebo prostředek, na něž se zámek vztahuje
sp_spaceused	zobrazí odhad aktuálního množství místa na disku a velikost databází
sp_monitor	zobrazí statické údaje, včetně využití procesoru, využití I/O a dobu nečinnosti od doby, kdy byla procedura spuštěna

- Příkazy konzoly databáze (DBCC – Database Console Commands), umožňují kontrolovat statistiky výkonu o logické a fyzické konzistenci výkonu.
- Předdefinované funkce

Tab. 5.2: Porovnání vybraných nástrojů na sledování SQL Serveru

Událost nebo aktivita	SQL Server Profiler	System Monitor	Activity Monitor	Transact-SQL	Error logs
Analýza trendů	Ano	Ano			
Přehrání sběru událostí	Ano				
Sledování ad hoc	Ano		Ano	Ano	Ano
Generování výstrah		Ano			
Grafické rozhraní	Ano	Ano	Ano		Ano
Použití v rámci vlastní aplikace	Ano			Ano	

6. Microsoft SQL Server Management Studio

SQL Server Management Studio je integrované prostředí pro správu infrastruktury SQL Serveru. Management studio poskytuje nástroje pro konfiguraci, monitorování a správu instancí SQL Serveru. Také poskytuje nástroje pro vývoj, monitorování a aktualizování datových součástí, například databází a datových skladů používaných aplikací, a tvoření dotazů a skriptů. SQL Server Management Studio kombinuje širokou skupinu grafických nástrojů s množstvím bohatých skriptovacích editorů zajišťující přístup k SQL Serveru jak pro vývojáře, tak i pro správce všech úrovní. [5]

Funkce management studia: [5]

- Podpora většiny administrativních úkolů pro SQL Server
- Jednotné, integrované prostředí pro SQL Database Engine
- Nové vedení dialogů pro správu objektů v SQL Server Database Engine, Analysis Services, Reporting Services, Notification Services a SQL Server Compact 3.5 SP1, které vám umožňuje provádět své akce okamžitě, odeslat do kódového editoru nebo do skriptu pro pozdější provedení.
- Přístup k více nástrojů najednou
- Společné plánování, které umožňuje provést akce později
- Export a import dat
- Uložit nebo vytisknout prováděcí plán XML nebo vzájemně zablokovat soubory generovanými SQL Server Profiler, nebo odeslat správci pro analýzu
- Odeslání případné chyby na Microsoft
- Sledování činnosti s filtrováním a automatická aktualizace
- Integrovaná pošta
- Zápis T-SQL

7. Srovnání verzí Microsoft SQL Server

7.1 Microsoft SQL Server 2000

Relační databázový a analytický systém pro komerční, podnikový a datové využití. SQL Server 2000 obsahuje podporu pro XML a HTTP, výkon, dostupnost funkcí při rozdělení zatížení, zajištění provozuschopnosti, pokročilou správu, ladění funkcí pro automatizaci rutinních úloh a nižší celkové náklady na vlastnictví. Podporuje náročné aplikace jako OLTP (Online Transaction Processing) a aplikace usnadňující rozhodování, jejichž jádrem je Transact-SQL (verze jazyku SQL od firmy Mircrosoft). MS SQL Server 2000 také podporuje replikaci, která umožňuje udržovat více kopií dat. [6]

Edice MS SQL Server 2000: [7]

- SQL Server 2000 Enterprise Edition
- SQL Server 2000 Standard Edition
- SQL Server 2000 Personal Edition
- SQL Server 2000 Developer Edition
- SQL Server 2000 Windows CE Edition
- SQL Server 2000 Enterprise Evaluation Edition

7.2 Microsoft SQL Server 2005

Microsoft SQL Server 2005 rozšiřuje výkon, spolehlivost, dostupnost, programovatelnost a snadnost používání SQL Server 2000. SQL Server 2005 obsahuje několik nových funkcí, díky nimž je vynikající databázovou platformou pro rozsáhlé on-line transakční zpracování (OLTP), datové sklady a e-komerční aplikace. [8]

Funkce a nové nástroje MS SQL 2005: [8]

- Notification Services (oznámení služby) můžete zaslat včas personalizované zprávy na tisíce nebo miliony účastníků, kteří používají širokou škálu zařízení.
- Reporting Services je nová serverová platforma, která podporuje hlášení vytváření sestav, distribuci, správu a přístup koncových uživatelů.
- Service Broker je nová technologie pro vytváření databázových aplikací, které jsou bezpečné, spolehlivé a škálovatelné.
- Database Engine přináší nové programovatelnost vylepšení, jako je integrace s Microsoft. NET Framework, Transact-SQL vylepšení, nové XML funkce a nové datové typy. To také zahrnuje vylepšení škálovatelnosti a dostupnosti databází.
- SQL Server 2005 přináší zlepšení v programovacích rozhraní používané pro přístup k datům v databázích SQL Serveru.
- Analysis Services představuje nové nástroje pro správu a integrované vývojové prostředí a integraci s .NET Framework.
- Integration Services zavádí novou rozšiřitelnou architekturu a nový návrh, který odděluje pracovní tok z datového toku a nabízí bohatou sadu sémantiky kontroly toků. Integration Services poskytuje také zlepšení správu balíků a nasazení společně s mnoha novými úkoly.
- Fulltextové vyhledávání
- Replikace nabízí zlepšení ve správě, dostupnosti, programovatelnosti, mobilitě, škálovatelnosti a výkonu.
- SQL Server Management Studio
- Business Intelligence Development Studio

7.3 Microsoft SQL Server 2008

SQL Server 2008 obsahuje řadu funkcí a nástrojů, které lze použít na rozvoj a správu vašich databází a jejich řešení.

Další nástroje, které SQL Server 2008 nabízí: [9]

SQL Server Profiler
Jedná se o nástroj, který zachycuje události ze serveru. Ty ukládá do souboru, který může být později analyzován nebo použit k zopakování určité řady kroků.

SQL Server Configuration Manager

SQL Server Configuration Manager je nástroj pro správu služeb spojených s SQL Serverem, konfigurace síťových protokolů používaných SQL Serveru a správu sítě, připojení konfiguraci z počítače serveru SQL klienta.

Database Engine Tuning Advisor (DETA)

Pomůže vybrat a vytvořit optimální nastavení indexů, indexovaných pohledů a oddělení bez expertních znalostí databáze nebo vnitřního rozhraní MS SQL Serveru.

7.4 Microsoft SQL Server 2008 R2

SQL Server 2008 R2 přidává některé funkce ke stávajícím SQL Serveru 2008, včetně správy kmenových dat systému (Master Data Services), centralizované konzoly sloužící ke spravování více instancí SQL Serveru s podporou více než 64 logických procesorů.

7.5 Microsoft SQL Server 2012

SQL Server 2012 obsahuje několik vylepšení oproti předchozí verzi SQL Server 2008 R2. Mezi nové funkce patří AlwaysOn SQL Server instance převzetí služeb při selhání clusteru a dostupnost skupin, které poskytují řadu možností jak zlepšit dostupnost databáze. Zjednodušení přesouvání databází mezi instancemi, nové modifikované zobrazení pro dynamickou správu a funkce, vylepšení programovatelnosti včetně nových prostorových funkcí, podpora metadat, zvýšení výkonu pomocí nového indexování ColumnStore, bezpečnostní vylepšení během instalace, nové vyhledávaní oprávnění, nové uživatelské definované role serveru a nové způsoby řízení serveru a databázové role. [10]

7.6 Podporované hardwarové prostředky jednotlivých verzí

	MS SQL 2000	MS SQL 2005	MS SQL 2008	MS SQL 2012
maximální velikost databáze	1 048 516 TB	524 258 TB	524 272 TB	524 272 TB
maximální počet procesorů	Podle edice Enterprise: OS maximum	Podle edice Enterprise: OS maximum	Podle edice Enterprise: OS maximum	Podle edice Enterprise: OS maximum
maximální množství paměti	Podle edice Enterprise: OS maximum	Podle edice Enterprise: OS maximum	Podle edice Enterprise: OS maximum	Podle edice Enterprise: OS maximum
maximální počet instancí na jednom PC	16	50	50	50

Tab. 7.1: Podporované hardwarové prostředky jednotlivých verzí

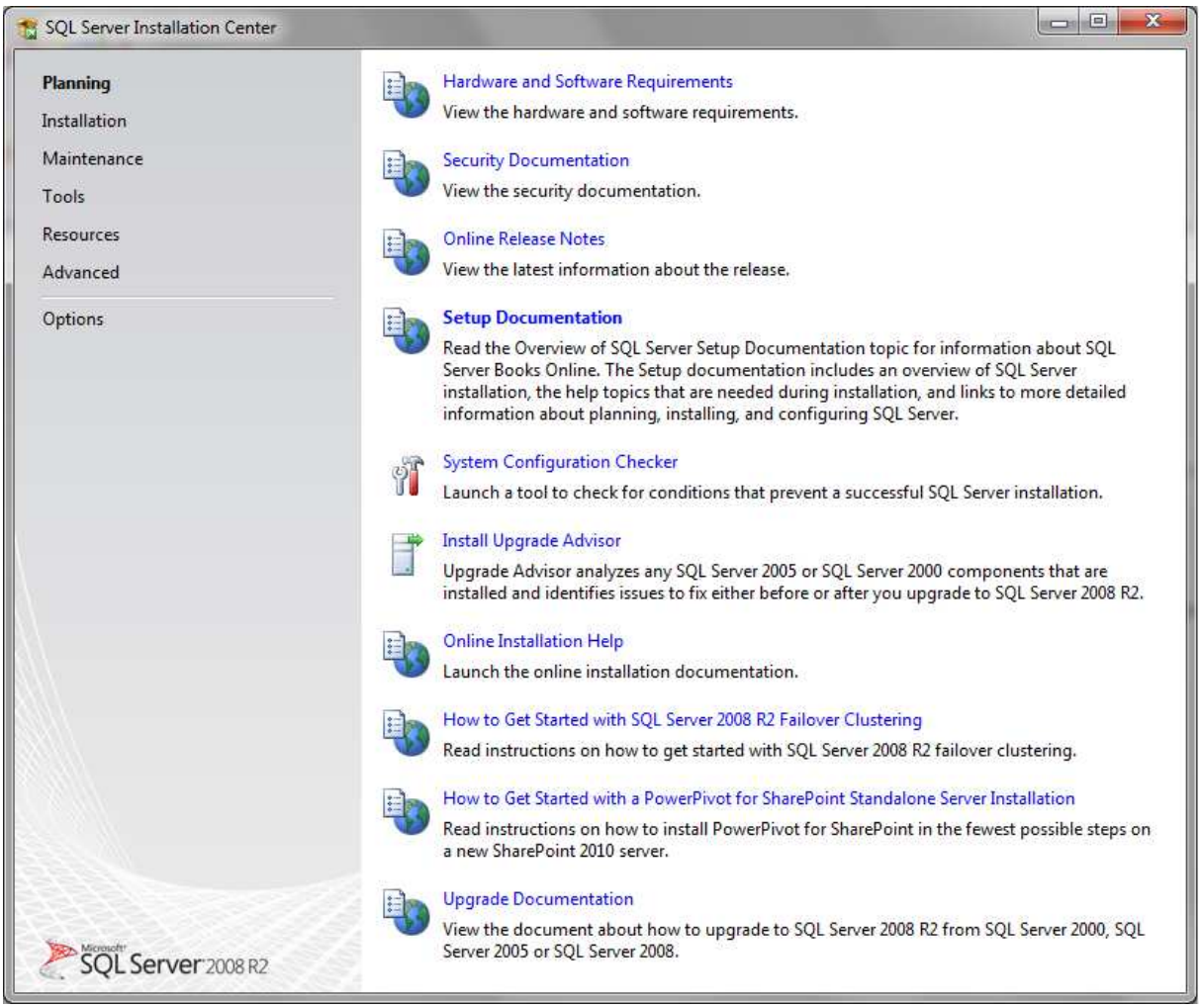
8. Úvod k praktické části

V praktické části bakalářské práce se zaměřím na sledování výkonu při manipulaci s databází na jednotlivých verzích SQL Serveru a při vzdáleném a lokálním přístupu k databázi. V další části se budu zabývat sledování komunikace mezi SQL Serverem a klientem pomocí programu na analýzu paketů. Z dosažených výsledků vyvodím závěr a nastíním nejvhodnější optimalizaci databáze.

8.1 Příprava

8.1.1 Instalace SQL Serveru

Jako první krok je samozřejmě opatření instalačního DVD či image. Pro svou práci jsem použil trial verzi na dobu 180 dní, kterou jsem stáhl na stránkách Microsoft. Při spuštění instalačního menu je jako první položka *Planning.*



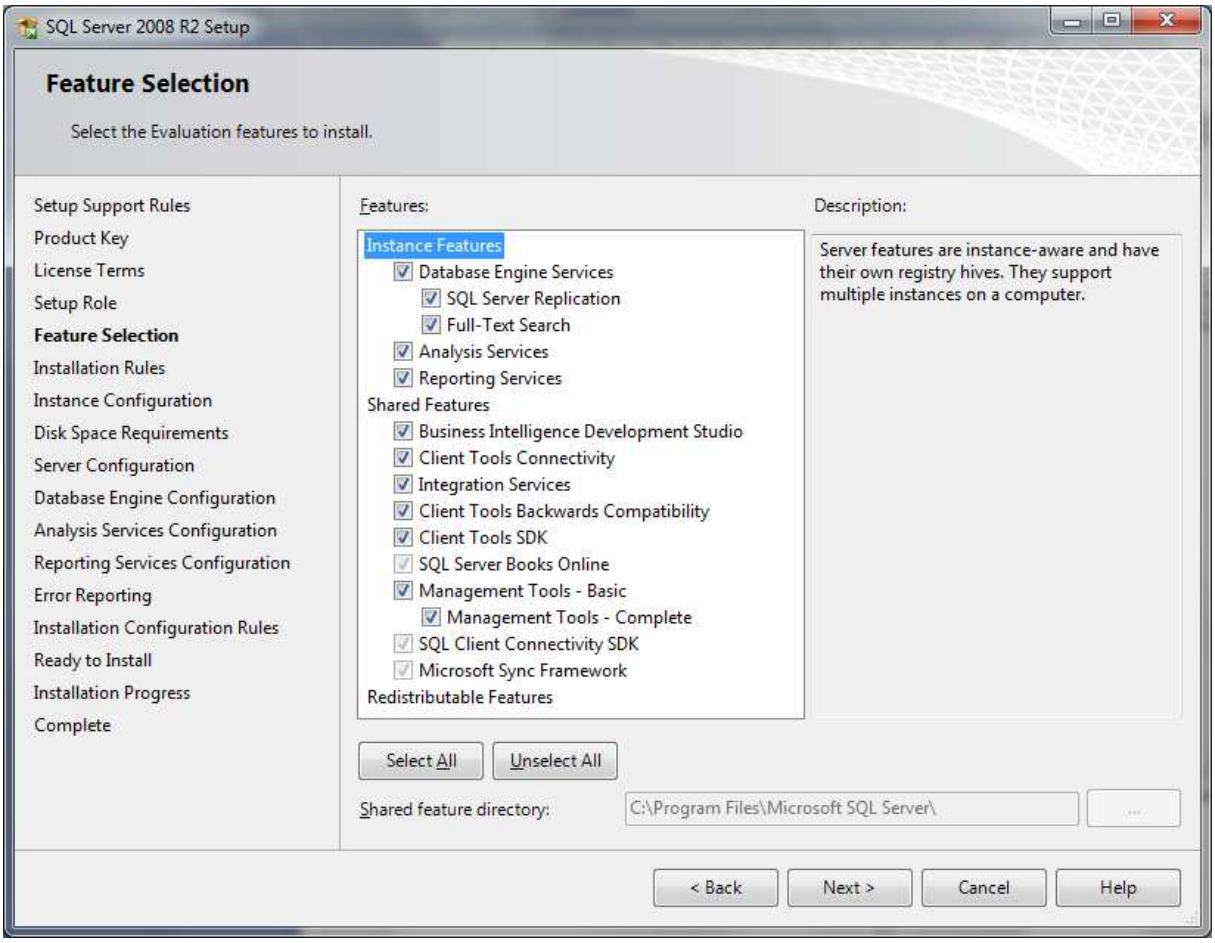
Obr. 8.1: Instalační menu - Planning

Zde je uvedena všechna potřebná dokumentace pro instalaci SQL Server a příslušných nástrojů. Planning neboli plánování obsahuje: dokumentaci o minimálních požadavcích k instalaci SQL Server, zabezpečení před instalací, poznámky k vydání, instalační dokumentace, nástroj pro zkontrolování podmínek pro úspěšnou instalaci SQL Serveru, instalaci nástroje Upgrade Advisor, který slouží

k aktualizaci SQL Serveru na novější verzi, online instalační pomoc, návod začínáme s SQL Serverem u převzetí služeb při selhání, návod začínáme s PowerPivot pro SharePoint, a dokumentace k aktualizaci na nejnovější verzi.

Obr. 8.2: Instalační menu - Installation

Dalším položkou instalačního menu je Installation,kde vybereme New installation or add features to an existing installation.



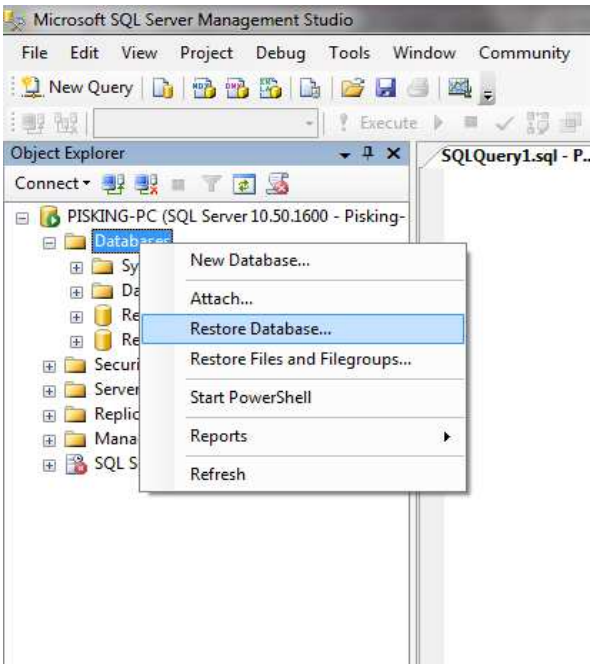
Obr. 8.3: kroky instalace, výběr funkcí

Nejdůležitější kroky instalace a jejich stručný popis:
Feature Selection (výběr funkcí) – vybrání jednotlivých nástrojů a služeb pro instalaci
Instance Configuration (konfigurace instancí) – určujeme, zda chceme nainstalovat výchozí či vlastní instance
Disk Space Requirements (diskový prostor) – vypočítá prostor na disku pro instalované funkce
Server Configuration (nastavení serveru) – nastavení přihlašovacích účtů a kolace
Database Engine Configuration (konfigurace účtu) – nastavení ověřování (Windows, kombinované) a instalačního adresáře
Analysis Services Configuration (nastavení služeb pro analýzu) – nastavení uživatele nebo účtu, který bude mít oprávnění správce pro Analysis Services.
Reporting Services Configuration (nastavení služeb reportu) – tři možnosti nastavení: nativní režim výchozí konfigurace, SharePoint režim výchozí konfigurace a nekonfigurovaná instalace.
Error reporting (chybové hlášení) – nastavit, zda posílat chyby Microsoftu
Installation (instalace)

8.1.2 Nahrání testovací databáze na SQL Server

Otevřeme si Management Studio a vytvoříme si novou databázi: v object exploreru pravým tlačítkem myši kliknene na Databases a z nabídky vybereme New Database. Dalším krokem je nahrání testovací

databáze do naší vytvořené databáze, kterou jsem si pojmenoval bc_prace: v object exploreru pravým tlačítkem myši kliknene na Databases a z nabídky vybereme Restore Database.



Obr. 8.4: Postup při restorování databáze

V otevřené okně Restore Database vybereme cíl – To database: bc_prace a následně zdroj – From device testovací databázi. Označímě Restore a dáme OK.

9. Výkonnostní testy

Výkonnostní testy budu provádět nad tabulkami jm_template.Template a jm_template.Template_test.

jm_template.Template
Tabulka obsahuje přehled šablon a to, jestli je šablona aktivní, či nikoli. Šablony jsou v podobě varchar. Každá šablona je složena ze 128x128 ASCII znaků. Každý ASCII znak je možné převést do int (0..255) a je tak možné rychle získat popis 128x128hodnot, kde každá hodnota pochází z rozsahu (0..255). Každý subjekt je charakterizován jednou nebo více šablonami, které ho jednoznačně identifikují na základě hlasu, obličeje, postavy a dalších ukazatelů.

jm_template.Template_test
Tabulka obsahuje přehled šablon a to, jestli je šablona aktivní, či nikoli. Tabulka je stejná jako jm_template.Template s tím rozdílem, že jsou zde šablony uloženy ve formě binárních dat. Je tak možné testovat, který přístup k ukládání šablon je lepší (jestli varbinary, nebo varchar). Každý subjekt je charakterizován jednou nebo více šablonami, které ho jednoznačně identifikují na základě hlasu, obličeje, postavy a dalších ukazatelů.

Testy spočívají v pouštění dotazů jazyka Transact SQL různých složitostí na vybraných verzích a na vzdáleném či lokálním přístupu k SQL Serveru. Z výsledků testů určím výhodnější verzi, přístup, a zda je data lepší uchovávat v datových typech varchar nebo varbinary.

9.1 Hardware konfigurace použitých serverů a počítačů.

Hardware konfigurace lokálního serveru/klienta:
Procesor: Intel ® Core™2 Duo CPU P8600 @ 2,40GHz
Paměť: DDR2 4096 Mb, Dual symetric
Operační systém: Windows 7

Hardware konfigurace vzdáleného serveru:
Procesor: AMD Phantom™ II X2 560 Procesor 3,3 GHz
Paměť: DDR3 4096 Mb, single
Operační systém: Windows 7

9.2 Operace nad tabulkou jm_template.Template

9.2.1 Lokální přístup

Microsoft SQL Server 2008 R2

1. V programu dodaného vedoucím bakalářské práce jsem pustil dotaz na zobrazení 30 000 řádků z tabulky `jm_template.Template`:

```
select top 30000 * from bc_prace.jm_template.Template
order by TemplateID asc
```

Měření prováděno při využití procesoru 7% a využití paměti 4096/2511 Mb.

Výsledky:

Query time is 00:00:20.3971667
Total rows count 30000
Time for 1 row (ms) 0,679905556666667

Query time is 00:00:06.9193958
Total rows count 30000
Time for 1 row (ms) 0,230646526666667

Query time is 00:00:08.2014691
Total rows count 30000
Time for 1 row (ms) 0,273382303333333

Query time is 00:00:06.7813878
Total rows count 30000
Time for 1 row (ms) 0,22604626

Query time is 00:00:06.2873596
Total rows count 30000
Time for 1 row (ms) 0,209578653333333

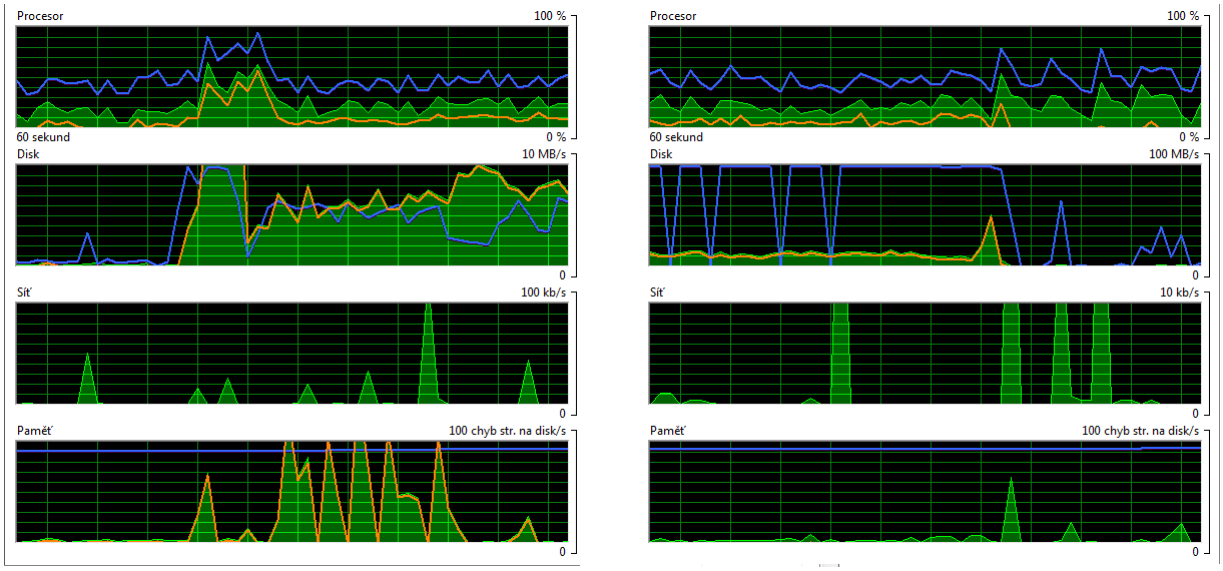
Při vyloučení nejlepšího a nejhoršího výsledku a následné zprůměrování zbylých hodnot vychází celkový výsledek:

7,30075 s na zpracování dotazu
0,24335 ms na zpracování jednoho řádku.

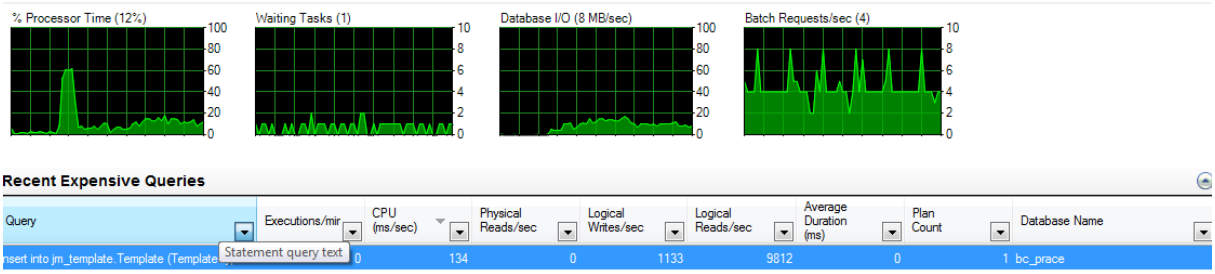
Pro srovnání jsem stejný dotaz pustil v Management Studio, kde vyšly výsledky zpracování dotazu 58, 39, 44, 38 a 39 sekund. Při stejném postupu zpracování vychází výsledek 40,66666 s na zpracování dotazu a 1,35555 ms na zpracování jednoho řádku. Z toho vyplývá, že zpracování dotazu v Management Studiu je mnohem pomalejší. Je to způsobeno grafickým zobrazení výsledku

2. Vložení 30 000 řádků do tabulky `jm_template.Template` pomocí dotazu spuštěném v Management Studiu:

```
insert into bc_prace.jm_template.Template
(TemplateTypeID,TemplateFile,TemplateDisabled)
select top 30000 TemplateTypeID,TemplateFile,TemplateDisabled
from bc_prace.jm_template.Template order by TemplateID asc
```



Obr. 9.1: Graf využití disku, paměti, procesoru a sítě na začátku a na konci spuštěného dotazu



Obr. 9.2: Nástroj Activity monitor při spuštění dotazu na lokálním serveru

Na obr. 9.1 pozorujeme chování hardwarových komponentů počítače a jejich zatížení na lokálním serveru při spuštění a dokončení dotazu, který slouží k vložení záznamů do databáze. V okně procesor modrá křivka značí nejvyšší frekvenci, zelená využití procesoru a oranžová využití procesoru programem MS SQL. V okně Disk vyznačuje modrá křivka nejvyšší aktivní čas, zelená využití disku a

oranžová využití disku MS SQL. V okně Síť zelená značí využití sítě a oranžová využití sítě MS SQL. V okně Paměť modrá značí využitou fyzickou paměť, zelená chybovost a oranžová chybovost způsobenou MS SQL.

Na obr. 9.2 pozorujeme vytížení procesoru, počet úloh čekající na zpracování, příchozí a odchozí data z databáze a dávkové požadavky lokálního serveru při spuštění dotazu, který slouží k vložení záznamů do databáze. Při spuštění dotazu zaznamenáme nárůst požadavků na hardware a tím i jeho větší vytížení a naopak při dokončení dotazu pokles vytížení.

Měření prováděno při využití procesoru 7% a využití paměti 4096/2770 Mb.

Výsledky:
114, 84, 60, 132 a 112 sekund

Při vyloučení nejlepšího a nejhoršího výsledku a následné zprůměrování zbylých hodnot vychází celkový výsledek:
103,33333 s na zpracování dotazu
3,44444 ms na zpracování jednoho řádku.

3. Smazání 30 000 řádků z tabulky `jm_template.Template` pomocí dotazu spuštěném v Management Studiu:

```
delete from bc_prace.jm_template.Template where TemplateID in (
select top 30000 TemplateID from bc_prace.jm_template.Template order by
TemplateID desc)
```

Měření prováděno při využití procesoru 7% a využití paměti 4096/2800 Mb.

Výsledky:
26, 41, 51, 39 a 33 sekund.

Při vyloučení nejlepšího a nejhoršího výsledku a následné zprůměrování zbylých hodnot vychází celkový výsledek:
37,66666 s na zpracování dotazu
1,25555 ms na zpracování jednoho řádku.

Microsoft SQL Server 2012 Express Edition

1. V programu dodaného vedoucím bakalářské práce jsem pustil dotaz na zobrazení 30 000 řádků z tabulky `jm_template.Template`:

```
select top 30000 * from dbguard01.jm_template.Template
order by TemplateID asc
```

Měření prováděno při využití procesoru 8% a využití paměti 4096/2300 Mb.

Výsledky:

Query time is 00:00:09.7865598
Total rows count 30000
Time for 1 row (ms) 0,32621866

Query time is 00:00:07.3664213
Total rows count 30000
Time for 1 row (ms) 0,245547376666667

Query time is 00:00:06.1423513
Total rows count 30000
Time for 1 row (ms) 0,204745043333333

Query time is 00:00:06.4343680
Total rows count 30000
Time for 1 row (ms) 0,214478933333333

Query time is 00:00:06.2573579
Total rows count 30000
Time for 1 row (ms) 0,208578596666667

Při vyloučení nejlepšího a nejhoršího výsledku a následné zprůměrování zbylých hodnot vychází celkový výsledek:
6,68604 s na zpracování dotazu
0,22286 ms na zpracování jednoho řádku.

2. Vložení 30 000 řádků do tabulky `jm_template.Template` pomocí dotazu spuštěném v Management Studiu:

```
insert into dbguard01.jm_template.Template
(TemplateTypeID,TemplateFile,TemplateDisabled)
select top 30000 TemplateTypeID,TemplateFile,TemplateDisabled
from dbguard01.jm_template.Template order by TemplateID asc
```

Měření prováděno při využití procesoru 8% a využití paměti 4096/2178 Mb.

Výsledky:
150, 116, 133, 120 a 133 sekund

Při vyloučení nejlepšího a nejhoršího výsledku a následné zprůměrování zbylých hodnot vychází celkový výsledek:
128,66666 s na zpracování dotazu
4,28888 ms na zpracování jednoho řádku

3. Smazání 30 000 řádků z tabulky `jm_template.Template` pomocí dotazu spuštěném v Management Studiu:

```
delete from dbguard01.jm_template.Template where TemplateID in (
select top 30000 TemplateID from dbguard01.jm_template.Template order by
TemplateID desc)
```

Měření prováděno při využití procesoru 7% a využití paměti 4096/2172 Mb.

Výsledky:
40, 37, 32, 38 a 40 sekund

Při vyloučení nejlepšího a nejhoršího výsledku a následné zprůměrování zbylých hodnot vychází celkový výsledek:

38,33333 s na zpracování dotazu
1,27777 ms na zpracování jednoho řádku.

9.2.2 Vzdálený přístup

Microsoft SQL Server 2008 R2

1. V programu dodaného vedoucím bakalářské práce jsem pustil dotaz na zobrazení 30 000 řádků z tabulky `jm_template.Template`:

```
select top 30000 * from dbguard01.jm_template.Template
order by TemplateID asc
```

Měření prováděno při využití procesoru 17% a využití paměti 4096/1748 Mb.

Výsledky:

Query time is 00:00:49.6618405
Total rows count 30000
Time for 1 row (ms) 1,65539468333333

Query time is 00:00:47.7157292
Total rows count 30000
Time for 1 row (ms) 1,59052430666667

Query time is 00:00:47.6987282
Total rows count 30000
Time for 1 row (ms) 1,58995760666667

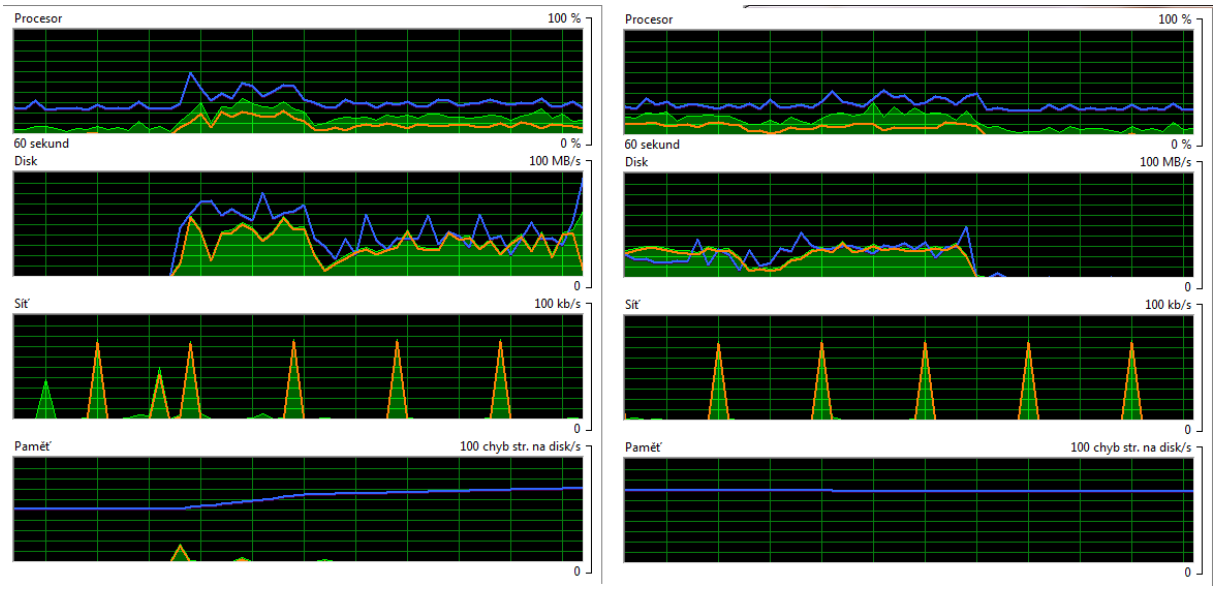
Query time is 00:00:47.6437250
Total rows count 30000
Time for 1 row (ms) 1,58812416666667

Query time is 00:00:48.3937680
Total rows count 30000
Time for 1 row (ms) 1,6131256

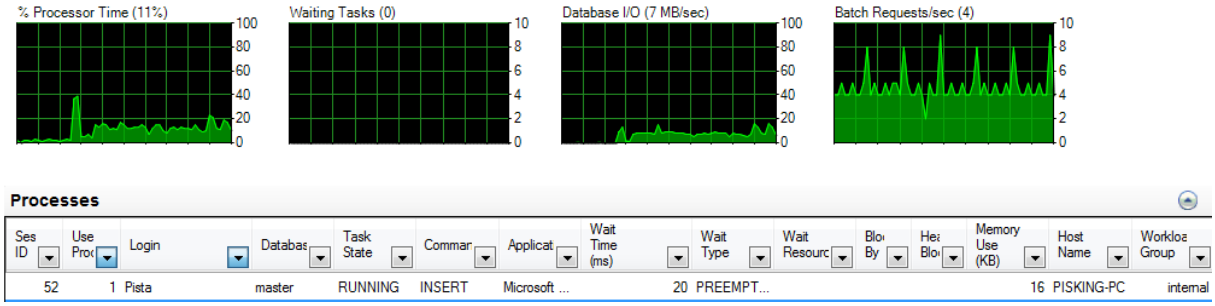
Při vyloučení nejlepšího a nejhoršího výsledku a následné zprůměrování zbylých hodnot vychází celkový výsledek:
47,93606 s na zpracování dotazu
1,57869 ms na zpracování jednoho řádku.

2. Vložení 30 000 řádků do tabulky `jm_template.Template` pomocí dotazu spuštěném v Management Studiu:

```
insert into dbguard01.jm_template.Template
(TemplateTypeID,TemplateFile,TemplateDisabled)
select top 30000 TemplateTypeID,TemplateFile,TemplateDisabled
from dbguard01.jm_template.Template order by TemplateID asc
```



Obr. 9.3: Graf využití disku, paměti, procesoru a sítě na začátku a na konci spuštěného dotazu vzdáleného serveru



Obr. 9.4.: Nástroj Activity monitor při spuštění dotazu na vzdáleném serveru

Na obr. 9.3 pozorujeme chování hardwarových komponentů počítače a jejich zatížení na vzdáleném serveru při spuštění a dokončení dotazu, který slouží k vložení záznamů do databáze. V okně procesor modrá křivka značí nejvyšší frekvenci, zelená využití procesoru a oranžová využití procesoru programem MS SQL. V okně Disk vyznačuje modrá křivka nejvyšší aktivní čas, zelená využití disku a oranžová využití disku MS SQL. V okně Síť zelená značí využití sítě a oranžová využití sítě MS SQL. V okně Paměť modrá značí využitou fyzickou paměť, zelená chybovost a oranžová chybovost způsobenou MS SQL.

Na obr. 9.4 pozorujeme vytížení procesoru, počet úloh čekající na zpracování, příchozí a odchozí data z databáze a dávkové požadavky vzdáleného serveru při spuštění dotazu, který slouží k vložení záznamů do databáze. Můžeme tak pozorovat, že při spuštění dotazu zaznamenáme nárůst požadavků na hardware a tím i jeho větší vytížení a naopak při dokončení dotazu pokles vytížení.

Měření prováděno při využití procesoru 15% a využití paměti 4096/2393 Mb.

Výsledky:
85, 58, 55, 53 a 65 sekund

Při vyloučení nejlepšího a nejhoršího výsledku a následné zprůměrování zbylých hodnot vychází celkový výsledek:

59,33333 s na zpracování dotazu
1,97777 ms na zpracování jednoho řádku.

3. Smazání 30 000 řádků z tabulky `jm_template.Template` pomocí dotazu spuštěném v Management Studiu:

```
delete from dbguard01.jm_template.Template where TemplateID in (
select top 30000 TemplateID from dbguard01.jm_template.Template order by
TemplateID desc)
```

Měření prováděno při využití procesoru 18% a využití paměti 4096/2283 Mb.

Výsledky:
21, 20, 17, 25 a 21 sekund

Při vyloučení nejlepšího a nejhoršího výsledku a následné zprůměrování zbylých hodnot vychází celkový výsledek:
20,66666 s na zpracování dotazu
0,68888 ms na zpracování jednoho řádku.

9.3 Operace nad tabulkou jm_template.Template_test

9.3.1 Lokální přístup

Microsoft SQL Server 2008 R2

1. V programu dodaného vedoucím bakalářské práce jsem pustil dotaz na zobrazení 30 000 řádků z tabulky `jm_template.Template_test`:

```
select top 30000 * from bc_prace.jm_template.Template_test
order by TemplateID asc
```

Měření prováděno při využití procesoru 7% a využití paměti 4096/1890 Mb.

Výsledky:

Query time is 00:00:28.1706113
Total rows count 30000
Time for 1 row (ms) 0,939020376666667

Query time is 00:00:04.3792504
Total rows count 30000
Time for 1 row (ms) 0,145975013333333

Query time is 00:00:03.4841993
Total rows count 30000
Time for 1 row (ms) 0,116139976666667

Query time is 00:00:03.7852165
Total rows count 30000
Time for 1 row (ms) 0,126173883333333

Query time is 00:00:03.1211785
Total rows count 30000
Time for 1 row (ms) 0,104039283333333

Při vyloučení nejlepšího a nejhoršího výsledku a následné zprůměrování zbylých hodnot vychází celkový výsledek:

3,88288 s na zpracování dotazu
0,12942 ms na zpracování jednoho řádku.

2. Vložení 30 000 řádků do tabulky `jm_template.Template_test` pomocí dotazu spuštěném v Management Studiu:

```
insert into bc_prace.jm_template.Template_test
(TemplateTypeID,TemplateFile,TemplateDisabled)
select top 30000 TemplateTypeID,TemplateFile,TemplateDisabled
from bc_prace.jm_template.Template_test order by TemplateID asc
```

Měření prováděno při využití procesoru 7% a využití paměti 4096/2875 Mb.

Výsledky:
32, 66, 54, 53 a 64 sekund

Při vyloučení nejlepšího a nejhoršího výsledku a následné zprůměrování zbylých hodnot vychází celkový výsledek:
57 s na zpracování dotazu
1,9 ms na zpracování jednoho řádku.

3. Smazání 30 000 řádků z tabulky `jm_template.Template_test` pomocí dotazu spuštěném v Management Studiu:

```
delete from bc_prace.jm_template.Template_test where TemplateID in (
select top 30000 TemplateID from bc_prace.jm_template.Template_test order
by TemplateID desc)
```

Měření prováděno při využití procesoru 7% a využití paměti 4096/2900 Mb.

Výsledky:
33, 59, 39, 32 a 24 sekund

Při vyloučení nejlepšího a nejhoršího výsledku a následné zprůměrování zbylých hodnot vychází celkový výsledek:
34,66666 s na zpracování dotazu
1,15555 ms na zpracování jednoho řádku.

Microsoft SQL Server 2012 Express Edition

1. V programu dodaného vedoucím bakalářské práce jsem pustil dotaz na zobrazení 30 000 řádků z tabulky `jm_template.Template_test`:

```
select top 30000 * from dbguard01.jm_template.Template_test
order by TemplateID asc
```

Měření prováděno při využití procesoru 17% a využití paměti 4096/2643 Mb.

Výsledek:

Query time is 00:00:03.9262245
Total rows count 30000
Time for 1 row (ms) 0,13087415

Query time is 00:00:06.3273619
Total rows count 30000
Time for 1 row (ms) 0,210912063333333

Query time is 00:00:05.0482888
Total rows count 30000
Time for 1 row (ms) 0,168276293333333

Query time is 00:00:04.1532376
Total rows count 30000
Time for 1 row (ms) 0,138441253333333

Query time is 00:00:05.3263047
Total rows count 30000
Time for 1 row (ms) 0,17754349

Při vyloučení nejlepšího a nejhoršího výsledku a následné zprůměrování zbylých hodnot vychází celkový výsledek:
4,84260 s na zpracování dotazu
0,16142 ms na zpracování jednoho řádku.

2. Vložení 30 000 řádků do tabulky `jm_template.Template_test` pomocí dotazu spuštěném v Management Studiu:

```
insert into dbguard01.jm_template.Template_test
(TemplateTypeID,TemplateFile,TemplateDisabled)
select top 30000 TemplateTypeID,TemplateFile,TemplateDisabled
from dbguard01.jm_template.Template_test order by TemplateID asc
```

Měření prováděno při využití procesoru 14% a využití paměti 4096/2508 Mb.

Výsledky:
110, 50, 74, 76 a 66 sekund

Při vyloučení nejlepšího a nejhoršího výsledku a následné zprůměrování zbylých hodnot vychází celkový výsledek:
72 s na zpracování dotazu
2,4 ms na zpracování jednoho řádku.

3. Smazání 30 000 řádků z tabulky `jm_template.Template_test` pomocí dotazu spuštěném v Management Studiu:

```
delete from dbguard01.jm_template.Template_test where TemplateID in (
select top 30000 TemplateID from dbguard01.jm_template.Template_test order
by TemplateID desc)
```

Měření prováděno při využití procesoru 7% a využití paměti 4096/2737 Mb.

Výsledky:
47, 37, 38, 32 a 29 sekund

Při vyloučení nejlepšího a nejhoršího výsledku a následné zprůměrování zbylých hodnot vychází celkový výsledek:

35,66666 s na zpracování dotazu
1,18888 ms na zpracování jednoho řádku.

9.3.2 Vzdálený přístup

Microsoft SQL Server 2008 R2

1. V programu dodaného vedoucím bakalářské práce jsem pustil dotaz na zobrazení 30 000 řádků z tabulky `jm_template.Template_test`:

```
select top 30000 * from dbguard01.jm_template.Template_test
order by TemplateID asc
```

Měření prováděno při využití procesoru 22% a využití paměti 4096/1583 Mb.

Výsledky:

Query time is 00:00:56.4732301
Total rows count 30000
Time for 1 row (ms) 1,88244100333333

Query time is 00:00:51.6829561
Total rows count 30000
Time for 1 row (ms) 1,72276520333333

Query time is 00:00:51.6899565
Total rows count 30000
Time for 1 row (ms) 1,72299855

Query time is 00:00:52.7000143
Total rows count 30000
Time for 1 row (ms) 1,75666714333333

Query time is 00:00:47.3637091
Total rows count 30000
Time for 1 row (ms) 1,57879030333333

Při vyloučení nejlepšího a nejhoršího výsledku a následné zprůměrování zbylých hodnot vychází celkový výsledek:
59,33333 s na zpracování dotazu
1,97777 ms na zpracování jednoho řádku.

2. Vložení 30 000 řádků do tabulky `jm_template.Template_test` pomocí dotazu spuštěném v Management Studiu:

```
insert into dbguard01.jm_template.Template_test
(TemplateTypeID,TemplateFile,TemplateDisabled)
select top 30000 TemplateTypeID,TemplateFile,TemplateDisabled
from dbguard01.jm_template.Template_test order by TemplateID asc
```

Měření prováděno při využití procesoru 12% a využití paměti 4096/2282 Mb.

Výsledky:
83, 58, 74, 58 a 40 sekund.

Při vyloučení nejlepšího a nejhoršího výsledku a následné zprůměrování zbylých hodnot vychází celkový výsledek:
63,33333 s na zpracování dotazu
2,11111 ms na zpracování jednoho řádku.

3. Smazání 30 000 řádků z tabulky `jm_template.Template_test` pomocí dotazu spuštěném v Management Studiu:

```
delete from dbguard01.jm_template.Template_test where TemplateID in (
select top 30000 TemplateID from dbguard01.jm_template.Template_test order
by TemplateID desc)
```

Měření prováděno při využití procesoru 10% a využití paměti 4096/2286 Mb.

Výsledky:
46, 22, 18, 36 a 12 sekund.

Při vyloučení nejlepšího a nejhoršího výsledku a následné zprůměrování zbylých hodnot vychází celkový výsledek:
25,33333 s na zpracování dotazu
0,84444 ms na zpracování jednoho řádku.

9.4 Vyhodnocení

Tab. 9.1: Porovnání verzí SQL Serveru

Dotazy spouštěny nad tabulkou *jm_template.Template*		
Dotaz v jazyce T-SQL	Microsoft SQL Server 2008 R2 30 000/1 řádek/využití RAM (s/ms/Mb)	Microsoft SQL Server 2012 Express Edition 30 000/1 řádek/využití RAM (s/ms/Mb)
Select	7,30075/0,24335 /2511	6,68604/0,22286 /2300
Insert	103,33333/3,44444 /2770	128,66666/4,28888 /2178
Delete	37,66666/1,25555/2800	38,33333/1,27777 /2172
Dotazy spouštěny nad tabulkou *jm_template.Template_test*		
Select	3,88288/0,12942 /1890	4,84260/0,16142 /2643
Insert	57/1,9 /2875	72/2,4 /2508
Delete	34,66666/1,15555/2900	35,66666/1,18888 /2737
Součet	**243,85028/8,12831/15746**	**286,19529/9,53984/14538**

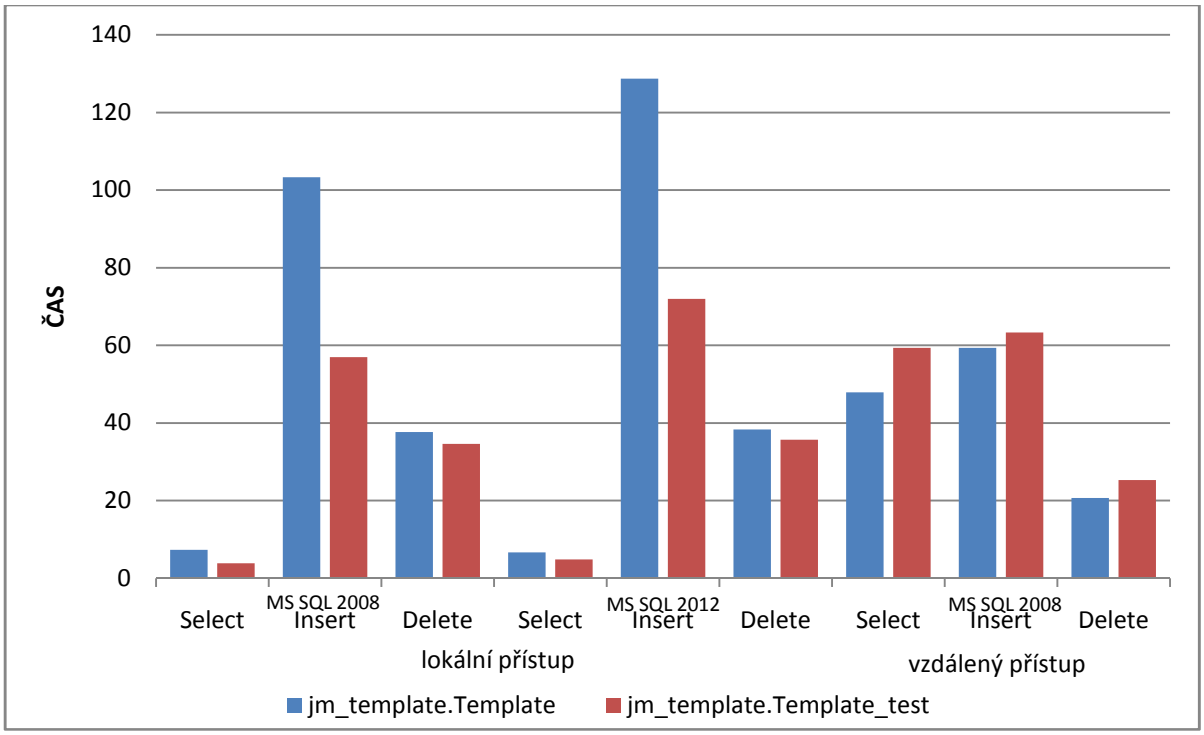
Podle dosažených výsledků je zřejmé, že rychlejší verze je Microsoft SQL Server 2008 R2. Může to být způsobeno tím, že Microsoft SQL Server 2012 Express Edition je na trhu krátce a možné zrychlení bude při další aktualizaci. Další možnost zpomalení může být způsobeno novými nástroji či funkcemi, které v MS SQL 2008 R2 nenaleznete, viz kapitola 7. Srovnání verzí Microsoft SQL Server. Jako další možnost omezení rychlosti zpracování dotazu jsou jednotlivé edice SQL Serveru. Zatímco verze SQL Serveru 2008 R2 je kompletní, edice SQL Server 2012 Express je značně „osekaná“ a chybí v ní mnoho funkcí.

Tab. 9.2: Porovnání přístupu k SQL Serveru

Dotazy spuštěné nad tabulkou *jm_template.Template*		
Dotaz v jazyce T-SQL	lokální SQL Server 30 000/1 řádek/využití RAM (s/ms/Mb)	vzdálený SQL Server 30 000/1 řádek/využití RAM (s/ms/Mb)
select	7,30075/0,24335/2511	47,93606/1,57869/1748
insert	103,33333/3,44444/2770	59,33333/1,97777/2393
delete	37,66666/1,25555/2800	20,66666/0,68888/2283
Dotazy spuštěné nad tabulkou *jm_template.Template_test*		
select	3,88288/0,12942 /1890	59,33333/1,97777/1583
insert	57/1,9 /2875	63,33333/2,11111/2282
delete	34,66666/1,15555/2900	25,33333/0,84444/2286
Součet	**243,85028/8,12831/15746**	**275,93604/9,19786/12575**

Měření bylo prováděno na stejné verzi SQL Serveru, SQL Server 2008 R2. Z dosažených výsledků vyplývá, že výhodnější je mít databázi na lokálním serveru. Při vzdáleném připojení na SQL Server prochází požadavky a odpovědi přes síť a z toho logicky vyplývá výrazné zpomalení.

Porovná datových typů Varchar (*jm_template.Template*) **a Varbinary** (*jm_template.Template_test*): Z dosažených výsledků měření vyplývá, že data v binární podobě, tedy Varbinary, jsou rychlejší na zpracování, než data v textové podobě, tedy Varchar. Ale z důvodu, že šablony jsou uloženy v podobě ASCII znaků, tedy textového řetězce, je výhodnější uchovávat data v datovém typu Varchar.
U Varbinary se totiž musí převést text do binární hodnoty a v případě potřeby se data převádí zpátky na text, zatímco u Varchar data nemusíme převádět, tudíž je to rychlejší a výhodnější.



Graf 9.1: Porovnání zpracování dotazu v tabulkách jm_template.Template a jm_template.Template_test na jednotlivých verzích MSSQL a přístupu k serveru

Složitost jednotlivých dotazů:
Dotaz na zobrazení záznamu z tabulky, select, je nejméně náročný na zpracování, což je způsobeno tím, že s tabulkou neprovádíme žádné operace „jenom“ zobrazujeme data. Naopak nejnáročnější na zpracování je dotaz na vložení záznamu do tabulky, insert. Zápis dat je mnohem náročnější operace než čtení či smazání dat.

10. Analýza dat zachycených paketovým analyzátorem

Ke komunikaci mezi SQL Serverem a SQL klientem používá MS SQL protokol TCP (Transmission Control Protocol). TCP je spojově orientovaný protokol pro přenos toku bajtů na transportní vrstvě se spolehlivým doručováním. TCP používá služby IP protokolu opakovaným odesíláním nespolehlivých paketů, při ztrátě paketu zajišťuje spolehlivost a neuspořádáváním přijatých paketů zajišťuje správné pořadí. Dále jsou používány dynamické porty, které jsou v rozsahu 49152 – 65535. V našem případě je nastaven dynamický port na serveru 53189. Zachycená data jsou zobrazeny v hexadecimálním a v ASCII formátu. Na monitorování komunikace mezi SQL Serverem a klientem použiji nástroj Wireshark, více o programu v příloze 1.

Protože TCP je spojovaná transportní služba, musí se před odesíláním dat navázat spojení mezi klientem a serverem. K tomu slouží trojcestný handshaking:

- Klient odešle na server paket s nastaveným příznakem SYN
- Server odešle klientovy paket s nastavenými příznaky SYN a ACK
- Klient odešle paket s nastaveným příznakem ACK

Obě strany si pamatují číslo sekvence své i protistrany. Používají se totiž i pro další komunikaci a určují pořadí paketů. Když úspěšně proběhne trojcestný handshaking, je spojení navázáno a odesílají se data. Data jednotlivých paketů jsou vyobrazeny v hexadecimálním a ASCII formátu.

Ukončení spojení probíhá podobně jako jeho navázání. Používá se k tomu příznaky FIN a ACK:

- Klient odešle paket s nastaveným příznakem FIN
- Server odpoví paketem s nastaveným příkazem ACK
- Server odešle paket s nastaveným příznakem FIN
- Klient odpoví s nastaveným příkazem ACK

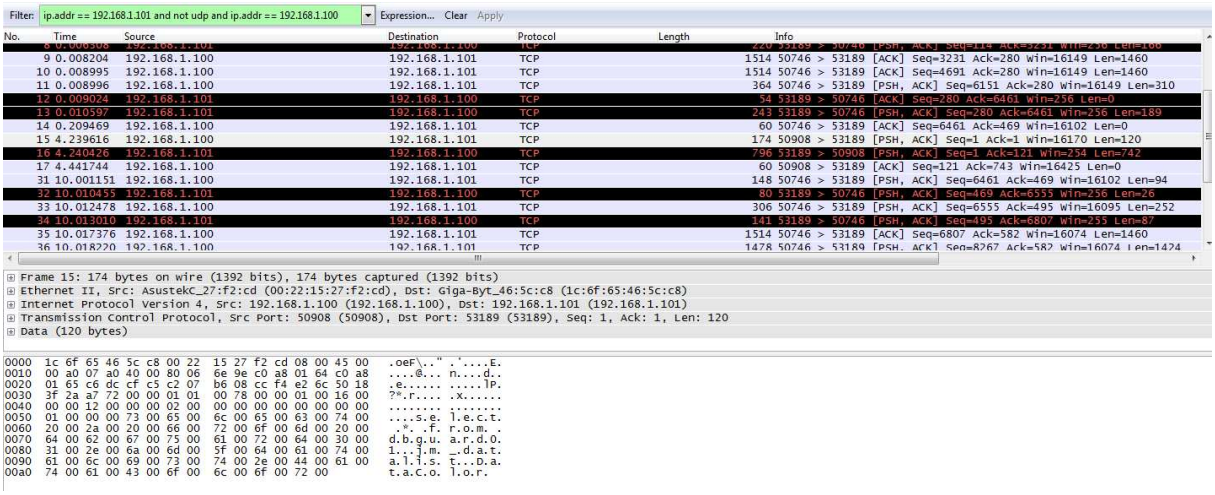
Teprve po těchto čtyřech krocích je spojení ukončeno.

10.1 Analýza

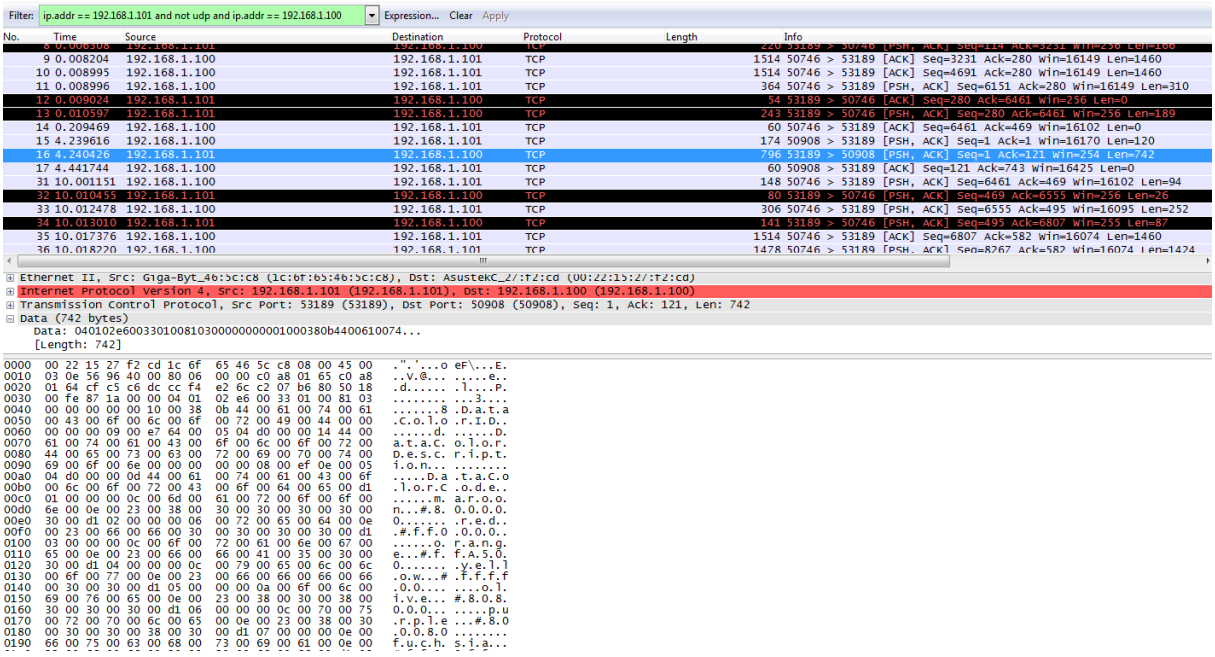
V Management Studiu, které je připojeno na instanci vzdáleného SQL Server jsem pustil dotaz:

```
select * from dbguard01.jm_datalist.DataColor
```

Komunikaci mezi serverem a klientem jsem zachytil ve Wiresharku viz obr. 10.1 a 10.2

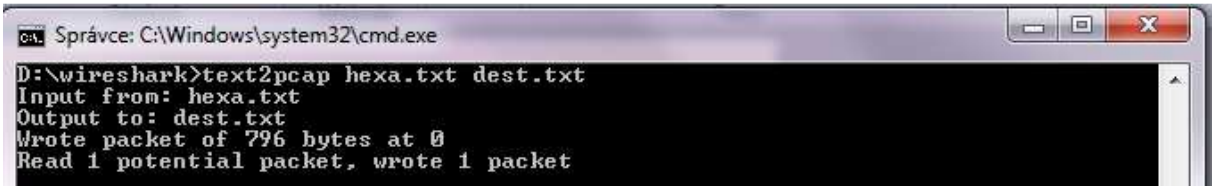


Obr. 10.1: Zachycení dotazu poslaného na vzdálený SQL server

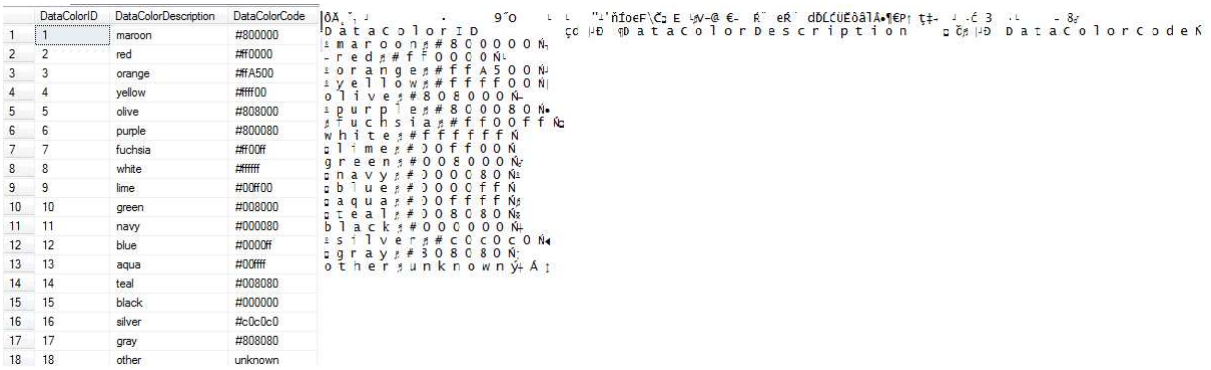


Obr. 10.2: Zachycení odpovědi na dotaz poslaného na vzdálený SQL server

Wireshark obsahuje podpůrný program Text2pcap, který čte data uložená v hexadecimálním a ASCII formátu a zapisuje je do výstupního souboru.



Obr. 10.3: Spuštění čtení dat pomocí podpůrného programu text2pcap

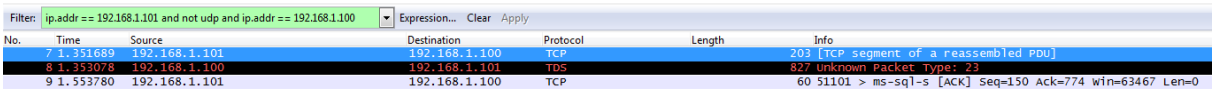


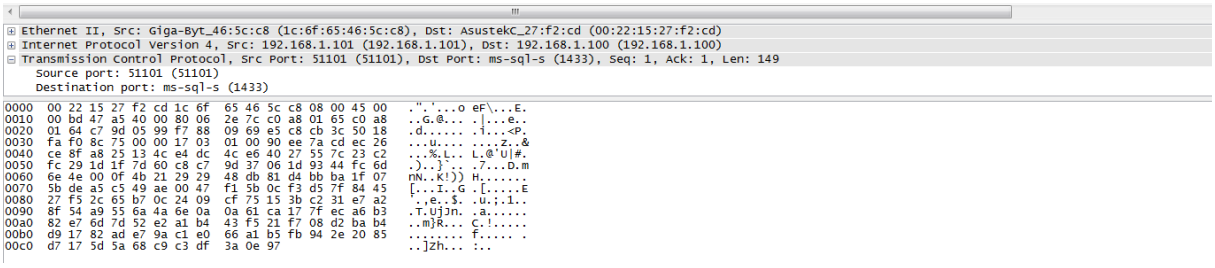
	DataColorID	DataColorDescription	DataColorCode
1	1	maroon	#800000
2	2	red	#ff0000
3	3	orange	#ffA500
4	4	yellow	#ffff00
5	5	olive	#808000
6	6	purple	#800080
7	7	fuchsia	#ff00ff
8	8	white	#ffffff
9	9	lime	#00ff00
10	10	green	#008000
11	11	navy	#000080
12	12	blue	#0000ff
13	13	aqua	#00ffff
14	14	teal	#008080
15	15	black	#000000
16	16	silver	#c0c0c0
17	17	gray	#808080
18	18	other	unknown

Obr. 10.4: Porovnání tabulek, zleva Management Studio, zprava výsledek programu Text2pcap

Při defaultním nastavení SQL Serveru jsem paketovým analyzátorem přečetl spuštěný dotaz i výsledek dotazu. Zamezení reprodukce dat do původní podoby zajistíme následujícím postupem:

1. otevřít nástroj konfigurační manažer (SQL Server configuration manager)
2. otevřít SQL Server Network Configuration
3. na vybraný server pravým tlačítkem myši a vlastnosti
4. v záložce Flags změnit Force Encryption na YES a použít
5. v posledním kroku restartovan SQL Server





Obr. 10.5: Zachycení dotazu poslaného na vzdálený SQL server při nastaveném kódování

10.2 Zhodnocení bezpečnosti přenášených dat

Při defaultním nastavení SQL Server lze pomocí paketového analyzátoru přečíst jak požadavek, tak i odpověď zaslanou na SQL Server. Z pohledu firmy/osoby, která vlastní citlivé data, je velice důležité, aby měli kódování nastaveno. Z tohoto důvodu doporučuji hned po instalaci SQL Server zapnout kódování, aby nedocházelo k úniku informací.

11. Návrh optimalizace

V databázi jsem díky předchozím testům objevil několik bezpečnostních hrozeb. Jednotlivé hrozby v této kapitole odstraním a navrhnu nejoptimálnější řešení databáze. Databáze „bc_prace“ obsahuje

informace o osobách detekovaných systémem CCTV a o hledaných osobách. Tyto záznamy jsou na základě uložených šablon porovnávány a je tak možné identifikovat hledané osoby, které se vyskytují ve videozáznamech připojených systémů CCTV. Optimalizace zohledňuje fakt, že poměr zápis/čtení je v rámci databáze cca 1:1. To znamená, že výrazně nepřevažuje čtení nad zápisem a ani zápis nad čtením.

V tabulkách *jm_template.Template* a *jm_template.Template_test* jsou stejné data, je vhodné vybrat nejlepší variantu a jednu tabulku odstranit, čímž výrazně snížíme velikost databáze. Tabulka *jm_template.Template_test* je sice rychlejší na zpracování, ale z důvodu, že jsou data v binární podobě, a tudíž se data musí převádět, ji odstraním následujícím dotazem:

```
drop table bc_prace.jm_template.Template_test
```

Pro zrychlení zpracování dotazů nad tabulkou Template, se nabízí použití indexů nad nejnáročnějším sloupcem TemplateFile, jenomže při použití full-textového vyhledávání [3]to mělo opačný efekt. Zaznamenal jsem výrazné zpomalení, řádově desítek minut. Další možnost pro zrychlení výpočtu náročného sloupce je použití nonclustered index, který se používá pro zrychlení vyhledávání dat za podmínky WHERE, ORDER BY, GROUP BY. Tento index má omezenou délku sloupce a tudíž jsem ji nemohl použít, a tak k zrychlení tabulky Template způsobilo pouze smazání tabulky Template_test.

11.1 Bezpečnost

V první řadě, hned po instalaci SQL Serveru, zapnout Force Encryption v konfiguračním manažeru, aby jednotlivé pakety zasílané mezi SQL Serverem a klientem nešly reprodukovat do původní podoby, a tím zabránit úniku informací. Jako další bezpečnostní opatření proti úniku informací doporučuji zapnout Transparentní šifrování dat. Transparentní šifrování dat šifruje každou stránku vaší celé databáze a automaticky dle potřeby dešifruje každou stránku během přístupu. Tato funkce vám umožňuje zabezpečit celou databázi, aniž byste se starali o podrobnosti zašifrování na úrovni sloupců.

Postup k zapnutí transparentního šifrování dat:

1. Pokud neexistuje, tak vytvořit hlavní klíč databáze (DMK). Ujistěte se, že hlavní klíč databáze, je šifrován pomocí služby hlavního klíče (SMK).

CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'some password';

2. Pro co nejlepší zabezpečení, je doporučeno vytvořit nový certifikát, jehož jedinou funkcí je chránit hlavní klíč databáze (DEK)

CREATE CERTIFICATE tdeCert WITH SUBJECT = 'TDE Certificate';

3. Vytvořte zálohu certifikátu se soukromým klíčem a uložte jej na bezpečném místě.

BACKUP CERTIFICATE tdeCert TO FILE = 'path_to_file'

WITH PRIVATE KEY (

[3] full-textové vyhledávání - funkce SQL Serveru, která umožňuje rychlé a efektivní dotazování velkého množství nestrukturovaných dat.

FILE = ‘path_to_private_key_file’,

ENCRYPTION BY PASSWORD = ‘cert password’);

4. Vytvoření databázového klíče (DEK) šifrovaného pomocí certifikátu určeného v kroku 2.

```
CREATE DATABASE ENCRYPTION KEY

  WITH ALGORITHM = AES_256

  ENCRYPTION BY SERVER CERTIFICATE tdeCert
```

5. Zapnout transparentní šifrování dat

```
ALTER DATABASE myDatabase SET ENCRYPTION ON
```

K bezpečnosti databáze samozřejmě také patří oprávnění pro uživatele. Nastavení oprávnění pro jednotlivé uživatele záleží na správci databáze a v jaké míře chce uživateli povolit oprávnění. V našem případě jsem správcem, vlastníkem databáze já, tudíž mám nejvyšší oprávnění sysadmin.

11.2 Umístění

Podle dosažených výsledků shrnutých v kapitole 9, je při mých hardwarových možnostech serveru a klienta, nejlepší řešení použít lokální přístup k SQL Serveru. Použití MS SQL Serveru 2008 R2 je v našem případě nejvhodnější volba. Dále doporučuji použít edici Enterprise, která má oprávnění na všechny metody šifrování.

Závěr

V teoretické části jsem nastínil problematiku bezpečnosti databází Microsoft SQL. Zejména zabezpečení, šifrování, používané kryptografické algoritmy, módy ověřování, vysoká dostupnost a obnovení po havárii. Dále jsem porovnal jednotlivé verze MS SQL a nástroj MS SQL Management Studio, které je pro práci s databázemi klíčovým prvkem. V poslední řadě jsem rozebral možnosti sledování a ladění výkonu.

V praktické části jsem se po instalaci MS SQL zabýval měřením výkonu při spuštění dotazů jazyka Transact SQL různých složitostí nad tabulkami obsahující stejné data, ale v rozdílném datovém typu, a v poslední řadě na vybraných verzí MS SQL. Z výsledků měření jsem vyvodil závěr a určil nejvhodnější variantu zpracování dat. V další části jsem paketovým analyzátorem zachytil komunikaci mezi SQL Serverem a klientem při zapnutém i vypnutém kódování.

Nejefektivnější verze je Microsoft SQL 2008 R2 nainstalovaná na lokálním počítači, kde šablony tabulky jm_template.template jsou v datovém typu varchar.

Použitá literatura

[1] WALTERS, E. Robert. Mistrovství v Microsoft SQL server 2008. 1. vyd. Brno: Computer Press, 2009. 864 s. ISBN 978-80-251-2329-4.

[2] Internetové stránky technické podpory firmy Microsoft. Authentication Modes [online], [cit. 21. 4. 2012]
URL: <http://msdn.microsoft.com/en-us/library/aa905171(v=sql.80).aspx>

[3] Internetové stránky technické podpory firmy Microsoft. Microsoft Kerberos [online], [cit. 27. 5. 2012]
URL: <http://msdn.microsoft.com/en-us/library/windows/desktop/aa378747(v=vs.85).aspx>

[4] Internetové stránky technické podpory firmy Microsoft. Nástroje pro výkon pro sledování a ladění [online], [cit. 17. 5. 2012]
URL: <http://technet.microsoft.com/cs-cz/library/ms179428(v=sql.100).aspx>

[5] Internetové stránky technické podpory firmy Microsoft. Představujeme SQL Server Management Studio [online], [cit. 27. 5. 2012]
URL: <http://technet.microsoft.com/cs-cz/library/ms174173(v=sql.100)>

[6] Internetové stránky technické podpory firmy Microsoft. Microsoft SQL Server 2000 [online], [cit. 29. 5. 2012]
URL: < http://msdn.microsoft.com/en-us/library/ms950404>

[7] Internetové stránky technické podpory firmy Microsoft. Edice Microsoft SQL Server 2000 [online], [cit. 29. 5. 2012]
URL: < http://msdn.microsoft.com/en-us/library/aa933150(v=sql.80).aspx>

[8] Internetové stránky technické podpory firmy Microsoft. Co je nového v SQL Server 2005 [online], [cit. 29. 5. 2012]
URL: < http://technet.microsoft.com/en-US/library/ms170363(v=sql.90).aspx>

[9] Internetové stránky technické podpory firmy Microsoft. Vlastnosti a nástroje SQL Server 2008 [online], [cit. 29. 5. 2012]
URL: < http://msdn.microsoft.com/en-us/library/bb500397(v=sql.100)>

[10] Internetové stránky technické podpory firmy Microsoft. Knihy online pro SQL Server 2012 [online], [cit. 29. 5. 2012]
URL: < http://technet.microsoft.com/en-us/library/ms130214.aspx>

[11] Internetové stránky produktu Wireshark. About Wireshark [online], [cit. 27. 5. 2012]
URL: <http://www.wireshark.org/about.html>

Seznam použitých zkratek, veličin a symbolů

ACK	Acknowledge
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
CAS	Code Access Security
CCTV	Closed Circuit Television
CLR	Common Language Runtime

CryptoAPI	Cryptographic API
DBCC	Database Console Commands
DDL	Data Definition Language
DETA	Database Engine Tuning Advisor
DMK	Database Master Key
DPAPI	Windows Data Protection API
DVD	Digital Video Disc
EKM	Extensible Key Managment
FIN	Finish
HSM	Hardware Security Module
IP	Internet Protocol
KDC	Key Distribution Center
MS SQL	Microsoft SQL
OLTP	Online Transaction Processing
SMK	Service Master Key
SQL	Structured Query Language
SYN	Synchronization
TCP	Transmission Control Protocol
TGS	Ticket Granting Service
TGT	Ticket Granting Ticket
WMI	Windows Management Instrumentation
XML	eXtensible Markup Languag

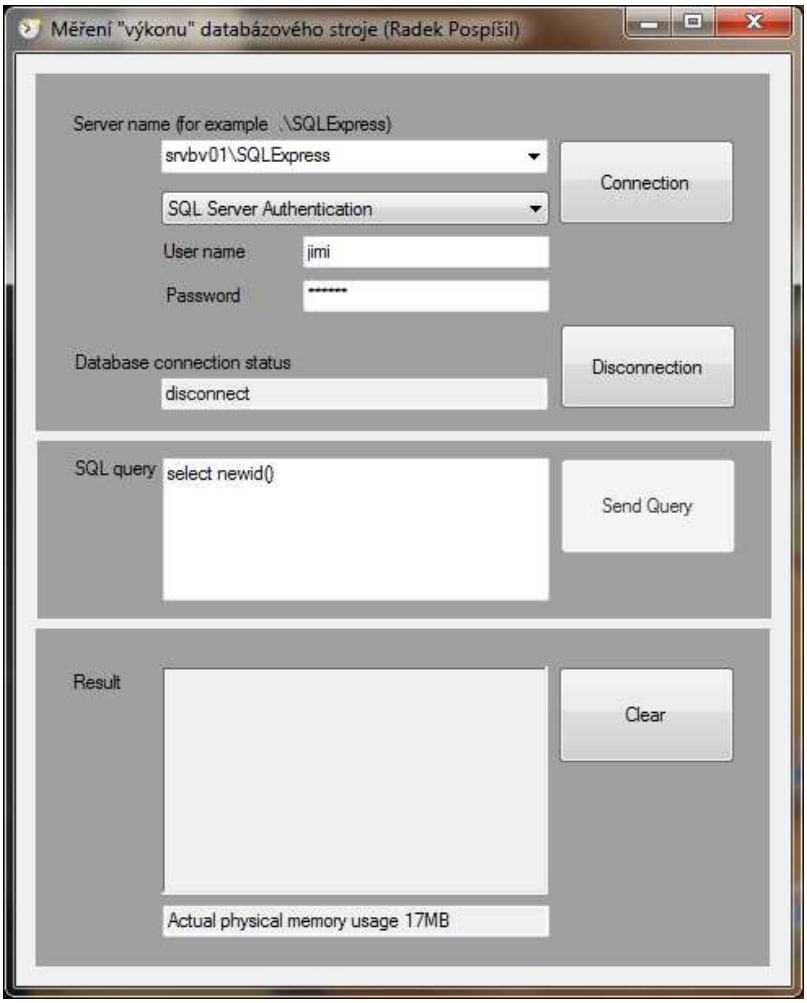
Seznam příloh

A. Program dodaný vedoucím bakalářské práce

Napsaný v c# (MS Visual Studio 2010 Professional) a slouží k měření toho, jak dlouho trvá sada databázových dotazů uvedených v "SQL query“. V případě dotazu SELECT se chová utilita takto:

- Předá dotaz SQL serveru a odpověď (přijatá data) ukládá do operační paměti (respektive do tabulky DataTable datatable = new DataTable();).
- Po ukončení plnění tabulky je vypsán čas (spolu s počtem záznamů), který byl pro vykonání úkonu třeba. Takto je získán "surový" čas, není ovlivněn zápisem do GUI apod.

Protože může být v rámci dotazu přijato velké množství dat, je aplikace 64bitová a odpadá tak omezení alokovatelného množství paměti. V aplikaci běží GUI v hlavním vlákně a příkazy jsou spouštěny v samostatných vláknech Thread t = new Thread(delegate() { dataset(); });
K databázi je možné připojit se prostřednictvím Windows Authentication (Integrated Security=SSPI) i SQL Server Authentication.



Obr. A.1: Náhled programu dodaného vedoucím bakalářské práce

B. Wireshark

Wireshark je síťový analyzátor. Umí číst pakety ze sítě, dekódovat je a zobrazit ve srozumitelném formátu. Jednou z důležitých vlastností Wiresharku je skutečnost, že je distribuován jako open-source,a tedy zdarma. Následuje výpis některých dalších vlastností tohoto programu: [11]

- Wireshark je distribuován zdarma pod open-source licencí Gnu's not UNIX (GNU) General Public License (GPL).
- Pracuje v promiskuitním i nepromiskuitním módu.
- Může zachytávat data ze sítě nebo je číst ze souboru.
- Má srozumitelné a konfigurovatelné rozhraní.
- Má bohaté možnosti nastavení zobrazovacích filtrů.
- Podporuje formát souborů dat zachycených programem tcpdump. Také má nástroje pro rekonstrukci relace protokolu TCP a dokáže je zobrazit v kódu American Standard Code for Information Interchange (ASCII), Extended Binary Coded Decimal Interchange Code (EBCDIC), dále pak v hexadecimálním tvaru nebo ve formátu pole jazyka C.
- Je dostupný v podobě předkompilovaných binárních souborů i v podobě zdrojového kódu.
- Pracuje na více než 20 platformách, včetně operačních systémů založených na Uniplexed Information and Computing Systém (UNIX), Windows a od jiných dodavatelů jsou dostupné i instalační balíky pro operační systém Mac OS X.
- Podporuje více než 750 protokolů, a protože je open-source, nové protokoly jsou komunitou přidávány poměrně často.

- Dokáže číst data zachycená více než 25 odlišnými programy.
- Může ukládat zachycená data v různých formátech (jako libpcap, Network Associated Sniffer, Microsoft Network Monitor (NetMon) a snoop z operačního systému Sun).
- Umí zachytávat data z různých přenosových médií (jako Ethernet, Token-Ring, bezdrátové protokoly 802.11 a dalších).
- Obsahuje verzi programu ovládanou z příkazové řádky, nazvanou *tshark.*
- Obsahuje další podpůrné programy jako *editcap*, *mergecap* a *text2pcap*.
- Výstup může být uložen nebo vytištěn jako prostý text nebo PostScript.